SMART INTERFACE BOX FOR SOLAR PANELS

Design Document

October 9, 2019

ECE 445 (Fall 2019)

Team #3

Dillon Vadgama (dvadga2)

Douglas Lee (dlee242)

Sachin Reddy (ssreddy2)

0.1 Introduction

1. Problem and Solution Overview:

In 2018, a fire broke out on the roof of a Walmart in Beavercreek, Ohio due to Tesla's unmaintained solar panels suffering from hot-spots [6]. These hot-spots resulted in the cracking of the back sheets of the solar modules and compromised their electrical insulation. There was no protection system to detect this type of unwanted behavior and shut down the system before a fire broke out. As a result, Walmart sued Tesla over the flaws present in their solar panels and an overall lack of system protection.

The Electrical and Computer Engineering building (ECEB) at the University of Illinois at Urbana-Champaign has a roof of 60 solar panels that are used for research; however, there are no protection interfaces between the solar panels and the load they are delivering power to. As a result, these solar panels are at risk for suffering the same type of failure as the one stated above.

A smart interface box attached to each solar panel that has the ability to shut off the entire operation in the event certain failure conditions (over-current, over-voltage, and overheating) are met and allows users to remotely monitor system behavior/parameters could help 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 would be particularly useful to researchers.

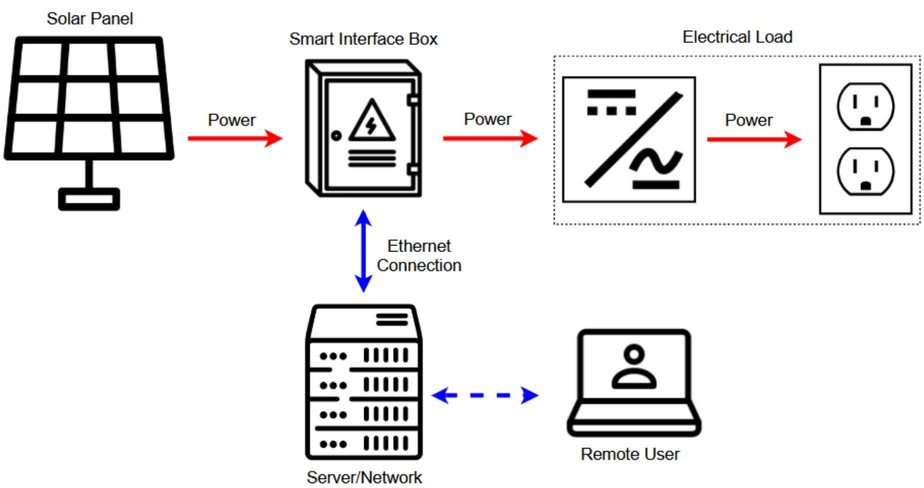


Figure 1: Visual aid.

2. High Level Requirements:

- The output of the interface box will be configurable to be connected to either 32 cells, 64 cells, or 128 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).

0.2 Design

1. Block Diagram

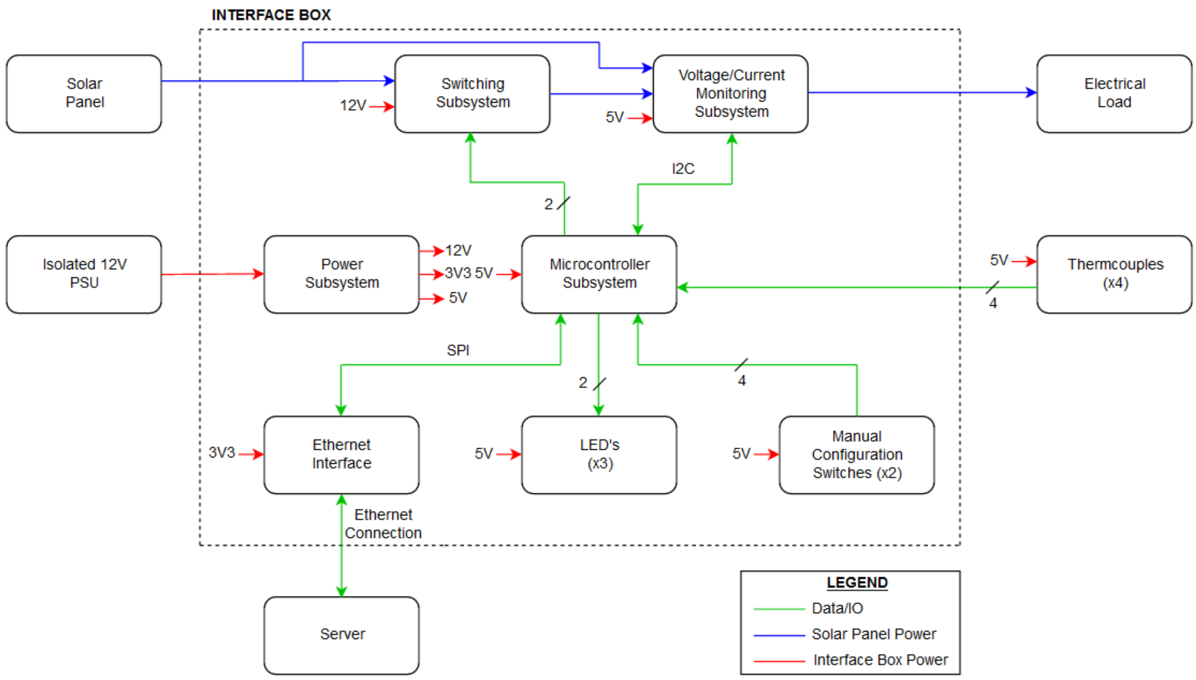


Figure 2: System Block Diagram

The interface box for the solar panel will consist of seven important subsystems: the Voltage/Current Monitoring Subsystem, Ethernet Interface, Microcontroller/Processing Subsystem, Power Subsystem, Switching Subsystem, LED's, and Manual Configuration Switches.

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.

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 this subsystem 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 when requested by the Microcontroller Subsystem.

Thermocouples that will be mounted directly on the solar panel will feed temperature data directly into 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 the Ethernet Interface is not being utilized, users can control the output configuration and toggle the output via the Manual Configuration Switches which will be mounted on the interface box.

LED's aid in manual configuration by indicating interface box power, whether the output is enabled, and whether the Ethernet interface is being used.

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.

The correct implementation of each of these blocks will ensure that all of the high-level requirements are met.

2. Mechanical Design

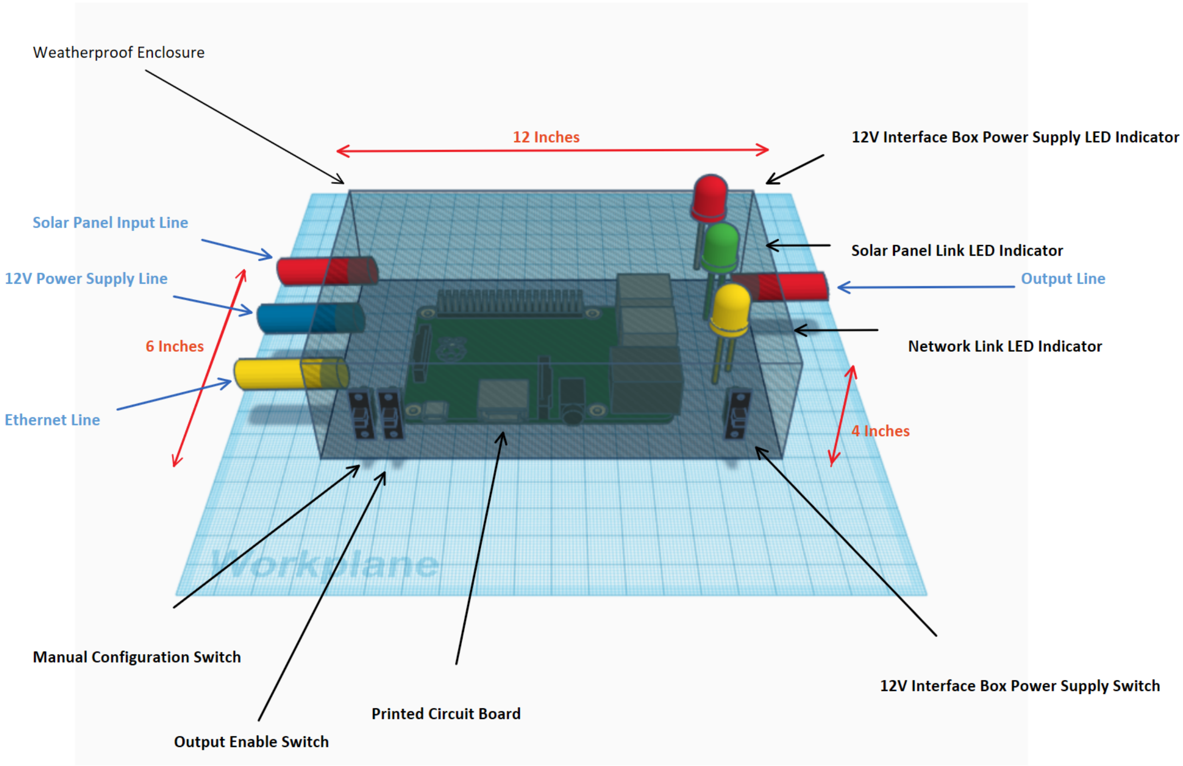


Figure 3: Weatherproof Enclosure for Interface Box

Our design does not require any extensive mechanical components, but it does require a weatherproof enclosure to protect it from outside elements (e.g. rain and water). The PCB will be housed in a 12 x 6 x 4 (inches) weatherproof enclosure as shown in Figure 3.

