DESIGN AND IMPLEMENTATION OF A BACKPLANE ELECTRICAL SYSTEM FOR AN AUTONOMOUS UNDERWATER VEHICLE

Submitted by:
Hoang The Huan
A0092259X
Department of Electrical and Computer Engineering

In partial fulfilment of the requirements for the
Degree of Bachelor of Engineering
National University of Singapore

Abstract

The electrical system of Bumblebee Autonomous Underwater Vehicle version 2.0 has been tested extensively during pool tests and competitions from March 2014 till now. Several problems have been identified and analyzed for future improvements. As part of the development for Bumblebee Autonomous Underwater Vehicle version 3.0, an electrical backplane system is proposed which will incorporate changes to fix the existing problems, reduce the amount of wiring, enhancing reliability and testability.

This thesis discusses the existing problems in the electrical system of Bumblebee version 2.0, presents possible solutions for those problems as well as new features. It describes the development and testing methodologies of the solutions and new features. Next, the design considerations for incorporating the solutions and new features on the backplane are discussed. Finally, it presents the final electrical backplane product.

Acknowledgement

First and foremost, I would like to express my heartfelt gratitude towards my project supervisor, Professor Mandar Anil Chitre, for giving the opportunity to undertake my final year project under his supervision. Besides his constant guidance and encouragement, he has shared with me his expertise and insight which have greatly helped me complete this project.

In addition, I would like to express my gratitude towards the team members of Bumblebee Autonomous Underwater Vehicle team for providing moral support and technical assistance throughout this project.

Last but not least, I would like to thank the staff of Advanced Robotics Center and Control Mechatronics Laboratory 1 for allowing me to use the facilities as well as providing technical support for me.

Contents

List of Figures

List of Symbols and Abbreviations

AC	Alternating Current
ADC	Analog to Digital Converter
AHRS	Attitude Reference Heading System
AUV	Autonomous Underwater Vehicle
CAN	Controller Area Network
CANH	CAN high
CANL	CAN low
DC	Direct Current
DTR	Data Terminal Ready
DVL	Doppler Velocity Log
FPGA	Field-programmable Gate Array
LAN	Local Area Network
LED	Light Emitting Diode
Li-po	Lithium-polymer
MCU	Microcontroller
MOSFET	Metal-oxide Field Effect Transistor
NTC	Negative Thermal Coefficient
N-MOSFET	N-channel Metal-oxide Field Effect Transistor
P-MOSFET	P-channel Metal-oxide Field Effect Transistor
PCB	Printer Circuit Board
PoE	Power over Ethernet
RJ45	Registered Jack 45
RXD	Receive Data
SBC	Single Board Computer
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus

A	Ampere
Ah	Ampere hours
Hz	Hertz
s	second
V	Volt
V_{GS}	Gate-source voltage
W	Watts
Ω	Ohm

Chapter 1 Background of the Project

1.1 Bumblebee Autonomous Underwater Vehicle (AUV)

Bumblebee Autonomous Underwater Vehicle (BBAUV) team was founded in 2013 by 5 members from Mechanical Engineering department of National University of Singapore. The team's aim was to build AUV and compete in two international AUV competitions annually. They are the Singapore AUV challenge (SAUVC) and Robosub, held in San Diego, United States. Throughout the years, the team has developed 2 AUV systems. Figure 1 and figure 2 shows Bumblebee v1.0 and Bumblebee v2.0 respectively.

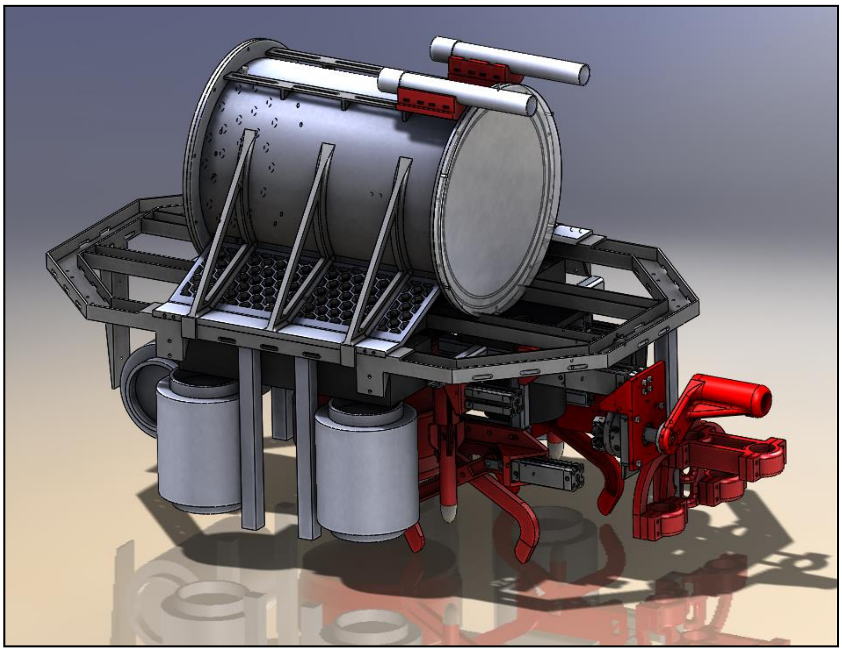


Figure 1: Bumblebee v1.0. Source: bbauv.com

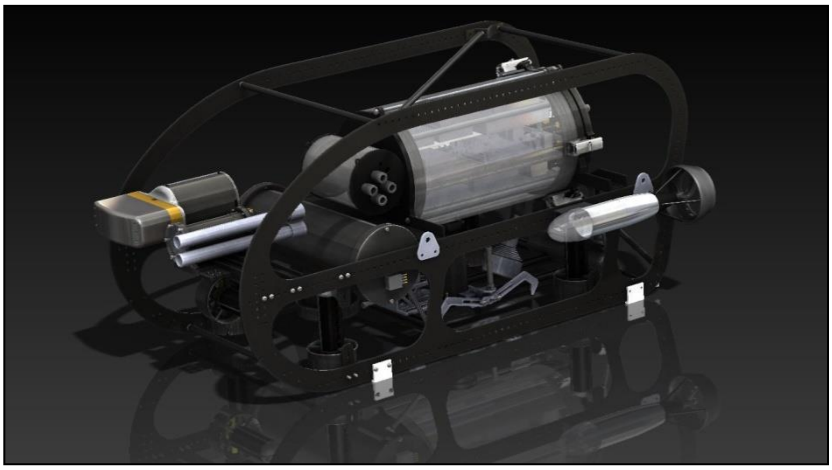


Figure 2: Bumblebee v2.0. Source: bbauv.com

Though the form of the AUV has changed from version 1 to version 2, there are still similarities in the two systems. In the center of the AUV lies the electronic housing (main hull). There are other enclosures outside of the electronic housing which house other sub-systems such as the solenoid manifolds, batteries, cameras, Doppler velocity log, thrusters, etc. To ensure connectivity between the main hull and the external enclosures, underwater connectors are used. Figure 3 and 4 show the electronic housing of Bumblebee v2.0 and the underwater connectors.

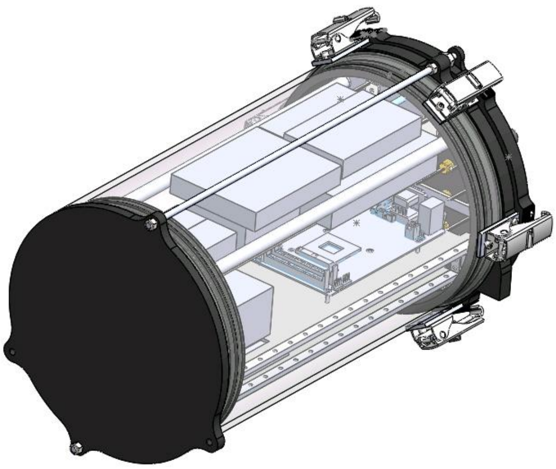


Figure 3: Bumblebee v2.0 electronic housing. Source: bbauv.com

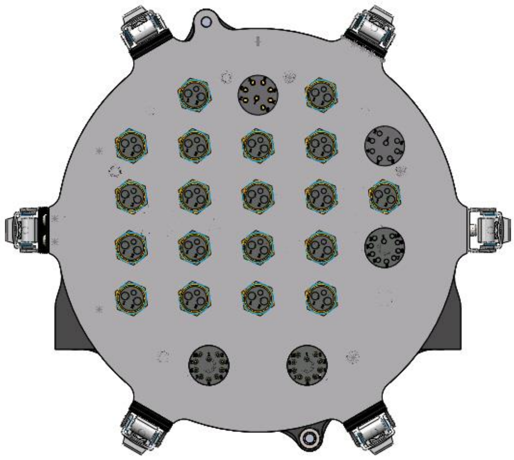


Figure 4: Underwater connectors on end cap of electronic housing. Source: bbauv.com

1.2 State of the current electrical system of the AUV

The current electrical system of Bumblebee v2.0 is described in the following block diagram.

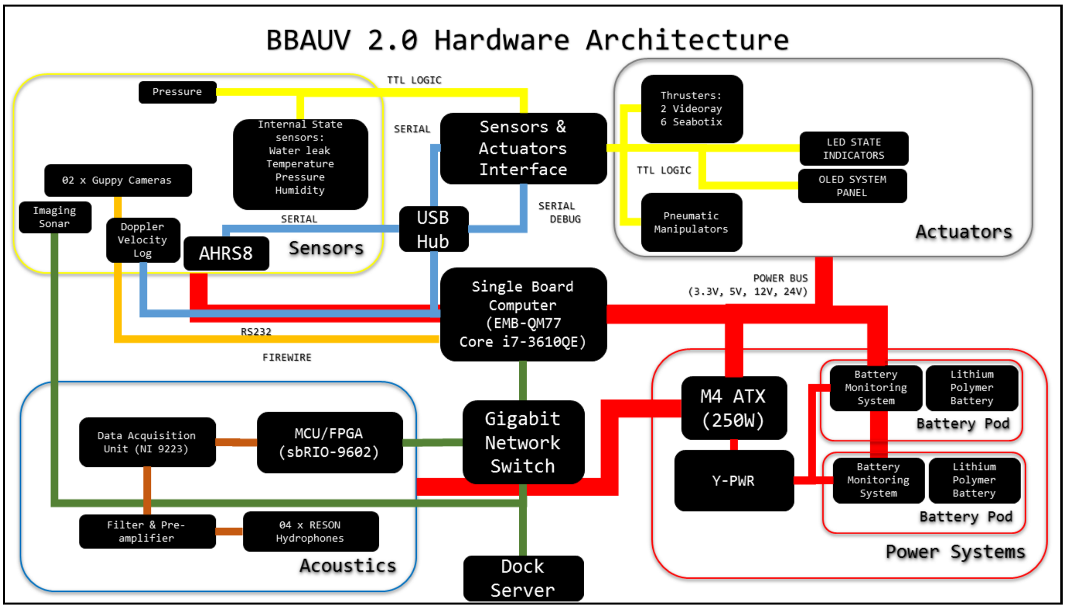


Figure 5: Bumblebee v2.0 electrical architecture. Source: bbauv.com

In terms of power, Bumblebee v2.0 runs on two Lithium-polymer (Li-po) batteries each with 7800mAh capacity. The thrusters take power directly from the voltage rail of the batteries. Out of 8 thrusters, 4 are powered by each battery. The M4-ATX takes in battery voltage level and regulates down to 12V and 5V rail for the consumption of the electronics on the system.

In terms of data connectivity, the Single Board Computer (SBC) is the central processing unit of the AUV. Together with peripherals such as the sensor & actuator board, Doppler velocity log (DVL), Attitude reference heading system-8 (AHRS-8) Inertia measurement unit, the SBC forms a star configuration. The SBC connects to these peripherals via serial links. Apart from the serial links, the SBC is also connected to a LAN network which connects the SBC, sbRIO-9602 Field-programmable gate array (FPGA) module, Blueview P450 sonar and a dock server together.

Chapter 2 Existing Problems

2.1 Unreliable wire connectivity

The current electronics housing of Bumblebee v2.0 is compact. Most of the components in the housing are off-the-shelf products. To integrate these components together, a lot of wires are used. These wires are cut to length and crimped to the appropriate connectors. Figure 6 and 7 show the current state of the electronic housing.

Figure 6: Bumblebee v2.0 electronics housing

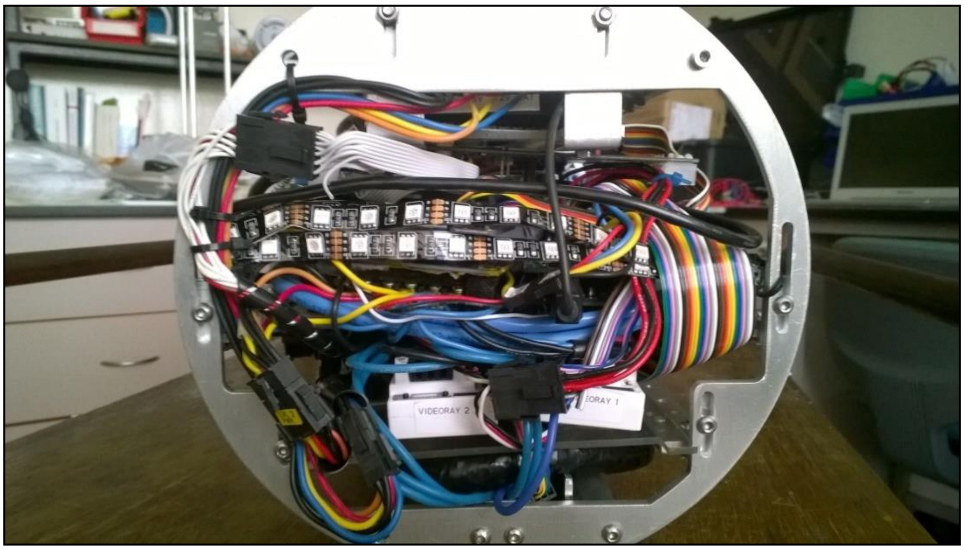


Figure 7: Bumblebee v2.0 electronics housing

Throughout the development and testing phase of Bumblebee v2.0, unreliable data and power connectivity have been experienced. Most of the time, problems arise from loose connections of wires and improper contacts of the wire crimps. The frequency at which the electronic rack is taken out and in from the housing also introduces mechanical impacts on the wires and the components, exacerbating connectivity problem. Figure 8 shows a burned

Molex connector; the loose connection introduces a high resistance which results in heating of the connector.

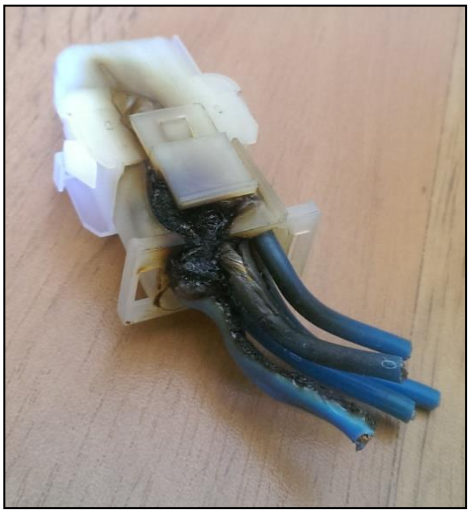


Figure 8: A burnt Molex connector in Bumblebee v2.0 electronic housing

The high amount of wires and connectors increases the susceptibility to failure of the system.

2.2 Power supply failure

As mentioned in the previous chapter, M4-ATX is a DC-DC power supply which is being used on Bumblebee v2.0. It takes in 24 V power from the batteries and regulates down to 12 V and 5 V for the consumption of other electronics.

Figure 9: M4-ATX DC-DC power supply. Source: http://www.mini-box.com/M4-ATX

Through intensive testing of Bumblebee v2.0, the M4-ATX has been known to fail unpredictably. When M4-ATX fails, the entire AUV system is down. This is a critical issue as power failure leads to disruption during testing and competitions.

2.3 Difficulty in hardware debugging and maintenance

Due to the high amount of wirings in the electronics housing, it is difficult to troubleshoot the system. Firstly, it is difficult to access electrical points in the system due to the compactness. Secondly, it is difficult to identify a suspected sub-system for further troubleshooting. Furthermore, during troubleshooting, many connections need to be removed; the repetitive mating and disconnecting action results in wear and tear of the connectors.

Without proper documentation of the wire connections, disassembly and assembly of the electrical system is troublesome and time-consuming. It is especially difficult for new members of the team to understand the system and troubleshoot it without the knowledge of the seniors.

2.4 Inrush current

Inrush current is a transient peak in current that happen when an electronic system is powered up. Inrush current happens because input filtering capacitors of an electronic system act as short circuits when power is first applied across them. In-rush current has degrading effect over the connectors, capacitors and batteries due to the high current magnitude. In-rush current can be several orders of that of normal operating current.

A set-up was arranged in order to measure the inrush current when a 6-cell Li-po battery is connected to the electrical system.

Figure 10: Set-up to measure the inrush current into the system

A 0.33 Ω power shunt resistor is connected in series with the AUV electrical system load in order to measure the current. The voltage across the shunt resistor is measured by an oscilloscope. The oscilloscope was changed to single mode to capture rising edge changes in the voltage signal. Figure 11 shows the inrush voltage across the shunt resistor captured by the oscilloscope.

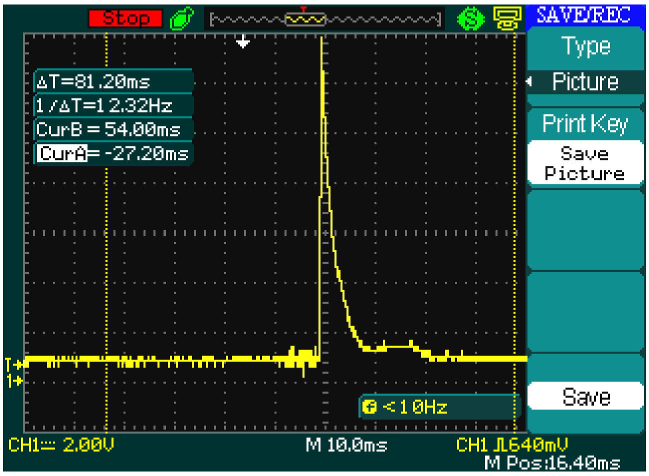


Figure 11: Inrush current registered on an oscilloscope

The peak voltage current across the shunt resistor is measured to be 13.596 V. Hence, the peak inrush current registered is:

$$I_{inrush} = \frac{V_{shunt}}{R_{shunt}} = \frac{13.596\ V}{0.33\ \Omega} = 41.2\ A$$

The inrush current is 27 times the normal operating current, which is 1.5 A.

