POWER BOARD FOR ILLINISAT-3

By

Dongze Lu

Robert Maksimowicz

Final Report for ECE 445, Senior Design, Fall 2018

TA: Zhen Qin

12 December 2018

Project No. 35

Abstract

The purpose of this project is to design a power board for the new IlliniSat-3 satellite bus. The power board delivers power to different subsystems of the satellite based on the status of the battery pack. A nanosatellite bus system handles all communication, power, command and data handling, and attitude determination and control need for the satellite. By introducing a CAN bus as a form of communication between the subsystems and an automobile grade, high performance microcontroller, the power board becomes more reliable than its previous IlliniSat-2 version. The CAN bus decreases power board dependency on the system which results in shorter development time for new missions. This project demonstrates the effort to create a power board prototype that includes the features described above.

Contents

1. Introduction

1.1 Purpose and Functionality

CubeSats are a class of nanosatellites that use a standard size and form factor. The standard CubeSat size uses a “one unit” or “1U” measuring 10x10x10 cm and is extendable to larger sizes [1]. The development of CubeSats grew significantly over the past few years and that it has advanced into an industry with government and academia involvement for increasing capabilities. CubeSats provide a low-cost, effective platform for science investigators, new technology demonstrations and advanced mission concepts using constellations--swarms for disaggregated systems [1]. With a rapid increase in number of CubeSats launched, there are associated problems. During the years of 2010-2015, there have been 187 CubeSat missions launched into orbit, of which only ~50% were successful [2]. Around 15% experienced early loss and around 23% of them were declared dead on arrival (DOA).

The CubeSat program at University of Illinois at Urbana-Champaign (UIUC) is responsible for designing an IlliniSat-2, a generic picosatellite bus system. IlliniSat-2 is a scalable CubeSat platform that handles all communication, power, command and data handling, and attitude determination and control need for the satellite [3]. The problem with IlliniSat-2 is that it is “too centralized”. This results in longer development time for each mission due to over complicated software. Also, certain parts of the system are too fragile and too slow and the whole bus is more expensive than previously anticipated.

The goal of this project is to develop a power board that is more reliable than its previous version. By introducing a Controller Area Network (CAN) bus as a form of communication between the subsystems we decreased the development time for new missions, since the subsystems will not be highly dependable on each other to function properly. The system is designed to be as simple as possible, minimizing the points of failure. The choice of components is based on reliability, such as a Dual CPU’s running in Lockstep feature for the microcontroller. The microcontroller selection is also an important aspect of the project. Our choice is resistant to radiation and magnetic effects, it has low power consumption, and supports the communication protocols such as CAN. The power board can deliver the power to other subsystems through hot swap controllers. Hot swapping is the technique used in systems whenever a module must be added or removed without disturbing the whole system. Throughout the duration of the mission, the power board must always be functional. However, certain subsystems of the satellite do not. These subsystems, such as scientific payloads, must be turned on and off whenever not in use, without interrupting the satellite power delivery system.

1.2 High Level Design

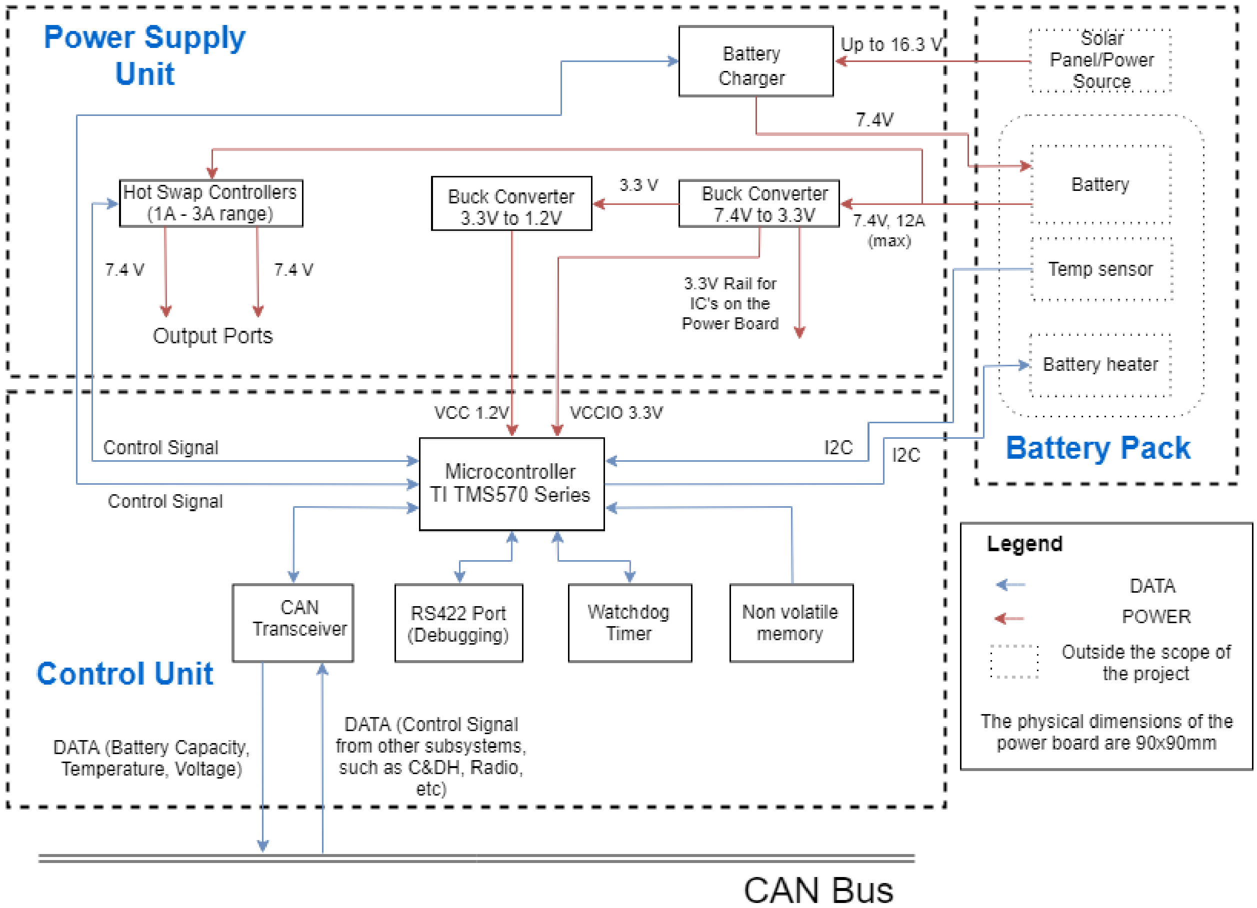

Figure 1 Block Diagram

Figure 1 shows top-level diagram of the power board. The system is divided into two subsystems: the power unit and the control unit. The battery pack is outside the scope of the project and it was replaced by the power source to simulate the battery.

The power unit is responsible for delivering the power to the rest of the satellite subsystem--for example a radio board or a scientific payload--through user configurable hot swap controllers. The hot swap devices, Texas Instruments TPS 2420, provide highly integrated load protection for our application. The reason behind this model is that the controllers are configurable with the resistor and capacitor values based on the user application. The programmable current limit threshold sets the maximum current allowed into the load, for both inrush and severe load faults. Both events use the programmable timer which inhibits all current to the load when it expires [4]. There are two buck converters in the design. One of them steps down the voltage to power the Integrated Circuits (ICs) on the power board, such as CAN transceiver and RS422 port IC, as well as Input/output (I/O) pins of the microcontroller. The other one steps down the voltage to power the microcontroller.