3. Block Descriptions

(a) Solar Panels

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 is designed for. Table 1 shows important specifications [2] that were used throughout the design process.

As a part of requirements laid out above, 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

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

the structure of the solar panel, 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. There are also bypass diodes between each of these partitions to ensure none of the partitions are consuming power if they are shaded. For the purposes of the design, this information was used to create a voltage source model of the panel as shown in Figure 4. 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, we can calculate the expected voltages of the sources: $V_{32} = 21.4$ V and $V_{64} = 42.8$ V.

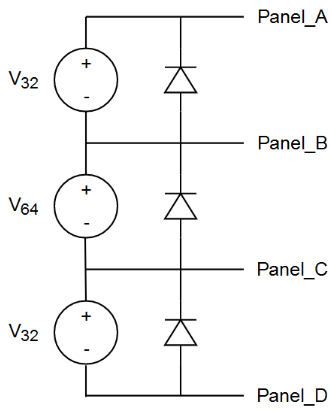


Figure 4: Solar Panel Equivalent Circuit

Knowing that the output of the interface box must be connected to either 32 cells, 64 cells, or 128 cells, the possible output configurations chosen to satisfy this requirement 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.

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.

(b) Switching Subsystem

After defining the four required output configurations of the interface box above, we can discuss the implementation of this in the Switching Subsystem. Figure 5 shows the schematic of the Switching System. The primary switching components responsible for setting the output configuration are two double pole double 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.

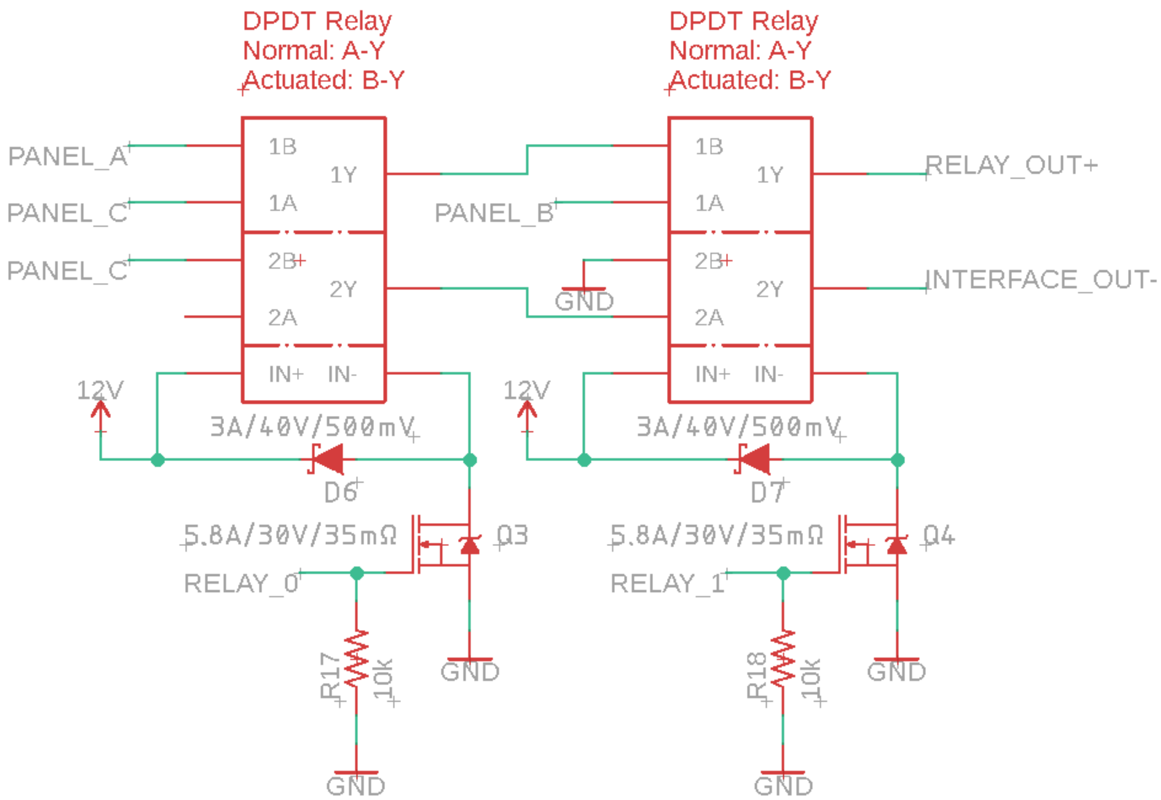


Figure 5: Solar Panel Equivalent Circuit

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 (e.g. if the 5V regulator fails or the 12V supply is disconnected), 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.

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 are 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.

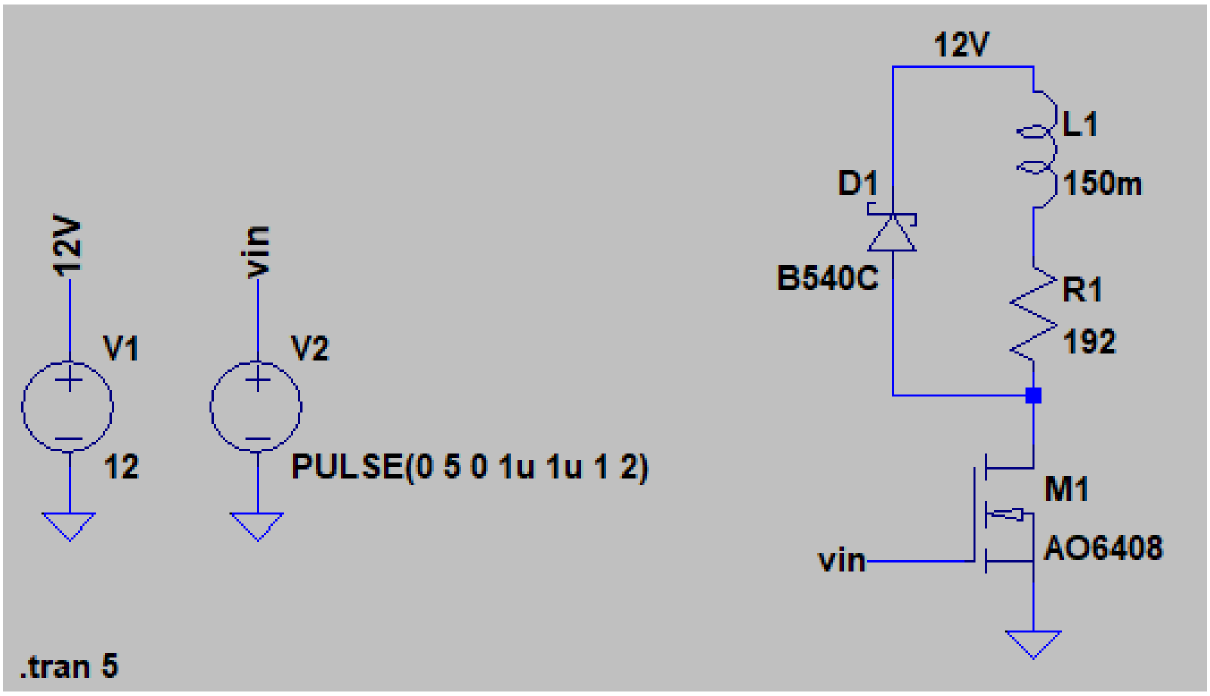


Figure 6: Simulation of relay driver circuit.