2.5 Surge over serial data line

As mentioned in chapter one, the SBC communicates with a few other peripherals via serial links. The peripherals are connected to a USB hub via USB-UART converter. It has been observed that USB-UART converters that are connected to the battery monitoring system are damaged when the batteries are connected into the AUV system. A few USB-UART converters have been replaced in order to keep the battery monitoring system functioning.

Figure 12: A damaged USB-UART converter

Chapter 3 Solutions and Extra Features Testing

In this chapter, the possible solutions to the existing problems of the electrical system of Bumblebee v2.0 are discussed. Prototypes of the possible solutions are built and tested.

Extra features such as voltage and current measurement are also explored to facilitate debugging of the electrical system.

A backplane electrical system is the solution to the unreliable connectivity problem. Having an electrical backplane also allows the solutions and extra features to be incorporated in a unified platform. Hence, the prototypes of the solutions and extra features are made in such a way that they simulate as close as possible to how they would be incorporated on the backplane.

3.1 DC power input soft-start

As mentioned earlier in section 2.4, in-rush current has been detected in the current electrical system of the AUV. In the case of our existing AUV system, in-rush current upon applying battery to the system is measured to be 41.2 A maximum, as compared to 1.5A normal operating current. This inrush current can last up to 10ms.

There are various methods to dampen or eliminate in-rush current into a system. The most common method is to have Negative Thermal Coefficient (NTC) thermistor in series with the load. This method is described in the following schematic.

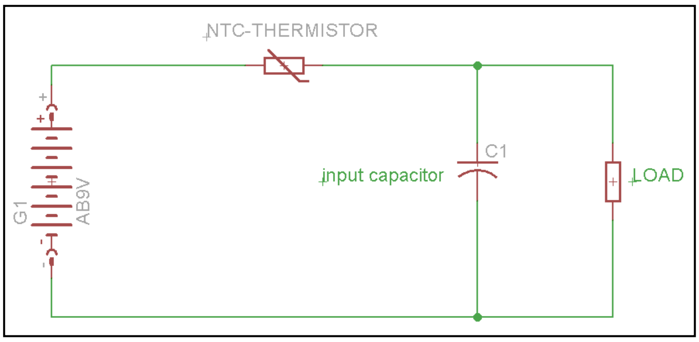


Figure 13: Negative Thermal Coefficient Thermistor inrush current limiter

When a battery is first connected to the system, NTC thermistor's temperature is low and therefore its resistance is high. This high resistance reduces the initial inrush current. As current start to conduct through the NTC thermistor, its temperature rises and its resistance thus decreases, allowing current to rise to the normal operating magnitude.

In steady state, however, the resistance of NTC is non zero. CL-60 NTC inrush current limiter has 0.18 Ω of resistance at 5 A.

$$P_{loss} = {I_{NTC}}^2 \times R_{NTC} = 5^2 \times 0.18 = 4.5\ \mathrm{W}$$

Power dissipation is therefore very substantial especially in high DC current application. An alternative method is to use a MOSFET-based soft-start circuitry. MOSFFETs act as a switch which can be turned on slowly by gradually increasing V_{GS} gate to source voltage. This can be done by charging a capacitor connected between the gate and the source of a MOSFET. Below is a low-side N-MOSFET slow start circuit for DC application:

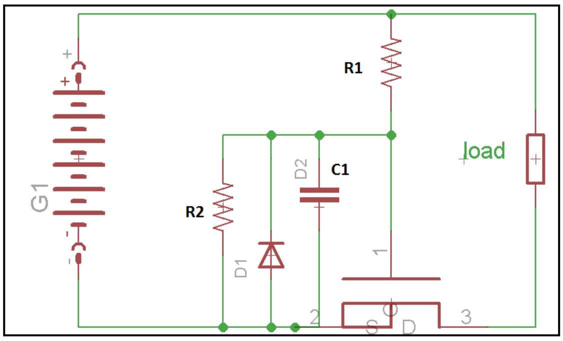


Figure 14: N-channel MOSFET based soft-start circuitry

The drain-source resistance of IRLR3636pbf N-MOSFET is 6.8 mΩ at maximum. The power dissipation at a drain current of 5 A is:

$$P_{loss} = {I_D}^2 \times R_{DS,on} = 5^2 \times 0.0068 = 0.17\ \mathrm{W}$$

This is negligible as compared to 4.5 Watts as in the case of using a CL-60 NTC.

Resistor R6 and capacitor C1 form a RC charging circuitry. The time constant of this RC circuit is given by $\tau = R \times C$, defined as the time taken for the voltage of the capacitor to reach 63% of the maximum voltage across itself.

To effectively control the inrush current, the time constant of the RC circuit should be higher than 10ms. Choosing τ to be 1 s, and R1 to be 1 MΩ:

$$C1 = \tau \div R1 = 1 \div 10^6 = 1\mu F$$

R2 is then chosen such that the R1 and R2 voltage divider network would generate a voltage greater than the clamping voltage of the zener diode when the diode is not in place. Choosing R2 to be 1 MΩ, voltage at the top of zener diode would be:

$$V_{divided} = 24V \div 2 = 12V$$

The clamping voltage of Zener diode is chose to be 10V. This voltage determines the V_{GS} of the N-MOSFET. At $V_{GS} = 10V$, IRLR3636pbf is completely open and acts as a short circuit and is in its linear zone.

With the values of the passive components decided, a prototype of the soft-start circuitry was made. The module is as seen in the figure 15.

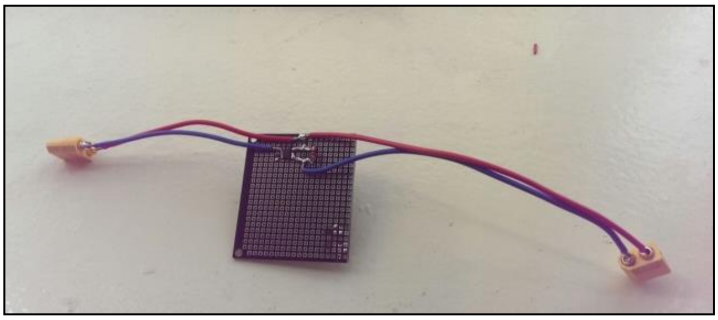


Figure 15: Soft-start prototype module

Figure 16: Soft-start module connected to the AUV

Figure 16 shows a set-up to test the soft-start prototype. A pack of Li-po battery is connected in series to a shunt resistor, then in parallel with the soft-start prototype and in parallel with the AUV load. The electrical connection is shown in the following schematic.

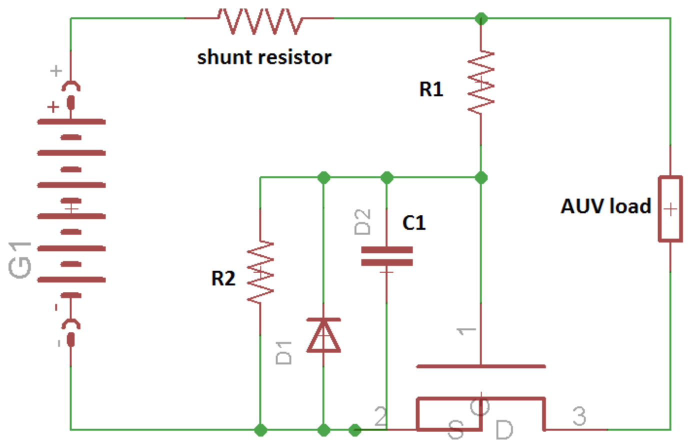


Figure 17: Connection of the soft-start prototype to existing AUV system

Upon plugging in the battery to this soft-start and AUV load system, the voltages across the shunt resistor is measured by an oscilloscope to verify that the inrush current is suppressed. The screenshot of oscilloscope is shown in figure 18.

Figure 18: In-rush current suppressed with soft-start prototype

As compared to figure 11, no high peak is seen in the voltage reading of the shunt resistor. The small peaks observed in figure 18 are due to the start-up current drawn by the start-up buzzer sequence in the AUV system.

3.2 DC power path switcher in load sharing mode

As there is a need for balancing the two batteries on the AUV system, there must be a mechanism to balance out the load incurred by each battery. In order to achieve that, dedicated power path switcher IC LTC4416 from Linear Technology is employed. In load sharing mode, the LTC4416 compares the two DC inputs' voltages and connect the output to the input with higher voltage. When two DC inputs are within 100 mV difference, both DC inputs are connected to the output. LTC4416 switches each DC input on and off by controlling the gate of P-MOSFETs. The schematic of LTC4416 and P-MOSFETs in this application is shown in figure 19.

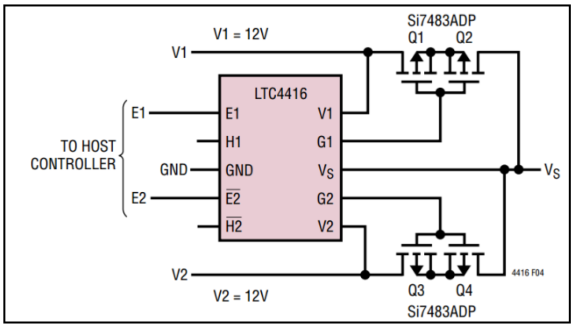


Figure 19: LTC4416 load sharing configuration. Source:

http://cds.linear.com/docs/en/datasheet/4416fa.pdf

The P-MOSFETs on both channels are in the drain to drain configuration in order to bi-directionally block DC current flows in the off state.

Based on this application schematic, a Printed Circuit Board was designed to test out the functionality of LTC4416 in load sharing configuration. The prototype is as shown in figure 20.

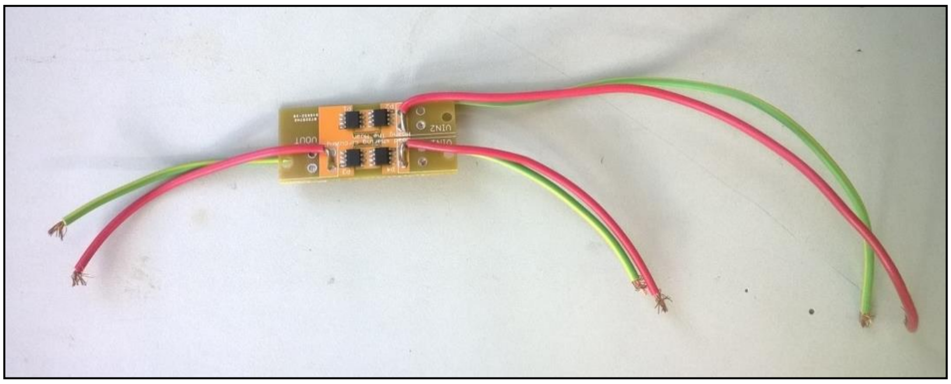


Figure 20: LTC4416 load sharing PCB prototype.

The LTC4416 load sharing prototype is then tested. Two DC inputs are simulated using two power supply. The output of the prototype is connected to an oscilloscope to measure the output voltage. The prototype is evaluated by varying the input voltages and examining the output voltage. It was observed that the output voltage vary correctly as expected, i.e. output voltage is always the higher DC input. Figure 21 shows an oscilloscope screenshot when testing the load sharing prototype.

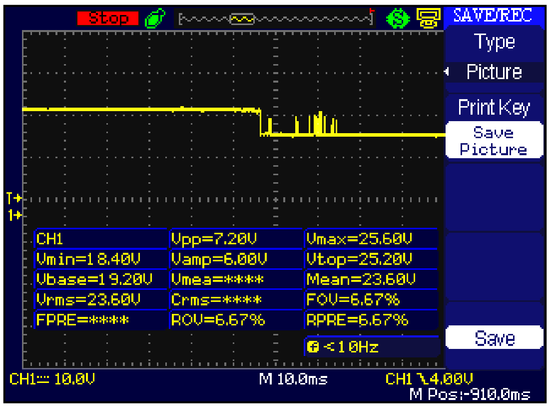


Figure 21: Output voltage of LTC4416 load sharing prototype during switching.

3.3 Buck converter (step-down voltage regulator)

As mentioned in part 2.2, there is frequent power failure from the M4-ATX power supply that is currently being used in our AUV electrical system. Furthermore, there is a need for a stable 5 V voltage rail on the electrical backplane so that voltage and current monitoring can happen without disruption in the event of a failure of the M4-ATX.

The input battery voltage can be stepped down to provide a 5 V voltage rail on the backplane. In order to do this, a buck converter circuitry is employed. A typical buck switching regulator circuitry is shown in figure 22.

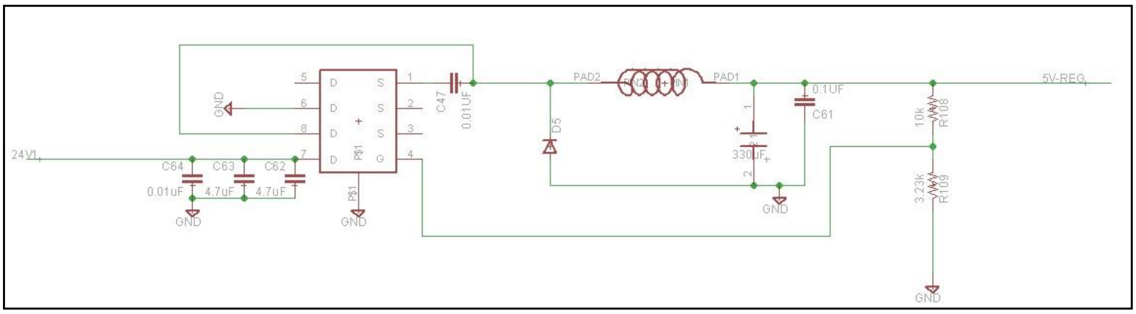


Figure 22: A typical buck regulator switching circuit.

TPS5450 is chosen as the switching regulator IC as it can support up to 5A of continuous current drawn. Along with TPS5450, SRR1280A-150M inductor and B540C-13-F are chosen as they can support over 5 A.

The output voltage is configured by having a voltage divider network which feedback the output voltage to TPS5450. The resistor values are calculated based on the formula

$$R_2 = \frac{R_1 \times 1.221}{V_{out} - 1.221}$$

where 1.221 V is the internal voltage reference of TPS5450, V_{out} is the desired output voltage. Letting R1 = 10 kΩ, V_{out} = 5V, we have R2 = 3.23 kΩ. The circuitry was tested on a breadboard as shown in figure 23.

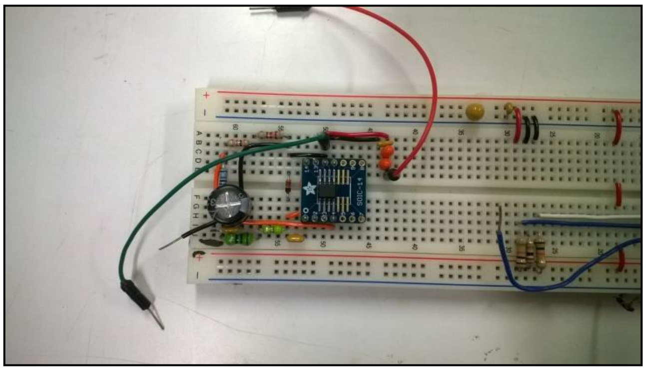


Figure 23: TPS5450 buck regulator prototype

The inductor used in the prototype is of lower current rating. The prototype was just to verify that a 5V voltage rail can be generated at the output. No load current testing was carried out.

3.4 DC power path switcher in back-up DC mode

In normal operating condition, all daughter boards and electronics on backplane will draw power from the 5V rail provided by the M4-ATX. In the case of power failure of M4-ATX, the buck regulator needs to provide back-up power. Hence, there is a need for a power path switcher. The switcher needs to detect a fall in the voltage in the primary DC source and quickly switch over to the secondary DC source. LTC4416 is capable of this application. The schematic of LTC4416 and P-MOSFETs for this application is shown in figure 24.

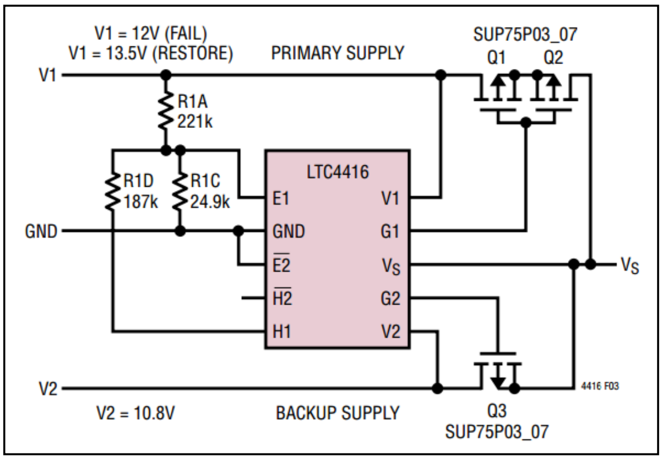


Figure 24: LTC4416 as primary/secondary power path switcher. Source: http://cds.linear.com/docs/en/datasheet/4416fa.pdf.

Primary DC input, in this case the 5V voltage rail from the M4-ATX, is feedback to the enable pin E1 of LTC4416. The feedback is connected to E1 via a resistor divider network of R1A and R1C. R1D is connected from E1 to pin H1. In normal operating condition, voltage of the primary DC source is divided by R1A and R1C and feedback to E1, R1A and R1C are calculated based on the following application equation:

$$V_{fail} = V_{eth} \times \frac{R_{1A} + R_{1C}}{R_{1C}}$$

where $V_{eth} = 1.222$ V which is the internal voltage reference of LTC4416. Choosing $R_{1A} = 51$ kΩ , $V_{fail} = 4.34$ V we have $R_{1C} = 20$ kΩ.

When primary DC source voltage drops below 4.34, voltage at E1 drops below V_{eth} , LTC4416 activates the secondary DC source channel by outputting appropriate voltage at G2 to open the P-MOSFET on secondary DC voltage line. At the same time, H2 goes from high impedance to low, connecting to ground. This effectively transforms the resistor divider network into R1A in series with R1C//R1D. The restore voltage is given by:

$$V_{restore} = V_{eth} \times \frac{R_{1A} + R_{1C}//R_{1D}}{R_{1C}//R_{1D}}$$

Choosing $V_{restore} = 4.8$ V, we have $R_{1D} = 120$ kΩ. The final schematic is shown in the figure 25. The addition of a LED connected from secondary 5V to pin H1 is a visual indication when M4-ATX 5V voltage rail fails. A 100µF capacitor is added at the 5V output to smooth out any transient voltages during switching.