The control unit is responsible for all the communications of the power board with other satellite subsystems. The microcontroller is the brains of the control unit that performs decisions, such as enabling hot swap controllers based on the current battery status or decides when to turn on the battery heaters based on the battery operating temperature. In the previous satellite bus design, the Control and Data Handling (C&DH) board processed all the housekeeping data and performed decisions described above. In the situation where the C&DH board fails, the satellite would be no longer be functional. However, with the introduction of the CAN bus, all subsystems can communicate with each other and can process data and make decisions internally. The control unit consist of a watchdog timer that automatically generates a system reset if the main program neglects to service it periodically [5]. The non-volatile memory features low power consumption, is not affected by magnetic fields, and is radiation resistant [6]. The non-volatile memory in our design serves as a reliability feature and holds the operating code of the microcontroller. The RS422 port exists for debugging purposes.

2 Design

2.1 Power Unit

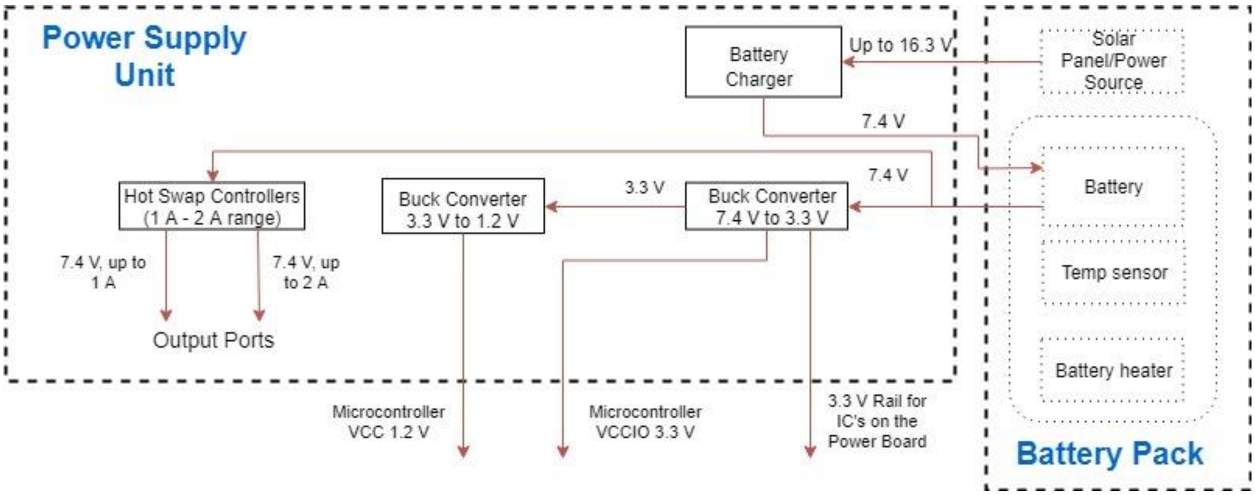

Figure 2 Power Supply Unit Block Diagram

2.1.1 Buck Converters

Figure 2 shows two buck converters present in the power supply unit. Buck converters come in a variety of different versions. We've decided to choose a fixed-voltage output version for both.

The choice for the buck converter that steps down the voltage from 7.4 V to 3.3 V is Texas Instruments TPS 62110 chip. The choice for the buck converter that steps down the voltage from 3.3 V to 1.2 V is Texas Instruments TPS 62002 chip.

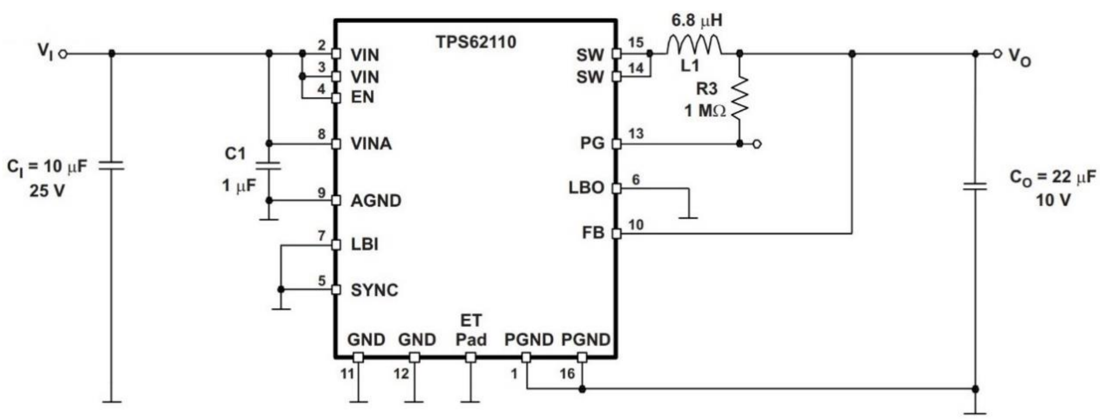

Figure 3 TPS62110 Step-Down Buck Converter

Figure 3 shows a standard connection for the buck converter fixed-output version and placement of components in the circuit. For testing purposes, we have used a standard connection schematic from the manufacturer provided manuals for both converters. There are a few capacitors and an inductor that must be selected for an optimal performance.

For good performance, a peak-to-peak inductor-current ripple should be less than 30% of the maximum DC output current [7]. Since the difference between the input and output voltage was not significant, we have decided to use 6.8 µH inductor. When the input voltage is greater than 12 V, it is recommended to increase the inductor value to 10 µH or 22 µH. This is not the case, since the input voltage from the battery pack is around 7.4 V. Inductor choice was based on the maximum output current to avoid saturation of the inductor, and the lowest DC resistance we could find for highest efficiency. The maximal output current is calculated using Equation 1 where δI_L is defined in Equation 2.

$$I_L max = I_O max + \frac{\delta I_L}{2} \tag{1}$$

$$\delta I_L = V_O \times \frac{1 - \frac{V_O}{V_I}}{L \times f} \tag{2}$$

Equation 2 calculates δI_L, which is a peak-to-peak inductor ripple current. Variable L is the inductor value and f is the switching frequency. Substituting these numbers for the real values yields Equation 3.

$$I_L max = 1.5\,A + \frac{3.3\,V \times \frac{1 - \frac{3.3\,V}{7.4\,V}}{6.8\mu\text{H} \times 1\,MHz}}{2} = 1.65\,A \tag{3}$$

Based on these calculations, we picked a Coilcraft inductor model MSS6131-682 with the maximum DC current rated at 2.8 A and maximum DC resistance at 52 mOhms.

A typical 22 µF output capacitor is needed with a 6.8 µH inductor [7]. The overall output ripple voltage is the sum of the voltage spike caused by the output capacitor ESR plus the voltage ripple caused by charging and discharging the output capacitor [7] calculated using Equation 4.

$$\Delta V_O = V_O \times \frac{1 - \frac{V_O}{V_I}}{L \times f} \times \left(\frac{1}{8 \times C_O \times f} + R_{ESR}\right) \tag{4}$$

$$3.3\,V \times \frac{1 - \frac{3.3\,V}{7.4\,V}}{6.8\ \mu\text{H} \times 1\ \text{MHz}} \times \frac{1}{8 \times 22\ \mu\text{F} \times 1\text{Mhz}} = 0.0015\,V \tag{5}$$

Based on Equation 5, using a 22 µF output capacitor should yield a minimal ripple at the voltage output.

The nature of the buck converter is a pulsating input current [7]. An input capacitor is required for best input voltage filtering and to minimize the interference with other circuits. The input capacitor should have a minimum value of 10 µF and can be increased without any limit for better input voltage filtering [7]. We've decided to use 10 µF ceramic capacitor rated for 10 V applications. Ceramic capacitors show a good performance because of their low ESR value, and they are less sensitive against voltage transients compared to tantalum capacitors [7].