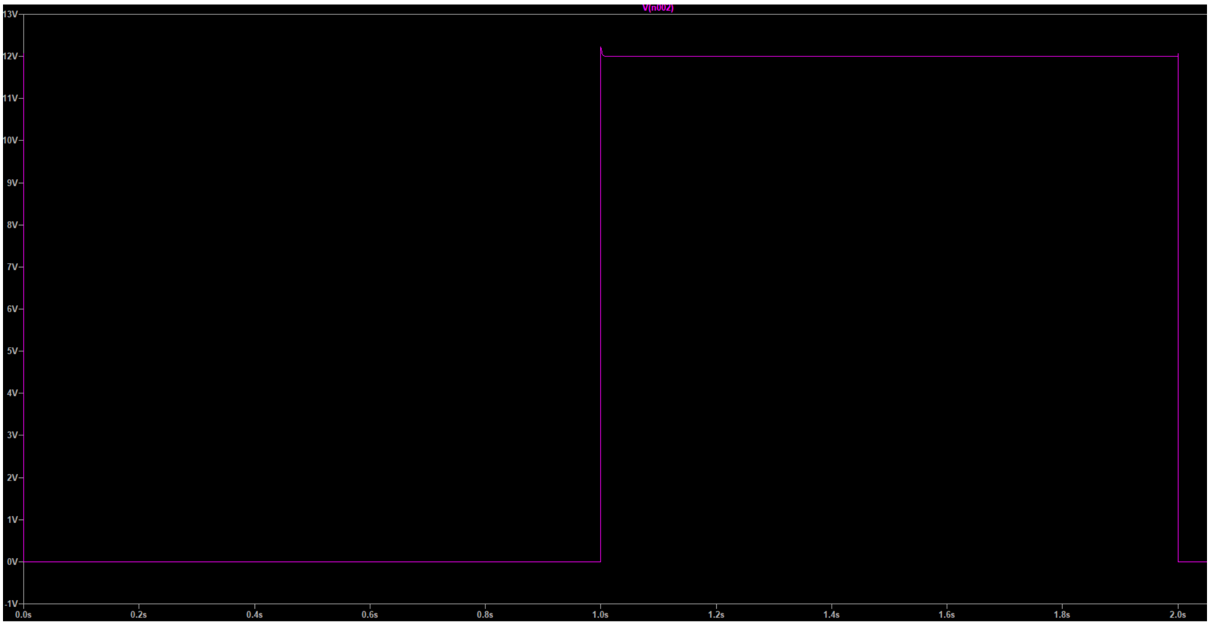


Figure 7: Driver circuit simulation results. Voltage spikes are less than 13 V which is suitable for the chosen MOSFET.

Table 4 shows the requirements and verification process for 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. · 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. · 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. · Ensure that when the control input is toggled that the voltage across the MOSFET does not exceed 30V.

Table 4: Requirements and verification table for the Switching Subsystem. All requirements assume that the circuit in Figure 5 has been constructed and powered with a 12V 1.725A power supply.

(c) Voltage/Current Monitoring Subsystem

Connecting directly to the output of the switching subsystem, the Voltage/Current Monitoring Subsystem is responsible for measuring current delivered by the solar panel to the electrical load as well as the voltage across the three sources in Figure 4. The schematic of the Voltage/Current Monitoring Subsystem is shown in Figure 8. The subsystem has two outputs: one that communicates current/voltage data to the microcontroller via an I2C bus (labeled SDA and SCL on the schematic) 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.

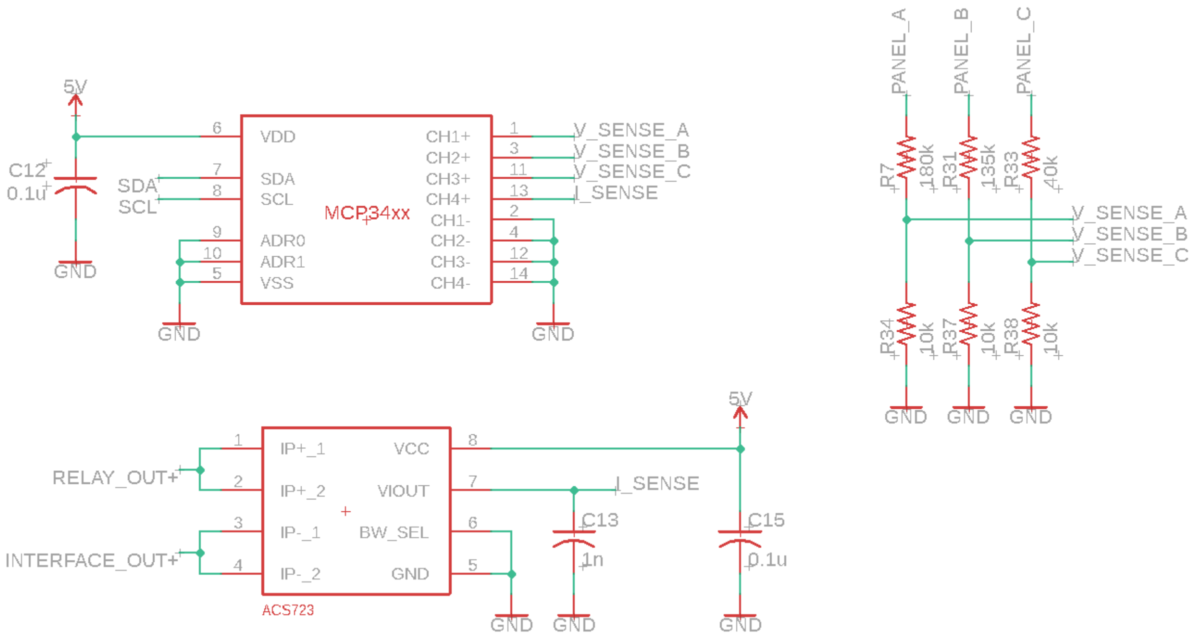


Figure 8: Voltage/Current Monitoring Subsystem schematic.

Because these solar panels are going to be used by researchers, having more accurate measurements for voltages and currents will be beneficial so long as costs are reasonable. Given 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 10mV and 10mA resolution on voltages and currents would be suitable for this application.

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 IC 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 voltage 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 voltage resolution of the first channel. Equation 2 shows the voltage resolution of the second channel. Equation 3 shows the voltage resolution of the third channel. All of these are within the 10mV specification.

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

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

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

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. As current is delivered to the electrical load a small voltage is generated across this resistor. This can then be amplified using an instrumentation amplifier and then connected directly to the fourth channel of the ADC. This method did not make it into the design because of the difficulty of finding differential amplifiers able to withstand high common-mode voltages on the order of 90 V.

The second method is to use a hall effect current sensor. These types of sensors do not require a shunt resistor in series with the primary current path. Rather, they detect the strength of the magnetic field generated when current is flowing. These sensors are more suited for high voltage current sensing because the primary current path and any associated voltages are isolated from the internal circuitry used to sense the magnetic field. 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 is shown in Equation 4 and meets the10mA specification defined above.

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. Table 5 shows the requirements and verification methods for the Voltage/Current Monitoring Subsystem.

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

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. · Connect three voltage sources to PANEL_A, PANEL_B, and PANEL_C. · 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. · 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. · Connect a current source across RELAY_OUT+ and INTERFACE_OUT+. · Run firmware to allow the microcontroller to read the voltage at Channel 4 of the ADC and extrapolate the current value. · Verify measurements are within the resolution determined in Equation 4.

Table 5: Requirements and verification table for the Voltage/Current Monitoring Subsystem. The requirements above assume that the circuit in Figure 8 is constructed and powered with a 5V 800mA supply.

(d) Temperature Monitoring Subsystem - Thermocouples

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 the schematic as shown in Figure 9. This subsystem is powered by the 5V voltage rail 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). There were three interfaces available for communicating temperature data to the microcontroller. Not including all of the components of the Temperature Monitoring Subsystem, there is one digital I/O pin available for use without sacrificing features of the other subsystems. An I2C bus is available because it is already being used by the Voltage/Current Monitoring Subsystem, and an SPI bus is available to communicate with the Ethernet Interface. The SPI bus was not a good choice because each device on the bus requires a Slave Select pin. This would be problematic when multiple temperature sensors are added. The I2C bus was not a suitable option because it is meant for short distance communication. As a result a 1-wire digital interface was determined to be the most ideal.

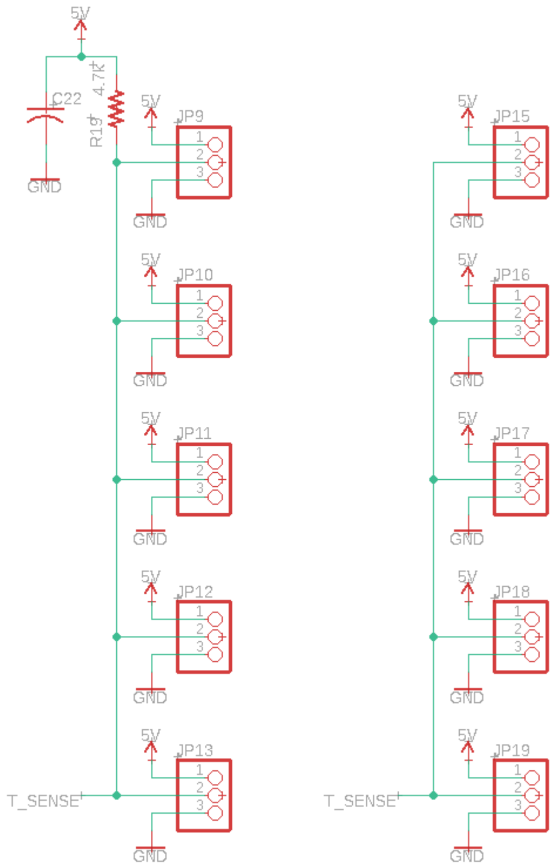


Figure 9: Temperature Monitoring Subsystem schematic.