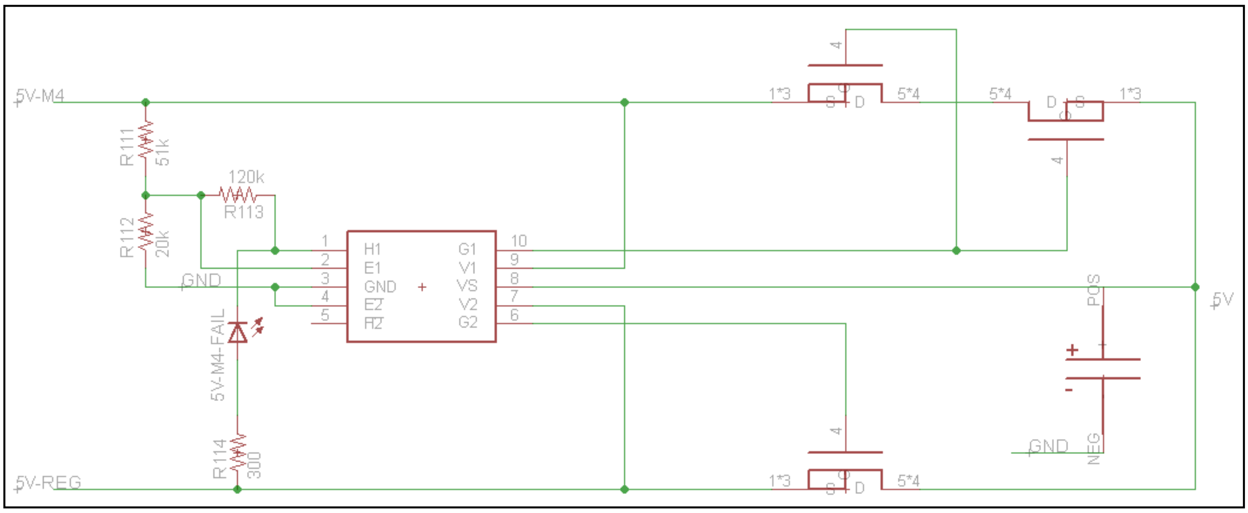


Figure 25: Complete LTC4416 as back-up power path switcher

There was no prototype built to test this circuitry. However, it was tested on the backplane PCB after it was fabricated. The LTC4416 back-up power path switcher on backplane PCB is shown in figure 26.

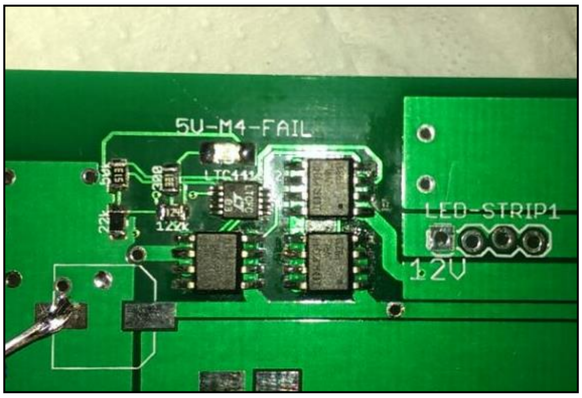


Figure 26: LTC4416 and P-MOSFETs in back-up power path switcher configuration

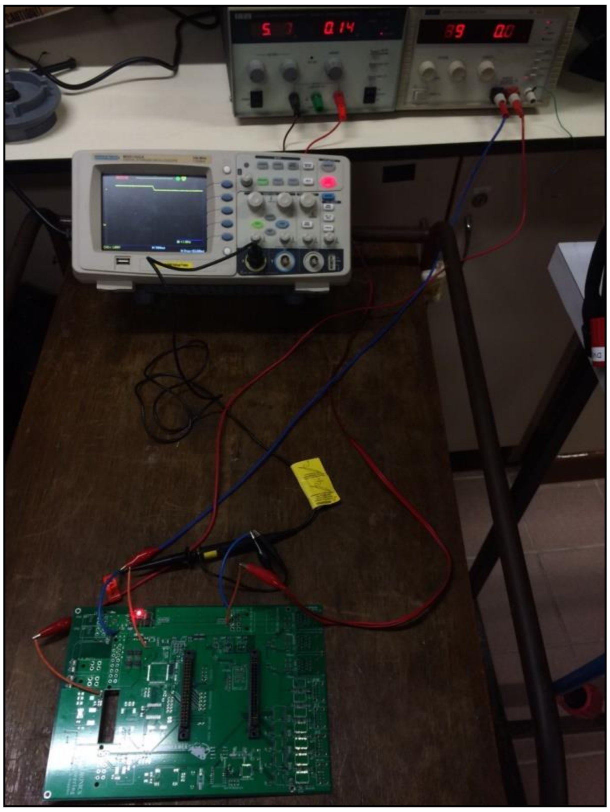


Figure 27: Test set up for LTC4416 as back-up power path switcher

In the test setup for LTC4416 as back-up power path switcher, two power supplies were used to simulate the primary and secondary 5 V DC sources. The output voltage is measured by an oscilloscope. By varying the primary DC voltage source to below V_{fail}, LED turns on, this indicates that the primary supply voltage is failing. Correspondingly, the output voltage on the oscilloscope changes to the DC level of the secondary DC voltage source. This is shown in figure 28. Conversely, during secondary supply mode, when primary voltage DC hits 4.8V, output DC switches back to the same DC level as the primary source.

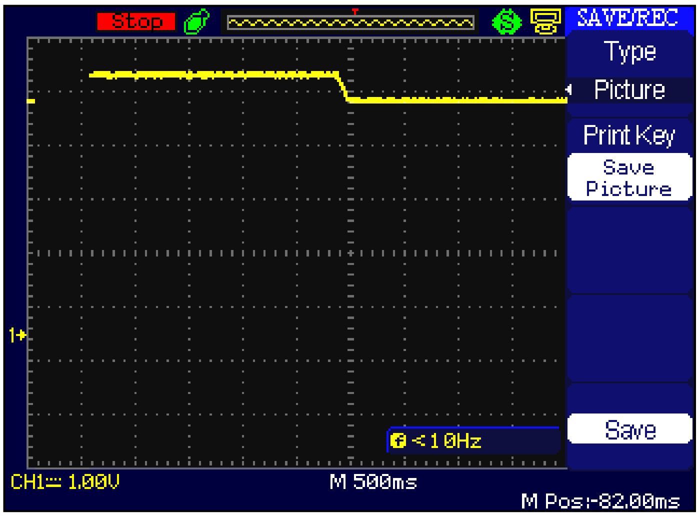


Figure 28: Output voltage changes to secondary DC source.

3.5 Power over Ethernet injection

In the electrical system of Bumblebee AUV version 3, there are 3 electronic sub-systems that are physically separated from the main electronic housing. In order to provide connectivity and data to these external modules on the AUV, Power over Ethernet (PoE) is employed. PoE uses RJ45 Ethernet cable to transfer both data and power to a sub-system. The pin allocation of the RJ45 cable is as shown in figure 29.

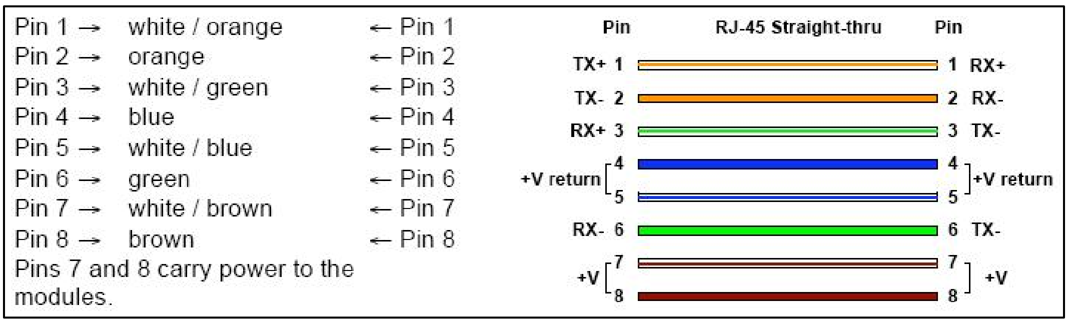


Figure 29: POE pin allocation for RJ45 Ethernet cable. Source: http://www.wisptech.com/images/252-Pin_out_PoE.jpg

There are two main methods of PoE. The first one involves separate wires for power and data as shown in figure 29. The second method utilizes the same wires for power and data. The data signal, which has high frequency of 10-100 MHz, is superimposed on the DC power level. Power and data are conserved without interfering with each other. The PoE method employed in this project is the first method as it is easier to implement than the second. Out of 8 wires, 4 wires are used for DC power (2 positive wires and 2 negative wires); the other 4 wires are used to transfer data.

A PCB prototype was developed to test the PoE circuitry. The aim was to make sure that DC power is injected while data connectivity is still ensured. The PCB design layout design is shown in figure 30.

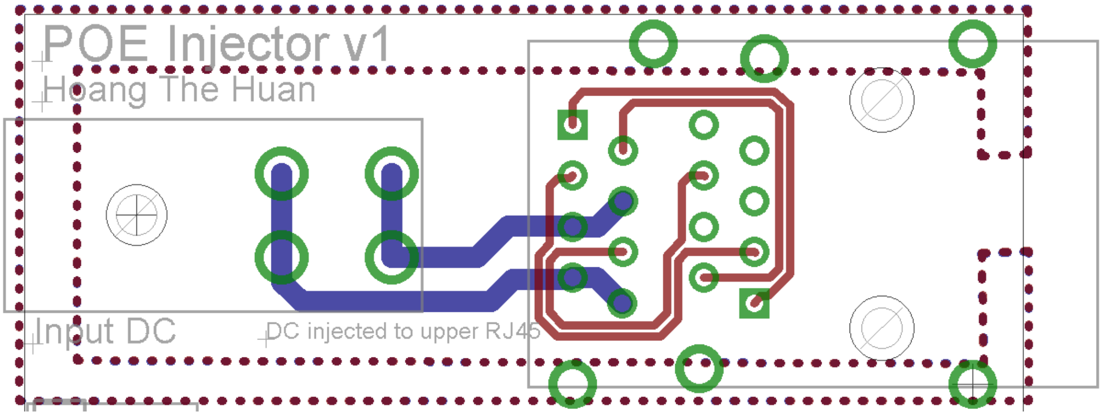


Figure 30: PoE injector PCB board design.

The PoE design was sent for fabrication. The finished PCB was populated with a stacked RJ-45 receptacle and a Molex 4 pin connector as the input of DC power. The assembly was tested with Blueview P450 sonar which is a PoE-powered device to test for functionality. The result is shown in figure 31 and figure 32.

Figure 31: PoE injector PCB prototype. 24V DC power is injected on the left.

Data cables connected to the stacked RJ-45 receptacle on the right in Fig. 31.

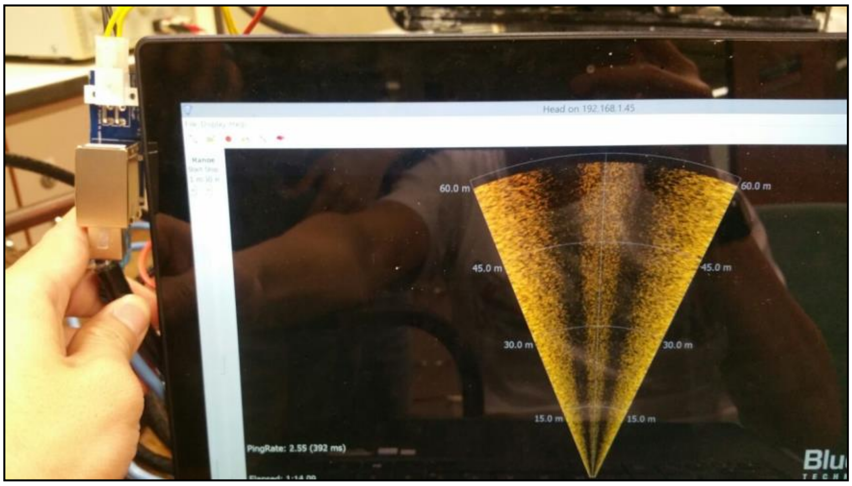


Figure 32: Blueview P450 sonar functions properly over PoE injector prototype.

3.6 USB to UART converter with signal isolation

There is a need for communication between the Single Board Computer (SBC), which is the main processing unit of the AUV, and the other peripherals seated on the backplane. These peripherals include the microcontroller (MCU) on the backplane, thrusters control board, sensors and actuators control board and telemetry board. Apart from the Controller

Area Network (CAN) bus, dedicated serial links between the SBC and each peripheral is also needed in case of failure of Controller Area Network. The protocol for this one to one communication is Universal Asynchronous Receiver/Transmitter (UART) as the microcontrollers on each peripheral has built-in capability for that. SBC, however, does not have UART capability. Therefore, the UART signals need to be converted to USB signal for communication between SBC and the peripherals.

3.6.1 USB to UART converter

To convert UART to USB signals and vice versa, FT232RL is employed. FT232RL translates the UART's single ended unipolar signals into USB's differential signals and vice versa. FT232RL can be powered from the USB's 5V power rail or from the UART peripheral's 5V power rail. In the application of this project, the USB 5V power rail configuration is used. This configuration is illustrated in figure 33.

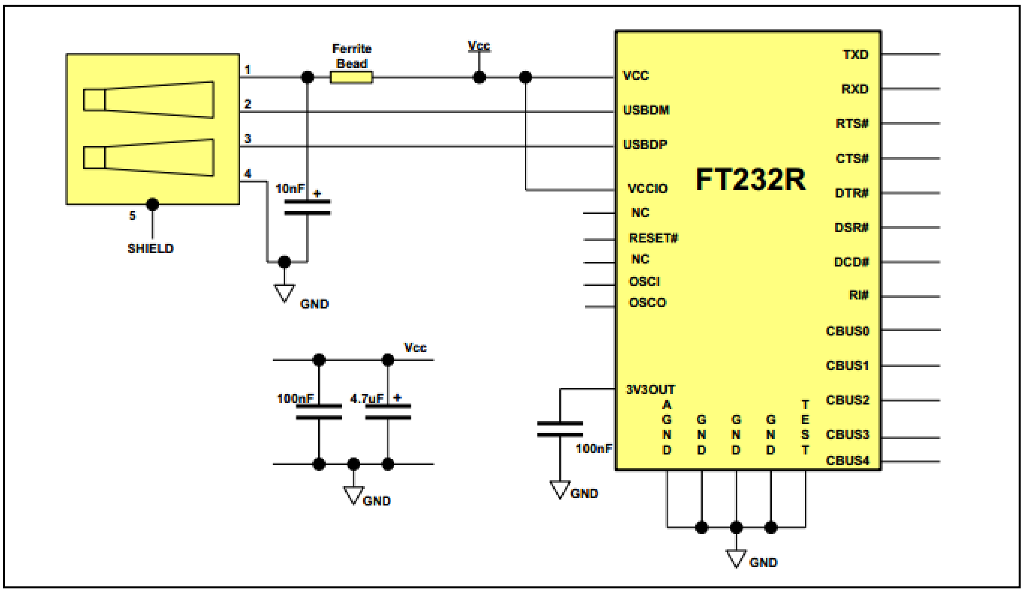


Figure 33: FT232RL's USB bus powered configuration. Source: http://www.ftdichip.com

3.6.2 Signal isolation

Signal isolation is a method of maintaining data connectivity while removing the physical connectivity between a signal driver and a signal receiver. This can be done by a few methods. These methods include optic isolation, inductive isolation (transformer) and capacitive isolation. The advantages of signal isolation include protection of devices against high voltages in close vicinity, protection against transient voltages which might appear on the signal lines, removing electrical noise coupling from system to system and removing ground loop.

ADUM14022 is employed as the signal isolator for the UART–USB interface on the backplane. ADUM1402 is an inductive isolation device with 4 channels of isolation, 2 forward (transmitting) and 2 backward (receiving) channels. Adum1402's channel configuration fits the UART signals' directions. The two forward signals (transmitted) include TXD and CTS while the 2 backward signals (received) include RXD and DTR. ADUM1402 supports up to 90Mbps data rate which is sufficient to accommodate for the maximum data rate of 1Mbps of UART. The schematic of the isolated UART-USB interface is shown in figure 34.

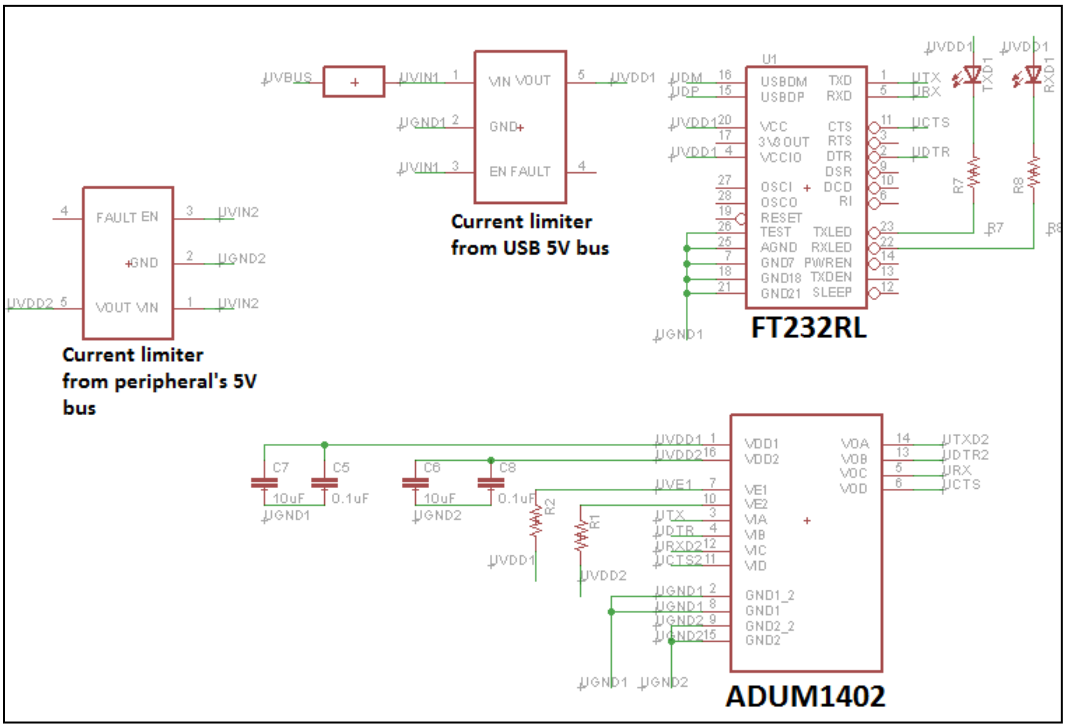


Figure 34: Isolated UART-USB converter with current limiters.

