Worcester Polytechnic Institute
Digital WPI

Major Qualifying Projects (All Years) | Major Qualifying Projects

April 2009

Design of Autonomous Underwater Vehicle and Optimization of Hydrodynamic Properties and Control

Akhil Kejriwal
Worcester Polytechnic Institute

Brandon Michael Habin
Worcester Polytechnic Institute

Maxwell E. French
Worcester Polytechnic Institute

Radu Alin David
Worcester Polytechnic Institute

Follow this and additional works at: https://digitalcommons.wpi.edu/mqp-all

Repository Citation
Kejriwal, A., Habin, B. M., French, M. E., & David, R. A. (2009). *Design of Autonomous Underwater Vehicle and Optimization of Hydrodynamic Properties and Control*. Retrieved from https://digitalcommons.wpi.edu/mqp-all/292

This Unrestricted is brought to you for free and open access by the Major Qualifying Projects at Digital WPI. It has been accepted for inclusion in Major Qualifying Projects (All Years) by an authorized administrator of Digital WPI. For more information, please contact digitalwpi@wpi.edu.

Project Number: AE-IIH-0001

Design of Autonomous Underwater Vehicle and Optimization of Hydrodynamic Properties and Control

A Major Qualifying Project Report
Submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Bachelor of Science
in Aerospace, Electrical and Mechanical Engineering

by

Radu A. David _____
Maxwell E. French _____
Brandon M. Habin _____
Akhil Kejriwal _____

Date: Tuesday, April 28, 2009

Approved:

_____
Prof. I.H. Hussein, Major Advisor

keywords
1. AUV Design and Control
2. Submarine
3. Hydrodynamics

_____
Prof. W.R. Michalson, Co-Advisor

Certain materials are included under the fair use exemption of the U.S. Copyright Law and have been prepared according to the fair use guidelines and are restricted from further use.

Abstract

Autonomous vehicles are becoming more and more prevalent in military, private industry and residential applications. Research currently being done in the area of autonomous underwater craft is often hindered by expense. It was desired to build a craft at WPI which could serve as an inexpensive test-bed for future research and implementation of control algorithms, etc. The vehicle's construction required the design and manufacture of a number of components including water-jet stabilization thrusters, propeller driven main thrusters, a complete multi-hub electronic sensor and drive control system and individual sensors such as a tri-axial sonar unit as well as a capacitance based water-leak sensor. A Lexan heat forming process was also developed for hull construction.

Executive Summary

This purpose of this project is to create a low-cost, Autonomous Underwater Vehicle (AUV) which can serve as a test-bed for testing and implementation of software and hardware systems such as control algorithms and navigation units, and can also be used for educational purposes to study the dynamics of underwater craft. Although cheap AUV's do exist out in the market, adaptable multi-role test-beds are expensive, costing upwards of $50,000. Thus, the market has inexpensive craft – usually intended for shallow water applications, and more expensive craft – intended for a multitude of operations, but no craft in between which can serve as an inexpensive, yet adaptable system test bed. This project aims to provide such a vehicle. The completed vehicle is shown in 1, below.

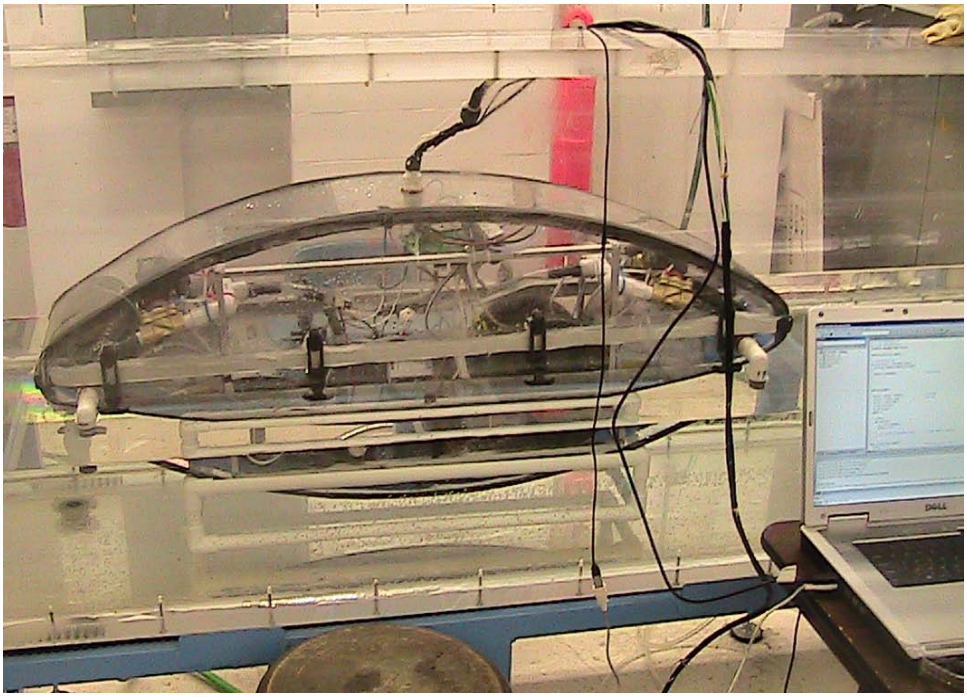


Figure 1. Completed Vehicle.

As the craft was intended to be used for further research, it was important that it be easily modifiable and that it afford room for drastic design changes and installation of additional systems without a need to overhaul any of the current systems. This meant that each system was designed to work and operate as is, but easily accommodate additions, design changes or complete removal. To keep the cost of the craft low and meet our design needs, most subsystems were designed from the ground up, and a majority of the

manufacturing was performed in house at WPI. This includes the design and manufacture of an application specific ballast system, magnetic coupling driven thruster system and a sonar system for range finding. Where needed, new tools – such as thermoforming ovens – were designed and built to carry out processes such as molding Lexan for the hull design.

On completion, this system will be a prime candidate for research test beds in AUV control. Furthermore, due to the inexpensive nature of this AUV, this craft could also be utilized to research and test simultaneous control of multiple vehicles exhibiting swarm behavior, where a fleet of autonomous vehicles coordinate (autonomously) to carry out larger operations such as quick systematic sweeping of vast areas. This type of technique can be useful for anything from underwater hazard detection such as mine sweeping to ocean floor mapping operations.

Acknowledgements

We would like to express thanks to Pat Morrison, Tom Angeliotti, Randy Harris, Neil Whitehouse and Prof. J. Blandino for their technical assistance.

Nomenclature

E, V	Volts
I	Amperes
W	Watts
Ah	Ampere-Hour
Wh	Watt-Hour
L	Inductance
n	Turns ratio
N	Number of turns
P	Power
Z	Impedance
GHz	Gigahertz
Gb	Gigabyte
ADC	Analog to Digital Converter
DAC	Digital to Analog Converter
PCB	Printed Circuit Board
SPI	Serial Peripheral Interface
I^2C	Inter Integrated Circuit Control
I/O	Input/Output
C	Celsius
F	Fahrenheit
", in	Inch
' , ft	Foot
°	Degree
DC	Direct Current
AC	Alternating current
JTAG	Joint Test Action Group
ATX	Advanced Technology Extended
m	Meters
m/s	Meters per Second
s	Seconds

Table of Contents

List of Figures

List of Tables

1 Introduction

In order to explore the hydrodynamic navigation and control schemes for an autonomous multi-vehicle system, it was deemed necessary to design and build a low-cost underwater vehicle. This vehicle is to serve as a test platform for communication, navigation and positioning algorithms and hardware, as well as future research and project work to be performed at WPI. The following vehicle subsystems were devised in order to complete the task: Attitude control and positioning system, navigation system, and a full electronics system to handle all sensors and communications.

The attitude and positioning control system was designed to include two propeller-driven main thrusters for yaw and forward/reverse propulsion as well as a series of four solenoid actuated water jet thrusters (for pitch and roll control a water jet thruster would be positioned at each corner of the craft). These thrusters would be powered by a water pump that delivers high pressure flow and a high mass flow rate. The water jet thruster system is integrated with the cooling and ballast tank systems. These three systems will utilize the water pump and the flow of water through the craft. The need for a cooling system due to excess heat produced by all electronic components was explored. Also, the main thrusters feature a machined magnetic coupling which eliminates the need for any dynamic seals.

The navigation system required the development of a 3-axis sonar sensor for main position determination as well as an integrated 3-axis gyroscope and accelerometer. For the navigation system, an ultrasonic range finder was designed and built based on the echo sounder transducers used in fish finder applications. By using three range finders controlled by a central unit, the distance from the front, side and bottom of the vehicle can be obtained. Together with the information from the gyroscope and accelerometer can give a very accurate position and would allow a trajectory mapping. A two-axis magnetic compass was also used to provide heading information. Fore and aft pressure sensors were used in conjunction with the sonar system to determine depth.

The electronics system implements a three hub serial bus (fore, aft and center) to handle all sensors and drive components and interface with the central computer. A single printed circuit board was designed with provisions for multiple configurations. A capacitive water-leak sensor was also developed to detect seal leaks and pooling. The

central computer, running Debian Linux, was outfitted and interfaced with both an analog to digital converter and frame grabber to allow the use of various data collection devices (such as acoustic imaging hardware).

A number of hull solutions were also examined. The first hull was built out of polystyrene sheet. The second iteration of the hull was formed out of Lexan using a purpose built heat molding oven. Several hull sealing solutions were examined. Ultimately, a custom sealing system was tested and designed, resulting in a rubber lined 'H-seal' to waterproof the hull. Rubber draw latches were utilized to keep the hull together and keep the seal intact.

2 Background Information

Submarines are a class of craft capable of operating underwater. Initially developed as craft for anti surface-ship warfare, they have diversified from traditional war machines to perform roles such as oceanographic research, salvage and space-craft simulation.

In recent years, use of autonomous craft has become much more prevalent. This is due, in part, to the cheaper cost per hour of operation for one of these craft as compared to a manned submersible, their ability to access highly hazardous areas and their ability to run continuous tests for days autonomously without human interaction. Thus, significant research efforts worldwide are being made toward optimal controls for these craft.

However, research test-beds for craft can be expensive – adaptable systems such as the IVER AUV cost upwards of $50,000[1]. Admittedly, less expensive vehicles do exist, such as the Serafina[2], a low cost shallow water craft. However, these vehicles do not have the adaptability to test out different subsystems. Thus it is apparent that there exists a niche in the market – for low cost but adaptable AUV test-beds – which still needs to be filled.

Before construction of the various components needed to complete this craft, research was conducted on design principles and existing variations of the components.

2.1 Hull

The external surface of the submarine is called its hull. The hull essentially serves as a barrier between the surrounding water and the working innards of the craft, as well providing an optimized hydrodynamic body for motion through water. There are two main categories for hull design – a single hull system and a double hull system.

As the name implies, the two functions of a hull are integrated into a singular hull in the single hull system. The same hull provides both a water and pressure barrier, and a hydrodynamic front for the submarine. This system allows for a larger payload capacity for a particular hull design, but can be hard to design effectively as optimal designs to withstand pressure and maintain the integrity of the hull are different from optimized hydrodynamic shapes.

The double hull system separates the two functions of the hull, with an external hull acting as a hydrodynamic surface for motion through water and a second internal hull acting as a pressure barrier. The space between the two hulls is either flooded, used to house ballast tanks, or used for other storage purposes. [3]

As the craft was intended for shallow water applications, plastics were explored as potential hull materials. An initial mockup of the hull was made using polystyrene, with the final hull being made from ¼" Lexan. Lexan, although sturdy at a ¼" thickness, is rigid and cannot undergo bending easily. Thus to form Lexan to the curves we needed, heat forming techniques for the material were examined.

As opposed to simply mechanically deforming a plastic to form its desired shape which induces stresses and strains into the material, heat forming allows for deformation of materials with only minute residual stresses remaining. The process of heat forming plastics involves heating a plastic from its initial solid state to its plastic region, mechanically deforming and constraining the plastic to the desired shape and finally letting the plastic cool and cure to its final shape. The plastic can be constrained to its final shape utilizing a combination of a mold or applied suction/pressure.

Due to its hygroscopic nature, Lexan has a tendency to bubble when formed. To prevent this, Lexan must be prebaked at 250°F for about an hour to get rid of all surface moisture before it can be heated to the forming temperatures of 350 – 400°F. When formed, the Lexan needs to be cooled slowly without exposure to cool or moist air as this can induce residual strains and bubbling in the Lexan surface. [4]

2.2 Ballast System

In order to be able to dive below the surface, a submarine needs to be able to change its buoyancy from positively buoyant for surface operations to neutrally or negatively buoyant for underwater operations. This is achieved with the help of the submarine's ballast system.

A ballast tank is a compartment within a craft which holds water. The tank can be filled or emptied – effectively changing the displaced volume and thus the buoyancy of the craft – to allow the craft to descend or ascend. Larger vehicles will have multiple ballast tanks, which include the main ballast tank, the negative buoyancy tank,

emergency tank and the foreward and rear trim tanks (see Figure 2). The main ballast tanks are the tanks which account for the primary change in the buoyancy of the vehicle – flooding these tanks usually brings the craft to neutral buoyancy. They are located towards the centre of the craft. The foreward and rear trim ballasts are smaller tanks located at the front and rear extremities of the craft. They are used for fine control of the buoyancy of the craft as well as zeroing the pitch of the craft. Traditionally, these are differentiated from the blast tank because a large displacement of water within the main ballast tanks as the submarine maneuvers could potentially upset the craft. The safety tank is kept flooded throughout normal operation and can be emptied during emergencies to add to the positive buoyancy of the craft. The negative buoyancy tank can be flooded

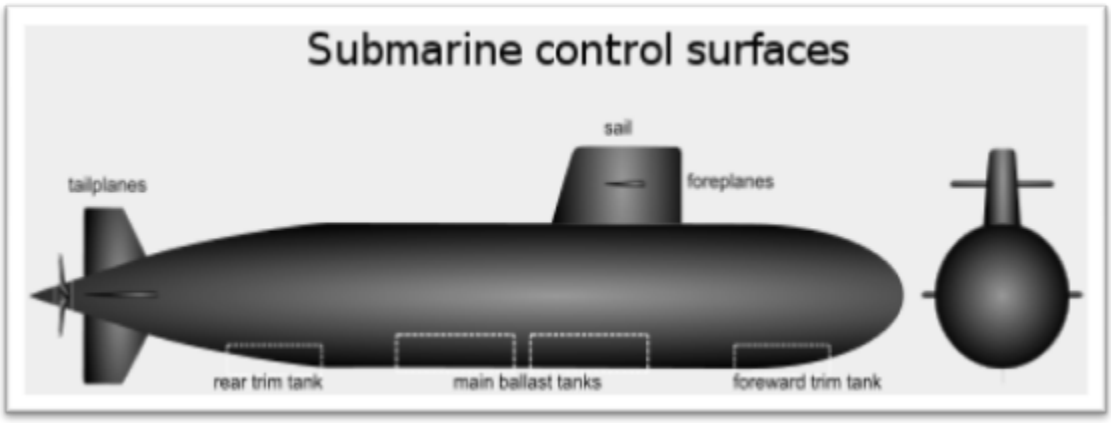


Figure 2. Ballast Layout in a Submarine. [5]

to make the craft negatively buoyant, a maneuver used to initiate sudden dives.

Regulation of water levels in the ballasts is performed via pressurized air. To fill the ballasts, valves to the surrounding water and vents to the atmosphere (or surrounding water) are opened. This allows air to escape the tanks, filling them with water. To empty the tanks, compressed air is utilized to force out the water[6].

One other method utilized in submersibles to control buoyancy is the release of weights attached to the craft. Submersibles often carry extra plates of steel or lead to add to negative buoyancy for the craft. When needed, these plates can be released, making the craft positively buoyant and allowing it to rise to the surface.

2.3 Attitude Control

In addition to the ballast system, which can help with the trim of the vehicle, there are two methods through which one can control the attitude of the craft – hydroplane control and thruster control.

Hydroplane control involves controlling yaw, pitch and roll of the craft with the help of control surfaces, much the same as the elevator, rudder and aileron system of an airplane. In order to work effectively they need a minimum craft velocity, and thus cannot be utilized for hovering applications. However, they have a relatively low power requirement for lengthy control maneuvers at larger velocities.[7] Many larger vehicles use this system along with their ballast systems to control the craft.

Thruster control systems are typically utilized in low velocity applications for vehicles, as they become ineffective at higher velocities. Thruster actuation can be achieved via water jet thrusters and traditional propeller driven thrusters. The water jet thrusters are more effectual for short low velocity applications as compared to propeller driven thrusters.[8] Given the nature of our system, a thruster control system was designed for the vehicle.

2.4 Magnetic couplings

Now, a thruster system had to be designed to drive the craft. The inherent problem with designing a thruster system for a water-craft is sealing, as that the main driving motor cannot come in contact with water, while the driving propellers have to. Instead of attempting to design and manufacture a system with a dynamic seal on a rotating shaft, a static seal system utilizing a magnetic coupling was envisaged and designed.

Permanent magnetic couplings (PMC) allow for power transmission between a primary drive shaft and a secondary follower via magnetic fields and no mechanical contact. Thus the two components – the drive shaft and the follower – can be located in two different media with each permanently sealed from the other. This static seal makes it much easier to ensure that the motor does not come into contact with any harmful media. Furthermore, the coupling can act as a torque limiter for the follower. Thus, if the follower's motion is damped or arrested in any fashion, the coupling will break ensuring no damage is done to the motor or drive-shaft. Magnetic couplings can be made co-

axially and linearly or face to face. The co-axial coupling allows for a higher torque transmission, while the linear coupling allows for some misalignment of the coupling. Given the nature of a thruster system, a co-axial coupling was designed to transmit torque to the propellers.[9]

2.5 Navigation Systems

The purpose of a navigation system is to track the location of the vehicle and to detect objects with respect to the surroundings. Initially, a series of existing navigation systems were considered and analyzed. The first system considered was Light Detection and Ranging (LIDAR). This technique uses laser scattering to detect distance very precisely, and it is commonly used for atmospheric measurements, air distance measurements or astronomy. Although very accurate, it is very susceptible to small particles and the cost is rather high.

A DVL (Doppler velocity logger) uses the more classic approach of emitting ultrasonic pulses. By calculating the frequency shift in the return wave, it determines the speed of the vehicle. Integrating this, a relative displacement of the vehicle can be determined. This device is ideal for the needs of the project, but the cost and complexity don't make it a very good candidate.

Inertial navigation systems (INS) use accelerometers and gyroscopes to accurately determine the heading and the acceleration of the vehicle. They rely heavily on the accuracy of the data, and it is the only system that doesn't require interaction with the outside, which also means they cannot detect objects or obstacles. However, they can validate data from secondary sources, and will be implemented on the submarine together with a ranging device for accurate position determination.

An altimeter was another application that was considered. Tritech's Digital Precision Altimeters work on either 200 KHz or 500 KHz and have both analog and digital data output, which would suit the needs of our project. The device already calculates the time of flight and the corresponding distance and returns the data. Although this device works and is used in the industry, its size or price would not allow its use in the design of the AUV we are building.

The Devantech SRF04 air based range finder was a useful device, and although not for water, the sensor had a very simple design and gave a lot of insight on the design of rangefinders[10].

2.6 Ultrasonic range finding

To gain a better understanding of the technology, research on the physics of the ultrasound was conducted. The biggest concern was the difference in the acoustic characteristics of water and air. Some acoustic equations and constants were identified:
Sound Pressure Level: SPL [dB] = 20 log (P/P_r), where P is the wave pressure and P_r is the reference pressure. For water P_r = 1 µBar while for air P_r = 20 µBar. As our device will be operating in a pool, the relevant speed of sound - 1521 m/s – will be utilized, which is comparably larger than the speed of sound in air (343 m/s) resulting in smaller flight times for same distances.

The attenuation between air and water also differs. For water the attenuation is 8.8 dB/km while in air the value is 1320 dB/km. There were three orders of magnitude difference between the two media, which were taken into account when working with the sensors.

3 Methodology and Implementation

The vehicle itself was broken down into a number of mechanical and electrical subsystems. The mechanical subsystems consisted of a hull in which to house all other systems, a thruster system to provide stabilization and propulsion and a ballast system to allow depth control and trim. The electrical subsystems consisted of a main electronics system to interpret position feedback and control the ballast and thruster systems. The main electronics system also included a number of sensors to provide system health information and redundant positioning information. A sonar navigation system was developed as the primary positioning sensor to interface with the main electronics system. Finally, an initial set of control equations were developed. Once implemented these equations will bring together all subsystems and allow control of simple autonomous operations (i.e. auto-stabilization).

For the construction process, a hull had to be built first to house all of the vehicle's components. Once the hull was completed the inner components could be built. The hull construction process required the development of a heat-molding process and heat-molding equipment. On completion of the hull, a series of heat transfer experiments and analyses were conducted to decide whether or not an internal cooling system would be necessary to maintain a safe temperature. In the end it was shown that such a system would not be necessary.

The thruster system consists of propeller driven propulsion thrusters and water jet driven stabilization thrusters. The propulsion thrusters use a magnetic coupling created specifically for this application to drive propellers external to the hull. The propulsion thrusters can control heading (yaw) and forward/reverse motion. The water-jet powered stabilization thrusters use convergence nozzles (also designed and machined at WPI) and a water pump installed in the craft to provide pitch and roll control.

The ballast system consists of two tanks, built in-house, positioned at either end of the craft and can control depth as well as pitch. This system shares the water pump installed in the vehicle with the water jet thruster system. Together with the thruster systems, this allowed us complete controllability of the vehicle.

Once the thruster and ballast system designs had been decided upon an electronics system was designed to control them. This system consists of a central processing unit to

receive position and system health information from a network of sensors and output control information to the thruster and ballast system actuators through the “Sub-Hub System”, which was devised to handle all sensors and actuators in the vehicle.

The main position determination sensor, a 3-axis sonar navigation system, was also developed at WPI specifically for this vehicle.

Finally, basic control equations were developed and will soon be implemented to make full use of the interface between the various electrical and mechanical subsystems and allow for autonomous control.

The interaction between the thruster and ballast systems and the electronics is described in Figure 3, below.

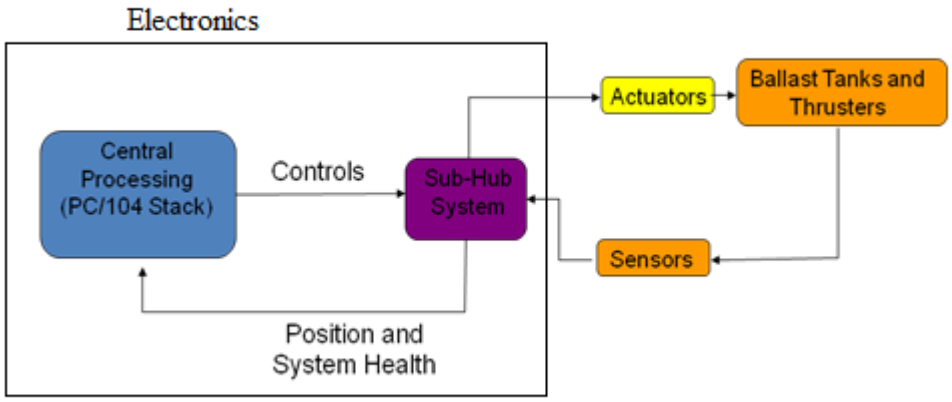


Figure 3. Sub-System Interactions Block Diagram.

3.1 Hull

To allow for a maximum payload capacity for a given design, a single hull design was chosen for our craft. As can be seen in Figure 5, the hull consists of two symmetrical pieces (top and bottom) which came together to form an oval shape. It measured 46” in length and 8” in width, with the curved profile being that of a circle passing through the two ends and a point 6” above the centre of the hull. This design allowed complete and easy access to all internal components while still maintaining a degree of hydrodynamics. It also provided flat surfaces on the sides to securely mount the stabilization and propulsion thrusters.

3.1.1 Polystyrene Hull

The curved top and bottom obviously posed some obstacles when it came to construction (as compared to fully flat surfaces). Because of this, the first iteration was made out of 1/8” thick polystyrene sheet. This material was relatively flexible and could easily be bent by hand to follow the contours of the curve. The four side panels of the polystyrene hull were cut out by hand using hobby knives and following a printed template. The curved panels were then clamped against the side panels and plastically welded using Ambroid ProWeld Plastic solvent. This process is shown in Figure 4, below.

Figure 4. Example of Polystyrene Hull Build Process.

While the polystyrene hull allowed a mock-up of the internal layout of the vehicle’s components, the hull was relatively weak and did not hold up well against the light pressures that the vehicle would be subjected to. Large deflection of the side-panels under pressure also made it difficult to reliably seal the hull. The completed polystyrene hull is shown in Figure 5. Along with the impracticality of the polystyrene hull, a transparent hull was desired. As this vehicle is to be used as an educational development tool it was deemed important to be able to monitor functionality such as ballast tank filling and water leak sensing. It was also desirable to leave the option of mounting imaging hardware throughout the hull for data collection purposes.


Figure 5. Completed Polystyrene Hull.

3.1.2 Lexan Hull

The second iteration of the hull, which is currently in use, was built out of ¼" Lexan. Lexan at this thickness is much stiffer than polystyrene and the top and bottom pieces could not simply be bent like the polystyrene. In order to achieve the curve needed, a special oven was built and the Lexan was heat formed to fit the curve.

3.1.3 Oven

The oven built to heat-form the curved top and bottom pieces of the hull consists of a sheet metal (steel) enclosure with a hinged top. The enclosure measures two feet in width and height and five feet in length. The aluminum legs hold the enclosure two feet off the ground for a total height of four feet. The top and sides are covered in 1/4" thick fiberglass insulation.

The first tests of the oven used an AC dummy load resistor array as an electrical heating element. This unit is shown in Figure 6, below. No bottom was installed on the oven and the resistor array was turned on its side and raised into the oven (from the bottom) on cement blocks. At full power, the unit drew 50A at 120V or 6kW (20,473 Btu/hr), all of which can be assumed to have been converted to heat. Aluminum tape was placed around the edges of the top in an attempt to minimize heat losses from the enclosure area. A K-type thermocouple was inserted from the back of the oven to allow digital monitoring of internal temperature.

After 2 hours at full power, the internal temperature of the enclosure stabilized at around 240°F. The heat forming process requires that the Lexan be heated in two stages, the first at 250°F and the second at 300°F. Consequently, a more powerful heating element needed to be explored.

Figure 6. AC Dummy Load Resistor Array.

The AC dummy load was removed and a bottom panel was installed in the enclosure with cutouts for two 45,000 Btu/hr propane gas burners. As the breakdown temperature for Lexan is about 420F, with heat forming temperatures nearing 400F, the oven was tested in the WPI Fire Lab under a large hood for safety. With a total heating element power of 90,000 Btu/hr the oven instantly reached temperatures in excess of 300°F with the burners on a medium setting. The burners were therefore set to their lowest setting and the top was vented using Kaowool ceramic fiberboard, to keep the oven at a steady 250°F. Reducing ventilation through the top of the oven allowed the necessary molding temperature of 350°F to be reached as well.

Both male and female molds were created to form the curve of the top and bottom pieces of the hull. The hull design was slightly modified to include a square sealing surface for the two halves of the hull. The molds were made out of 1/8” hardboard bent around 1/2” plywood ribs and secured with wood screws. The bottom of the male mold was covered with fiberglass insulation. The insulation extended six inches beyond the mold on each end to protect against direct heating of the end flaps of the Lexan which were to be bent straight down. The Lexan was placed on top of the male mold at the start

of the molding process. When the heat forming temperatures were achieved, the female mold was simply pushed down on top of the male mold.

Two grates were placed inside of the enclosure about four inches above the tops of the burners. Kaowool ceramic fiber blanket was placed on the grates, over the burners, in an attempt to dissipate heat. The male mold (with fiberglass insulation attached to the bottom) was placed directly on top of the Kaowool blankets.

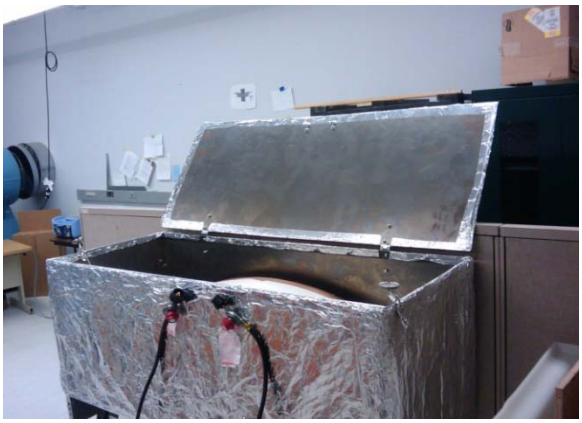


Figure 7: Final Oven with Gas Burners

3.1.4 Construction

Once the top and bottom panels had been formed, the side panels were cut-out with a band-saw and a construction process similar to that of the polystyrene hull was used. A slightly stronger methylene-chloride solvent solution (as compared to the ProWeld used for the polystyrene) was used to plastically weld the constituent parts of the Lexan hull.

When the two hull halves were constructed they were each cut down on a band saw to ensure that their adjoining portions were square and level with respect to each other.

3.1.5 Sealing

The internal seams where the side panels meet the top and bottom panels were lined with clear silicon sealant. The external seams were sealed with cyanoacrylate to fill any gaps and then with a layer of silicone sealant. Grip-Dip liquid rubber was painted on to the external seams to ensure that no leaks would occur.

To seal the two halves of the hull together, a custom sealing system was designed. A framework which fit the two halves of the hull was made from H-bar aluminum extrusions (shown in Figure 8). This was attached to one half of the hull using silicon sealant. The other half of the H-bar was lined with rubber gasket material, and was used to form a compression seal with the hull. To help maintain the integrity of this dry seal, a combination wet seal was also utilized – the H-bar was filled with petroleum jelly on top of the EPDM gasket, which ensured a complete seal. Rubber draw latches were used to keep the two halves of the hull together and keep the seals intact. The completed hull is shown in Figure 9.

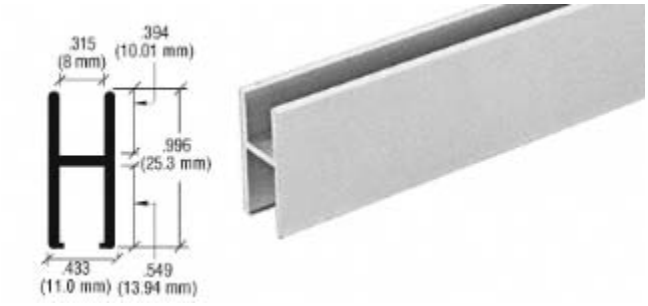


Figure 8. Extruded H-Bar

© 2008 C.R. Laurance Inc.[11]

Figure 9. Completed Hull.

3.1.6 Frame

The internal frame of the sub serves a dual purpose. It provides support to the sides, decreasing deflection and thus improving the seal as well as providing a system to which all internal components are mounted.

The current frame simply consists of a rectangular bottom and top connected by four supports. Aluminum C-Channel (1/4”) was used to construct the frame in such a way that all components can be mounted on ¼” thick plates which can be slid along the frame to adjust the center of gravity. The plates are secured with set screws. The bottom of the frame is press fit into the bottom of the hull. The individual components of the frame are secured with screws allowing the entire frame to be removed if necessary. The completed frame can be seen in Figure 10 with a slide mount for the pump.

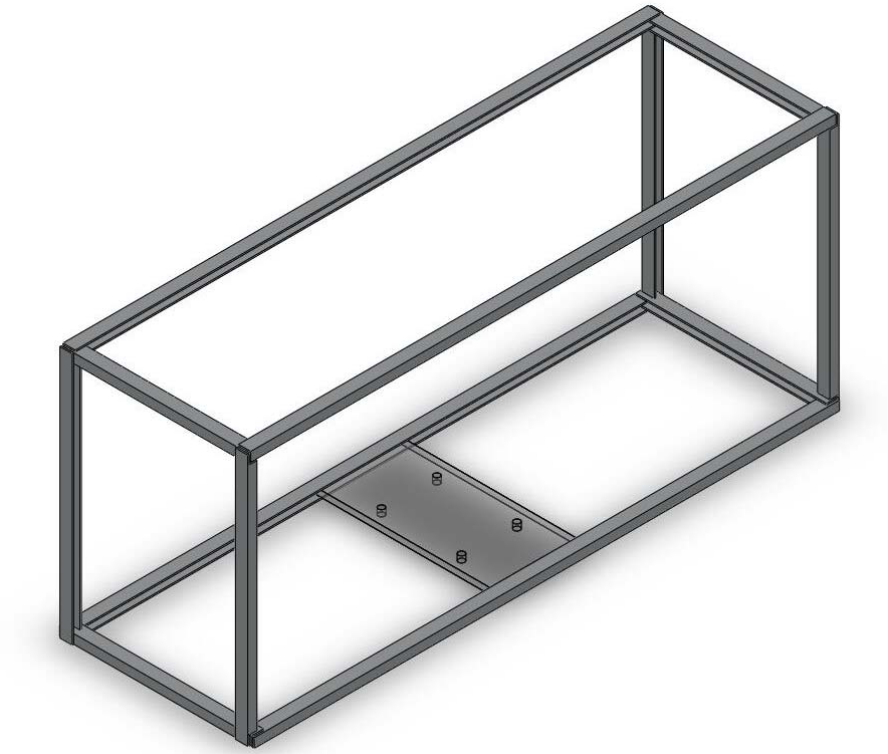


Figure 10: Hull Frame with Pump Slide Mount

3.1.7 Heat Transfer Analysis

As a number of electrical components are housed inside of the hull. Most of these components can operate at temperatures up to 125 °C. The limiting component, however, is the PC-104 which can only operate at temperatures up to 60 °C. It was important, therefore, to find out how much heat would be produced inside of the hull and how quickly the hull could dissipate this heat. This, in turn, would allow design for any necessary cooling system components.

3.1.7.1 Heat Transfer Through the Hull

An experiment was performed to test heat transfer through the polystyrene hull into surrounding water at ambient temperature. A 25W incandescent light bulb was installed in an 8x8x8 in box made of 1/8" polystyrene. A wireless temperature sensor (Texas Instruments EZ430 RF2500 Wireless development tool) was also installed in the enclosure. The base of the light socket was lowered from the top of the box leaving about one inch of space between it and the box to minimize heat conduction through the base of the bulb. The temperature sensor was mounted such that it was shielded from direct radiation by a piece of white construction paper and mounted one inch from the side of the box.

The actual power consumption of the bulb was measured to be 24.8W.

Assuming that 90% of the energy consumed by an incandescent light bulb is converted to heat[12] and that there was minimal heat conduction through the wires powering the bulb this means that there can be assumed to be a 22.3W heat source inside the box.

The box was submerged in a water tank with an ambient temperature of 14°C or 287K and allowed to cool to that temperature. The light bulb was then switched on until a steady state temperature was reached within the box. After about 20 minutes the internal temperature reached 27°C or 300K (see Figure 11). The experiment was continued for another 30 minutes while the internal temperature remained at its steady-state of 300K.

This temperature rise of 13K from a 22.3W heat source given a surface area of 0.24 m^2 yields a calculated total heat transfer coefficient to water of 7.147 W/mK for the

polystyrene sheet used in this experiment. The thermal conductivity of polystyrene is around 0.08 W/mk[13,14].