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, we wish to disconnect the panel 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 this behavior and shut off the system if necessary. Because the Temperature Monitoring Subsystem only consists of off the shelf temperature sensors, it was just a matter of finding the correct part to fit this need and have the ability to interface with our system.

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 [5]. This is more than suitable for our temperature accuracy needs. 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 fine.

The chosen sensor is packaged in a waterproof housing at lengths of 3 meters. The sensor also uses the 1-wire Dallas protocol for communication which means that all of the sensors could be connected to a single digital I/O pin of the microcontroller. 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. Ten headers were attached to schematic in the event the user wishes to add more sensors. Table 6 shows the requirements and verification processes for the Temperature Monitoring Subsystem.

```
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 10: 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 11: Multiple DS18B20 Temperature Sensor Test Outputs

Figure 10 and Figure 11 display successful results of tests with DS18B20 temperature sensors connected to an Arduino development board. In Figure 10, a single temperature sensor was tested. In figure 11, to test the "One-Wire" configuration, two temperature sensors were configured to one digital I/O pin. In total, five sensors were attached to the Arduino and tested simultaneously. Each measurement was within the tolerance stated above.

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. · Connect the Microcontroller Subsystem such that T_SENSE is connected to a PWM enabled digital I/O pin. · Run firmware on the microcontroller to read the data coming from each sensor using the 1-wire interface. · Verify that the temperature given by each sensor is within ±5° F of the actual ambient temperature.

Table 6: Requirements and verification table for the Temperature Monitoring Subsystem. The requirements above assume the circuit in Figure 9 has been constructed and is powered with a 5V 800mA supply.

(e) 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 rail. Figure 12 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 because it contains all of the pins and interfaces needed in the system being built. By burning the Arduino bootloader into the microcontroller, we will be able to program it using the Arduino programming language as well as retain access to useful libraries used to communicate with other subsystems such as the Ethernet Interface and the temperature sensors.

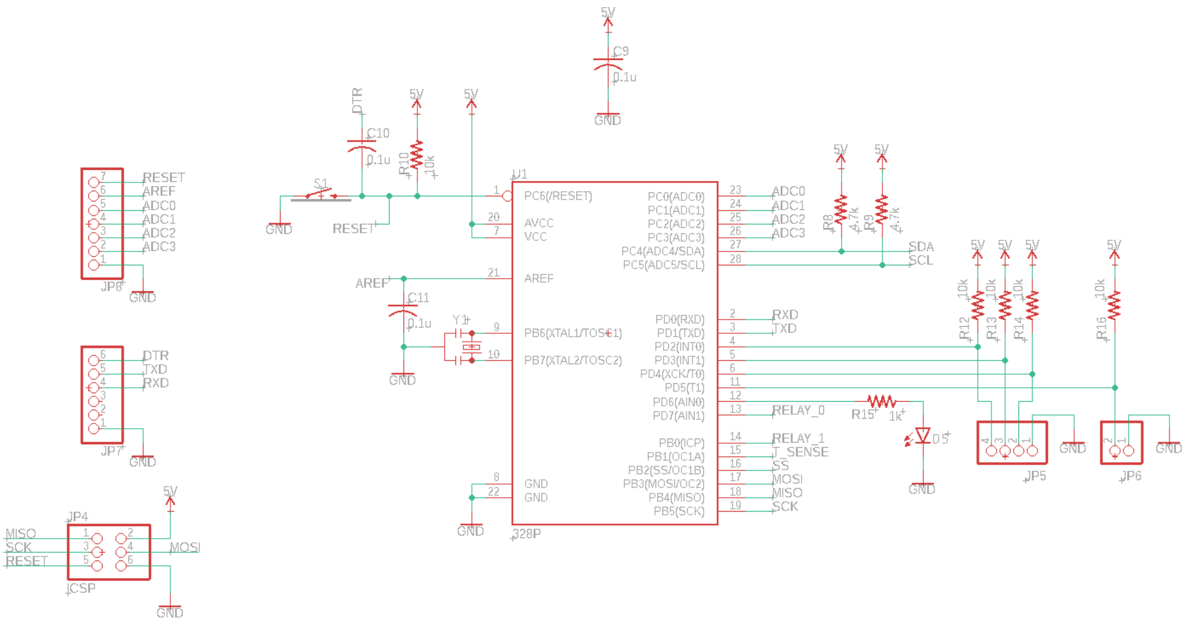


Figure 12: Microcontroller Subsystem schematic.

On the right side of schematic there are two headers used to connect to the manual configuration switches. Each position on both of those switches is connected to a digital I/O pin for reading. An LED is connected to pin 12 and is responsible for indicating if the output of the interface box is enabled. Pins 13 and 14 are used to control the relays as discussed in the Switching

Subsystem section. Pin 15 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 in the same way the Arduino is. An oboard tactile switch connected to pin 1 allows a developer to reset the microcontroller if necessary.

Three additional headers on the left side of the schematic were added for debugging purposes. A 6 pin SPI header allows developers to program the microcontroller without removing it from the board. Another 6 pin header is connected to the microcontroller's serial data lines. This will be very useful in debugging as developers can write to the serial port and debug as long as a USB-to-Serial converter is attached. Lastly, a 7 pin header contains other miscellaneous or unused pins. These headers do not need to be populated; however, during the debugging process, they may prove to be useful. Figure 7 shows the requirements and verification processes for the 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. · Run firmware to read the states of these switches. · 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. · Run firmware to toggle the output at 10 Hz. · Verify the output alternates between 0 and 5V at 10 Hz. · 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. · Run firmware to read temperature data through the PWM pin. · 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. · Run the i2cScanner sketch on the microcontroller. · 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. · Run firmware to read data on the SPI bus and echo it back. · From a terminal, send data on the SPI bus and verify the data sent back is the same.