Based on the design described above, the buck converters performed to our expectations when tested as separate units. There was no need to change the design based on the results. However, the results after testing the converters on the actual PCB were not the same. The voltage ripple for the TPS62002 was beyond the requirements. We believe that the reason was a bad design of the PCB. The input capacitors should be placed as close as possible to the voltage input pins of the IC for best performance. The same follows for the output voltage pins.

2.1.2 Hot Swap Controllers

Hot swapping is a technique used in systems where a module must be removed or added without disturbing the system. A good application example are servers or network switches. On the satellite, the power board must always remain operational, while turning on and off different subsystems, such as radio or scientific payloads. The TPS2420 Hot-Swap device provides highly integrated load protection for applications up to 20 V. The device protects loads, minimizes inrush current, and safely shuts down in the event of a fault [4]. The programmable fault current makes the device a perfect choice for our application. Figure 4 shows an example schematic for the hot swap controller connections. In our design there was no need to include C_{VIN} or C_{LOAD} capacitors, as well as the diode shown in schematics.

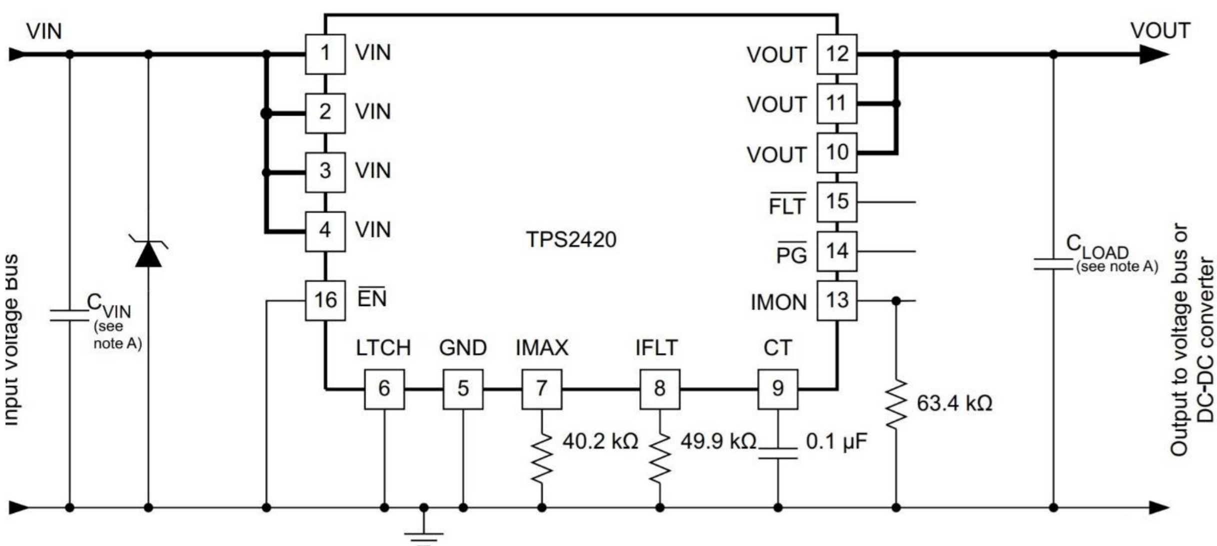

Figure 4 Hot Swap Controller Schematics

The resistor at pin 8, I_{FLT} sets the fault current level. The resistor at pin 7, I_{MAX} sets the current-limit level. Currents between the fault current threshold and the current-limit are permitted to flow unimpeded for the period set by the fault timer programmed by capacitor C_T at pin 9. The fault timer starts when the fault current threshold is exceeded. The fault timer starts charging C_T when current through V_{IN} exceeds

I_{FAULT}. If the current doesn't drop below the I_{FAULT} level before V_{CT} reaches the upper threshold the output will be shut off [4].

The fault current resistor is set by Equation 6 where R_{IFLT} is in Ω and I_{FAULT} is in Amperes (A). The current-limit level is set by Equation 7 where R_{IMAX} is in Ω and I_{MAX} is in Amperes (A). The fault timer period is selected using the Equation 8 where T_{FAULT} is the minimum timer period in seconds and C_{CT} is in Farads.

$$R_{IFLT} = \frac{200\ k\Omega}{I_{MAX}} \tag{6}$$

$$R_{IMAX} = \frac{201\ k\Omega}{I_{MAX}} \tag{7}$$

$$C_{CT} = \frac{T_{FAULT}}{38.9 \times 10^3} \tag{8}$$

The recommended operating voltage range is 3 V to 18 V, in our design, the controller will operate at 7.4 V. The output voltage for the TPS2420 device is approximated by Equation 9.

$$V_{OUT} = V_{IN} - 0.04 \times I_{OUT} \tag{9}$$

The power board consists of two hot swap controllers. One controller is set for I_{FAULT} at 1 A, I_{MAX} at 2 A, and has a T_{FAULT} set to 0.86 ms using a 0.022 µF capacitor C_{CT}. The resistor value for R_{IFLT} is 200 kΩ, and the resistor value for R_{IMAX} is 100 kΩ. The other controller is set for I_{FAULT} at 2 A, I_{MAX} at 3 A, and has a T_{FAULT} set to 3.89 ms using a 0.1 µF capacitor C_{CT}. The resistor value for R_{IFLT} is 100 kΩ, and the resistor value for R_{IMAX} is 65.5 kΩ. All of the resistor and capacitor values were calculated based on Equation 6, 7, and 8. When operating near the I_{FAULT}, based on Equation 9, Equation 10 and 11 yields the voltage output V_{OUT} to be approximately 7.36 V for the first hot swap and 7.32 V for the other.

$$7.4\,V - (0.04 \times 1\,A = A) = 7.36\,V \tag{10}$$

$$7.4\,V - (0.04 \times 2\,A) = 7.32\,V \tag{11}$$

The I_{MON} pin is used to indicate the current through the device was not used. FLT pin is used to indicate if the fault time has expired and the controller switched off.

2.1.3 Battery Charger

Due to the time limit and extremely small form factor of the chip, battery charging was not incorporated in this project.