Based on this schematic, a PCB prototype board layout was designed and sent for fabrication. The resulting PCB was populated with the required components for testing. Two isolated UART-USB channels are included in the layout and connected to a stacked USB female socket. This is to simulate the exact layout that would be used on the backplane. Figure 35 shows the PCB layout and figure 36 show the populated PCB.

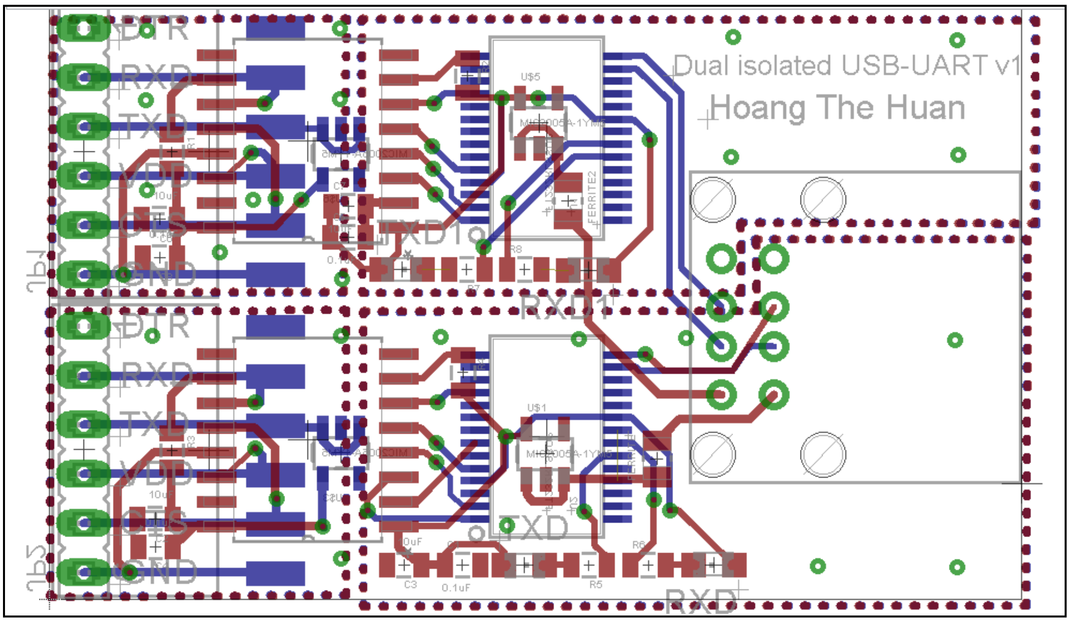


Figure 35: Dual-channel isolated USB-UART converter prototype PCB layout

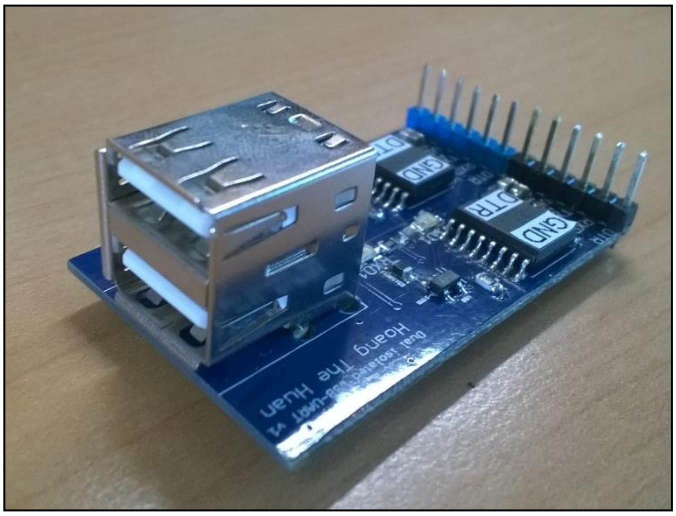


Figure 36: Populated dual-channel isolated USB-UART converter prototype

The prototype was used to programme an Arduino Mini board as a connectivity test. A blinky program was programmed onto the Arduino Mini board. The LED blinked, indicating that connectivity is guaranteed.

3.8 High-side current sensing

The ability to monitor currents on different branches on the electrical backplane allows user to know the status of the sub-systems and the peripherals. This also makes way for over-current protection capability and facilitates debugging.

LTC6103 high-side current sense amplifier is employed. The application schematic for LTC6103 for single-direction current measurement is shown in figure 37.

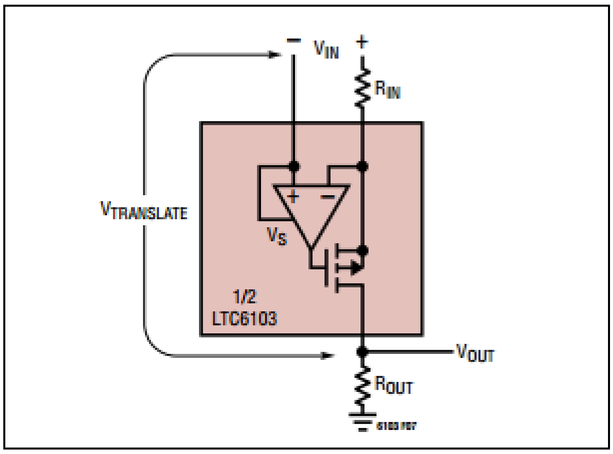


Figure 37: LTC6103 single direction current measurement configuration. Source: http://cds.linear.com/docs/en/datasheet/6103f.pdf

The negative and positive inputs into LTC6103 as seen in figure 37 are connected to two ends of a shunt resistor. The shunt resistor has to have low resistance in order to reduce the power loss across it; it also needs to be able to withstand the heat generated by its own resistance. LVK25R002FER was chosen as the shunt resistor based on the above-mentioned criteria. The resistance of LVK25R002FER is 0.002 Ω and is rated 2 W. Hence:

$$I_{max} = \sqrt{\frac{P_{rating}}{R_{sense}}} = \sqrt{\frac{2}{0.002}} = 31.66 \text{ A}$$

None of the current branch on the backplane is to exceed 31.66 A, therefore LVK25R002FER is suitable for our usage.

$\mathrm{R_{IN}}$ controls the transconductance of the current sense circuit since:

$$\mathrm{I_{out}} = \frac{\mathrm{V_{sense}}}{\mathrm{R_{IN}}}, \text{transconductance gm} = \frac{1}{\mathrm{R_{IN}}}$$

$\mathrm{R_{IN}}$ must be chosen so that a suitable amplification ratio is obtained while limiting the output current. $\mathrm{R_{OUT}}$ converts the output current into a voltage which is measured by the analog to digital converter (ADC). Most ADC has an input voltage tolerance of 5V. Hence $\mathrm{R_{IN}}$ and $\mathrm{R_{OUT}}$ must be selected such that a high amplification ratio is attained while maintaining $\mathrm{V_{OUT}}$below 5V. $\mathrm{V_{sense,max}} = 31.66\ \mathrm{A}\ \times 0.002\ \Omega = 0.06332\ \mathrm{V}$. Choosing $\mathrm{R_{IN}} = 100\Omega$, $\mathrm{I_{out,max}} = \frac{0.06332}{100} = 0.0006332\ \mathrm{A}$. Choosing $\mathrm{R_{OUT}} = 5\ \mathrm{k\Omega}$, $\mathrm{V_{out,max}} = 5000\ \Omega\ \times 0.0006332\ \mathrm{A} = 3.166\ \mathrm{V}$. This is within the 5 V tolerance limit of the ADC.

To measure the $\mathrm{V_{out}}$ across $\mathrm{R_{out}}$, ADS1115 16-bit analog to digital converter is used. ADS1115 translates analog measurement into digital format. The digital readings are accessible via the I^2C bus.

A PCB prototype using LTC6103 and LVK25R002FER was developed to test this current measurement method. The schematic is as shown in figure 38.

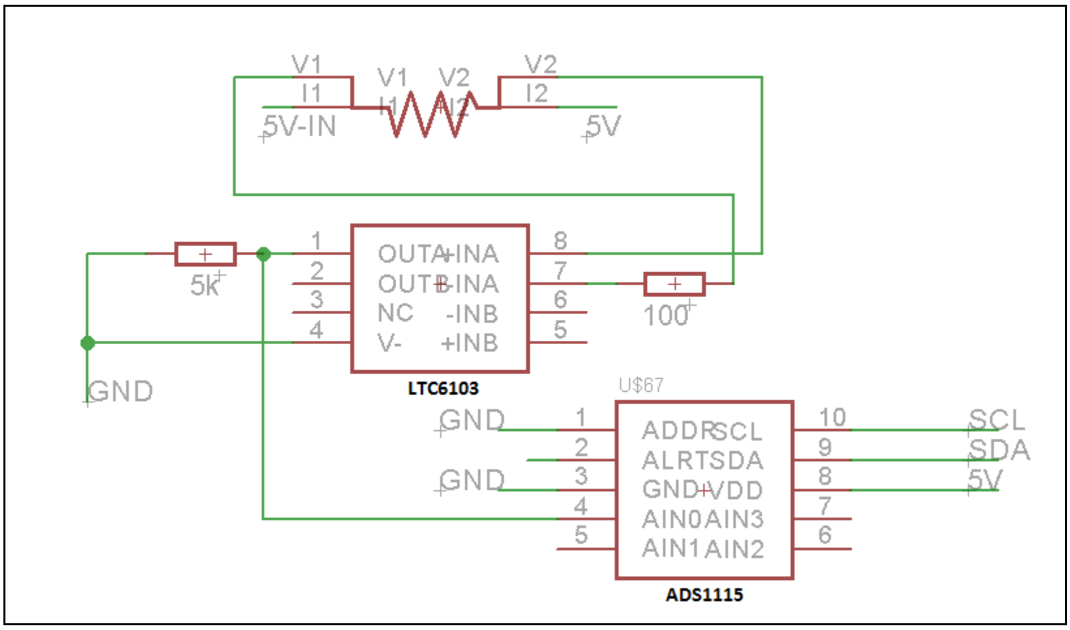


Figure 38: Schematic of current measurement prototype

A PCB prototype was then developed from this schematic and populated with necessary components for current measurement testing. The complete prototype is shown in figure 39.

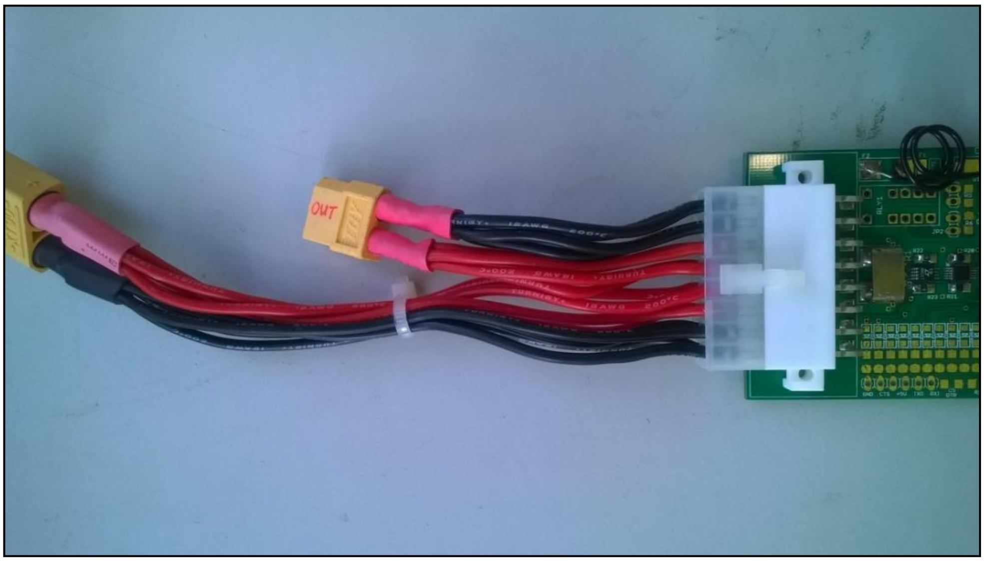


Figure 39: Populated current measurement PCB prototype.

With the PCB prototype ready, a setup was arranged to record the measured current readings and compare with actual current readings from a Fluke 115 multimeter. A power supply is used as the power source. An adjustable array of power resistors is used to simulate a load and to vary the current running over the shunt resistor. An Arduino Pro Mini board is connected to the PCB prototype to read voltage readings from ADS1115 over the I^2C bus. The readings are printed onto a serial port on a laptop. The set-up is shown in figure 40.

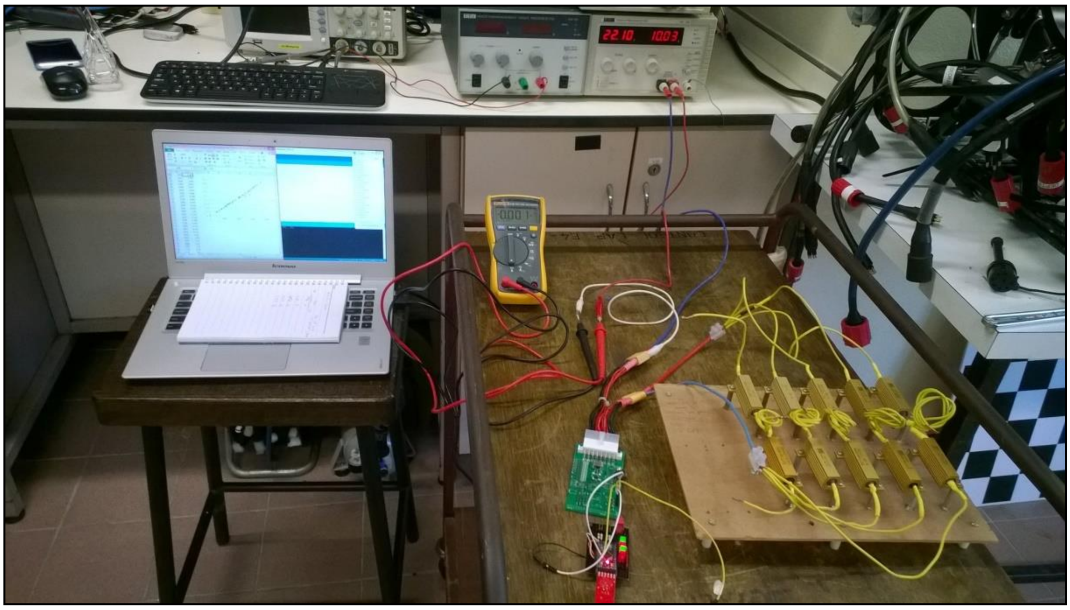


Figure 40: Set-up to test and calibrate current measurement PCB prototype.

The amplified shunt resistor voltage readings and actual current readings from the multimeter are recorded with a cell phone camera and is played back to extract data. A graph of measured voltages against readings of multimeter is then constructed to facilitate calibration of the measurement prototype. Outliers are removed from the graph. Figure 41 shows the graph with outliers and figure 42 shows graph with outliers removed.

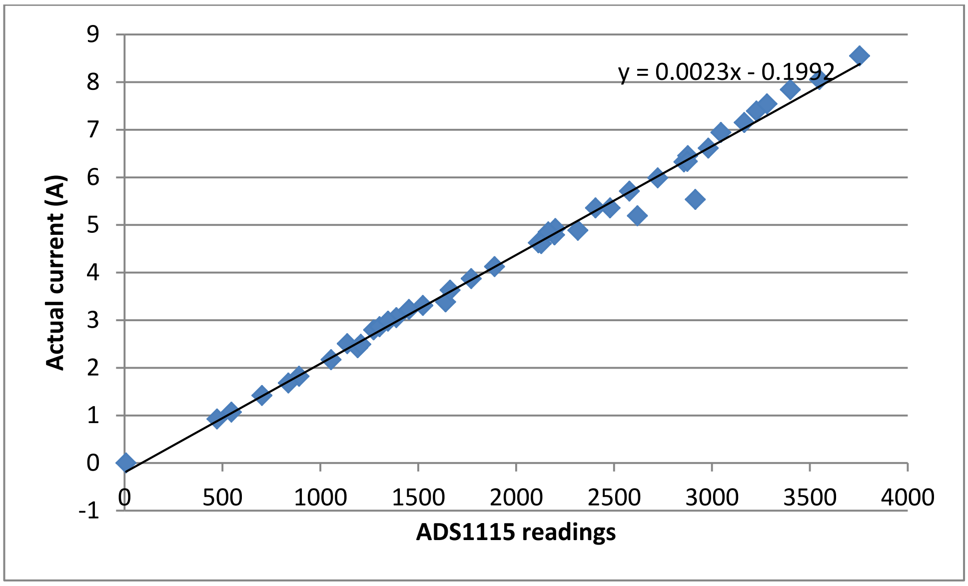


Figure 41: Measured voltage against current readings by Fluke 115 multimeter (with outliers)

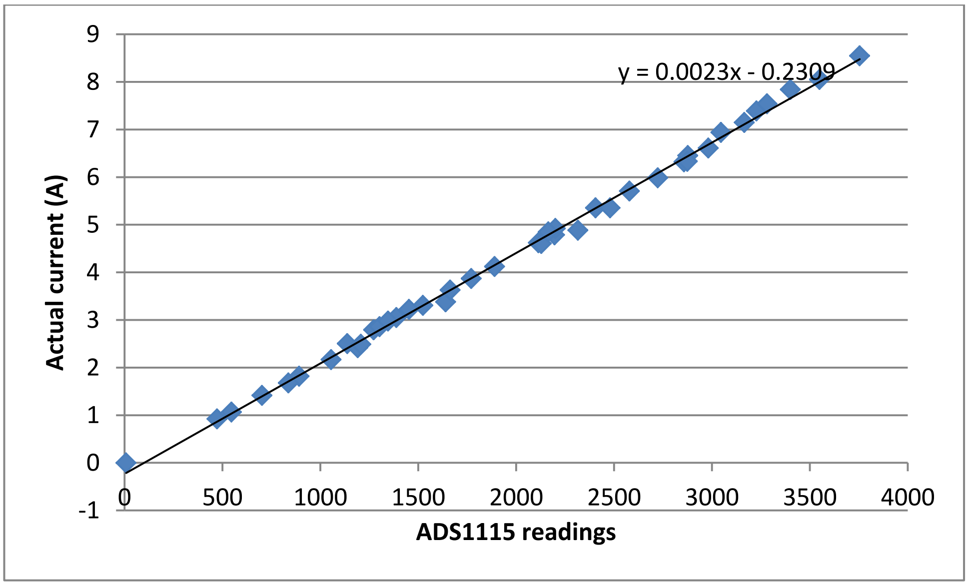


Figure 42: Measured voltage against current readings by Fluke 115 multimeter (without outliers)

The linearity correlation coefficient between the measured voltage and the current readings is 0.998971. Hence, a linear relationship between the measured voltage and the actual current is assumed. The linear equation of the line of best fit is $y = 0.0023x - 0.2309$. Hence, the scalar calibration factor for the PCB prototype is 0.0023 and the offset factor is -0.2309.