Table 7: Requirements and verification table for the Microcontroller Subsystem. The above requirements assume that the circuit in Figure 12 has been constructed and is powered with a 5V 800mA supply.

(f) LED‘s

Several LED‘s visible to the end user will be present to show information about the status of the Ethernet connection, status of interface output connection (whether or not the interface box has connected the solar panel to the electrical load), and status of interface box power (whether or not the isolated 12V supply is powering the interface box).

The interface output connection and Ethernet connection LED’s will be controlled by the micro-controller while the interface box power LED will be powered by the 5V line from the Power Subsystem.

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. · Run the ’Blink’ example program and verify the LED blinks at 0.5 Hz.

Table 8: Requirements and verification table for the LED‘s. The requirements above assume the LED’s have a current limited series resistor and are powered with a 5V 800mA supply or connected to a digital output of the microcontroller.

(g) Manual Configuration Switches

In the case of loss of connection to the network and the server is unable to access the interface box, there will be manual switches mounted on the enclosure to control the configuration of the solar panel and shutoff the system entirely. One will be an SP3T switch with three outputs (one for each pole) responsible for changing the output configuration and the another will be an SPST switch with a signal output responsible for enabling the output of the interface box. The circuit configuration is shown in Figure 12 as headers JP5 and JP6 and their corresponding pull up resistor. Each output is connected to a digital input on the microcontroller for processing. The subsystem receives power from the 5V line from the Power Subsystem.

Requirements	Verification
The voltage outputs of this subsystem must be readable by the Microcontroller Subsystem	· Connect an SP3T switch to JP5. · Connect an SPST switch to JP6. · 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. · Connect an SPST switch to JP6. · Measure the voltages on the four outputs of the subsystem. · Verify that changing any switch position will change at most two outputs from HIGH to LOW or vice versa.

Table 9: Requirements and verification table for the Manual Configuration Switches. The requirements above assume the are powered by a 5V 800mA supply.

(h) Ethernet Interface

The Ethernet Interface is responsible for facilitating communication via Ethernet 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. In achieving this, the Ethernet Interface must utilize one of available data buses on the microcontroller. Primarily, this means that it must use I2C or SPI.

The interface was implemented using WizNet W5500 IC [4]. This IC was primarily chosen because it allows the system to be connected to the network, it can easily connect to a microcontroller's SPI bus (which is available), and because it is the Ethernet controller used in the Arduino Ethernet Shield 2. This means that any libraries already developed for this will be accessible during the firmware implementation. The circuit of the Ethernet Interface can be found on page 4/5 in the schematic at the end of this document [4]. Table 10 shows the requirements and verification processes used for this subsystem.

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. · Connect the Ethernet Interface to a network using an Ethernet cable. · Connect a PC to the network. On the microcontroller, run the UDPSendReceiveString sketch. · From the remote PC, send data over the network to the Ethernet Interface. · Verify that a response is received saying "acknowledged."