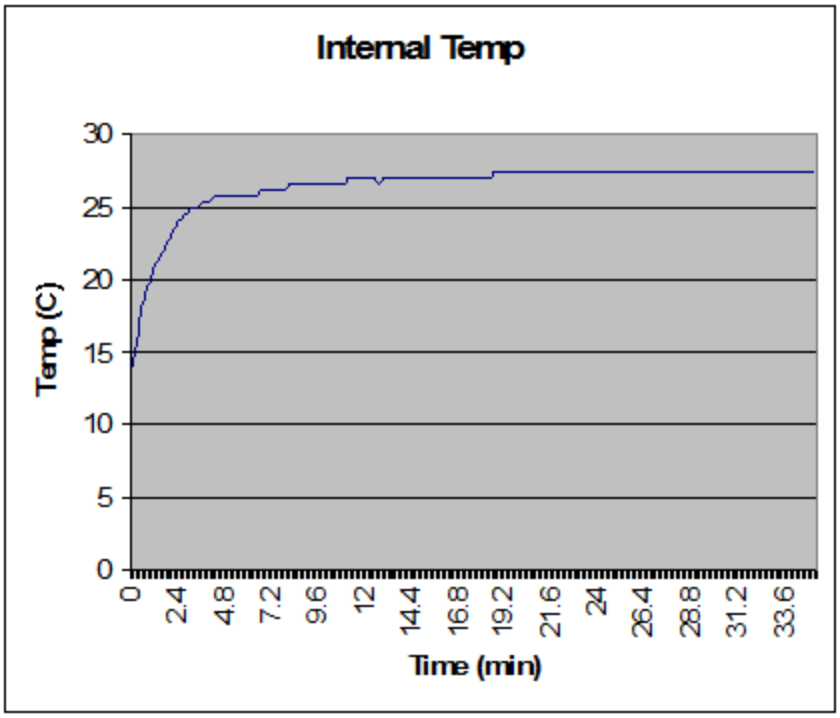


Figure 11. Hull Heat Transfer Experiment: Plot of Internal Temperature vs. Time

This experiment was not repeated for Lexan, however the thermal conductivity of Lexan, 0.2W/mK[15], is significantly more than that of the polystyrene. It can be expected that the total heat transfer coefficient to water would be about twice that of polystyrene as well. Since the Lexan used is twice the thickness of the polystyrene, about the same thermal properties are expected.

As the absolute maximum ambient temperature inside of the hull is 60 °C, as limited by the PC-104, a temperature rise of 20 °C is calculated for (leaving a 15 °C buffer assuming that the external ambient temperature is 25 °C). Given a conservative external hull surface area of about 1.2m^2 (for exposed Lexan alone and excluding the H-Bar surface area), the hull itself can dissipate over 170 W. Allowing a 25 °C temperature rise inside the hull (reducing the buffer to 10 °C) yields a hull heat dissipation of over 214 W.

As was determined in the “Power Analysis” section, the maximum expected power usage is about 175 W. This analysis assumes that 100% of the power used by the

electronics in the sub is converted to heat. It also ignores the fact that the main thruster motors are mounted external to the hull and therefore don't contribute any heat to the inside of the hull.

The hull, therefore, will be able to dissipate all generated heat safely on its own, without the need for a cooling system. It is important to note that this analysis ignores the added heat transfer properties of the extensive ballast tank and water jet thruster piping which are expected to dissipate significant heat on their own.

3.1.7.2 Water Pump Heat

In determining the heat produced in the hull, it was assumed that 100% of the power consumed by the electronics was converted directly to heat. To get a better idea of how much heat would actually be produced in the hull a quick experiment was done to determine the heat given off by the water pump which accounts for 45% of the total power consumption . The pump inlet and outlet were connected through a pressure gauge and valve to a bucket holding water at room temperature. The valve was adjusted such that the backpressure that the pump had to work against was about 35-40 psi (the pump shuts off automatically at 45 psi) to ensure that the pump was near its maximum load. A thermocouple was taped to the outside of the motor housing and the temperature was monitored for 60 minutes. The temperature reached a steady-state after about 15 minutes with a maximum temperature rise over ambient of about 2.5 °C. This relates to very small actual heat dissipation from the pump to ambient air. The results of this experiment suggest that the actual heat dissipated in the hull will be much significantly smaller than expected.

3.2 Thruster System

Several iterations of thruster system design were considered for our craft. The traditional propeller drive, hydroplane actuated design was discarded due to the low velocity applications for this craft.

A rotatable four-thruster system was conceived and designed, which would allow for maneuverability in low speed applications and stability for hover control. However, this system was deemed hard to implement and build, and a simpler system with fixed-position thrusters was conceived.

The current design utilizes two propeller based thrusters for yaw and forward/reverse propulsion as well as four solenoid activated water jet thrusters for pitch and roll control. This design allows for control of the craft at low velocities as well as stable hovering capabilities.

3.2.1 Main thrusters

The original thruster design was to consist of three main components: a pressure vessel to house the motor and drive system, a drive system utilizing a magnetic coupling to connect the motor to the propeller and the aforementioned motor driver circuit to interface the computer and thruster motor.

3.2.1.1 Pressure Vessel

The pressure vessel was the most simple of the three components consisting merely of a tube sealed at either end by end-caps. The two wires to power the motor enter through a hole in the “front” end-cap (see Figure 12), which is sealed with epoxy and gap-filling cyanoacrylate (CA) glue. A 2 x 2 terminal block was then fitted to the inside of the end-cap to connect to the motor.

Figure 12. Front End Cap.

The “rear” end-cap features a pocketed hole to fit the components of the magnetic coupling (see Figure 13).

Figure 13. Rear End Cap.

Both end-caps were machined from solid 3” diameter aluminum bar-stock using a CNC mill and manual lathe. Two o-ring grooves were cut in each to allow a watertight seal. An access point was also drilled into the side of the pressure vessel to allow adjustment of the magnetic coupling housed inside. An exploded view of the final pressure vessel components is seen in Figure 14, below.

Figure 14. Exploded View of Final Pressure Vessel.

3.2.1.2 Magnetic Coupling

The final component of the thruster design was much more complicated. The magnetic coupling was designed to use an "external" cup (housed inside the pressure vessel and connected directly to the motor) lined with two rows of N52 grade neodymium magnets with alternating polarities. The external cup then coupled with the "internal" cup (external to the pressure vessel and connected to the propeller) lined with two rows of magnets aligned with polarities opposite those of the external cup as seen in the drawing in Figure 15 below. The internal cup is connected to the propeller by a shaft machined from ¼" stainless steel bar-stock.

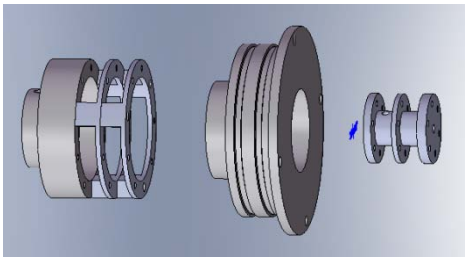


Figure 15. Magnetic Coupling.

The two cups of the magnetic coupling attract each other through the walls of the pocketed hole on the rear end-cap allowing them to turn in unison with no direct mechanical connection. This design, though more complicated in theory, removes the need for a dynamic mechanical seal thus making the design inherently more leak proof than a conventional direct drive shaft and shaft seal. The magnetic coupling design also removes the need for considerations of extreme pressure differentials on a mechanical shaft seal when optimizing the thruster for depth performance.

The magnetic coupling cups were machined out of 2.5" aluminum bar stock. To make manufacturing simpler, the external cup is designed as an assembly of three separately machined parts and the internal cup is a single piece (see Figure 16).

Figure 16. Machined Magnetic Coupling.

The fully assembled magnetic coupling (outside of the pressure vessel) as well as the motor and external cup (installed in the pressure vessel) is shown in Figure 17, below.

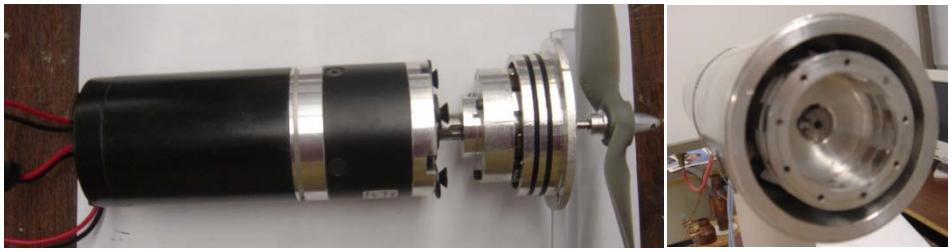


Figure 17. Assembled Magnetic Coupling.

The first iteration of this thruster design was based around an Anaheim Automation BDGP-60-110-24V-3000-R3.6 24V DC motor with a planetary gear box. This motor provides a rated torque of 57 oz-in at a rated speed of 694 rpm[30].

The first prototype of the thruster (using a 3" outer diameter aluminum tube with 0.24" walls) was first water tested by submersion for 24 hours. As no leaks were found, thrust measurements were then conducted in a large testing tank. The thruster was mounted to a steel rod (using hose clamps) which passed through a bar that was free to pivot. A scale was then attached to the top of the rod (the scale and thruster were equidistant from the center bar) so that thrust could be measured. The test rig and scale attachment are shown in Figure 18, below.

Figure 18. Main Thruster Test rig.

To operate at its rated torque and rpm the motor was expected to draw 2.2 amps. When tested, however, the magnetic coupling would decouple from a current draw of only 1.23 amps. This test is shown in Figure 19, below. This condition provided the maximum measured thrust of 3lbs, 2oz. Once the magnetic coupling had decoupled it provided barely enough torque to spin the propeller and no measurable thrust, requiring that the motor be stopped to allow the coupling to reconnect. At the maximum power (as limited by the magnetic coupling) the slightest current or jerky motion of the thruster would cause coupling failure. Which the motor drawing only 1 amp, however, (producing a thrust of 2 lbs, 1 oz) the coupling was still strong enough to resist failure from these conditions.

Figure 19. Main Thruster Test.

Though the measured thrust was much lower than expected or hoped it is believed that further adjustment of the magnets in the coupling system and possibly slight redesign of the magnetic coupling itself may increase the effectiveness of the design. The almost non-existent torque when decoupled raises concern but the relative strength of the

coupling when only slightly below its maximum torque suggests that slight adjustments may cause a decent increase in the torque required to decouple the drive system. Because the motor's capabilities were found to be in great excess of the drive coupling's capabilities a smaller motor was ordered. The Anaheim Automations BDGP-38-86-12C-4000-R5.2 uses a 12V supply and provides 26.4 oz-in of torque at a rated speed of 467 rpm. This motor will allow the use of a much shorter pressure vessel and will be more fitted to the torque achievable by the magnetic coupling.

The propeller used for this testing was a standard 9" model airplane propeller. The original intent was to machine a marine propeller and nozzle. A scale drawing of a Rice Speed Nozzle (shown in Figure 20, below) was created as a starting point for the design. However, this portion of the design has not yet been examined in depth.

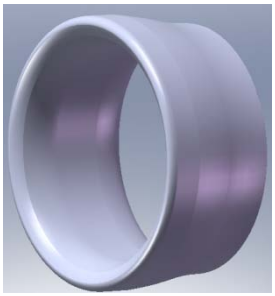


Figure 20. Rice Speed Nozzle.

3.2.2 Directional Thrusters

The water jet thrusters are positioned at the corners of the hull as to provide torque orthogonal to the forward facing axis for optimum efficiency. These thrusters are powered by a water pump that delivers high pressure flow, 45 psig max, and a high mass flow rate, 5 GPM. This pump allows for one water jet thruster to produce approximately 1.8 lbf. The thruster nozzles are made of Polyvinyl Chloride (PVC) piping for ease of machining, low cost, and low weight. Multiple tests were performed under ideal conditions, varying water flow rate and nozzle sizing in order to determine optimum nozzle size, maximum thrust, and backpressure produced. According to the data, the optimal nozzle was designed with a 30° taper, a 0.5in inlet and a 0.1875in exit. The data collected from these tests is shown in Appendix F.

3.2.2.1 Theoretical Thruster Calculations:

Initially a pump that produces a flow of 1100 GPH was utilized for testing. Applying the conservation of mass equations for this pump, the theoretical thrust produced by the nozzles can be determined:

$$T_{thrust} = \dot{m}V_2 - \dot{m}V_1$$

From 1100 GPH:

$$\dot{m} = 2.55\ \frac{lbm}{s}$$

From a pump hose Diameter of ¾ in:

$$= 13.32\frac{ft}{s}$$

With a reduction nozzle from ¾ in to 3/8 in:

$$\rho V_1 a_1 = \rho V_2 a_2$$

$$V_2 = 52.2\frac{ft}{s}$$

Assuming $\dot{m}$ stays constant and no losses:

$$T_{t\square rust} = 99.2\frac{ft \cdot\ lbm}{s^2} = 13.7\ N = 3.1lbf$$

See Table 14 for ideal thrust produced (no losses) by alternative sized reduction nozzles.

A variable flow meter was utilized for the second set of testing. This flow meter allowed for up to 5 GPM or 300 GPH of flow. The flow meter was attached to a standard water line which could work against up to 65 psig. Again, applying the conservation of mass equations for this pump:

$$T_{thrust} = \dot{m}V_2 - \dot{m}V_1$$

From 300 GPH:

$$\dot{m} = 0.694\ \frac{lbm}{s}$$

From a pump hose Diameter of ¾ in:

$$V_1 = 3.625\frac{ft}{s}$$

With a reduction nozzle from ¾ in to 3/16 in:

$$\rho V_1 a_1 = \rho V_2 a_2$$

and

$$V_2 = 56.86 \frac{ft}{s}$$

Assuming $\dot{m}$ stays constant and no losses:

$$T_{thrust} = 135.7 \frac{ft \cdot lbm}{s^2} = 18.7\ N = 4.2 lbf$$

See Table 15 for ideal thrust produced (no losses) by alternative sized reduction nozzles.

3.2.2.2 Nozzle Testing

Simple nozzles were constructed out of PVC with various outlet sizes and an 1100 GPH water pump was used to determine actual thrust produced for the control of the craft. Three tests were performed. The nozzles were constructed with outlet diameters of 3/8 in, 5/16 in, 1/4 in, 1/8 in, and 1/16 in. At a later date, converging nozzles were machined out of PVC, all with a 30° taper and with outlet diameters of 3/8 in, 5/16 in, 1/4 in, and 3/16 in. Two tests were performed with these nozzles with a flow meter and pressure gage attached to a standard water line connection.

3.2.2.3 Nozzle Test 1

Only one nozzle was connected to the hose attached to the pump. This test was performed in a water tank with the nozzle submerged, where the assembly was attached to a lever arm to measure a torque produced in order to calculate thrust from the water jet thrusters, as seen in Figure 21. The distance from the thruster to the fulcrum was twice that of the distance from the spring scale measurement to the fulcrum. Therefore, the thrust produced by the water jet thrusters was half the amount measured by the spring scale. Each size diameter nozzle reducers was tested. The data for this test is shown in Table 16.

Figure 21. Nozzle Test 1 (Single nozzle).

3.2.2.4 Nozzle Test 2

For the second test, two nozzles of the same diameter were connected to the hose attached to the pump. This test was performed in a water tank with the nozzles submerged, where the assembly was attached to a lever arm to measure a torque produced in order to calculate thrust from the water jet thrusters, as seen in Figure 22. Again the distance from the thruster to the fulcrum was twice that of the distance from the spring scale measurement to the fulcrum. Each set of different size diameter nozzle reducers were tested. The data for this test is shown in Table 17.

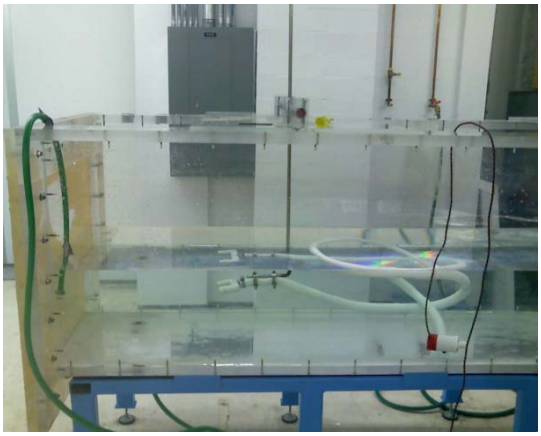


Figure 22. Nozzle Test 2 (Dual Nozzle).

3.2.2.5 Nozzle Test 3

For a third test, a 3/8 in nozzle was attached to the hose connected to the pump. This was held at a height of 8 in above the water surface as seen in Figure 23. The horizontal length of the arc of the water was measured. A measurement of 58 in was taken.

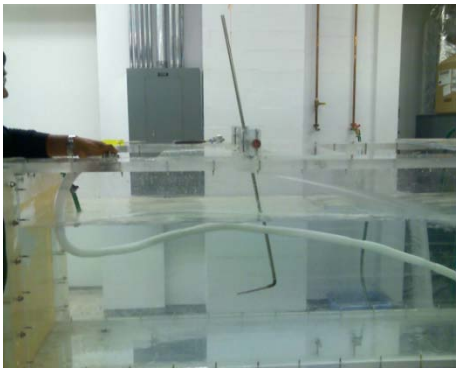


Figure 23. Nozzle Test 3.

3.2.2.6 Machined Nozzle Test 1

For this test, the machined nozzles were attached to a hose attached to a flow meter attached to a building water line. This test was performed in a water tank with one nozzle submerged, where the assembly was attached to a lever arm to measure a torque produced in order to calculate thrust from the water jet thrusters. The distance from the thruster to the fulcrum was the same as that of the distance from the spring scale measurement to the fulcrum. Therefore, the thrust produced by the water jet thrusters was the same amount measured by the spring scale. This test was performed with a water flow of 2 GPM. Each size diameter nozzle was tested. The data from this test is shown in Table 18.

3.2.2.7 Machined Nozzle Test 2

This test was the same as Machined Nozzle Test 1 except the flow of water was set to 5 GPM. The data from this test is shown in Table 19.

3.2.2.8 Results

Test 1 and 2 produced nominal results for the smaller nozzles. This was due to equations using an ideal moment, whereas the mass of the lever arm was not accounted for. From these results it is also seen that the water jet thrusters with this pump did have head losses due to the pump's inability to work at higher pressures. By utilizing basic projectile physics, test 3 was used to calculate the exit velocity from a 3/8 in nozzle. This resulted in a 23.7 ft/s exit velocity which shows a 50% head loss for this nozzle, or 1.5lbf of thrust produced by this nozzle. In the machined nozzle tests, nominal results were produced from the larger nozzles due to head losses but also from low flow. The smaller nozzles produced better results since this test allowed for work against back pressure, yielding almost 2lbf of thrust from the 3/16 in outlet nozzle. Therefore, the head losses were not due to backpressure but rather nozzle design. Future work can be performed in optimizing the nozzle design.

3.3 Ballast system

3.3.1 Tank Design

In order to keep the design of the ballast system simple while allowing the maximum maneuverability and control for our craft, a two tank ballast system was decided upon. Instead of a traditional main ballast tank with additional trim tanks, we would have two ballast tanks – one positioned at either end of the craft – to control the buoyancy and the trim of the craft.

To overcome the problem of having the ballast fluid move during attitude control and potentially destabilize the craft, a ballast tank with an expandable membrane enclosure was utilized to hold the fluid. Thus fluid entering the tank would be constrained within the membrane, limiting its motion within the tank.

The tanks were designed to be airtight with a constant volume. So as the tank filled, the air within the tank would get pressurized. This pressurized air was utilized to force the water out of the tank when needed. This eliminated the need to pump water out of the tanks, and also supplied a safety mechanism for the ballasts – in case of a power failure the air pressure could be utilized to force water out of the tanks via a Normally Open (NO) solenoid valve held closed during normal operation.

The ballast system was integrated into the directional thruster system to utilize only one water pump. Thus, the ballast tanks were designed to withhold the maximum pressures the pump was capable of generating – 45 psi. A study of pressure vessel properties indicated that 6” diameter, ¼” wall thickness Lexan tubing with flat end caps would serve as functional ballast tanks (Ref. Appendix G.). The water capacity of the ballast tanks at this pressure would be 72% of the total volume of the tank.

3.3.2 Building Ballast Tanks

For the main body of the ballasts, 6” diameter, ¼” wall thickness Lexan tubing was cut down to size on the vertical band-saw. A standard tire valve was inserted into the body to allow us to pre-charge the tank, ensuring that all water would be removed from the system. To allow the flow of water in and out of the tanks, a custom through wall connector was machined on a lathe. A compression o-ring seal on the inside ensured no

pressure losses/leaks took place through the walls of the tubing. As the membrane holding the liquid would experience no pressure differential across its membrane, the material strength was not of high consideration. Simple balloons secured via o-rings were utilized to form the membrane barrier for the fluid.

Initial ballast tank prototypes utilized a flat end-cap Lexan welded to the ends of the tubing and sealed externally with silicone sealant (Ref. Figure 24). The tank was then clamped together via a hose clamp to ensure that it would not come apart under high pressures. However, this setup lead to inherent sealing issues as the pressure differential was acting against the seal, causing depressurization of the tanks. Furthermore, at higher pressures the end caps would 'bow' around the hose clamp, furthering degrading the seal. This was due to the non-perfect Lexan bond and low bonding area between the end-cap and the tube body.

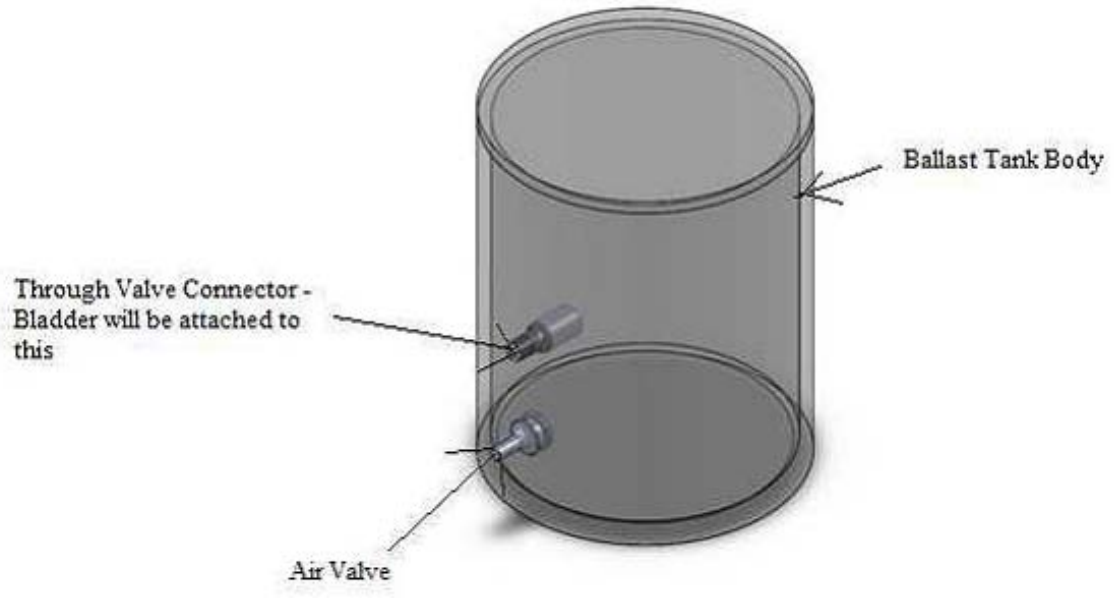


Figure 24. Initial Ballast Tank Design

The completed ballast tank can be seen in Figure 25 below. As can be observed, the final ballast system design utilizes a modified end cap design to inset part of the caps in the Lexan tubing. To seal the system, an internal silicone seal was utilized along with cyanoacrylate to fill any possible gaps. The ends of the caps were then Lexan welded to the tubing using methylene-chloride solution and Lexan dust as a filler. Dual hose clamps were utilized to overcome the bowing problem for the end caps. These ballast tanks were

pressure tested to ensure they could hold the required 45psi pressure for the system. A push-to-connect fitting was added to allow for easy connection to ballast system tubing.

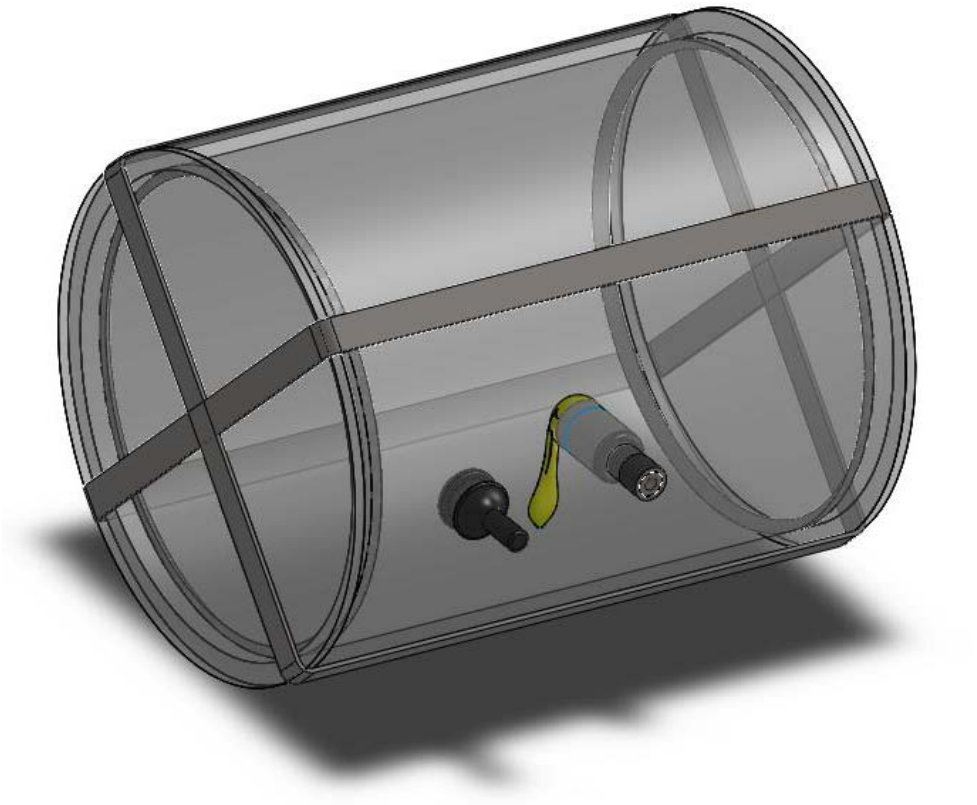


Figure 25: Final Ballast Tank

3.4 Integration of the ballast and thruster system

To reduce the power consumption of the craft, a plumbing circuit integrating both the ballast system, and the directional water jet thruster system was devised. Thus all the water based systems were powered by a central pump, for which the flow was rerouted via a multitude of solenoid valves. The ballast tanks were kept as independent pressure systems via the help of check valves. A complete schematic can be seen in Figure 26 below.

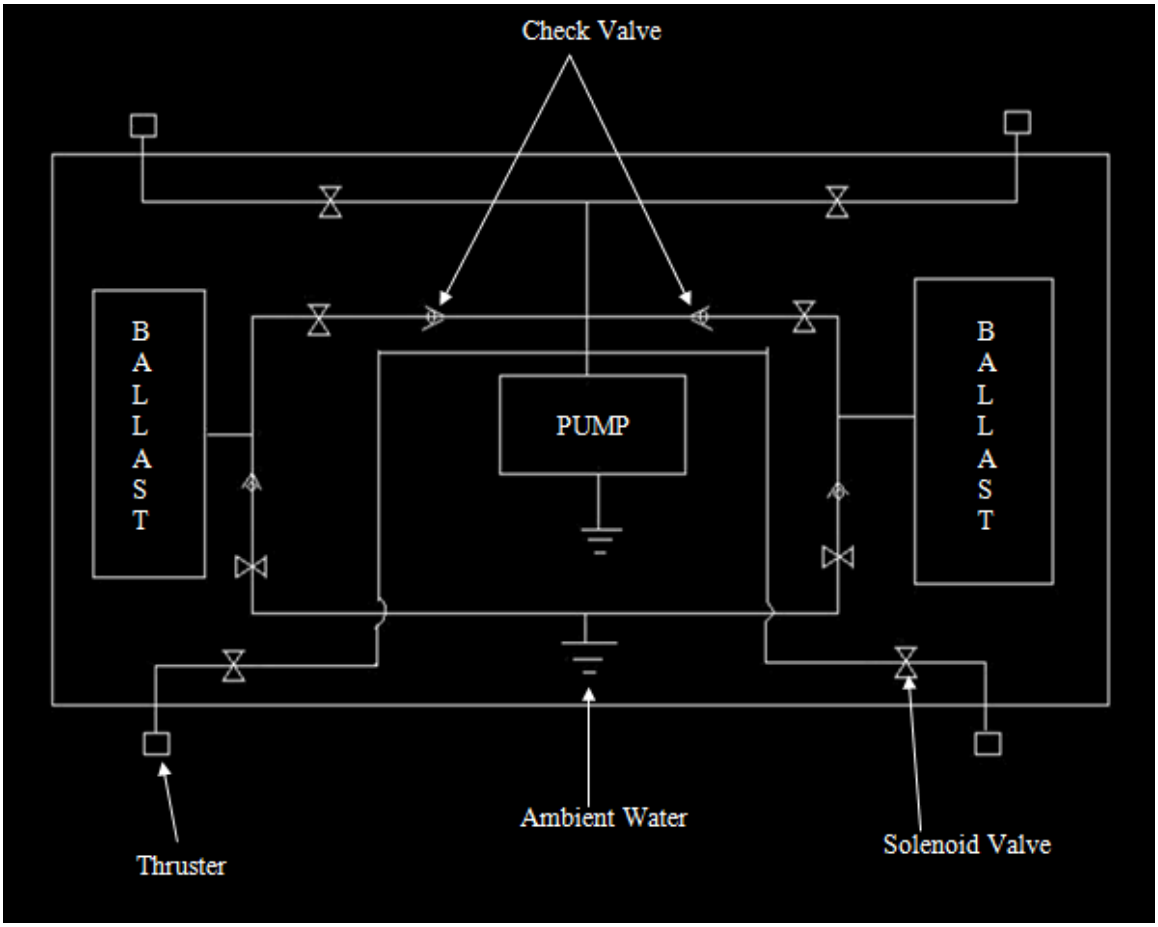


Figure 26. Integrated Plumbing Schematic.

3.5 Electronics

The vehicle electronics can be broken up into three main categories: Central Processing, Sensors and Actuators. The main function of the central processing unit is to implement control algorithms to interface readouts from the vehicle sensors and translate these into commands for the attitude and position control system through some sort of interface (the Sub-Hub system, described below). This is very simply described in Figure 27, below.

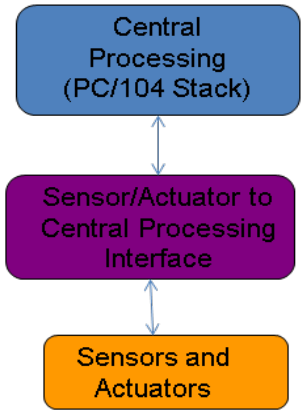


Figure 27. Electronics System Block Diagram.

The vehicle sensors include navigation units such as the 2-axis magnetic compass, as well as system health units like the cabin temperature sensor. The actuators (providing attitude and position control) consist of a series of solenoids and motors which direct flow for the water jet thrusters and ballast tanks, and drive the main propulsion thrusters.

3.5.1 Central Processing

The central processing unit includes both the central computer and a set of data acquisition devices which, together, allow both full processing of control algorithms, etc. as well as data acquisition from various sources such as acoustic imaging hardware or video cameras. Specifically, a frame grabber unit and an analog to digital data acquisition unit are installed in the central processing unit. These components follow either the PC/104 or PC/104 Plus embedded computer systems specification and form a "PC/104 Stack" which is described in more detail below. The development of these components focuses on software and interfacing. A block diagram of the central processing unit is included in Figure 28, below.

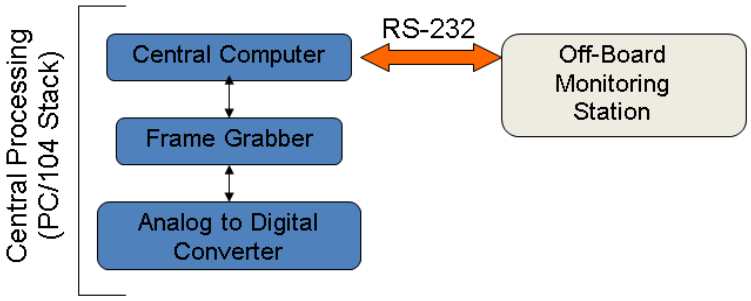


Figure 28. Central Processing (PC/104 Stack) Block Diagram.

3.5.1.1 PC/104 and PC/104 Plus Standards

The PC/104 standard (IEEE P996.1 Standard for Compact Embedded Modules), maintained by the PC/104 Consortium[16] defines both a self-stacking form factor and a communication protocol. PC/104 Plus, also maintained by the PC/104 Consortium, uses the same form factor as PC/104 with an expanded communication bus.

The PC/104 form factor defines a compact form factor of 3.6" by 3.8" with a height of 0.6" as shown in Figure 29. PC/104 devices communicate over an ISA bus. The 104 pin ISA bus connections are comprised of a 64 pin header and a 40 pin header. As mentioned above, PC/104 devices are made to be self stacking. This is accomplished through female bus connections on the top of the board and male bus connections on the bottom. PC/104 devices are available as both 8-bit and 16-bit modules.

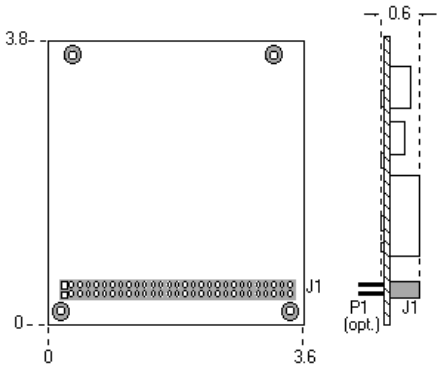


Figure 29. PC/104 Basic Mechanical Dimensions
© 2009 PC/104 Embedded PC Modules.[17]

PC/104 Plus devices include the full ISA bus connections of the PC/104 and an additional 120 pin PCI bus. The PC/104 Plus basic mechanical dimensions are shown in Figure 30. PC/104 devices can be interfaced with PC/104 Plus devices as PC/104 Plus devices include the 104 pin PC/104 ISA bus connections. As PC/104 devices do not include the 120 pin PCI connection used for PC/104 Plus, the PC/104 stack must be ordered such that any PC/104 Plus devices that are used go on the stack first followed by 16-bit PC/104 devices and finally 8-bit PC/104 devices.

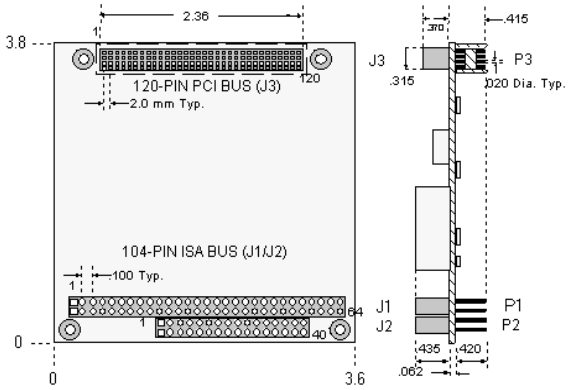


Figure 30. PC/104 Plus Basic Mechanical Dimensions
© 2009 PC/104 Embedded PC Modules.[18]

The PC/104 stack requires +5V, +12V and -12V supplies through an ATX connector. This supply is currently provided by a pico-PSU-120 DC-DC ATX power supply. This supply requires a 12V regulated DC input voltage and supplies 3.3V, 5V, +12V and -12V rails.

3.5.1.2 Computer

The VersaLogic Cheetah EPM-32c PC-104-Plus single board computer, see Figure 31. VersaLogic Cheetah EPM-32c., was chosen as the central computer. The EPM-32c features a 1.6 GHz Pentium M Processor. Currently, 1 GB of RAM and a 4 GB solid state hard drive are installed. The unit also features two USB 2.0 ports and two COM ports, one of which is RS-232/422/485 configurable [19]. The EPM-32c can support up to four PC/104 Plus and four PC/104 expansion modules.

Figure 31. VersaLogic Cheetah EPM-32c.

The computer is currently running Debian Linux 3.1, which was installed as part of the DEV-CD-L2-2.02 quick start package supplied by the manufacturer[20]. As of February 2, 2009 a new version of the development package, DEV-CD-L2-3.00 is available featuring Debian 5.0. To acquire either of these software packages a software request must be sent to VersaLogic. Upon acceptance, a download code will be issued allowing the software to be downloaded from the VersaLogic website.

A description of the setup of the computer and the operating system installation process is included in Appendix D.

3.5.1.3 Frame Grabber

Sensoray's Model 311 Frame Grabber, shown in Figure 32, is installed first on the PC-104 Stack (closest to the EPM-32c). This module features a PC/104 Plus compliant data bus and can support up to four cameras.

