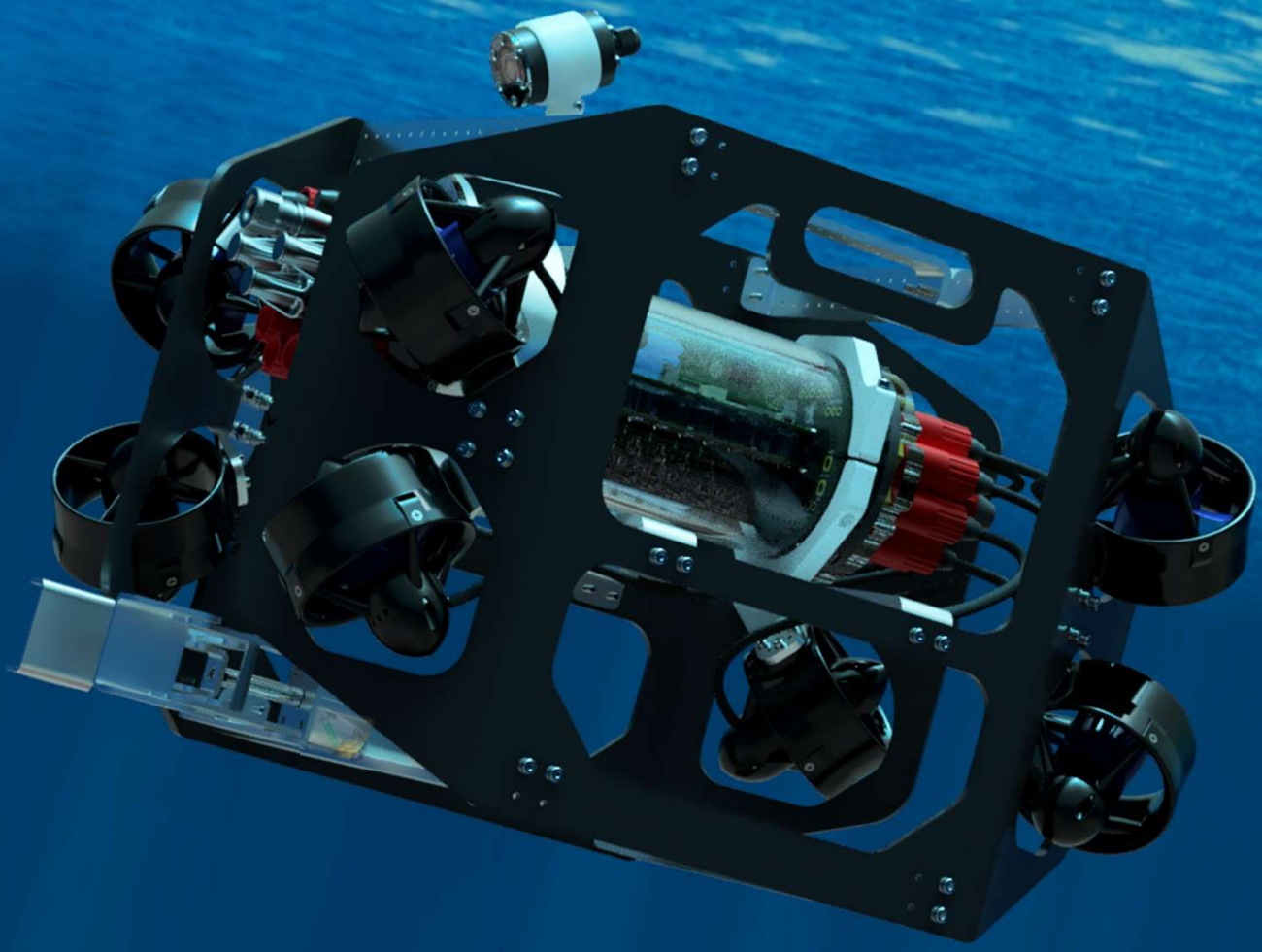
The University Of Sheffield.

AVALON
- Underwater Robotics -

4th Year Group Project Final Report

Group 1: *Design and Manufacture of Electrical and Control Systems for an Underwater ROV*



Contributors: Benjamin Griffiths
Henry O'Keeffe
Joseph Orford
George Osmond

Supervisors: Prof. Dan Gladwin
Prof. Martin Foster
Mr Ken Mitchell

Word Count: 21730

Abstract *(HO)*

Exploration of the environment has been aided by the development of new technology. One of the most difficult locations to explore on earth is underwater, in lakes, rivers, and the sea. Whilst many techniques have been available for some time based on people going underwater, it is only with the recent development of new technology that allows Remotely Operated underwater Vehicles (ROVs) to take their place, allowing safer underwater operations to be carried out as well as reducing setup time and increasing convenience. When operating in very deep water (>4500 m) only ROVs can be used, as the pressure becomes too great for a capsule big enough for a human to withstand.

An ROV typically contains cameras for surveying and to aid navigation, thrusters to propel the ROV and a tether which provides a backup mechanical connection, as well as provides cables for power and signals to be sent to and from the ROV from the surface station. Linking these components together is the electrical system, which must do signal and power conversion and control.

To promote the development of underwater ROVs, the Marine and Technology Education (MATE) ROV competition was set up. This is an international competition held in the USA where undergraduate students compete with their own design of a small (< 30 kg) ROV to complete challenges. These challenges usually involve identifying objects underwater, moving, placing, or retrieving objects, and interacting with other devices, such as valves. The competitors are expected to form, manage, and maintain a team to develop an ROV from scratch or previous design during the academic year.

Avalon ROV is such a team operating at the University of Sheffield. Avalon ROV is split into three sub teams: Software, Electrical and Mechanical. This paper documents the efforts of 4 team members across the Electrical and Software sub teams to develop the electrical and control system of the ROV for 2020.

Table of Contents

1. Introduction

1.1 Project Background *(HO)*

The MATE ROV team 'Avalon ROV' was formed in 2016 with a small multi-disciplinary group of engineering students. Since then, it has competed in four international competitions held in the US whilst the team constantly grew and evolved to include the current members.

The ROV built in the academic year 2018-2019 featured a self-contained modular control system in a watertight enclosure with separate PCBs for thruster/actuator control and low power operations (such as signal decoding, camera control).

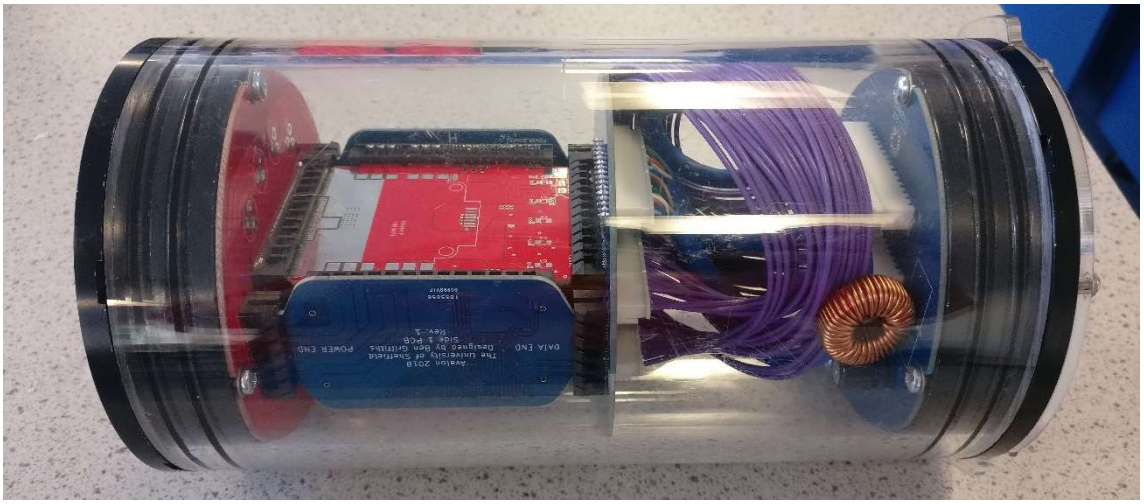


Figure 1: Previous control capsule design.

Due to multiple issues with the design in the previous year (which are beyond the scope of this document), it was decided at the beginning of this project to keep only the basic ethos of a modular, board to board system and design a whole new electronics system for the on-board watertight housing, known as the Control Capsule seen in Figure 1. The acrylic tube housing would be reused.

1.2 ROV Specification *(BG)*

The electronic control systems primary responsibility was manoeuvring the ROV and controlling the on-board devices, hence the specification for the design was highly dependent on the functionality of the ROV.

In particular, the ROV the system is controlling will contain:

- → Thrusters
  - Blue Robotics T200 3-phase Brushless-DC thrusters
- → DC brushed motors
- → Pneumatic solenoid valves used for actuators
- → Digital cameras (USB and Ethernet)
  - Full HD Webcams, PoE IP Cameras
- → Analogue cameras
  - Generic reverse parking cameras
- → Sensors
  - Temperature, Depth, Inertial Measurement Unit
- → A deployable mini-ROV

The extent to which these devices are used depends on the tasks the ROV has to complete at the competition [1]. Furthermore, the control system must be able to interface with the external devices mounted to the ROV whilst isolating the electronics.

1.3 Aims (BG, from PID)

The aims of the project that we initially set out to complete from the project initialisation document are shown below.

1) **Identify functional requirements of the electronics control system and software.**
    a) Identify the ROVs hardware requirements, such as the number and type of the thrusters, actuators, sensors, motors and camera.
    b) Identify the software features required to operate the electronics control system to control all aspects of the ROV.

2) **Design the electronics control system and develop software.**
    a) Split the overall system into its core components to divide the development between all members of the group.
    b) Use computer aided design (CAD) to design the physical structure of the electronics control system so that I can be mounted inside a watertight enclosure.
    c) Identify required underwater connectors and design the mounting plates.
    d) Research and source required components such as microcontrollers and PCB connectors.
    e) Design a mock-up user interface for the ROV control program.
    f) Develop control programs in Python and PyQt5.
    g) Design the circuits for the data, power, interface, electronics speed controller (ESC), testing and surface PCBs.
    h) Route each PCB to the dimensions specified by the CAD model.

3) **Manufacture**
    a) Submit PCBs for fabrication.
    b) Procure necessary components.
    c) Populate PCBs with components.
    d) Manufacture connector mounting plates and assemble capsule end caps.

4) **Test system**
    a) Test functionality of each board separately using breakout boards.
    b) Test the electronic speeds controller thermal response to a range of operating conditions.
    c) Test ROV control program with electronics system.
    d) Identify faults and areas where each board has not met the requirements.

5) **Re-design**
    a) Identify faults and areas where each board has not met the requirements.
    b) Redesign boards where necessary.
    c) Restructure code into libraries for better readability and maintenance.
    d) Fabricate and populate new PCB revision.

6) **Test completed system**
    a) Install electronics control system into the ROV and connect external hardware.
    b) Test full system functionality with the control program and ROV.

In addition to the initial aims, as the project progressed in the electrical and mechanical sub-teams, the need for two additional systems emerged. The first system was a prototype ROV chassis, that would allow the electrical sub-team to mount the electronics control system and test the thrusters, without disrupting the workflow of other sub-teams. The second system was a surface control station, which would be an all-in-one solution to piloting the ROV from the surface. This system would contain a computer and monitors to run the ROV control program and view the camera feeds, as well as external connectors to connect to the ROVs tether.

2. Physical Structure of The Electronics Control System *(BG)*

The electronics is isolated from the water using an underwater capsule, designed by Blue Robotics, that contains a 200mm long clear acrylic tube (4" series) [2], with anodised aluminium end caps at each end, shown in Figure 2. A water-tight seal is achieved with rubber O-rings on the end caps. Underwater connectors. manufactured by MacArtney and Bulgin. are secured to the capsule via acrylic mounting plates bolted onto the end caps. Custom 3D-printed capsule clamps secure the capsule to the ROV chassis, and a capsule extension was designed to mount more connectors than we previously could in past competitions.

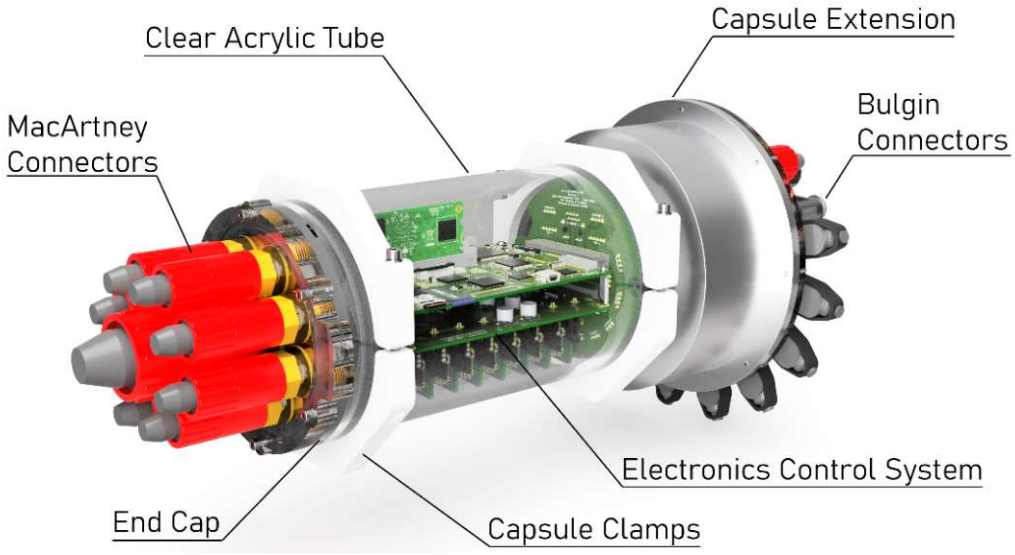


Figure 2: Underwater capsule that contains electronics control system.

The physical structure of the electronics had to be designed to fit inside the control capsule. Special care was taken when designing the shape of each PCB in CAD, taking the tolerance of the capsule's length into consideration to guarantee a reliable connection between the board-board connectors. The system takes the form of a modular, interconnected stack of printed circuit boards (PCB), shown in Figure 3. This design was chosen for a few reasons. Firstly, it allowed us to design the system as a team, with each member being responsible for the design and development of one of the boards. Secondly, it introduced a form of modularity and redundancy into the system, where if a board was to fail, only that single board would be replaced, as opposed to replacing the entire system.

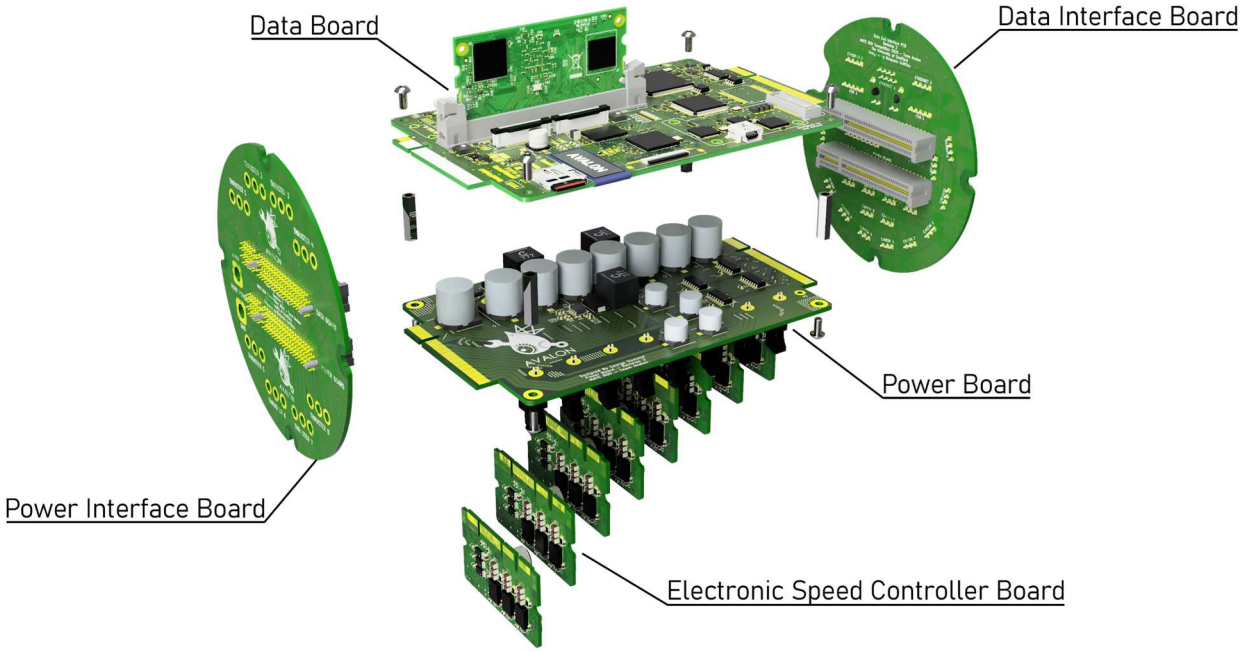


Figure 3: Exploded view of the electronics control system.

3. Data Board *(IO)*

The Data Board is the main controller of the ROV. This PCB receives all of the inputs from the sensors around the ROV, performs PID control, general data processing and sends feedback to the surface and to all the onboard actuators. This board can take both digital and analogue cameras and send this data to the surface computer through analogue, digital and fibre-optic links for multiple layers of redundancy. The second board revision is shown in Figure 4.

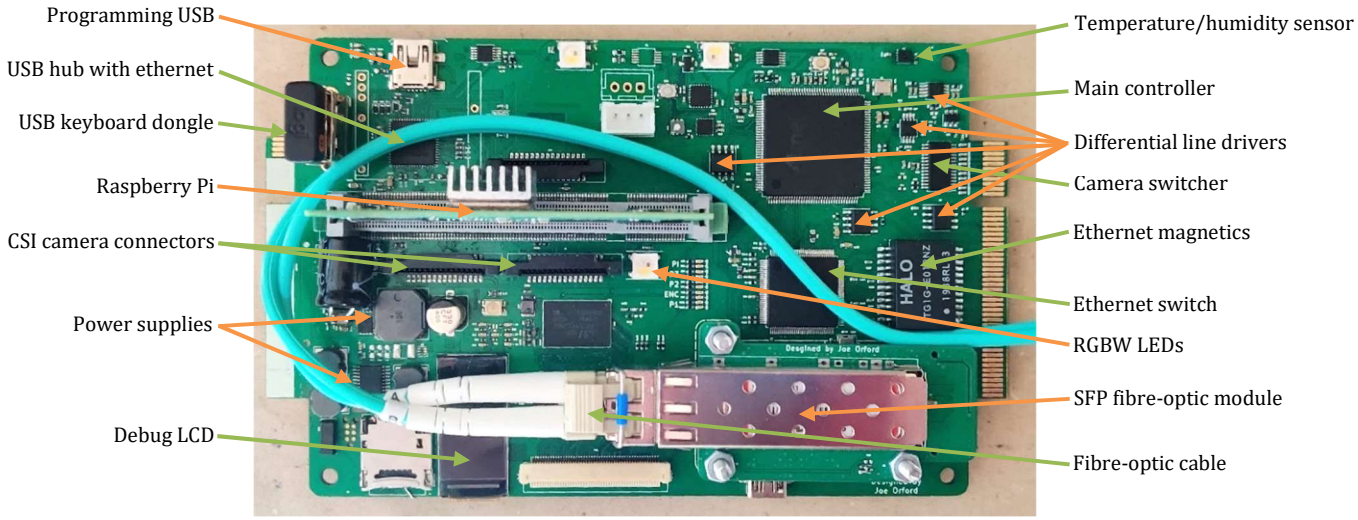


Figure 4: The data board from the top

3.1 Board Specification

- → Powered from +48 V, with on-board DC-DC converters
- → Communicates with the control program on the surface station via serial commands using RS-422 protocol.
- → Receives, switches, and transmits camera signals up the tether.
- → Outputs 4 PWM and 8 direction digital signals to drive 4 H-Bridges on the Power Board.
- → Sends speed commands to ESC modules via RS-485 protocol.
- → Reads values from temperature, depth, and gyro sensors on an I2C bus.

3.2 Design & Implementation

3.2.1.1 Analogue communications

To get low latency video feeds to the surface, analogue video was used. Analogue video has the benefit of not requiring a buffer and is real time. Analogue video in this instance is lower quality, but the trade-off is that low latency can be achieved. As the cameras are analogue, they are more susceptible to EMI, so the decision was made to convert the single ended analogue feed to differential signal up the tether. Over 25m the analogue wire would otherwise act as an antenna creating interference. This was achieved using AD8130 and AD8131 analogue differential line drivers along with resistors for termination and biasing [3] [4]. These line drivers do however need a -3.3V rail in order to generate the differential signals. To achieve this an LM2776 charge pump was used, which is able to generate the -3.3V required [5].

As the tether needs to be lightweight, only two pairs of wire were able to be used for analogue video. This meant only two camera feeds could be sent. As we wanted more than two cameras, an I²C FMS6501 switching matrix was used. This meant up to 12 analogue inputs could be sent to any one of 9 outputs, but only two are used in this instance [6]. However, one of the analogue feed inputs was used to connect the Raspberry Pi's video signal.

3.2.1.2 Digital communications

To communicate with the surface, a full duplex RS-422 link was used. This meant there was a differential pair in the tether for both send and receive so that the ROV could send data whilst receiving commands. To achieve this, a MAX3488 was used at either end [7]. The MAX3488 interfaces with a UART port on the ROV and a USB to UART on the surface computer and acting as a passive bridge.

Throughout the ROV there is an I²C communication bus pulled high to 3.3V. This bus ensures that even if a device were to crash and pull the bus low, no other devices would be damaged, it would just halt communications. The bus is relatively low speed so is only used for the sensors as they only require up to a 10Hz refresh rate. I²C in this implementation runs at 400kHz.

For higher speed interconnects, like the ethernet link for the main controller, SPI was used. SPI has been implemented at 50MHz which gives rise to a very high communication speed, especially because it is full duplex. The main controller acts as a host and controls the slaves on the bus with chip select lines.

For the motor drivers on the power board, GPIOs were wired through the end cap. The motor drivers are simple, so require only two signals for each. The GPIOs were chosen such that each that was sent supported complementary PWM operation. The PWM signals were generated at 20kHz as the drivers this is the drivers rated maximum [8].

3.2.1.3 Optical communications

For video data transfer at 1Gbps with low latency communications over the 25m tether it was necessary to use fibre-optic communications. Fibre optics are not susceptible to Electromagnetic Interference (EMI). Fibre-optics are reasonably flexible with a ~7.5mm bend radius but is extremely lightweight which are both ideal for the tether [9]. The optical communications are full duplex and uses two separate multi-mode fibre optic cables. 1310nm wavelength was selected as this gives the furthest range reasonably achievable only compromising on extra cost [10].

The cable used was OM4 cable, which is rated for 100Gbps over 100m [9]. The connectors on each end were a pair of pre-terminated LC connectors. These are industry standard connectors and are compatible with most SFP modules.

3.2.2 Microcontrollers and processors

The main controller for the data board is the ATSAM3X8EA microcontroller from Microchip. This is a 32-bit microcontroller with multiple UARTs, USB programming, high speed SPI with direct memory addressing and more [11]. This microcontroller was chosen as it was one of the most powerful chips used for the Arduino development environment with a CoreMark score of 95 [12]. This is more than 13x more powerful than the Arduino Mega, the next suitable option [12] [13]. This was chosen as everyone on the Electrical Team was familiar with Arduino. This microprocessor is responsible for the main communications between everything on board the data board. It connects using UART to the RS-485 and RS-422 differential line drivers which link to the surface and the ESCs. It also has two I^2C busses going throughout the board to all the sensors and out of the end cap to external sensors.

To handle extra digital cameras and also image processing, a Raspberry Pi compute module was used. This has a microprocessor which runs Linux and allows for USB or CSI cameras to be connected [14]. This module uses the DDR2 form-factor (it does not use a DDR2 standard interface) and is connected to the board with a locking vertical connector. This is programmed over USB and uses multiple transistors and a USB switch so that it can be easily switched between programming mode and host mode. When in host mode, the USB line is connected to a LAN9514i USB hub which has a built in 100 Mbps ethernet connection [15]. The hub has a USB A port connected to it on the data board so a wireless keyboard dongle can be used, allow for control over the Pi with the tube sealed.

3.2.2.1 Ethernet switching and fibre

To connect multiple ethernet modules to one another, a switch is needed. To achieve this, the KSZ9477, a 7-port gigabit ethernet switch from microchip was used. This switch is managed meaning settings such as port type, speed and IP addresses can be changed over SPI to ensure each port is setup correctly. Five of the ports on the switch are generic 4 pair ethernet, but two use SGMII and RMII protocols [16]. The RMII link was connected to the ATSAM3X8EA, which, although a primary interface, was never utilised. The SGMII link is a serializer-de-serializer (SERDES) meaning it has two higher speed differential lines than 4 pair ethernet, whilst still carrying a gigabit. The main difference between this and needing 4 pairs for the ethernet is range. SGMII runs at 625MHz [16]. This means that taking $1/10^{th}$ of its wavelength as the maximum trace length, results in a ~4.8cm maximum trace. This is much less than the un-boosted 100m maximum specified for a 4 pair CAT-5 cable.