Table 10: Requirements and verification table for the Ethernet Interface. The above requirements assume that the circuit on page 4/5 in the schematic at the end of document is constructed and powered with a 3.3V 800mA supply.

(i) 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 subsystem. The interface box as a whole will be supplied with an isolated 12V power supply. Given the current requirement of all of the subsystems, this supply will need to have current carrying capacity of at least 1.725A. The system is not expected to be drawing this current continuously; however, this will be the maximum.

The 3.3V and 5V voltage rails will be generated using two 800 mA linear voltage regulators. Additionally, another switch accessible to the end user will completely shut off the 12V supply to the box. Doing so will automatically disconnect the solar panel from the output of the interface box and disconnect the Ethernet interface. The circuit for the Power Subsystem is shown in Figure 13. The header JP1 represents the SPST switch used to control the interface box power. Additionally, a series diode is present to prevent damage due to a reverse polarity at the 12V input. Each regulator, also has an LED attached to indicate their respective voltage rails are active. The requirements and verification processes for the Power Subsystem are shown in Table 11.

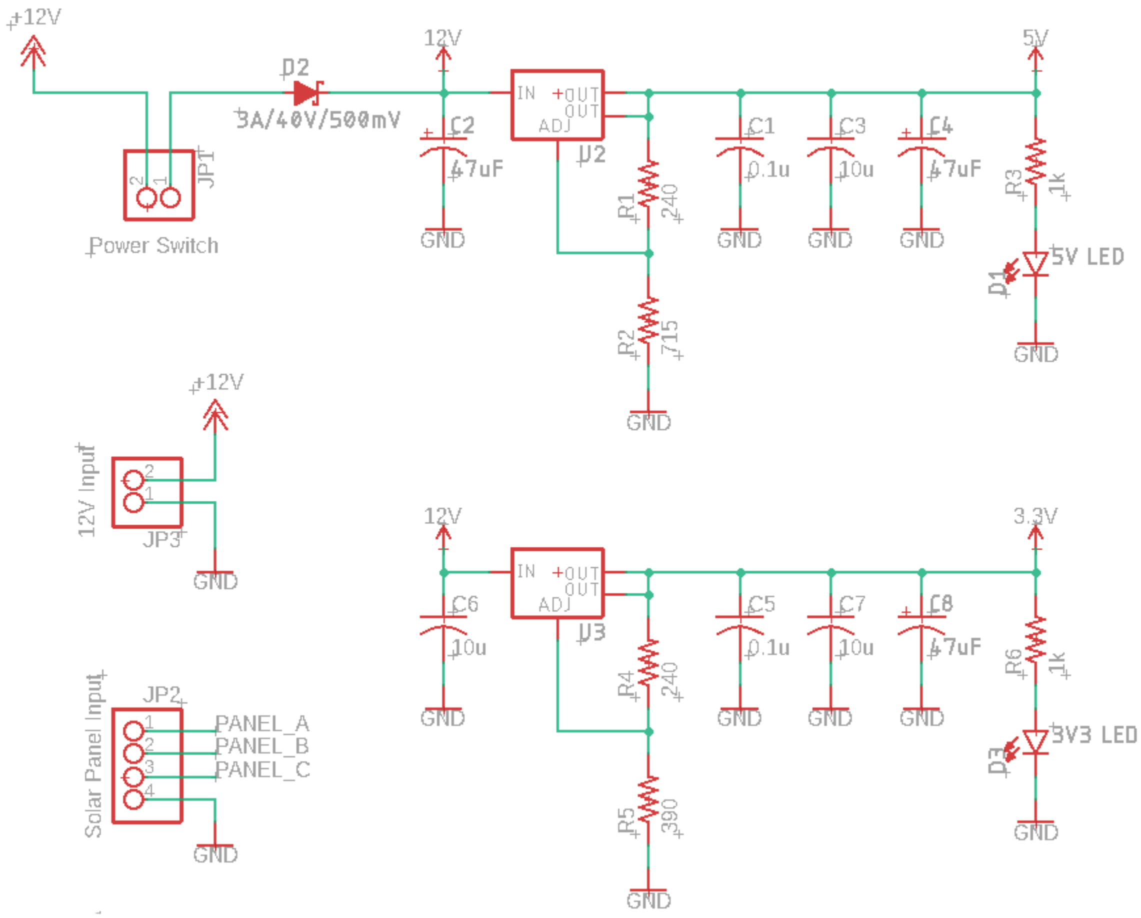


Figure 13: Power Subsystem schematic.

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. · Verify it is between 4.5-5.5V. · Attach a programmable load to the output of the regulator that is set to 800 mA. · 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. · Verify it is between 3.0-3.6V. · Attach a programmable load to the output of the regulator that is set to 800 mA. · 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. · Make sure the switch is opened. · Measure the voltages at the 5V and 3.3V output. · Verify they are both 0V.

Table 11: Requirements and verification table for the Power Subsystem. The requirements above assume that the circuit in Figure 13 has been constructed and is powered with a 12V 1.725A supply.

4. Risk Analysis

The monitoring system is highly dependent on the functionality of the switching subsystem, voltage/current monitoring subsystems, and the micro-controller all working in-tandem to ensure that the solar panel can be disconnected if any failure conditions should occur. This is most essential part of the monitoring system and is responsible for preventing a hot-spot disaster.

Along with the statement above, it is essential that all the subsystems of the interface box can work along side the micro-controller, so that they can send/receive the proper data and control the output accurately. Each subsystem is fed power from the power subsystem and it is essential that if this subsystem fails for any reason, the solar panel will be disconnected from the output and not deliver any power. Additionally, because this interface will be mounted on the back of the solar panel outside, these components must all be kept safe from environmental damage in a weather-proof enclosure.

0.3 Software and Firmware

1. The ATMega328p chip is capable of servicing both hardware and software interrupts [3]. In order to ensure that the system is protected (e.g. Over-voltage, Over-current, and Overheating), the program is designed so that when hardware or software interrupts are received from the temperature sensors, current monitoring system, or the voltage monitoring system, it will be able to service these interrupts in a timely manner.

2. In Figure 14, a simple closed loop program is implemented for the temperature sensors ideally at a rate of The user will define a temperature threshold that will shut the solar panel down completely and notify the user of overheating if the measured temperature is above the set or default threshold.

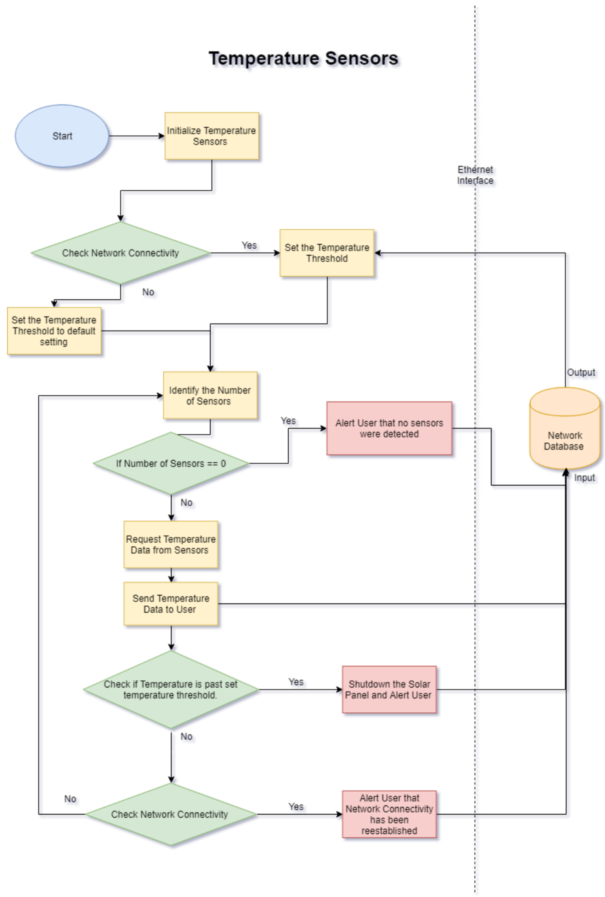


Figure 14: Temperature Monitoring Subsystem Software/Firmware Flowchart

Additionally, the program notifies the user if no temperature sensors are detected. The system will not continue in the case the temperature sensors are not connected or malfunction. If the temperature sensors are connected and the threshold is set, the software will continuously loop through each sensor and compare the temperature against the set or default temperature threshold until overheating is detected.