Figure 32. Sensoray Model 311 Frame Grabber with Sample Camera
© 2009 SENSORAY EMBEDDED ELECTRONICS.[21]

The Model 311 can transfer up to 30 frames per second in either color or monochrome. NTSC and PAL video formats are supported with 640x480 and 768x576 line resolution, respectively. This is accomplished through either four composite video inputs or three composite inputs and one S-Video. The supported frame rates for both formats are shown in Table 1. The Model 311 also provides eight TTL compatible digital I/O lines. Linux compatible drivers for the Model 311 Frame Grabber are available from the manufacturer but have not yet been installed and verified on the EPM-32c. The Model 311 requires 5V power.

Table 1. Sensoray Model 311 Frame Grabber Frame Rates.[22]

Number of cameras	NTSC per channel frames/sec	PAL per channel frames/sec
1	30	25
2	8	6
3	3	2
4	1	1

3.5.1.4 Analog to Digital Converter

In addition to the Frame Grabber, a VersaLogic VCM-DAS-1 "Data Acquisition and Control Module"[23] (see Figure 33) is installed second on the PC-104 stack. This card features a 16-bit, fully compliant PC/104 data bus.

Figure 33. VersaLogic VCM-DAS-1
Data Acquisition and Control Module
© 2004-2009 Versalogic Corporation.[24]

This DAS-1 has sixteen analog input channels and two analog output channels. The analog inputs are single ended with high impedance (listed as $>10^{10}\Omega$, 20pF) and 16-bit resolution. The analog inputs offer a jumper selectable range of ± 5V or ± 10V (all 16 channels are configured the same) with ±35 V over-voltage protection. The analog outputs offer 12-bit resolution and can source up to 5 mA each. The outputs are individually configurable for either 5V or 10V range. This board runs off of 5V +12V and -12V supplies. Damage can occur if all three supplies are not present.

The DAS-1 also features two 8-bit, TTL compatible digital I/O ports. The digital I/O are byte programmable as either input or output and can either source or sink 24 mA at a logic high of 2.4V (minimum) or logic low of 0.55V (maximum), respectively.

A description of DAS-1 operation along with example code is provided in Appendix D.

3.5.2 Sensors

The vehicle requires a number of drive components and sensors. The main sensors to be implemented in the current design include everything from system health and general sensors to navigation sensors. The sensors are listed below:

- External Pressure
- Ballast Pressure
- Cabin Temperature
- Central Processor Temperature
- Main Thruster Current
- 2- Axis Magnetic Compass
- 3-Axis Gyroscope/Accelerometer
- 3-Axis Sonar
- Water Leak Sensor (WLS)
- Supply Voltage Monitor (SVM)
- Hub Addressing Module

Provisions were also included for a pressure monitor for an emergency ballast system discharge cylinder should the current design for the ballast and stabilization thruster system need to be changed.

3.5.2.1 External Pressure and Ballast Pressure

Measurement Specialties MSI-1451 pressure sensors were chosen to determine the external pressure (allowing determination of depth and, when used in a fore and aft pair, pitch). The same series of sensor was also used for the ballast tank pressure, allowing indication of fill level, and the emergency discharge cylinder pressure monitor.

The MSI-1451 is a simple Wheatstone bridge based sensor with a supply voltage range of 3 to 12VDC. The sensor is offered in a variety of pressure ranges with the output

of the sensor swinging from 0-60mV across its specified pressure range allowing a single amplifier design to accommodate the sensors for all three applications. All sensors were chosen to sense absolute pressure so that changes in pressure inside of the hull due to deflection of the hull itself would not affect measurements. Sensors were also chosen with ported tubes to allow connection of hosing between the sensor and the location for the measurement.

A pressure range of 0-30 psia was chosen for the external pressure measurements. This was chosen to correspond with a maximum expected test depth of 2.4m in the WPI pool (an increase of about 3.5 psi over the surface) or a total maximum sensed pressure of about 18 psi. The 0-30 psia range allows for precise measurement with a 12-bit ADC (0.008 psi resolution) while still allowing a reasonable buffer to ensure that the sensor will not be harmed by dynamic pressures.

A pressure range of 0-100 psia was chosen to sense ballast tank pressure and determine ballast tank fill level. This was based off of the maximum backpressure of the pump chosen to supply water to the ballast tanks (45 psi as regulated by an automatic shut-off switch).

If an emergency ballast system discharge cylinder is ever required due to design changes, the same pressure sensor series can be used. Sensors in the range of 0-250 psia and 0-500 psia are available to monitor the charge pressure.

3.5.2.2 Cabin Temperature and Central Processor Temperature

The National Semiconductor LM-20 temperature sensor was used for cabin temperature and central processor temperature monitoring. This sensor was chosen for its simplicity and low power consumption. With a power supply operating range of 2.4V to 5.5V this sensor can be run off of either of the 3.3V or 5V supplies regulated by the ATX power supply. Its output voltage is nearly linearly related to the temperature and swings from 2.5V (corresponding to -55 C) to about 0.3V (corresponding to 130 C) and thus requires no output signal conditioning. It can also source up to 10 mA and therefore is well suited for remote applications (i.e. off-board temperature monitoring applications such as central processor temperature monitoring).

3.5.2.3 Main Thruster Current

The main thruster current sensing is actually provided for by the LMD18200T H-Bridge DC motor drivers used to drive the thruster motors. These chips use an array of DMOS power transistors to deliver up to 3A to the motors that they are driving. A small portion of the individual cells that make up the DMOS transistors are used to send a scaled down representation of the current through the whole transistor. This current can then be run through a resistor to convert the signal to a voltage which can be read by an ADC.

3.5.2.4 2-Axis Magnetic Compass

The on-board magnetic compassing (used to provide heading data) is based off of Honeywell's HMC1052 2-Axis magnetometer. This sensor consists of two Wheatstone bridges (one for each axis) and therefore only needs simple signal conditioning (similar to that of the pressure sensors). The magnetometer also requires the implementation of a set/reset circuit which is used periodically to realign the magnetic domains of the bridge sensors during operation to increase performance. A simple compassing circuit will provide better than 3 degree resolution.

3.5.2.5 3-Axis Gyroscope/Accelerometer

The Analog Devices ADIS16354 3-Axis Gyroscope and Accelerometer is to be used to detect X, Y, and Z-axis acceleration as well as X, Y, and Z-axis angular rates (pitch, roll and yaw). This unit has an on-board 12-bit ADC and sends its data digitally over a serial peripheral interface (SPI) bus. It includes a number of additional features which are not currently in use such as a Digital to Analog Converter (DAC) and general purpose digital I/O.

3.5.2.6 3-Axis Sonar

A 3-Axis sonar system was developed specifically for use on the vehicle. It uses transducers from AIRMAR's P23 Fishfinder. The sonar unit is controlled by a Texas Instruments MSP430 microcontroller and sends its data digitally over an SPI bus.

3.5.2.7 Water Leak Sensor

The water leak sensor was also developed specifically for this vehicle. This sensor relies on a moisture dependant change in the dielectric constant of an absorptive material (general automotive chamois cloth) sandwiched between two wire mesh plates. The sensor is supplied with a waveform generated by a Texas Instruments TLC556-timer. This functions the same as a general LM555 timer but is better suited for low supply voltages (down to 2V). The output of the 556-timer passes through a high pass filter creating a detectable signal on the ADC input channel. Because of the simple nature of the sensing element itself, it can be cut to and shape and size and installed around any seals or low points in the vehicle, allowing quick detection of hazardous leaks.

Initial tests of the water leak sensor element showed impressive results. A 4" by 6" element placed vertically in ¼" of water (with a 4" edge facing downward, in the water) showed a capacitive increase from 0.241 nF (dry) to 20 nF. The water leak sensor is, however, very sensitive to orientation in that any bending or force on the wire mesh plates causes a significant change in its capacitance.

3.5.2.8 Supply Voltage Monitor

The supply voltage monitor watches the voltage supplied by the battery pack and issues a warning when that voltage begins to near a level too low for continued operation. This is simply made up of two resistors forming a voltage divider to scale down the supply voltage and compare it to an internal reference.

3.5.2.9 Hub Addressing

The hub-addressing "sensor" is really just a jumper configurable switch which allows a 4-bit address to be mechanically selected on a pc-board and read into software so that software specific to that address can be loaded. This, of course, means that up to 15 unique hubs can be used. The sub-hub system is described in detail in a later section.

3.5.3 Actuators

The actuators listed below control flow through the stabilization water jet thrusters, fill and purge operations for the ballast tanks and main propulsion:

- Stabilization Thruster Control Solenoids

- Ballast Tank System Solenoids
- Main Propulsion Thruster Drivers
- Pump Switch Relay Driver
- Pump Routing Control Solenoids

Provisions were included for extra solenoid controls as well. These are labeled as ballast solenoids in the schematics but can be used to drive any of the solenoids.

3.5.3.1 Stabilization Thruster Control Solenoids

The water-jet stabilization thrusters are controlled by individual solenoid valves. These valves provide binary control over the thrusters (on/off). STC Valve's 2P160-1/2 were chosen for the job. These valves offer ½" connections and 16mm (0.625") orifice size. The solenoids were chosen based on their low-cost and large orifice size which allows a minimal pressure drop across the valve when open. They also feature a response time of less than 20ms which is sufficient for all required switching operations.

3.5.3.2 Ballast Tank System Solenoids

Solenoid valves were also used to control and direct flow through the ballast system. STC Valve's 2V015 1/4, quarter inch, 12 V, 2-way, direct acting, normally closed solenoids were chosen to perform this function. These valves also feature a maximum response time of less than 20ms. All ballast hosing was chosen to be ¼" as there is no express need for fast filling or purging of the tanks unless there is an emergency. In fact, slower filling and purging is desired as the vehicle's depth is very sensitive to small fluctuations of weight when at or around neutral buoyancy. Even with the smaller hose lines the tanks can be completely filled or purged in a matter of seconds. This was considered to be reasonable even in the event of an emergency. In the case of an emergency where catastrophic failure will occur within the few seconds that it takes for the ballast to purge and the vehicle to surface this failure would probably not be preventable by such an action so larger hoses and valves would not improve performance.

3.5.3.3 Main Propulsion Thruster Drivers

National Semiconductor's LMD18200T H-Bridge DC motor drivers are used to power and control the 12V DC motors used in the external propeller driven thrusters. These drivers require a pulse width modulated (PWM) input signal to control variable speed of the thrusters as well as a logic level direction input to reverse propulsion and a logic level brake input to stop the motors. These drivers also provide a current output which is proportional to the current draw through the motor (allowing simple current sensing).

3.5.3.4 Pump Switch Relay Driver

A 4 amp pump switch relay driver was also included to allow use of a different pump for the integrated thruster/ballast system. The current pump being used features a pressure activated automatic shut-off. The pump switch relay driver provides a provision for control of a pump without an automatic shut off or bypass of such an automatic shut-off.

3.5.3.5 Pump Routing Solenoid Controls

Pump routing solenoid controls were included to allow the installation of extra solenoids on the inlet and outlet of the pump. This would allow the pump to be used to actively discharge the ballast tanks by routing the pump inlet to the ballast tanks rather than the water external to the hull. This would also allow water routing through a water cooled radiator.

3.5.4 Sub-Hub System

A number of possible solutions were considered to handle the vehicle electronics. The main design choice was between a single, central printed circuit board (PCB) servicing all of the vehicle needs or a series of PCB's spread through the interior of the hull and serving all sensors and drive components locally. The latter would require a data bus to interface the individual boards. Regardless of this choice a serial bus was to be used for communication between the drive and sensor components, the central computer and an on-shore monitor.

All sensors and drive components could have been handled by a single board and the small size of the vehicle does not necessitate local handling of any sensors (the farthest sensors are still located within relatively close proximity to the center of the hull). However, to allow more efficient operation as well as more room for configuration and possible redesign of electronic systems, a multiple hub system was chosen.

A single PCB design which could be either fully or partially populated and equipped with an addressing scheme to allow hub-specific software to be loaded and run on startup was created to comprise the sub-hub system. The current design uses three hubs (central, fore and aft) to handle all sensors and drive controls and spreads them throughout the hull. This system is illustrated by a block diagram in Figure 34, below.

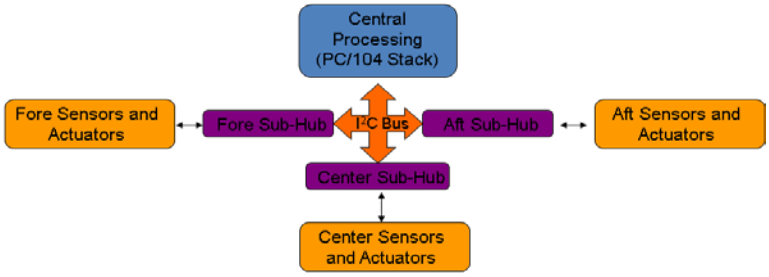


Figure 34. Sub-Hub System Block Diagram.

At the center of each hub is an MSP430F233 microcontroller which facilitates data processing and control of all sensors and drive components as well as communication with other hubs, the PC-104 stack and an on-shore monitor. The MSP is also meant to take on some of the software load, simplifying controls from the PC-104 stack and monitor computer.

The hub circuitry consists of the sensors and control devices with their required signal conditioning, I^2C and SPI serial buses, the required circuitry associated with the microcontroller (i.e. crystal oscillators and JTAG interface) and the power connections. The specific circuits included on each hub and the quantities of each type of circuit are listed below:

- 4-Bit Hub Addressing, 1
- 2-axis magnetic compass, 1
- Magnetic compass Set/Reset, 1
- Solenoid driver, 15

- Emergency ballast solenoid driver, 1
- Pump Driver, 1
- Main thruster motor driver, 2
- On-board temperature sensor (Cabin Temperature), 1
- Remote temperature sensor (PC104 Temperature), 1
- 1.25V reference generator, 1
- Pressure sensor, 3
- Supply Voltage Monitor, 1
- Water leak sensor, 1
- I^2C bus lines and pull-up resistors, 1
- SPI bus lines, 1
- MSP430 circuitry
    - High frequency crystal oscillator (for MSP430), 1
    - Low frequency crystal oscillator (for MSP430), 1
    - JTAG programming interface (for MSP430), 1
- Power connections, 1x3.3V, 1x5V, 4x12V

The suggested use of components (for the three respective hubs) is outlined in Figure 35, below.

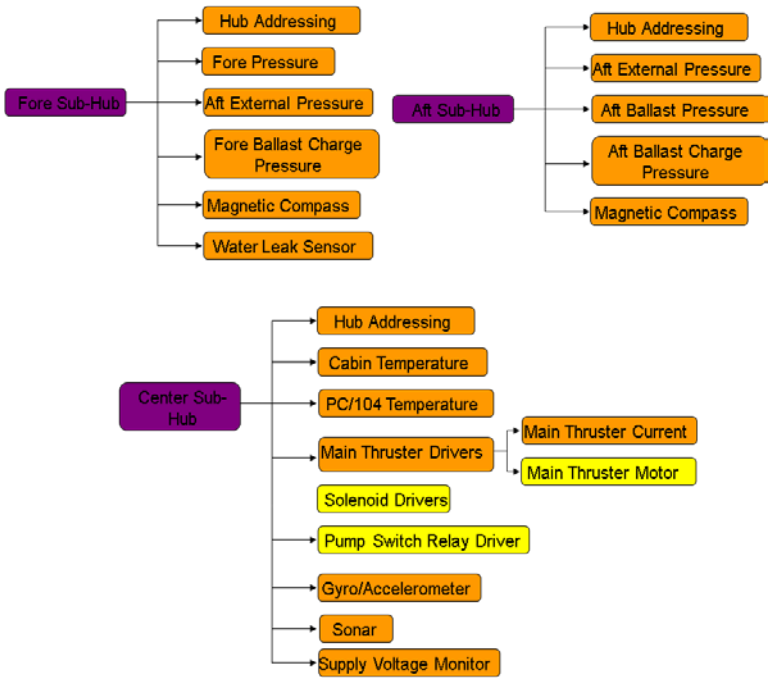


Figure 35. Fore, Aft and Center Sub-Hub Layout Block Diagram.

The majority of these components can operate simultaneously. There are a few, however, which share I/O pins on the MSP430 and must be manually selected with jumpers via three pin headers (i.e. tying the left pin to the center selects one peripheral signal while tying the right pin to the center selects the other). These headers are shown in the FUNCTION_SEL circuit schematic.

The magnetic compass's x-axis shares a pin with the cabin temperature sensor. The magnetic compass's y-axis shares a pin with the PC104 temperature sensor. The water leak sensor and supply voltage monitor also share an input pin. These peripherals, therefore must be selected via jumpers J19, J20 and J21, respectively.

Each board has connections for 3.3V, 5V and 12V power supplies.

3.5.4.1 Four-Bit Hub Addressing Circuit

The 4-bit hub addressing circuit, shown in Figure 36, connects the signals HUBADDRx to the +3.3V supply through four 10kΩ pull-up resistors. The four signals area also connected to the ground plane through four pairs of two pin headers (realized by a single 4x2 pin header, J17). The MSP430 pin connections for these four signal lines are shown in Table 2, below.

Table 2. Hub Addressing Connections.

Signal	MSP430 Pin	MSP430 Port	4x2 Header Pin Signal Side	4x3 Header Pin Ground Side
HUBADDR3	37	P4.1	1	2
HUBADDR2	36	P4.0	3	4
HUBADDR1	35	P3.7	5	6
HUBADDR0	34	P3.6	7	8

When a given signals jumper is installed (i.e. header pins 1 and 2 are connected, using HUBADDR3 as an example) the signal is shorted to ground, and therefore registers as a logic low at the MSP430's input. When the jumper is removed the signal is connected to the 3.3V supply through the 10kΩ resistor. Due to the high impedance inputs of the MSP430's digital I/O pins, negligible current flows through the resistor and a 3.3V logic high is seen at the MSP430's input.

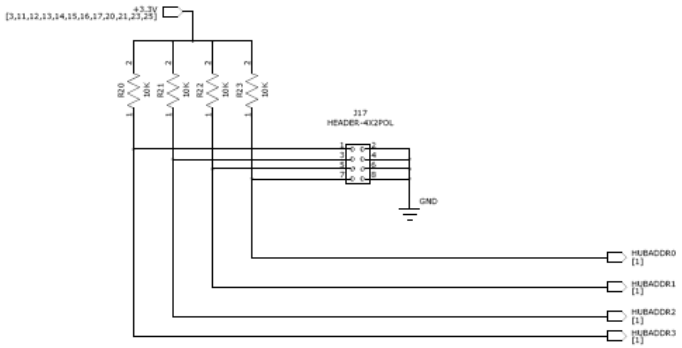


Figure 36. Hub Addressing Circuit.

3.5.4.2 Two-Axis Magnetic Compass Circuit

The magnetic compass circuit, Figure 37, uses Honeywell's HMC1052 2-Axis Magnetic Sensor with some signal conditioning to send two channels of analog data to the MSP430's ADC. Each axis is run through an inverting amplifier made with TI's OPA358 operational amplifier. This op-amp is well suited for use with a single 3.3V supply rail. The op-amps are given a DC offset using the 1.25V reference voltage (described later) allowing the magnetic sensors outputs to make efficient use of the 2.5V range of the ADC.

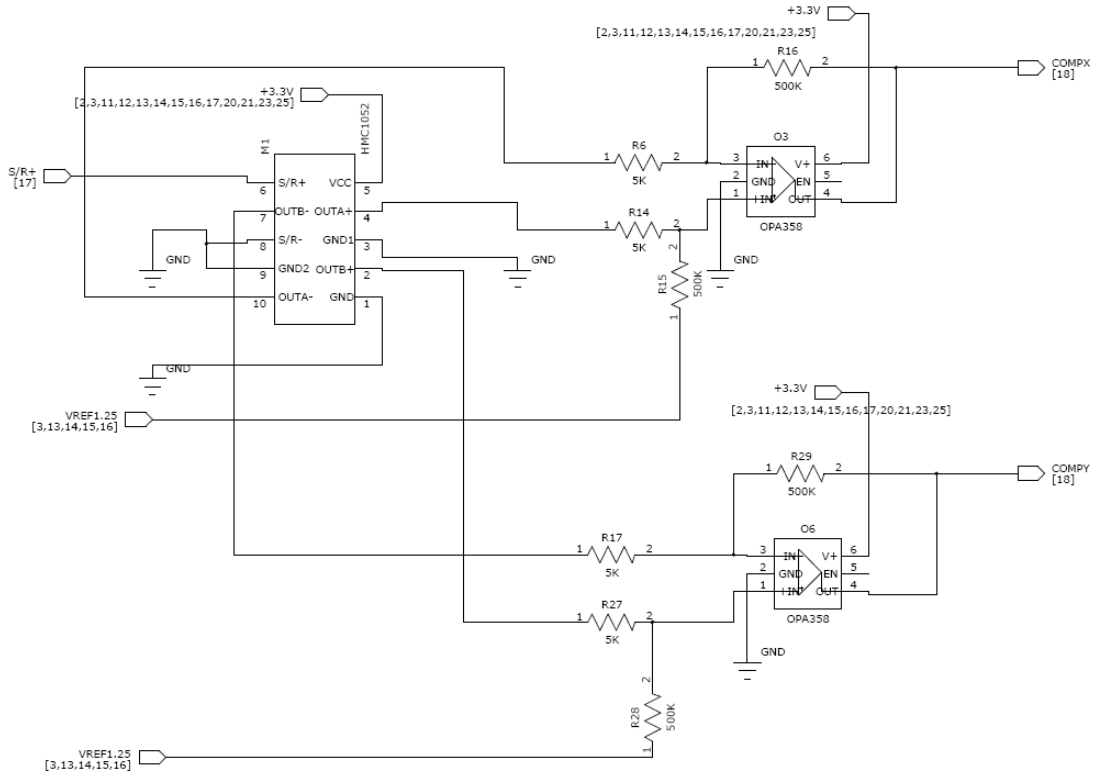


Figure 37. Magnetic Compass Circuit.

The magnetic compass circuit signal outputs: COMPX and COMPY, share pins with the two temperature sensor outputs sent by the cabin temperature sensor and the PC-104 temperature sensor: CABINTEMP and PC104TEMP, respectively. The signals are jumper selectable via two 3-pin headers: J19 and J20. Only one signal from each pair can be used at a time on each board. The jumper positions for function selection are labeled as COX, CAB, COY and PCT for COMPX, CABINTEMP, COMPY and PC104TEMP,

respectively. The MSP430 pin connections for these signals (after the 3 pin jumpers) are shown in Table 3, below.

Table 3. Magnetic Compass Connections.

Signal	MSP430 Pin	MSP430 Port	Function Selection Jumper	Signal 1	Signal 2
TEMP/COMPASS_SELECT_1	59	P6.0	J19	COMPX	CABTEMP
TEMP/COMPASS_SELECT_2	60	P6.1	J20	COMPY	PC104TEMP

3.5.4.3 Magnetic Compass Set/Reset Circuit

The compass set/reset circuit (Figure 38) sends a pulse through a magnetic coil perpendicular to the sensitive axis of the compass periodically to "condition the magnetic domains" of the sensing elements for better performance[25].

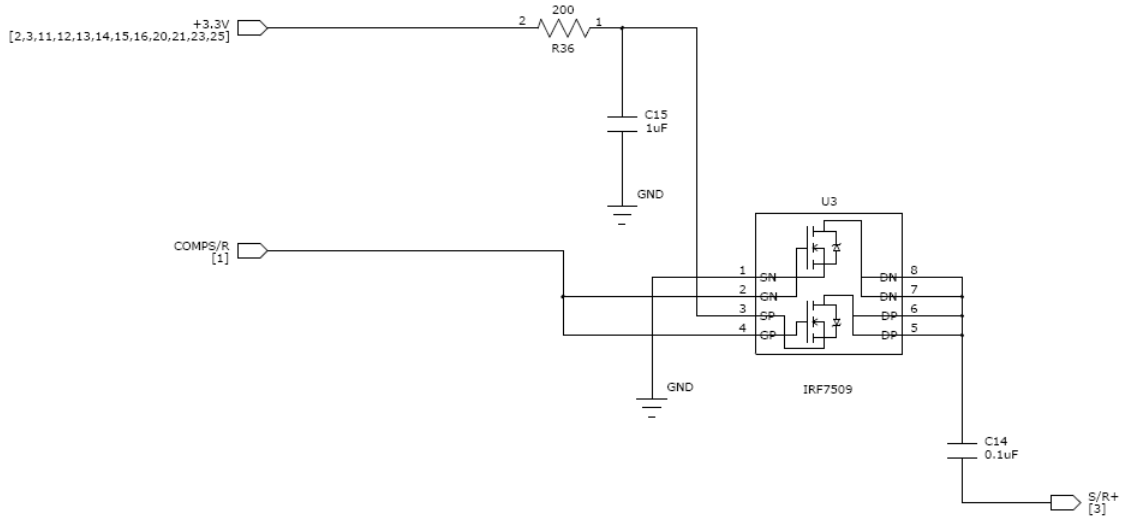


Figure 38. Compass Set/Reset Circuit.

The circuit uses an IRF7509 CMOS transistor pair to send a short pulse through the set/reset coils when the 3.3V logic input COMPS/R (sent from the MSP430) goes high. The COMPS/R signal is connected to MSP430 pin 51 (P5.7).

3.5.4.4 Solenoid and Emergency Ballast Solenoid Driver Circuits

The solenoid driver circuits, shown in Figure 39, use an ST Microelectronics STD17NF03L power NMOS transistor. The gate of the transistor is connected directly to an output pin from the MSP430 (signal STABx, BSOLx or PRSOLx) as shown in Table 4. The minimum logic high voltage of the MSP430 (2.6V) will allow a drain to source current (I_{ds}) of up to 3.10A for a drain to source voltage (V_{ds}) of 12V. This is more than sufficient to drive the ballast system solenoids which draw a maximum of 0.25A for pressures up to 60 psi as well as the water jet thruster solenoids which draw a maximum of 1A for pressures up to 60 psi. A fly-back diode is connected across the solenoid connection header to prevent damage to the MOSFET when the solenoid is switched off.

Table 4. Solenoid Driver Connections

Signal	Header	MSP430 Pin	MSP430 Port
STAB1	J1	44	P5.0
STAB2	J2	45	P5.1
STAB3	J3	46	P5.2
STAB4	J4	47	P5.3
BSOL1	J5	12	P1.0
BSOL2	J6	13	P1.1
BSOL3	J7	14	P1.2
BSOL4	J8	15	P1.3
BSOL5	J9	16	P1.4
BSOL6	J10	17	P1.5
BSOL7	J11	18	P1.6
BSOL8	J12	19	P1.7
BSOL9	J13	20	P2.0
PRSOL1	J14	21	P2.1
PRSOL2	J15	22	P2.2
EBSOL	J16	49	P5.5

It should be noted that this current design does not utilize a pull-down resistor on the gate signal. As the MSP430 will actively pull the line low there is no need to dissipate stray voltages on the signal line. The MSP430 also features a software selectable pull-down resistor which can be used if this component is deemed necessary. Without the use of a pull-up resistor the board's 12V supplies should be connected after the 3.3V supplies to allow the MSP430 to initialize its outputs. If this is not done some solenoids may operate shortly after start-up, while the MSP430 is initializing.

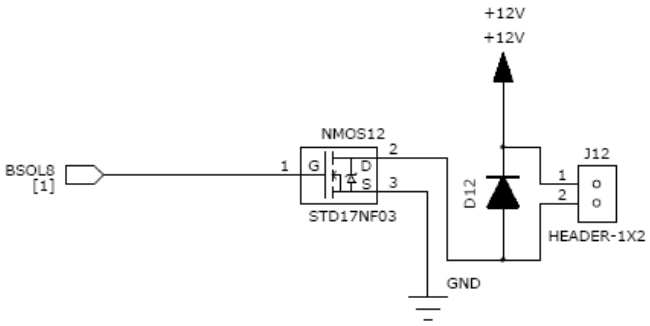


Figure 39. Solenoid Driver Circuit.

The emergency ballast solenoid driver (see Figure 29) is slightly different. It uses an IRF7509 CMOS power transistor to charge a 10,000 uF capacitor while the MSP430 applies a logic high to its gate. In the event of a recognized failure or power loss, the EBSOL signal will switch to a logic low and the IRF7509 will discharge the capacitor through the solenoid connected to J14. A small pull-down resistor (R45) is connected to the EBSOL signal to ensure that no stray voltages occur in the line preventing correct operation of the IRF7509. This circuit allows a latching valve to be used as an emergency purge valve in an attempt to reduce power consumption.

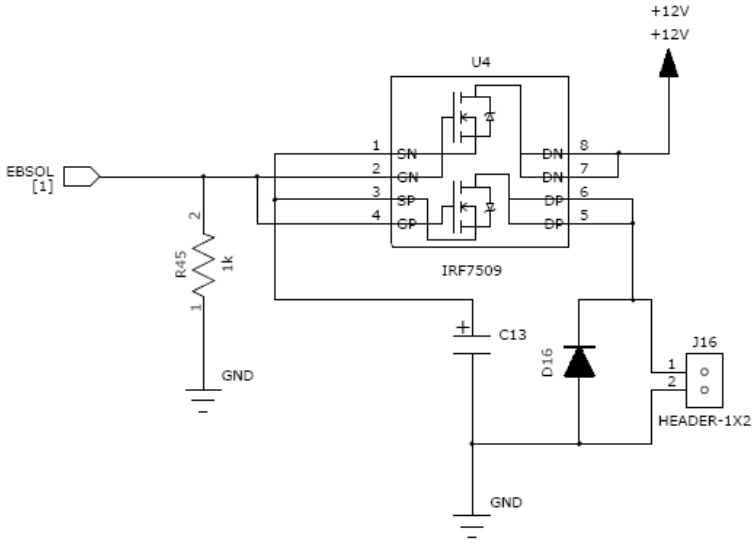


Figure 40. Emergency Ballast Solenoid Driver Circuit.

Currently the vehicle is using a normally open solenoid valve (run by a normal solenoid driver circuit), which is forced shut by a signal from the MSP430. In the event of a failure or power loss the absence of a logic high from the MSP430 will turn off the valve allowing it to re-open and purge the ballast tanks.

3.5.4.5 Pump Driver Circuit

The pump currently used for the integrated ballast and thruster system has an automatic shut-off switch that activates at a pressure of 45 psi. In the event that test results or a design change necessitate active control over the pump, a manual pump driver circuit was included on the circuit board (see Figure 41). Because of the high current needs of the pump (a maximum of 13 amps), driving the pump directly off of the PCB would have required trace widths in excess of 200 mils[26]. In order to run the pump off of the STD17NF03L MOSFET, the 3.3V signal from the MSP430 would also have needed to be stepped up significantly to meet the current requirements. It was decided that supplying a signal to control a relay which would, in turn, switch the pump was a more reasonable and efficient solution. The logic high gate voltage level was still stepped up slightly to 5 V to ensure that the transistor could source up to 4 amps comfortably. The pump driver circuit uses a Texas Instruments OPA358 configured as a comparator. The comparator checks the PUMP_DRIVE signal against the MSP430's 2.5V reference voltage. The output of the comparator feeds an STD17NF03L MOSFET, with a 1kΩ pull-down resistor (to ensure that any stray voltages on the signal do not activate the MOSFET. The output of the MOSFET is to be connected to the off-board relay through header J18.

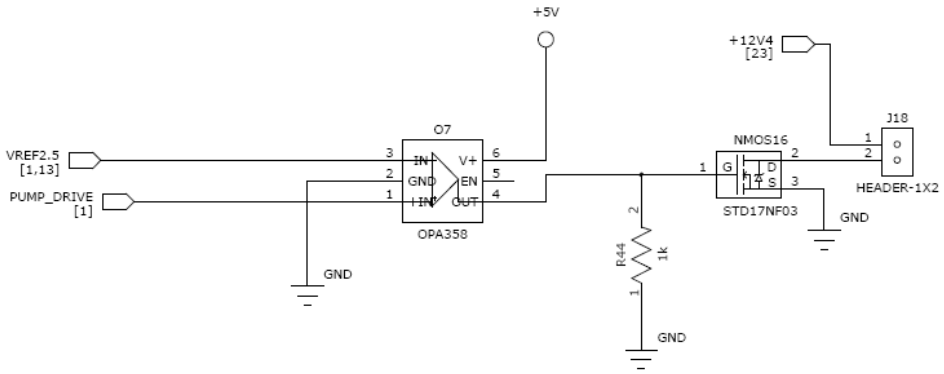


Figure 41. Pump Driver Circuit.

3.5.4.6 Main Thruster Driver and Current Sensing Circuit

The two main thruster drivers and current sensing circuits use a National Semiconductor LMD18200T H-Bridge DC motor driver (see Figure 42) controlled by inputs from the MSP430. The pulse width modulated MTxPWM controls the motors duty cycle, the 3.3V logic level MTxDIR controls the direction and the 3.3 logic level MTxBRK actively slows and stops the motor. The connections to the MSP430 are shown in Table 5, below. The motor is connected through a 2-pin header (J25 and J24 for Main Thruster Drivers 1 and 2, respectively).

Table 5. Main Thruster Driver and Current Sensing Connections.

Signal	MSP430 Pin	MSP430 Port
MT1PWM	38	P4.2
MT1DIR	39	P4.3
MT1BRK	40	P4.4
MT2PWM	41	P4.5
MT2DIR	42	P4.6
MT2BRK	43	P4.7
CURSENSE1	4	P6.5
CURSENSE2	5	P6.6

For faster switching applications above 1kHz[27], two bootstrapping capacitors are required for efficient operation of the charge pump which ensures that the output voltage

stays above 10V. These capacitor values are labeled as DNP (Do Not Populate) as they are unnecessary for current designs.

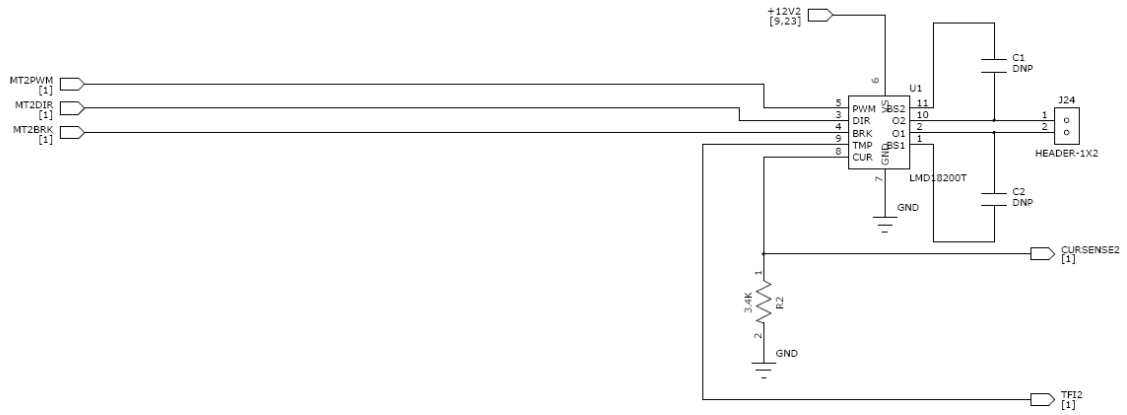


Figure 42. Main Thruster Driver and Current Sensing Circuit.

Also included in the circuit is a resistor which converts a scaled version of the output current (CURSENSEx), provided by the current sensing output of the LMD18200T, to a voltage and sends it to the MSP430's ADC.

3.5.4.7 Cabin Temperature and PC104 Temperature Sensor Circuit

The cabin temperature sensor circuit, shown in Figure 43Figure 29, is relatively straightforward. It uses a National Semiconductor LM20 temperature sensors which requires a 3.3V power supply and sends a temperature dependant voltage signal that is already within the 0 to 2.5V range of the MSP430's ADC. As suggested in the LM20 manual a low pass filter is built on the LM20's output signal to reduce noise. This is not necessary for cabin temperature sensing as the sensor is located on-board in a relatively low noise environment but the extra components are included to improve performance. Example code to read the cabin temperature sensor's voltage output through the MSP430's ADC is given in Appendix A.