The SGMII link goes through a data connector capable of 25Gbps to a custom daughter board [17]. This daughter board contains an SFP module with a hot-swappable connector and EMI shield [18]. The SFP module takes the SGMII signal and converts it straight to fibre-optic with low latency digital integrated circuits. The SFP module is rated to just over 1Gbps and is plug and play, it can then be monitored over its built in I2C port for diagnostics and identity verification [10]. On the surface an off-the-shelf SFP to ethernet module was used, as there is adequate space to place such a module.

As a backup to the ATSAM3X8EA RMII link, was ethernet module known as the ENC28J60 which was connected to the SPI bus of the ATSAM3X8EA [19]. This was used as the primary network interface for the ATSAM. This module is capable of 10Mbps and uses MDI or MDI-X ethernet to communicate with the hub. It was discovered that 7 ports would not be enough if the ATSAMD microprocessor was to use its RMII link for internet connection. The next hub size would have been 16 ports which is too large. This is because there are few applications of hubs between 7 and 16 ports. To combat this, another 3-port switch, the KSZ8563, was used. This has two ethernet links and an RMII link and is capable of gigabit communications [20].

On the main switch, one of the ethernet links was routed as gigabit to the end cap with 4 pairs and two others were routed but only with two pairs (100 Mbit/s) due to size constraints. These two other connections are for backup and are only capable of up to 100Mbps. The KSZ feature auto negotiation between MDI and MDI-X [16]. MDI and MDI-X are the same apart from the two pairs are switched around. This helps for if an ethernet cable is wired wrong or it is connected to a poorly designed device.

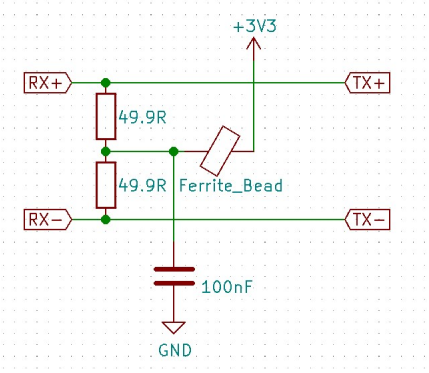


Figure 5: network of passives to bias each ethernet pair

The KSZ series of ethernet switches have built in biasing resistors meaning they can be wired directly to them, however for the ENC28J60 and the Raspberry Pi's USB hub, there is no such luxury [16] [21]. Figure 5 shows the network of passives was added at the inputs and outputs of the ethernet ports [22].

When wiring two ethernet devices, it is necessary to have some level of isolation to prevent mismatched voltages at either end causing unwanted current flow [23]. This can be achieved with magnetics, but these are big bulky units. They do however provide galvanic isolation, which for PoE, which runs at 48V, needs DC biasing and is not necessarily at the same potential as the device connected.

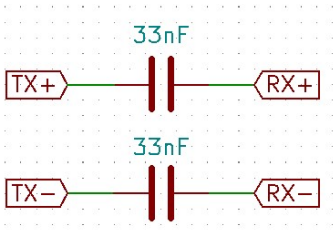


Figure 6: use of capacitors as a way to connect two physical ethernet layers

In this case, this is unnecessary as the devices are connected to the same power rails. To wire the two physical ethernet (Phy) layers together, 33nF capacitors were placed in series which allow for only the signals to pass with no DC biasing [24] [22]. This was added in series with all on board ethernet connections as shown in Figure 6.

3.2.2.2 Sensors

Sensors throughout the board are used in order to monitor all system vials and the current position of the ROV. All are connected via I2C with one internal bus and a second bus for external sensors. Table 1 shows the list of sensors chosen.

Sensor	IC	Location	
Temperature and humidity	SHT31 x2	Either end of data board	[25]
Inertial measurement unit	MPU6050 x2	Centre of data board	[26]
	ICM20948	Centre of data board	[27]
Voltage rail measurement	MCP3423 x3	Distributed across data board	[28]
	MCP3424	Distributed across data board	[28]
Volatile organic compounds	SGP30	On the data board	[29]
Pressure sensor	MS5803 – 15 BAR	Outside on the end cap	[30]
Waterproof temperature	Generic thermistor + MCP3423	Outside on the end cap	[28]

Table 1: list of sensors and their locations on the data board

3.2.2.3 Power Over Ethernet (PoE)

As high-quality cameras were required as well as the low latency analogue cameras, ethernet IP cameras were used as they provide up to 4k video streaming at up to 30fps. This is however a lot of data which uses a theoretical data rate of around 35Mbps. But in reality, this may be higher with lost packets. The issue arises when powering the cameras as it would require two cables. Waterproofing connectors is a difficult procedure and also space hungry. To reduce the number of cables required, PoE was used. This meant the 48V tether input could be used to power the camera over its existing gigabit ethernet link.

The PoE was implemented using the TPS23881, an 8 channel PoE controller. This IC takes configuration from I2C to set the max channel power and controls MOSFETs in order to switch the 48V rail [31]. The magnetics isolate the ethernet differential pairs from the switch and for every two pairs, one pair connected to ground and the other switched to 48V. The chokes in the magnetics ensure that the PoE supply does not affect the data.

When PoE starts up, the TPS23881 does a check to find out if a device is connected, and if so, what classification it is. Some devices have lower power limits than others, so a safety check is performed by seeing how the load responds to pulses [31]. With the camera connected to the PoE bus the start-up signal is seen. An initial pulse is the discovery pulse which is followed by the classification pulse. The main supply then ramps up to 48V once the controller has verified that the load requires it. To protect the PoE module, fuses were installed on each line rated to 1A each. Based on 48V PoE, and with the main 4 pair ethernet, a maximum power delivery of 96W can be achieved.

3.2.2.4 Voltage Rail Demand and Supply

On board the data PCB there are seven voltage rails. All of which are monitored with the aforementioned I^2C Digital to Analogue Converters (DACs) to verify bus stability and also added protection in case of low impedance faults. All power supplies were selected in order to allow for this by over-specifying current ratings with a Safety Factor (SF) and each module has over current protection. The voltage rails and current demand of each rail is shown in Table 2.

Rail	Peak demand	Main uses
48V	4.15A	Powering the 5V rail, power over ethernet devices
5V	5A	All other rails, Raspberry Pi, PoE module, camera servos
3.3V	2.5A	Raspberry pi, microcontrollers, USB switch, fibre optics, general ICs, and sensors
2.5V	1A	Ethernet switch rails, generating analogue video for Raspberry Pi
1.8V	1A	1V35 rail, Raspberry Pi, USB hub, backup IMU
1.35V	0.5A	Microprocessor and RAM
1.2V	1A	Ethernet switch

Table 2: voltage rails and peak current demand

All regulators with the exception of the 1.35 V rail are switch mode power supply units. All switch mode supplies have external inductors and have integrated MOSFETs with the exception of the 5V rail which has external MOSFETs as integrated switches have poor performance at high voltage and current [REF]. The 1.35 V rail uses an ultra-low drop out linear voltage regulator which employs a 3.3 V biasing. This is due to the only 0.45 V drop required going from 1.8 V to 1.35 V. This low drop was chosen as a linear supply is needed as RAM requires a ripple free supply, but linear regulators are inefficient. In using the lower input voltage, an efficiency of around 75% can be achieved.

3.2.2.5 Programming

The main ATSAM3X8EA uses a USB input to program. For this, a mini-USB port was implemented as this is a fairly robust connector. As this microcontroller only has one USB port which is also used as a COM port, an FSUSB42 USB switch was used. By changing a jumper, the USB port on the ATSAM3X8EA could either be connected to the mini-USB or the Raspberry Pi's USB hub. The microcontroller could then be programmed both via the mini-USB port by connecting it to a laptop or via remotely controlling the Pi over the fibre-optic connection.

As remote programming was a major consideration to prevent water ingress, it was important to have a backup method of programming. To do this, a UART bridge was created using an XBEE module. This module runs at 2.4GHz and is wireless [32]. It allows for the pass through of both UART and GPIO pins, meaning the ATSAM could be remotely reset, erased or re-programmed [33]. This can only be done if ROV is above water as 2.4GHz signals at this power can only pass roughly 20 cm through water [34].

3.2.2.6 Other hardware

For debugging purposes, a 0.93 inch 160*80 LCD display was added. This had its own ATmega2560 microcontroller which updated the screen over SPI. There was also an SD card on the SPI bus which allowed the ATmega to display the Avalon logo and any other graphics. It could also be used to display the statistics of the ROV, such as bus voltages, temperatures, and any other warnings.

There was also 13 RGBW addressable LEDs placed on and around the data board. This type of LED was chosen because they only require one GPIO of the ATSAM. The LEDs were added so that the pool could be lit up for the cameras or the tube could glow a specific colour to indicate a fault, i.e. red for "retrieve ROV from pool".

3.2.3 PCB Considerations

3.2.3.1 Impedance Control

To ensure that high speed signals are noise free, differential pairs have been used. These are of a fixed impedance that is controlled with trace width, trace gap and distance from a plane. For USB signals, CSI camera feeds and the DSI display, 90 ohms impedance was needed [35] [36] [37]. For ethernet, analogue cameras and the communications link, 100 ohms was needed. For the single ended video feeds, 75 ohms was used as this is the standard for PAL video [38].

To improve the differential signals, the board was routed to ensure that wherever possible, differential pairs did not have to change layers and also did not cut across two planes. To reduce the space needed for a given differential pair, it was important to have the ground layer underneath the traces be close [37]. The distance between layer 1 and 2 was 0.2mm and the same was true of layer 3 and 4. All layers on the board were 1oz, and due to the manufacturer's capabilities, 0.1mm track and gap spacing had to be observed. Based on this information, using an impedance calculator, Table 3 could be produced giving the dimensions needed to achieve a given impedance.

Target impedance (ohms)	Trace width (mm)	Trace gap (mm)	Actual impedance (ohms)
100 (differential)	0.16	0.10	101.93
90 (differential)	0.22	0.1	89.37
50 (single)	0.39	N/A	50.06
75 (single)	0.16	N/A	75.66

Table 3: dimensions and impedances of differential and single ended traces JLCPCB 4 layer boards

To match the impedance of each line of the differential pair, small additions, called "meanders", to the shortest trace can be added as shown in Figure 7. These ensure that the traces are the same length. The same may be done to both lines in order to match the length of two sets of pairs, like with the ethernet lines. This was however not required for this design as the ethernet switch automatically compensates for any miss-match

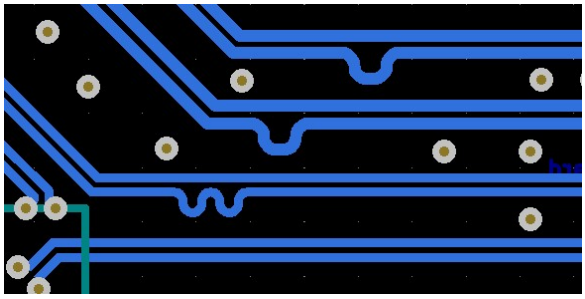


Figure 7: example of multiple differential pairs with meanders

3.2.3.2 EMI Shielding

For high speed communication links like for the SFP, which is very sensitive to EMI, it was important to protect the traces. Even though the signals are differential, there is never 100% coupling between the two traces, this means that a design consideration is to minimise any EMI reaching the SFP lines. To do this, a large ground plane was placed under the traces and they were routed on the top layer of the PCB, away from any other high frequency data lines. The large ground plane meant that EMI from the power board beneath it would be reduced.

For the power supplies, to reduce EMI, shielded inductors were used. This meant that the ferrite core of the inductor surrounded the windings. While this will reduce thermal performance of the inductor, it reduces the emitted EMI, protecting the communication lines. The power supplies operate in the hundreds of kHz which will interfere with signals.

3.2.3.3 Conditioning

For particularly sensitive electronics like the ethernet switch, microcontroller or SFP module, ripple or spikes in the power supply can cause abnormal operation. To combat this, passive components are used to minimise these effects.

For each power pin on the affected devices, a 100nF capacitor was placed as close as possible to the pin. This acts to couple any high frequency noise away from the pin to the ground plane. The SFP module was especially sensitive so a 10nF was also added to further decouple high frequency noise.

As most of these devices are digital, they use an internal or external clock signal. This is in the order of MHz but does give rise to more power being drawn as the data is clocked. At this point, a spike in current is observed. A capacitor in the order of 1 to 10 µF is placed in the local vicinity of each chip on each rail. They are connected with as larger power places as feasible to ensure a low impedance connection. When a current spike occurs, the capacitor acts as a charge pool and means the power supply does not have to respond to the rapid changes.

For certain rails, where EMI is a major consideration, a choke can be used in series with the rail. This helps to further reduce EMI by providing a higher impedance for high frequency noise. The chokes allow DC to pass through whilst blocking the EMI created by other devices on the same power bus.

3.3 Results

3.3.1 Fibre and ethernet

the fibre and ethernet it was important to measure the maximum data rate of the connection. To do this a program called LST (LAN Speed Test) was used [39]. An LST server was setup on a laptop with a known gigabit connection to the surface side of a 30m fibre optic cable. This had an off the shelf SFP module and SFP to gigabit ethernet module. The fibre-optic module was connected to the data board as it would be in the competition with the SFP daughter board and a second laptop connected to the gigabit camera ethernet on the end cap. This meant a full speed test from the camera to the surface could be tested. LST was ran on the second laptop and sent multiple packets back and forth to the server to see how quickly the data could be transferred, the results log is shown in the Appendix, Figure 60.

This shows that a link speed of ~900Mbps was achieved. This is more than enough for the required applications but is not quite the gigabit that all devices are rated to. This could be due to slight imperfections in the ethernet routing or simply overheads with error checking not being accounted for.

To show that the Raspberry Pi was able to connect to the internet via its USB ethernet, the on board ethernet switch and the fibre module, Figure 8 shows an image of a speed test being performed on the Pi. A speed of 94Mbps was achieved. This used the service “Fast.com” and is hosted by Netflix to give you your maximum download speed [40]. The expected maximum speed was 100Mbps as the Pi is only connected with two ethernet pairs through a USB 2.0 [15]. The result of 94 Mbps is a fast enough for the 2 or 3 1080p 30fps camera feeds. As each 1080p feed is compressed, a bit rate of 4mbps will be need per camera feed given the lack of fast moving objects.

Figure 8: internet speed test for the on-board Raspberry Pi compute module

The advantage of having most of what the rover is looking at can be comprised of white, black, and blue (excluding the odd prop which is coloured) and moving relatively slowly reduces bitrate needed. This is because compression algorithms like it when a lot of the data is the same. A firework or confetti would need a much higher bitrate to maintain quality due to the vast number of colours and fast changing scene [41].

The 3 ethernets, which went out of the end cap, are all capable of PoE. To test that PoE was functioning correctly, probes were connected to the power lines on a camera which supported PoE. Figure 9 shows the results of initialising PoE and shows the initial pulses to detect if a module is connected before turning on to full power. The initial 3 pulses are used as a negotiation between the data board and the camera to find its power capabilities so faults can be detected.

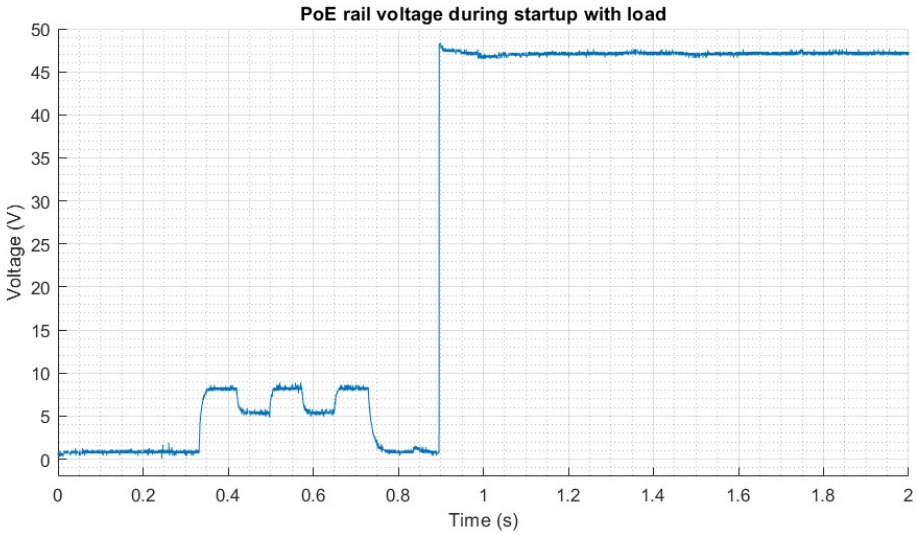


Figure 9: normal operation of PoE at 48v

Issues do however arise when the bus voltage is lowered too far. In this test, the bus voltage was lowered from 48V down to 40V as shown in Figure 10. While the controller was able to negotiate a power level, when it went to full power it was only able to stay online for ~500ms before the camera initialised and due to the lower voltage, resulted in increased currents which the controller registered as a fault and was stuck constantly negotiating and then faulting.

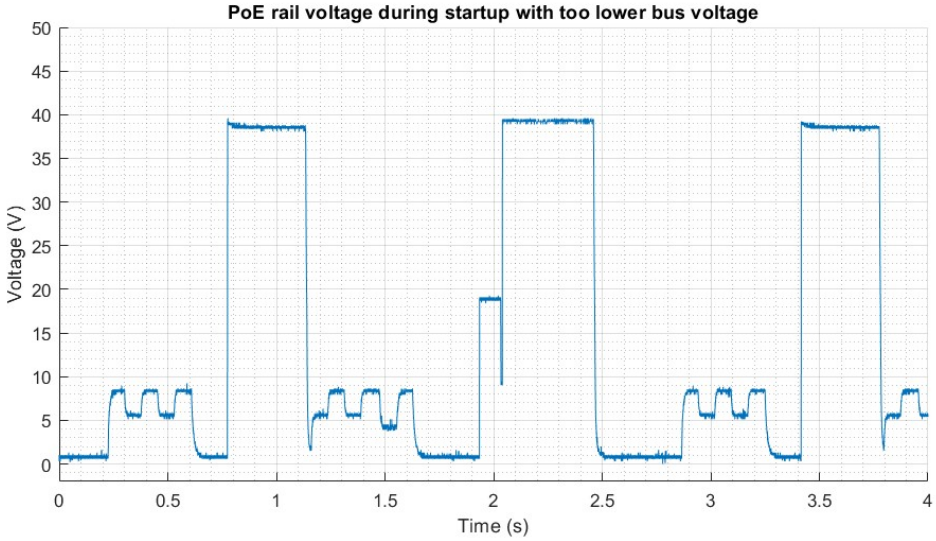


Figure 10: PoE response when the main 48v supply is reduced to 40v

3.3.2 Power supplying

While all voltage rails worked successfully, as shown in Figure 13. The board could successfully power itself from the 48V input and all rails came online. The Raspberry Pi requires the supplies come on in voltage magnitude order, which was evidently successful. The only issue which arose while testing was a ground loop issue. When connecting two USB cables to the same laptop or using a non-isolated scope whilst programming, would result in the 3.3V rail destroying itself. The running theory is that in adding a second ground would cause a large current to flow, even for a brief second. The 3.3V rail controller is rated to 12A whereas the inductor was rated to only 7A [42]. During the current spike this will have caused magnetic saturation and caused the controller to damage itself. In changing the inductor to a 12A rated version resulted in the 3.3V rail not being damaged any more.

3.3.3 Electromagnetic interference

To check how much EMI was being generated by the power supplies, or other items on the board, a test was performed by connecting an oscilloscopes ground clip to the tip. This acted as an antenna for the oscilloscope and could be moved round the board to locate EMI hotspots. Figure 11 shows the Fast Fourier Transform of the noise around the supplies.

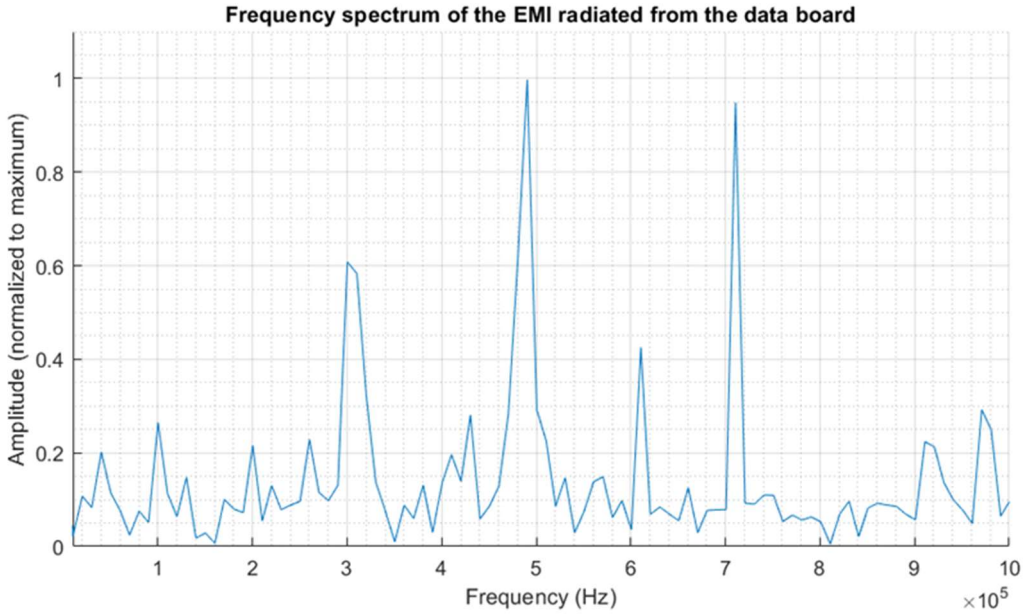


Figure 11: Frequency response of the EMI from the data board constructed with an FFT

From the FFT in Figure 11, and comparing it to the values in Table 4, the 3 most major peaks line up with the frequency at which the power supplies run at [43] [42] [44] [45]. There is a spike just after 600kHz, but this will be a second harmonic of the 300kHz 5V regulator. This data shows that more work can be done to better reduce the EMI from the supplies. The 48V supply has little impact as there is a choke placed in line with the supply, resulting in no visible peak. Also, better inductors could have been used with EMI shields to further reduce these spikes

Supply	Switching frequency
48V tether	260kHz
5V	300kHz
3.3V	500kHz
2.5V	700kHz
1.8V	700kHz
1.35V	N/A (linear)
1.2V	700kHz

Table 4: power supply switching frequencies

3.3.4 Debugging

Figure 12: debug screen in operation with a test card

Figure 13: bus voltages measured and then displayed on Blynk over the internet

To test the display, text was programmed into the ATmega and displayed. This is shown in Figure 12. To test that the voltage rails were all being monitored correctly by the I²C DACs, an app known as Blynk was used and the voltage rails streamed to a mobile phone. The rails can be shown below being monitored in Figure 13.

The LEDs on the ROV for debugging can be shown lighting up in Figure 14. They are in rainbow mode to show their range of colours achievable. They can update their colours at 800Hz which allows for fast moving patterns to be used which can be used to better get the operators attention. Having the white LED as well as the coloured ones means that you can get a purer white for lighting the pool, and compared to without the white LED, leaves you with a very bright LED.

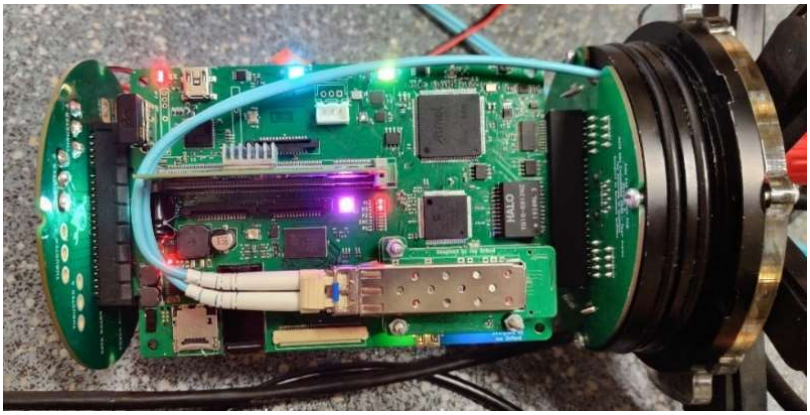


Figure 14: the data board debug LEDs online

3.3.5 Thermals

To test that the board was not overheating and to discover hot spots, a thermal camera was used to measure the boards temperature. During this test, all items on the board were made to be active and online with all three ethernets connected to a PoE device. The power being drawn from the main supply was 7.2W (40V at 0.18A). It was left for 20 minutes until the temperature stabilised. The thermal images of the front and back are shown in Figure 15.