An additional feature of the software is to store the measured temperature data to a database, so that the user is able to view the logged data. This will be implemented if there is time after all of the high-level requirements are satisfied.

3. In Figure 15, another simple closed loop program is implemented for the Switching Subsystem and Voltage/Current Monitoring Subsystem. The user will be able to set the desired relay configuration for the solar panel cells (e.g. Cells A to D, B to D, or C to D) remotely through the network. Voltage and current will be measured through the two respective monitoring subsystems and check these measured values against the set or default thresholds continuously.

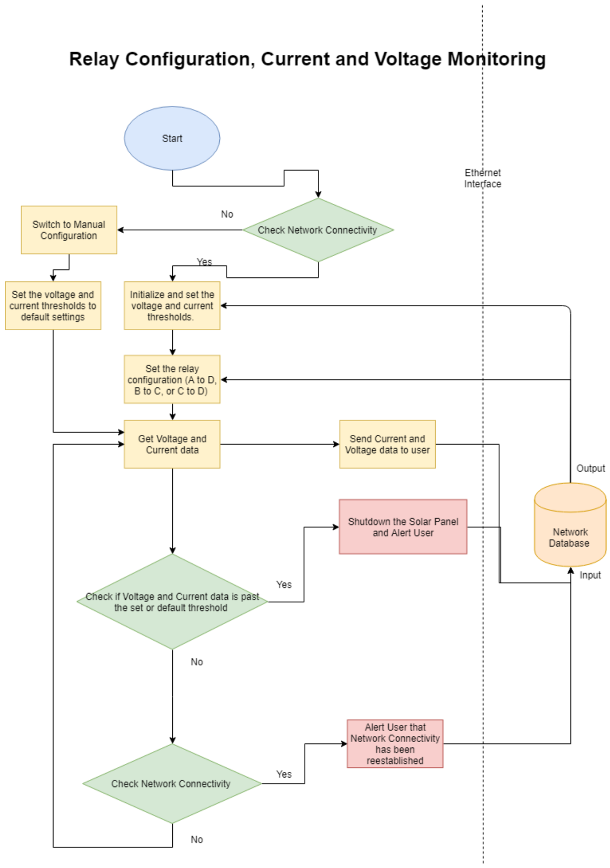


Figure 15: Relay Configuration and Current/Voltage Monitoring Software/Firmware Flowchart

If the received data is above the set or default threshold, the system will shutdown the solar panel and alert the user. Voltage and current data will then be stored into the database through the network.

4. In the case of loss in network connectivity, both programs will be able to loop continuously through the designated loops shown in Figures 14 and 15. Temperature, voltage, and current data will not be stored in this case. The program will continuously check for network connectivity until connection has been reestablished. When connection is reestablished, the program will notify the user.

5. The database and graphical user interface design is not discussed since it is outside the scope of this project.

0.4 Cost and Schedule

1. Cost Analysis

   (a) 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.

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

(b) Parts Cost:

#	Name/Description	Manufacturer	Manufacturer P/N	Quantity	Unit Price ($)
1	IC REG LIN 3.3V 800MA SOT223-4	Texas Instruments	LM1117MPX-3.3/NOPB	1	1.14
2	IC REG LINEAR 5V 800MA SOT223-4	Texas Instruments	LM1117MPX-5.0/NOPB	1	1.14
3	MOSFET SOT-23 N-CH LOGIC	ON Semiconductor / Fairchild	NDS351AN	2	0.39
4	MOSFET N-CH 30V 5.8A SOT23	Alpha & Omega Semiconductor Inc.	AO3404A	2	0.49
5	DIODE SCHOTTKY 40V 3A SMA	Diodes Incorporated	B340A-13-F	3	0.42
6	IC ADC 16BIT SIGMA-DELTA 14SOIC	Microchip Technology	MCP3428-E/SL	1	3.67
7	IC MCU 8BIT 32KB FLASH 28DIP	Microchip Technology	ATMEGA328P-PU	1	2.14
8	RELAY GEN PURPOSE DPDT 12A 12V	TE Connectivity Potter & Brumfield Relays	PT270012	2	7.73
9	SENSOR CURRENT HALL 20A DC	Allegro MicroSystems, LLC	ACS723LLCTR-20AU-T	1	5.27
10	IC CTLR 3-1 ETH TCP/IP 48LQFP	WIZnet	W5500	1	3.95
11	IC SUPERVISOR PP ACT LO SOT143-4	ON Semiconductor	CAT811TTBI-GT3	1	0.43
12	IC BUF NON-INVERT 5.5V SC70-5	Texas Instruments	SN74LVC1G125DCKR	3	0.33
13	CONN MAGJACK 1PORT 100 BASE-T	WIZnet	RB1-125BAG1A	1	2.55
14	LED YELLOW CLEAR 0603 SMD	Inolux	IN-S63AT5Y	5	0.32
15	LED RED DIFFUSED T-1 3/4 T/H	Rohm Semiconductor	SLR-56VR3F	2	0.6
16	CERAMIC RES 16.0000MHZ 30PF T/H	Murata Electronics	CSTLS16M0X55Z-A0	1	0.48
17	RES SMD 0 OHM JUMPER 1/10W 0603	Yageo	RC0603JR-070RL	2	0.1
18	CAP CER 0.1UF 25V X7R 0603	Yageo	CC0603KRX7R8BB104	23	0.05
19	RES ARRAY 4 RES 10K OHM 1206	Bourns Inc.	CAY16-103J4LF	2	0.1
20	RES SMD 10K OHM 1% 1/10W 0603	Yageo	RC0603FR-0710KL	19	0.1
21	CAP CER 10000PF 25V X7R 0603	Yageo	CC0603KRX7R8BB103	8	0.1
22	CAP CER 10PF 50V C0G/NPO 0603	Yageo	CC0603JRNPO9BN100	2	0.1
23	CAP CER 10UF 25V X5R 0805	Yageo	CC0805KKX5R8BB106	4	0.51
24	RES SMD 12.4K OHM 1% 1/10W 0603	Yageo	RC0603FR-0712K4L	1	0.1
25	Thin Film Resistors - SMD 1/10W 135K Ohms 0.1%	Yageo	RT0603BRD07135KL	1	0.38
26	RES SMD 180K OHM 1% 1/10W 0603	Yageo	RC0603FR-07180KL	1	0.1
27	RES SMD 1M OHM 1% 1/10W 0603	Yageo	RC0603FR-071ML	1	0.1
28	RES SMD 1K OHM 1% 1/10W 0603	Yageo	AC0603FR-071KL	4	0.1
29	CAP CER 1000PF 25V C0G/NPO 0603	Yageo	CC0603JRNPO8BN102	1	0.1
30	RES SMD 22 OHM 1% 1/10W 0603	Yageo	AC0603FR-0722RL	1	0.1
31	CAP CER 22UF 25V X5R 0805	Murata Electronics	GRM21BR61E226ME44L	4	0.49
32	RES SMD 240 OHM 1% 1/10W 0603	Yageo	AC0603FR-07240RL	2	0.1
33	CRYSTAL 25.0000MHZ 18PF T/H	Abracon LLC	ABL-25.000MHZ-B2F	1	0.33
34	RES SMD 390 OHM 1% 1/10W 0603	Yageo	RC0603FR-07390RL	1	0.1
35	DIODE SCHOTTKY 40V 3A SMA	Diodes Incorporated	B340A-13-F	3	0.42
36	RES SMD 4.7K OHM 1% 1/10W 0603	Yageo	AC0603FR-074K7L	3	0.1
37	Thin Film Resistors - SMD 150mW 40kohm 0.1% 25PPM 0603	Vishay / Thin Film	PAT0603E4002BST1	1	0.75