2.2 Control Unit

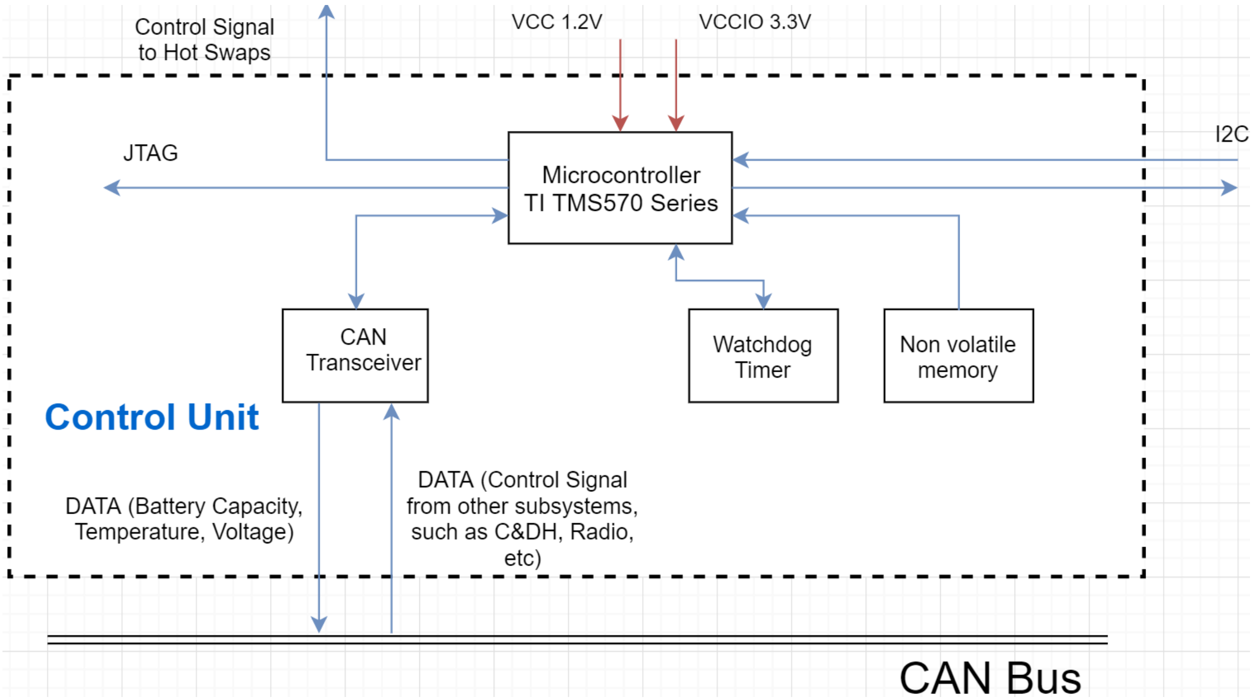

Figure 5 Control Unit Block Diagram

The control unit of the power board consist of the microcontroller, CAN transceiver, watchdog timer, and non-volatile memory ICs.

2.2.1 Microcontroller

Microcontroller handles all the logic control behind enabling and disabling hot swap controllers using General-Purpose Input/output (GPIO) pins. The communication with the battery pack is established using Inter Integrated Circuit (I^2C). Controller Area Network (CAN) bus is used to communicate with other satellite subsystems such as radio board or Control and Data Handling Board. Based on the housekeeping data, such as temperature from the battery pack, the microcontroller performs decisions such as when to switch the battery heaters.

Our choice of microcontroller is a Texas Instruments Hercules series TMS570LS1227. Table 1 shows the advantages over the previous version of the microcontroller that was used for the power board, MSP430 series.

Table 1 Microcontroller Selection Features

Features	TMS570LS1227	MSP430
# of GPIO Pins	58	44
Flash Memory (KB)	1280	32
I2C	1	2
CAN	Yes	No
Cost ($)	23.95	4.63

TMS570 microcontroller was chosen for its high performance and safety critical applications. This model features detection probability of errors induced by voltage spikes or ionizing radiation [8]. It features dual CPUs running in lockstep, error signaling module with error pin and voltage and clock monitoring [9]. TMS570 series microcontroller flash memory size provides scalability as the memory size determines the maximum program size.

2.2.2 CAN Transceiver

The CAN transceiver operates with a single 3.3 V supply. The device is designed for data rates up to 1 megabit per second (Mbps) and includes many protection features providing device and CAN network robustness. The chip is automotive graded and designed to operate in harsh environments making it ideal for satellite application.

CAN bus messages are represented using two square wave channels, CAN high and CAN low. CAN bus data packets cannot be interpreted by the microcontroller directly without the CAN transceiver. CAN transceiver provides the functionality to decode CAN messages into a single channel square wave format which could be interpreted by the microcontroller. The transceiver encodes the data packets send from the microcontroller to be recognizable by another CAN bus device.

2.2.3 Watchdog Timer

When a satellite operates in space, no human interference is possible. The microcontroller might run into a problem such as that the main program ending up in an infinite loop. It is not possible to manually reset the microcontroller when an error occurs. Therefore, a watchdog timer automatically generates a system reset if there’s no signal from the main program after some user defined timer expires [10].

2.2.4 Non-Volatile Memory

The non-volatile memory features low power consumption, is not affected by magnetic-fields, and is radiation resistant [6]. The non-volatile memory in our design serves as a reliability feature. Non-volatile memory will hold the operating code that is loaded into the microcontroller. We chose the FRAM chip manufactured by Cypress Semiconductor for our non-volatile memory since FRAM devices are impervious to external electric fields [6].

2.2.5 RS422 Port

In our initial design, RS422 port is utilized for connecting the power board with a PC. The microcontroller uses serial communications interface (SCI) to communicate with a PC where a differential line driver and differential-input line receiver is needed. However, in later development, RS422 port tested to be insufficient. The serial communication failed to obtain microcontroller register value as well as the return values in user specified address. Furthermore, SCI communication is able to output the error message to PC only in a program segment and lacks the ability to report error location. In later development Joint Test Action Group (JTAG) was found to be capable of identifying error location as well as providing register values to a PC.

3. Design Verification

Shown in Appendix A, Table 4 lists the requirements and verifications discussed in this chapter.

3.1 Power Unit

Each component of the power unit was tested separately on a breadboard. However, in this section we will discuss the results obtained by testing the components on our designed PCB.

3.1.1 Buck Converters

Buck converter efficiency is defined in Equation 12.

$$\eta = \frac{P_{OUT}}{P_{IN}} = \frac{V_{OUT} \times I_{OUT}}{V_{IN} \times I_{IN}} \tag{12}$$

Shown in Figure 3 the input to the buck converter is shown as V_I and output is shown as V_O. To satisfy Requirement 1, 2, and 3, buck converters were tested under the load. At V_I power supply was set to deliver 7.4 V. At V_O the programmable load was set to deliver a current in range from 250 mA to 600 mA. The results are shown in Figure 6.

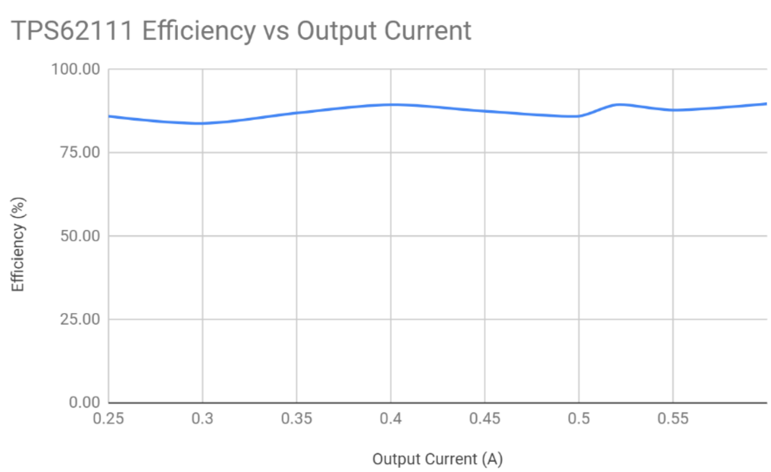

Figure 6 TPS62111 Efficiency vs Output Current

Based on results obtained for TPS 62111, the requirement 1 was satisfied. Since TPS 62111 converter feeds power to TPS 62002 converter, power supply was set to deliver 7.4 V at the V_I of the 62111 converter. At V_O of 62002 converter, the programmable load was set to deliver a current in range from 250 mA to 600 mA. The results are shown in Figure 6. Since 62002 provides power to the microcontroller, the maximum usage falls slightly below 400 mA, therefore an efficiency at the peak performance is rated at 70 %. Based on test results shown in Figure 7, TPS 62002 did not satisfied requirement 1.