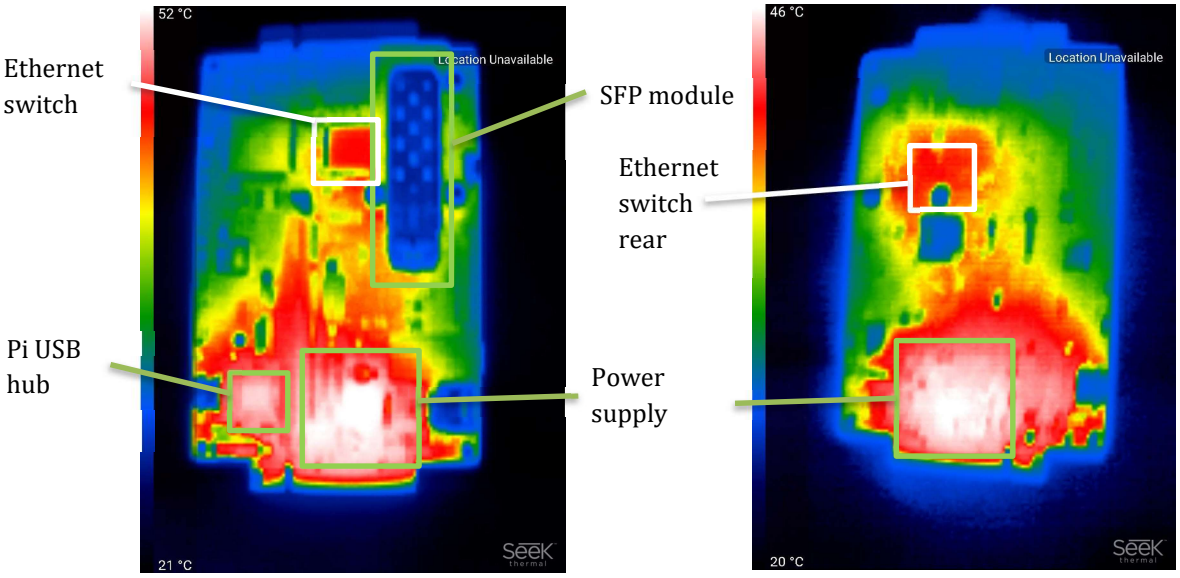


Figure 15: thermal images of the front of the board (left) and rear of the board (right)

The maximum temperature recorded was 66 degrees resulting in a ΔT of 45 degrees. This is higher than optimum as it will be inside a sealed container with powerful motor drivers and regulators. However, this is a worst-case scenario and during the competition, devices which are not being used could be switched off to reduce power consumption. The hot spot on the board can be seen to be where the power regulation is occurring, which is to be expected. There are also hot spots on the Pi's USB hub and the ethernet switch. The SFP module was also warm to the touch but not hot, the EMI shield is very reflective so the thermal camera cannot detect its temperature. A close up on the power supply reveals that the power is being dissipated equally across the 48V to 5V power supply as shown in Figure 16.

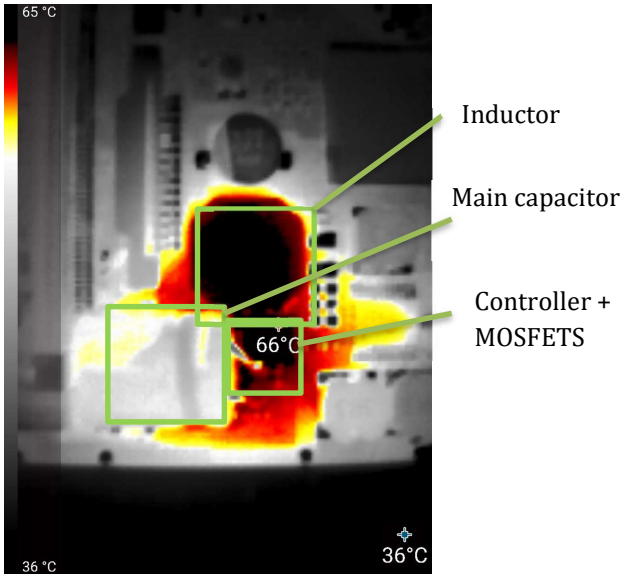


Figure 16: high contrast thermal image of the 5v power supply

3.3.6 Camera latency

To perform the test of camera latency, the camera under test was placed such that it was watching a stopwatch. The stopwatch is being played back on a 90Hz display meaning a resolution of 11.1ms could be achieved. The camera was capable of filming at 240fps and was able to see both the camera feed and the stopwatch. The test setup and results from the PoE camera and Analog cameras is shown in Figure 17.

Figure 17: results of the camera latency tests with the PoE camera left and analogue right. Cameras are to the right of the images watching the phone. The phone is on the right and the camera feed is the phone on the left.

The results from Figure 17 show that the PoE IP camera was able to achieve a latency in the order of 359ms. These results may be inaccurate by ±77.8ms due to the 90Hz camera and due to the 15Hz update frequency which resulted in slow frame loading times. For the analogue camera, 12ms latency was recorded. This again may have been off, but only by 11.1ms due to the camera. The analogue camera is clearly much better in terms of latency, but the IP camera holds victorious in terms of image quality.

3.4 Evaluation

Comparing the results to the initial specifications, it can be seen that, to the best of our testing, all specification points have been met. It was proved the data board could be powered on 48V power and thus does not rely on the power board to function. The data board is however, not the most efficient device, using up to 7.2W when running at full power (excluding PoE devices). If a future revision was to be completed, more care should be taken to reduce this number to just a watt or two if possible.

Communications were tested and shown to work very well, although due to Covid-19, no official tests that could be documented could be performed. The cameras are however displayed correctly over all the communication links as well as the raspberry Pi feeds which was documented. The low latency of the analogue cameras means that piloting the ROV is much easier and gives the pilot more of a feel for the ROV. The control of the motor drivers worked just fine and although not fully documented, could be controlled with the ATSAM. With the real-time nature of the ATSAM, running PID control loops and controlling motor speed is highly reliable.

The RS-485 communication link to the ESCs was successful and allowed for the ESCs to receive their data as will be covered later in the report. The only issue found with the link was potentially the data rate, as often the ATSAM would report that devices did not respond, increasing the data rate would hopefully mitigate this issue. Lastly, while not documented, the I2C sensors including the IMU, temperature sensors and pressure sensors all worked successfully.

Despite the fact that Covid-19 had a profound impact on the ability to test the data board, with minor design changes, the data board is ready for the competition. All critical aspects of the design worked very well and allows for a design which is both robust and feature rich.

4. Power Board (GO)

The power board is responsible for the aspects of the ROV classed as high power, whether they demand higher currents or require a higher operational voltage such as DC-DC converters and H-Bridge controllers. It was decided that the high-power aspects would be dealt with separately to the data aspects due to the high current traces that, if run in close proximity to sensitive data lines, could interfere with high-speed communication lines and sensitive camera signals. By having separate boards for data and power, the control enclosure can be made smaller with the two boards stacked on top of each other, with interface boards that connect them all together and allow for communication between all parts of the control system.

4.1 Board Specifications

With the basic description set out for the overall functionality of the ROV a more in depth and board specific specification can be developed to ensure the power board fulfils all the requirements.

- → It must be able to convert 48 V supplied from the tether down to 12 V at 10 A using a DC-DC convertor to power external actuators, analogue cameras and the mini ROV. This can be achieved with two 12 V at 5 A converters if necessary.
- → Convert 48 V to 5 V at 5 A using a regulator to power logic onboard each ESC module and serve as a backup for the 5 V rail on the data board
- → It must be capable of controlling external actuators such as motors or solenoid valves.
- → ESC modules using individual connectors sufficiently rated to full operation of each ESC.
- → The board must be able to withstand temperatures of up to 160°C and current levels of around 40 A while remaining fully operational for a minimum of 20 minutes.

Board Interface	Inputs to Power Board	Outputs from Power Board
Power End Cap	→ 48 V	→ 8x 3 phase thruster traces
Data End Cap	→ RS485 communication → 4x PWM signals → 8x Enable lines (2 per H-Bridge)	→ 12 V for cameras and Mini ROV → H-Bridge lines for motors or actuators → 5 V for rail redundancy
ESC Interface	→ 8x 3 phase thruster traces	→ RS485 communication → 48 V → 5 V → Ground → Hardwired address for slot recognition

Table 5: Requirements for Each Form of Power Board Interface

4.2 Design & Implementation

4.2.1 Considerations

The major concern for the power board was being able to handle the 7 A RMS drawn by each phase of each of the eight ESCs. This required twenty-four traces that would withstand the high current. To do this a piece of software called Saturn PCB was used which accounted for different attributes of a conductor and was able to calculate set parameters. The most important parameters which would help the decision were the conductor width, conductor current and the temperature rise (ΔT). The first step was to work out how wide the conductors could be, after many different routing attempts it was found that a trace width of 1.8 mm was as wide as possible, while still allowing for all twenty-four traces to be routed. Figure 18 shows how the thickness of copper affects the current carrying and thermal capabilities of a trace. The sets of data labelled as “4 layer” assume a plane present on the two middle layers of the board at a depth of 0.22mm from the top layer with a copper density of 1oz/sq. ft. This however will not be the case in the final power board as the thickness of copper available for internal layers is only 0.5 oz/sq. ft and there will not be a continuous plane due to the various requirements on the board. Therefore the two data sets for a four-layer board do not accurately represent the performance of the power board; thus they shall be used as guidance when making the final

decision. However it can be seen how a ground plane affects the capabilities for a conductor to carry current. In both 1 oz and 2 oz cases there being a significant increase in current at a set temperature.

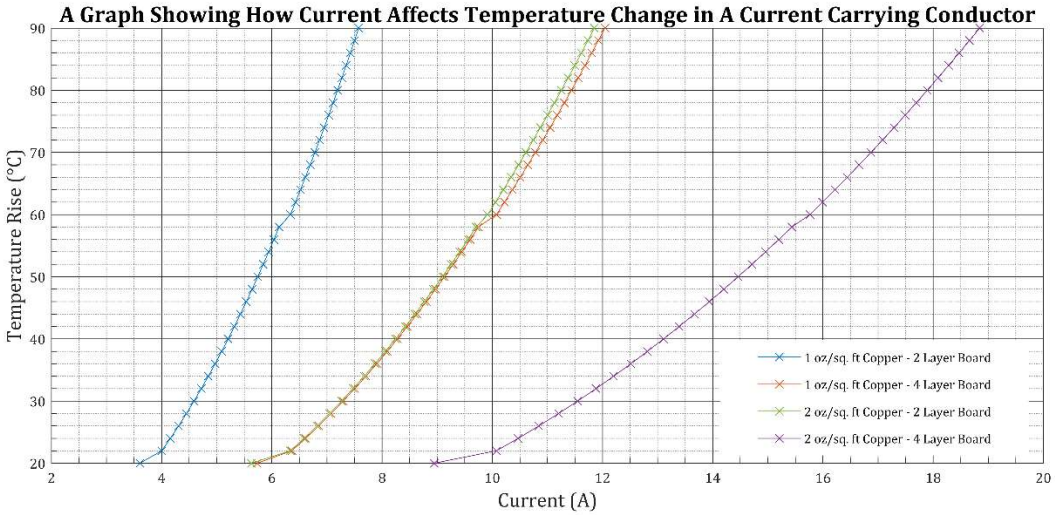


Figure 18: A Graph Showing How Current Affects Temperature Change in A Current Carrying Conductor

This shows that for a 1 oz/sq. ft board of two layers, that in order to carry 7 A, the ΔT would be 75°C which is far too high for the enclosed control capsule. The only way to reduce this temperature would be to make the traces wider, this however cannot be done as previously mentioned, 1.8mm is the widest possible for the traces. This rules-out a weight of 1 oz with two layers, leaving a 2 oz board which yields a ΔT of 28°C when operating at 7 A. A much more suitable operating temperature can be achieved with a 2 oz/sq. ft copper density and can be further improved by moving to a four-layer board with a combination of planes on the middle layers. This wouldn't yield as high a benefit as shown on the graph for a four-layer board, however it will help to reduce the thermal effects. Both of these options will be used in different revisions of the board, the final design utilising 2 oz/sq. ft with four layers. The added copper weight also brought with it an issue when soldering, as the board took longer to heat up.

4.2.2 Voltage Regulation

4.2.2.1 Revision 1

The initial design for the power board used a single 12 V DC-DC convertor capable of 10 A which would be used to power the cameras, mini ROV and external motors and actuators. For this a Texas Instruments L5146-Q1 synchronous buck controller [43] was used to provide the power to all components requiring 12 V. During testing of this converter when there was no load attached to the output, the rail happily sat at just below 12 V but as soon as a load was applied the controller was not able to react correctly and the rail then fell to 0 V. It was initially assumed that there was a pin miss soldered or a bridged pad, so the first step was to re-flow the components and try to clean up any dry looking joints. This however didn't resolve the issue and a lack of suitable equipment and materials (stencil and solder paste) made the recovery of the PCB unlikely, so the board was redesigned.

As well as this, there was a 5 V regulator capable of up to 5 A, used to provide 5 V to each ESC for on board logic as well as a jumper connection to the 5 V rail on the data board which would allow for power to be shared between the boards and try to keep both boards operational. This was achieved using a MIC2128 synchronous buck controller [43] from Microchip. Unfortunately during manufacture the 5 V regulator faced the same issues which were faced by the 12 V converter with difficult to solder pads. When testing this regulator it was obvious that there were bridged pads as continuity tests showed up multiple traces which shouldn't be connected to be connected. Similarly to the 12 V regulator, without the correct equipment recovery was unlikely, so this also required a redesign.

4.2.2.2 Revision 2

The second revision of the power board branded a complete redesign of both the 5 V and 12 V regulators. The redesign made use of the TPS54560BDDA by Texas Instruments [46] for both the 5 V and 12 V regulators. Despite this regulator being limited to 5 A it was decided that the simpler IC design would make manufacture easier due to ease of soldering. With this device only able to deliver a maximum of 5 A, two were used to deliver 12 V, with one responsible for delivering power to the cameras and mini ROV and the second being responsible for delivering power to the H-bridges

which would then power the external actuators and motors. It was thought that this would increase reliability as there was less strain being placed on a single regulator and spreading some of the load between two. The other bonus which came from using this IC was the reduced footprint which allowed for more space on the board to be taken up by planes and motor traces.

The TPS54560BDDA is most commonly used in a 5 V at 5 A setup so for this the standard application was taken from the data sheet and used within the control system. This did however require a redesign to be able to deliver the 12 V required. Following the design rules set out in the IC's datasheet, a new design was able to be created with a new list of components.

The first step in the design was to set out the requirements for the regulator, Table 6 shows these parameters.

Design Parameter	Required Values
Output Voltage (V_o)	12 V
Transient Response 1.25 A to 3.75 A Load Step	ΔV_{OUT} = 4% of V_o
Maximum Output Current (I_o)	5 A
Input Voltage (V_{in})	48 V
Output Voltage Ripple (V_{ripple})	0.5% of V_o
Start Input Voltage (Rising V_{in})	6.5 V
Stop Input Voltage (Falling V_{in})	5 V

Table 6: Design Parameters for 12 V at 5 A Regulator

With the design parameters set out, the next step is to calculate the switching frequency of the controller. Equation 1 and Equation 2 are used to calculate the maximum switching frequency, taking the lowest of the two values as the maximum. The standard on time for the controller (t_{onmin}) is 135ns which means that the frequency needs to be low enough to ensure that the controller can switch on and off correctly within in a cycle. With a maximum input voltage (V_{max}) of 60 V, assuming the voltage drop of the diode (V_D) of 0.7 V, inductor resistance (R_L) of 11 mΩ, switch resistance (R_s) of 92 mΩ and a short circuit output voltage (V_{SC}) of 0.1V the maximum frequency is calculated as 841 kHz. For this design a switching frequency of 500 kHz was chosen to ensure the controller sits comfortably below the maximum operating frequency ensuring no pulses are skipped.

$$f_{sw(maxskip)} = \frac{1}{t_{onmin}} \times \left(\frac{I_o R_L + V_o + V_D}{V_{max} - I_o R_s + V_D}\right) \tag{1}$$

$$f_{sw(maxskip)} = \frac{1}{135 \times 10^{-9}} \times \left(\frac{5 \times 11 \times 10^{-3} + 12 + 0.7}{60 - 5 \times 92 \times 10^{-3} + 0.7}\right) = 1.57\ MHz$$

$$f_{sw(shift)} = \frac{8}{t_{onmin}} \times \left(\frac{I_o R_L + V_{SC} + V_D}{V_{max} - I_o R_s + V_D}\right) \tag{2}$$

$$f_{sw(shift)} = \frac{8}{135 \times 10^{-9}} \times \left(\frac{5 \times 11 \times 10^{-3} + 0.1 + 0.7}{60 - 5 \times 92 \times 10^{-3} + 0.7}\right) = 841\ kHz$$

In order to set this frequency a resistor (R_T) is placed between the clock pin and ground. Equation 3 is used to calculate the correct value to set the frequency. For this a 193 kΩ resistor is required.

$$R_T = \frac{101756}{(f_{SW})^{1.008}} = \frac{101756}{(500 \times 10^3)^{1.008}} = 193\ k\Omega \tag{3}$$

Next is to calculate the output inductor value (L_o), this requires a ratio which is used to represent the amount of inductor ripple current relative to the maximum output current (K_{ind}), a standard value for this is K_{ind} = 0.3. Equation 4 is used to calculate the minimum inductor value which then allows for current ripple ($I_{L(ripple)}$), RMS current ($I_{L(RMS)}$) and peak current ($I_{L(peak)}$) through the inductor using Equation 5, 6 and 7 respectively. The minimum inductor value calculated is 12.8 µH and the nearest standard value which doesn't go below this minimum value and is able to handle the current demands is 15.3 µH.

$$L_{O(min)} = \frac{(V_{max} - V_o)}{(I_o \times K_{ind})} \times \frac{V_o}{V_{max} \times f_{SW}} = \frac{60 - 12}{5 \times 0.3} \times \frac{12}{60 \times 500 \times 10^3} = 12.8\ \mu H \tag{4}$$

$$I_{ripple} = \frac{V_o \times (V_{max} - V_o)}{(V_{max} \times L_o \times f_{SW})} = \frac{12 \times (60 - 12)}{60 \times 15.3 \times 10^{-6} \times 500 \times 10^3} = 1.28\ A \quad (5)$$

$$I_{L(RMS)} = \sqrt{I_o^2 + \frac{1}{12}\left(I_{L(ripple)}\right)^2} = \sqrt{5^2 + \frac{1}{12}(1.28)^2} = 5.014\ A \quad (6)$$

$$I_{L(peak)} = I_o + \frac{I_{L(ripple)}}{2} = 5 + \frac{1.28}{2} = 5.64\ A \quad (7)$$

The next calculation in the design was to aid selection of a flyback diode which needs to have a larger reverse voltage than the maximum input voltage and have a peak current rating greater than the peak current expected to flow through the inductor to ensure no damage occurs. Equation 8 shows how much power is dissipated in the diode which meant that this could then be checked against a chosen diodes datasheet to see if it was capable of handling it. The diode which was initially chosen was the B560C from Diodes Incorporated [47] as this is the same diode used in the 5 V regulator. This particular diode has a junction capacitance (C_j) of 300 pF which is required for this calculation. The power dissipation calculated is 3.08 W which is well within the operating region of the diode.

$$P_D = \frac{(V_{max} - V_o) \times I_o \times V_D}{V_{max}} + \frac{C_j \times f_{SW} \times (V_{max} + V_D)^2}{2} \quad (8)$$

$$P_D = \frac{(60 - 12) \times 5 \times 0.7}{60} + \frac{300 \times 10^{-12} \times 500 \times 10^3 \times (60 + 0.7)^2}{2} = 3.08\ W$$

The output capacitor required is four 22 µF capacitors, however when it comes to selecting the correct capacitor, the RMS output current needs to be calculated to ensure the capacitor will be able to handle the current levels. Equation 9 is used to calculate the RMS output current through the capacitor. The calculated value for the current is 370 mA, this will be taken into account when selecting a capacitor to use.

$$I_{C(OUT)} = \frac{V_o \times (V_{max} - V_o)}{\sqrt{12} \times V_{max} \times L_O \times f_{SW}} = \frac{12 \times (60 - 12)}{\sqrt{12} \times 60 \times 15.3 \times 10^{-6} \times 500 \times 10^3} = 370\ mA \quad (9)$$

The input capacitor needs to be rated to a higher voltage than the maximum input voltage and also have a greater ripple current rating to the maximum ripple current at the input. Equation 10 calculates the input ripple current to aid selection, 11, taking the minimum input voltage (V_{min}) as 40 V. The input ripple current is calculated as 2.29 A, for this, four 2.2 µF capacitors were chosen rated to 75 V and a ripple current rating of 3 A, this gives a total input capacitance (C_{IN}) of 8.8 µF. The input capacitance is used to determine the ripple of the input voltage, this is shown in Equation.

$$I_{C(RMS)} = I_o \times \sqrt{\frac{V_o}{V_{\min}} \times \frac{(V_{min} - V_o)}{V_{min}}} = 5 \times \sqrt{\frac{12}{40} \times \frac{40 - 12}{40}} = 2.29A \quad (10)$$

$$\Delta V_{IN} = \frac{I_o}{C_{IN} \times f_{SW} \times 4} = \frac{5}{8.8 \times 10^{-6} \times 500 \times 10^3 \times 4} = 284\ mV \quad (11)$$

The bootstrap capacitor required which needs to be placed between the BOOT and SW pins must be a 0.1 µF capacitor rated to a minimum of 10 V to ensure damage isn't incurred during operation.

The undervoltage lockout setpoint (UVLO) is used to monitor the voltage across the bootstrap capacitor and turn off the controller when the voltage goes below the minimum operational voltage of the controller. In order to set the undervoltage lockout point, two resistors are used to produce the 6.5 V and 5 V start stop voltages. The controller must be able to turn on when the input voltage rises above 6.5 V and be able to turn off when the voltage falls below 5 V. Equation 12 and Equation 13 are used to calculate the two resistor values required to create the set point, where I_{HYS} is the hysteresis current at 3.4 µA, V_{ENA} is the voltage required on the enable pin to turn the device on at 1.2 V and I_1 is the internal pullup current source that enables operation at 1.2 µA.

$$R_{UVLO1} = \frac{V_{START} - V_{STOP}}{I_{HYS}} = \frac{6.5 - 5}{3.4 \times 10^{-6}} = 441\ k\Omega \quad (12)$$

$$R_{UVLO2} = \frac{\frac{V_{ENA}}{V_{STAR} - V_{ENA}}}{R_{UVLO1}} + I_1 = \frac{1.2}{\frac{6.5 - 1.2}{442 \times 10^3} + 1.2 \times 10^{-6}} = 90.9\ k\Omega \quad (13)$$

There are two resistors which create a potential divider and set the output voltage by holding the feedback pin at a certain voltage. Equation 14 is used to calculate the high side (R_{HS}) resistor, where the low side (R_{LS}) resistor can be sensibly chosen and the equation will create the required high side value to select the correct voltage. Taking the low side resistor to be 10 kΩ, the high side resistor was calculated to be 140 kΩ.

$$R_{HS} = R_{LS} \times \frac{V_{OUT} - 0.8}{0.8} = 10 \times 10^3 \times \frac{12 - 0.8}{0.8} = 140\ k\Omega \quad (14)$$

Finally, the compensation network can be calculated. The compensation network is used to vary the operational frequency to ensure that the output voltage stays within the desired region. This is a feedback network that takes the current voltage and compares it to the desired through the network of resistors and capacitors which will be calculated. First the modular pole ($f_{P(mod)}$) and modular zero ($f_{Z(mod)}$) must be calculated, this is done in Equation 15 and Equation 16 respectively where R_{ESR} is the equivalent series resistant of the output capacitors, which from their datasheet is 46.9 mΩ. Using these two values the crossover frequency can be calculated which is the geometric mean value of two separate crossover frequencies (f_{co1} & f_{co2}) calculated in Equation 17 and Equation 18 with the final crossover frequency calculated in Equation 19. The final crossover frequency was taken as 9 kHz in an attempt to improve transient
response.

$$f_{P(mod)} = \frac{I_o}{2\pi \times V_o \times C_{OUT}} = \frac{5}{2\pi \times 5 \times 88 \times 10^{-6}} = 753.6\ Hz \quad (15)$$

$$f_{Z(mod)} = \frac{1}{2\pi \times R_{ESR} \times C_{OUT}} = \frac{1}{2\pi \times 0.0469 \times 88 \times 10^{-6}} = 38.56\ kHz \quad (16)$$

$$f_{co1} = \sqrt{f_{P(mod)} \times f_{Z(mod)}} = \sqrt{753.6 \times 38.56 \times 10^3} = 5.39\ kHz \quad (17)$$

$$f_{co} = \sqrt{f_{P(mod)} \times \frac{f_{SW}}{2}} = \sqrt{753.6 \times \frac{500 \times 10^3}{2}} = 13.73\ kHz \quad (18)$$

$$f_{co} = \sqrt{f_{co1} f_{co2}} = \sqrt{753.6 \times 38.56 \times 10^3} = 8.6\ kHz \quad (19)$$

The compensation resistor (R_C) is calculated using Equation 20, assuming that the transconductance of the power stage (gmps) is 17 A/V and the transconductance of the amplifier (gmea) is 350 μA/V with a reference voltage (V_{ref}) of 0.8 V this yields a resistance of 125.5 kΩ. For the compensation capacitor (C_C) in series with the resistor, Equation 21 is used taking the previously calculated resistor value. There is also a capacitor which runs in parallel (C_P) to the capacitor and resistor which is used to introduce a compensation pole, for these two values are calculated using Equation 22 and Equation 23 and the largest value is taken. Therefore for this capacitor, 32.9 pF was chosen.

$$R_C = \left(\frac{2\pi \times f_{co} \times C_{OUT}}{gmps}\right) \times \left(\frac{V_o}{V_{REF} \times gmea}\right) \quad (20)$$

$$R_C = \left(\frac{2\pi \times 9 \times 10^3 \times 88 \times 10^{-6}}{17}\right) \times \left(\frac{12}{0.8 \times 350 \times 10^{-6}}\right) = 125.5\ k\Omega$$

$$C_C = \frac{1}{2\pi \times R_C \times f_{P(mod)}} = \frac{1}{2\pi \times 125.5 \times 10^3 \times 753.6} = 1.68\ nF \quad (21)$$

$$C_P = \frac{C_{OUT} \times R_{ESR}}{R_C} = \frac{88 \times 10^{-6} \times 0.0469}{125.5 \times 10^3} = 32.9\ pF \quad (22)$$

$$C_P = \frac{1}{R_C \times f_{SW} \pi} = \frac{1}{125.5 \times 10^3 \times 500 \times 10^3 \times \pi} = 5.07\ pF \quad (23)$$

4.2.3 H Bridges

In order to control the external actuators that the ROV will use, four h-bridges were used. The tasks set out within the competition will require different external actuators, such as motors or solenoid valves. The motors require a h-bridge in order to have bidirectional control or if just a singular direction of rotation is required then a simple half bridge can be used. The solenoid valves also require a half bridge to operate as they automatically close again when power is removed. To fulfil this requirement the VNH7070ASTR by STMicroelectronics [8] were chosen as they are capable of an output of 15 A, with a high operating temperature of up to 150°C and capable of being configured in either a full h-bridge set up or as two half bridges allowing for a variety of combinations for external actuators as shown in Figure 19.