3.9 DC voltage measurement

Voltage measurement capability is necessary on the backplane as there is a need to constantly monitor the various voltage rails in order to detect power fault. This facilitates hardware debugging on the AUV. Atmega2560 has built-in 5 V tolerant ADC channels which are capable of 10-bit resolution voltage reading. 10-bit resolution is suitable for DC voltage monitoring as high accuracy is not required.

In order to measure voltage accurately, an accurate voltage reference is needed for the microcontroller. There are two options of voltage reference. The first option is an external voltage reference; the second option is the internal 1.1 V bandgap voltage reference of the microcontroller. Since there is no stable external voltage reference, internal voltage reference was chosen. A simple voltage divider network is used to divide the input voltage down to a level below 1.1V for proper voltage measurement. A breadboard prototype was developed to test this voltage measurement technique. The schematic of this prototype is shown in figure 43.

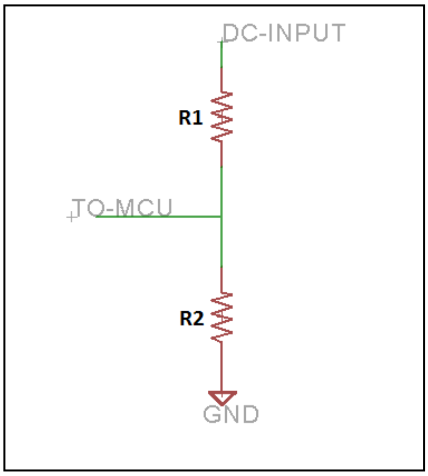


Figure 43: Voltage divider in voltage measurement prototype

Power supply is used to give a variable input DC source; this voltage is divided across R1 and R2. The value of R1 and R2 is chosen such that:

$$V_{\mathrm{input}} \times \frac{\mathrm{R2}}{\mathrm{R1} + \mathrm{R2}} < 1.1\ \mathrm{V}$$

Furthermore, the ADC channel of Atmega2560 has an input impedance of 100 MΩ. Together with the output impedance, this forms a voltage divider. Therefore, R2 should be of sufficiently low resistance in order to have accurate reading value. On the other hand, if R2 is too low, the power loss across it could be too high:

$$P_{\mathrm{loss}} = \frac{{V_{\mathrm{DC-input}}}^2}{\mathrm{R2} + \mathrm{R1}}$$

For a DC input of approximately 12 V, choosing R1 = 200 kΩ and R2 = 7 kΩ, $V_{\mathrm{to-MCU}} = \frac{7}{207} \times 24\ \mathrm{V} = 0.812\ \mathrm{V}$ which is below 1.1 V. Power loss across this resistor network is:

$$P_{\mathrm{loss}} = \frac{{V_{\mathrm{DC-input}}}^2}{\mathrm{R2} + \mathrm{R1}} = \frac{24^2}{207000} = 0.0028\ \mathrm{Watts}$$

This power loss is negligible.

The ADC readings are printed onto a serial port on a laptop. These readings are recorded down together with the actual DC voltage measured by Fluke 115 multimeter. Figure 44 shows the set-up to measure DC voltage.

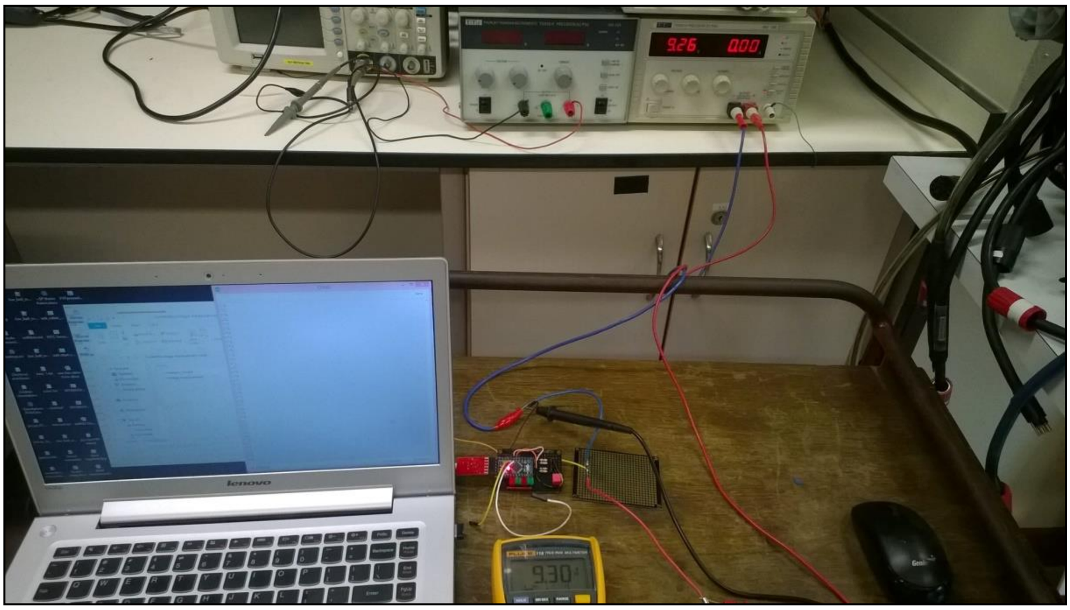


Figure 44: Set-up to calibrate ADC voltage measurement

A graph of voltage readings against actual readings is plotted to determine the calibration factors. The graph is shown in figure 45.

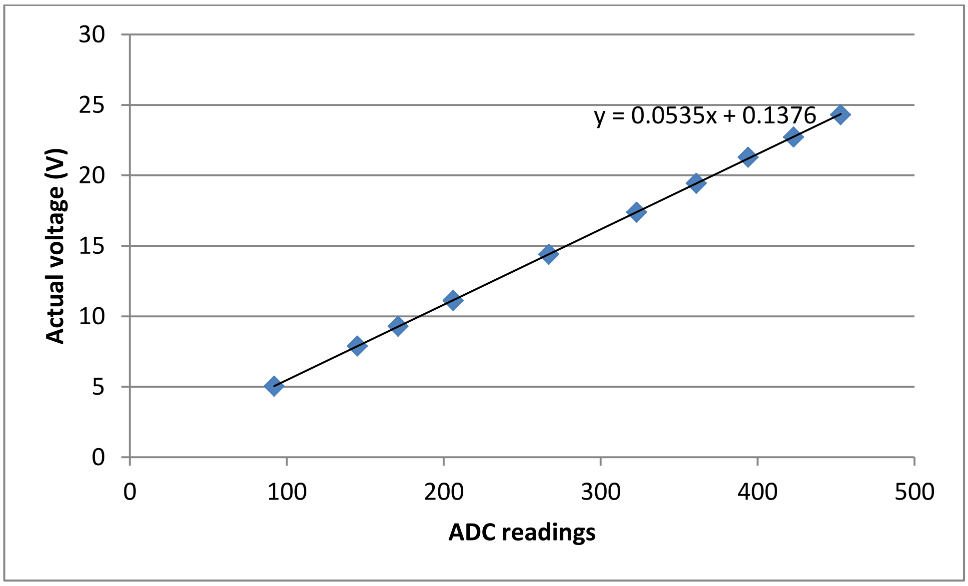


Figure 45: Measured voltage against actual voltage from power supply.

The linearity correlation coefficient of the data points was 0.999985. This indicates a linear relationship between the measured voltages and the actual voltages. The equation of the line of best fit of the graph is $y = 0.0535x +$

0.1376. Hence, the scalar calibration factor is 0.0535 and the offset is 0.1376.

3.10 Controller Area Network circuitry

Controller Area Network (CAN) bus is explored in order to facilitate a communication network between the microcontrollers on the AUV system. CAN is a differential signalling standard, capable of 1Mbps data transfer rate. There is no master or slave on the CAN bus; messages are broadcasted onto the bus and all nodes can receive the messages. The CAN communication protocol is a carrier-sense, multiple-access protocol with collision detection and arbitration on message priority (CSMA/CD+AMP). CSMA means that each node has to wait for a prescribed period of bus inactivity before attempting to broadcast the message. Collisions are resolved by bit-wise arbitration based on pre-programmed priority bits in the identifier fields of each message.

At the physical layer, the CAN bus has two signal lines namely CAN high (CANH) and CAN low (CANL). The bus status consists of dominant and recessive states. In the recessive state, CANH and CANL has the same voltage level (usually 2.5 V); in the dominant state, CANH swings to high voltage level (3.5 V) whereas CANL swings to low voltage level (1.5 V). Being a differential signaling standard, CAN bus is therefore more resistant towards electromagnetic interference (EMI). Furthermore, it can support up to 30 plug-and-play nodes. Figure 46 shows an illustration of the bus configuration.

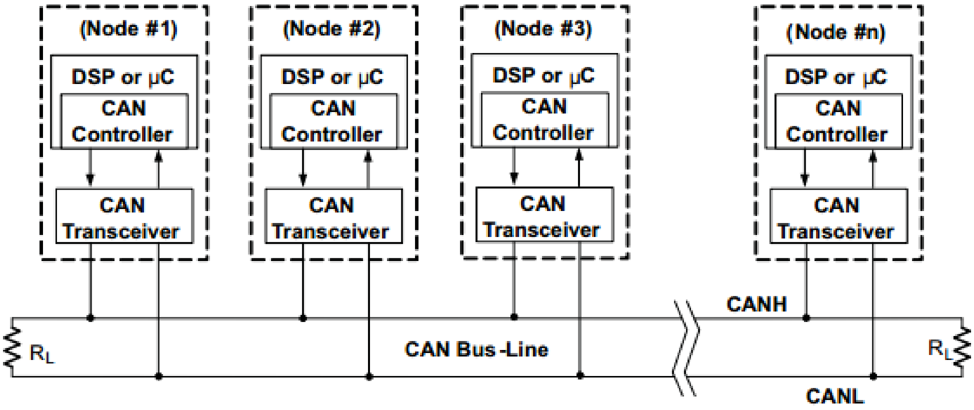


Figure 46: Details of a CAN bus. Source: Texas Instrument)

A schematic of a CAN circuitry was constructed with MCP2515 CAN controller IC and ISO1050 isolated CAN transceiver. MCP2515 belongs to the Data-link layer of the network; it controls the logic, bus arbitration and packaging of the bits. ISO1050 belongs to the physical layer; it controls the electrical signaling. Figure 47 shows the schematic of the CAN circuit and figure 48 shows the PCB layout.

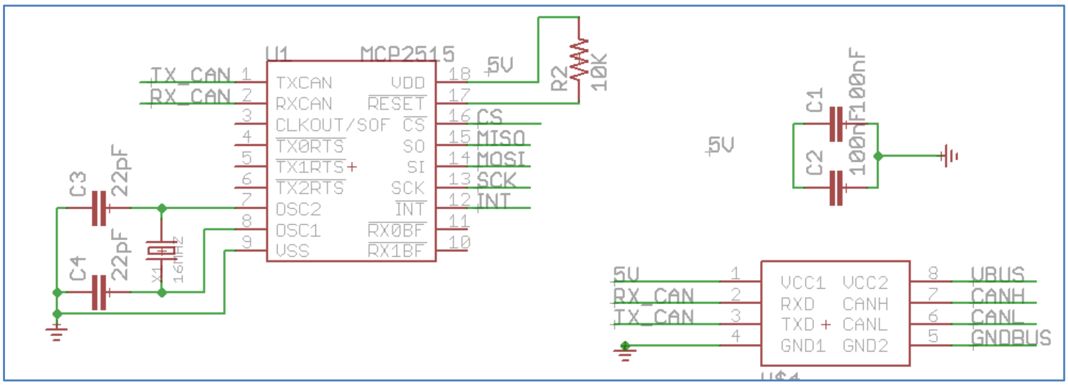


Figure 47: Schematic of the CAN circuitry

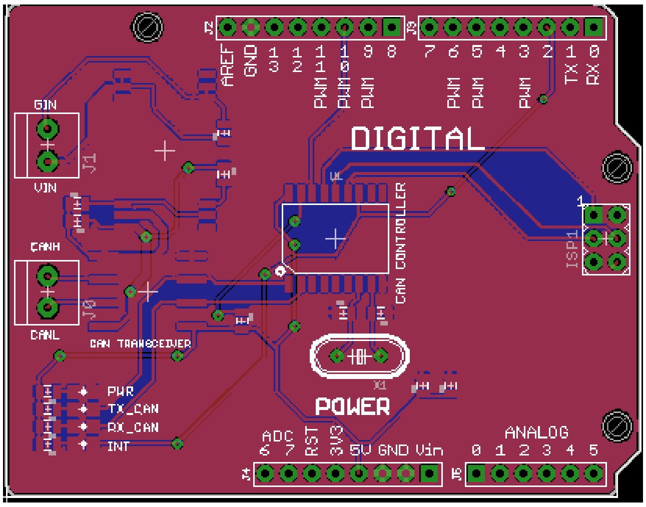


Figure 48: CAN prototype PCB layout

Two prototypes were built in order to test out the communication between two microcontroller nodes.

Chapter 4 Backplane Design Considerations

4.1 Electrical backplane and other electronics

When designing the electrical backplane, the interaction of the backplane and other electronics on the AUV system has be considered. Overall, the main purpose of the backplane includes:

- Provides power to peripherals
- Provide data connectivity to peripherals
- Current and voltage monitoring
- Power path controls

Figure 49 explains the basic power and data connections to the major electronic components on the AUV.

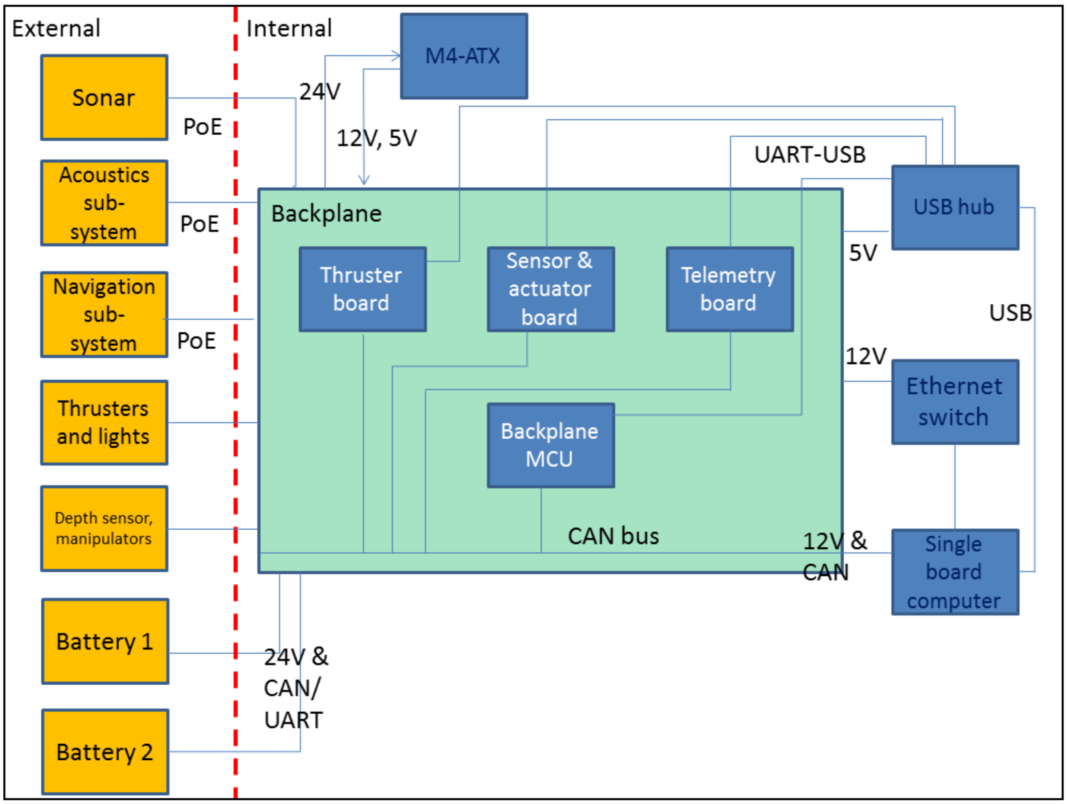


Figure 49: Power and data links between backplane and other major electronics

As can be seen from the above figure, the electrical backplane acts as a power distribution point. It takes power from two external batteries and load-balances them. It powers the M4-ATX with the load-balanced 24 V output and receives back regulated 12 V and 5 V. With 24 V, 12 V and 5 V, the backplane provides appropriate voltage to each of the electronics outside of the housing as well as the inside the housing.

The backplane Atmega2560 MCU is the main processing unit of the backplane. It samples the voltage and current readings of the monitoring circuitries, detects over current condition and cut power to the daughter board that is causing the overcurrent. The MCU sends the telemetry back to the SBC.

The backplane also acts as data distribution point. It houses 6 UART-USB converter circuitries to facilitate one to one communication between the SBC and other peripherals with microcontroller. The backplane also houses a CAN bus. It has a dedicated microcontroller Atmega328 to convert CAN messages to USB, enabling the SBC to join the CAN bus.

4.2 Backplane electrical architecture

Figure 50 explains the entire electrical architecture of the backplane.