The oscilloscope probes were placed at the output of TPS 62111 converter, which is also an input of the TPS 62002, and at the output of TPS 62002 converter to measure the output voltage ripple shown in Figure 8. The blue waveform shows the output of the 62111 converter, which averages around 3.31 V, which satisfies requirement 2. The yellow waveform is the output of the 62002 converter where minimum and maximum voltage variation is too high to satisfy requirement 3.

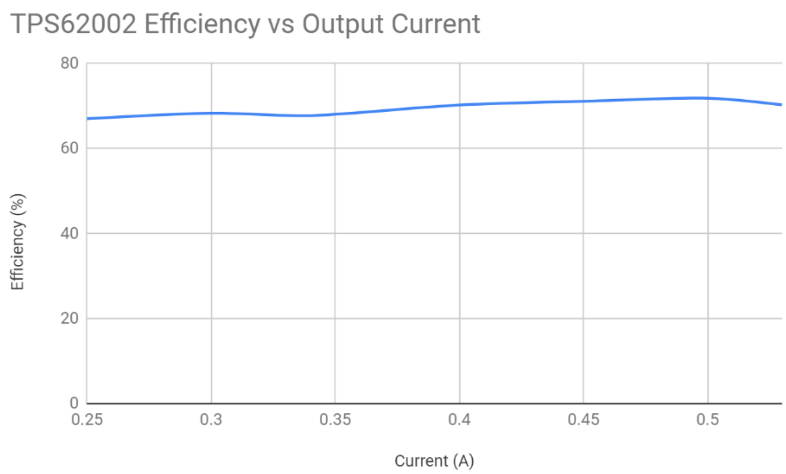

Figure 7 TPS 62002 Efficiency vs Output Current

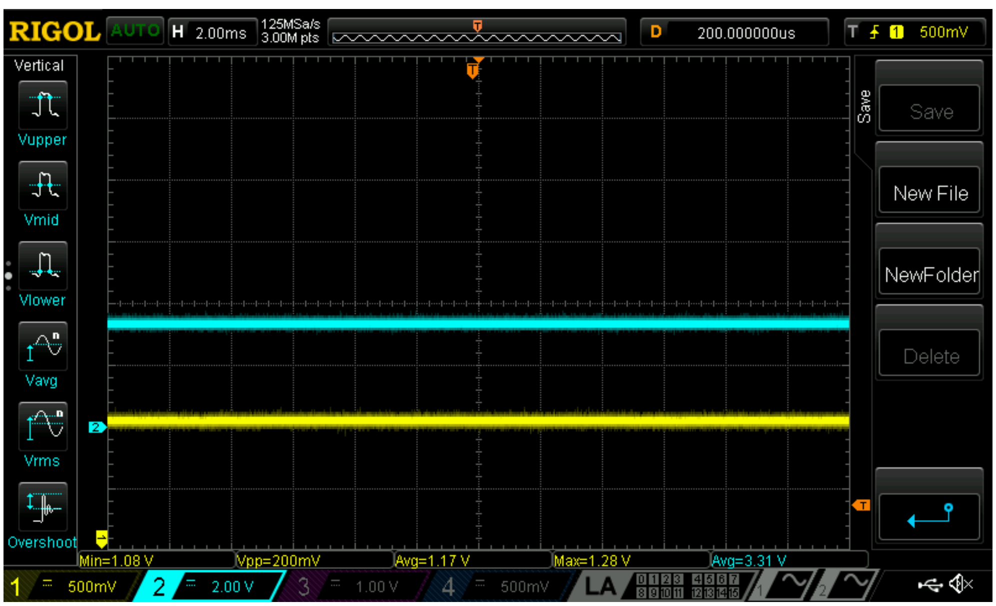

Figure 8 TPS 62002 Ripple Measurement

Failed requirement 1 and 3 for TPS 62002 converter requires more testing with different values for input and output capacitors. We believe that it was a poor design choice to feed the output of one buck converter to the next, instead of feeding the power directly from the battery.

3.1.2 Hot Swap Controllers

Each hot swap controller on the power board was tested separately. Referring to Figure 4, power supply delivered power to V_{IN} and a programmable load was connected to V_{OUT}. Requirement 4 was verified by setting the programmable load to a value over 1 A for requirement 4A, and over 2 A for requirement 4A. Both requirements were successfully verified by the oscilloscope probe at the V_{OUT}. Figure 9 shows the controller under a load over 1 A. Based on the capacitor value described in the design chapter, the controller allows the current to pass for 0.86 ms before the fault timer is exceeded and controller shuts down.

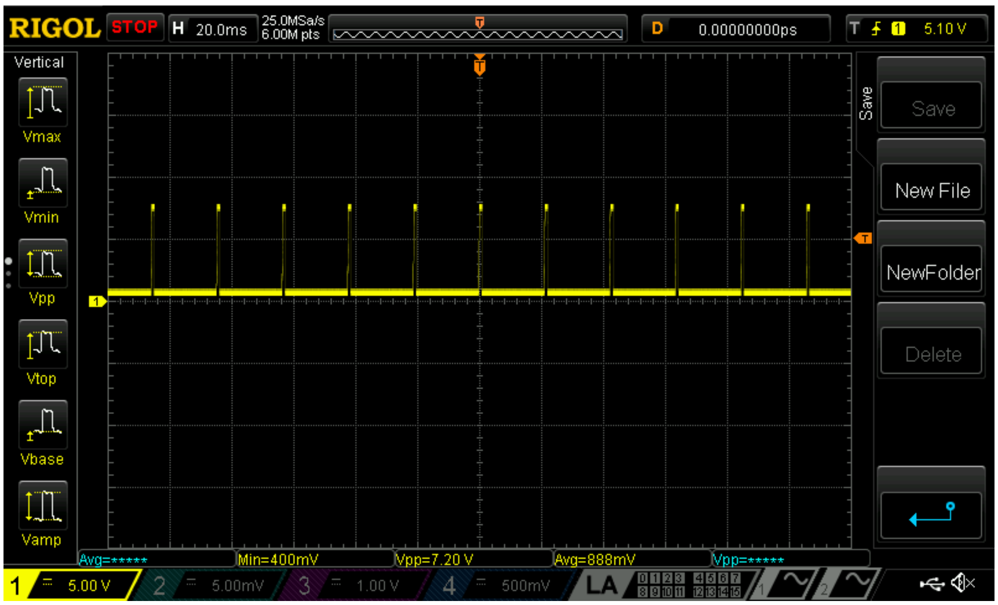

Figure 9 Hot Swap Trip Example

Unfortunately, requirement 5 was not satisfied by both hot swap controllers. Separate unit tests on breadboards were successful, however the PCB we designed had a standard width traces of 0.25 mm and that caused this problem that was later verified by KiCAD trace width calculator. The trace width for both controllers should be more than 0.3 mm.

3.2 Control Unit

3.2.1 Microcontroller

Requirement 8 discusses the microcontroller ability to enable and disable hot swap controllers through GPIO pins. During testing, the microcontroller toggled GPIO pin 2 and pin 5 output every couple of seconds to verify the controllers are switching on and off. For example, when GPIO pin generated a low signal, the hot swap controller was enabled, and vice versa. The functionality was tested by probing the output pins of the controllers and measuring the voltage.

The microcontroller should communicate with the battery pack using I2C as described in requirement 6. The communication with other satellite subsystems is performed over CAN bus as described in requirement 7. We performed the verification of communication protocols using an Arduino Uno equipped with a CAN bus shield to emulate a CAN bus and to test I^2C communication. I^2C protocol is very simple and easy to use, therefore Arduino I2C pins connect directly to microcontroller I^2C pins as shown in Figure 10. I^2C consists of two wires, a data wire (SDA) and a clock wire (SCL). Pull up resistors are required to enable the transmission, which value is specified in the microcontroller datasheet.