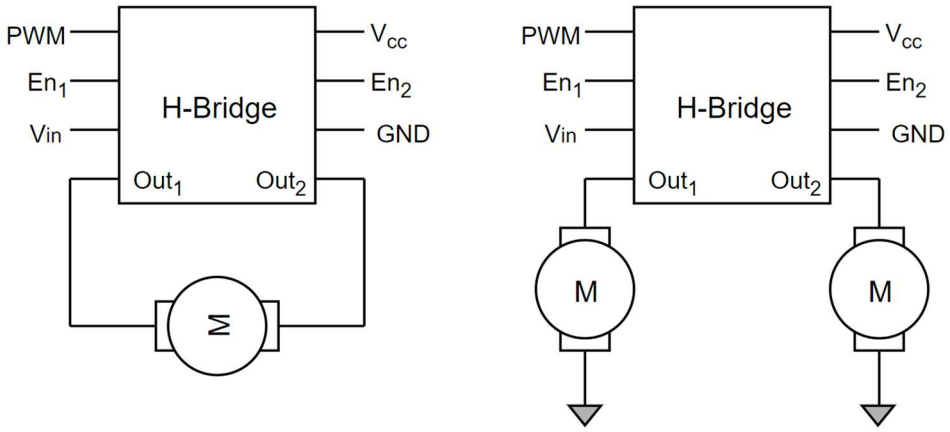


Figure 19: Basic Diagram Showing the Configuration of the VNH7070 H-Bridge as a Full Bridge (left) or Two Half Bridges (right)

Configuration of the h-bridges uses two enable pins to set the device in either a full h-bridge setup or as two half bridges. The signal required for this comes from the Arduino on the data board via the data end cap interface board, allowing for simple reconfiguration when testing different actuator combinations. This set up allows for a maximum of four external motors with bidirectional control or up to eight solenoid valves and unidirectional motor control or a middle combination of both to meet various requirements. A PWM line is also required for the full h-bridge to operate successfully. This signal also comes from the Arduino and is capable of controlling the speed of the motor. An example of the use for this is the mini ROV tether which needs to be long enough for the mini ROV to complete its task but when not in use needs to be tided away to avoid getting in the way of other challenges. This is achieved using a small motor which is used as a winch to reel the mini ROV back in and keep the tether neatly packed away.

4.2.4 Routing and Component Selection

Initially, routing the PCB was thought to be very difficult if using a 1 oz/sq. ft board as the traces would have to be at least 4 mm wide to ensure they could carry the current required without reaching a high temperature. Due to the connectors for the ESCs taking up a lot of the space on the underside of the board, running 4mm traces was going to prove difficult due to the small gap between the edge connector and the first ESC connector which would quickly become full. The decision to use a 2 oz/sq. ft board meant that the trace width could be reduced down to 1.8 mm which allowed for more traces to leave the edge connector and get passed the first ESC without getting trapped. Where possible, traces were routed from the underside of the board, to a space where a minimum of five vias could be placed and the trace then continued on the more open and free space that was available on the top of the board, then finishing off with vias back to their respective pads on the ESC connector footprint. At last five vias were required to ensure that the required 7 A could easily flow through from one side of the board to the other without causing damage by trying to get through a single via.

When choosing components for the regulators the main consideration had to be the size, with height being especially important. This was due to the way the boards were stacked in the capsule, leaving only 16 mm of space between the power board and data board. There needed to be enough clearance between the top of any components and the underside of the data board to ensure that the two didn't come into contact and potentially short out. The main components that needed to be measured were the large electrolytic capacitors and inductors as they protruded from the board the most.

4.3 Results

With the final revision design completed, all the boards were sent off for manufacture, however this time with a stencil ordered as well, so that solder paste could be used and a re-flow oven to make sure that all components were soldered correctly unlike during the manufacture of revision 1. Figure 20 shows the top of final design with all components populated except for a couple which were removed during testing. Figure 21 shows the bottom of the power board with each ESC slot populated with a custom ESC that also has a large copper block on to act as thermal mass for thermal management during operation.

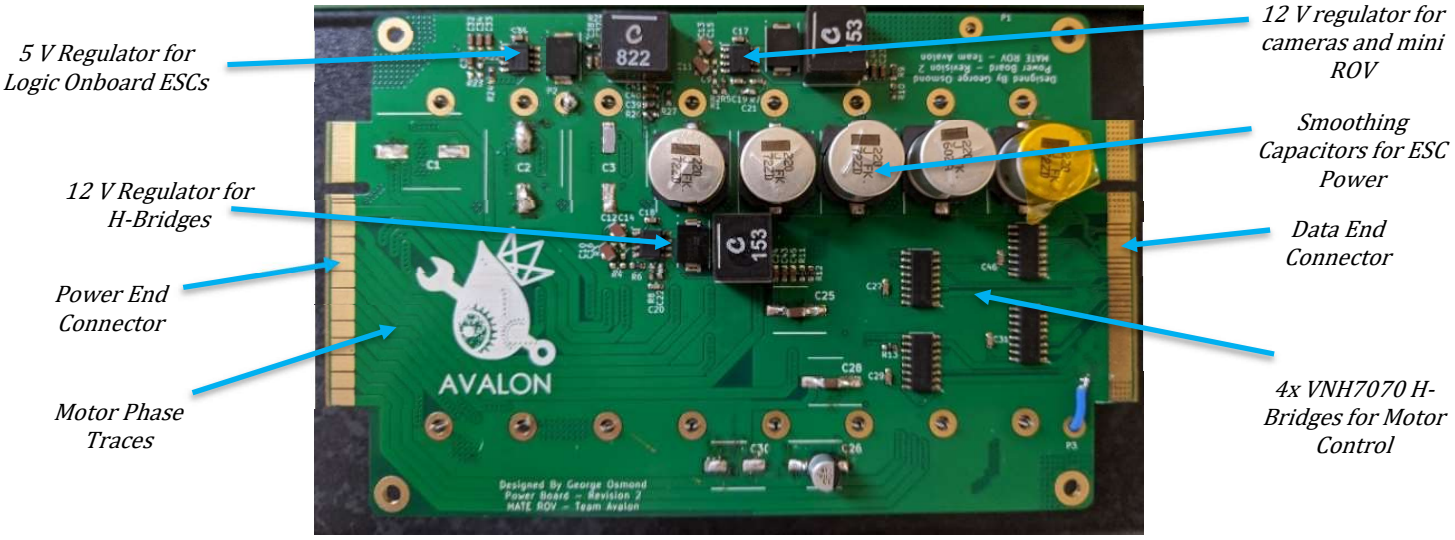


Figure 20: Fully Populated Power Board – Top

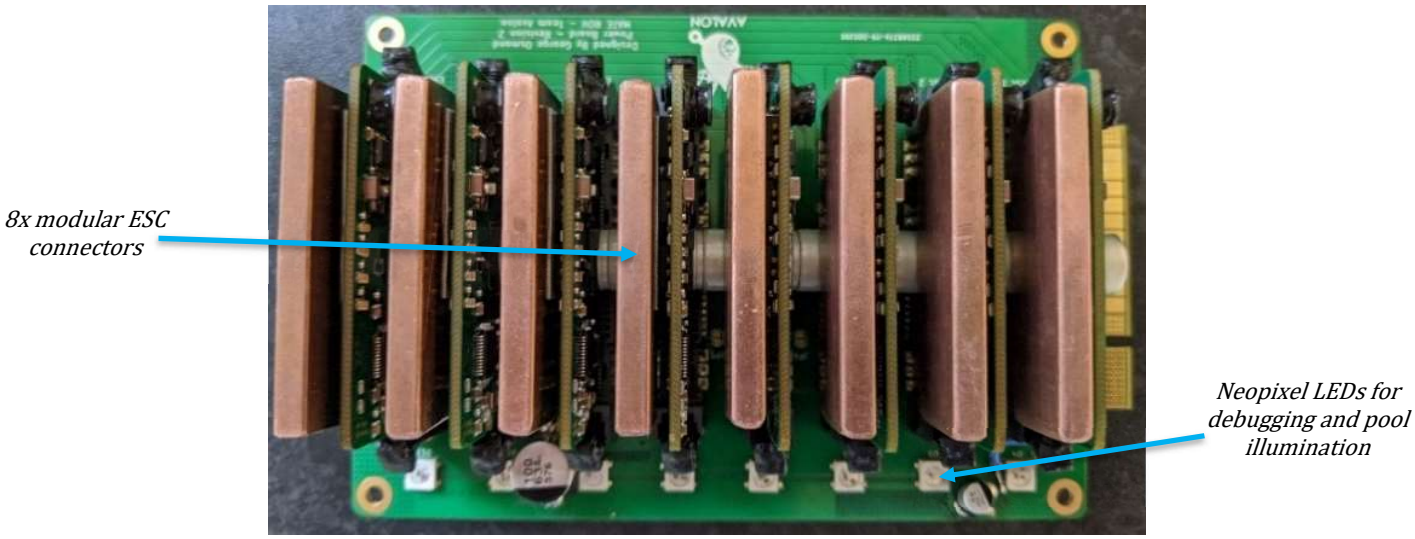


Figure 21: Fully Populated Power Board with Mounted ESCs - Bottom

4.3.1 Output Voltage Regulation

The most important aspects of the power board that needed testing were the 5 V and two 12 V regulators to ensure the board was going to receive all the power it needed to operate. If these didn't work, then the h-bridges and ESCs couldn't be tested on the board or have the whole system tested as a completed capsule. Figure 22 shows how the output voltage of the 5 V regulator is regulated by the controller to ensure it is kept at the desired voltage. A varying load was applied to the output to simulate the different current demands that the regulator could face. It can be seen on the graph that apart from a couple of anomalous points when the output current is less than 0.5 A, the regulation is linear with increasing current demand. However when the scale of the Y-axis is taken into account, it can be seen that the output voltage only changes by 12mV which is a regulation of 0.24%. The point at which the measurements were taken for the 5 V regulator was directly at the output terminals which means that there is no voltage drop to take account of. This set of results shows that the 5 V regulator is working as intended and will be able to provide 5 V to all the components that require it in the power board.

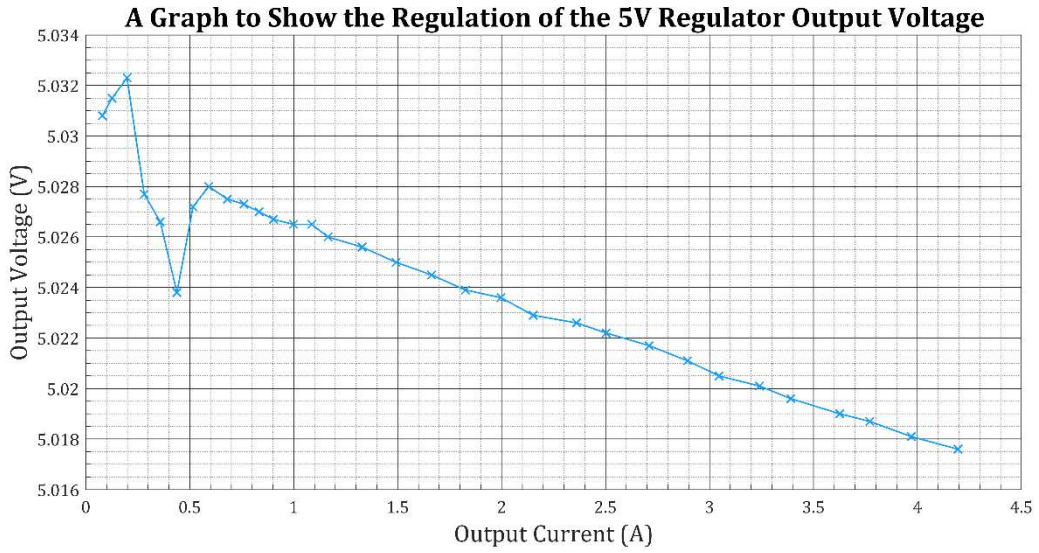


Figure 22: A Graph Showing the Regulation of the 5 V Regulator Output Voltage

Figure 23 shows the regulation of the 12 V regulator output voltage, the immediate difference between this data set and the previous dataset is the sudden rise at 0.25 A and the continuous rise up until 1.7 A. This suggest the compensation network calculated was slightly incorrect and is over compensating the output voltage because of this. Due to the scale of the Y-axis the voltage change would at first appearances look to be quite a large change, it is however only a rise of around 400 mV, with the voltage settling back down again to just over 12 V at 4.3A. For these measurements, the test point wasn't directly at the output of the regulator, so the trace resistance causes a voltage drop. The resistance of the trace is 0.13 Ω which at 4 A causes a voltage drop of 0.52 V, this however isn't sufficient enough to have brought the 12 V rail to below the required 12 V. Further testing can be done to calculate the correct compensation components required and correct the values.

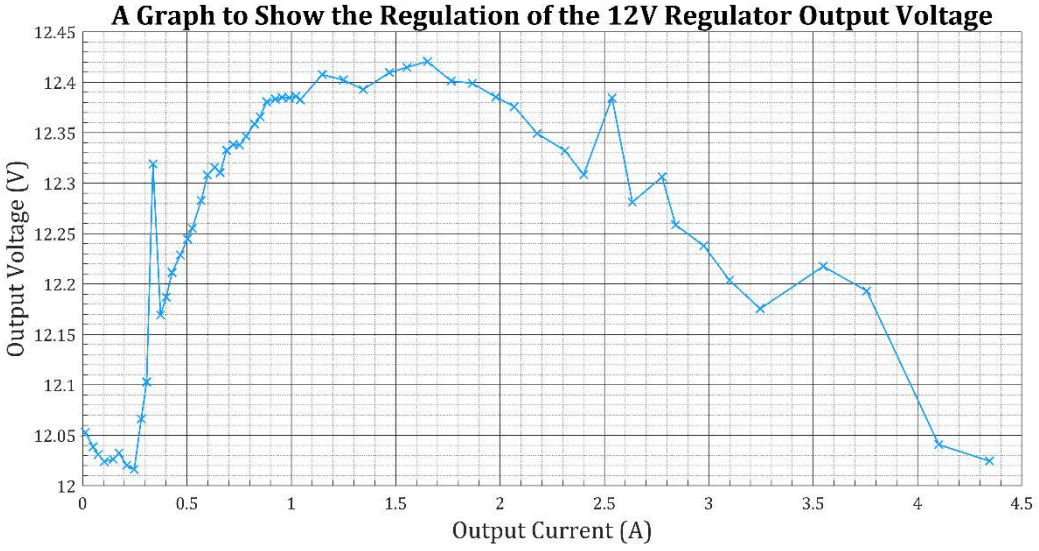


Figure 23: A Graph Showing the Regulation of the 12 V Regulator Output Voltage

4.3.2 Efficiency

In order to try and reduce the thermal effects within the capsule, the regulators need to be as efficient as possible to produce as little heat as possible. Figure 24 shows the efficiency of the 5 V regulator through a load sweep, which allows for simulation of different current demands. It can be seen that at 4.2 A, the regulator has a maximum efficiency of 89%. It starts at low currents with an efficiency of 64% and increases constantly. In the operating region of the regulator the efficiency will be well over 80% which is higher than expected. However the loss in efficiency will introduce heat into the capsule and will need managing and monitoring throughout the competition.

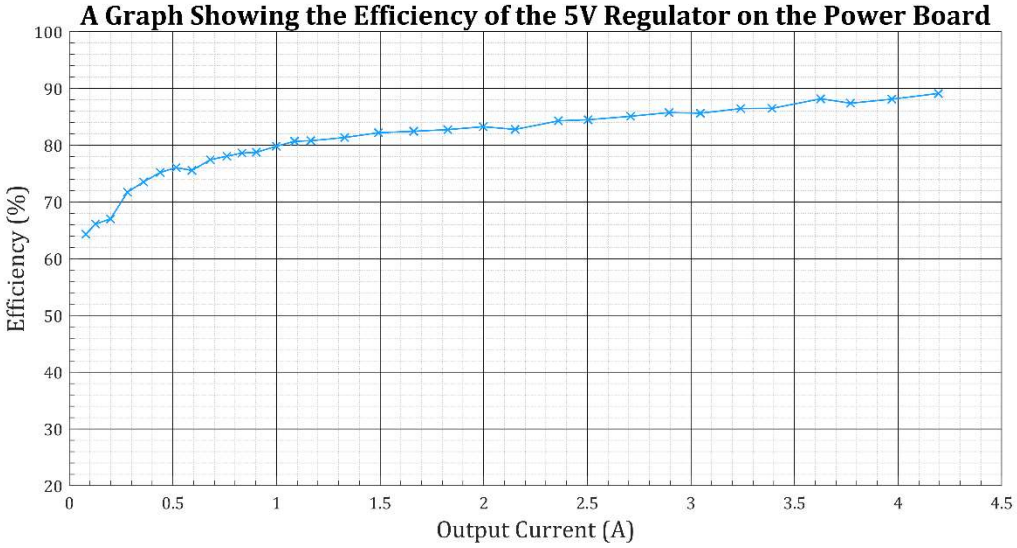


Figure 24: A Graph Showing the Efficiency of the 5 V Regulator on the Power Board

Figure 25 shows the efficiency of the 12 V regulator and it can be seen that at low current levels the efficiency is poor, being 60% efficient 0.1 A. This low efficiency could be due to the initial voltage rise caused by the incorrect compensation network as the efficiency very quickly normalises to an efficiency of greater than 80%. The maximum efficiency witnessed is 95% at 4.35 A. This is the most likely operation region for both 12 V regulators, rather than running in the lower current region.

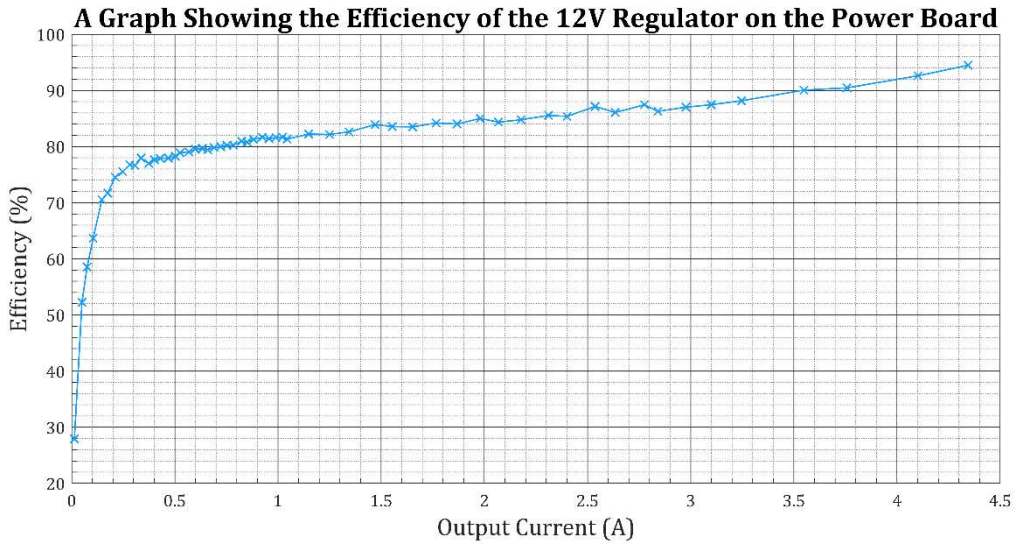


Figure 25: A Graph Showing the Efficiency of the 12 V Regulator on the Power Board

4.3.3 Thermal Results

With the power board being situated inside a sealed capsule the thermals were an important factor of the design process, attempting to keep the temperature rise of the board as low as possible. Figure 26 shows a thermal image of the 5 V regulator when running at 4 A. under these conditions the regulator operates at about 88% efficiency as can be seen on Figure 24, this means that 12% of the power is being converted into heat and heated the capsule. Over an extended period of time it can be seen that both the diode and IC get up to a temperature of over 100°C while the inductor manages to sit at a much cooler, but still warm 70°C. This however assumes the regulator running at a constant 4 A which in the competition is an unlikely occurrence.

Figure 27 shows the 12 V regulator also running at 4 A, at this operating point the regulator is 92% efficient as shown in Figure 25. However when you look at the thermal image, it can be seen that the diode and controller are operating at 140°C while the inductor sits at around 100°C. This is a considerably high temperature for the enclosed capsule and would cause the control enclosure to heat up very quickly. The temperature would need to be closely monitored as the sensitive electronics on the data board might not be able to handle such high temperatures.

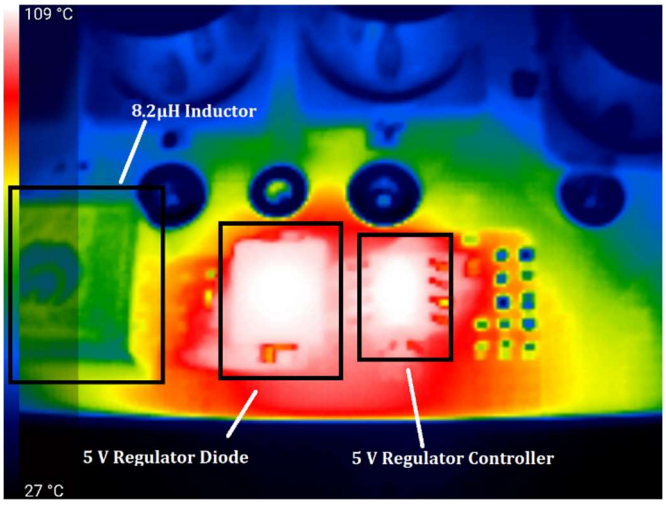


Figure 26: Thermal Image Showing The 5 V Regulator running at 4 A

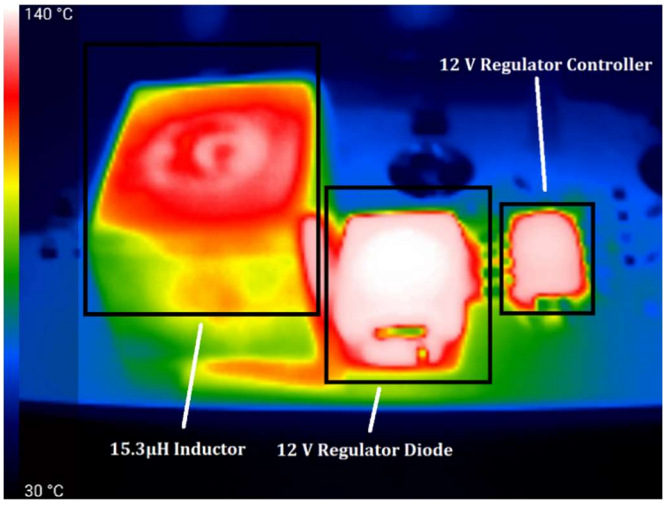


Figure 27: Thermal Image Showing The 12 V Regulator running at 4 A

4.4 Evaluation

Comparing the final board through testing and the initial specification, the power board meets all the specification points which were set out. Some points are met better than others, however. The connectors which interface the ESCs with the power board work better than expected, with easy access and easy removal of the modular ESCs when required. They can easily handle the current requirements and are capable of handling the data that the ESC requires.

Both regulators deliver the required power to the system, however they do operate at a higher temperature than would be preferred. Further tests could be carried out to see what the thermal impact is on the inside of the capsule when running a simulated competition run. It would also be helpful to see the thermal images of each regulator when running at various current demands to see how much the temperature differs between operating points.

With more time the components used in the compensation network could be changed to try and improve the regulation of the output voltage to give a constant output of as close to 12 V as possible. This could be as simple as changing a resistor value but could also require additional components to compensate for the abnormalities witnessed in testing. This could also help with the thermal issue but cannot be guaranteed until further testing is done to prove it.

The thermal tests show that the boards operate at a lower temperature than set out in the specification, however the test were done in open air conditions were passive cooling occurred from air flow within the room. To get a better overview of the thermals of the board, a full test would be carried out with a fully assembled control enclosure that can be run for a simulation period under water to see how the thermals differ to open air conditions.