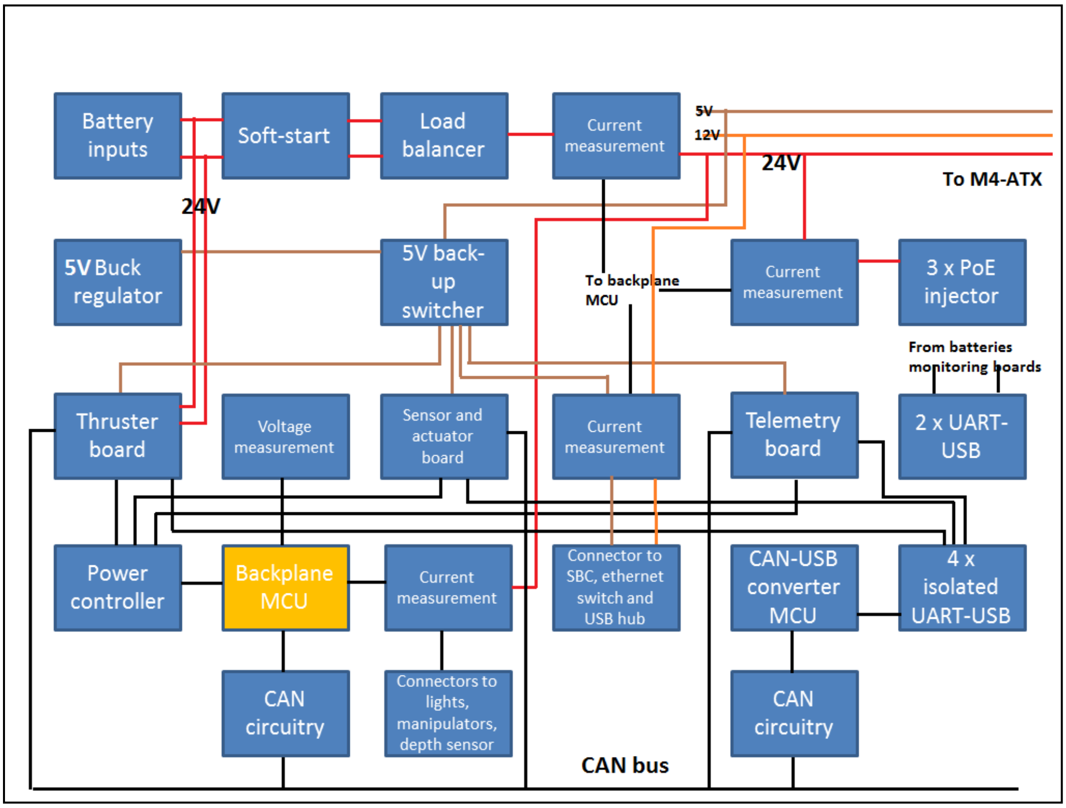


Figure 50: Inter-connection of the functional circuitries on backplane

At the heart of the backplane, the backplane MCU receives current and voltage measurements and controls the power input to thruster, sensor & actuator and thruster board. The power circuitries provide load balancing capability, soft-start upon input plug-in and 5V back-up power path switcher. CAN circuitries include CAN controller and transceiver, enabling backplane MCU to join the CAN bus. A dedicated MCU acts as a CAN-UART converter. This MCU translates CAN messages into UART messages. UART signal is converted to USB signal and sent to the SBC, enabling SBC to join the CAN bus. There are three PoE injector modules on the backplane with current monitoring capability.

4.3 Spatial considerations

The dimensions of the backplane are constrained by the diameter of the cylindrical electronic housing. The housing has a diameter of 17cm; the backplane therefore has a maximum allowable width of 15.5 cm. The height of the daughter boards and the mating heights of the daughter board connectors are also constrained within 15.5 cm. An illustration of the constraints is shown in figure 51.

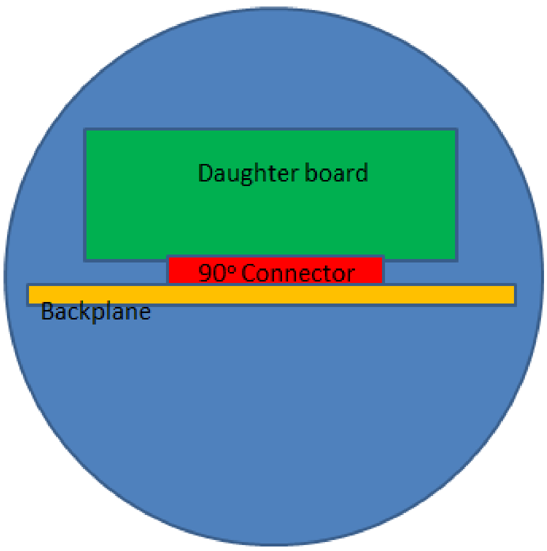


Figure 51: Backplane assembly inside cylindrical electronic housing

With the high number of functional circuitries on board, proper planning of physical placement of each functional circuitry is needed. This process is done with considerations of the internal interaction and connection between the functional blocks as well as between the internal functional blocks and external electronics.

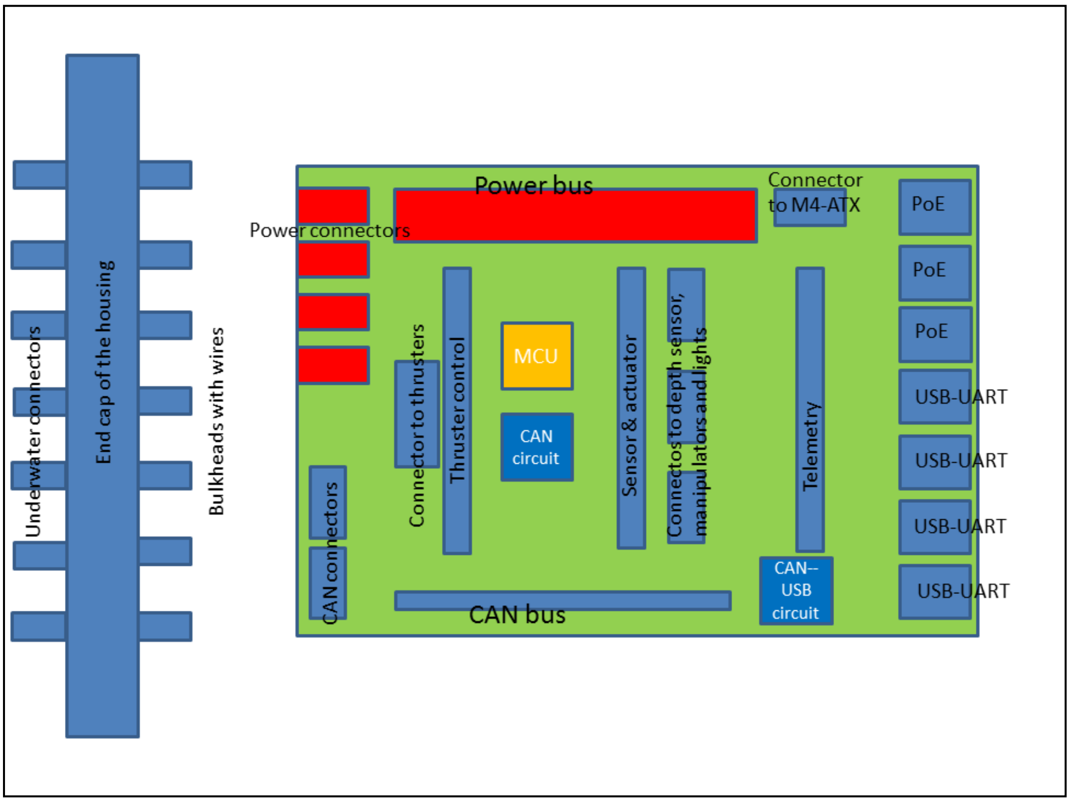


Figure 52: Lay-out of major parts on the backplane

The power bus and CAN bus are separated as far apart as possible to minimize crisscrossing between data lines and power lines. Power lines are a great source of noise which needs to be separated from data lines. The PoE and USB connectors are located at the edge of the backplane for ease of access. The daughter boards are orientated in such a way so as to access the power lines and the data lines most easily. The connectors to components outside of the electronics housing such as thruster, lights, manipulators and depth sensor are located parallel with the daughter board. They are within close vicinity of the daughter boards so as to minimize the length of the signal and power traces.

4.4 Conducting high current

There are many high current-carrying conductors on the backplane. Depending on the amount of current to be accommodated and the thickness of the copper traces, the widths of the traces are then determined.

Saturn PCB toolkit V6.81 was used to calculate the trace width. A screenshot of the software is shown in figure 53. In the case of the backplane, the copper thickness of the PCB is 1.5 oz (specification given by the fabrication house).

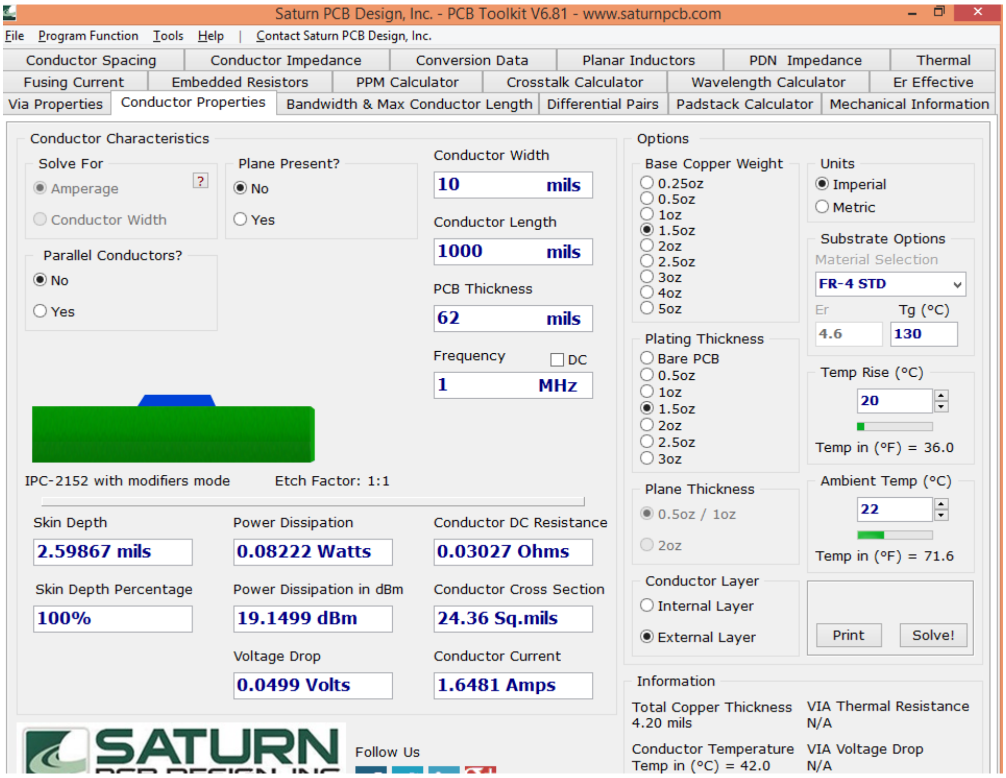


Figure 53: Calculating trace width with Saturn PCB toolkit V6.81

4.5 Noise and grounding considerations

Grounding configurations on PCB can directly affect the noise level of signals and hence the performance of the backplane. The first grounding guideline is a star configuration for ground return path as can be seen in figure 54.

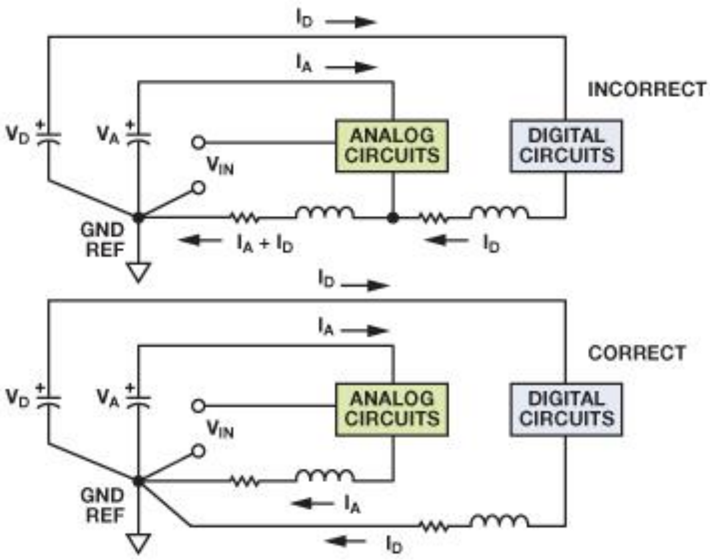


Figure 54: Star ground configuration. Source: http://www.analog.com/library/analogDialogue/archives/46-06/staying_well_grounded.html

In the first configuration in the above figure, the ground return paths of the analog circuits and that of the digital circuits overlap. This is also called a ground loop. A ground loop will create unwanted signals picked-up by the analog circuits in this case. This is because the return current of digital circuits I_D create extra voltage drop due to the resistance and inductance on the overlapped return path.

However, in practical PCB layout, it is difficult to create a star-ground configuration. A more common technique used is a ground plane. A ground plane, due to its large size, provides low impedance return current path. In so doing, ground plane minimizes signal pickups/interference between systems. In the backplane, a ground plane configuration is used. The ground plane provides return current path for the digital circuits. There exist a small number of analogue circuits on board, which is contained within ICs such as ADS1115. These ICs already have its own analog ground within themselves. The ground plane on backplane PCB only connects to the power ground at only one point to avoid forming a ground loop with the power ground. This configuration is illustrated in figure 55.

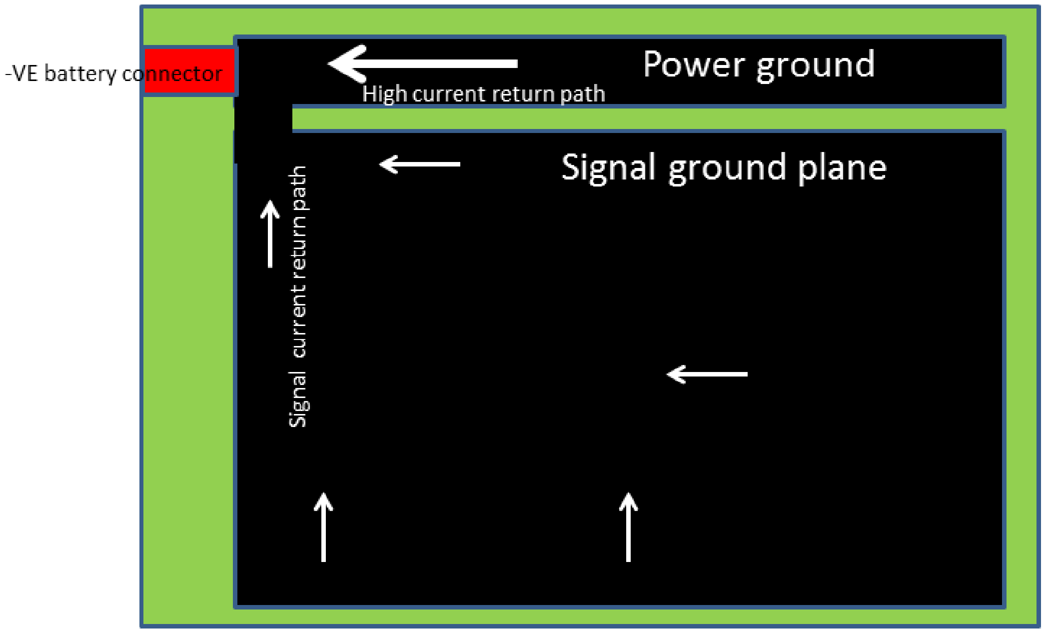


Figure 55: Ground plane configuration on backplane PCB

4.6 High speed signal routing considerations

When designing high speed signal traces on a PCB, guidelines must be taken to ensure that the traces will not introduce impedance mismatch, which has adverse effect on the high speed signals. On the backplane PCB, there are two kinds of high speed signals namely Ethernet and CAN signals.

The PCB layout guidelines for Ethernet traces are:

- For each Tx+/Tx- and Rx+/Tx+ pair, their separation should be under 0.25mm
- Differential impedance should be 100Ω, each trace should have impedance less than 50Ω.
- Keep the Tx+/Tx- and Rx+/Rx- trace lengths as equal as possible.

Using Saturn PCB toolkit V6.81, the trace width and trace separation were determined in order to fulfill the criteria. The resulting Ethernet trace layout in PoE injector module is as seen in figure 56.

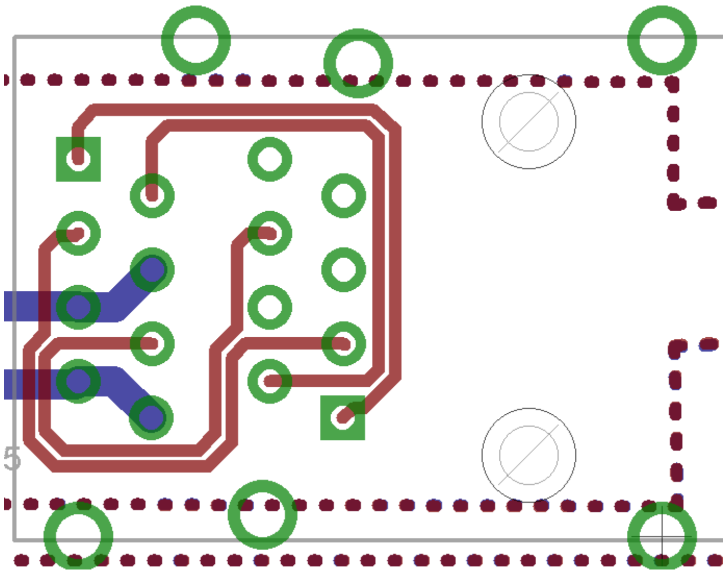


Figure 56: Ethernet traces in PoE injector prototype.

The PCB layout guidelines for CAN traces are:

- Differential impedance between the CANH and CANL lines must be approximately 120Ω
- Ensure that a ground plane is present underneath the differential pair

Again, Saturn PCB toolkit V6.81 is employed to determine the trace width and separation of the differential CANH CANL pair.