38	RES ARRAY 4 RES 470 OHM 1206	Bourns Inc.	CAY16-471J4LF	1	0.18
39	CAP ALUM 47UF 20% 35V SMD	Panasonic Electronic Components	EEE-1VA470WP	3	0.41
40	RES ARRAY 4 RES 49.9 OHM 1206	Bourns Inc.	CAY16-49R9F4LF	1	0.14
41	RES SMD 715 OHM 1% 1/10W 0603	Yageo	RC0603FR-07715RL	1	0.1
42	FERRITE BEAD 30 OHM 0805 1LN	Murata Electronics	BLM21PG300SN1D	2	0.11
43	SWITCH TACTILE SPST-NO 0.05A 24V	TE Connectivity ALCOSWITCH Switches	1825910-6	1	0.1
44	MOSFET P-CH 20V 3.5A SOT23	Nexperia USA Inc.	PMV48XP,215	1	0.39
45	SWITCH TOGGLE SPST 6A 125V	NKK Switches	M2011S3A1W03	2	3.46
46	SWITCH TOGGLE SP3T 5A 120V	E-Switch	100DP6T1B1M1QEH	1	4.28
47	CONN HEADER R/A 40POS 2.54MM	Sullins Connector Solutions	PRPC040SBAN-M71RC	2	0.83
48	CONN HEADER R/A 40POS 2.54MM	Sullins Connector Solutions	PRPC040SGAN-M71RC	1	0.97
49	CONN QC TAB 0.250 SOLDER	TE Connectivity AMP Connectors	1217861-1	6	0.03554
50	CONN QC RCPT 10-12AWG 0.250	TE Connectivity AMP Connectors	3-350819-2	6	0.34
51	CONN HEADER VERT 6POS 2.54MM	Amphenol ICC (FCI)	68602-406HLF	1	0.25
52	Universal Multifunctional C-Clamp Clip	Julius Studio	JSAG381	1	$10.10
53	Wing Nut, 1/4-20-Inch	The Hillman Group	659	1	6.41
54	1/4-20 x 3/4-Inch Button Socket Cap Screw, Stainless Steel	The Hillman Group	44007	1	6.22
	Total				97.13324

Table 12: Cost of Parts

(c) Total Cost:

Labor	$20,250
Parts	$97.13
Grand Total	$20,347.13

Table 13: Grand Total

2. Schedule

Week	Dillon	Doug	Sachin
10/7	Draft PCB layout and pinout	Prototype and Test Ethernet Interface.	Order weatherproof enclosure for PCB
10/14	Finalize orders for any additional components for the PCB	Draft and design database (back end)	Draft and design GUI (front end)
10/21	Write firmware for Voltage/Current Montoring Subsystem	Program and Test Ethernet Interface	Program and test Temperature Sensor
10/28	Assemble and debug Switching Subsystem	Assemble and debug Power Subsystem	Test and debug Manual Configuration Switches
11/4	Revise and order version 2 of PCB	Program and test database	Program and test GUI
11/11	Test and debug final PCB	Interface database and GUI to PCB	Modify weatherproof enclosure and install PCB in the enclosure
11/18	Test and debug data transfer over network	Test and debug data transfer over network	Test and debug data transfer over network
11/25	Prepare for final demo	Prepare for final demo	Prepare for final demo
12/2	Prepare PowerPoint for final presentation	Begin final paper	Begin extra credit poster
12/9	Final Presentation	Final Presentation	Final Presentation

Table 14: Project Schedule

0.5 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 should 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 casing must protect the internal electrical circuitry 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 closed 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'[1].

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'[1].

Citations and References

[1] "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]

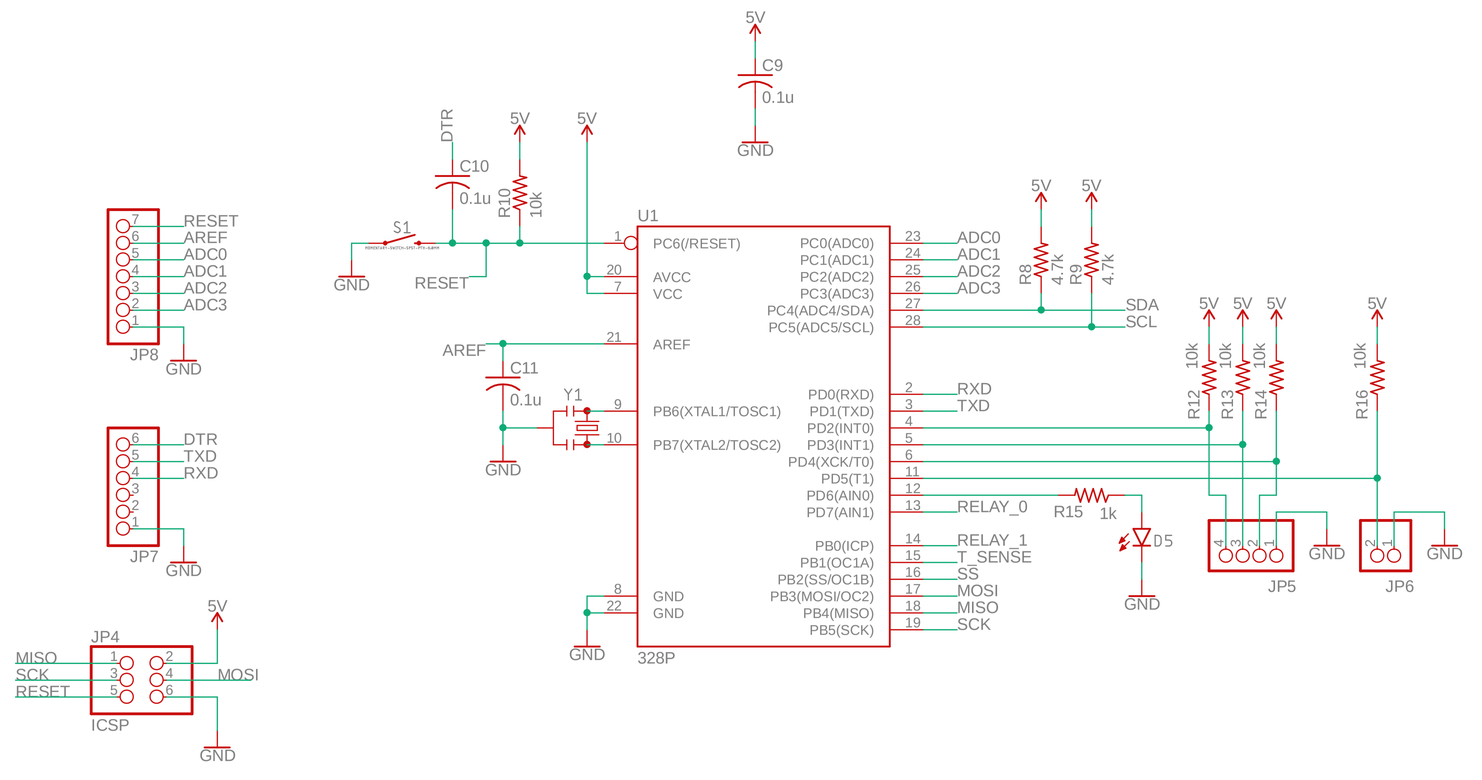
[2] SPR-425E-WHT-D Solar Panel from SunPower Posharp, Inc. [online]. Available at: http://www.posharp.com/photovoltaic/solarpanel.aspx?pid=32ceb0e4-27f3-47f6-a60a-935cfd768488 [Accessed Oct 1 2019]

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

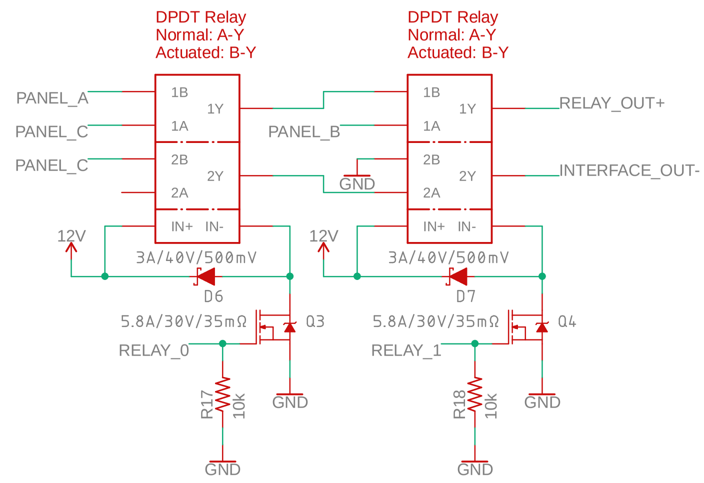
[4] Store.arduino.cc. (2019). Arduino Ethernet Shield 2. [online] Available at:https://store.arduino.cc/usa/arduino-ethernet-shield-2 [Accessed Oct 3. 2019]

[5] 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]

[6] 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].



Microcontroller Subsystem	
TITLE: smartInterface	
Document Number:	REV:
Date: 9/30/2019 8:10 AM	Sheet: 1/5



Switching Subsystem

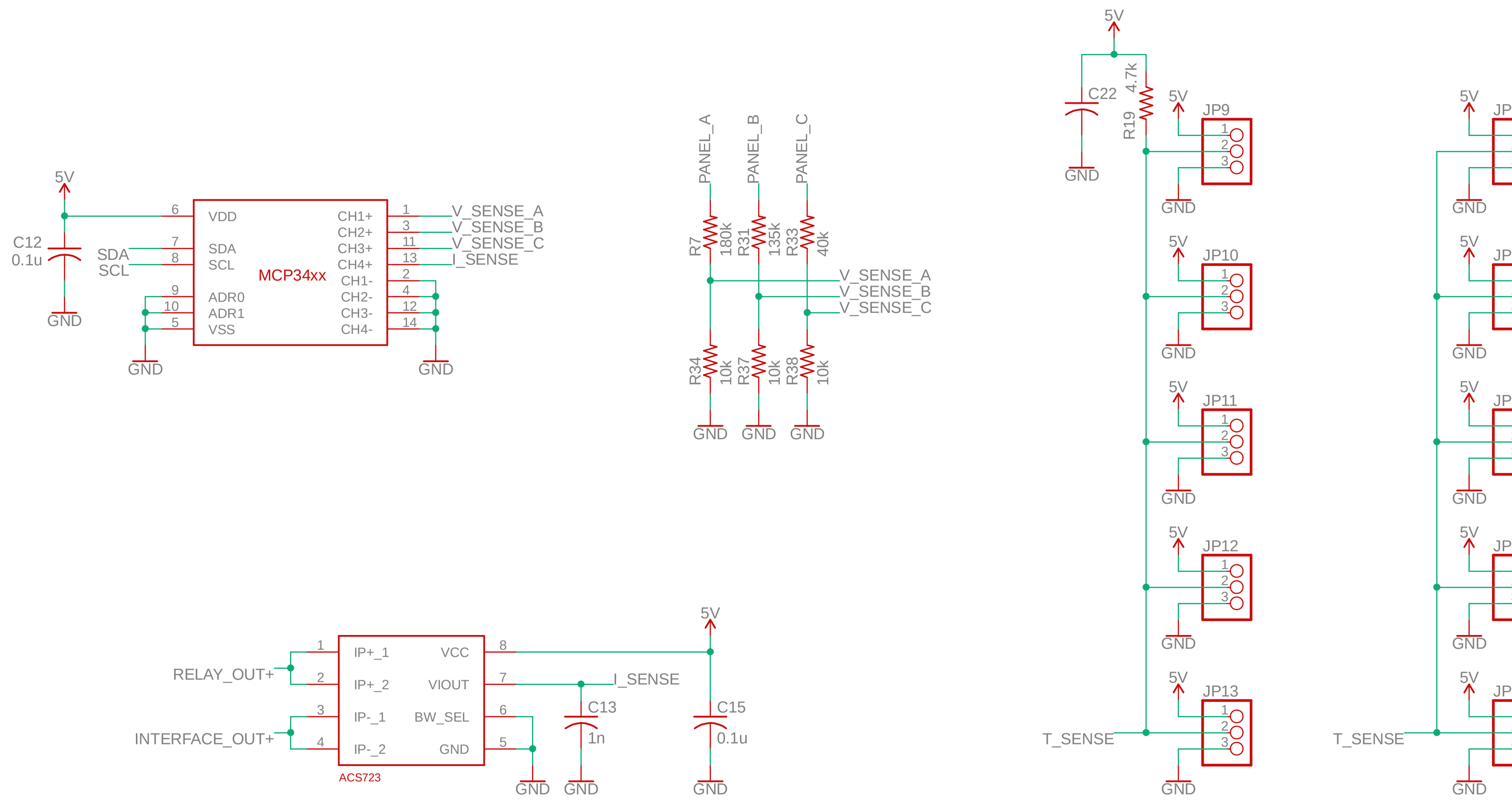
TITLE: smartInterface

Document Number:

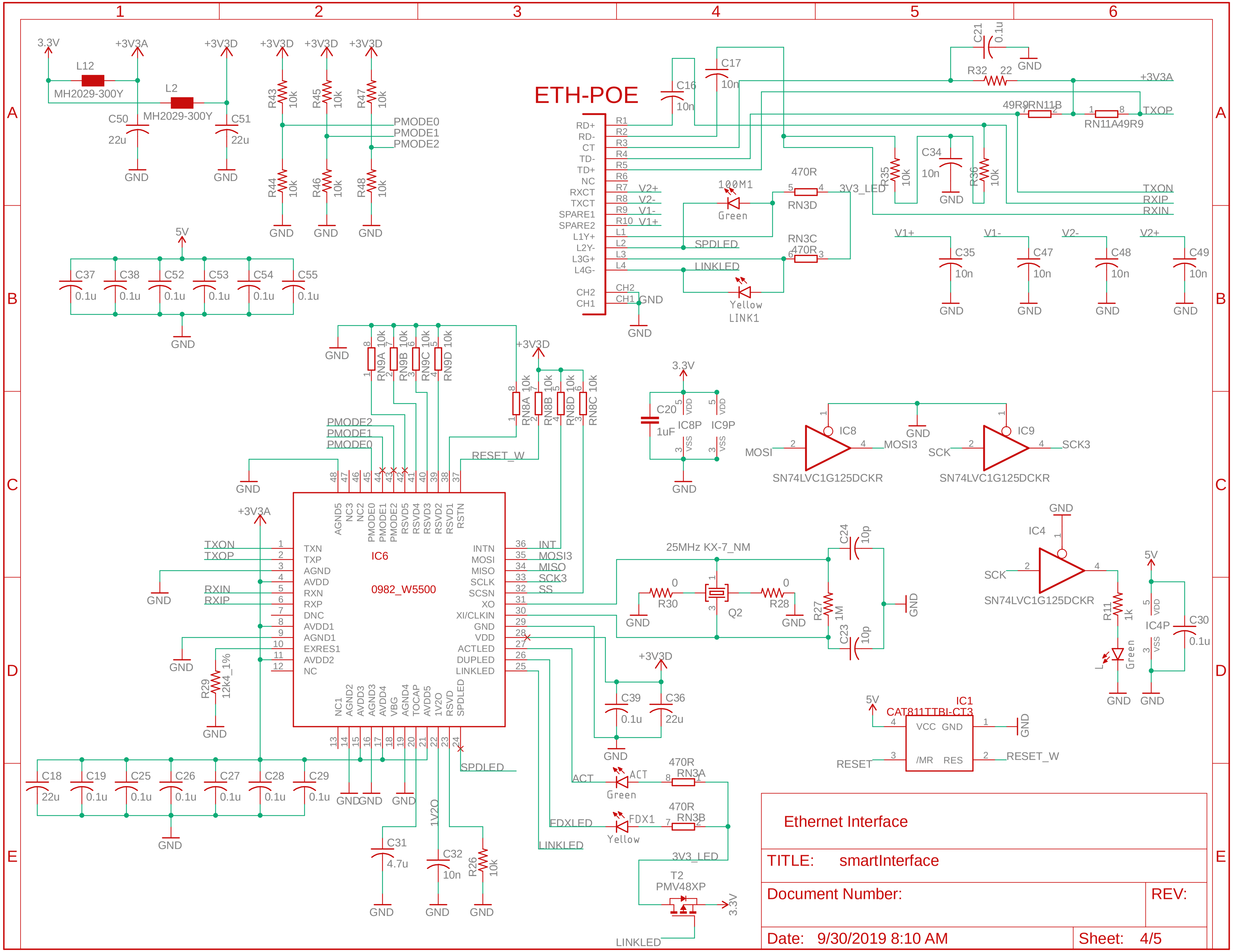
REV:

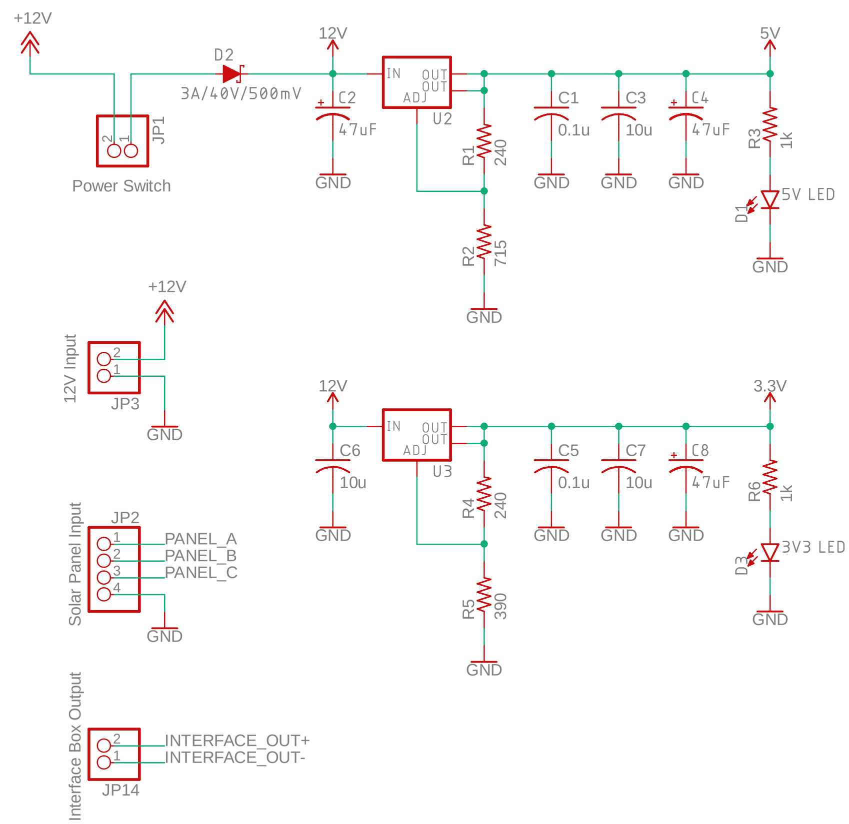
Date: 9/30/2019 8:10 AM

Sheet: 2/5



Voltage, Current, and Temperature Monitoring	
TITLE: smartInterface	
Document Number:	REV:
Date: 9/30/2019 8:10 AM	Sheet: 3/5


ETH-POE
IC6
0982_W5500
SN74LVC1G125DCKR
25MHz KX-7_NM
CAT811TTBI-CT3
PMV48XP
Ethernet Interface
TITLE: smartInterface
Document Number:
REV:
Date: 9/30/2019 8:10 AM
Sheet: 4/5



Power Subsystem

TITLE: smartInterface

Document Number:

REV:

Date: 9/30/2019 8:10 AM

Sheet: 5/5