The h-bridges were proved to operate successfully during the manufacture process, but unfortunately no test data was able to be collected due to the effects of the COVID-19 lockdown. they were proven to work with both motors and solenoid valves when set up in a full bridge or half bridge configuration allowing maximum customisation of the external actuators. Overall the power board has met all the specification point but could be improved through further testing and improvements.

5. Electronic Speed Controllers *(HO)*

5.1 Specification

5.1.1 Original Specification

→ Powered from +48 V and +5 V rail.
→ On-board 5 v -3.3 V regulator to power logic.
→ Drives a single brushless DC (BLDC) thruster at a maximum power of 200W.
→ Modular design - easily replaceable in the event of failure.

5.1.2 Final Specification

→ Four different versions of the ESCs for comparison to find the most effective were designed:

<table>
<tr><th>GaN MOSFET Motor Driver (primary)</th><th>Silicon MOSFET Motor Driver (primary)</th><th>Commercial ESC Motor Driver (backup)</th><th>Motor Driver with Integrated Driver and Controller (backup)</th></tr>
<tr><td colspan="4">→ Powered only from the 48 V rail, with optional 5 V input for backup.
→ Control via RS-485 (multiple units on one bus), analogue speed input, I2C or UART
→ Form factor of 54.6 x 32 mm with an edge connector for all inputs/outputs
→ Maximum of 4-layers, 35 µm thick copper PCB
→ Four power-level indication LEDs</td></tr>
<tr><td colspan="2">→ A dsPIC33CH512MP505 dual-core 16-bit microcontroller for control
→ Three INA240 Current sense amplifiers for current feedback
→ Permanent Magnet Synchronous Machine Field Oriented Control System (PMSM FOC)</td><td colspan="2">→ An ATMega4809 to interface with either the Commercial ESC or integrated driver
→ 48 to 12 V DC/DC converter to supply either the Commercial ESC or integrated driver.</td></tr>
<tr><td>→ Three LMG5200 Gallium Nitride Half-Bridge Modules for the output stage
→ Capable of driving one thruster at 40 W</td><td>→ Three LM25101 MOSFET Gate Driver ICs
→ Three different types (for comparison) of 6 SO-8 packaged MOSFETs for the output stage
→ Capable of driving one thruster at 80 W</td><td>→ Footprint for a 20 A Off-the-shelf Electronic Speed Controller
→ Capable of driving one thruster at 35 W</td><td>→ A Texas Instruments DRV10987 2 A integrated motor driver and controller
→ Capable of driving one thruster at 20 W</td></tr>
</table>

Table 7: A Table Outlining the Final Specification for Each Model of ESC

5.2 Design

5.2.1 Hardware

To meet the original specifications, it was decided to design custom Electronic Speed Controllers (ESCs) to reduce the power electronics required to drive the thrusters, increasing efficiency; the custom ESCs would use a single 3-phase bridge directly from a 48 V input, rather than a commercial ESC driven from a 12 V rail post buck regulation.

When researching hardware that could implement the required control as a temporary development platform, the Texas Instruments (TI) BOOSTXL-3PhGaNInv Evaluation Module was found. This utilises three LMG5200 Gallium Nitride (GaN) half bridge modules each capable of delivering 10 A, switching at several MHz at up to 80 V, with integrated gate drive [48]. As the small form-factor of the LMG5200 [49] would be particularly suitable for the small space requirements, an Evaluation Module was purchased and microcontroller board (TI Launchpad F28027) was used to provide control. This was tested using example code from TI [50], a 3 V PSU and several different motors, including the thrusters that would be used on the ROV.

After the basic principle had been tested, the hardware was redesigned to meet the specification; the microcontroller was changed to the dsPIC33CH512MP505 due to the availability for sample, the auxiliary DC/DC converter was changed for one that could fit into the space requirements, and the MAX3495 was added to provide the required communication protocol [7]. This microcontroller has two cores, a master and a slave. The master runs at 200 MHz (generated from a PLL based on an external 8 MHz clock), whilst the slave runs at 180 MHz as most instructions can be executed in two clock-cycles, this results in a combined maximum speed of 190 MIPS (Million Instructions Per Second) [51].

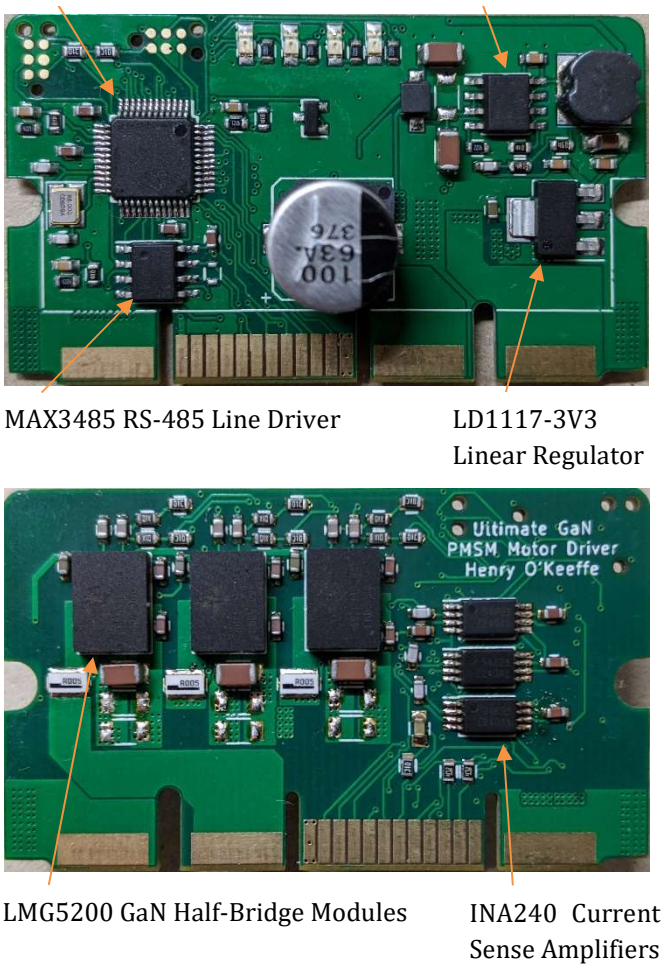


Figure 28: GaN Electronic Speed Controller PCB

Figure 28 shows the GaN ESC with key components marked.

The Maxim Integrated MAX3485 is the 3.3 V version of the common MAX485. It takes UART transmit and receive, receiver enable, and transmitter disable. The output of the transmitter and input of the receiver are connected internally, and externally connected to the differential RS-485 bus. At idle, the transmitter is disabled, and receiver enabled, allowing the receiver to convert the data on the bus into a single ended UART signal. The microcontroller synchronises using an all-zero byte, then checks for the address of the ESC (set by the host with a permanent binary high/low on the female connector). If the microcontroller recognises the address, it records the data sent to it. When the pre-determined amount of data (including a checksum) has been received, it disables the receiver, enables the transmitter, and replies with either requested data in the case of a request, or a repetition of the received data in the case of a command. To finish the transaction, the microcontroller sends another checksum (addition of all data transmitted modulo 255 + 1). If the checksum does not match the internally generated checksum, the microcontroller will ignore the data received.

The TI LM5164 buck regulator controller IC [52] is used to provide 5 V to the gate-drive portions of the LMG5200 Half-Bridge Modules. It chops the 48 V input at a duty cycle of approximately 10.4 %, with feedback from the output to control the duty. The pulsed DC at the output node is filtered to reduce the AC component with a 33 µH inductor and 10 µF capacitor. About 100 mA is expected to be drawn at 5 V, so the requirements for the regulator were not very high. Never-the-less, a safety factor of 5 was used, and the regulator should be capable of 0.5 A.

The STMicroelectronics LD1117-3V3 linear regulator [53] reduces the voltage from the 5 V rail to 3.3 V to power the MAX3485, the microcontroller and current-sense amplifiers. A linear regulator was chosen, as a stable supply and ease of implementation was preferred over a higher efficiency. The linear regulator is approximately 66 % efficient, whilst a switching one could be 90 % efficient. As the power demand on the 3.3 V rail is only approximately 120 mW, an additional 48 mW loss would be negligible during operation, causing a temperature rise from about 40°C to about 45°C (assuming the switching regulator has a similar thermal resistance of 135 K/W).

As the PMSM control system required current feedback, current sensors were required. The original TI evaluation module used the INA240 current sense amplifiers [54] and 5 mΩ current sense resistors. As these had proved themselves in operation and were of a suitable small size, these were chosen for this design, they provide an output referenced from 50 % of the supply voltage (3.3 V) and have a gain of 20 V/V. Coupled with a 5 mΩ resistor, this results in a trans-impedance of 0.1 A/V. A measurable current range of -16.5 to 16.5 V is therefore obtained.

Figure 29 shows the ESC designed with silicon MOSFETs with key components marked. The changes over the GaN version are the replacement of the GaN modules with three gate drivers and six silicon MOSFETs. As the gate drivers require 12 V to operate optimally, rather than 5 V, the LM5164 was reconfigured to output 12 V with the use of a different feedback and compensation network.

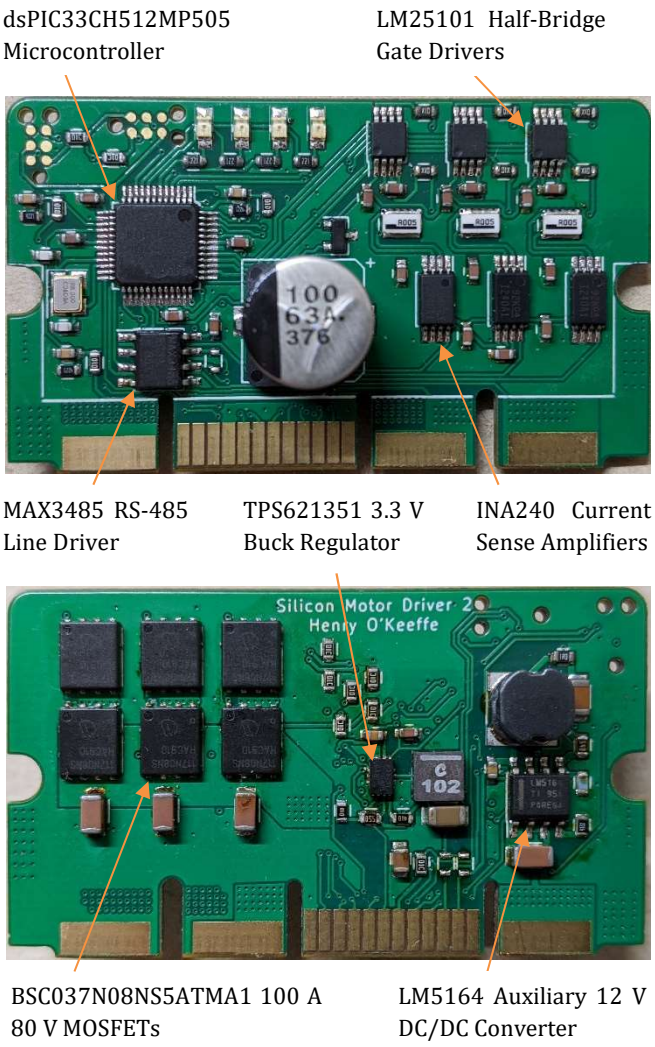


Figure 29: Silicon Electronic Speed Controller PCB

The requirement to replace the 5 V but with a 12 V bus would have drastically reduced efficiency in a 3.3 V linear regulator (to 27.5 %) This would increase the dissipation in the regulator to about 316 mW, which would cause a 40 degree temperature rise, and significantly impact efficiency at low demand. As this was undesirable, a switched-mode DC/DC regulator was chosen and designed to suit.

The TI TPS621351 was chosen for its small size, high current capability (4 A) and ease of design with a minimum of supporting components (feedback resistors, stabilisation network, 1 µH output inductor and smoothing and decoupling capacitors) [55]. Difficulty soldering reliably would mean that if more devices are required in the future, they could be designed with a different IC such as the SOIC-packaged TPS56628 [56].

Although a switching module (3-phase bridge, half-bridge, or single switch) with an integrated driver would be preferable to separate devices due to the potential space saving from the integration, no suitable device could be found. Modules such as the TI DRV8332 were considered, however their high cost, complexity and large $R_{ds\,(on)}$ (which would result in high conduction losses) [57] made them undesirable. Also considered was the TI DRV8301 [58] coupled with 6 discrete MOSFETs however after one PCB was designed with this module it proved difficult to set up and debug (over SPI), and so was abandoned.

The final solution was to combine three small TI LM25101 half-bridge gate drivers with 6 discrete MOSFETs. The gate drivers can provide 1 A peak, with a high side offset of up to 80 V and come in an appropriately small 8 lead MSOP (with power pad) package [59]. These should be capable of driving the MOSFET gates through the switching region in 16 ns. Losses and predicted temperature rise have not been calculated for the primary ESCs, as these depend on the control algorithm, and the ROV in operation is only manoeuvred slowly, resulting in the ESC being rarely the limiting factor.

To provide a backup in case other ESCs fail, either due to poor design, or failure of multiple ESCs, two other ESCs were designed. The Commercial ESC was designed with an on-board 48 to 12 V DC synchronous buck converter, supposedly capable of a 10 A output current. The Commercial ESC can be seen in Figure 30.

The TI LM5116 [60] Controller IC was originally designed as a synchronous buck converter, controlling the main 12 V output to the commercial ESC (as neither the commercial ESC nor the thruster can withstand 48 V). This drives the two CSD19534Q5A MOSFETs [61] in a half-bridge, the output being filtered by the 4.7 µH inductor and 100 µF capacitor. The controller has a mode to disable the low side driver to emulate a non-synchronous driver for greater efficiency at low output current. During design, this feature was overlooked, and unintentionally enabled. This resulted in poor efficiency at high loads, so to mitigate this, the lower MOSFET was replaced with a Schottky diode of a similar package (not seen in figure).

The other main difference between the backup ESCs and the primary ESCs, is the microcontroller used. The ATmega4809 was chosen, as it is easy to program with Atmel Studio, has all of the required features, and could be sampled in bulk. Running with a 20 MHz internal clock, it can achieve 20 MIPS [62].

The fourth ESC to be designed (seen in Figure 31) used an TI DRV10987 for driving the thruster. This IC is set up and can be controlled over I²C, outputs up to 2 A continuously, and is designed for driving sensor less Brushless DC (BLDC) motors [63]. Whilst a 2 A output cannot meet the originally specified 200 W, it was deemed a suitable replacement if it could be proved reliable for an emergency situation.

The integrated motor driver also has an integrated buck converter, which provides a 5 V output at up to 100 mA. This was used to power the ATmega, as well as the MAX3485 via the linear regulator. This integration removed the need for a separate auxiliary power supply and allowed the PCB to only have components on one side.

The TI LMR16030 3 A buck converter IC [64] was used to provide the 12 V rail. This IC requires an external flyback diode, however during design, the pinout of the 10 A diode was reversed. To correct for this without a redesign, two 2 A smaller diodes were used instead.

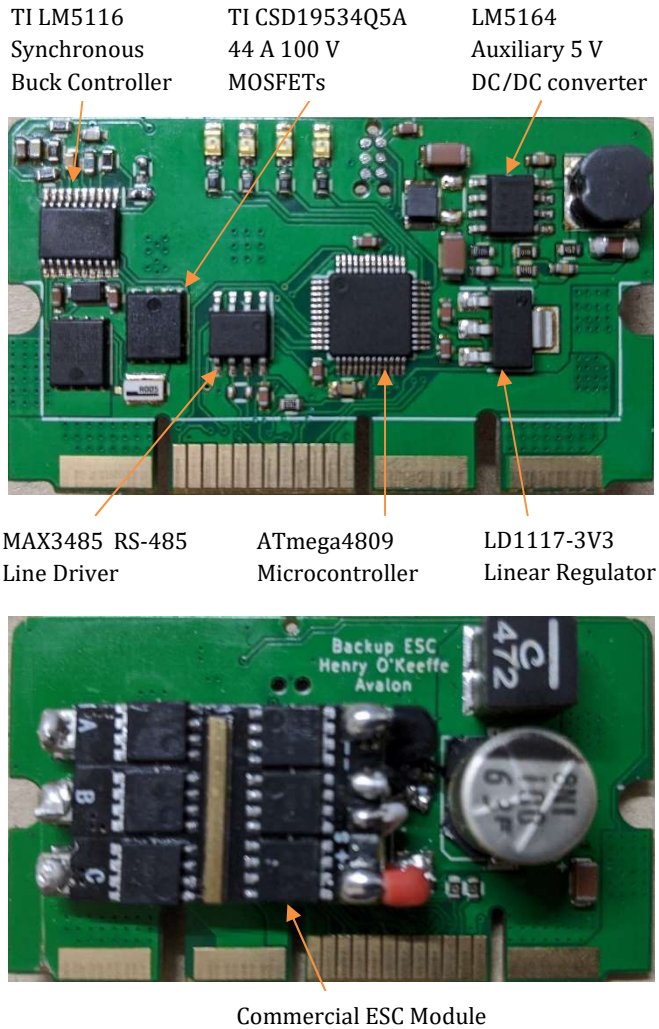


Figure 30: Commercial-ESC Electronic Speed Controller PCB

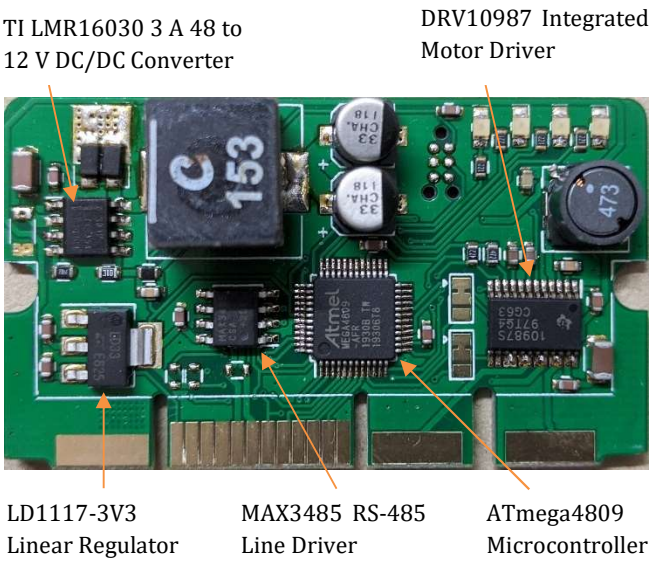


Figure 31: Integrated Driver Electronic Speed Controller PCB

On this design, the ATmega has the additional duty of setting up the driver IC. Required information such as motor resistance, open-loop current ramp time, etc, is stored in the program ROM of the ATmega and written to the EEPROM of the motor driver on start-up.

5.2.2 Software

To provide low acoustic and electrical noise, high efficiency and tight speed control, it was decided to use a Permanent Magnet Synchronous Machine control system on the primary ESCs, shown in Figure 32. By applying voltage to all three of the motor phases and sensing the current, it is possible to provide power to all three phases simultaneously, increasing available instantaneous power over the standard Brushless DC controller by 50% (with the same line current). Better utilisation of the windings also increases efficiency for the same output power. The simple implementation of a sinusoidal output voltage reduces the harmonic content of the current, provided a similar back-EMF, reducing acoustic noise [65].

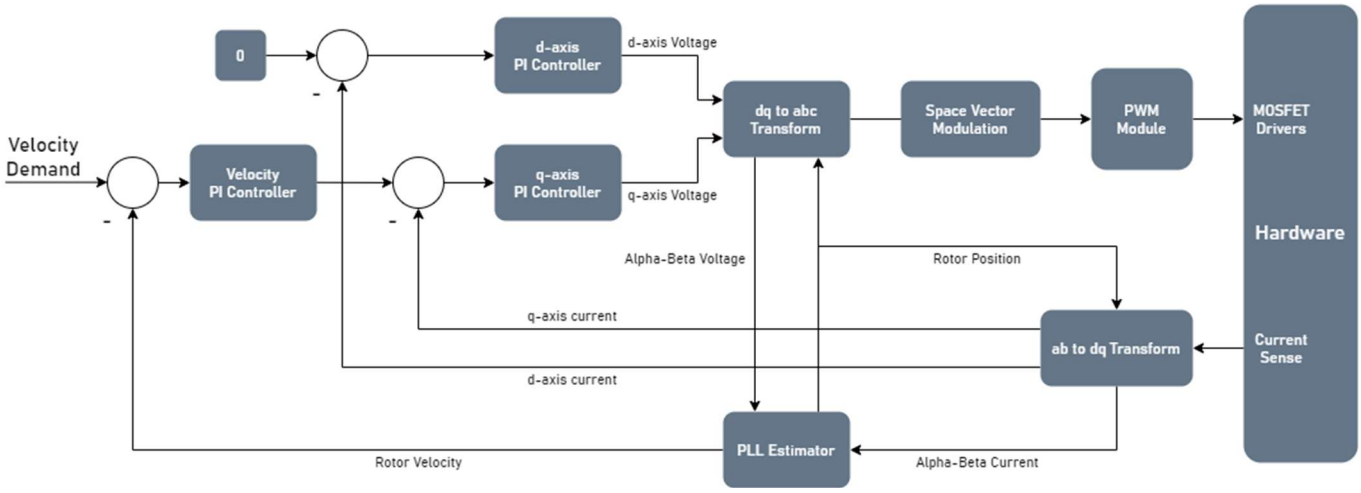


Figure 32: PMSM Field Oriented Control (FOC) System Diagram

Early in the project, it was decided to simulate the control system with Simulink, to determine the optimum PI constants for the three PI controllers. The thruster motors were tested and the required parameters (phase inductance, phase resistance, constant of angular velocity and inertia) were obtained. To optimise the PI constants, a Particle Swarm Optimiser (PSO) was written in MATLAB. This interfaced to a Simulink model, shown in Figure 33. The PSO initially generated a random PI value for each controller and ran the simulation multiple times with different random initial values. The time to achieve steady state (if possible) was recorded. The next set of simulations changed the PI values a random amount plus an amount that would bring the PI values towards (in n-dimensional space, where n is the number of controlled variables, in this case 6) both the simulation that had the lowest time to steady state in each of the previous runs with that initial value, as well as the best across all previously run simulations [66]. The simulations were repeated 20 times with 30 simulations per set, by which time the values had converged toward the optimum for the current set-up.

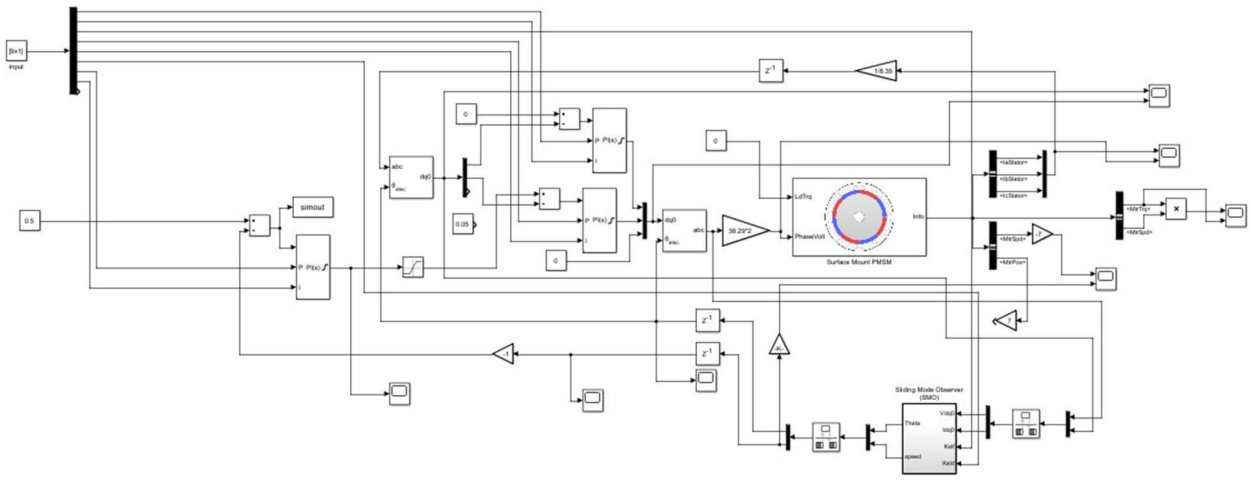


Figure 33: PSO Optimisation of the PMSM FOC System with Motor Model

After the TI Evaluation Module had been acquired and the test program set-up, the optimised PI values were tested on the thruster after accounting for the sample rate of the integrator. The thruster did not operate stably, and the PI values had to be adjusted by an order of magnitude to achieve stable operation. It was supposed that the poor choice of PI values was due to too many variables and discrepancies between the thrusters and the model of the thrusters, as well as between the control system model and the version that was implemented on the microcontroller. The PSO system was abandoned, and all tuning was done manually.

After it was decided to move to a dsPIC33 for control, a new program had to be written. This was based on an example by Microchip [28], however it used floating point maths, and the dsPIC33 does not have a floating-point unit, so it was rewritten to use fixed point maths. Other libraries used were the Motor Control Library [67] also by Microchip, and contains main function blocks, such as transforms, modulation, etc, written with fixed point maths.