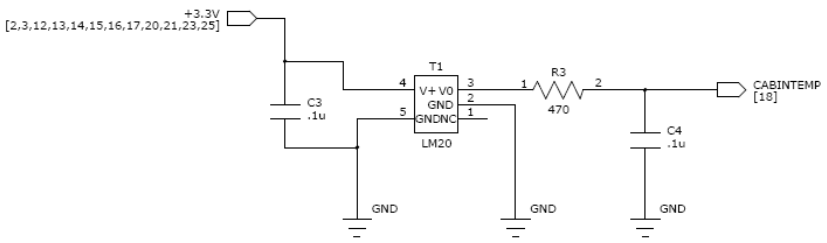


Figure 43. Cabin Temperature Sensor Circuit.

As the LM20 output can source up to 10mA and is capable of driving high capacitive loads it is also well suited for remote applications. Therefore the same circuit was used for the PC104 Temperature sensor with the exception that the sensor itself was replaced by a 3-pin header (see Figure 29).

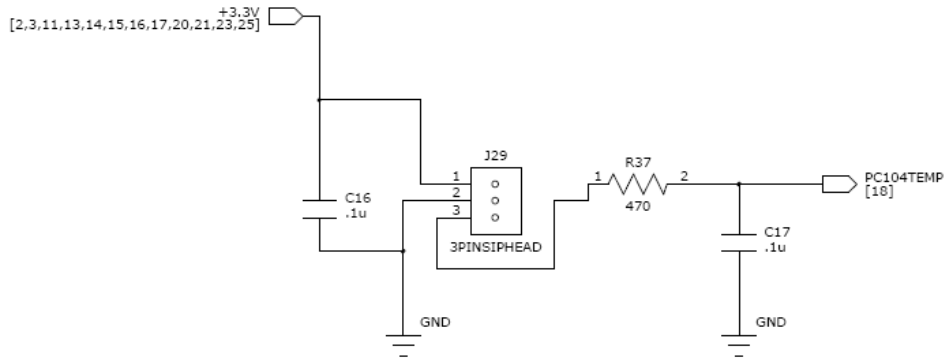


Figure 44. PC104 Temperature Sensor Circuit.

Using this circuit, the sensor can be located remotely (on the PC104) and return its signal to the MSP430. The PC104 temperature sensor connections are shown in Table 6, below.

Table 6. PC104 Temperature Sensor Circuit Connections.

Header Pin	Signal
1	3.3V Power
2	Ground
3	Output

The cabin and PC104 temperature sensor outputs share pins with the magnetic compass outputs. This is explained in detail in section 3.5.4.2 Two-Axis Magnetic Compass Circuit.

3.5.4.8 1.25 Volt Reference Generator Circuit

The 1.25 volt reference generator circuit (Figure 45) uses a Texas Instruments OPA358 operational amplifier configured as a unity gain buffer. The op-amp's input takes a ½ voltage divider which cuts the MSP430's 2.5V reference down to 1.25V.

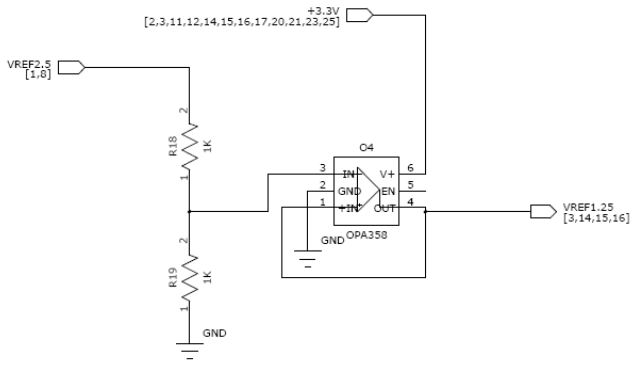


Figure 45. 1.25 Volt Reference Generator Circuit.

3.5.4.9 Pressure Sensor Circuit

The pressure sensor circuit uses the MSI-1451 Wheatstone bridge pressure sensor (Figure 46). The differential outputs of the pressure sensor are amplified by an inverting amplifier. The pressure sensor requires a 3.3V power source.

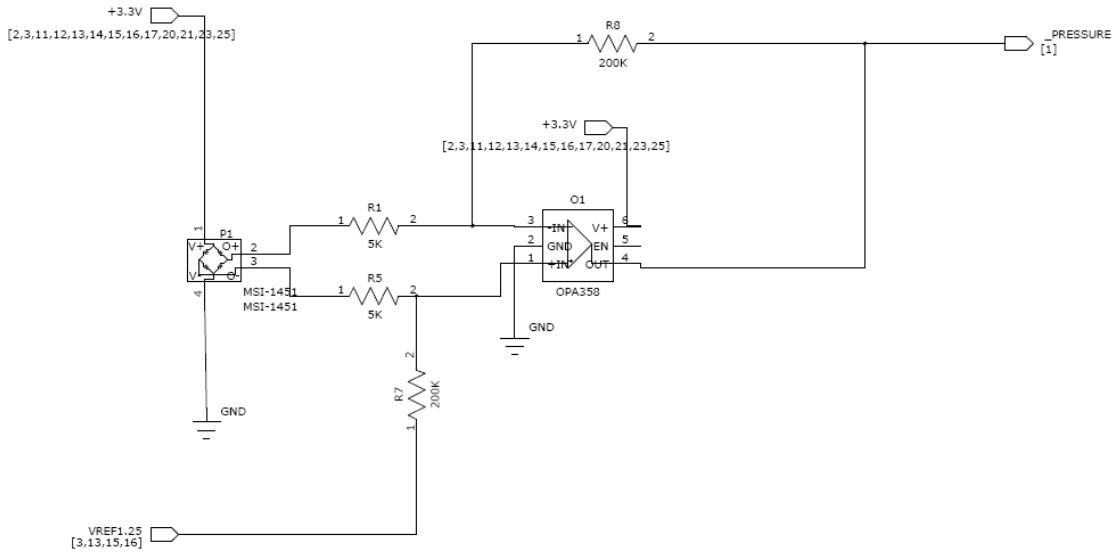


Figure 46. Pressure Sensor Circuit.

The same circuit is used for all three of the pressure sensors that can be populated on each board. The signal connections are shown in Table 7.

Table 7. Pressure Sensor Connections.

Sensor	Signal	MSP430 Pin	MSP430 Port
P1 (Depth Pressure)	DEPTH_PRESSURE	61	P6.2
P2 (Ballast Pressure)	BALLAST_PRESSURE	2	P6.3
P3 (Emergency Ballast Charge Pressure)	EMERGENCY_PRESSURE	3	P6.4

3.5.4.10 Supply Voltage Monitor Circuit

The MSP430 is meant to be able to take a scaled down version of the supply voltage and internally compare it to its 1.1V reference. The supply voltage monitor circuit makes use of this function with a simple voltage divider as shown in Figure 47. The scaled down supply voltage signal SVM is jumper selectable as described in Appendix A. .

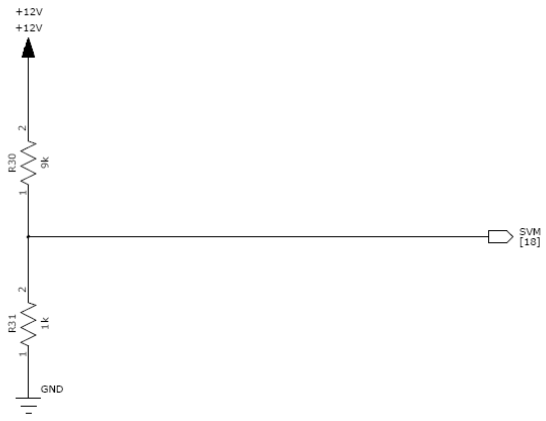


Figure 47. Supply Voltage Monitor Circuit.

3.5.4.11 Water-Leak Sensor Circuit

The water-leak sensor circuit uses a sensing element developed specifically for the vehicle (see Figure 48).

The sensing element is used as the capacitive portion of an astable monovibrator circuit built using a Texas Instruments TLC556 timer. This functions the same as a standard LM555 timer but is suited for lower supply voltages. A Microchip MCP41050 digital potentiometer is used in conjunction with the capacitive water-leak sensor to calibrate the output frequency of the TLC556. The digital potentiometer is set using the SPI bus connections from the MSP430 and is selected with signal CS4. The MSP430 connections are shown in , below. The output of the water leak sensor is jumper selectable as described in 0. More testing is required to determine correct values for the passive components in the water-leak sensor circuit. As the size of the sensing element can change drastically depending on the number of seals in the finished hull and the extent to which water leak sensing is deemed to be necessary these values can cover a broad range. The digital potentiometer is meant to allow calibration in the event that additional water-leak sensing elements are added as well as to negate the effects of other parameters such as humidity changes. The chip select and SPI signal pin connections are described in Table 9.

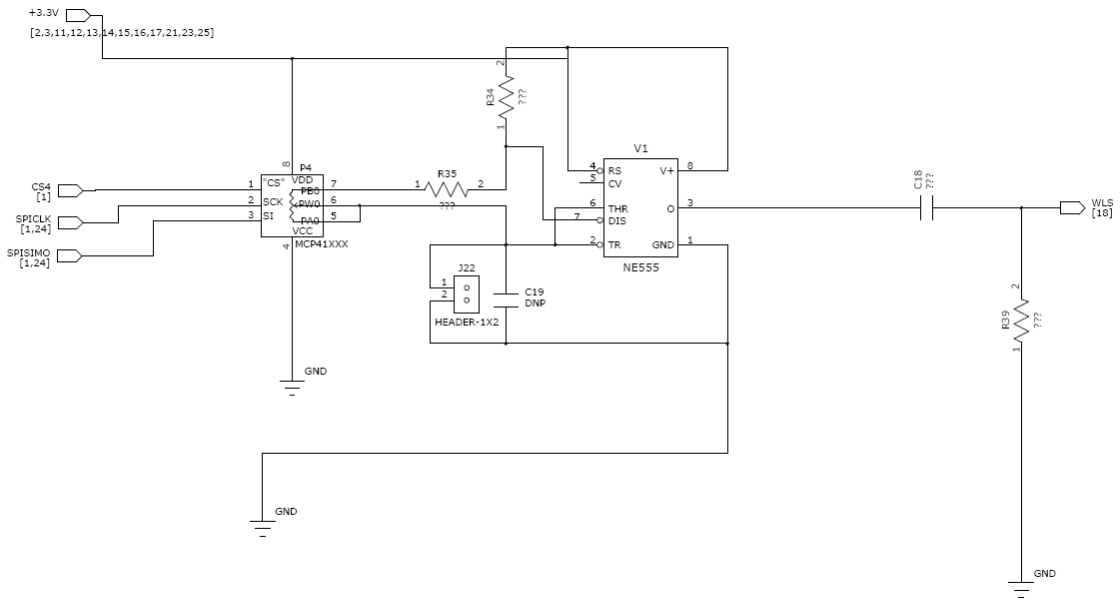


Figure 48. Water-Leak Sensor Circuit.

3.5.4.12 Inter-Integrated Circuit, I²C, Bus Circuit

The I^2C circuitry uses the MSP430's provided I^2C capabilities. A 2-pin header, J31, provides off-board connections for the I^2C bus. The I^2C bus connections are shown in Table 8.

Table 8. I^2C Bus Circuit.

Header Pin	**Signal**	**MSP430 Pin**	MSP430 Port
1	I2CSCL	30	P3.2
2	I2CSDA	29	P3.1

Pull-down resistors from the 3.3V supply are tied to the I^2C bus lines. The I^2C bus circuit is shown in Figure 49, below.

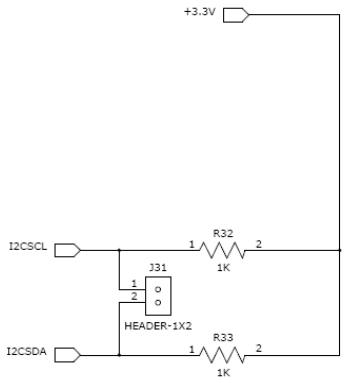


Figure 49. I^2C Bus Circuit.

3.5.4.13 Serial Peripheral Interface, SPI, Bus Circuit

The SPI bus connections are accessible through 24-pin header S29 as shown in Table 9. Three chip select outputs are provided allowing each board to handle up to three off-board SPI devices. Signal CS4 (Chip Select 4) is also provided and is used for components used in the water-leak sensor circuit. No off-board connections are provided for CS4. This bus is used for a number of components such as the 3-axis gyro/accelerometer and the sonar modules.

Table 9. SPI Connections.

Header Pin	Signal	MSP430 Pin	MSP430 Port
1	n/a	n/a	n/a
2	n/a	n/a	n/a
3	SPICLK	28	P3.0
4	SPISOMI	33	P3.5
5	SPISIMO	32	P3.4
6	SPISTE	31	P3.3
7	CS2	26	P2.6
8	CS1	25	P2.5
9	CS3	27	P2.7
10	5V	n/a	n/a
11	5V	n/a	n/a
12	5V	n/a	n/a
13	n/a	n/a	n/a
14	n/a	n/a	n/a
15	n/a	n/a	n/a
n/a	CS4	50	P5.6

3.5.4.14 MSP430 Circuitry

The crystal oscillator circuit provides precision clocking frequencies for the MSP430. A 32760 Hz quartz watch crystal is used for the low frequency crystal. A 4.0 MHz crystal is used for the high frequency crystal. Through software, these frequencies can be divided to a number of frequencies. The crystal oscillator circuits are shown below in Figure 50. Crystal Oscillator Circuit..

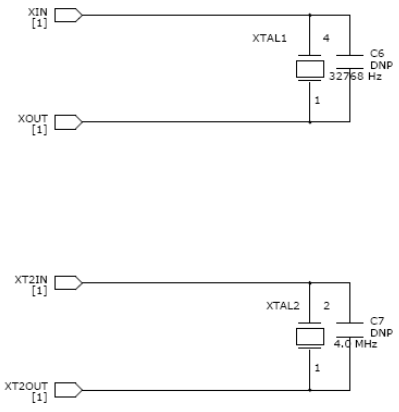


Figure 50. Crystal Oscillator Circuit.

The JTAG programming interface circuit is shown in Figure 51. This allows a standard programming tool like the MSP-FET430U64 to be used, along with software like IAR Embedded Workbench, to program the MSP430. This interface follows the standard suggestions provided by Texas Instruments in the MSP-FET430U User's Guide[28]. The JTAG circuit features a 3-pin jumper (J27) which allows the use of an external power supply for the MSP430 by connecting J27 pin 3 to pin 2. When pin 1 is connected to pin 2 the debugger voltage supply is used. For convenience, an additional 2-pin jumper was included between the board's 3.3V supply and the MSP430's AVCC. This allows programming and debugging of the MSP430 without power-up of the rest of the 3.3V components on the board. The ~RST/NMI pin is pulled high by a resistor.

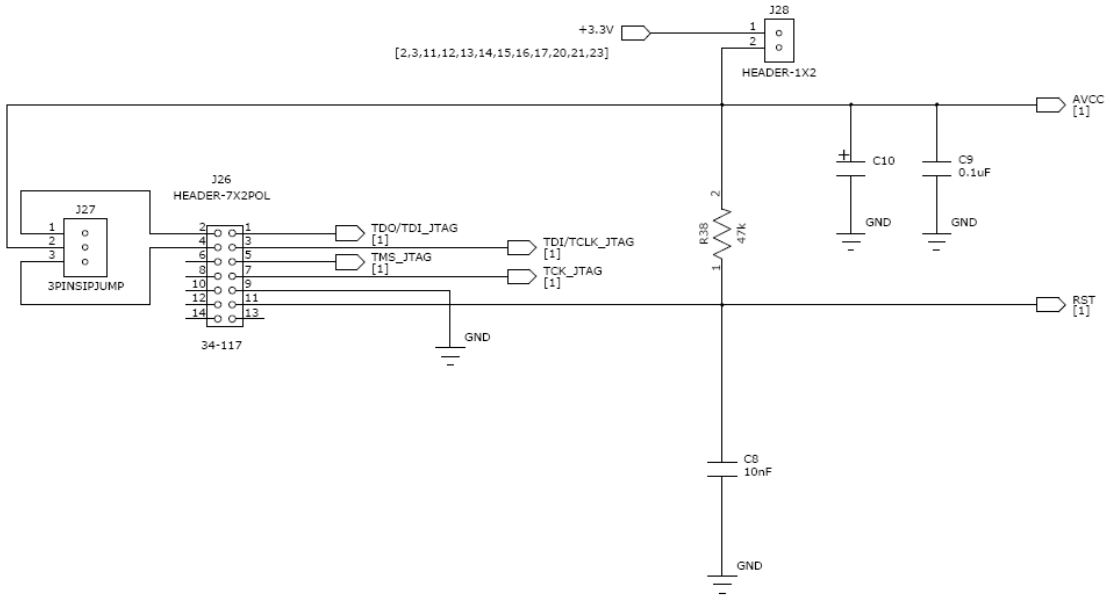


Figure 51. JTAG Programming Interface Circuit.

3.5.4.15 Power Connections Circuit

Each board has six fused power supply inputs. The 3.3V and 5V power rails are meant to be connected to an ATX power supply. The other four power supply inputs are for 12V rails. The general fusing circuitry for the power supplies is represented in Figure 52.

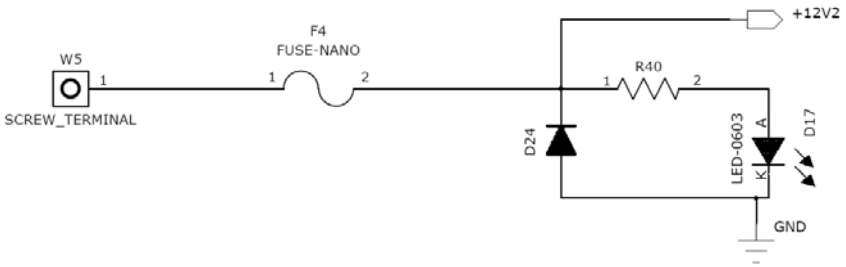


Figure 52. Power Supply Connection/Fusing Circuit.

The regulated power supply is connected to the screw terminal connector. That supply is then run through a Little Fuse Nano (with surface mount clip). These fuses offer very small packages with simple replacement. The power line after the fuse is the power supply connection used by on-board components. This line is also connected to ground through a reverse polarity rectifier diode which, along with the fuse, comprises the

reverse polarity protection circuit. In the event of a reverse polarity connection current will flow unrestricted through the diode and fuse and blow the fuse. Lastly, this power line is connected to ground through a resistor and Light Emitting Diode (LED) which serves as a power indicator.

The power supply connections are as follows:

Table 10. Power Supply Connections and Fuse Currents.

Supply Name	Voltage (V)	Terminal	Fuse Current
+3.3V	3.3	W2	1A
+5V	5	W3	1A
+12V	12	W4	6A
+12V2	12	W5	6A
+12V3	12	W6	4A
+12V4	12	W7	4A

The thruster motors, solenoid drivers and pump driver are broken up between the four 12V power supplies.

- Thruster solenoid drivers 1, 2, 3, and 4, ballast solenoids 5, 6, 7, 8 and 9, pump routing solenoids 1 and 2, and the emergency ballast solenoid driver are run off of the +12V rail.
- The two main thruster motor drivers are run off of the +12V2 rail.
- Ballast solenoid drivers 1, 2, 3, 4 and 5 are run off of the +12V3 rail.
- The pump relay driver is powered by the +12V4 rail.

Power supply connections are provided as screw hole terminals sized to fit M-40 screws.

3.5.4.16 Function Select Circuit

There are a total of three pairs of signals that are jumper selectable (Figure 53). These are the x-axis compass and cabin temperature, the y-axis compass and PC104 temperature and the water-leak sensor and supply voltage monitor.

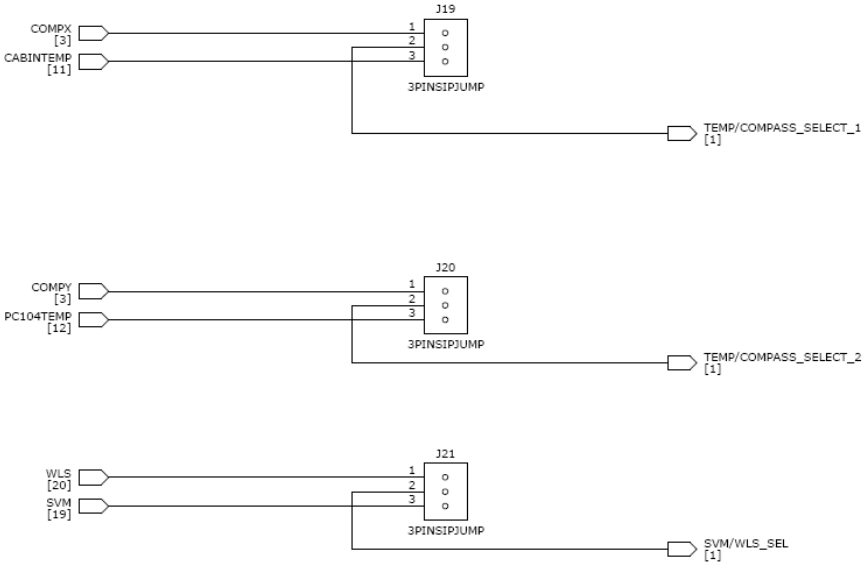


Figure 53. Function Select Circuit.

Three 3-pin headers allow selection of the various signals. These jumper positions are labeled on the PCB silkscreen. They are also outlined along with the MSP430 connections in Table 11, below.

Table 11. Function Select Connection.

Header	Signal Choices	MSP430 Signal	MSP430 Pin	MSP430 Port
J19	COMPX/CABINTEMP	TEMP/COMPASS_SELECT1	59	P6.0
J20	COMPY/PC104TEMP	TEMP/COMPASS_SELECT2	60	P6.1
J21	WLS/SVM	SVM/WLS_SEL	6	P6.6

3.5.4.17 Sub-Hub Printed Circuit Board

The circuit schematics and printed circuit boards were designed with PADS Logic and PADS Layout, respectively. The finished PCB's measure 4 in by 5.5 in. As is common with the first revision of any such design, a number of errors were found in both the circuit design and the board design. These are described in Appendix A. .

The top and bottom layers of the unpopulated boards, oriented with their top edge facing up, are shown in Figure 54.

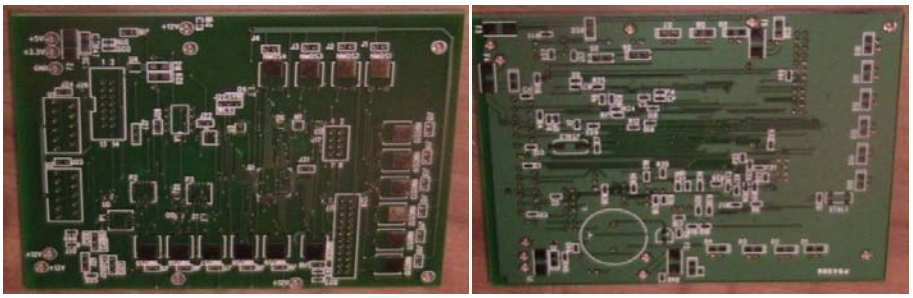


Figure 54. Unpopulated PCB Top and Bottom Layer.

It should be noted that the convention for jumper connections is as follows:

With the header in question oriented such that its reference designator reads left to right, pin 1 of the header is on the left, unless otherwise noted.

This means that, for instance, to connect a solenoid to J1 with the top layer facing up and the top edge facing up the positive pin should be connected to the left pin of the header (pin 1) and the negative pin should be connected to the right pin of the header (pin 2).

A partially populated board is shown in . This board includes a number of completed power supply connections, solenoid drivers, thruster motor drivers and the pump driver circuit. The hub-addressing, cabin temperature, I^2C bus and 1.25V reference generator circuitry is also populated.

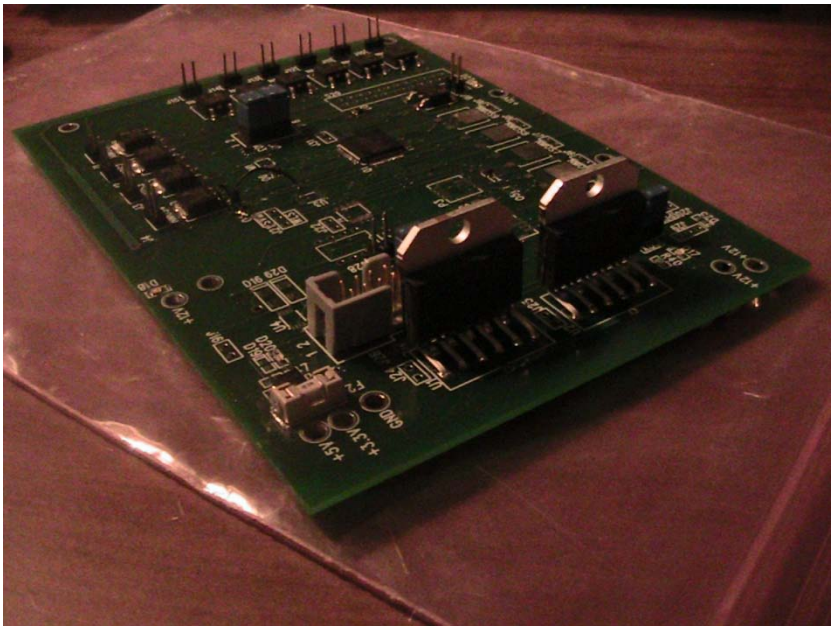


Figure 55. Partially Populated Sub-Hub PCB.

The full circuit schematics, parts lists and PCB design are provided in Appendix A.

3.5.5 Power

Before understanding the power supply needs for the vehicle and deciding on a solution to meet those needs, the vehicle's power consumption needed to be estimated. This estimation also provided a starting point from which a thermal analysis to determine the hull's cooling needs, if any, could be performed.

3.5.5.1 Power Analysis

To determine the necessary power supply requirements as well as estimating the heat produced in the hull a full power analysis was done on the electronics. Two operating points were chosen, the first with all components operating on a 100% duty cycle (worst case scenario) and the second with components operating at expected duty cycles.

Assuming a 100% duty cycle, the components with the largest power consumption (before power supply efficiencies) are the pump which draws a maximum 13A at 12V[29], or 156W, the main thruster drive motors with a stall current of 4.4A[30] but limited to 3A at 12V by the drivers[27]or 36W, and the computer which consumes a

maximum of 25W[19]. Aside from these components the majority of power consumption is driven by the solenoid valves which consume a maximum of 3W each at 12V (with a pressure of 60 psi)[31] and the H-Bridge thruster motor drivers which can use up to 3W on their own.

Using the minimum power supply efficiencies listed for the pico-PSU-120[32] (85% for the 3.3V supply, 86% for the 5V supply and 100% for the 12V supply) the maximum possible power usage was calculated as 524W. This is assuming a 100% duty cycle for all components and includes the extra peripherals mentioned above that are not actually in use (i.e. extra solenoid drivers and solenoids). This power consumption is therefore higher than the actual maximum power consumption of the current design and not of much use.

Of more use is the maximum power consumption at the expected operating point of all of the components in question. Reasonable duty cycles were chosen for all components based on their probable maximum usage. The main thrusters, for example, will probably only be used at half speed for missions requiring significant travel distances as this is more efficient and almost not at all for stationary stabilization missions. Full thrust will only be applied in an attempt to quickly stop the vehicle and will not occur for significant lengths of time. Because of this, it was assumed that the maximum duty cycle occurring for any length of time would be around 40%. In truth, this estimate is still probably a bit high especially with respect to the average duty cycle over the entire length of the mission. At a 40% duty cycle, the main thrusters only consume 18W as compared to the estimated 36W for a 100% duty cycle.

At the reasonable expected maximum operating point power consumption is calculated to be 175W. The full component list and power analysis is provided in Appendix B.

3.5.5.2 Supply Requirements

At the reasonable maximum operating point, the maximum power consumed by the vehicle's electronics is found to be 175W. From a 12 volt source, this necessitates a minimum continuous current draw ability of 15A. Estimating a reasonable mission time as the time required to complete 1 full lap of the WPI pool (18m x 9m or a 54m perimeter) and assuming a minimum safe travel speed of 0.1 m/s (allowing ample time to react to any system malfunction) the minimum mission time to travel the total distance for one charge of the battery pack would be 9 minutes. Allowing an extra 20 minutes for system preparation and testing yields a required minimum of 30 minutes to discharge one battery pack. This means that, at a 15A discharge rate, a 12V battery pack would need to have a capacity of at least 7.5 Ah. Alternatively, a battery pack would need to have a 87.5Wh capacity.

This analysis was based on a 12V supply. All components run on either a 12V supply or either the 3.3V or 5V supply provided by the DC-DC ATX power supply except for the sonar module. The sonar module requires 24V DC. If using a 12V battery pack, this necessitates a 12V-24V DC-DC converter to power the sonar module which can be extremely cost prohibitive at the high currents (8A peak) drawn by the sonar module. Another option is to use a 24 battery pack and 24V-12V DC-DC converter such as the pico-PSU-120WI-25V DC-DC ATX supply which acts the same as the pico-PSU-120 currently being used except that it accepts an unregulated input from 12V-25V. The last option is to simply connect two 12V battery packs in series providing a 24V source and a 12V source.

The most cost effective option (requiring the fewest additional components) is the use of a 24V battery pack with a 24V-12V DC-DC converter. It should be noted that the vehicle is currently still under tethered operation (an on-shore power supply is used in place of a battery).

3.6 Sonar Navigation System

3.6.1 Objectives

Once an approach was determined and information on the ultrasound physics was reviewed, the design process was split in several parts and a series of objectives were laid. It was determined that the system will contain three components: the ultrasonic transducers (piezo-electric), the transmit circuit and the receive circuit. It was also decided to use a microcontroller for controlling the pulses and for gathering and analyzing the data.

Since the first tests will be conducted in a pool, the maximum length that the navigation system should measure was chosen to be the maximum length of the pool – around 10m. Also, since a sonar device has a resonance problem, a minimum distance that could be identified was also chosen. The smallest side of the vehicle is around 20cm which was chosen to be the smallest resolved distance for the system.

The resolution for the device will probably be given by how accurately we will be able to record the data, which relates to the analog to digital converter used. This subject will be analyzed in the future sections.

Also, the physical size of the device is not of great concern, since we are just building a test apparatus to get an idea of the returning waveforms and what analysis techniques we should use.

Another feature that the sensor should have would be to detect multiple echoes from objects that are different distances away. This is the main reason we will need raw data analysis, compared to the classic on-off trigger that available ultrasound sensors are using.

3.6.2 Design approach and testing

Based on the prior art research, the decision was to first test the hypotheses obtained in the research by creating an air-based apparatus. The purpose of this step was to understand the behavior of the system as well as to test the key points of the design.

Once the experiment succeeded in air, the test would be moved to water, by insulating the transducers and dipping them in different level buckets to observe any changes.

The final step was to design and construct the water based apparatus based on the previous tests. Adapting the device to work in water proved more complicated than in air, as water is an incompressible fluid. Transmitting sound waves in water required a higher energy than in air, although the losses were much smaller. Based on these results, a new set of objectives was determined in the design process.

First, an underwater transducer had to be found. This differs from the air based ones, since it requires a higher energy and operating frequency. Next, a driving circuitry for this transducer was necessary. This involved the design of a balanced matching network that would transfer the energy to the transducer with minimal loss.

The final step was to put everything together, once all the elements of the apparatus were identified and carefully designed. A schematic of the circuit was built together with a printed circuit board (PCB) that could be ordered and tested.

3.6.2.1 Initial Design

As mentioned above, the first step was the construction of an air based ultrasonic range finder.

The system block diagram was created based on the previous research and objectives. It contains the blocks identified in the objectives. The microcontroller was identified and more detail on its purpose and operation is given. Also, several stages in the transmit and receive components are shown, such as the voltage driver, the small signal amplifier, the signal conditioning stage and the link to the PC.

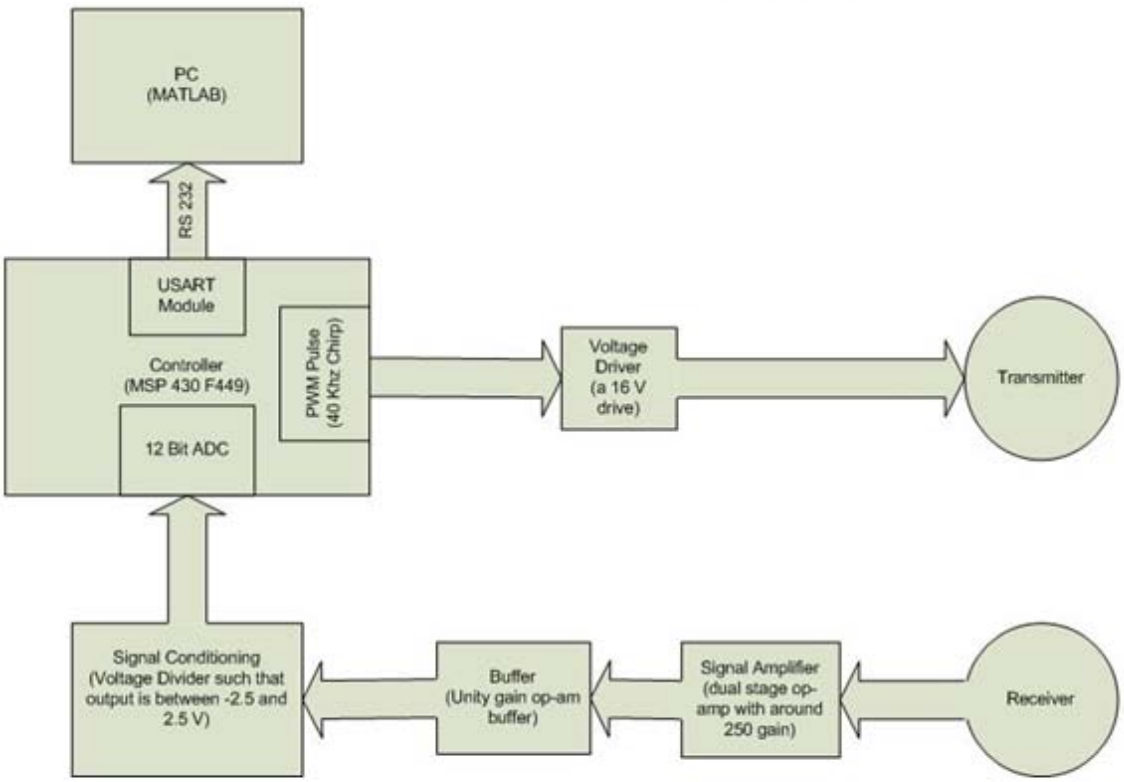


Figure 56. System Block Diagram.

3.6.2.2 Analog Design

The diagram shows the different components of the circuit. The main parts are the transmitter and the receiver that represent the ultrasonic transducers. On the transmitter side, the controller will do the job of sending out a pulse (chirp) of 40 kHz to the transmitter through the voltage driver (ST232). The driver will give a higher SPL level output, which results in a higher range. Whether this will be needed in the water, remains debatable. Furthermore, it also serves as the negative rail of the amplifying block used in the receiving side. Since the ST232 also has a voltage doubler and inverter, it is perfect for a higher rail voltage on the operation amplifiers.

Both transducers work at 40 kHz and are commonly used for air ultrasonic ranging.

Figure 57. Ultrasonic Transducer.

The receiver has a sensitivity of -65 dB SPL, according to the datasheet 0Using the SPL formula described above, and knowing that the sensor has an output of 1V/µBar, the minimum detectable signal coming out of the sensor was calculated to be: $-65 = 20\log\left(\frac{P}{P_r}\right) => P = P_r * 10^{-\frac{65}{20}}$. With P_r = 20µBar for air (the sensor has parameters for air), P is found to be: $P = 0.01125\ \mu Bar => V_{min} = 0.01125V$. Also, the maximum voltage must correspond to a 20 cm range, since this is the minimum resolved distance.

The signal amplifier block will be formed by a two stage operational amplifier with the gain of 250. The gain is open to changes, especially because of the difference between the water and air attenuation. A buffer and signal conditioning unit might be needed before the analog signal will be fed into the ADC of the microcontroller, since we don't want to lose the accuracy of the signal. The MSP430 F449 micro-controller has a 12 bit ADC that works between 0V and 2.5 V. Based on these values, the gain and offset of the amplifier stage can be further varied.

The design was based on a rangefinder circuit for hobby robots. A schematic of this design can be seen in Figure 58 on the next page.