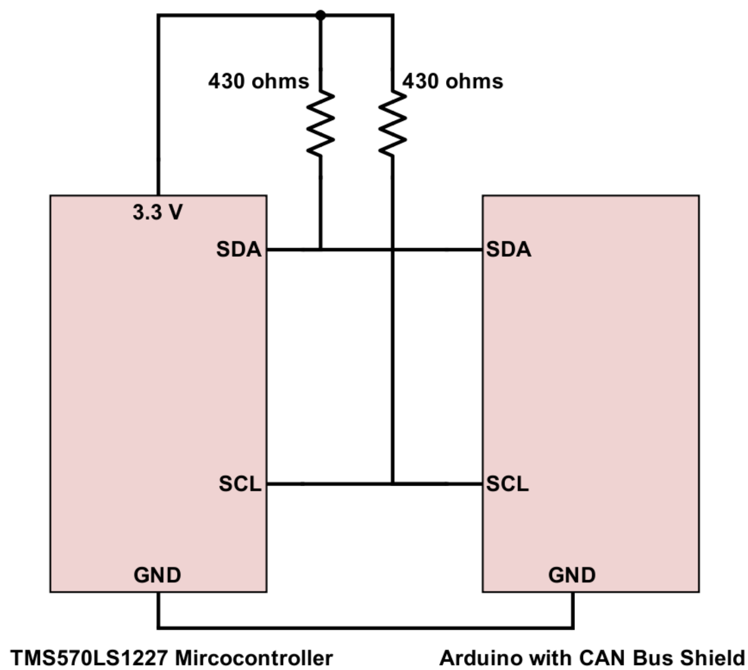

Figure 10 I2C Communication Connection Diagram

For testing purposes, a CAN transceiver resided in between an Arduino and the microcontroller. Since CAN bus has no clock available on the bus, it requires termination by a 120 Ω resistor to prevent reflection and oscillation on bit edge as shown in Figure 11.

Both communication protocols consist of a header, a data block, a checksum and acknowledge bits. In our design, the microcontroller successfully sent header of a data packet, but a data block was not transmitted. This behavior indicated that connection between the Arduino Uno and the microcontroller was established. Through investigation on the microcontroller side, we determined that a data block was not stored in the data registers after the connection was established. Data registers were used in communication protocols to store incoming or outgoing data. In the microcontroller data block was handles using SPI. One possibility is that SPI is synchronized, while CAN and I^2C communications are asynchronized. Another explanation is that different clock speed causes misinterpretation of data. The SPI uses 16 bits for data, while CAN and I^2C use 32 bits, for which the message would end up truncated in the data register.

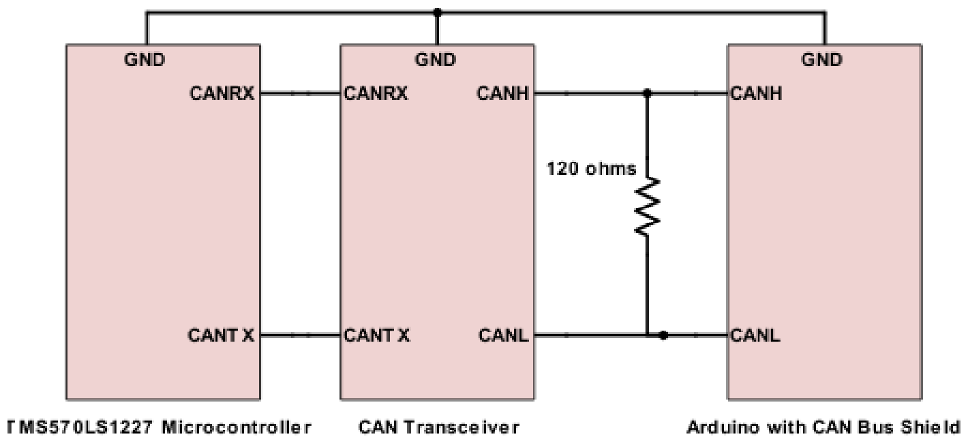

Figure 11 CAN Communication Connection Diagram

3.2.2 CAN Transceiver

The CAN transceiver must be able to decode messages from the CAN bus into format recognized by the microcontroller and encode messages from microcontroller which would satisfy requirement 9. During testing, we verified CAN transceiver functionality by sending a message from Arduino Uno and probing the CAN receive pin on the transceiver. Shown in Figure 12 is the CAN bus message, and as shown in Figure 13 the message was successfully decoded.