The PMSM system uses FOC but, as there are no position sensors on the thrusters, a PLL Estimator provides estimated position feedback to the FOC. This estimator is based on the d-axis current being zero due to the action of the d-axis current controller. The dq-axis back-EMF is calculated from known values of voltage and current and previously estimated speed and position. The sign of the q-axis back-EMF (to allow function when the motor is spinning in both direction) is multiplied with the d-axis back-EMF. As this would ideally be zero, this represents the error. The error is added to the q-axis back-EMF and using the (known) velocity constant of the motor, the speed is determined. Integrating the speed and feeding it back to the estimator determines position and closes the loop. The PLL Estimator was chosen as it was simple to implement and reliable, however the requirement for an accurate velocity constant limits interoperation with other motors without changing the control system.

Other responsibilities of the primary ESCs include the managing communications over RS-485, reading the temperature from a temperature sensor, and controlling the speed indicator LEDs. As the dsPIC33CH is split into two cores, the master core is responsible for the communications protocol, driving the LEDs and taking the temperature measurements, whilst the slave runs the FOC and has full control over the PWM module.

The master and slave communicate using a mailbox – a set of dedicated memory locations that both cores have either read or write access to (set up in registers). A flag system (implemented in hardware) delays read or write operations if one is already occurring, preventing a possible race condition. A clock-based system would not be suitable, as each core has an independent clock, and is not synchronised. Although not developed, the microcontroller has the potential to run a different algorithm on the slave, selectable on start-up by having the master program it with one of two (or more) programs (as the slave runs from RAM, and the program is initially stored in the master's ROM).

The microcontroller on the commercial-ESC backup version outputs a Pulse Position Modulated servo output signal to the commercial ESC which is also directly connected to the thruster output and on-board 12 V rail. The microcontroller only performs the RS-485 communication with the Data Board, reads any commanded speed, then sends it to the commercial ESC, using an interlinked set of timers. The servo signal requires a 1 – 2 ms pulse every 20 ms, which represents a full-scale full reverse to full forward; 1.5 ms is zero speed. One timer was set to trigger another every 20 ms using the ATmega Events System. The second timer period was adjusted from 1 to 2 ms with 16-bit resolution.

5.3 Testing and Results

During development of the ESCs they were tested with a less expensive T100 thruster without a load (in open air). Once the program and hardware were confirmed working, they were programmed for the T200 thruster, and a test was set up which can be seen in Figure 61. A tube was attached to a T200 thruster and placed in water with a height of 25 cm between the surface of the water and the output of the tube. The thruster would draw water up the tube and release it with a 25 cm head. This was the test setup used to simulate lifting a heavy object with the ROV, this is the High Load Test. A graph of efficiency vs output power (as speed demand was increased) is shown in Figure 34. At an output of 39 W the system had a flow rate of 2.5 L/s, resulting in a mechanical system output power of approximately 6.1 W.

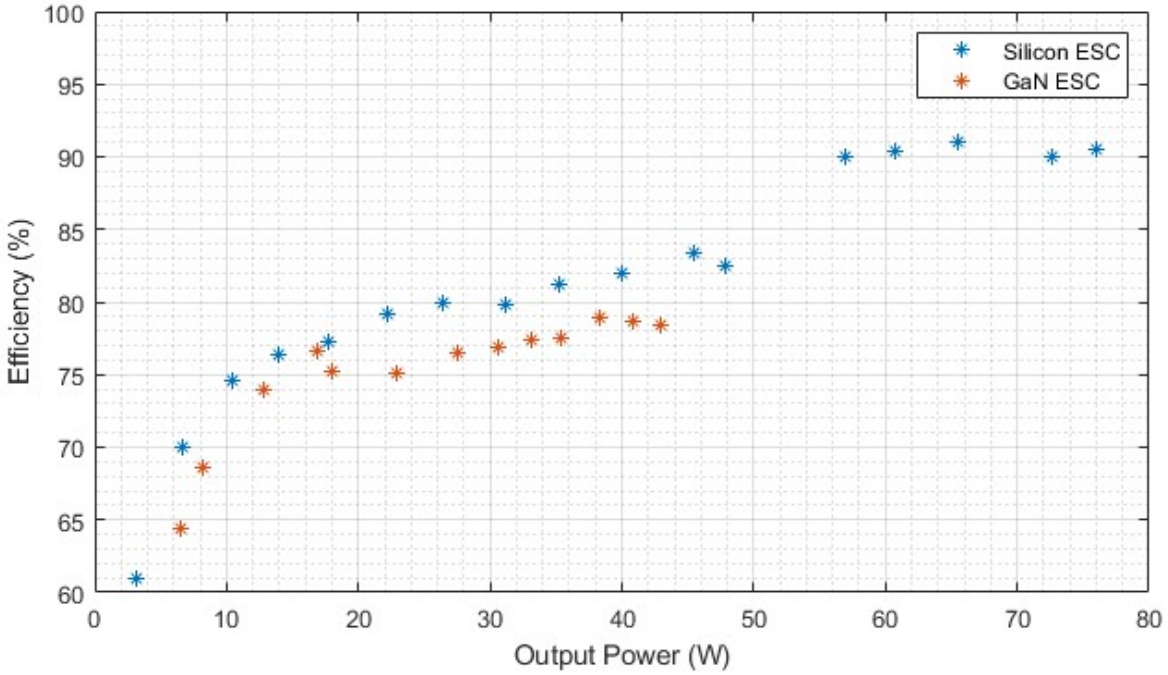


Figure 34: ESC Efficiency during the High Load Test

To determine likely performance when the ROV is moving laterally in the water, a Low Load Test was carried out. For this the pipe was made horizontal, so there was no drop, and the same test repeated. The results for this are shown in Figure 35.

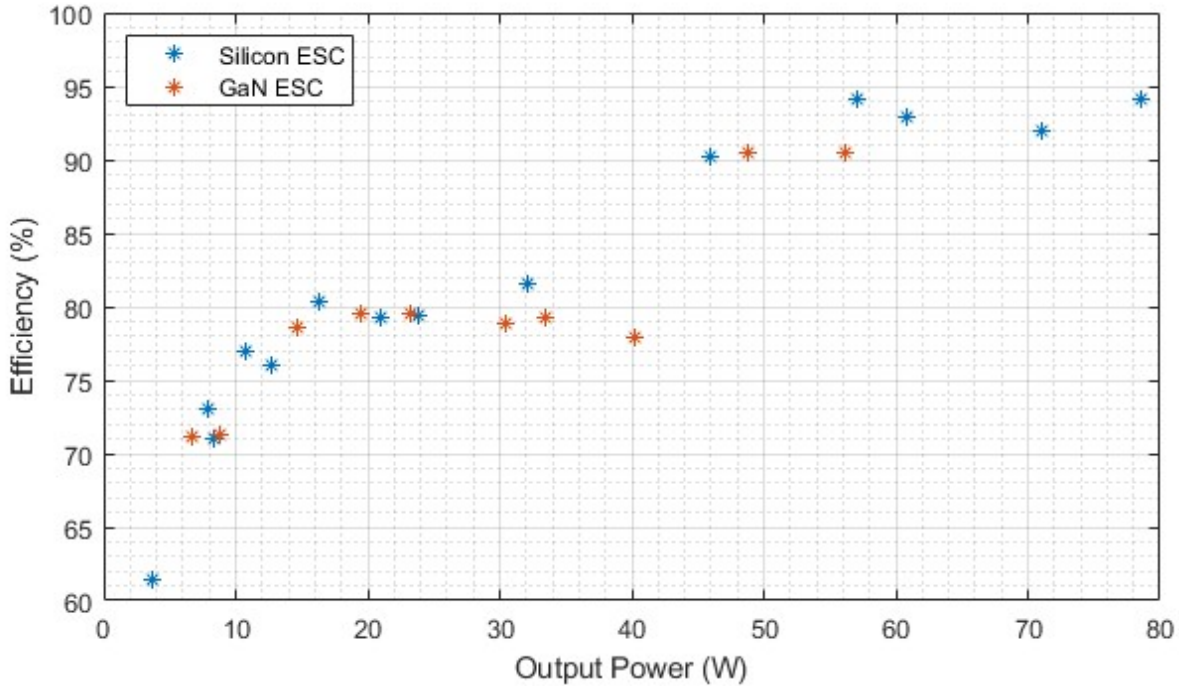


Figure 35: ESC Efficiency During the Low Load Test

Thermal tests were also carried out to assess the long-term effects of sustained operation at the common maximum output power (39 W, High-Load. limited by the GaN ESC). These thermal images can be seen in Figure 36 and Figure 37. Of particular note is the difference in temperature scale.

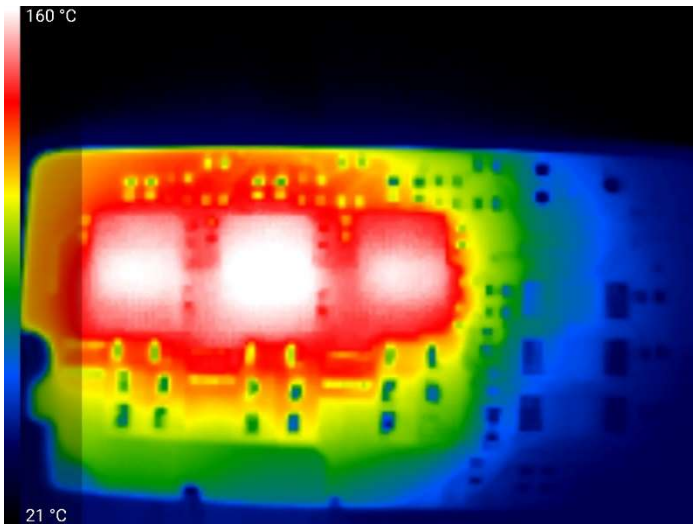


Figure 36: GaN ESC Thermal Image on 39 W High Load Test

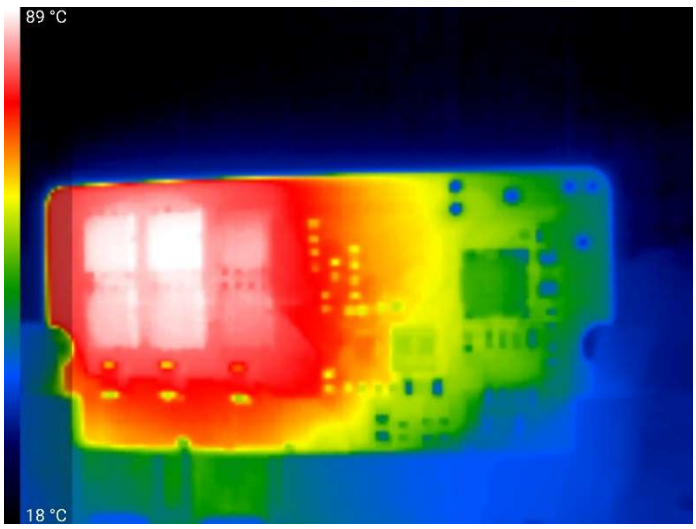


Figure 37: Silicon ESC Thermal Image on 39 W High Load Test

The backup ESC DC/DC converters were tested under a constant current load and thermal images were taken. Figure 38 shows the converter on the commercial-ESC board with a load of 3 A. Figure 39 shows the converter on the integrated ESC at 2.8 A (Over temperature protection shut down the converter beyond this current).

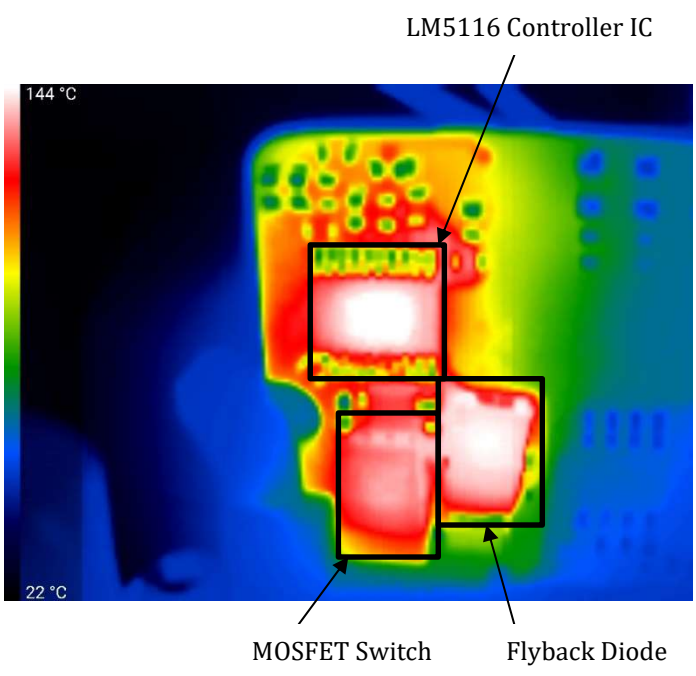


Figure 38: Commercial-ESC Converter Thermal Image at 3 A Load

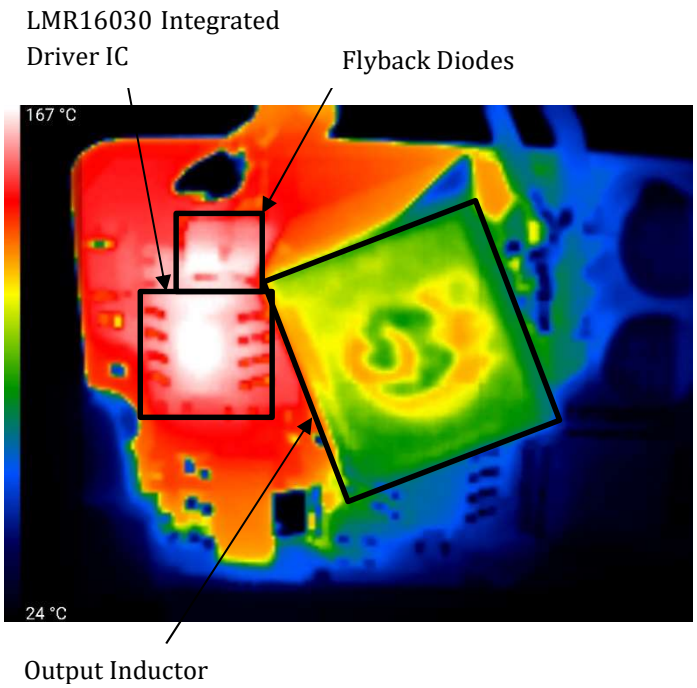


Figure 39: Integrated ESC Converter Thermal Image at 2.8 A Load

5.4 Evaluation

The High-Load test (Figure 34) shows a marked difference between the efficiencies of the GaN and Silicon ESCs. Up to the maximum power of the GaN ESC (43 W), the efficiency was 3-4 % less than the Silicon ESC. This was expected as at a switching frequency of 25 kHz even the total losses (mostly switching losses) of the Silicon ESC are low compared to the resistive losses of the GaN ESC. The Low-Load test (Figure 35) has almost equal efficiency for both ESCs at low power. As the current during the low load test was less than the high load test at the same power (voltage and speed were higher), this result is further evidence that the resistance of the GaN ESC was causing the lower efficiency in the high load test: whilst the Silicon ESC was largely unaffected by the reduced current, the higher resistance of the GaN ESC increased the effect of I^2R losses on efficiency.

The high spread of results in Figure 35 indicate that there was significant variation in the measurements. This is believed to be due to turbulence of the water during the test. Whilst the high load test will have had similar turbulence, the resultant change in current will have been negligible compared to the current required to lift the water.

In both the high and low load tests, a sudden jump occurs at around 50 W on the high load test and 40 W on the low load tests. As the jump happens at a lower power on the low load test, it seems that it happens at a specific speed demand, rather than a specific current or power. This is indicative of a flaw in the control system and may be improved (the efficiency increased at lower power) by adjusting control parameters. It is also interesting to note that the GaN ESC can attain an efficiency of 90 % comparable to the Silicon ESC when operating at low current load and a power greater than 48 W. It may be that the GaN will remain cool enough to be useful in the competition if the control system can be adjusted to obtain similar performance across the power range.

The thermal images in Figure 36 and Figure 37 show that the GaN ESC is not suitable for operation in the control capsule at a sustained 39 W, whilst the Silicon ESC might be, depending on how much heat is also being dissipated in the control capsule. A temperature of the outside of the package of 160°C is higher than the absolute maximum junction temperature of 125°C, and this test was done with a 20°C ambient temperature and in free space. Enclosure in the control capsule would rapidly result in device destruction. For the Silicon ESC, however, a temperature of 89°C is acceptable, and whilst it would rise in an enclosed space, the amount (and time taken to do so) cannot be determined without further tests. To provide a thermal mass to limit temperate rise during the competition, a small copper mass was added to each of the GaN ESCs. Whilst they would slow down temperature rise, their effectiveness is again subject to test (initial calculations indicate that a 38 g copper bar would heat by 100°C in less than 3 minutes if attached to a Silicon ESC operating at 40 W output. This is not including heat lost to the control capsule, which would raise the ambient temperature, but extend the time taken to overheat).

The thermal image in Figure 38 shows that the converter with the discrete switch cannot achieve the designed output current of 10 A. The package temperature of 144°C exceed the absolute maximum rating of the device at only 3 A output current. Whilst this was explained partially by the failure to account for switching losses in the power devices, the heat dissipated by the controller itself is surprising. This could be due to the losses in internal linear regulator (25 % efficient), but the only power this regulator has to supply is for the internal logic of the controller, and to drive the gate of the MOSFET. None of this power loss was accounted for either. Despite this, the DC/DC converter could still power a Commercial-ESC, albeit at a lower power.

Figure 39 shows the Integrated ESC converter. As the integrated ESC is only capable of outputting 2 A anyway, this regulator would have no problem suppling it in open air, however in the confined space of the control enclosure, it would most likely overheat in minutes. It is also worth noting that this is the only tested DC/DC converter that did not explode when the load was increased beyond its maximum capability. At 170°C it shut down safely, however this could not be relied upon in the competition. In addition, this converter tended to explode at inopportune moments (like when powering on, for instance) for seemingly no reason, so its use in the competition is limited.

Despite problems with certain components and control systems, several working, effective ESCs have been made and, with a little more tuning and redesign, could be instrumental in winning next year's competition.

6. Interface Boards *(BG)*

Alone, the data, power and ESC boards are not able to connect to anything external to the control capsule. The interface boards act as an intermediary interface between the electronics inside the capsule, and the underwater connectors mounted on the exterior of the capsule. The interface boards also allow signals to pass between the data and power boards. Each interface board is mounted via four fasteners to one of the control capsule end caps, with one end being used primarily for high power connections, and the other for high speed data connections. 98-pin through-hole PCI connectors are used to connect to the data and power PCB, which can handle 2.2A per pin [68].

A testing PCB for the ESC modules has also been developed, which is used for testing the ESC PCBs independently to the rest of the electronics control system.

6.1 Power Interface PCB

The power interface PCB, shown in Figure 40 and Figure 41 handles connections to:

→ +48V Power Input
→ 8x 3-phase thrusters

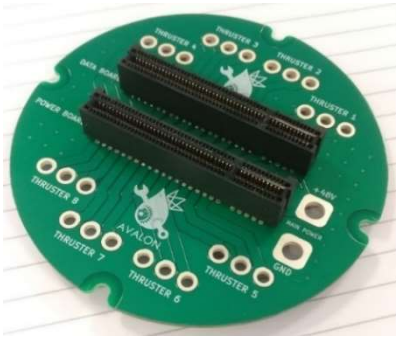


Figure 40: Power interface PCB front.

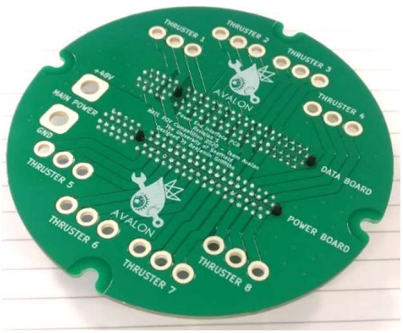


Figure 41: Power interface PCB back.

The current flowing through the traces on this PCB are relatively high, with a continuous current of 30A through the main power input, and 8A through each thruster phase. To accommodate these high currents, 2ozft^{-2} copper layers are used, and the trace width for each connection has been maximized (limited by the size of the board), with a 2.6mm trace width for each thruster phase. Large solid copper planes have been used for the main power input, with the distance between the connections being minimised where possible.

6.2 Data Interface PCB

The data interface PCB, shown in Figure 42 and Figure 43 handles connections to:

→ 8x Analogue cameras
→ 1x Ethernet camera
→ 2x USB cameras
→ 4x full bridge or 8x half bridge connections
→ Tether ethernet
→ 4x I^{2}C sensors
→ Fibre Optics
→ Mini-ROV

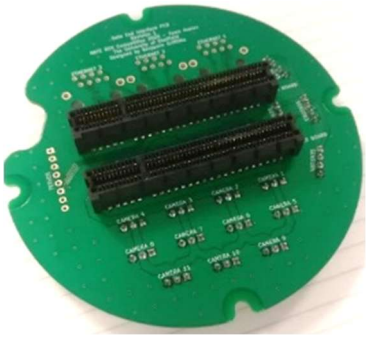


Figure 42: Data interface PCB front.

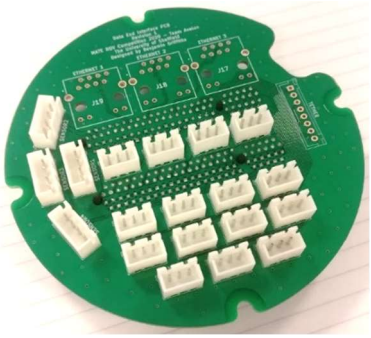


Figure 43: Data interface PCB back.

Due to the large number of connections on this PCB, JST-XH series connectors were used for the majority of signals such as analogue cameras and sensors. An RJ45 socket was used for the ethernet cameras. This makes assembly and maintenance of the data end-cap easier, which would be difficult if all the connections were solder terminated.

Some routing considerations had to be made for the traces belonging to high-speed USB, ethernet, and tether data signals. In particular, the length and skew of the differential pair traces were carefully tuned and were made as short as permitted by the board's layout.

The fully assembled power and data capsule end caps are shown in Figure 44 and Figure 45, which use MacArtney and Bulgin underwater connectors to interface with external devices.

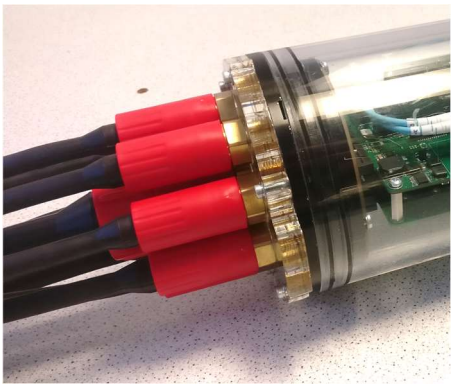


Figure 44: Power end-cap assembly.

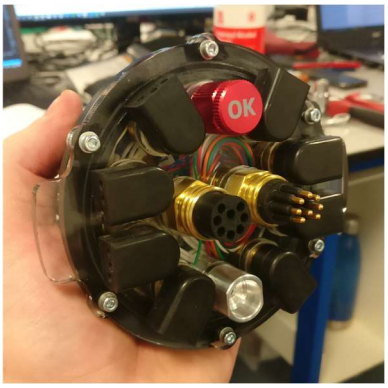


Figure 45: Data end-cap assembly.

6.3 ESC Test PCB

The ESC testing PCB was designed to isolate the development of the ESCs from the rest of the electronics control system. This was done to eliminate any potential development bottlenecks as a result of the inability to test the ESCs functionality until the rest of the system was complete. Furthermore, it prevented any damage to the other PCBs during initial testing.

The PCB contains a TE-Connectivity Mini Crown Edge connector to mount an ESC module to, and is powered by an ATmega 32u4 microcontroller, which has an embedded USB interface for easy code uploading and to display data on a serial terminal [69]. A Phoenix connector is used for the main +48V input, and a generic terminal block is used to connect any 3-phase brushless motor. The microcontroller communicates with the ESC module via RS485, using a MAX3485 line driver. A range of switches, buttons and a potentiometer are included to control the ESC, such as its speed, address and control schemes. Furthermore, LEDs and pin headers connected to key signals on the board are used for debugging. The assembled PCB is shown in Figure 46 and Figure 47.

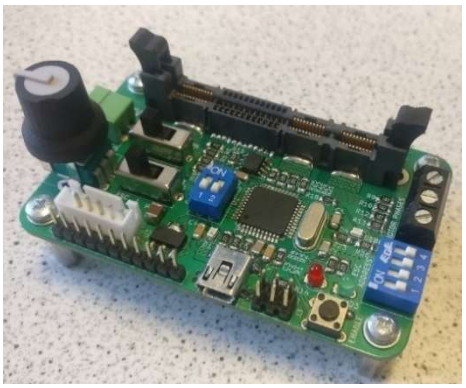


Figure 46: ESC testing PCB (without ESC).

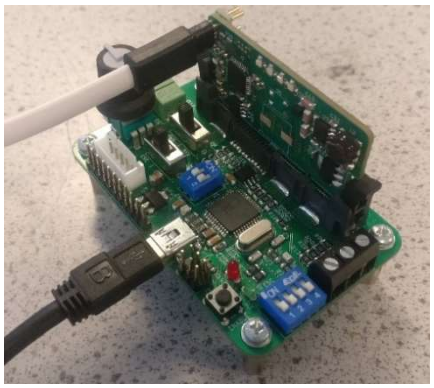


Figure 47: ESC testing PCB (with ESC).