4.7 Backplane PCB layout

With all the considerations mentioned in this chapter, backplane PCB was designed using EAGLE V6.6.0 PCB design software. A two-layered PCB was design with dimension 15.5cm x 22.0 cm. Figure 57 show the final layout

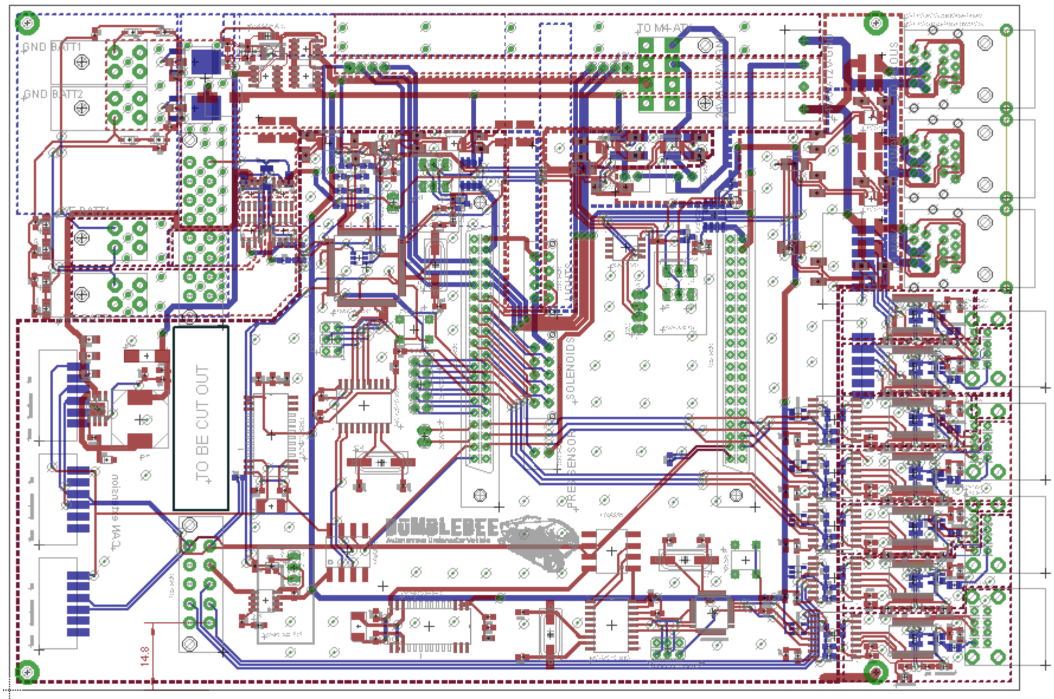


Figure 57: Backplane PCB layout in EAGLE V6.6.0

Chapter 5 Backplane PCB Manufacturing and Assembly

5.1 Manufacturing

The PCB board layout has to be run through a design check with parameters given from the fabrication house. Typical parameters include minimum spacing between pads, spacing between pads and vias, minimum trace width, etc. Failing the design check will results in the PCB manufacturing infeasible.

From the board layout, Gerber files are generated and sent to the fabrication house for manufacturing. Gerber files describe the features present on different layers of a PCB such as pads, vias, silk-screens, drillings and milling and PCB board dimensions. Figure 58 shows the top copper layer of the backplane PCB.

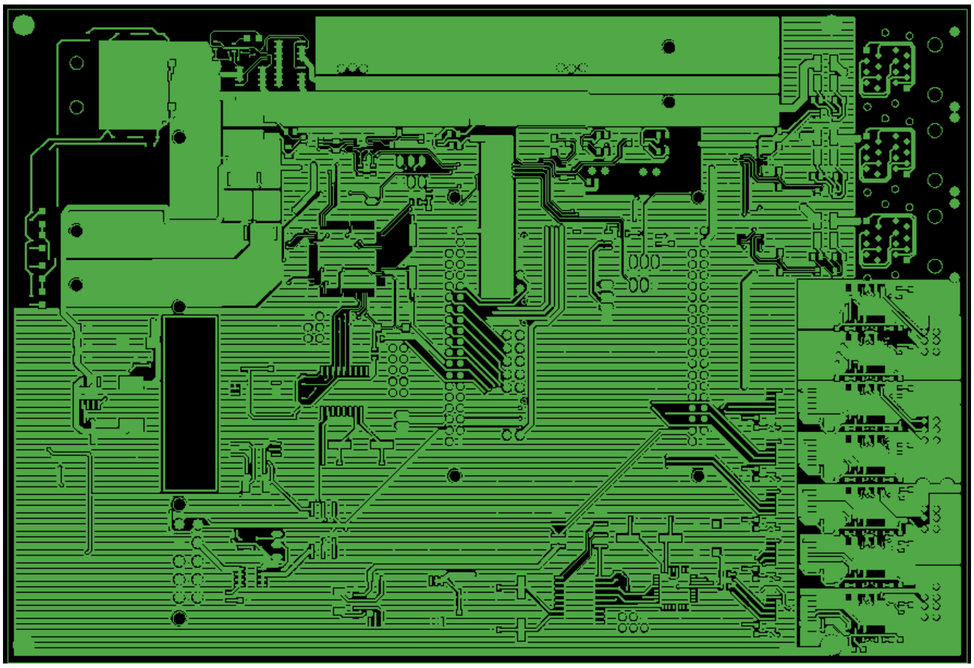


Figure 58: Backplane top copper layer Gerber file.

Figure 59 shows the front side of the bare backplane PCB after manufactured by the fabrication house.

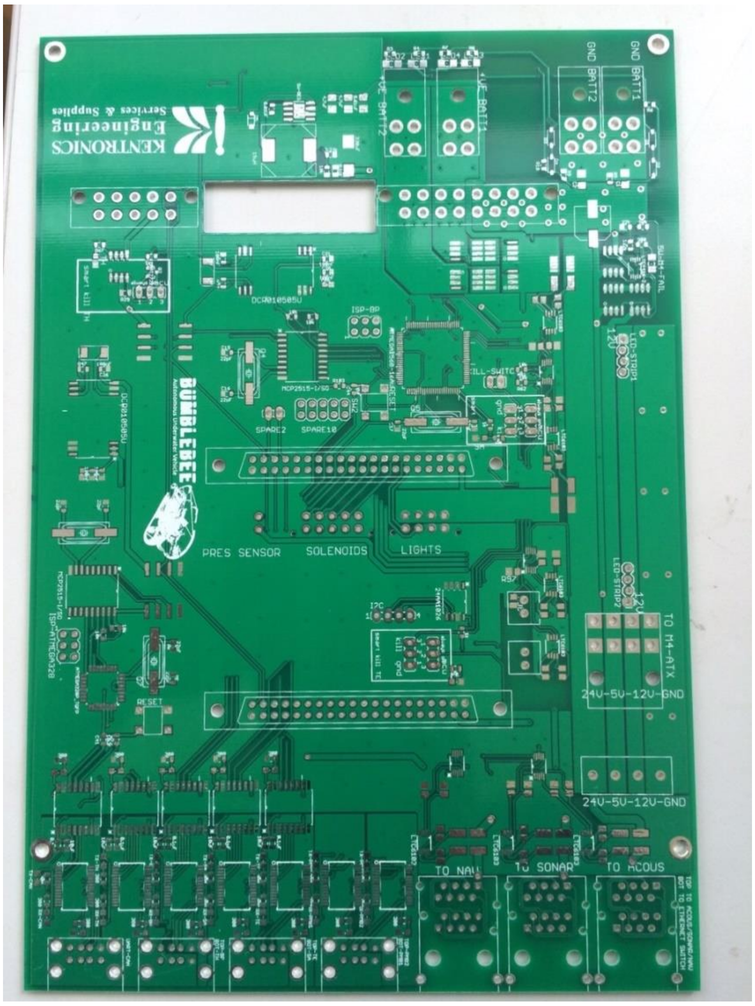


Figure 59: Bare backplane PCB

5.2 Populating and testing the backplane

After the backplane PCB is delivered from the fabrication house, ICs and passive components are populated onto the PCB. The backplane is populated in order of functional blocks. Power functional blocks are populated first followed by the data functional blocks. After each functional block is populated, functionality test is carried out. Only if block can carry out the functionality as its prototype is capable of, the next block is populated.

Figure 60 shows the set-up to test the soft-start block of the backplane

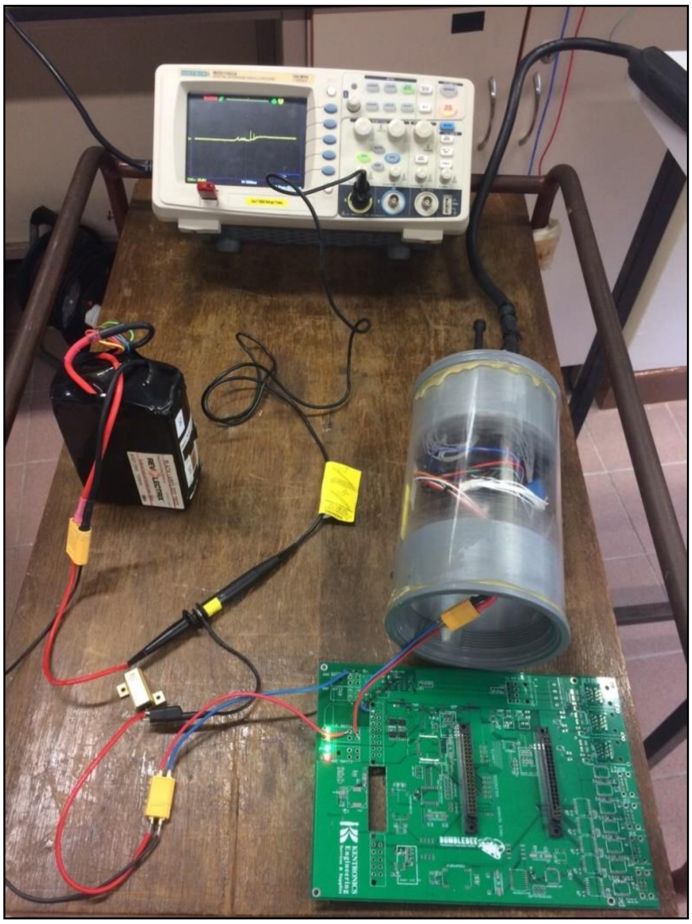


Figure 60: Testing the soft-start block of backplane PCB

Figure 61 show the set-up to test the 5V back-up switcher block on the backplane PCB. The red LED is on, indicating that the primary voltage has failed and the back-up 5V rail is being used.

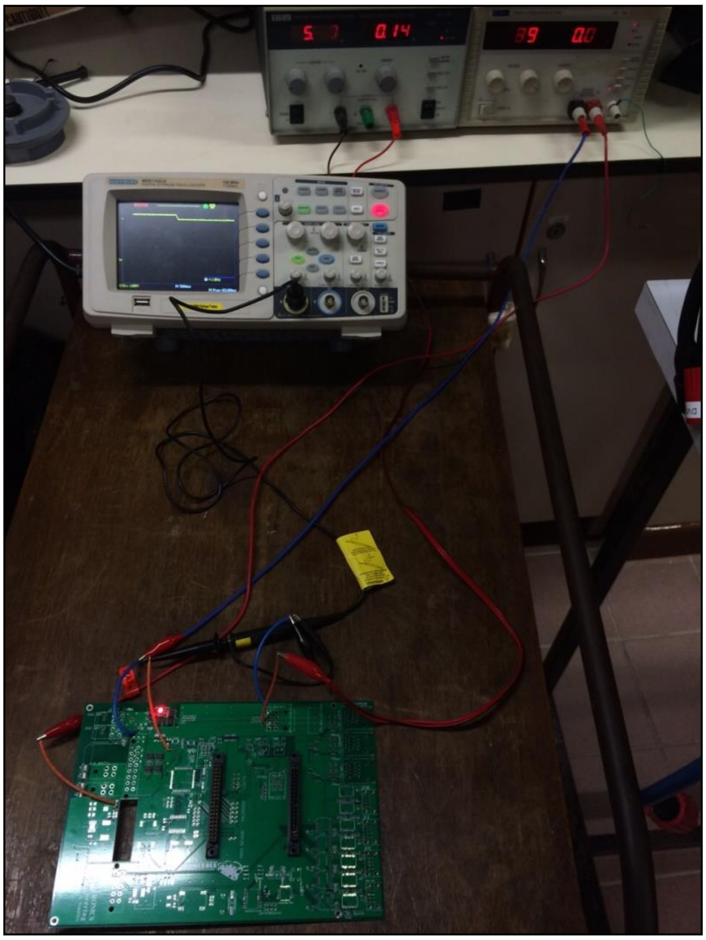


Figure 61: Testing 5V back-up block on backplane PCB

Chapter 6 Conclusion

In general, an electrical backplane system provides reliability to a system. It minimizes an electrical system's susceptibility to mechanical failure of cabling. A backplane also facilitates hardware debugging as it can incorporate various monitoring circuitries. It is therefore easier to identify the fault and isolate a sub-system during debugging. It is also mechanically easier to disassemble and reassemble electronic sub-systems in a backplane system.

The downside of having an electrical backplane system is the development overhead required. Much more resource is needed to design and test a backplane before it can be used reliably. More often than not, a few iterations of a system is necessary. Furthermore, when there are any changes to be made on the system, a new PCB has to be design. This process is time-consuming and can delay overall development or operation of a system.

There are various potential improvements that can be done on the current electrical backplane. Over-voltage protection is necessary in order to protect the backplane system. DC noise removal circuitries can be incorporated onto the backplane to remove noise from the power supply (M4-ATX). Finally, the current 2-layered backplane PCB design can be modified to a 4-layered PCB design. Having more layers facilitates routing and allows for dedicated power and ground plane, which is beneficial to noise reduction.

References

Analog Devices. (2012). *Printed Circuit Board Issues Partitioning.* Retrieved from Analog: http://www.analog.com/library/analogdialogue/archives/43-09/EDch%2012%20pc%20issues.pdf

Aono, K. (2011). *PCB Design with EAGLE.* Michigan: Michigan State University.

Butler, S. (2012). *VPT Power.* Retrieved from Controlling Inrush Current in DC-DC Power Converters: http://www.vptpower.com/wp-content/uploads/downloads/2012/01/info_inrushCurrent.pdf

Carter, B. (2008). *Circuit Board Layout Techniques.* Texas: Texas Instruments.

Comiskey, D. (2008). *Practical Guidelines to Designing an EMI Compliant PoE - Powered Device with Non-Isolated DC/DC.* Texas: Texas Instruments.

E2V. (2009). *Design Considerations for Mixed-Signal - How to Design a PCB Layout.* e2v.

Schmitz, T. (2011). *Grounding Techniques.* Intersil.

Texas Instruments. (1999). *PCB Design Guidelines for Reduced EMI.* Texas: Texas Instruments.

Texas Instruments. (2014). *USB 2.0 Board Design and Layout Guidelines.* Texas: Texas Instruments.

Williams, T. (2007). *The ground plane: Lord of the Board.* Elmac Services.

Zhu, W. (2013). *Isolated CAN Solution by Using ISO1050 and SN6501.* Texas: Texas Instruments.

Zumbahlen, H. (2012). *Staying Well Grounded.* Analog Dialogue.

Appendix A – Arduino code

Arduino code for current measurement calibration

```
#include <Wire.h>

boolean toggle=false;
uint16_t temp0=0, temp1=0;
float voltage0, voltage1;
float current;
float temp=0;

int writeCount=0;
unsigned int tick      =      0;
unsigned int second    =      0;

byte highbyteAIN0, lowbyteAIN0, highbyteAIN1, lowbyteAIN1;

float voltage_array[5]={0.0,0.0,0.0,0.0,0.0};
int current_index=0;
int float_index=0;
float temp_holder;
float voltage_filtered=0.0;

void begin_ads1115()
{
  Wire.begin();                // join i2c bus (address optional for master)
  Wire.beginTransmission(72);   // transmit to device #72 (0b1001000)
  Wire.write(1);                // point register set to config register
  Wire.write(194);                 // configure ADS chip, this if the MSByte of the
16bit config register
  Wire.write(227);              // LSByte of the 16bit config register
  Wire.endTransmission();       // stop transmitting
  Wire.beginTransmission(72);   // transmit to device #72 (0b1001000)
  Wire.write(0);                // point register set to config register
  Wire.endTransmission();       // stop transmitting

}

void read_AIN0()
{
  Wire.requestFrom(72, 2);        // request 6 bytes from slave device #72
(0b1001000)
  while(Wire.available())     // slave may send less than requested
  {
    highbyteAIN0 = Wire.read();
    lowbyteAIN0 = Wire.read();
```