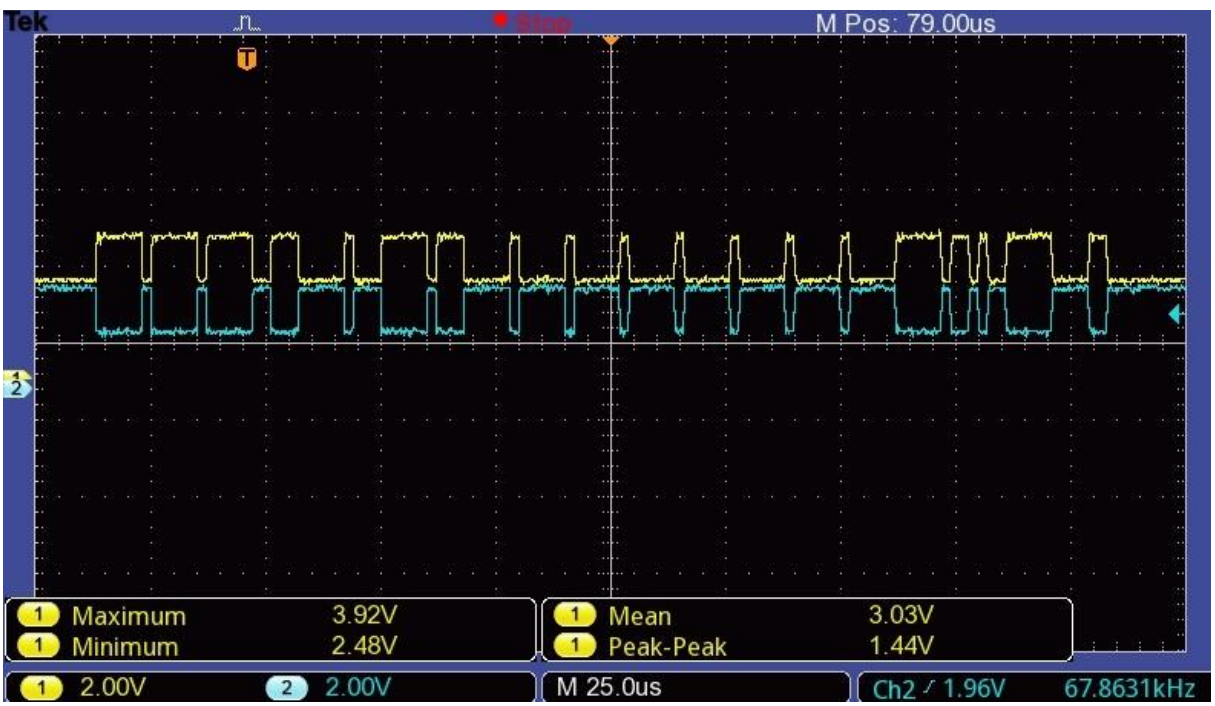

Figure 12 CAN Bus Message

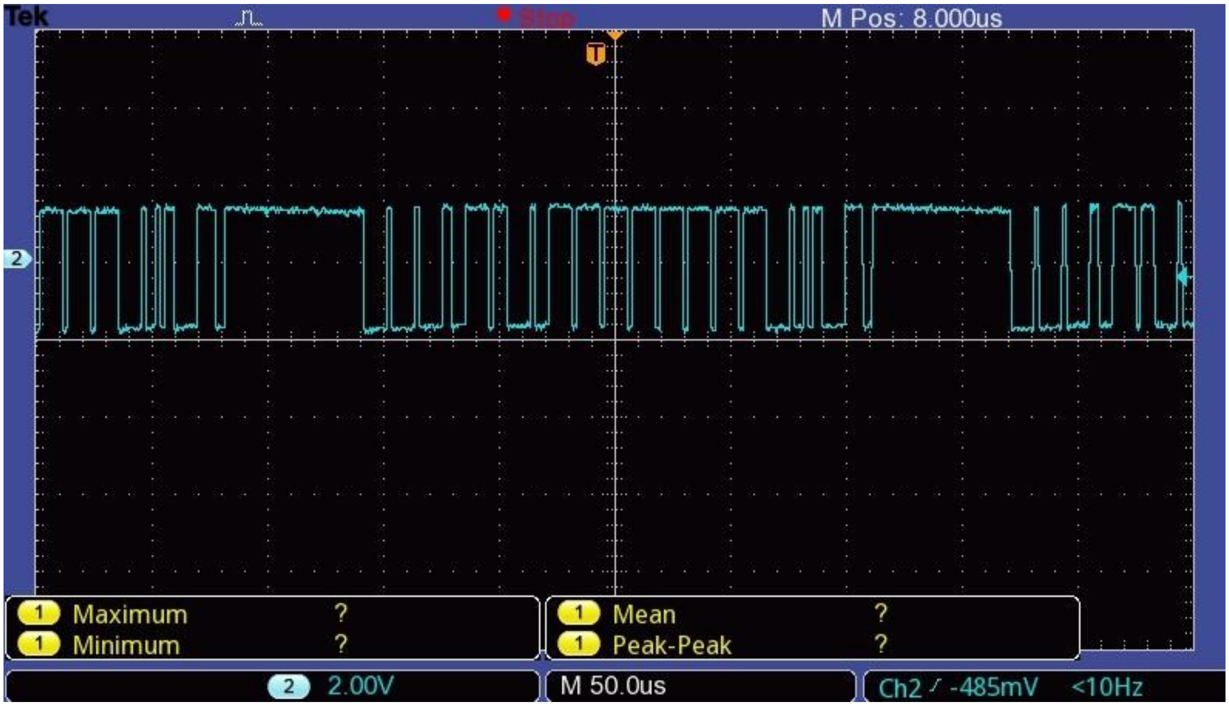

Figure 13 Output Message from CAN Transceiver

3.2.3 Watchdog Timer

The watchdog timer should reset microcontroller main program when program encounters faults or enters an infinite loop which satisfies requirement 11. The watchdog should send a reset signal to nRST pin whenever microcontroller fails to service the watchdog as per requirement 12. The implementation was verified using the microcontroller development board, which has nRST pin connected to one LED's for indication of a successfully send signal. During our tests, the watchdog performed as expected, and send the reset signal as indicated by nRST LED.

3.2.4 External Non-Volatile Memory

The external non-volatile memory (FRAM) must load the program into the microcontroller during the startup which satisfies requirement 10. Implementation of external memory failed because microcontroller required preinstalled bootloader to read the program from an external FRAM chip. The FRAM chip does not have the capability to indicate when to load the program into microcontroller. Currently the microcontroller has enough memory space to store its program. Lack of an FRAM chip in the design exposes risks such as data losses from radiation and magnetic fields.

4. Cost

4.1 Parts

Table 2 shows the cost of the parts used in the project.

Table 2 Parts Cost

Part	Manufacturer	Retail Cost ($)	Bulk Purchase Cost ($)	Actual Cost ($)
TMS570LS1227 Microcontroller	Texas Instruments	23.95	18.77	0
TPS62003 Buck Converter	Texas Instruments	1.78	1.31	0
TPS62111 Buck Converter	Texas Instruments	1.91	1.4	0
TPS2420 Hot Swap Controller	Texas Instruments	2.33	1.71	0
SN65HVD232 CAN Transceiver	Texas Instruments	2.13	1.54	2.13
LT3652 Solar Power Charger	Analog Device	6.66	3.77	6.66
SN65HVD379 RS422 Chip	Texas Instruments	4.47	3.64	4.47
TPS3850G09QDRCRQ1 Watchdog Timer	Texas Instruments	2.1	1.52	2.1
FM25V05-G FRAM	Cypress Semiconductor Corp	13.89	10.97	13.89
Total		**59.22**	**44.63**	**29.25**

4.2 Labor

For newly graduated engineers, the hourly salary ranged from $30 to $60. We choose the median of this range as our hourly salary. The labor cost of designing and prototyping the power board would be calculated as the following:

2 x $45/hour x 2.5 x 10 hours/week x 16 weeks = **$36,000**

5. Schedule

Table 3 Schedule

Week	Dongze	Robert
9/10/2018	Researching RTOS Block diagram design	Initial circuit design Block diagram design
9/17/2018	Researching parts	Verify block diagram Finalize parts, requirements and functionality
9/24/2018	Check RTOS specified requirements	Design documents
10/1/2018	Design documents Preparing for design review	Design documents Preparing for design review
10/8/2018	Developing system structure for the microcontroller on the development board	Implementing PCB design
10/15/2018	Programming required functionalities on the microcontroller using the development board	Verify PCB design
10/22/2018	Same as last week	Same as last week
10/29/2018	Work on I2C, CAN communication	Verify correctness of circuitry Solder components on the PCB board
11/5/2018	Fixing bugs on microcontroller system	Test PCB functionalities
11/12/2018	Test all functionalities of the microcontroller	Fixing problems associating with the power subsystem
11/19/2018 (Prototype should be finished)	Bug fixing the entire system	Bug fixing the entire system
11/26/2018	Prepare for presentation Minor changes based on mock demo feedback	Minor changes based on mock demo feedback
12/3/2018	Prepare for presentation Drafting the final report	Prepare for presentation Drafting the final report
12/10/2018	Finish final report	Finish final report

6. Conclusion

6.1 Accomplishments

One of the buck converters was successfully implemented to perform as expected. The correct components based on calculations from section 2.1.1 Buck Converters were selected at the design phase of the project. The power was successfully delivered to the input of the second buck converter within satisfactory voltage range. The hot swap controllers also performed as expected, neglecting the fact that the trace width was inadequate to successfully deliver a correct output voltage.

The microcontroller successfully controlled the operation of hot swap controllers using control signals generated from GPIO pins. A watchdog timer successfully performed microcontroller reset and send a reset signal to nRST pin, which indicated successful reset. A CAN transceiver encoded data packets from the CAN bus as well as data packets from the microcontroller, which were converted into CAN message format. Overall, the control unit successfully controlled components in the power unit.

6.2 Uncertainties

Several uncertainties occurred at one of the buck converters in the power unit. The converter failed to provide an output voltage within satisfactory voltage range as seen in Figure 8. Despite choosing the correct values for inductor and capacitor values based on section 2.1.1 equations, the design failed. The hot swap controllers failed to deliver the voltage output shown in Equations 10 and 11, despite working as expected when tested on the breadboard as separate units.

The microcontroller failed to send and receive data packets using both communication protocols, I^2C and CAN. The failure can be explained by the microcontroller data register's inability to obtain data. Data register is handled in the microcontroller using Serial Peripheral Interface, which is compatible with I^2C and CAN. External Non-Volatile Memory was not integrated since the implementation required pre-installed bootloader in the microcontroller. The chip would not detect a startup behavior and reset of the microcontroller independently.