7. Control Software *(BG)*

7.1 Specification

The software required to operate the ROV takes two forms, with the first being the software embedded in the electronics control systems microcontroller, and the second being a graphical user interface used by the pilot to control the ROV remotely. Both pieces of software work concurrently to fully control all the ROVs functions.

7.1.1 Graphical User Interface

The pilot uses the interface to control all aspects of the ROV remotely when it is submerged in water, which was run on a local computer on the surface and sends commands to the ROV over a tether. The graphical user interface was developed in Python, primarily using the PyQt5 library [70] - a Python binding of the popular Qt application framework [71]. Additional libraries such as Open-CV [72], PyGame [73], PySerial [74] and xml-ElementTree [75] as well as custom developed libraries were used to implement the necessary functionality, which will be discussed in detail.

The initial step in the development process was to establish all the functional requirements that the program must be able to perform. These requirements depend on the tasks the ROV has to complete at the competition as well as some usability aspects of the program.

Table 8 shows a breakdown of the functional requirements, categorized into the core components of the program.

Component	Functional Requirements
Serial Communication	→ Automatically detect the ROVs COM port → Connect to the COM port using a serial interface → Send/receive ASCII commands → Error handling
Pilot Controller	→ Read joystick and button values from an XBOX controller → Display the controller values for testing → Filter the joysticks to add a dead-band → Adjust the controller sensitivity
Usability	→ On-screen buttons to control ROV functions → Separate windows for controlling and configuring the ROV- → Self-explanatory user interface → Program must scale correctly for different display sizes → Saved user profiles for individual pilots and future ROV designs
Thrusters	→ Change the location of each thruster on the ROV → Reverse each thruster direction when required → Test individual thrusters → Perform thrust vectoring to achieve 6 axis motion control → Toggle control direction to pilot ROV in reverse
Actuators	→ Change number of actuators → Edit the name and on/off state label → Display current state of each actuator → Toggle each actuator using an on-screen button
Key-bindings	→ Add remappable XBOX controller key-bindings → Automatically detect button pressed to set binding → Use key-bindings to control ROVs actuators
Digital Cameras	→ Use OpenCV library to display USB and RTSP video streams → Enable/disable each camera feed → Display a large primary, and smaller secondary camera feeds → Click on a camera feed to move it to the primary feed → Change number of digital cameras → Select default camera feeds that are displayed at program launch → Change current camera feeds that are displayed

	→ Change source address of each camera → Change name of each camera
Machine Vision	→ Pop out window to display controls for each machine vision task → Pass camera frames through processing algorithms → Transfer data from algorithms to main program for processing → Easily implement external machine vision algorithms
Analogue Cameras	→ Change number of analogue cameras → Select default camera feeds that are displayed at program launch → Change current camera feeds that are displayed → Change name of each camera
Sensors	→ Change number of sensors → Select the type of each sensor → Retrieve sensor readings at a polling rate → Display the value of each sensor
Configuration	→ Saves all program settings to an XML configuration file → Opens XML file and applies program configuration upon launch → Program can be reset to default settings → Browse for specific configuration XML file

Table 8: ROV control program functional requirements.

7.1.2 Embedded Code

The code embedded in the electronics control systems microcontroller is written in C/C++ and is responsible for generating the signals to control the devices on the ROV. These include generating PWM signals for H-bridge modules, processing incoming data from the tether, and using communication protocols such as SPI and I2C to read data from sensors. The program was developed on the Arduino platform, which made preliminary testing possible on a standard Arduino Uno development board, independent of the data PCB.

Table 9 shows the functional requirements of the embedded program.

Component	Functional Requirements
Serial Communication	→ Respond to device identity request with “AVALONROV” → Parse incoming ASCII commands → Execute required functions
Thrusters	→ Output the arming sequence to activate the ESC → Set thruster speeds over RS485 protocol
Actuators	→ Output PWM signals to the power PCBs H-Bridges to control actuators and motors.
Cameras	→ Control camera switching matrix to select which analogue cameras are sent up the tether.
Sensors	→ Read multiple sensors on I2C bus → Transmit readings to control program

Table 9: Embedded code functional requirements.

7.2 Design Process

7.2.1 Graphical User Interface

7.2.1.1 Interface Design

The first step in the design process was to develop the user experience and design a prototype interface.

The user experience needed to be streamlined to allow the pilot to access the wide range of ROV controls unhindered, whilst offering a rich set of features with a high level of configurability. This was achieved by splitting the user interface between a *control panel* and *configuration* tab. The control panel contains all the essential controls the pilot requires to operate the ROV during the competition, such as live camera feeds, actuator controls and sensor readings. In contrast, the configuration tab is where the program is set up, such as setting the number of actuators, changing the communication settings or configuring the key-bindings for an individual user.

The *control panel* and *configuration* tab interfaces are shown in Figure 48 and Figure 49 respectively.

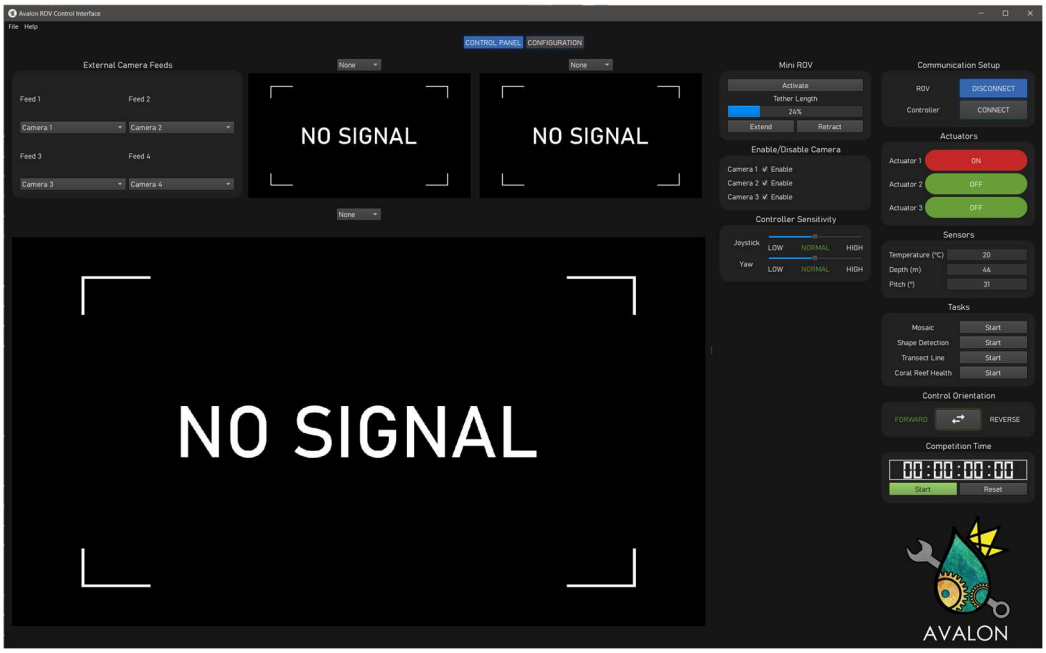


Figure 48: Programs control panel interface.

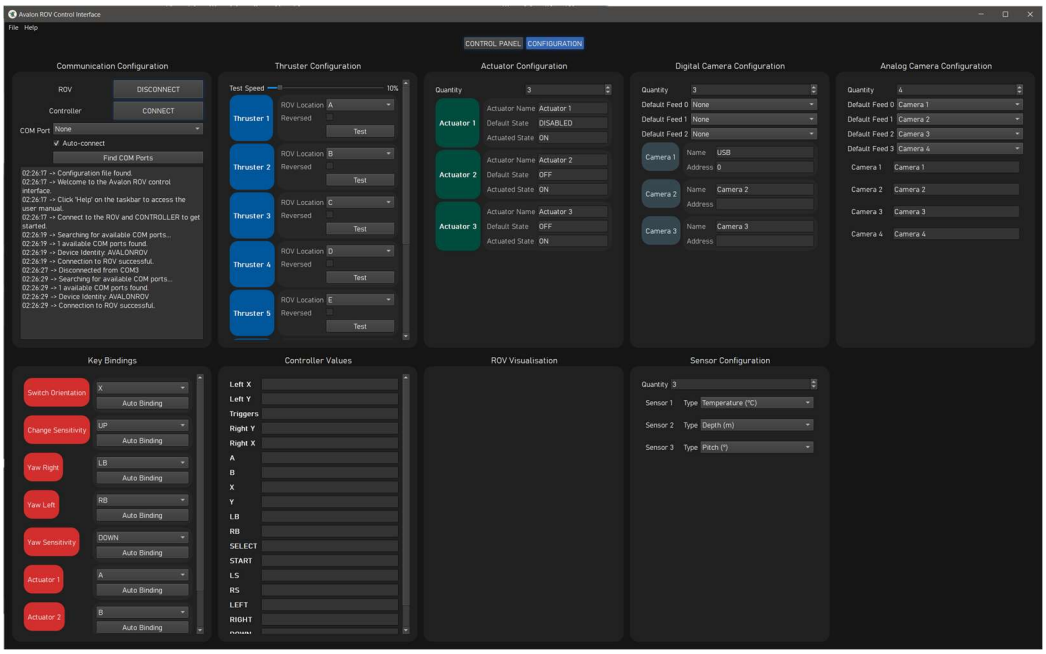


Figure 49: Programs configuration interface.

It took a few design iterations to achieve the final interface layout, but the end result is intuitive, with most controls being accessible with one click. The main component in the control panel is the live camera feeds, with one large primary feed, and two smaller secondary feeds. The pilot can select which camera source to display in each feed using the drop-down menus and can swap the primary feed with any secondary feed simply by clicking on the image. On the top left of the control panel, there are four drop-down menus used to select the analogue cameras to be sent up the tether from the data PCB.

The right-hand side of the control panel is populated with all the ROV controls, which are split into clearly labelled groups. For example, all the widgets relating to communications are contained by a group box labelled *Communication Setup*, and all the buttons to toggle the actuators are contained by a group box labelled *Actuators.*

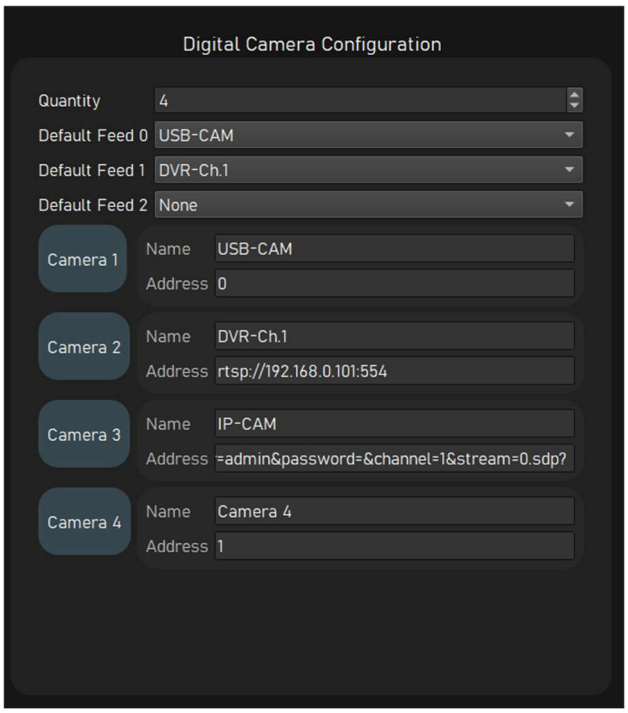


Figure 50: Digital cameras configuration tile.

The use of these group boxes or tiles makes the intent of the widgets obvious to the pilot. Each tile on the control panel has a matching tile on the configuration tab, which is where the data entry widgets are located to configure the program. For example, the actuators tile on the control panel, has a matching tile on the configuration tab, where the number of actuators can be set as well as their labels.

As an example, the digital camera configuration tile is shown in Figure 50, where the pilot can set the number of cameras on the ROV, the default feeds that are shown upon program launch, and the source address of each camera source.

Lastly, the taskbar on the top left contains features such as saving the programs current configuration, loading a different configuration file, changing the programs theme and opening the documentation.

7.2.1.2 Code Structure

To implement the large number of features required for the GUI, the structure of the code had to be organised and carefully planned. The program has to be stable, as the success of the ROVs performance at the competition depends on it, and it must be easily maintainable so that adding new features and tracking down bugs as the program is developed is easy.

During early development, it quickly became obvious that the code needed to be compartmentalised, allowing individual parts of the program to be tested independently. This was due to the complex interaction between different areas of the code, with multiple serial communication links, multiple threads [76] handling camera feeds and timers, and the dynamic nature of the interface because of its user configurability. It was often difficult to trace the root cause of bugs and program crashes, making further developments near impossible.

To alleviate this, a range of custom Python libraries were developed for the program to use. By containing the core functionality of the program in these libraries, or modules, the main Python script was greatly simplified, with its sole purpose being to import all the libraries, link them together and set up the program. In doing this, the structure and flow of the program is easily understood, and separate parts of the program can be easily tested and implemented.

The way that modules communicate with each other is through Qt's convention of *signals* and *slots* [77], where a signal represents the data you want to send, and the slot is where the data is sent to. For example, whenever an actuator is added to the program, a signal is emitted from the actuator library to the key-binding library (the slot), to create a key-binding to control that actuator.

A brief description of each library can be seen in Table 10.

Library	Description
Configuration File	Contains a class of functions to read and write the data to a configuration XML file. Uses the Python ElementTree library.
ROV Comms	Contains all functions relating to serial communication with the ROV, such as automatic COM port detection, and functions to format the commands to send to the ROV.
Controller	A thread running at 60Hz that reads joystick and button values from an XBOX controller. These values run through filtering functions before being emitted as a signal to the main program.
Camera Capture	A thread that captures frames from connected cameras, runs them through processing algorithms where required, and emits them as a signal. Other functions include changing the source address of the camera feed, and handling cameras connecting/disconnecting.
Visual Effects	Contains all the style sheets that determine the colours and styles of the widgets in the program, as well as functions to change the program theme.
Timer	A timer widget used by the pilot at the competition.
Thrusters	Contains widgets for the configuration tab, and the thrust vectoring algorithm.
Actuators	Contains widgets for the control panel and configuration tab, with functions to toggle each actuator.
Analogue Cameras	Contains widgets for the configuration tab, and functions to change camera settings.
Digital Cameras	Contains widgets for the configuration tab, and functions to change camera source addresses.
Key Bindings	Contains widgets for the configuration tab, and functions to automatically detect control button presses to set key-bindings.
Controller Display	Contains widgets for the configuration tab to visually indicate the values of the controller joysticks and buttons.
Sensors	Contains widgets for the control panel and configuration tab, with functions for filtering the incoming data.

Table 10: Custom developed libraries for the GUI.

A basic flow diagram that shows the interlinking of the libraries and threads is shown in Figure 51.

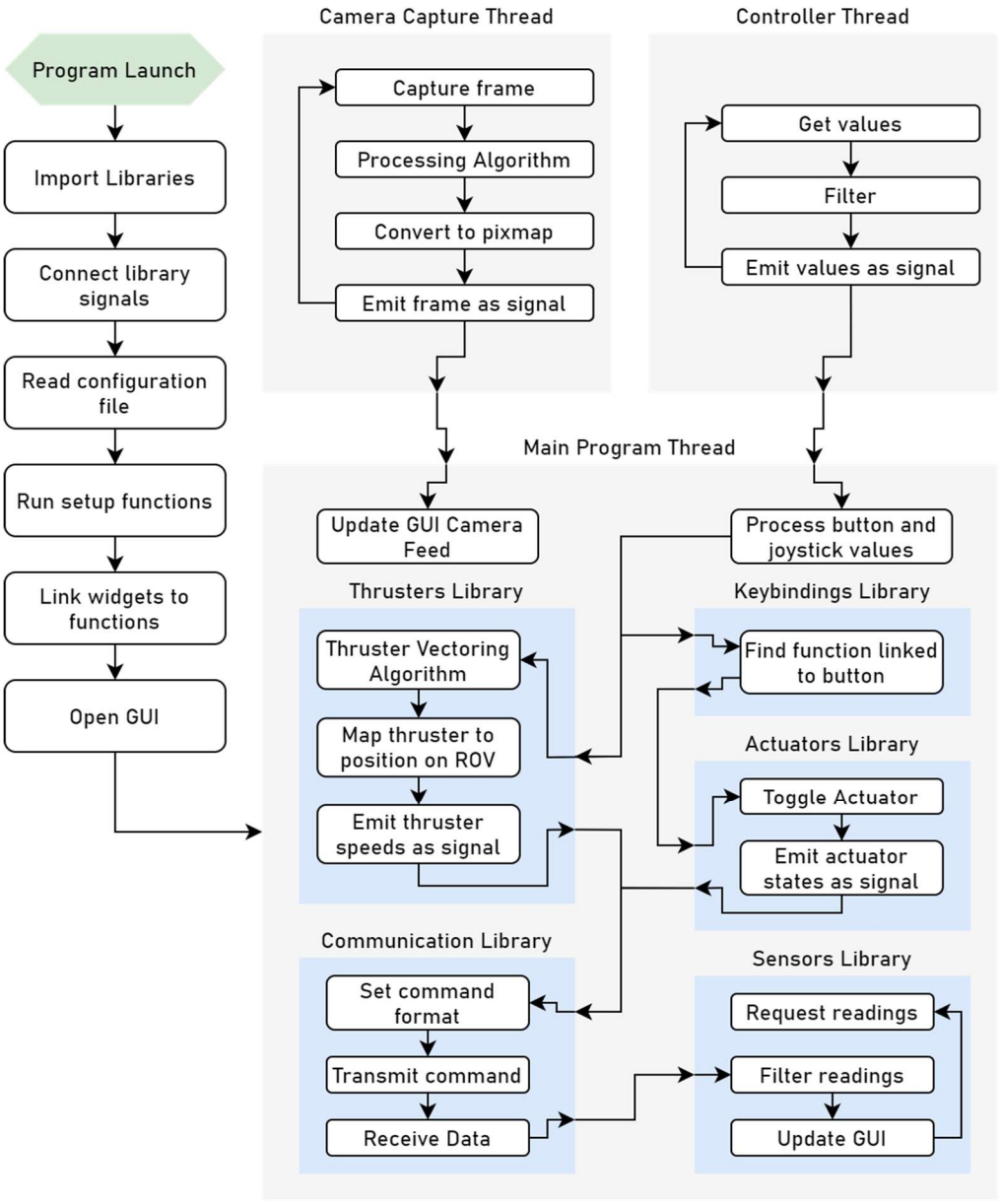


Figure 51: Basic flow diagram of GUI functionality.

7.2.2 Embedded Code

The functionality of the embedded program is centred around a custom communication protocol, which uses human-readable ASCII string commands to control the ROV. Therefore, the first step in developing this program was to establish a robust communication strategy. Each command consists of a) Command initialiser, b) Function identifier and c) Data where necessary, as shown in Figure 52.

Figure 52: ROV command structure.

All the commands required to control the ROV are documented in Appendix Table 2, with example commands shown in Appendix Table 3.

7.3 Results

The software was tested on an Arduino compatible platform, where all the required functionality could be simulated. For example, sensors readings were randomly generated, actuators were represented by LEDs, and the thrusters could be tested using commercially available ESCs that can accept a servo signal.

The automatic COM port detection functioned as designed, where even if there are multiple devices connected to the computer, the program can detect which one belongs to the ROV. This detection processing action can be seen in Figure 53. Furthermore, the sensor readings were able to be requested and received by the main program at a rate exceeding 100Hz, shown in Figure 54, even whilst heavily using the actuators and thrusters.

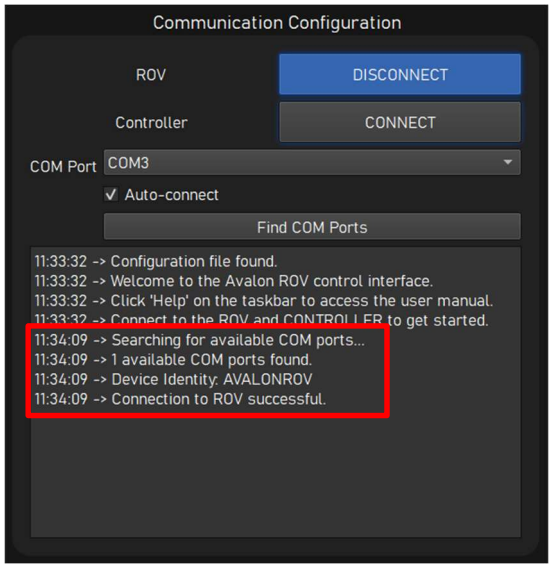


Figure 53: Automatic COM port detection.

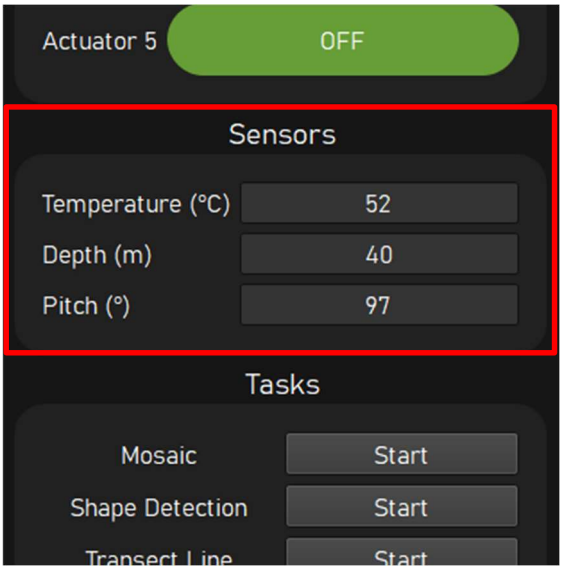


Figure 54: Displaying received sensor readings.

The XBOX controller thread and the key-bindings library worked as intended, with the pilot being able to bind any button on the controller to an ROV function, such as toggling an actuator, incrementing the controller sensitivity or controlling the ROVs yaw. Rather than selecting the desired button from a long list, the pilot can simply click the *Auto-bind* button, and the program will detect the first button that is pressed.

The performance of the thrust vectoring algorithm with the ROVs thruster configuration was tested using a test jig in a container filled with water, shown in Figure 55 and Figure 56, where the algorithm and controller bindings were fine tuned to achieve full 6-axis control.

Figure 55: ROV thruster test-jig.

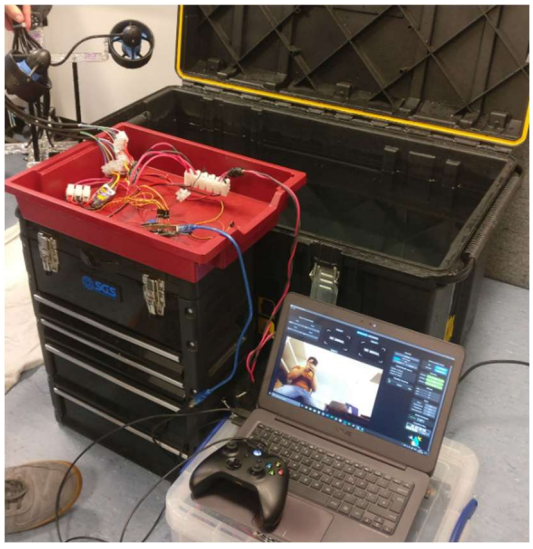


Figure 56: Testing thrust vectoring algorithm.

8. Additional Tasks *(BG)*

8.1 ROV Chassis

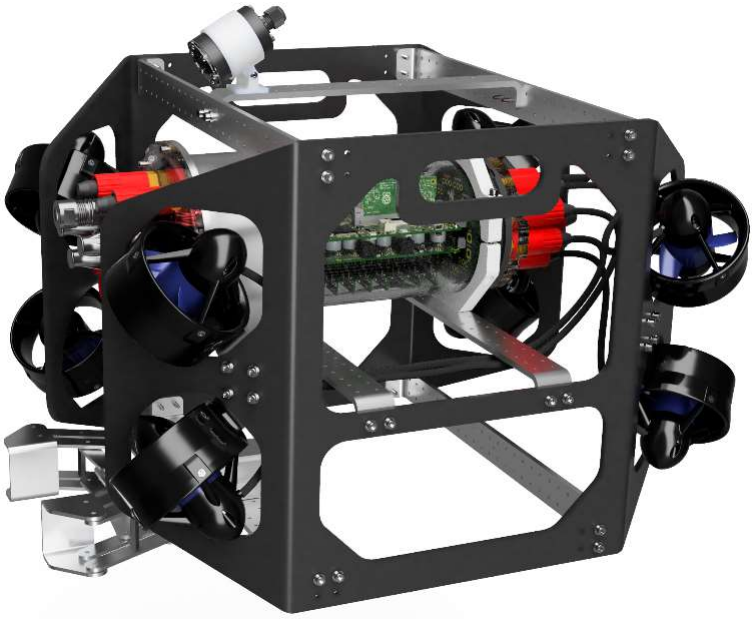


Figure 57: ROV chassis.

The prototype ROV chassis shown in Figure 57 was designed with two side panel made from Acetal to mount the thrusters, connected by an array of aluminium struts, that are used to mount the electronics control system and other devices.

This design was chosen to maximise the ROVs internal space to meet the competitions size constraints (largest dimension of the ROV must fit inside a 64cm diameter circle), so that we could mount all the devices such as sensors, cameras, and actuators, without interfering with the thrusters.

8.2 Surface Control Station

Figure 58: Surface control station.