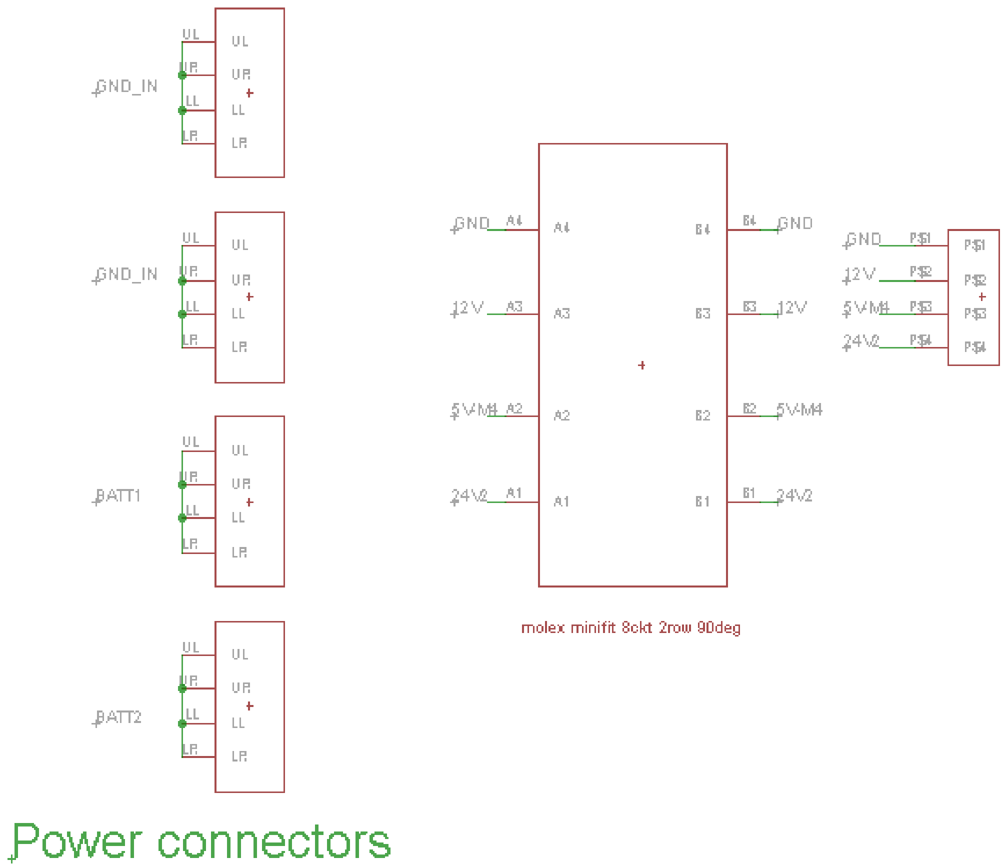
```
  }

  Wire.beginTransmission(72);  // transmit to device #72 (0b1001000)
  Wire.write(1);              // point register set to config register
  Wire.write(82);                // configure ADS chip, this is the MSByte of the 
16bit config register
  Wire.write(227);            // LSByte of the 16bit config register
  Wire.endTransmission();      // stop transmitting
  Wire.beginTransmission(72);  // transmit to device #72 (0b1001000)
  Wire.write(0);              // point register set to config register
  Wire.endTransmission();      // stop transmitting
}

void read_AIN1()
{
  Wire.requestFrom(72, 2);       // request 6 bytes from slave device #72 
(0b1001000)
  while(Wire.available())     // slave may send less than requested
  {
    highbyteAIN1 = Wire.read();
    lowbyteAIN1 = Wire.read();
  }

  Wire.beginTransmission(72);  // transmit to device #72 (0b1001000)
  Wire.write(1);              // point register set to config register
  Wire.write(66);                // configure ADS chip, this is the MSByte of the 
16bit config register
  Wire.write(227);            // LSByte of the 16bit config register
  Wire.endTransmission();      // stop transmitting
  Wire.beginTransmission(72);  // transmit to device #72 (0b1001000)
  Wire.write(0);              // point register set to config register
  Wire.endTransmission();      // stop transmitting

}

void calculate_current()
{
    read_AIN0();
    delay(5);
    read_AIN1();
    /* calculate voltage at nodes of the shunt resistor and hence current 
through */
   if((highbyteAIN0>>7)&0x1==1)
     temp0=0;
     else
     {
        temp0 = highbyteAIN0 * 256;
        temp0 = temp0 + lowbyteAIN0;
```

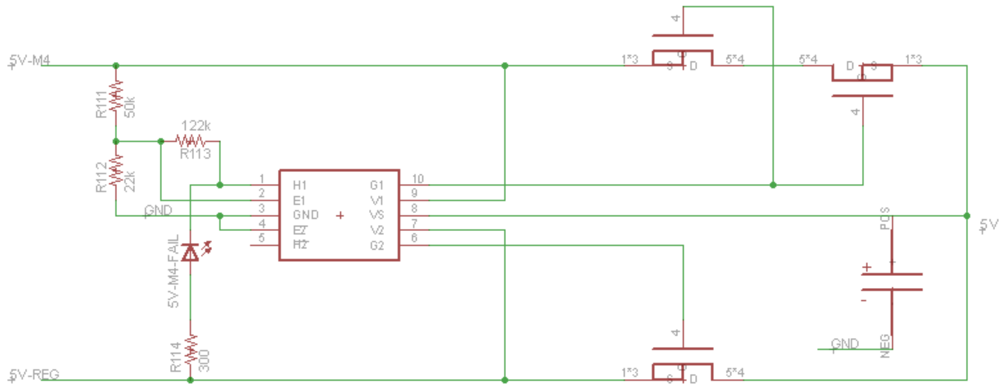
```
    }
   voltage0=temp0;

    if((highbyteAIN1>>7)&0x1==1)
     temp1=0;
     else
     {
         temp1 = highbyteAIN1 * 256;
         temp1 = temp1 + lowbyteAIN1;
     }
   voltage1=temp1;
   voltage_array[current_index]=voltage1;
   current_index++;
   if(current_index==5)
   {
     median_voltage_filter();
     current_index=0;
   }
}

void median_voltage_filter()
{
  for(int i=0; i<5; i++)
    for(int j=i+1; j<5; j++)
      if(voltage_array[i]>voltage_array[j])
      {
        temp_holder=voltage_array[i];
        voltage_array[i]=voltage_array[j];
        voltage_array[j]=temp_holder;
      }

  voltage_filtered=voltage_array[2];
}

void setup(){

    Serial.begin(115200);
    Serial.println("measure current");
    begin_ads1115();
    calculate_current();
}

void loop(){

    calculate_current();
    Serial.println(voltage_filtered,7);
    delay(100);
}
```

Arduino code for DC voltage measurement calibration

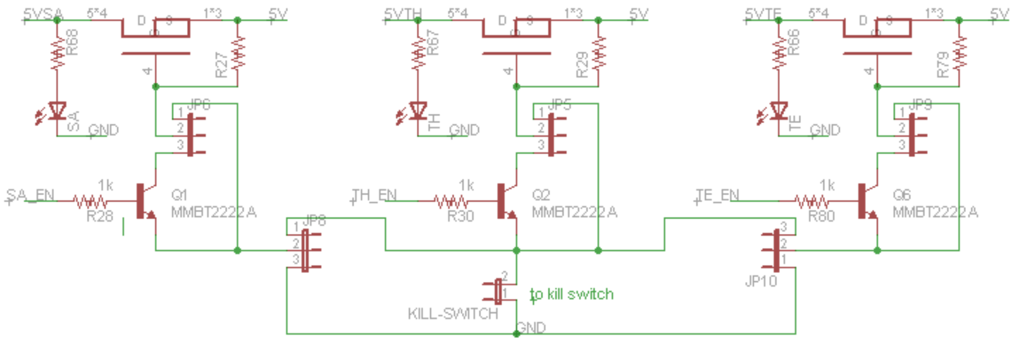
```
int voltage_digital=0;

void setup()
{
    analogReference(INTERNAL);
    Serial.begin(115200);
    Serial.println("voltage measurement");
}

void loop()
{
    voltage_digital = analogRead(2);
    Serial.println(voltage_digital);
    delay(500);
}
```

Appendix B – Backplane EAGLE schematic

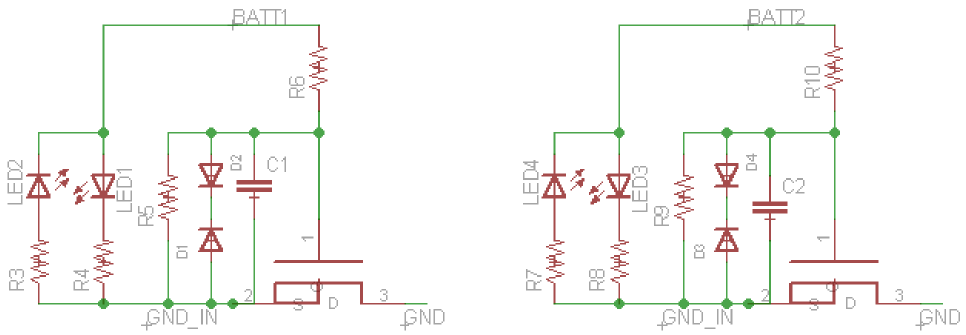




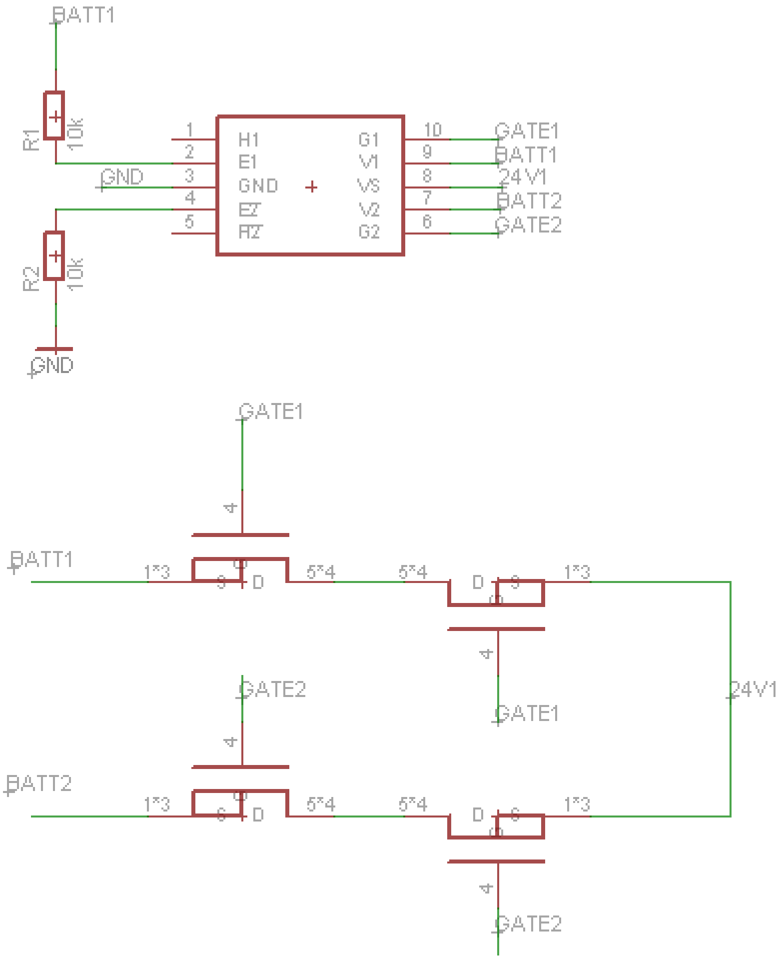
5V backup switcher



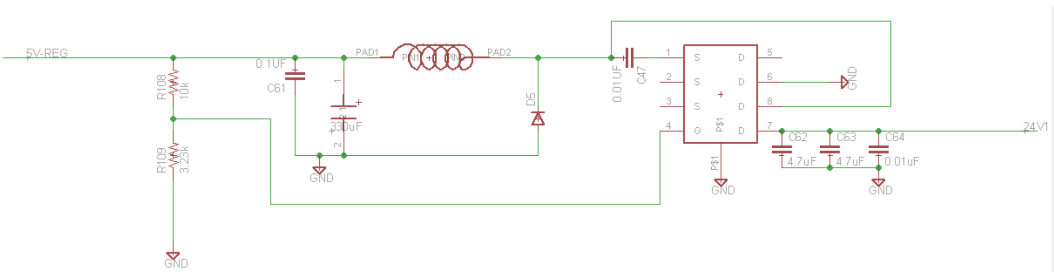
Daughter boards power control



Soft-start

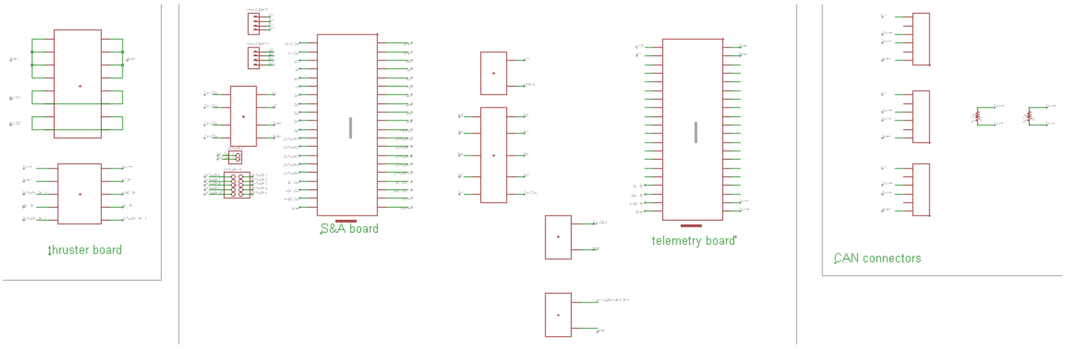


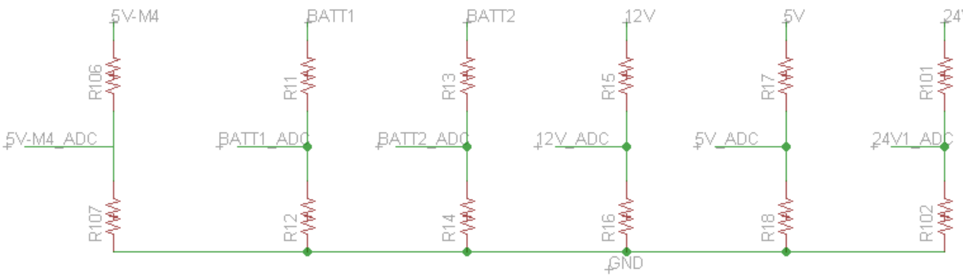
Load balancer



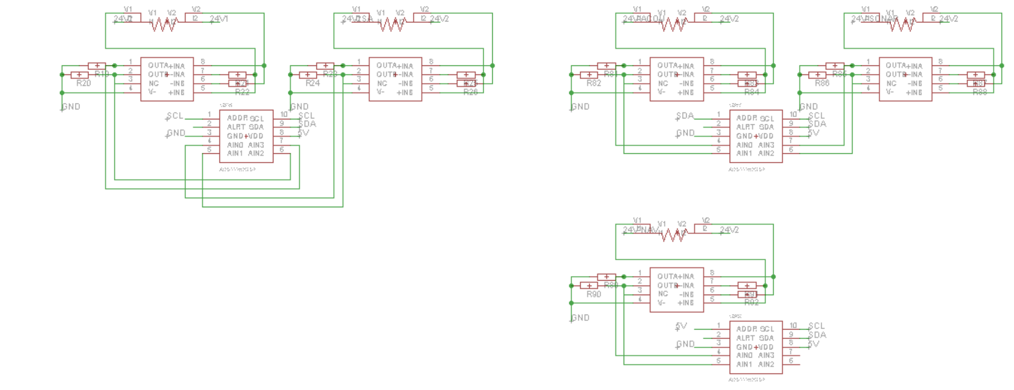
5V-buck regulator

Power over Ethernet injectors

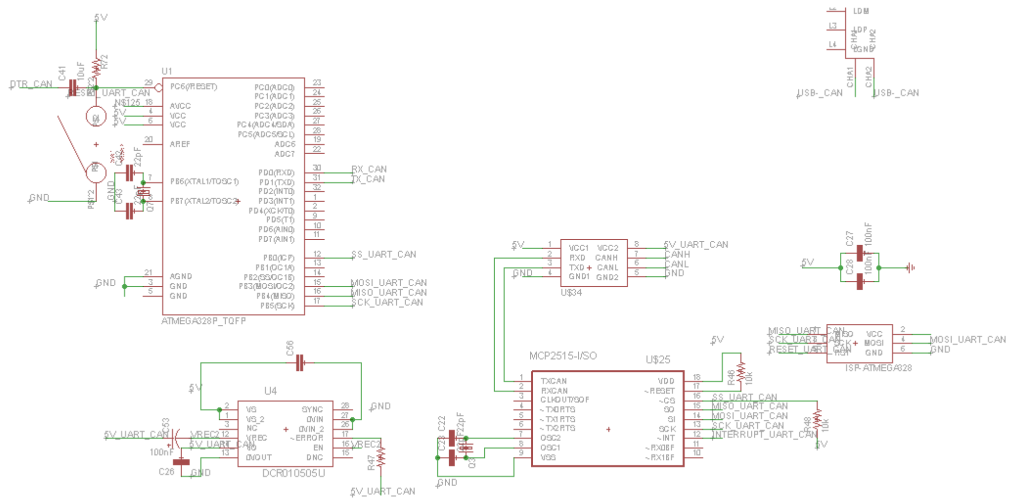




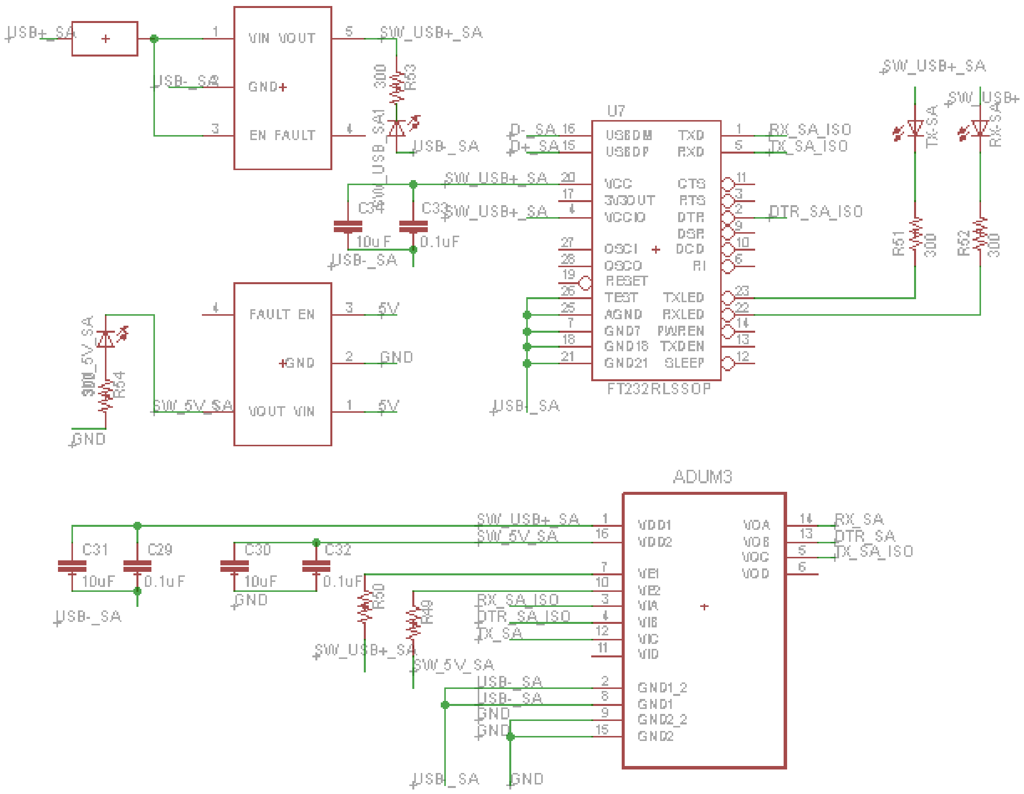
Voltage measurements



Current measurements



CAN-USB converter



Isolated USB-UART converters