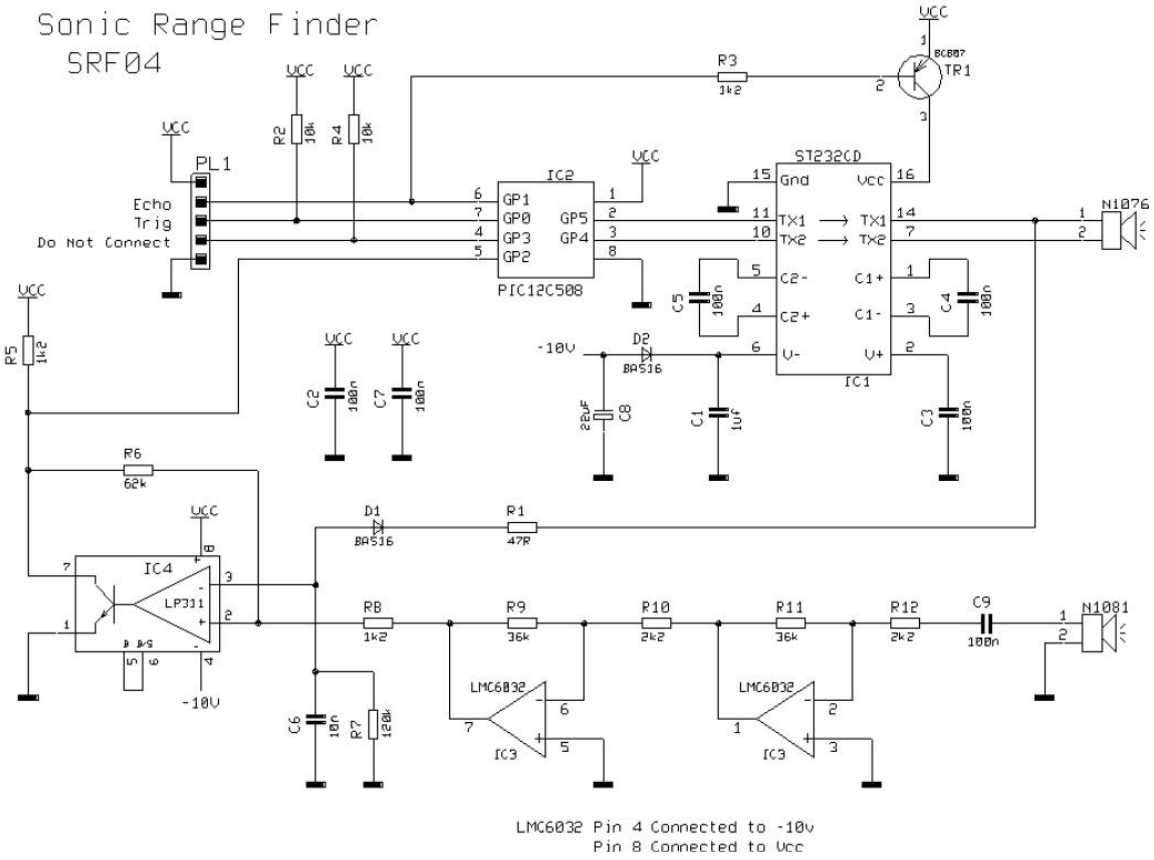


Figure 58. SRF04 Rangefinder Schematics

© 2001-2006 SuperDroid Robots Inc.[33]

The circuit uses a PIC processor to generate a chirp that gets amplified via the ST232 driver. This way a -10 V to 10 V pulse drives the transmitter (N1076). Then the processor holds the Echo pin high until a signal is received (N1081) and amplified trough the dual stage high gain amplifier (LMC6032) that triggers the LP311 comparator. This way, the duration that the Echo pin is high represents the time of flight and can be used to determine the distance.

The reason this design was chosen is its ease of construction and its ingenuity. Although similar, the design used for our experiment was different and didn't include all the components shown in this schematic.

As seen in the block diagram, the comparator of the receive stage will not be used since one of the goals of the project was to analyze the analog return signal and be able to distinguish between several objects that are in the same view angle but different distances from the transducer. Instead, the amplified output was conditioned such that it would be

between 0 and 2.5 V and 1.25 V offset. The reason for this is that the analog to digital converter (ADC) has only positive references, between 0 and 2.5 V maximum.

3.6.2.3 Software Design

With the circuit components identified, the software element was considered. Since the plan was to use the microcontroller to communicate data to a PC running MATLAB, both elements had to be configured. A basic flowchart was created to help program both the microcontroller, and MATLAB. The flowchart can be seen in Figure 59 on the next page.

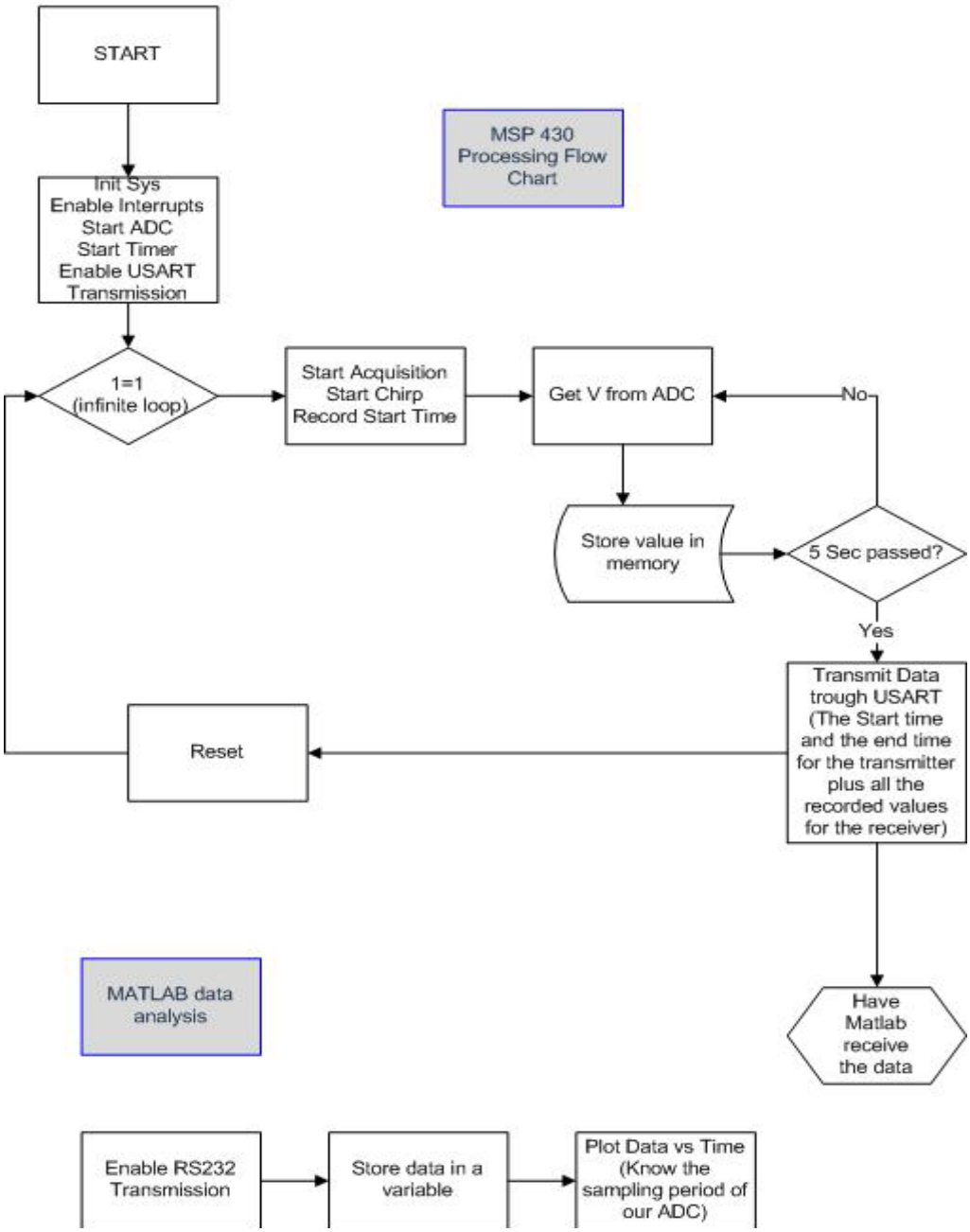


Figure 59. Software Flow Chart.

The flowchart includes steps for both the controller and MATLAB software. It can be seen that the MATLAB side doesn't require as much programming as the processor, but its configuration to communicate via RS232 was a little challenging.

The microcontroller will have to accomplish more than one goal. Firstly, it will be in charge of generating the chirp. As described above, this will be a 40 kHz pulse. The

duty cycle of the pulse is very small, around 1% or 2%. This means that 1 % of the time the system will send the pulse, and the other 99% of the time it will listen for return waves. Next, the ADC pin will be enabled and the data will be recorded in an array. At the end of the cycle, the data will be transmitted via RS232 to the computer. In the construction section, the operation of RS232 will be described more in depth. The data transmitted will be synchronized with the start time of the chirp so that the time of flight will be measured accurately.

On the computer side, MATLAB will be waiting for data to come on the RS232 communication port. The data will be stored in a variable, and then plotted. It is expected that the pulses will be recognized on the graph and the time easily measured. Some of the predicted issues are the shortage of memory on the microcontroller as well as the transfer rate of RS232.

3.6.3 Initial Design Construction and Testing

Once all the components of the design were identified together with the programming elements, we were ready to build the apparatus and test the hypotheses. For this a list of parts was ordered, shown in 0The parts were chosen to be suited for the proto board, since at this step the purpose was to test the functionality of the design. A MSP430F449 was borrowed from the ECE shop, together with wires and other elements. Also, components from the basic Lab Kit were used in the circuit.

For the construction of the circuit, it was decided that it would be split into two separate sections. The first step was to test the analog side of the design. This includes the voltage driver, the dual stage receive amplifier and most important, the two transducers. The second step was to integrate the analog circuit with the microcontroller and test the circuit together. On the digital side, the microcontroller could also be tested separately for RS232 communication, as well as the ADC.

The final goal of this test was to make the MSP430 controller capture the signals and be able to send them to the computer. Also, on the computer the data should be saved and analyzed.

3.6.3.1 Analog Circuit

The construction of the analog circuit was done on the proto board. First, wires were attached to the two transducers so they could be moved and tested on the breadboard. Also, since at this stage the microcontroller wasn't used, an LM555 timer was used to generate the 40 kHz pulse. The configuration schematic of the timer can be seen in the following figure.

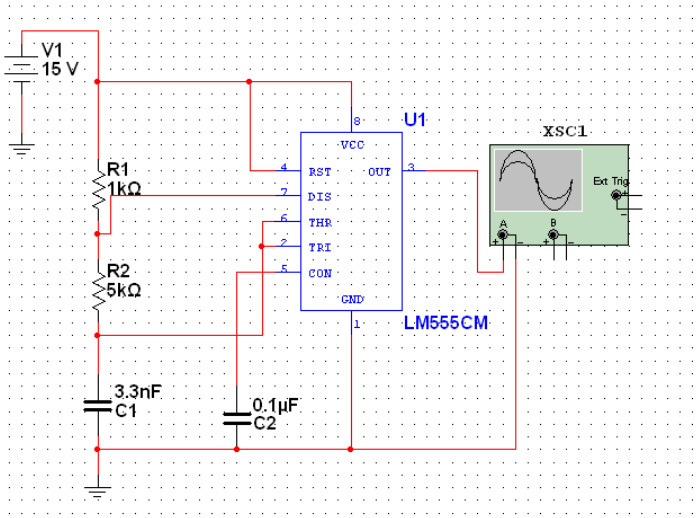


Figure 60. LM555 timer circuit.

The values were picked such that the duty cycle is very close to 50%. This is not a crucial requirement, but it is the expected operation of the transducers. Hence, R1 was picked to be 1kΩ, R2 5kΩ and C was calculated to be 3.3 nF.

To trigger the timer, the function generator available in the laboratory was used, set for a 1% duty cycle square wave. The function generator triggers the timer which drives the ultrasound transmitter. This circuit is basically separated from the receive side, but it proved that a common ground reduces the noise considerably. The receiver signal was connected to the dual stage amplifier, and the output was observed on the oscilloscope. The transmit signal was also observed on the oscilloscope, making it easy to trigger the view on the end of the transmit pulse. By adjusting the oscilloscope, the return wave was observed, as seen in Figure 61 below:

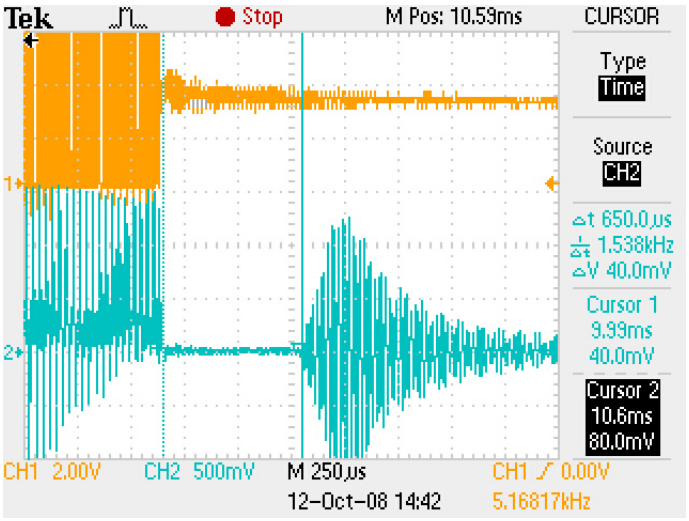


Figure 61. Return ultrasonic wave.

In the figure above, the top wave represents the transmit signal and the bottom wave represents the receive signal. It can be noticed that during the pulse, the receiver picks up the noise from the powerful transmit signal, but settles once this is stopped, allowing the return wave to be clear. It should also be noticed that the peak voltage for the return signal is a little over 1V meaning that with the proper offset this will fit into the 0 – 2.5 V window that the analog to digital converter expects.

Experiments were conducted with the two transducers facing each other or putting them next to each other and pointing them to surrounding objects. The experiments were successful, showing that the return wave is much bigger than the noise of the channel, and is easily recognizable.

To test the accuracy of the measurement, the transducers were placed facing each other, and the distance between them measured. Then the return wave was recorded on the oscilloscope and the time of flight determined. Figure 63 shows both the transducers while Figure 63 the obtained wave.

Figure 62. Ultrasound Transducer Test Setup (Air).

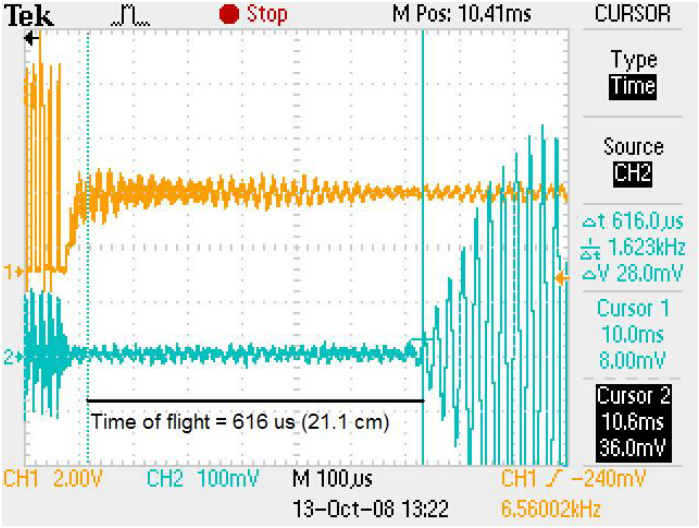


Figure 63. Transducers and the measured time of flight.

The first picture shows the transducers facing each other at a distance of around 21 cm. The results on the oscilloscope show that there are 616 microseconds which translates into 21.1 cm for a speed of sound of 343 m/s. These results were very accurate showing that the test apparatus works as expected.

Another hypothesis was tested, by placing two objects in front of the transducers and recording the result. The return wave showed two separate signals following one another, the first with higher amplitude than the second. The distance measurements matched the setup, proving that this apparatus could be used for detecting multiple objects.

Once this circuit was ready and proved functional, the next step was to test the transducers in water. For this, the two transducers had to be insulated. This was done by creating two cylindrical cases that the transducers would fit in, and then covering the ends of that cylinder. The cylinder was machined from a plastic tube. A rudimentary

approach was to enclose the transducers in plastic foil and glue the wires to the foil. This approach proved to work for our needs, since the transducer wouldn't get wet when dipped in water.

The test setup involved two buckets of water, of different heights. First bucket was around 15 cm deep while the second was around 40 cm deep. This way it was expected to see different time delays between the return waves. Also, it was expected that the amplitude of the signal will be much smaller due to the much higher density of the water. The two transducers were attached and dipped in the water facing downwards, and measurements were recorded. A set of recordings coming from the two buckets can be seen in the figure below:

Small Bucket **Deep Bucket**

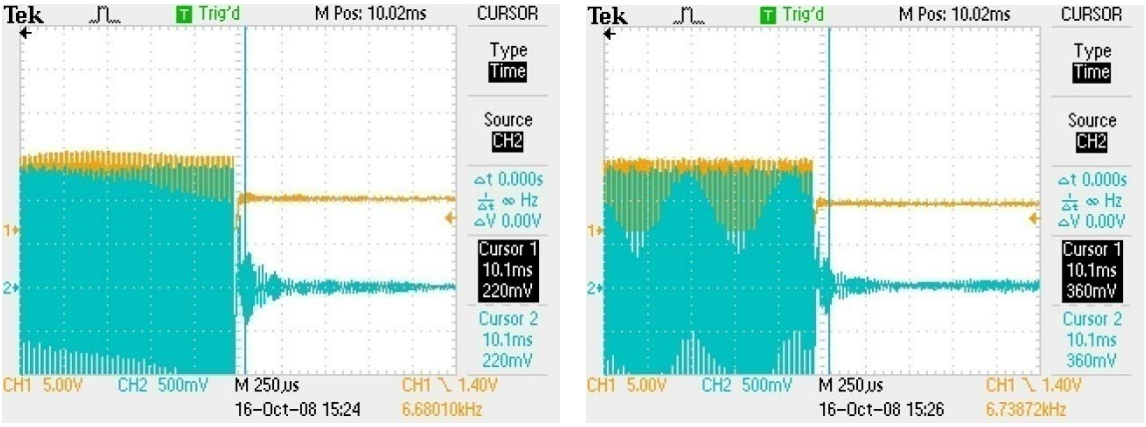


Figure 64. Underwater test return waves.

In this case, the results didn't prove as successful, as the return waves don't show on the oscilloscope. It was concluded that the transducers aren't powerful enough to generate sound waves in water due to the density and viscosity properties mentioned earlier. Hence, for the purpose of the project we focused our attention on the use of an underwater based transducer that would be able to successfully transmit ultrasonic pulses in water.

The air-based test setup was not abandoned since a lot of experiments were done on capturing and recording the data in order to analyze it and determine the distance, as described in the next section.

3.6.3.2 Digital Interface

Although the water test failed, the circuit was used for testing the microcontroller together with the MATLAB connection.

On the microcontroller part, the job was to get the analog data from the amplifier stage and convert it to digital using the analog to digital converter on the chip. This way, a series of data points representing the wave can be recorded and then analyzed.
At first the whole link between the transducer and the microcontroller (ADC) and then the microcontroller to the PC (RS232 port) was realized. The output of the transducer was connected to an ADC pin on the microcontroller. Then, the controller was programmed to sample the data on the pin and then convert it to the according voltage level.

The next step was to set the Universal Asynchronous Receiver/Transmitter (UART) interface on the microcontroller to be able to send data to the computer. This was done easily, since the setup of the UART unit involves enabling it and selecting the proper baud rate. This way the entire link was realized, and the MSP430 was able to get data in from the sensor and send data out to the computer. Moreover, Matlab software allows access to the RS232 port, hence a small program in Matlab was written to allow for data acquisition (Appendix E.2).

The microcontroller was programmed in C using IAR Embedded Workbench, and the source code can be seen in Appendix E.2.

The major problem in the system was the slowness of the RS232 connection. The highest baud rate that can be achieved is 115,200 baud. In the design, each data point requires 3 digits, each of these being one byte ASCII characters, and to transmit each data point, one more character (LF) is required by the software to tell the computer data will arrive. To transmit a byte over RS232, 10 bits of data are required (one start and one stop in addition to the eight data bits). Sparing the calculations, the final result is that each data point takes 347 μs to transfer which translates in a 2.88 kHz maximum sampling rate. This is a rather small rate, since the transducer operates at 40 kHz.

Experiments were conducted with the whole setup in place, and large sets of data were recorded. Figure 9 below shows a plot of 1000 data points obtained by transferring the sampled data from the controller to the PC. The vertical axis represents the voltage

level of the signal, calculated based on the ADC's conversion factor, while the horizontal axis shows the corresponding data point. The return pulses can be easily recognized in the plot, but in the same time no calculations can be done. The spikes contain both the return wave and the initial pulse, meaning that it's almost impossible to measure the time delay between them. These particular samples were recorded with the two transducers facing each other 6 cm apart. This translates to a time of flight of 175 µs, which is under our sampling rate of 347 µs.

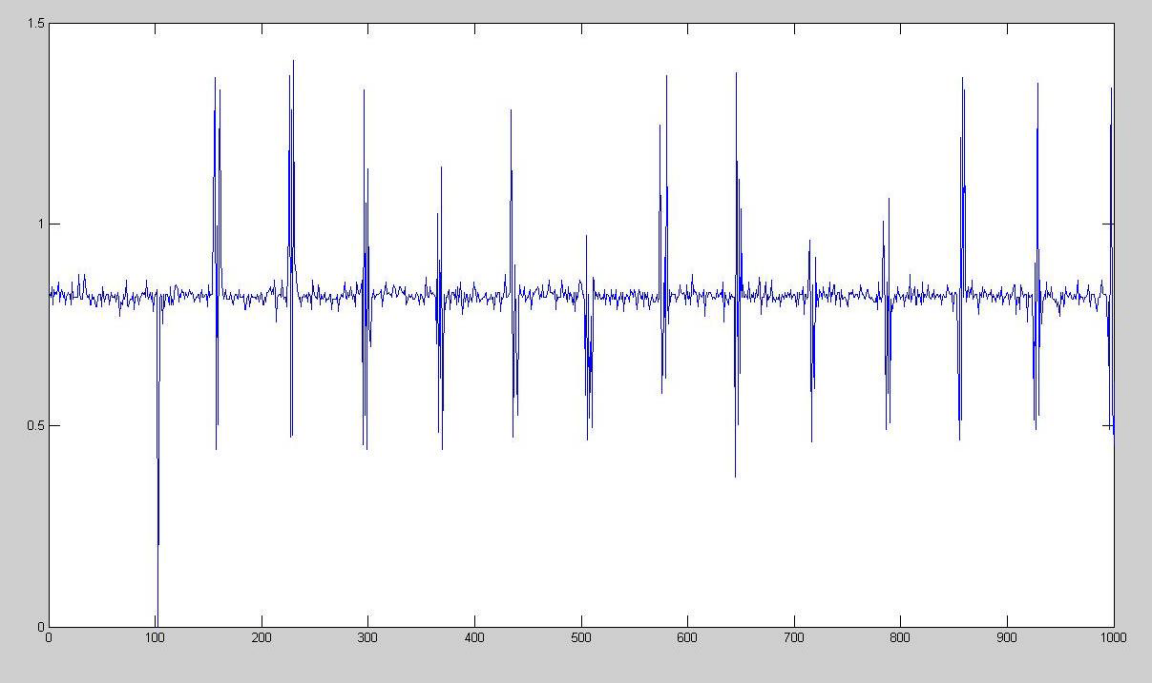


Figure 65. Data points recorded from the transducers.

The results show that the link is not working fast enough to be able to sample and send data in real time. This means that either a faster method of sending data must be used or a larger memory should be installed such that the microcontroller will save the data first, and then transmit it at the end of the sampling time. The second option is probably more reliable since a memory can be attached to the microcontroller in the final design allowing for the data to be saved very fast and in large quantities and then the transmission could be done while the device is in idle.

3.6.4 Underwater Transducer

Identifying an underwater transducer proved to be a very challenging task. Our focus was on devices that are used in regular fish finder equipment. For this, research was done in the area of fish finder operation.

Fish finders are devices used by fishermen to locate banks of fish in the water. The devices are comprised of a transducer that is usually mounted on the hull of a boat and an electronic display that shows the distance to the detected fish. The transducer operates based on the time of flight principle, the same approach that we are using. Also, some of these devices are powered by small AA batteries which also matched our energy needs. On the submarine we will need to conserve as much power as possible.

One hurdle was that usually, the required power of such a device varies between 200 W and 1000 W. This refers to instantaneous power, only needed during the pulse, but nevertheless, this meant that a power amplifier was needed to generate such a discharge.

3.6.4.1 Identifying the transducer

Intensive research was conducted to find the best underwater transducer. Unfortunately, most transducers are designed to be used with already built fish finders and they didn't have specifications, such as impedance, driving voltage, dynamic range or sensitivity. This made the research process more complicated.

Also, unlike the air based transducers used in the initial experiments, underwater transducers have the role of both transmitting and receiving signals, meaning that only one transducer is required for each ranging device.

There is a wide choice of devices on the market, but only a few matched our criteria. We focused our attention on the Norcross HawkEye DF2200PX[34] fish finder. This is a handheld device that shows the distance to underwater objects, mainly fish. Our main interest was in the power requirements of the device, and this particular one is powered by 4 AA batteries, which showed that the transducer will not require a lot of power.

The exact transducer used in the device couldn't be ordered separately, but we were directed to a similar replacement component produced by Airmar Technology Corporation.

The Airmar P23 transducer proved to fit the needs of the project, and all the specifications were available. Also, more in depth information about the theory of operation and the design of transducers and echo sounders was provided by the Airmar Technical Support.

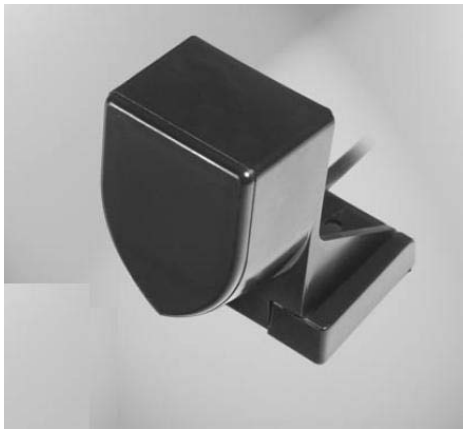


Figure 66. Airmar P23 Transducer
© 1995 Airmar Technology.[35]

The device has a resonant frequency of 200 KHz, which is standard for underwater ultrasonic transducers, it has a capacitive impedance of 465 – j170 Ω, and it is rated at 250 W at a 2% duty cycle. All these specifications are taken into account when the circuit is designed.

3.6.5 Driving Circuitry

Although the device will not require a lot of power on average, during the pulse it will need to be driven by a 200 W signal. This means that an amplifier will be required to drive the transducer.

Several techniques were researched in order to determine the best approach. From patent research and other papers or journals, it was determined that the best approach is to use a push-pull amplifier with a custom transformer that would match the impedance of the transducer. A basic model for such a device can be seen in Figure 67 below.

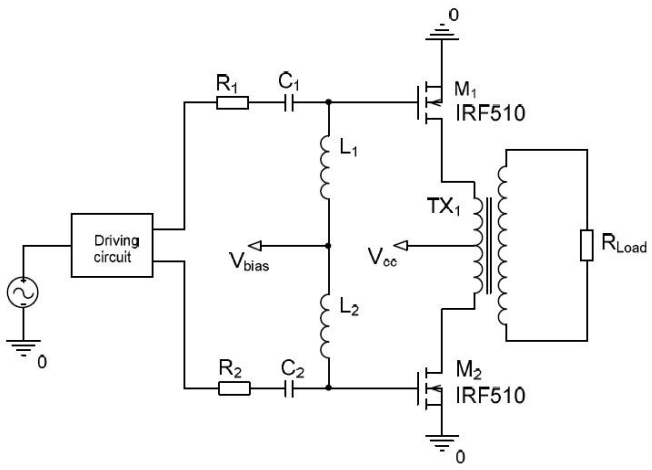


Figure 67. Basic Push-Pull amplifier

© 2006 Kaunas University of Technology.[36]

The design contains a driving circuit, which in our case will still be the microcontroller which has the capability of generating the high frequency pulse, the push-pull MOSFETs that act as switching elements, an amplifier with a center tap on the primary winding, allowing for push-pull operation and the load connected to the secondary winding.

To test this design, a simulation of the circuit was made. At first, just one side was simulated, to observe the behavior of the system. For this a generic 1 to 13 transformer was used together with a MOSFET, the 24 V supply and a complex impedance that would match the transducer. The MOSFET switch was triggered by a 200 KHz square wave similar to what the MSP430 would generate. Figure 68 and Figure 69 show the circuit and the resulting waves.

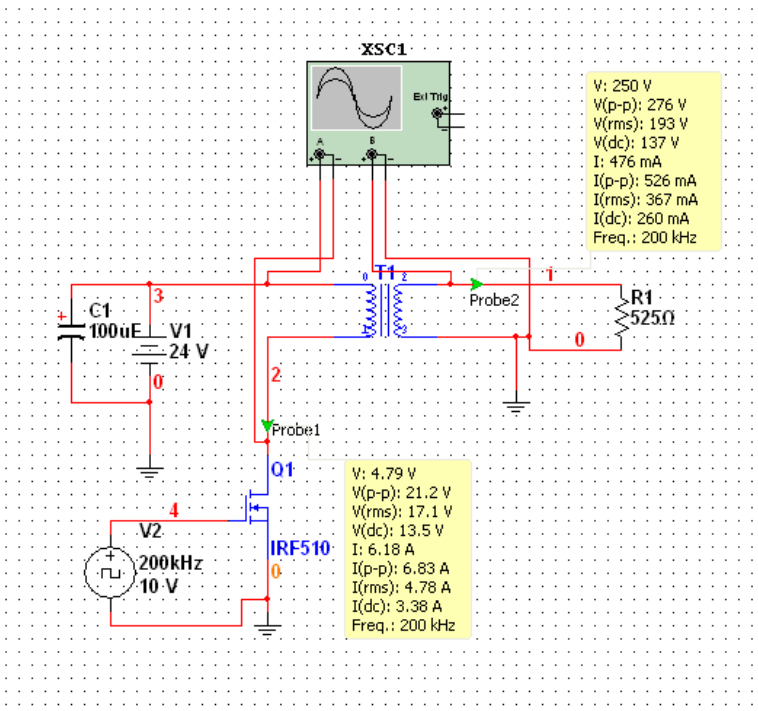


Figure 68. Transformer simulation.

The circuit simulation proved to be successful. It can be seen that the peak to peak voltage on the primary winding is around 21 V while the voltage on the secondary winding is around 276 V preserving the 1 to 13 ratio needed.

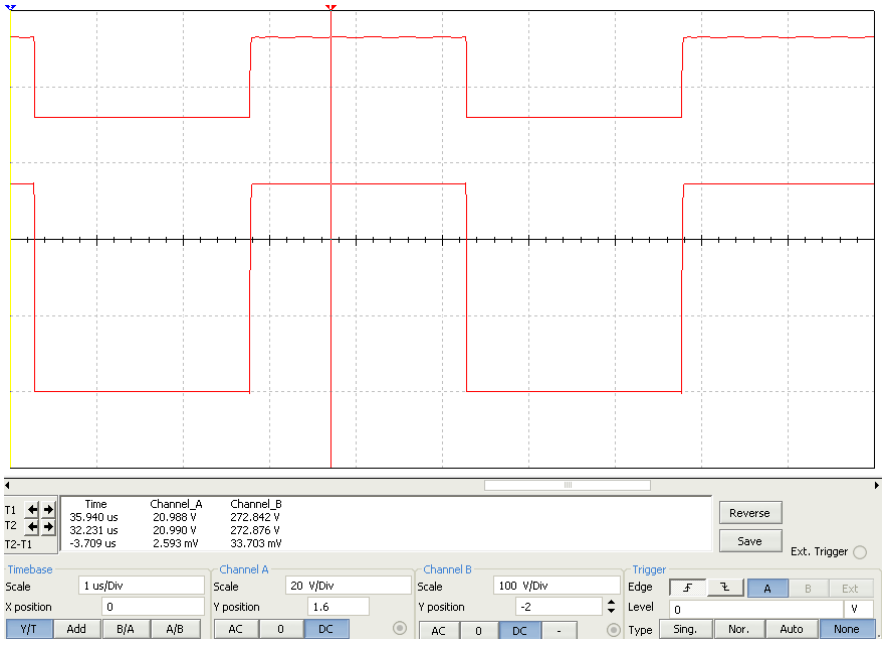


Figure 69. Simulation Wave Forms.

Once the ensemble was tested, we proceeded to the construction of the transformer. The balanced transformer would have to match the low impedance on the primary winding with the high impedance on the secondary winding. A ferrite toroid[37] was chosen for the transformer as it has a very good coupling coefficient and it is known to match the design parameters and theoretical values very closely. In order to choose the proper material and to calculate the right number of turns and to pick the right material, the impedance values were determined.

The calculation for the matching network is shown below, assuming a power of 200W on the transducer.

$$P = 200W$$

$Rezistive\ Z_S = 525\ \Omega$ (The resistance of the transducer which is on the secondary winding)

$$V_{cc} = 24V = E_{primary}$$ (The supply voltage)

$I_S = \sqrt{\frac{P}{Z_S}} \backsimeq 0.6\ A => E_S = 525 * 0.6 = 315\ V$ (Required voltage on the secondary winding)

$$n = \frac{N_S}{N_p} = \frac{E_S}{E_p} \simeq 13\ turns\ ratio => Z_p = Z_S * \left(\frac{1}{N}\right)^2 = 3.1\ \Omega$$

$$Hence, I_p = \frac{24}{3.1} \backsimeq 8A$$

At 200kHz, this corresponds to a 2.47 µH coil inductance.

Transformer design information was obtained from Amidon Corp where extensive information about core properties was found.

For an inductor, $L = \frac{A_L N^2}{10^6}$, where A_L is the inductance index, measured in mH/1000 turns and N is the number of turns. Based on these values and using the material size and type offered by Amidon, a chart was made to identify the best choice for the core. A copy of the chart used can be seen in Appendix E.3. Finally, the core material 61 FT 140 was selected. The core has a 1.4 inch outer diameter, an A_L value of 24.8 mH/1000 turns, and it requires 4 turns on the primary and 55 turns on the secondary. But, since the transformer will have a center tap, 8 turns are actually required on the primary winding.

For a current of 8A, a number 20 gauge wire size was chosen since it is the smallest that can hold the current. Considering the size of the wire, it means that the core was large enough for the 63 turns required.

3.6.6 Transmit/Receive Switch

Since the design is using only one transducer to both transmit and receive signals, the design required another component that would switch the transducer from the transmit state to the receive state. This needed to be done due to the high power transmit signal that could damage the receive amplifier.

A switch needed to be identified that would trigger between the two states. The two main requirements of this switch are to be fast and to be able to handle the high voltage of the pulse. The switching speed was chosen to be less than 1 ms, translating into a minimum resolved distance of one meter, which at this point was considered acceptable for our design. As for the high voltage, the switch must withstand a voltage of around 300 volts.

With these constraints in mind, research was done on available technologies. The first components that were considered were relays. Solid state relays have very fast switching, but cannot withstand the high voltage. Conventional electromagnetic relays usually have very slow switching, but can take the high power. Pin diodes were also considered for the switch, but it didn't prove a very useful technique due to their low voltage rating, as the highest voltage available was only 200 V.

In the circuit, the switching element should rapidly switch the two lines of the transducer between the driver lines and the receiver lines. For this purpose, the double pole double throw (DPDT) relay configuration was the best choice. A schematic of the relay circuit can be seen in Figure 70 below.