The surface control station shown in Figure 58 was developed as an all-in-one, plug and play solution to piloting the ROV on location. It contains a custom computer to run the program on, a monitor to view the GUI, and a monitor to view the analogue camera feeds via a DVR. Panel mounted fibre, ethernet and power connectors are used to attach the ROVs tether, and an IEC power connector allows the whole system to be powered from 230V or 120V. A Bluetooth keyboard and an array of reconfigurable buttons, each with an LCD screen can be connected to the control program and bound to ROV functions. Under the surface panels, the custom computer, an internet router, the surface PCB and necessary power supplies are mounted. Everything is secured inside a heavy-duty Maxcase MAX-800 waterproof equipment case, making the surface station easily transportable to testing locations and the competition.

9. Project Achievements

9.1 Impact of the COVID-19 Pandemic *(BG/JO)*

The COVID-19 pandemic had a profound impact on the ability to complete the projects final tasks. In particular, restricted access to labs containing specialist testing equipment such as thermocouples and multi-channel data loggers meant that the thermal behaviour of the assembled electronics control system could not be fully investigated under a range of operating conditions. This included important characteristics such as the impedance of traces which requires signal generators and high bandwidth oscilloscopes.

Further to this, having a lab facility which was equipped with enough space and tools to better facilitate the manufacture of the electronics, would have meant more tests could have been completed. Having the ability to organise tests better would have meant tests could have been taken more efficiently and would have been more repeatable. Having more repeats on tests would give a better indication as to their accuracy, which could not be facilitated given the circumstances.

Furthermore, the temporary closure of manufacturing facilities such as the iForge makerspace meant that manufacturing processes such as laser cutters and waterjet cutters could not accessed to produce parts for the ROV chassis and surface control station. For example. whilst work on constructing the surface control station was started, shown in Appendix Figure 1 and Appendix Figure 2. the panels to mount the hardware could not be cut out. Meanwhile, raw materials were procured to manufacture the ROV chassis, such as sheets of Acetal plastic and aluminium bar stock shown in Appendix Figure 3, but were never able to be used.

Due to not having an ROV chassis to mount the electronics control system to, or access to a pool, the full system was never extensively tested under the conditions it was designed for. This means we have no empirical method for assessing if the design of our electronics met the thermal requirements. It is not possible for us to simulate what would happen by placing the tube in a pool, whether than be an increase or decrease due to the presence of warm pool water.

Lastly, the 2020 MATE ROV competition was unfortunately cancelled, meaning our system was not able to perform under competitive conditions this year.

9.2 Conclusion (*All*)

In conclusion, whilst every attempt has been made to make a working ROV, foundations have been set for a fully functional, fully featured competition ready ROV. Multiple design iterations of each PCB and continuous refinement of the control software ensure the key criteria within each individual specification was met. Whilst basic functionality testing of the PCB was carried out successfully, in depth investigations, such as thermal performance and full system testing were postponed due to the aforementioned issues. The team has every confidence that our system would have been able to compete successfully this year, had situations remained unaffected and hope that the work will be continued and progressed by current and future members of the team.

10. References

[1] MATE ROV Competition, "Explorer Competition Manual," MATE, 27 11 2019. [Online]. Available: http://files.materovcompetition.org/2020/2020_EXPLORER_Manual%208_FINAL_cover.pdf. [Accessed 05 2020].

[2] Blue Robotics, "4" Series Watertight Enclosures," Blue Robotics, [Online]. Available: https://bluerobotics.com/product-category/watertight-enclosures/4-series/. [Accessed 05 2020].

[3] Analog Devices, "AD8131," 2005. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/AD8131.pdf. [Accessed 17 May 2020].

[4] Analog Devices, "AD8130," 2005. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/AD8129_8130.pdf. [Accessed 17 May 2020].

[5] Texas Instruments, "LM2776 Switched Capacitor Inverter," February 2017. [Online]. Available: http://www.ti.com/lit/ds/symlink/lm2776.pdf?&ts=1589723386319. [Accessed 17 May 2020].

[6] Fairchild; On Semiconductor, "FMS6501," April 2007. [Online]. Available: https://www.onsemi.com/pub/Collateral/FMS6501-D.pdf. [Accessed 17 May 2020].

[7] Maxim Integrated, "MAX3483/MAX3485/MAX3486/MAX3488/MAX3490/MAX3491 - 3.3V-Powered, 10Mbps and Slew-Rate-Limited True RS-485/RS-422 Transceivers," May 2019. [Online]. Available: https://datasheets.maximintegrated.com/en/ds/MAX3483-MAX3491.pdf. [Accessed 17 May 2020].

[8] ST Microelectronics, "VNH7070AS Automotive fully integrated H-bridge motor driver," May 2016. [Online]. Available: https://www.st.com/resource/en/datasheet/vnh7070as.pdf. [Accessed 17 May 2020].

[9] FS, "Standard Fiber Patch Cables," v, 2019. [Online]. Available: https://img-en.fs.com/file/datasheet/standard-fiber-patch-cable-datasheet-3.pdf. [Accessed 17 May 2020].

[10] FS, "1000BASE-LX SFP 1310nm 10km DOM Transceiver," 2019. [Online]. Available: https://img-en.fs.com/file/datasheet/sfp1g-lx-31-10km.pdf. [Accessed 17 May 2020].

[11] Microchip, "SAM3X / SAM3A Series datasheet," 23 March 2015. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-11057-32-bit-Cortex-M3-Microcontroller-SAM3X-SAM3A_Datasheet.pdf. [Accessed 17 May 2020].

[12] PJRC, "RSA Signature Speed - Simple CPU Performance Benchmark," 5 August 2019. [Online]. Available: https://github.com/PaulStoffregen/RSA_signature_speed. [Accessed 17 May 2020].

[13] Arduino, "Arduino Mega 2560 REV 3," [Online]. Available: https://store.arduino.cc/arduino-mega-2560-rev3. [Accessed 17 May 2020].

[14] Raspberry Pi, "Raspberry Pi Compute Module 3+," 1 January 2019. [Online]. Available: https://www.raspberrypi.org/documentation/hardware/computemodule/datasheets/rpi_DATA_CM3plus_1p0.pdf. [Accessed 17 May 2020].

[15] Microchip, "LAN9514i USB 2.0 Hub and 10/100 Ethernet Controller," 2 November 2016. [Online]. Available: http://ww1.microchip.com/downloads/en/devicedoc/00002306a.pdf. [Accessed 17 May 2020].

[16] Microchip, "KSZ9477s 7-Port Gigabit Ethernet Switch with Ring Redundancy, SGMII and RGMII/MII/RMII Interfaces," 9 July 2019. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/KSZ9477S-Data-Sheet-DS00002392C.pdf. [Accessed 2020 May 2020].

[17] Samtec, "QSH High speed ground plane socket," [Online]. Available: http://suddendocs.samtec.com/catalog_english/qsh.pdf. [Accessed 17 May 2020].

[18] Samtec, "SFP CAGES & CAGE/CONNECTOR KITS," [Online]. Available: http://suddendocs.samtec.com/catalog_english/sfpc.pdf. [Accessed 17 May 2020].

[19] Microchip, "ENC28J60 Stand-Alone Ethernet Controller with SPI Interface," November 2012. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/39662e.pdf. [Accessed 17 May 2020].

[20] Microchip, "KSZ8563R 3-Port 10/100 Ethernet Switch with RGMII/MII/RMII Interface and IEEE 1588v2," 11 June 2019. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/KSZ8563R-Data-Sheet-DS00002418D.pdf. [Accessed 17 May 2020].

[21] Raspberry Pi, "Raspberry Pi Compute Module IO Board," 13 October 2016. [Online]. Available: https://www.raspberrypi.org/documentation/hardware/computemodule/schematics/rpi_SCH_CMIO_3p0.pdf. [Accessed 17 May 2020].

[22] Intel, "Interfacing Intel® 8255x Fast Ethernet Controllers without Magnetics," November 2005. [Online]. Available: https://www.intel.com/content/dam/doc/application-note/8255x-fast-ethernet-controllers-without-magnetics-appl-note.pdf. [Accessed 17 May 2020].

[23] Microsemi, "Application note: Magnetics guide," June 2018. [Online]. Available: http://ww1.microchip.com/downloads/en/AppNotes/VPPD-01740.pdf. [Accessed 17 May 2020].

[24] Texas Instruments, "PHYTER Transformerless Ethernet Operation," April 2013. [Online]. Available: http://www.ti.com/lit/an/snla088a/snla088a.pdf. [Accessed 17 May 2020].

[25] Sensirion, "SHT3x-ARP," September 2018. [Online]. Available: https://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/2_Humidity_Sensors/Datasheets/Sensirion_Humidity_Sensors_SHT3x_Datasheet_analog.pdf. [Accessed 17 May 2020].

[26] TDK - Invensense, "MPU - 6050," 19 August 2013. [Online]. Available: https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/. [Accessed 17 May 2020].

[27] TDK - Invensense, "ICM - 20948," 6 February 2017. [Online]. Available: https://invensense.tdk.com/wp-content/uploads/2016/06/DS-000189-ICM-20948-v1.3.pdf. [Accessed 17 May 2020].

[28] Microchip, "MCP3422/3/4 18-Bit, Multi-Channel ΔΣ Analog-to-Digital Converter with I2C Interface and On-Board Reference," August 2009. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/22088c.pdf. [Accessed 17 May 2020].

[29] Sensirion, "Overview of Sensirion's Gas Sensors," April 2018. [Online]. Available: https://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/9_Gas_Sensors/Sensirion_Gas_Sensors_Overview_Table.pdf. [Accessed 17 May 2020].

[30] TE connectivity, "MS5803-01BA," June 2017. [Online]. Available: https://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Data+Sheet%7FMS5803-01BA%7FB3%7Fpdf%7FEnglish%7FENG_DS_MS5803-01BA_B3.pdf%7FCAT-BLPS0038. [Accessed 17 May 2020].

[31] Texas Instruments, "TPS23881 Type-4 4-Pair 8-Channel PoE PSE Controller with SRAM and 200 mΩ RSENSE," October 2019. [Online]. Available: https://www.ti.com/lit/ds/symlink/tps23881.pdf?&ts=1589726478134. [Accessed 17 May 2020].

[32] Digi, "DIGI XBEE 3 ZIGBEE 3.0," 2019. [Online]. Available: https://www.digi.com/pdf/ds_xbee-3-zigbee-3.pdf. [Accessed 17 May 2020].

[33] Arduino, "Arduino Due," 2020. [Online]. Available: https://store.arduino.cc/arduino-due. [Accessed 17 May 2020].

[34] X. Che, I. Wells, G. Dickers, P. Kear and X. Gong, "Re-evaluation of RF electromagnetic communication in underwater sensor networks," *IEEE Communications Magazine,* vol. 48, no. 12, doi: 10.1109/MCOM.2010.5673085, pp. 143-151, 2010.

[35] Analog Devices, "Switching in USB Consumer Applications," January 2006. [Online]. Available: https://www.analog.com/en/analog-dialogue/articles/switching-in-usb-consumer-applications.html. [Accessed 17 May 2020].

[36] Analog Devices, "Design Considerations for Connecting Analog Devices Video Decoders to MIPI CSI-2," 2014. [Online]. Available: https://www.analog.com/media/en/technical-documentation/application-notes/AN-1337.pdf. [Accessed 17 May 2020].

[37] Texas Instruments, "High-Speed Layout Guidelines for Signal Conditioners and USB Hubs," August 2018. [Online]. Available: http://www.ti.com/lit/an/slla414/slla414.pdf?&ts=1589731847508. [Accessed 17 May 2020].

[38] Maxim Integrated, "SWITCHING VIDEO USING ANALOG SWITCHES," 23 June 2006. [Online]. Available: https://www.maximintegrated.com/en/design/technical-documents/tutorials/3/3823.html. [Accessed 17 May 2020].

[39] Totusoft, "LAN speed test," 18 February 2019. [Online]. Available: https://totusoft.com/lanspeed. [Accessed 17 May 2020].

[40] NETFLIX, "Fast.com," [Online]. Available: https://fast.com/. [Accessed 17 May 2020].

[41] T. Scott, "Why Snow and Confetti Ruin YouTube Video Quality," 23 May 2016. [Online]. Available: https://www.youtube.com/watch?v=r6Rp-uo6HmI. [Accessed 17 May 2020].

[42] Texas Instruments, "TPS56C230 4.5-V to 18-V, 12-A Synchronous Step-Down Converter," August 2019. [Online]. Available: http://www.ti.com/lit/ds/symlink/tps56c230.pdf?&ts=1589641913290. [Accessed 17 May 2020].

[43] Microchip, "MIC2128, 75V, Synchronous Buck Controller Featuring Adaptive," 2019. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/MIC2128-Data-Sheet-20005620E.pdf. [Accessed 17 May 2020].

[44] Texas Instruments, "4.5V to 18V Input 1.5A, 2.5A, 1.5A Triple Synchronous Step-Down Converter," December 2013. [Online]. Available: http://www.ti.com/lit/ds/slvsc29b/slvsc29b.pdf?&ts=1589716254156. [Accessed 17 May 2020].

[45] Micrel, "MIC47100 1A High Speed Low VIN LDO," September 2008. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/mic47100.pdf. [Accessed 17 May 2020].

[46] Texas Instruments, "TPS54560B 4.5-V to 60-V Input, 5-A step-down dc/dc converter with Eco-mode™," January 2019. [Online]. Available: https://www.ti.com/lit/ds/symlink/tps54560b.pdf?&ts=1589640464324. [Accessed 17 May 2020].

[47] Diodes Incorporated, "B560C 5.0A SURFACE MOUNT SCHOTTKY BARRIER RECTIFIER," July 2015. [Online]. Available: https://www.diodes.com/assets/Datasheets/ds13012.pdf. [Accessed 17 May 2020].

[48] Texas Instruments, "BOOSTXL-3PhGaNInv Evaluation Module User Guide (Rev. A)," June 2018. [Online]. Available:

http://www.ti.com/lit/ug/sluubp1a/sluubp1a.pdf. [Accessed 17 May 2020].

[49] Texas Instruments, "LMG5200 80V, 10A GaN Half-Bridge Power Stage datasheet (Rev. E)," October 2018. [Online]. Available: http://www.ti.com/lit/ds/symlink/lmg5200.pdf. [Accessed 17 May 2020].

[50] Texas Instruments, "BOOSTXL-3PhGaNInv Software," 27 June 2017. [Online]. Available: http://www.ti.com/lit/zip/sluc631. [Accessed 17 May 2020].

[51] Microchip, "dsPIC33CH512MP508 Family Data Sheet," 2019. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/dsPIC33CH512MP508-Family-Data-Sheet-DS70005371D.pdf. [Accessed 17 May 2020].

[52] Texas Instruments, "LM5164 100-V Input, 1-A synchronous buck DC/DC converter with ultra-low IQ datasheet (Rev. A)," January 2019. [Online]. Available: http://www.ti.com/lit/ds/symlink/lm5164.pdf. [Accessed 17 May 2020].

[53] ST Microelectronics, "LD1117.fm," February 2020. [Online]. Available: https://www.st.com/resource/en/datasheet/ld1117.pdf. [Accessed 17 May 2020].

[54] Texas Instruments, "INA240 High- and Low-Side, Bidirectional, Zero-Drift, Current-Sense Amplifier With Enhanced PWM Rejection datasheet (Rev. B)," February 2018. [Online]. Available: http://www.ti.com/lit/ds/symlink/ina240.pdf. [Accessed 17 May 2020].

[55] Texas Instruments, "TPS62135, TPS621351 High Accuracy 3-V to 17-V 4-A Step-Down Converters with DCS-Control datasheet (Rev. B)," April 2017. [Online]. Available: http://www.ti.com/lit/ds/symlink/tps62135.pdf. [Accessed 17 May 2020].

[56] Texas Instruments, "4.5V to 18V Input, Synchronous Step-Down Converter with Advanced Eco-mode™ datasheet (Rev. A)," December 2013. [Online]. Available: http://www.ti.com/lit/ds/symlink/tps56628.pdf. [Accessed 17 May 2020].

[57] Texas Instruments, "DRV83x2 Three-Phase PWM Motor Driver datasheet (Rev. E)," December 2014. [Online]. Available: https://www.ti.com/lit/ds/symlink/drv8332.pdf. [Accessed 17 May 2020].

[58] Texas Instruments, "DRV8301 Three-Phase Gate Driver With Dual Current Shunt Amplifiers and Buck Regulator datasheet (Rev. F)," January 2016. [Online]. Available: www.ti.com/lit/ds/symlink/drv8301.pdf. [Accessed 17 May 2020].

[59] Texas Instruments, "LM25101 3-A, 2-A, and 1-A 80-V Half-Bridge Gate Drivers datasheet (Rev. C)," September 2016. [Online]. Available: http://www.ti.com/lit/ds/symlink/lm25101.pdf. [Accessed 17 May 2020].

[60] Texas Instruments, "LM5116 Wide Range Synchronous Buck Controller datasheet (Rev. H)," July 2015. [Online]. Available: http://www.ti.com/lit/ds/symlink/lm5116.pdf. [Accessed 17 May 2020].

[61] Texas Instruments, "CSD19534Q5A 100 V N-Channel NexFET Power MOSFETs datasheet," May 2014. [Online]. Available: http://www.ti.com/lit/ds/symlink/csd19534q5a.pdf. [Accessed 17 May 2020].

[62] Microchip, "ATmega4808/4809 Data Sheet," 2020. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega4808-4809-Data-Sheet-DS40002173A.pdf. [Accessed 17 May 2020].

[63] Texas Instruments, "DRV10987 12- to 24-V, Three-Phase, Sensorless BLDC Motor Driver datasheet (Rev. B)," February 2018. [Online]. Available: http://www.ti.com/lit/ds/symlink/drv10987.pdf. [Accessed 17 May 2020].

[64] Texas Instruments, "LMR16030 SIMPLE SWITCHER® 60 V, 3 A Step-Down Converter With 40 μA IQ datasheet (Rev. A)," May 2016. [Online]. Available: http://www.ti.com/lit/ds/symlink/lmr16030.pdf. [Accessed 17 May 2020].

[65] R. K. a. P. N. M. S. Sakunthala, "A study on industrial motor drives: Comparison and applications of PMSM and BLDC motor drives," in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, Chennai, 2017.

[66] H. C. a. T. Yigit, "Field-Oriented Control of the PMSM with 2-DOF PI Controller Tuned by Using PSO," in *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, Malatya, 2018.

[67] Microchip, "Motor Control Library for dsPIC33E and dsPIC33F," [Online]. Available: https://www.microchip.com/design-centers/motor-control-and-drive/motor-control-library. [Accessed 17 May 2020].

[68] Samtec, "PCI Express Card Sockets," Samtec, [Online]. Available: http://suddendocs.samtec.com/catalog_english/pcie.pdf. [Accessed 05 2020].

[69] Microchip, "ATmega16U4/32U4 Datasheet," Microchip, 04 2016. [Online]. Available: http://ww1.microchip.com/downloads/en/devicedoc/atmel-7766-8-bit-avr-atmega16u4-32u4_datasheet.pdf. [Accessed 05 2020].

[70] Riverbank Computing, "PyQt5 Reference Guide," Riverbank Computing, [Online]. Available: https://www.riverbankcomputing.com/static/Docs/PyQt5/index.html. [Accessed 05 2020].

[71] Qt, "All Classes | Qt Documentation," Qt, [Online]. Available: https://doc.qt.io/qt-5/classes.html. [Accessed 05 2020].

[72] OpenCV, "Reading and Writing Images and Video," OpenCV, [Online]. Available: https://docs.opencv.org/2.4/modules/highgui/doc/reading_and_writing_images_and_video.html. [Accessed 05 2020].

[73] PyGame, "PyGame Joystick," PyGame, [Online]. Available: https://www.pygame.org/docs/ref/joystick.html. [Accessed 05 2020].

[74] PySerial, "pySerial API," PySerial, [Online]. Available: https://pyserial.readthedocs.io/en/latest/pyserial_api.html. [Accessed 05 2020].

[75] Python, "The ElementTree XML API," Python, [Online]. Available: https://docs.python.org/3/library/xml.etree.elementtree.html. [Accessed 05 2020].

[76] Qt, "QThread Class," Qt, [Online]. Available: https://doc.qt.io/qt-5/qthread.html. [Accessed 05 2020].

[77] Qt, "Signals & Slots," Qt, [Online]. Available: https://doc.qt.io/qt-5/signalsandslots.html. [Accessed 05 2020].

[78] Texas Instruments, "LM5146-Q1 100-V Synchronous Buck DC/DC Controller With Wide Duty Cycle Range," November 2018. [Online]. Available: https://www.ti.com/lit/ds/symlink/lm5146-q1.pdf?&ts=1589562632399. [Accessed 17 May 2020].

11. Appendix

05-11-2020 15:07:33 - LST Server is Waiting.
05-11-2020 15:07:34 - Connect (1) : [DESKTOP-5KK2LGI: 192.168.0.87]
05-11-2020 15:07:34 - Write: 903.26 Mbps (1)
05-11-2020 15:07:34 - Read: 895.23 Mbps (1)
05-11-2020 15:07:34 - Connection Closed (1) : [Remote IP: 192.168.0.87]

05-11-2020 15:07:34 - LST Server is Waiting.
05-11-2020 15:07:35 - Connect (1) : [DESKTOP-5KK2LGI: 192.168.0.87]
05-11-2020 15:07:36 - Write: 905.63 Mbps (1)
05-11-2020 15:07:36 - Read: 899.56 Mbps (1)
05-11-2020 15:07:36 - Connection Closed (1) : [Remote IP: 192.168.0.87]

05-11-2020 15:07:36 - LST Server is Waiting.
05-11-2020 15:07:37 - Connect (1) : [DESKTOP-5KK2LGI: 192.168.0.87]
05-11-2020 15:07:37 - Write: 900.00 Mbps (1)
05-11-2020 15:07:37 - Read: 903.79 Mbps (1)
05-11-2020 15:07:37 - Connection Closed (1) : [Remote IP: 192.168.0.87]

Figure 60: Data log of LAN speed tests.

Figure 61: ESC Test Setup

Figure 62: JO Performing tests on the data board

Task	Description
Plastic Pollution	→ Seabin: ○ Disconnect old power connector from recently installed seabin ○ Remove mesh catch bag from previously installed seabin ○ Install new mesh catch bag into seabin ○ Reconnect power connector to recently installed seabin → Remediation: ○ Remove floating plastic debris from the surface ○ Remove a ghost net from midwater ○ Remove plastic debris from bottom of trench
Effects of climate change on Coral Reefs	→ Autonomously fly a transect line over a coral reef ○ Autonomously/manually map points of interest on the reef → Use computer vision to determine the health of coral reef by comparing current conditions to past data → Propagate corals onto the reef ○ Remove coral fragments from nursery structure ○ Outplant coral fragments to designated locations on coral reef → Cull an outbreak of sea stars → Collect samples of sponge species
Maintaining healthy waterways	→ Retrieve sediment sample from inside drain pipe to analyse for contaminants ○ Deploy a device into pipe and collect sediment sample ○ Return sample to the surface ○ Determine type of contaminants present in sample → Estimate number of mussels in mussel bed ○ Deploy a quadrat ○ Count number of mussels in the bed ▪ Estimate the total amount of water filtered by mussel bed ○ Eel restoration ▪ Remove a trap full of eels from a designated area ▪ Place empty eel trap in a designated area → Autonomously/manually create a photomosaic of a subway car submerged to create an artificial reef.

Appendix Table 1: Competition tasks.

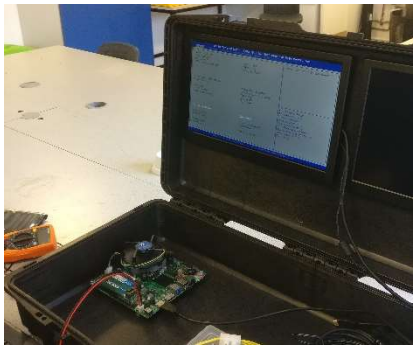
Command Initialiser	Function Identifier	Data	Return Data	Description
?I	-	-	AVALONROV	Gets the identity of the connected device.
?R	T	aaabbbcccdddeeefffgggghhh	-	Sets the speed of each thruster. ("aaa" = the speed of thruster 1 from *001 - 999*)
	A	abcdefgh	-	Sets the state of each actuator. ("a" = the state of actuator 1, *0* or *1*)
	C	aabbccdd	-	Sets the analogue cameras to display. ("aa" = the address of camera 1, from *01 - 99*)
	S	-	a,b,c	Gets the value of each sensor and returns them in a comma separated array. ("a" = the value of sensor 1)
	X	-	-	Initiates the arming procedure for the thruster electronics speed controllers.
?M	X	a	-	Activates the mini-ROV, arms the thruster ESC and turns the headlights on. ("a" = *1* for ON, *0* for off.)
	T	aaa	-	Sets the speed of the mini-ROV thruster. 'Aaa' represents the speed from '001' - '999'.

Appendix Table 2: ROV Serial Commands.

Operation	Command
Set thrusters 1 & 3 to full speed	?RT99950099995005005005O0
Turn actuators 1 & 3 ON, and actuator 2 OFF	?RA101
Get sensor readings	?RS

Appendix Table 3: Example ROV commands.

Appendix Figure 1: Surface station.



Appendix Figure 2: Surface station computer.

Appendix Figure 3: ROV chassis materials.