6.3 Ethical considerations

6.3.1 Safety

Our design will interact with a lithium ion battery pack. Although the battery pack has a safety mechanism such as over and under voltage protection and over current protection, we still need to be concerned with several lithium ion battery related safety issues. While storing the lithium ion battery packs, we need to assure the battery packet is away from combustible materials and within the environment with temperature between 5 ℃ to 20 ℃. These actions would prevent fire hazard relevant to the battery. We also should ensure a minimum of 50% capacity for the battery before long time storage. While using the charging or discharging the battery, we should prevent overcharge and over discharge, limiting the voltage within 4.2 V when charging, and minimum of 3 V when discharging [11]. Since our power board will be operating in outer atmosphere, temperature is a crucial safety factor to consider. The lithium ion battery charging below 0 Celsius will cause damage to metallic plating, the damage to metallic plating is permanent and can cause failure of the battery when charge in high rate

[12]. This is the main reason we have battery heater in our battery pack. On more concern regarding the lithium ion battery is that faulty charger would damage battery circuit protection [12]. When we test and building our power board, we need to avoid charging the battery with higher voltage than the permitted range. In our design, we plan on utilizing the DC-DC converter which will effectively regulate voltage passing from the battery pack through the power board to other integrated boards. It is important to provide adequate isolation/space for the DC-DC converter such that other parts of the system have minimum interference. Adequate isolation could protect the converter from high-energy transient which most likely will damage the converter [13].

6.3.2 Ethics

Our design is the redesign and improvement on the old power board used in the previous version of the IlliniSat bus. In the new design, several integrated parts such as battery packs and battery heaters will remain unchanged. In our report and design process, we need to clearly identify the unchanged parts and give credits to their designer/author, following the ACM Code of Ethics 1.5 and IEEE Code of Ethics #7, "Respect the work required to produce new ideas......" [14] and "......to credit properly the contributions of others" [15].

Our work involves understanding and building upon many engineers' work before our project. It is important that we do not share their work publicly and use their work to obtain benefits without the prior approval and acknowledgement of the original author. IEEE Code of Ethics #2 clearly stated, "to avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties ......" [15]. While working on this project, we will receive advice from professors and graduate students. It is crucial to acknowledge their advice and criticism. This practice would help our team to recognize and correct errors and mistakes quickly and effectively. According to IEEE Code of Ethics #7, "to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors......" [15].

6.4 Future work

The project needs more work to become a fully functional prototype. The power unit needs to be redesigned to support wider traces for the hot swap controllers on the PCB. The buck converters should be fed from the battery separately to make sure they won't produce a high ripple output and damage electronics. The control unit needs a new design for the RS422 connector to make sure the prototype can connect to a PC. The communication protocols such as CAN and I^2C need to be fixed and integration of an external non-volatile memory is crucial for a successful prototype.

References

[1] NASA, "What are SmallSats and CubeSats?," NASA, 7 8 2017. [Online]. Available: https://www.nasa.gov/content/what-are-smallsats-and-cubesats. [Accessed 17 9 2018].

[2] M. Swartwout, "University-Class Spacecraft: From Marginal Utility to Disruptive Research Platfroms," Saint Louis University, 2018. [Online]. Available: https://nepp.nasa.gov/workshops/etw2016/talks/14TUE/20160614-0900-Swartwoutetw2016.pdf. [Accessed September 2018].

[3] e. Directory, "LAICE (Lower Atmosphere/Ionosphere Coupling Experiment," 2002. [Online]. Available: https://directory.eoportal.org/web/eoportal/satellite-missions/l/laice. [Accessed September 2018].

[4] T. Instruments, "5-A, 20-V Integrated FET Hot-Swap Controller," 2014. [Online]. Available: http://www.ti.com/lit/ds/symlink/tps2420.pdf. [Accessed September 2018].

[5] a. MBED, "Watchdog Timer," 2008. [Online]. Available: https://os.mbed.com/cookbook/WatchDog-Timer. [Accessed September 2018].

[6] T. Instruments, "FRAM FAQ," 2008. [Online]. Available: http://www.ti.com/pub/fram/fram_faq.html. [Accessed September 2018].

[7] T. Instruments, "TPS6211x 17-V, 1.5-A, Synchronous Step-Down Converter," 2015. [Online]. Available: http://www.ti.com/lit/ds/symlink/tps62110.pdf. [Accessed September 2018].

[8] T. Instruments, "TMS570LS1227 16- and 32-Bit RISC Flash Microcontroller," 2015. [Online]. Available: http://www.ti.com/lit/ds/symlink/tms570ls1227.pdf. [Accessed September 2018].

[9] Webopedia, "Lockstep processor technique," 2010. [Online]. Available: https://www.webopedia.com/TERM/L/lockstep_processor_technique.html. [Accessed September 2018].

[10] a. mBed, "Watchdog Timer," [Online]. Available: https://os.mbed.com/cookbook/WatchDog-Timer. [Accessed September 2018].

[11] University of Washington, "Lithium battery safety," [Online]. Available: https://www.ehs.washington.edu/system/files/resources/lithium-battery-safety.pdf. [Accessed September 2018].

[12] Battery University, "Lithium-ion safety concerns," [Online]. Available: https://batteryuniversity.com/learn/archive/lithium_ion_safety_concerns. [Accessed September

2018].

[13] Flex, "Market Requirements and System-Level Considerations," [Online]. Available: https://flex.com/sites/default/files/TP018-SafetyRequirements_Flex.pdf. [Accessed September 2018].

[14] ACM, "ACM Code of Ethics," [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed September 2018].

[15] IEEE, "IEEE Code of Ethics," [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed September 2018].

Appendix A Requirement and Verification Table

Table 4 System Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
1. Buck converters must achieve at least > 80 % efficiency	1. Test the efficiency under the load using equation 12.	Y
2. The output voltage of TPS62110 should fall between 3.3 V ± 0.3 V	2. Test the output voltage range under the load.	Y
3. The output voltage of TPS62002 should fall between 1.2 V ± 0.06 V	3. Test the output voltage range under the load	Y
4. The controller must shut down if the current exceeds a certain value a. Must shutdown when current through the device exceeds 1 A b. Must shutdown when current through the device exceeds 2 A	4. Test the shutdown mode by applying current that exceeds the current required to trip the controllers	Y
5. The controller must deliver output current at a certain value range a. Output voltage should fall between 7.36 V ± 0.1 V b. Output voltage should fall between 7.23 V ± 0.1 V	5. Test the output voltage range by applying a load like number 4 using a programmable load	Y
6. CAN transceiver must be able to send and receive data packets on Controller Area Network (CAN) bus	6. A mock CAN message sent from Arduino should be received by the microcontroller	N
7. The microcontroller must be able to communicate using Inter-Integrated Circuit (I2C) communication	7. Successfully receive temperature data from the battery pack and successfully able to operate battery heaters	N
8. The microcontroller must be able to control hot swap controllers using signals from General Purpose Input/output (GPIO pins)	8. Successfully control hot swap controllers	Y
9. The microcontroller must be able to encode and decode data packets on CAN bus	9. Successfully convert message into\from CAN bus data format	Y
10. Non-volatile memory chip must be able to load program into microcontroller during startup	10. Microcontroller runs program from external memory during startup	N
11. Watchdog timer successfully resets the program when the	11. Program restart after reset	Y

timer expires		
12. The microcontroller reset signal is send to the nRST pin	12. LED connecting to nRST pin turns on after reset is performed	Y