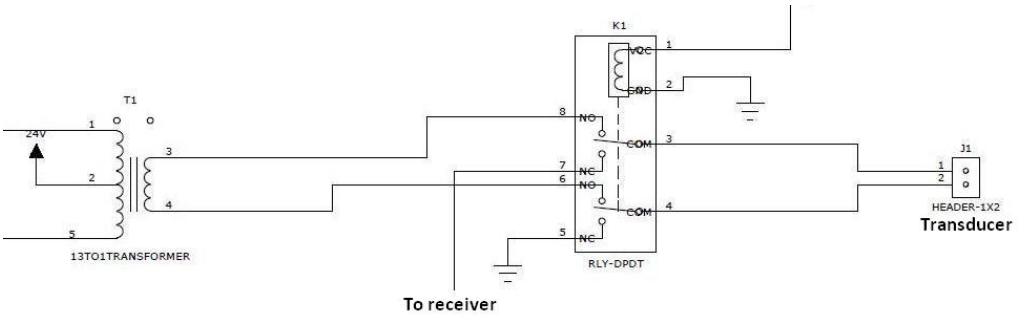


Figure 70. Relay circuit.

It can be seen that the relay connects the transducer to either the transformer or the receiver circuitry which is not shown in this schematic. Also, a 5V relay will be used, meaning a 5 V signal will trigger it on. The normally closed connection (NC) of the relay is connected to the receiver. This design approach was chosen due to the small amount of time that the transformer will be on. Most of the time the device will be in receive mode, which is the normal state of the relay that doesn't require energy.

We determined that the best design for the circuit is a Reed relay from US Relays, with 0.5 ms operates time and 0.25 ms switching time. These are one of the fastest electromagnetic relays, and although they have smaller voltage and current ratings, they are high enough for our needs.

3.6.7 Final design and testing

With all the components identified and tested we were ready to start the construction of the entire circuit. The goal was to create a circuit schematic and a printed circuit board design that would make the design and testing easier.

To design the circuit, Mentor Graphics PADS software was used. After gaining experience with the software, all the necessary components were created and circuit board footprints were created. The main units of the design are discussed separately in the following sections.

3.6.8 Transmit Driver

As discussed above, the transmitter consists of an impedance matching amplifier. Figure 71 below shows the circuitry related to the amplifier.

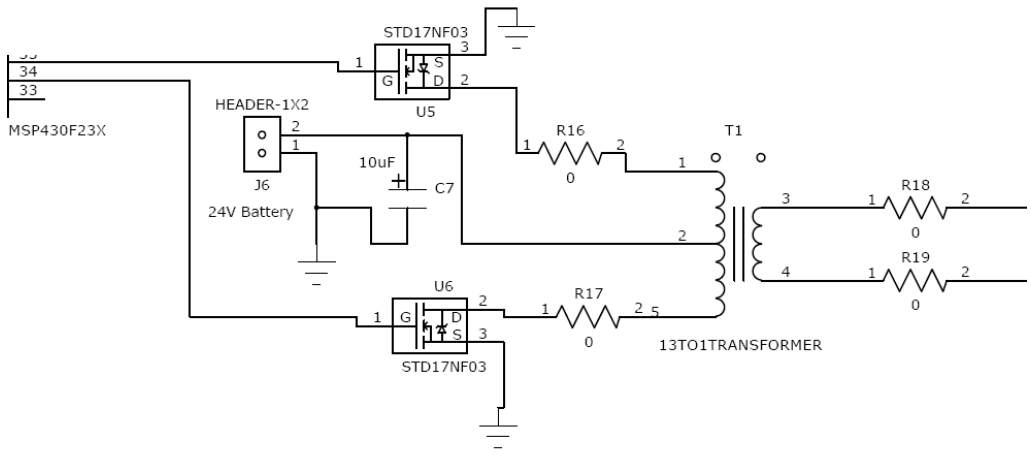


Figure 71. Power amplifier.

The center tap of the transformer is connected to the 24 V power supply. Capacitor C7 stops high currents from going back in the battery. The two MOSFETs are triggered by the 3.3 V signal from the microcontroller pins 34 and 35. They were chosen specifically for the high power they can withstand as well as for the small voltage they required to turn on. Also their impedance at 200 KHz is almost negligible, not affecting the calculations for the matching network. At a 3.3 V gate to source voltage, the drain to source current is up to 10 A which is more than the required 8 A. Three 0 Ohm resistors are test points that isolate different components and allow for easy debugging.

3.6.9 Receive Amplifier

It was decided to add a third op-amp that could also be configured as a low pass filter. This would enable us to increase the gain and to filter some noise that may appear. Currently the gain is set to 250 but this can easily be changed in further experiments. The components used are single rail op-amps that require 3.3 V to operate. Also, since the ADC input of the microprocessor is between 0 and 2.5 V it was decided to set the reference of the op-amps to 1.25 V by using the external reference pin on the controller and a voltage divider. The voltage divider was designed such that a very small current will be drawn.

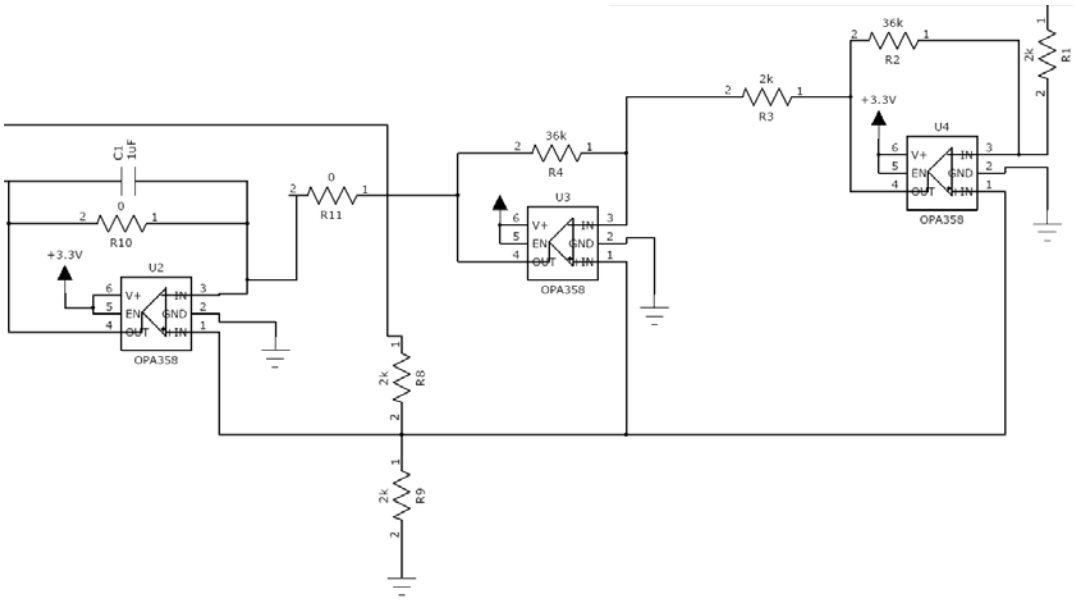


Figure 72. Receive Amplifier.

3.6.10 Transmit/Receive Switch

The switching between the power driver and the receiving amplifier was discussed in the previous chapter. It has a Reed DPDT Relay that switches between the two lines of the transformer and the dual stage op-amp amplifier. Also, the switching speed is below 1 ms which was one of the design constraints we had. Figure 72 shows the relay schematic in detail.

3.6.11 MSP430 Microcontroller

When the design was moved from the bread board to a printed circuit, a similar microcontroller was chosen that can easily be implemented in the design. The MSP430F235 is very similar to the F449, as it has all the features and it is programmed in the same way. The 64 pin LQFP package was chosen.

A detailed pin designation of the controller can be seen in Figure 73 below.

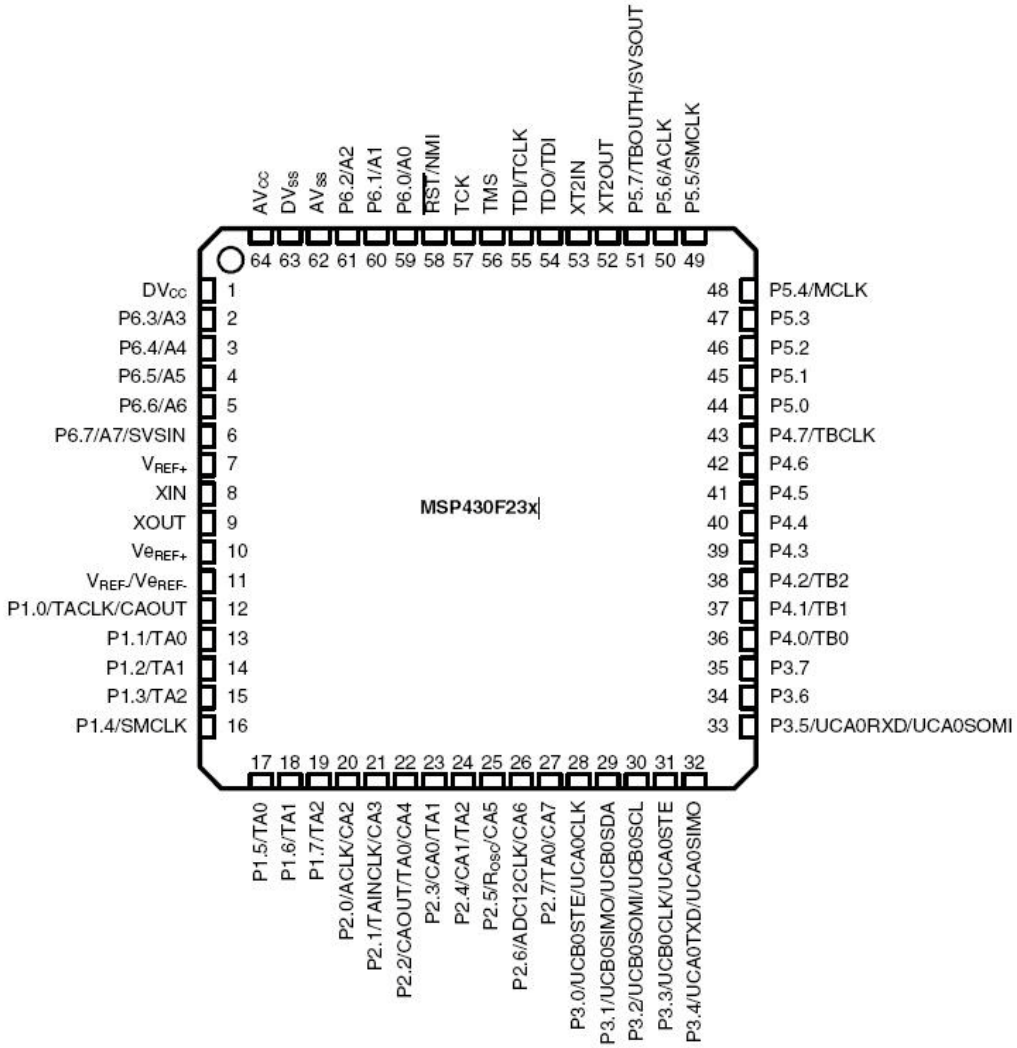


Figure 73. MSP430F23x Pin designation

Texas Instruments Inc. © 2008.[38]

This microprocessor was chosen because it is easy to use, we have a lot of experience from previous projects, and it provides all the necessary elements for our needs. Figure 74 shows detailed schematics of all the subassemblies of the microcontroller.

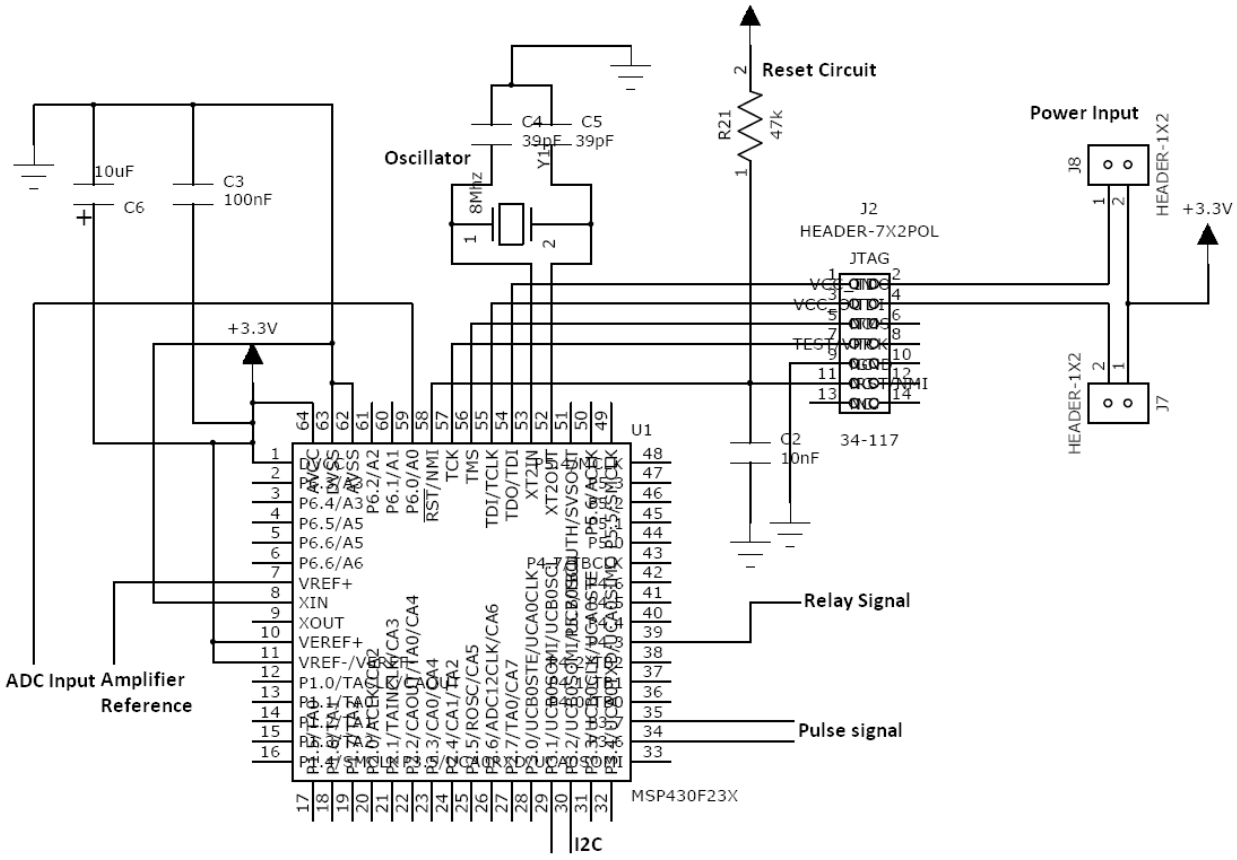


Figure 74. Microcontroller Schematic (MSP430F235).

The microcontroller will control all the assemblies of the board and analyze the information. Most of the attached components are peripherals that help its operation together with control and communication lines. All its components are discussed separately below:

3.6.12 Oscillator

The microcontroller requires an external oscillator, and for this an 8 MHz crystal was chosen. This will allow the microcontroller to operate at a decent speed. The clock will also be divided to create the 200 KHz pulse required to drive the transducer. This can be done using the output ports 34 and 35. As discussed above, additional circuitry is not required to control the two MOSFETs, since they require small voltage and very small currents to be triggered.

3.6.13 JTAG Connector

In order to allow the controller to be programmed a 14-pin JTAG connector was attached, according to the specifications provided by the manufacturer. This way, a programmer can be easily connected to the controller. Also, the two jumpers allow for two power inputs for the controller. This can be connected to either the boards power supply or the programmers power supply. This feature allows the processor to be run independently from the other components of the circuit. The reset circuit is also attached to the JTAG since it is a required option during the programming process.

3.6.14 ADC Input and Reference

The processor has a 12 bit analog-to-digital converter. For our purpose, this will be used with a 0 V to 2.5 V reference. Twelve Bits mean that a 0 V signal corresponds to the number 0 while a 2.5 V signal corresponds to the number 2^{12} or 4096. The signal coming from the op-amps will be connected to pin 60 which is the ADC input A0.
It was decided that the signal going into the ADC to be offset at half the maximum voltage, or 1.25 V. For this pin 7, or VREF+ was used together with a voltage divider circuit, since the reference is known to be very close to 2.5V.

3.6.15 I2C

In order to communicate to the main computer on the vehicle and transmit the information, it was decided among the team to use the I^2C format. This was chosen since all the communication between the vehicle's subassemblies is done this way. Also this protocol allows for fast and reliable data transfer. The communication only requires two lines, one data line and a clock, and the controller provides pins 29 and 30 (UCB0SDA and UCB0SCL) that can be set for I^2C mode. This will allow for the data gathered to be transferred to the main computer, where the distance to the obstacles can be calculated.

3.6.16 Control Signals

Finally, the microcontroller will be used to generate the control signals that would trigger the pulses. Input/Output ports 34 and 35 will be used to generate two square waves 180 degrees out of phase that will trigger the two op-amps and enable the push-pull operation of the power amplifier. The signal on pin 39 will switch the relay to connect the transformer and the transducer right before the pulse is generated, and immediately disconnect it once the pulse is over, to allow the return signal to be amplified in the op-am stage. A sample of the waveforms can be seen in Figure 75 below.

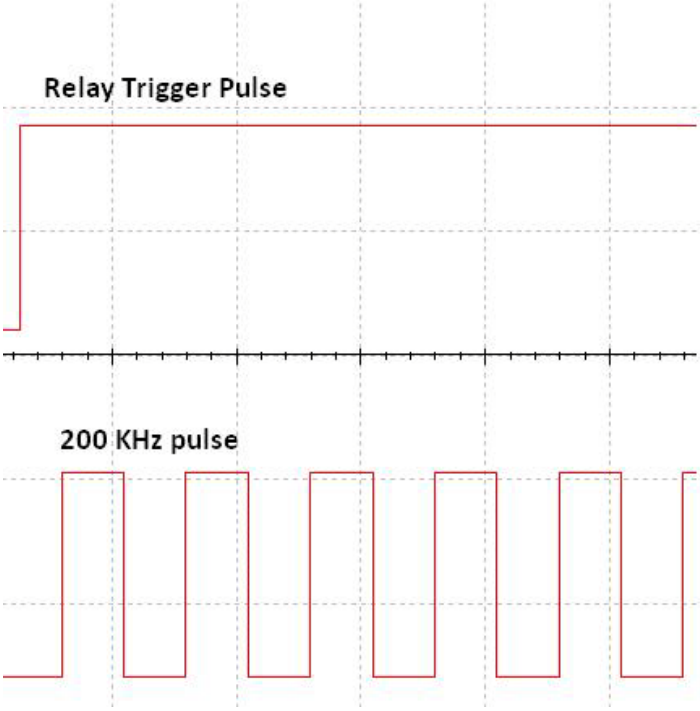


Figure 75. Relay switch and pulse waves.

Once all the modules were discussed and analyzed the entire circuit was put together.

3.6.17 Completed Circuit

The schematic for the entire circuit could fit on a standard schematic page, and nested schematics weren't needed. Also, all components have PCB decals attached to them that would make the PCB routing very easy using PADS. The transformer needed a custom decal, since it was built and designed by us. The relay also required a custom decal, since the model wasn't in the database. Since some of the components were used

by the control board that was designed by another team member, decals were used for this circuit as well. The headers were kept the same to assure a standard in the entire vehicle, and to allow for easy interconnections. Also, the microcontroller is used twice in the submarine.

Figure 76 shows the big picture of the circuit. All the components discussed above can be identified.

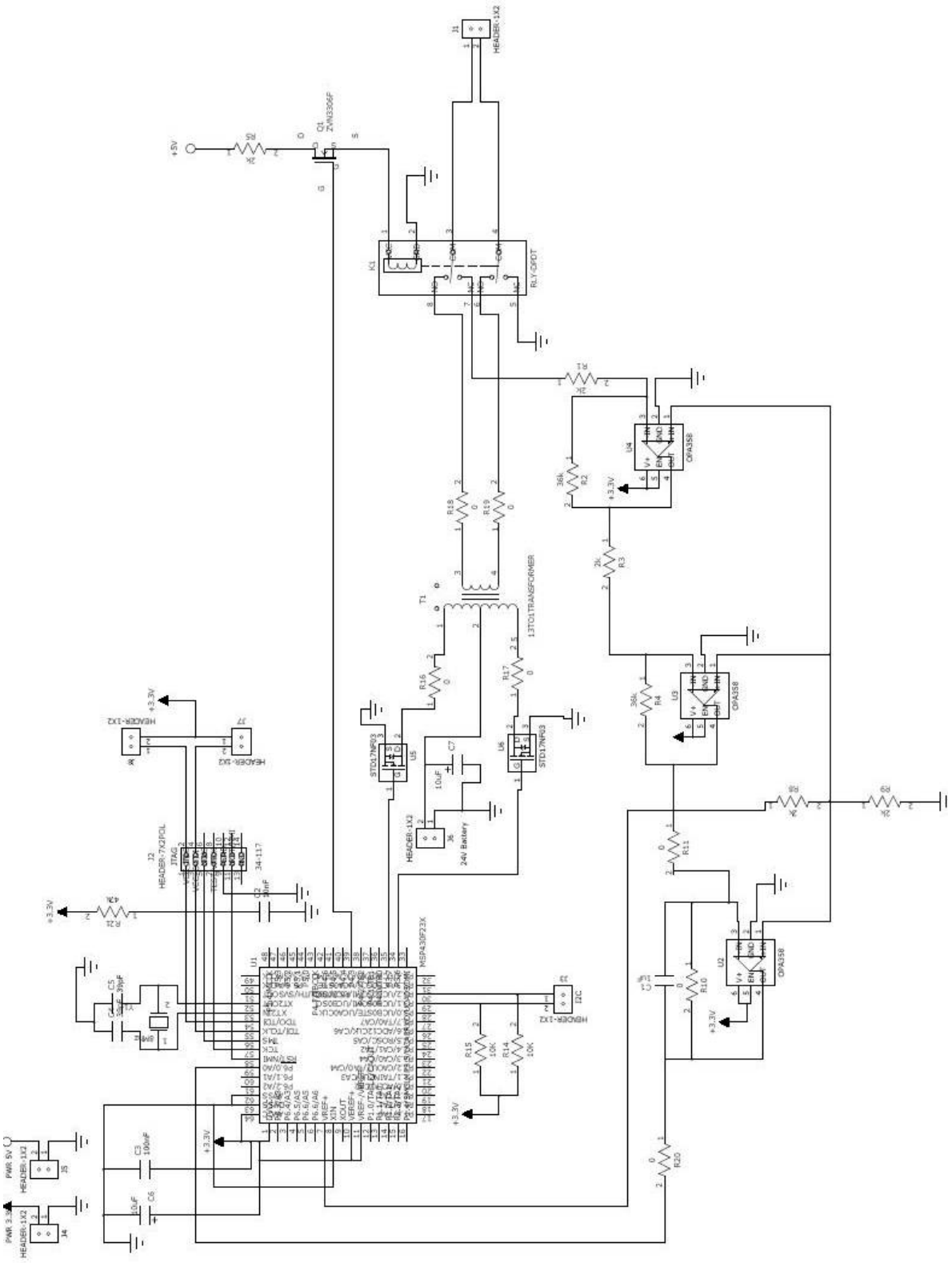


Figure 76.Completed Circuit.

A list of the order parts can be seen in Appendix E.2. The main components of the design were ordered. For initial tests and prototyping, a sample microprocessor could be ordered from Texas Instruments, while the passive components were all available in the department shop.

The next step was to create the printed circuit board of the circuit, build it and start testing the system. There are many components of the system that can be tested and debugged separately. Also, test points were inserted in the entire design to allow for easy debugging of separate assemblies.

Another aspect is to decide the data that would be transferred between the navigation system and the main computer. Considering the higher computing power of the PC104, most of the calculations could be done there and raw data could be transferred from the MSP430 microcontroller.

The tests will have to be conducted in water, the operating medium of the transducer. The ultimate goal would be to attach the device to the submarine and test the accuracy of the resulting data.

Ultimately, the project involved a lot of research on the ultrasonic technology. Understanding the behavior of sound waves in water and in air represented the beginning of constructing a functional device.

The process involved a series of iterations, by first creating an air based test apparatus, and then moving to a water system. A series of experiments were conducted to test the theoretical Hypothesis, and, as expected, some of them proved that the practice don't always match the theory.

The project involved a very broad area of electrical engineering proving that a successful engineer must have a very wide knowledge. Creating the power amplifier involved some knowledge of power electronics and transformers, while making a very sensitive receive circuit involved some knowledge of operational amplifiers.

Another aspect of the project was involving the digital aspect of electronics. A microcontroller had to be used to control the system. Computer engineering knowledge had to be used to program the processor and to run all the required tests.

One main thing that should have been done better is the time management for the project, and understanding that sometimes a challenge requires more time. Although most of the

subassemblies of the circuit were tested, the final circuit still needs to be put together and debugged. Even so, the project provides a very strong basis for the future iterations and for the system realization of an underwater navigation system.

A lot was learned in making this system, recommendations have been provided to ease the construction of the system in the future.

3.7 Controls

On completion, the multitude of sensors will be utilized to provide feedback for autonomous control of the craft. The first step to achieving autonomous control was to form a system of equations which modeled the dynamics of the craft.

Using Rigid Body dynamics, a system of equations were created to fully define the attitude and the position of the craft with respect to a defined world frame. The world frame was designed to have its Z-axis in the direction of gravity, and a fixed X and Y – Axis. Transformations between the world frame and the body frame would be achieved with the help of three euler rotations.

On solving the rigid body dynamics for the craft, the following set of equations were derived for the control of the craft:

$$\dot{u}^E := \frac{X}{m} - g \cdot \sin(\theta) - \left(q \cdot w^E - r \cdot v^E\right)$$

$$\dot{v}^E := \frac{Y}{m} + g \cdot \cos(\theta) \cdot \sin(\phi) - \left(r \cdot u^E - p \cdot w^E\right)$$

$$\dot{w}^E := \frac{Z}{m} + g \cdot \cos(\theta) \cdot \cos(\phi) - \left(p \cdot v^E - q \cdot u^E\right)$$

$$\dot{p} := \frac{L - q \cdot r\left(I_z - I_y\right) - q \cdot h_z^1 + r \cdot h_y^1}{I_x}$$

$$\dot{q} := \frac{M - r \cdot p\left(I_x - I_z\right) - r \cdot h_x^1 + p \cdot h_z^1}{I_y}$$

$$\dot{r} := \frac{N - p \cdot q\left(I_y - I_x\right) - p \cdot h_y^1 + q \cdot h_x^1}{I_z}$$

$$\dot{x}_E := u^E \cdot \cos\theta \cdot \cos\psi + v^E \cdot (\sin\phi \cdot \sin\theta \cdot \cos\psi - \cos\phi \cdot \sin\psi) + w^E (\cos\phi \cdot \sin\theta \cdot \cos\psi + \sin\phi \cdot \sin\psi)$$

$$\dot{y}_E := u^E \cdot \cos\theta \cdot \sin\psi + v^E \cdot (\sin\phi \cdot \sin\theta \cdot \sin\psi + \cos\phi \cdot \cos\psi) + w^E (\cos\phi \cdot \sin\theta \cdot \sin\psi - \sin\phi \cdot \cos\psi)$$

$$\dot{z}_E := -u^E \cdot \sin\theta + v^E \cdot \sin\phi \cdot \cos\theta + w^E \cdot \cos\phi \cdot \cos\theta$$

$$\dot{\phi} := p + (q \cdot \sin(\phi) + r \cdot \cos(\theta)) \cdot \tan(\theta)$$

$$\dot{\theta} := q \cdot \cos(\phi) - r \cdot \sin(\phi)$$

$$\dot{\psi} := (q \cdot \sin(\phi) + r \cdot \cos(\phi)) \sec(\theta)$$

Here, θ, φ and ψ are the three euler angles, p, q and r are the rate of change of roll, pitch and yaw, u, v and w are velocities and x, y and z are positions. X, Y, Z, L, M and N reflect the hydrodynamic properties and control inputs of the craft, which would be drag, buoyant forces, directional thrusters, main thrusters and the ballast system.

With the help of these equations, controllers can be formed to stabilize and control the craft for various runtime simulations.

4 Final Test of Sub

4.1 Vehicle Control

As no autonomous operation is required as of yet, all control of the sub is currently manual and provided through a monitor station (i.e. external computer) via a tether. This section outlines the communication methods between the Sub-Hub system board and external monitor station.

4.2 Communication

Communication between the vehicle and monitor station was achieved through the use of a Texas Instruments ez430-RF2500 development tool (on the monitor station side) connected to a Sub-Hub system board (located in the sub) through the I^2C bus.

The ez430 was configured to send and receive commands through a USB port using the target boards MSP430F2274 microcontroller. The target boards output pins P15 and P18 were configured as I^2C SCL (clock) and SDA (data), respectively. The ez430-RF2500 kit includes the MSP430 Application UART Driver which allows the USB dongle to be used as a serial port on COM3. In this way a new connection can be opened in HyperTerminal to allow the transmit and receive of ASCII characters through the keyboard. The HyperTerminal connection should be configured for a baud rate of 9600 kHz. Receipt of certain ASCII characters then triggers a software switch on the Sub-Hub system board which toggles the respective solenoids.

The ASCII characters used for this control along with their respective Sub-Hub outputs and actual operations are shown in Table 12, below.

Table 12. Vehicle Test Solenoid Control and Connections.

B	HEX	Solenoid Controller Header Number	Operation
1	0x31	J1	Fore Starboard Thruster
2	0x32	J2	Fore Port Thruster
3	0x33	J3	Aft Starboard Thruster
4	0x34	J4	Aft Port Thruster
a	0x61	J3, J4	Aft Thrusters
s	0x73	J1, J3	Starboard Thrusters
d	0x64	J2, J4	Port Thrusters
f	0x66	J1, J2	Fore Thrusters
j	0x6A	J9	Fore Ballast Fill
k	0x6B	J10	Fore Ballast Purge
l	0x6C	J11	Aft Ballast Purge
;	0x3B	J12	Aft Ballast Fill

Currently, stand-alone operation of the Sub-Hub system through IAR Embedded Workbench is erratic. For predictable operation the JTAG programming interface must be left connected to the board. The interface need not be in debug mode however and JTAG circuit jumper, J27, should be left in the lower position (connecting pin 2 to 3) to allow use of the external power supply. The Texas Instruments MSP-FET430U USB to JTAG programmer/debugger was used.

The code used for this connection is given in 0.

4.3 Tether

The original intention, with respect to vehicle control, was to use a four wire tether. The four wires of the tether were to be as follows:

- +12V Supply
- Ground
- I^2C Bus SCL (clock)
- I^2C Bus SDL (data)

Unfortunately, the current circuitry associated with the I^2C bus will only allow for a bus length of about two feet (results were the same using either 16 AWG stranded or 22 AWG solid core wire). The I^2C bus lines, therefore could not be controlled through the

tether (as they can barely reach through the sub). This required that the ezMSP-RF2500 be installed inside of the sub and a USB extension be run out to the monitor station through the tether. The I^2C bus lines, therefore, were run completely inside of the hull. As operation of the Sub-Hub still required the MSP-FET430U to be connected to the PCB, a second USB cable had to be run through the tether.

The +12V supply and ground wires remained unchanged from the original design. The wire used for these was 10 AWG stranded. Much smaller gauges of wire could have been used but 10 AWG was deemed flexible enough and offered a lower resistance than thinner wires.

The actual tether used includes the following:

- +12V (10 AWG)
- Ground (10 AWG)
- Type A Plug USB Extension (Female end located in hull)
- Type B Plug to Type A Plug USB Cord (Type B Plug located in hull)

5 Recommendations

5.1 Electronics

A number of modifications electronics and error revisions are also suggested for the electronics. Appendix A.6 outlines errors in both the circuit schematics and PCB design which should be fixed for any future revisions of the Sub-Hub boards.

There is also room for improvement through modification of the design itself. Upon completion and implementation of the boards it was found that using standard 0.1in spaced 2-pin headers for connection of the solenoids, motors, I^2C bus lines, etc. was not ideal. These connections are not very robust and have a tendency to be easily pulled out. It is suggested that this design be replaced with latching connectors with strain relief's.

It was also noticed that the power supply connections for the 3.3V and 5V supplies are located too close to each other. This makes it difficult to connect both supplies at once with the standard M-40 screws intended for the application. It is suggested that in future revisions of the board these be spaced out. It is also suggested that the screw terminal system be abandoned and replaced with latching polarized connectors for all power supply connections.

Upon actual implementation of the serial connection scheme, it was realized that the inclusion of an RS-232 driver on the Sub-Hub boards would have made connections much easier. It is suggested that provisions for this connection be included on any future revisions of the board. This would allow the boards to connect directly to the PC/104 and monitor station. The I^2C bus could still be used to connect the boards to each other.

5.2 Sonar System

The current system design contains three separate transducers that offer data in the three different coordinates. As a recommendation, additional research should be done in implementing the transducers on the same circuit and be controlled by the same controller. The project could be further expanded and improved by utilizing a more powerful microcontroller that would be able to transfer data faster. Also, the approach for the driving circuit of the transducer does not seem to be the most efficient although most echo sounders in the industry use it, and other techniques could be implemented.

5.3 Hull Modification

Completion of this project has uncovered a number of large obstacles that may have been avoidable through simple design change.

While construction of the current Lexan hull design was possible and the heat forming process was educational, this design does have a number of faults. Deflection of the side panels could cause seal failure at pressures higher than that seen in the test tanks. As explained, this hull shape was also difficult to construct and seal.

A recommended change in hull design would use large diameter Lexan tubing (around 8" outer diameter) to create a cylindrical hull. This would allow a modular design with modules connected either by threaded fittings or latches positioned around their circumference. The connected modules could simply be sealed with o-rings. This design would also be more spatially efficient as there would be no sharp angles creating, for example, the lost space toward the ends of the current hull design.

Such a hull system would allow for easy accommodation of drastic system design changes and quick and easy swap out of different components. One example of the usefulness of such a modular design would be the ability to add in an extra battery pack for longer missions. Due to the static size constraints of the current hull system, significant rearrangement of all internal components would be necessary to attempt such an addition. A modular hull, however, would allow an extra battery module to be added with little consequence.

Such a design is not perfect as additional modules would vary the size and shape of the sub requiring possible recalibration of control parameters, however, such a hull system could be desirable in comparison to the current design.

5.4 Magnetic Coupling

Although we were able to construct a working magnetic coupling, this system is yet to be optimized. Shape, size and orientation of the magnets, all affect the efficiency and torque limits of the coupling, and adopting the study conducted by Liu, Wang and Wu would benefit the efficiency and final performance of our submarine greatly.[39]

Furthermore, the gap distance between the drive shaft and the follower needs to be optimized for the pressure and performance conditions expected for the system. Added controllability for the system could be achieved utilizing the system suggested by

Chocron and Mangel[40] which would allow for a rotatable thruster system with the inherent advantages of a magnetic coupling system.

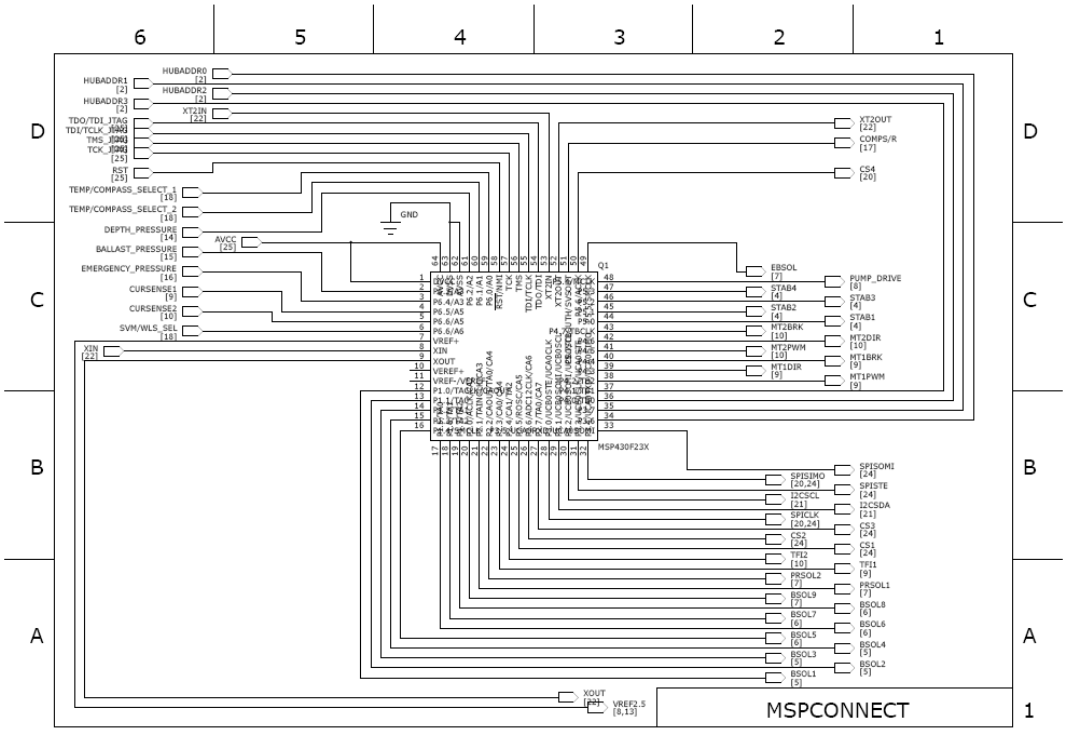
5.5 Ballast System

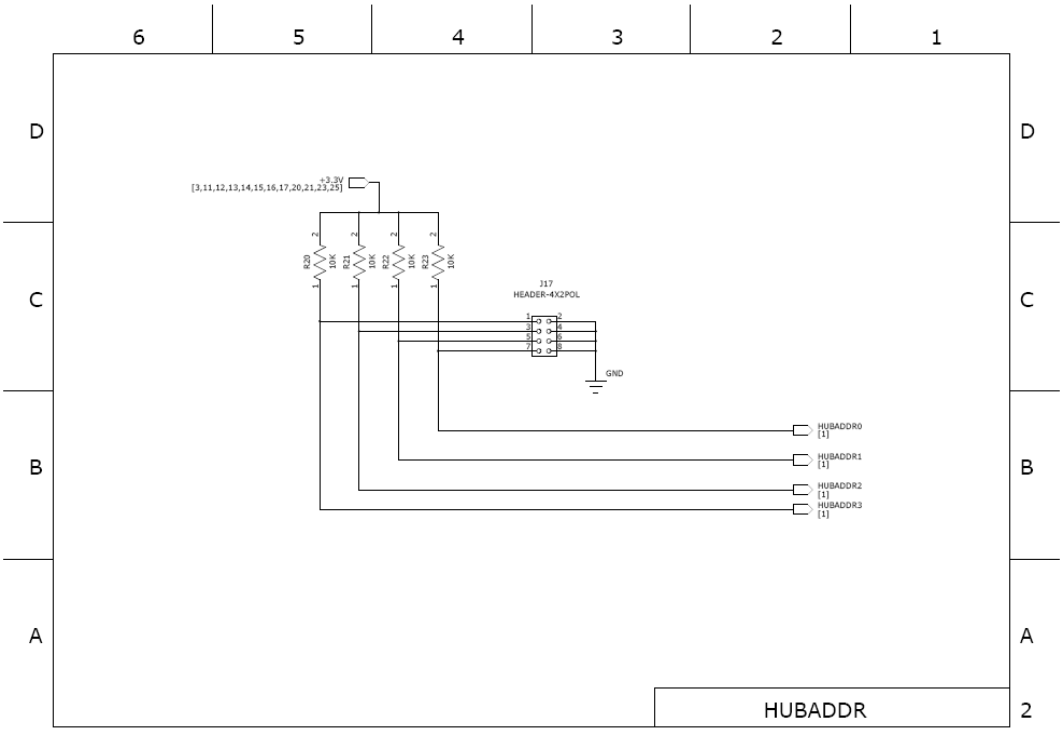
Though the ballast tanks created for this craft work, they have not been extensively tested. Extensive testing to determine capabilities of the tank would be important in determining final submarine capabilities. Furthermore, optimization of bladder material could ensure longevity of the ballast system tanks.

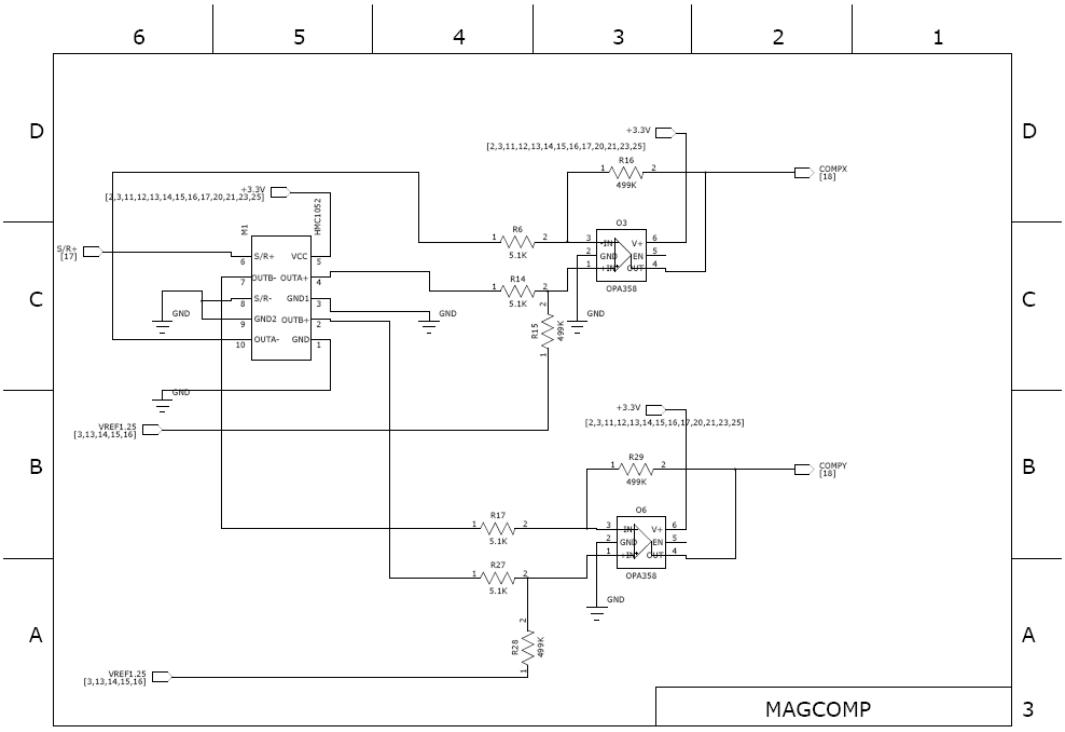
Appendix A. Sub-Hub System

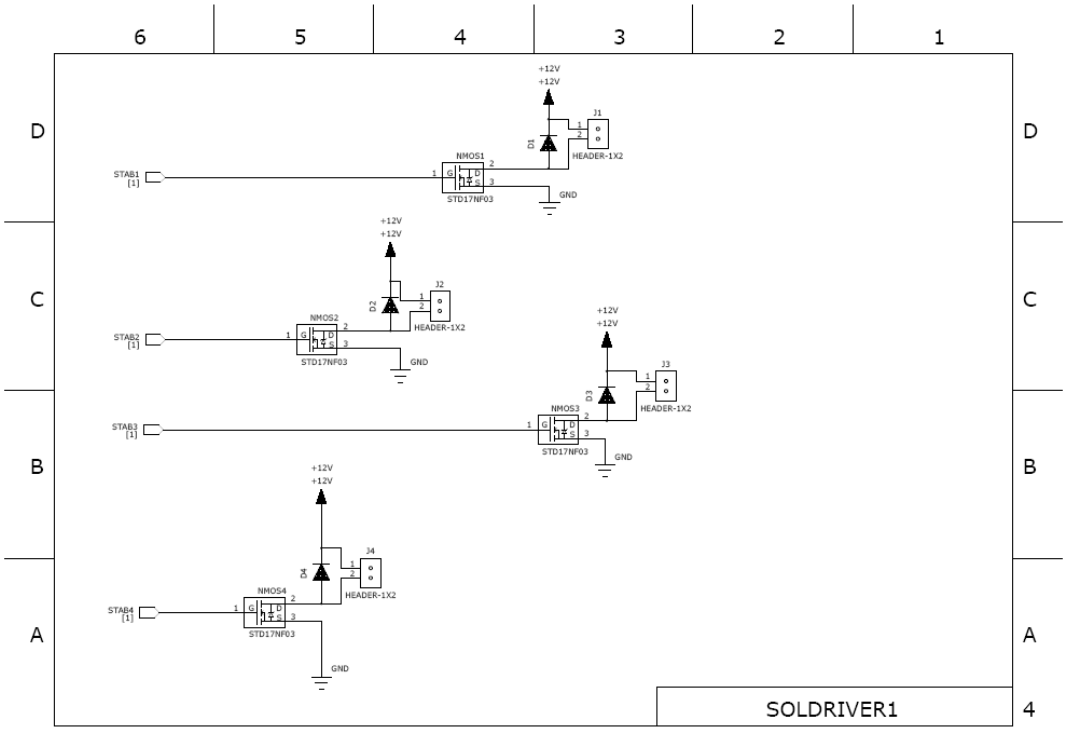
A.1 Sub-Hub Circuit Schematics

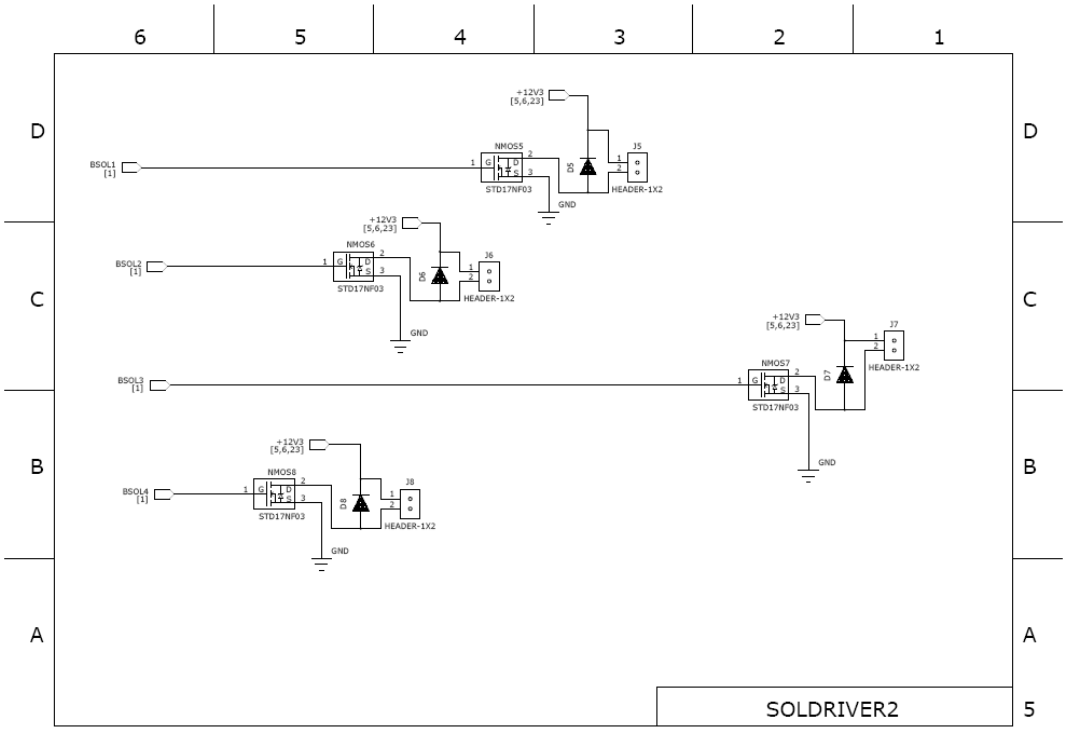
The full circuit schematics for the Sub-Hub system PCB are included below.

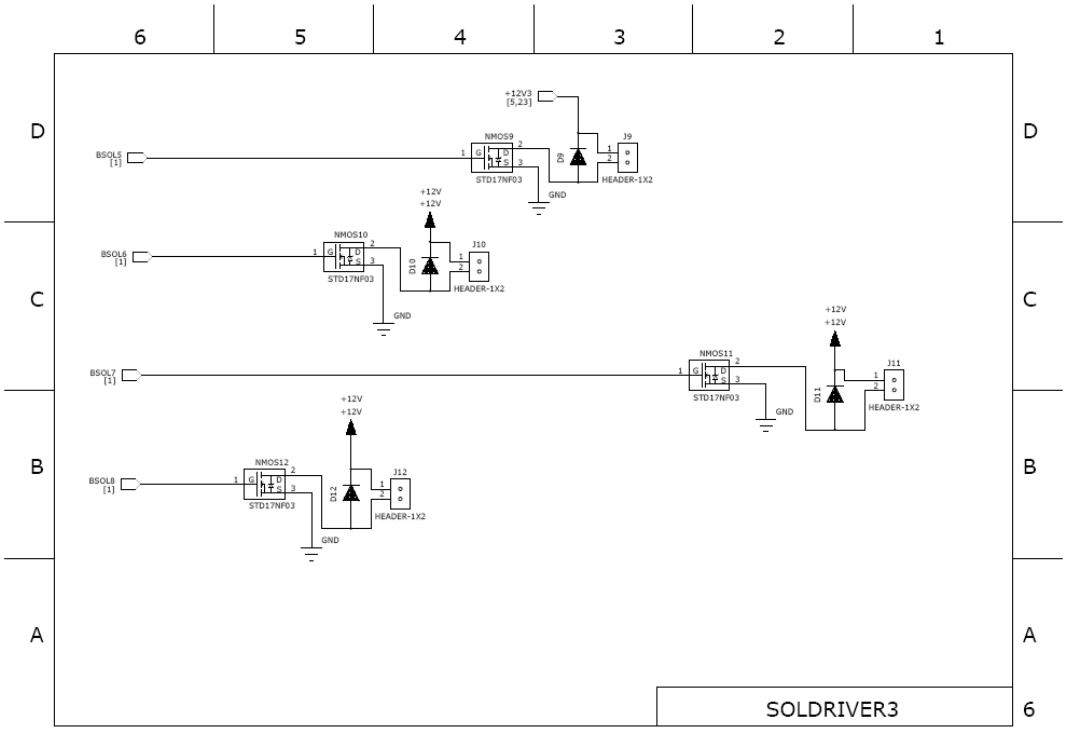


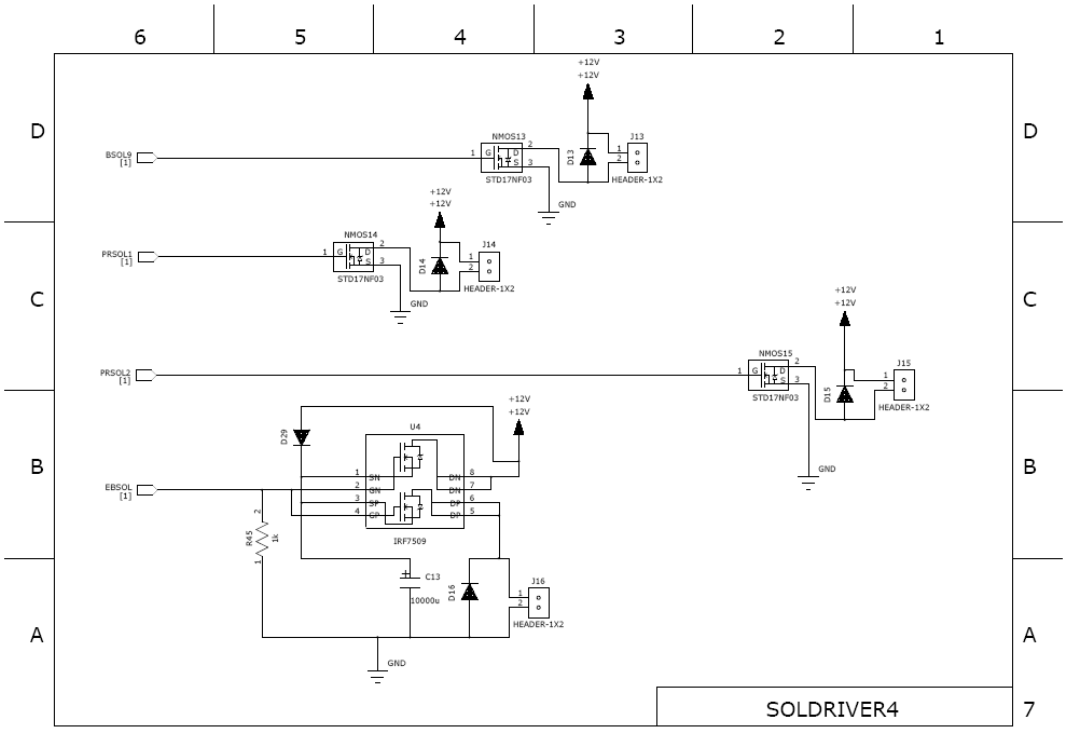

6
5
4
3
2
1
D
C
B
A
+3.3V
[3,11,12,13,14,15,16,17,20,21,23,25]
R20
10K
R21
10K
R22
10K
R23
10K
J17
HEADER-4X2POL
GND
HUBADDR0
[1]
HUBADDR1
[1]
HUBADDR2
[1]
HUBADDR3
[1]
HUBADDR
2

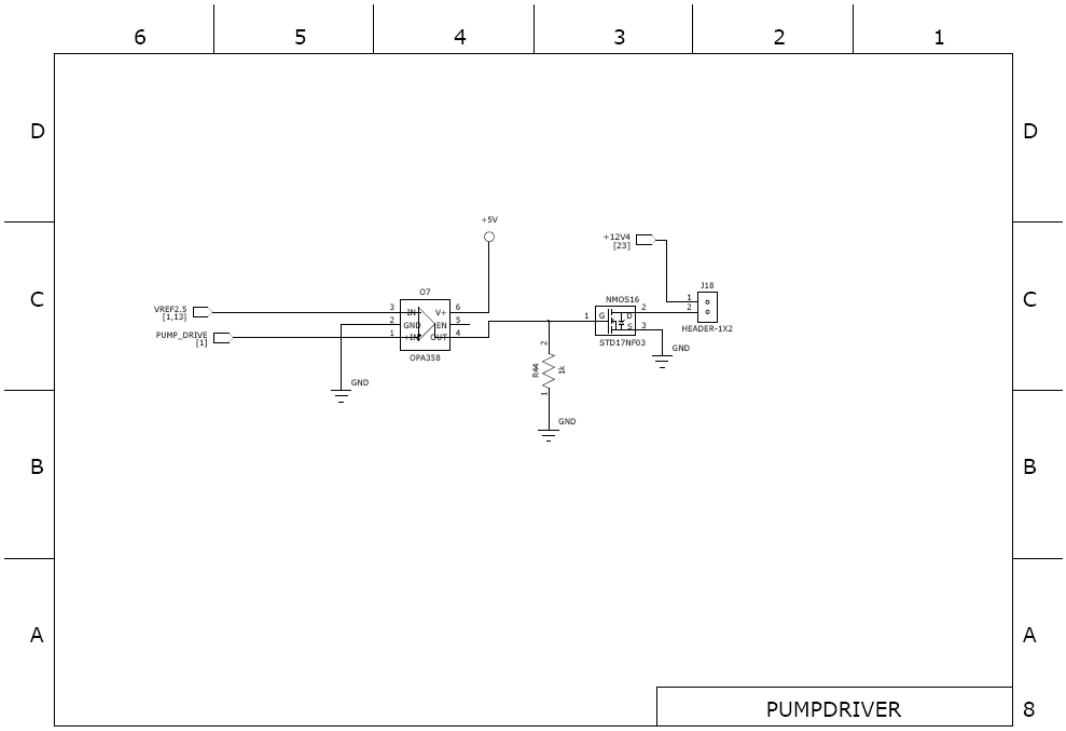

6
5
4
3
2
1
D
C
B
A
+3.3V
[2,3,11,12,13,14,15,16,17,20,21,23,25]
R16
499K
COMPX
[18]
M1
HMC1052
S/R+
[17]
S/R+
VCC
OUTB-
OUTA+
S/R-
GND1
GND2
OUTB+
OUTA-
GND
R6
5.1K
O3
OPA358
R14
5.1K
R15
499K
GND
VREF1.25
[3,13,14,15,16]
+3.3V
[2,3,11,12,13,14,15,16,17,20,21,23,25]
R29
499K
COMPY
[18]
R17
5.1K
O6
OPA358
R27
5.1K
R28
499K
MAGCOMP
3

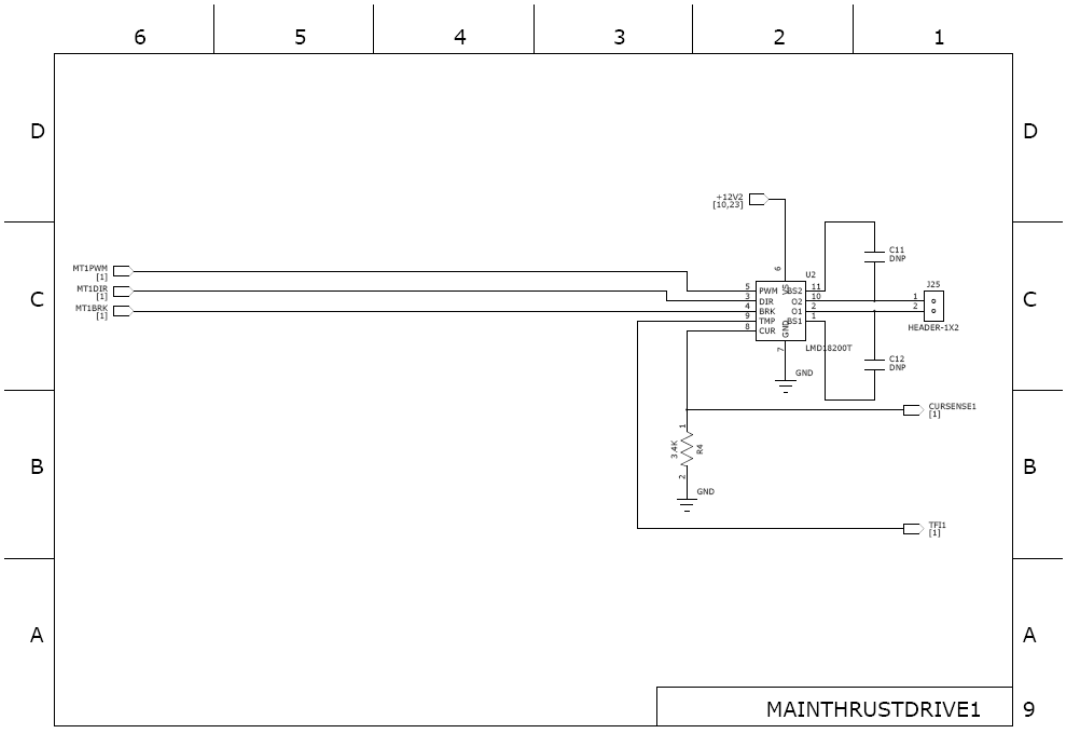

6
5
4
3
2
1
D
C
B
A
+12V
+12V
J1
D1
HEADER-1X2
NMOS1
STD17NF03
STAB1
[1]
GND
J2
D2
NMOS2
STAB2
J3
D3
NMOS3
STAB3
J4
D4
NMOS4
STAB4
SOLDRIVER1
4

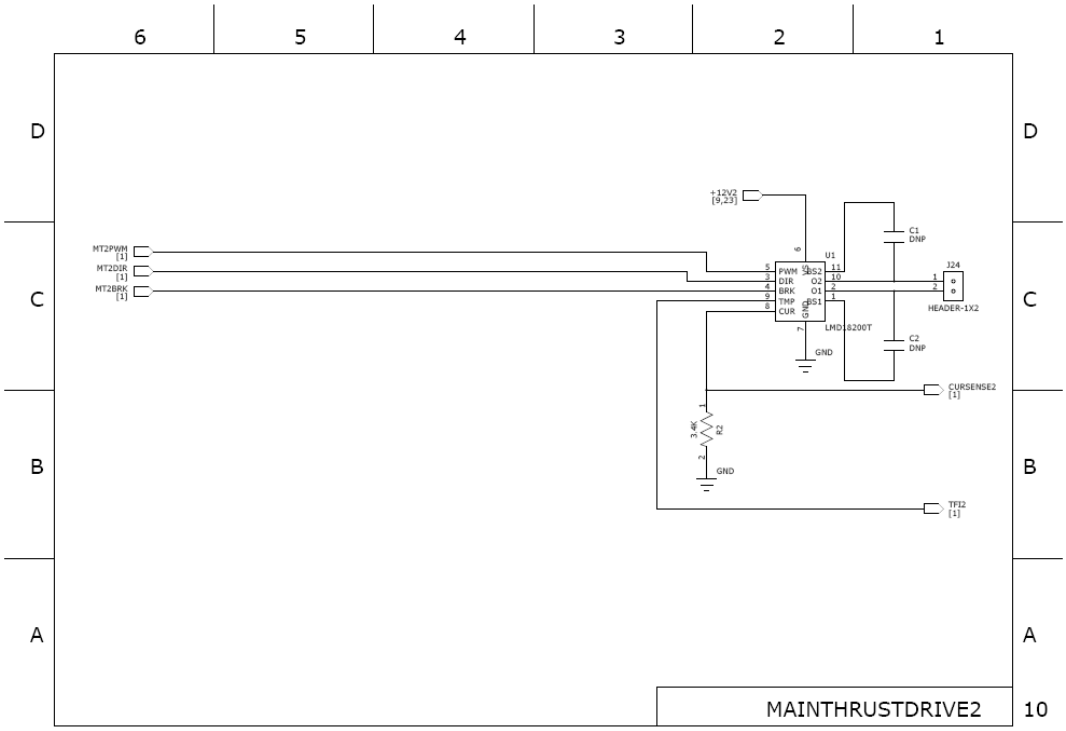

6
5
4
3
2
1
D
C
B
A
+12V3
[5,6,23]
J5
D5
HEADER-1X2
NMOS5
STD17NF03
BSOL1
[1]
GND
J6
D6
NMOS6
BSOL2
J7
D7
NMOS7
BSOL3
J8
D8
NMOS8
BSOL4
SOLDRIVER2
5

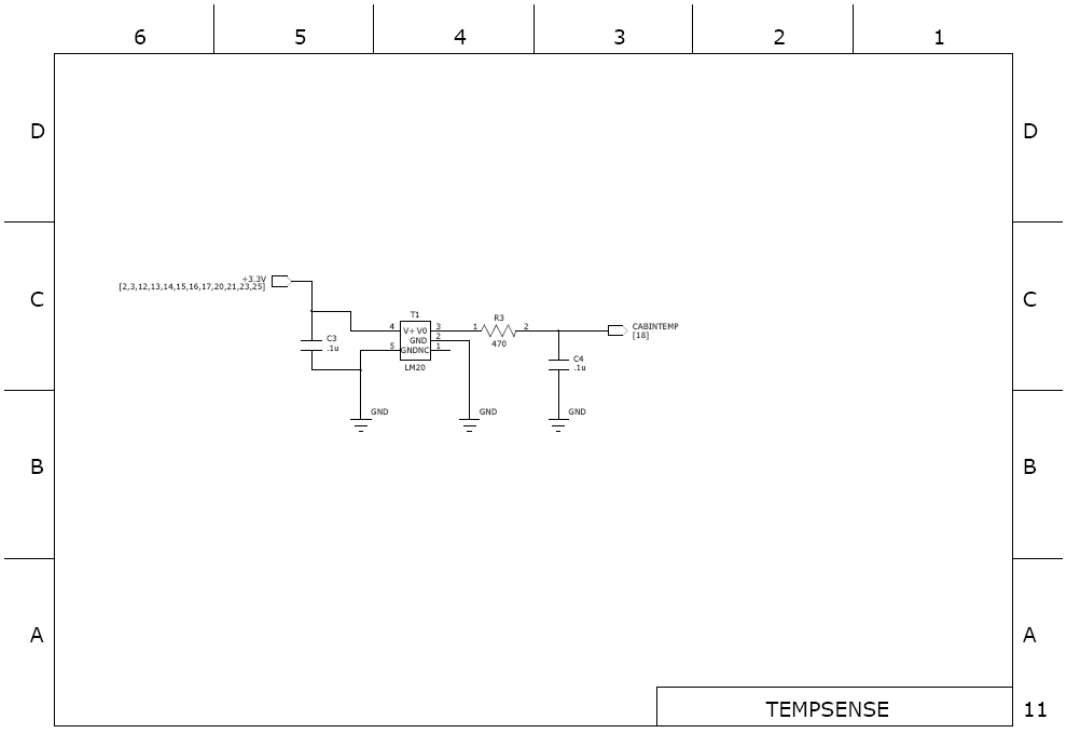

6
5
4
3
2
1
D
C
B
A
+12V3
[5,23]
BSOL5
[1]
NMOS9
STD17NF03
D9
J9
HEADER-1X2
GND
+12V
BSOL6
[1]
NMOS10
STD17NF03
D10
J10
HEADER-1X2
GND
BSOL7
[1]
NMOS11
STD17NF03
D11
J11
HEADER-1X2
GND
BSOL8
[1]
NMOS12
STD17NF03
D12
J12
HEADER-1X2
GND
SOLDRIVER3
6

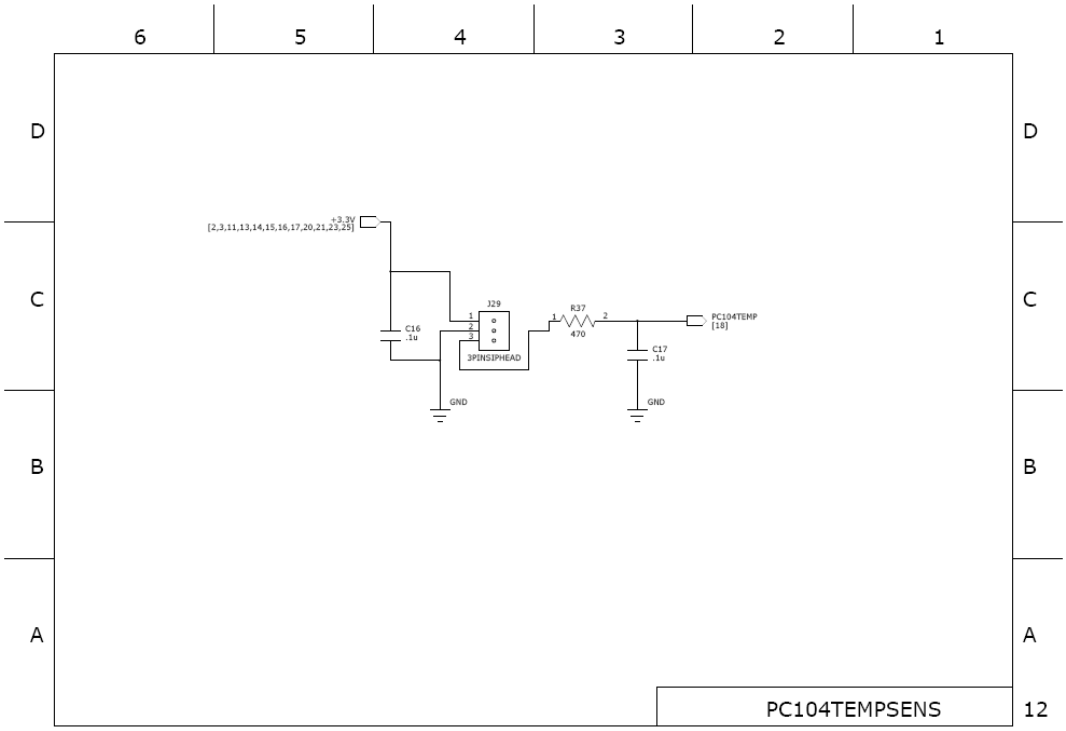

6
5
4
3
2
1
D
C
B
A
BSOL9
[1]
NMOS13
STD17NF03
D13
J13
HEADER-1X2
GND
+12V
PRSOL1
[1]
NMOS14
STD17NF03
D14
J14
HEADER-1X2
GND
PRSOL2
[1]
NMOS15
STD17NF03
D15
J15
HEADER-1X2
GND
EBSOL
[1]
D29
U4
IRF7509
R45
1K
C13
10000u
D16
J16
HEADER-1X2
GND
SOLDRIVER4
7

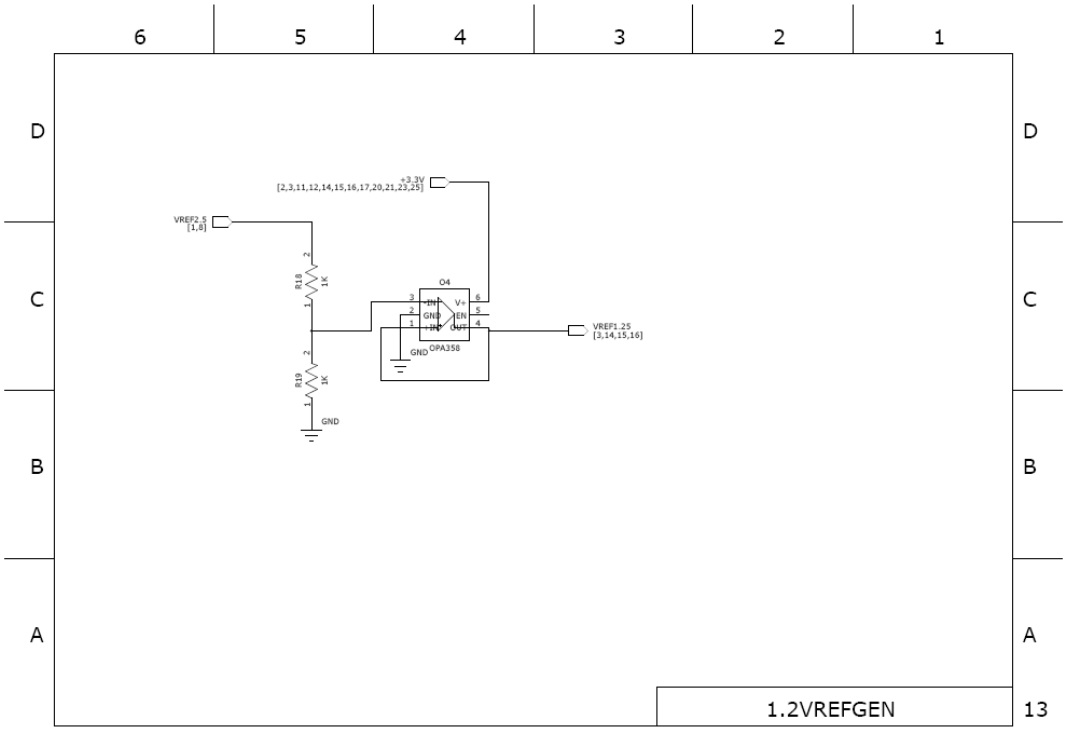

6
5
4
3
2
1
D
C
B
A
+5V
+12V4
[23]
VREF2.5
[1,13]
PUMP_DRIVE
[1]
O7
IN-
GND
IN+
V+
EN
OUT
OPA358
NMOS16
STD17NF03
J18
HEADER-1X2
GND
R44
1k
PUMPDRIVER
8

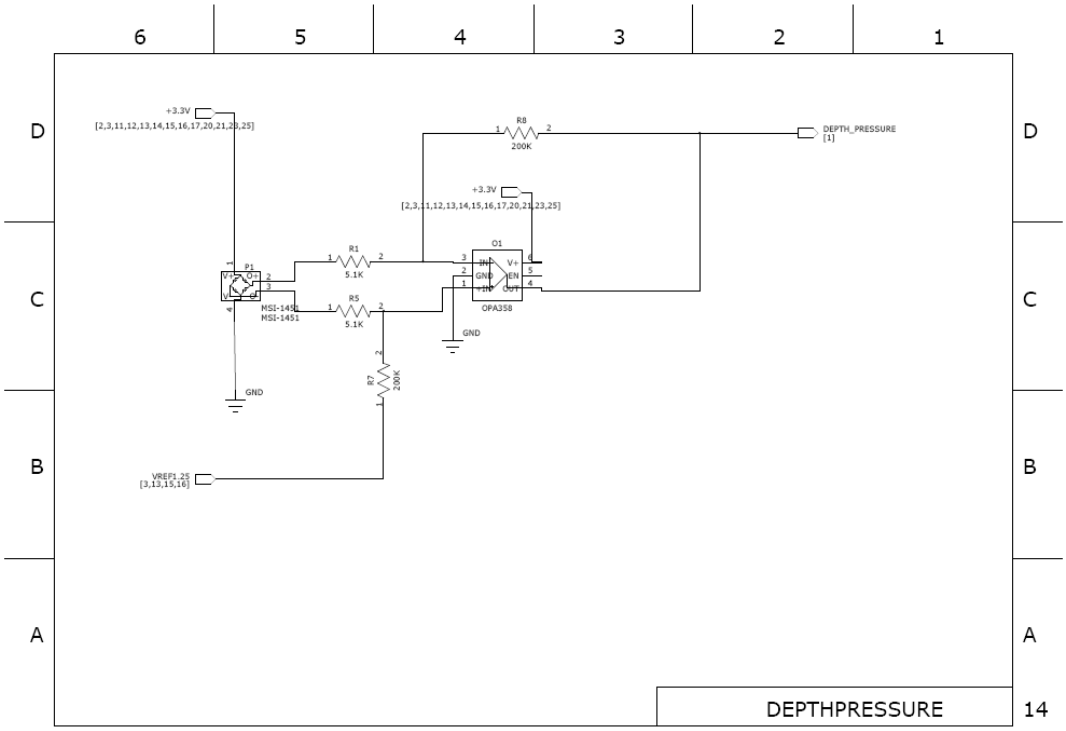

6
5
4
3
2
1
D
C
B
A
+12V2
[10,23]
MT1PWM
[1]
MT1DIR
[1]
MT1BRK
[1]
U2
PWM
DIR
BRK
TMP
CUR
VDD
VDD2
O2
O1
PGND
GND
LMD18200T
C11
DNP
C12
DNP
J25
HEADER-1X2
CURSENSE1
[1]
3.4K
R4
GND
TFI1
[1]
MAINTHRUSTDRIVE1
9

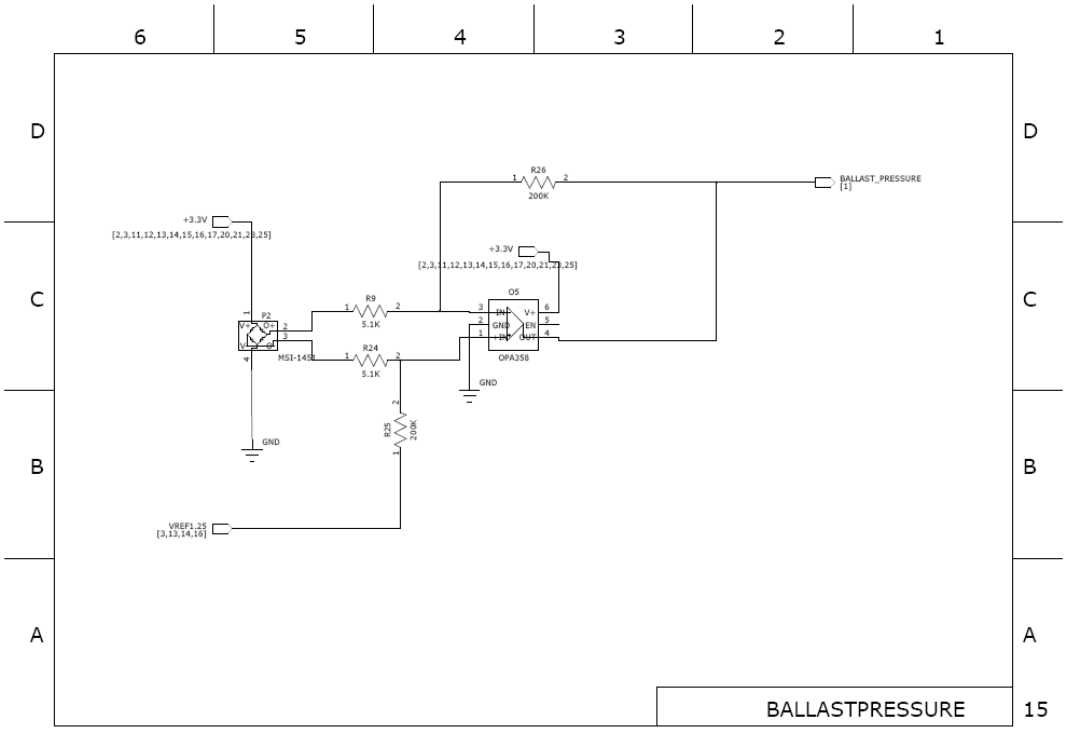

6
5
4
3
2
1
D
C
B
A
MT2PWM [1]
MT2DIR [1]
MT2BRK [1]
+12V2 [9,23]
U1
PWM
DIR
BRK
TMP
CUR
DS2
O2
O1
DS1
GND
LMD18200T
C1 DNP
C2 DNP
J24
HEADER-1X2
CURSENSE2 [1]
3.4K
R2
GND
TFI2 [1]
MAINTHRUSTDRIVE2
10

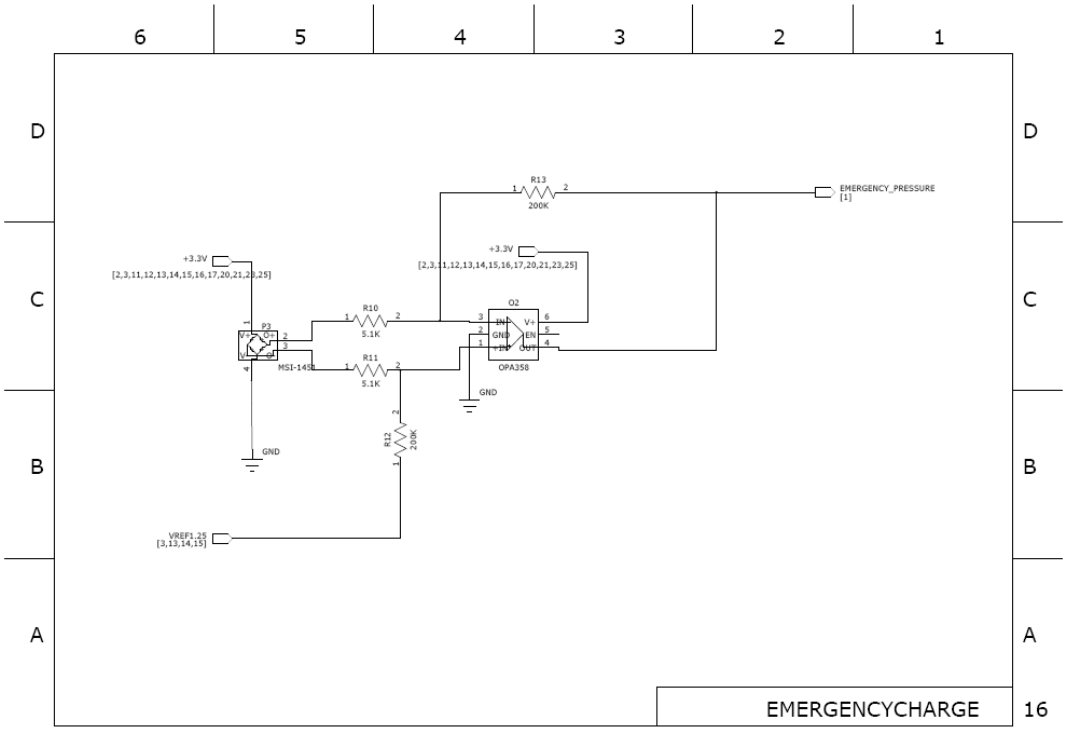

6
5
4
3
2
1
D
C
B
A
+3.3V [2,3,12,13,14,15,16,17,20,21,23,25]
T1
V+
VO
GND
GNDNC
LM20
C3 .1u
R3 470
C4 .1u
CABINTEMP [18]
GND
GND
GND
TEMPSENSE
11

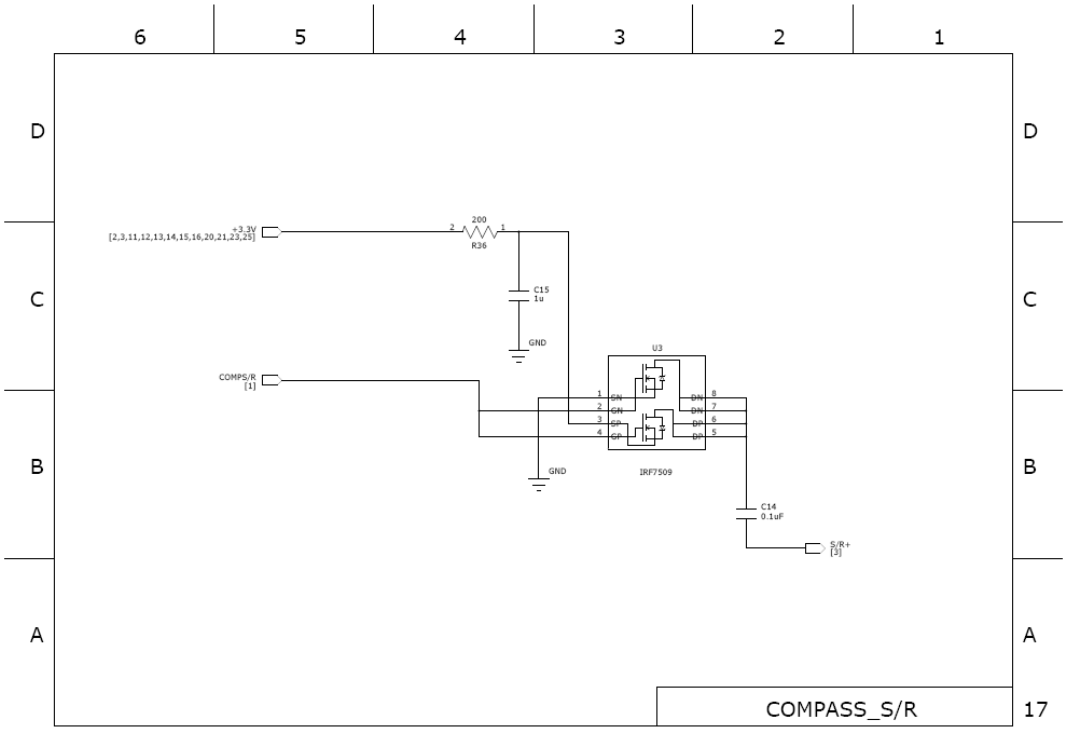

6
5
4
3
2
1
D
C
B
A
+3.3V
[2,3,11,13,14,15,16,17,20,21,23,25]
C16
.1u
J29
3PINSIPHEAD
R37
470
C17
.1u
PC104TEMP
[18]
GND
GND
PC104TEMPSENS
12

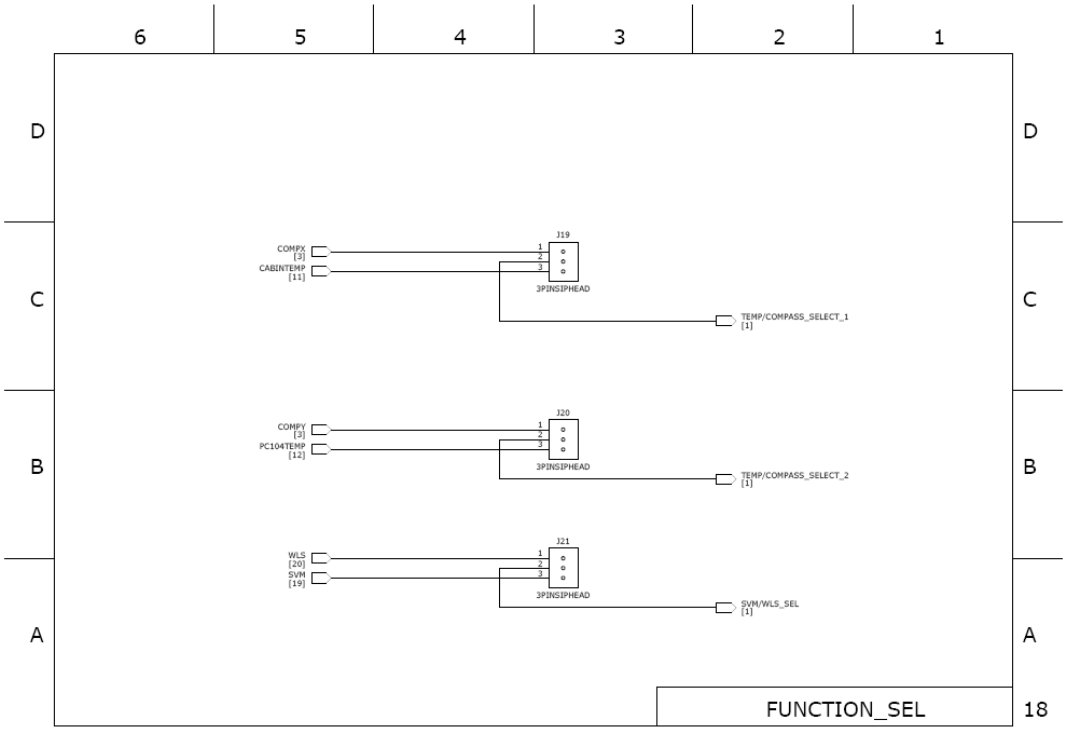

6
5
4
3
2
1
D
C
B
A
+3.3V
[2,3,11,12,14,15,16,17,20,21,23,25]
VREF2.5
[1,8]
R18
1K
R19
1K
O4
OPA358
VREF1.25
[3,14,15,16]
GND
GND
1.2VREFGEN
13

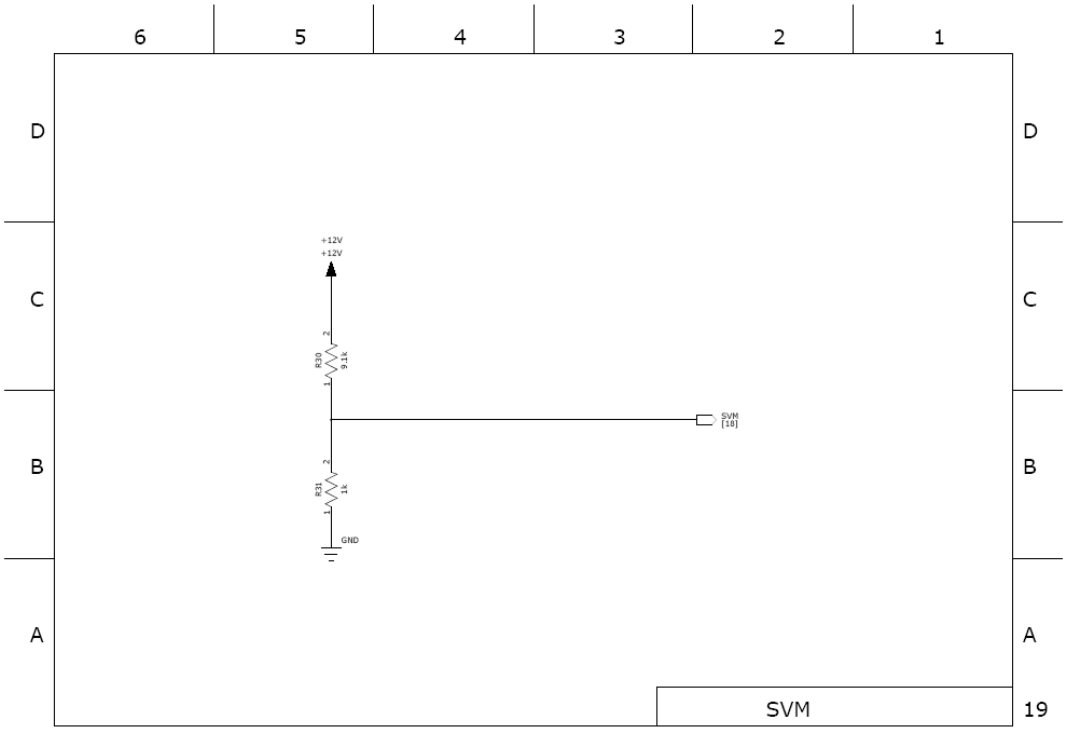

6
5
4
3
2
1
D
C
B
A
+3.3V
[2,3,11,12,13,14,15,16,17,20,21,23,25]
R8
200K
DEPTH_PRESSURE
[1]
+3.3V
[2,3,11,12,13,14,15,16,17,20,21,23,25]
O1
P1
R1
5.1K
MSI-1450
MSI-1451
R5
5.1K
OPA358
GND
R7
200K
GND
VREF1.25
[3,13,15,16]
DEPTHPRESSURE
14

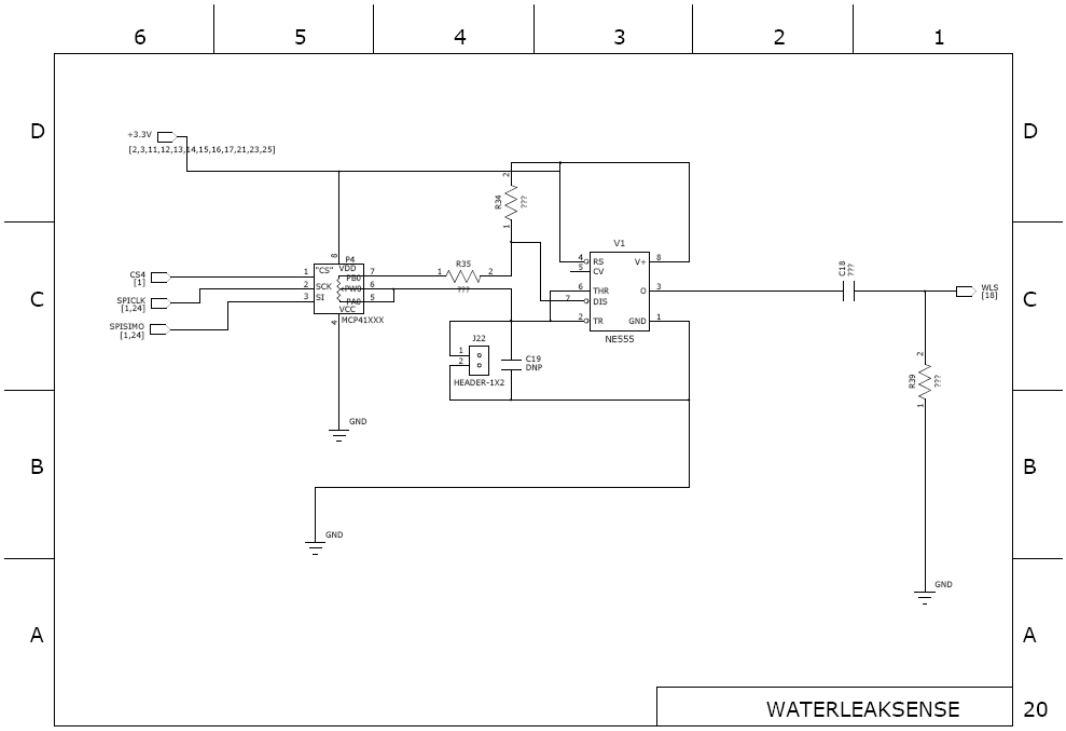

6
5
4
3
2
1
D
C
B
A
R26
200K
BALLAST_PRESSURE
[1]
+3.3V
[2,3,11,12,13,14,15,16,17,20,21,23,25]
+3.3V
[2,3,11,12,13,14,15,16,17,20,21,23,25]
O5
P2
R9
5.1K
MSI-1450
R24
5.1K
OPA358
GND
R25
200K
GND
VREF1.25
[3,13,14,16]
BALLASTPRESSURE
15

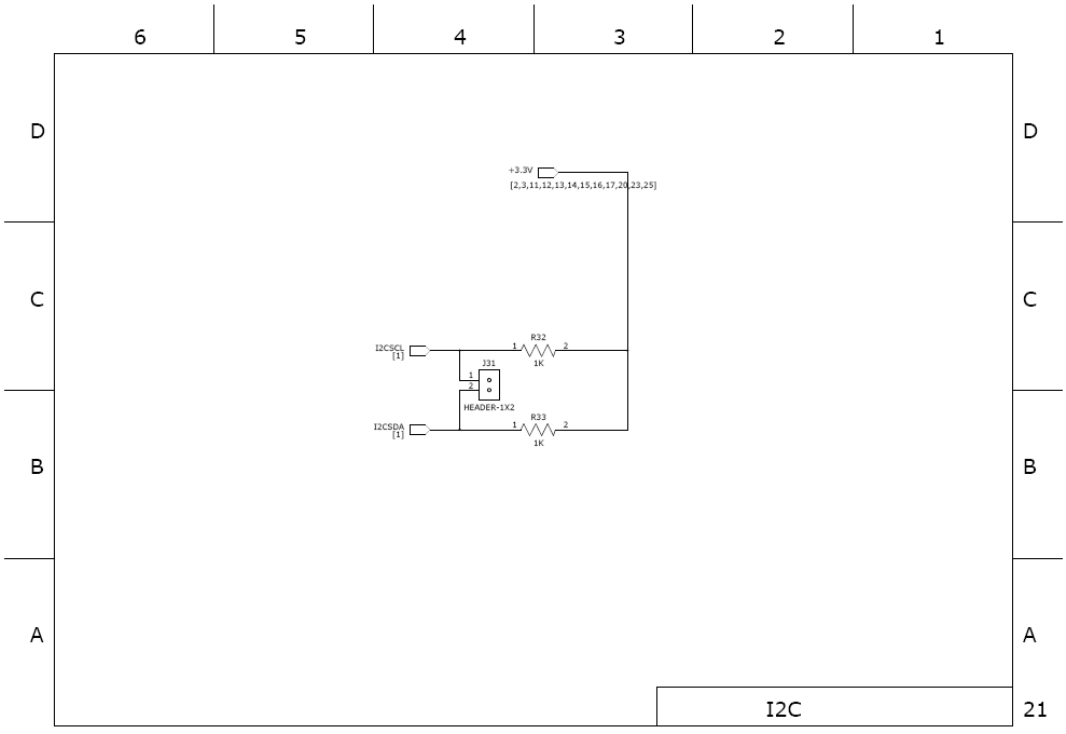

6
5
4
3
2
1
D
C
B
A
R13
200K
EMERGENCY_PRESSURE
[1]
+3.3V
[2,3,11,12,13,14,15,16,17,20,21,23,25]
R10
5.1K
Q2
MSI-1458
R11
5.1K
OPA358
GND
R12
200K
VREF1.25
[3,13,14,15]
EMERGENCYCHARGE
16

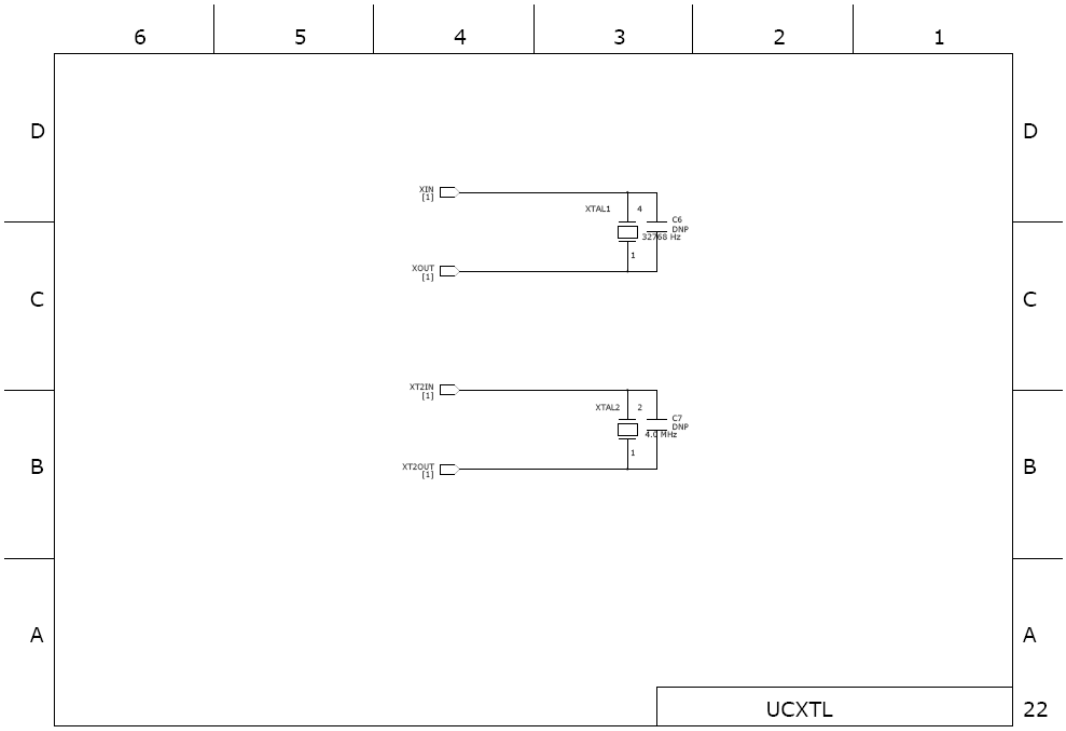

6
5
4
3
2
1
D
C
B
A
+3.3V
[2,3,11,12,13,14,15,16,20,21,23,25]
200
R36
C15
1u
GND
U3
COMPS/R
[1]
IRF7509
C14
0.1uF
S/R+
[3]
COMPASS_S/R
17

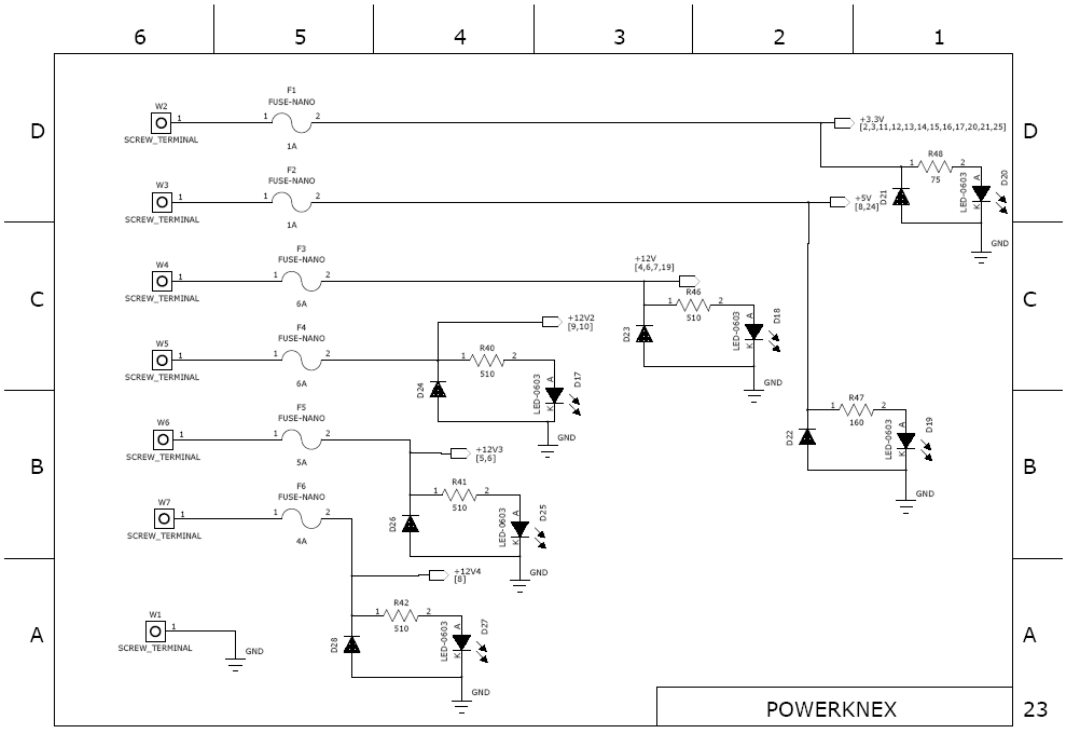

6
5
4
3
2
1
D
C
B
A
COMPX
[3]
CABINTEMP
[11]
J19
3PINSIPHEAD
TEMP/COMPASS_SELECT_1
[1]
COMPY
[3]
PC104TEMP
[12]
J20
3PINSIPHEAD
TEMP/COMPASS_SELECT_2
[1]
WLS
[20]
SVM
[19]
J21
3PINSIPHEAD
SVM/WLS_SEL
[1]
FUNCTION_SEL
18

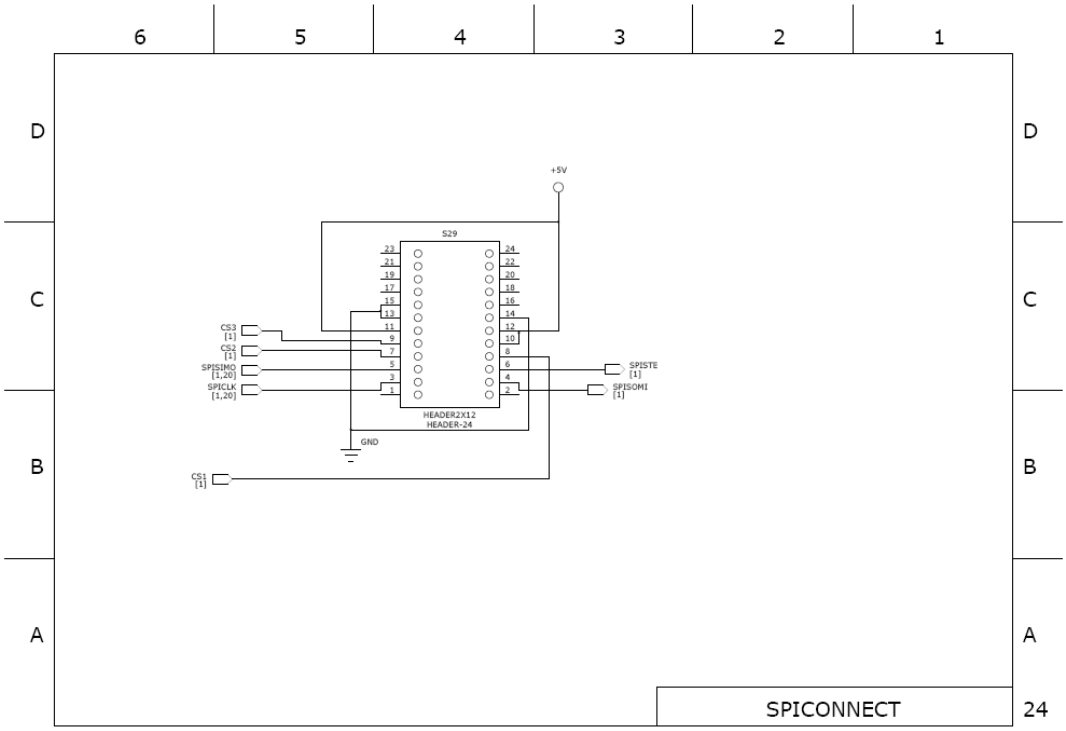

6
5
4
3
2
1
D
C
B
A
+12V
+12V
R30
9.1k
SVM
[18]
R31
1k
GND
SVM
19

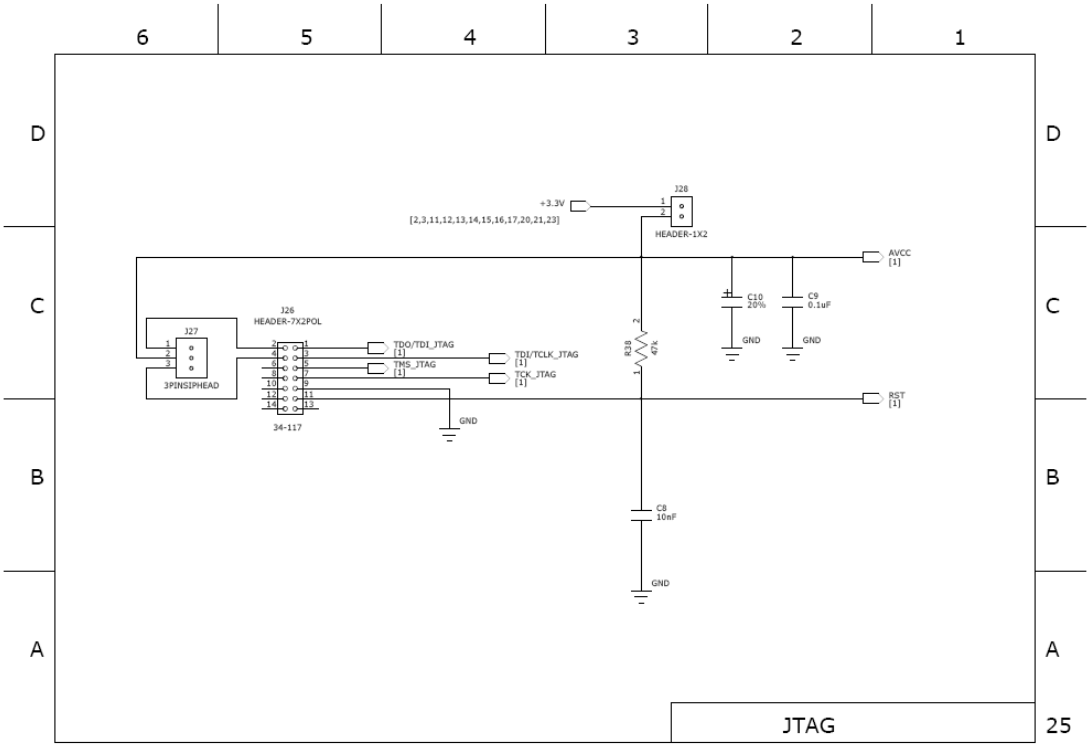

6
5
4
3
2
1
D
C
B
A
+3.3V
[2,3,11,12,13,14,15,16,17,21,23,25]
R34
???
V1
RS
CV
THR
DIS
TR
V+
Q
GND
NE555
P4
VDD
PB0
PW0
PA0
VCC
MCP41XXX
CS4
[1]
SPICLK
[1,24]
SPISIMO
[1,24]
SCK
SI
"CS"
R35
???
J22
HEADER-1X2
C19
DNP
C18
???
WLS
[18]
R39
???
GND
WATERLEAKSENSE
20

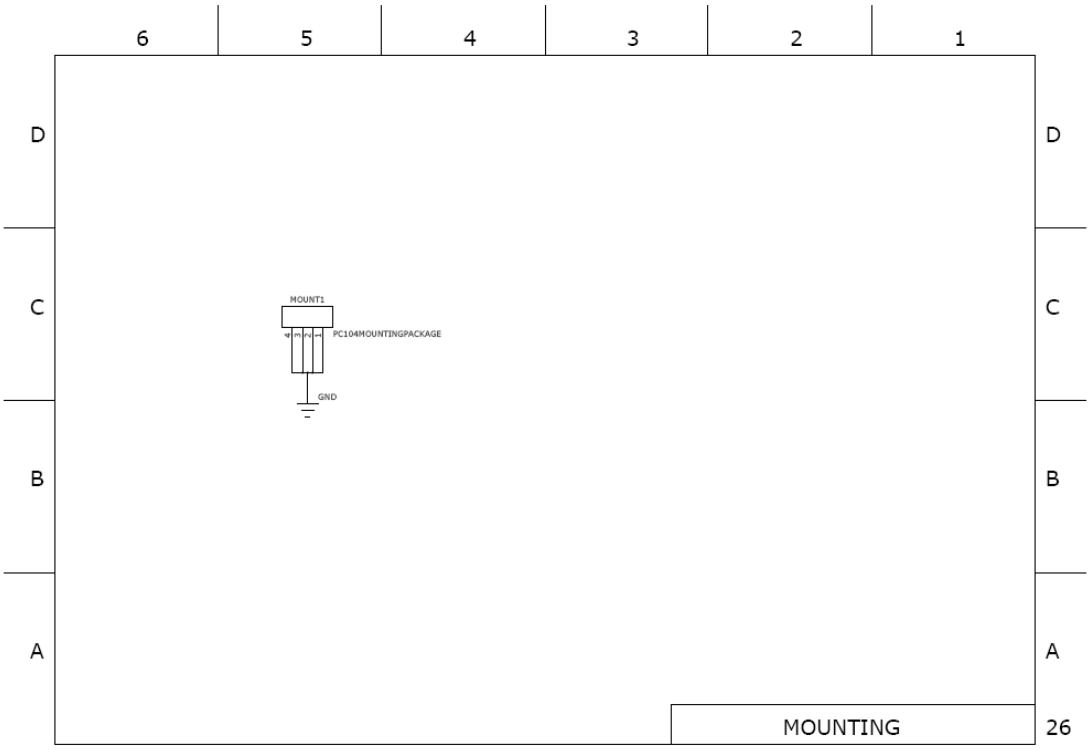

6
5
4
3
2
1
D
C
B
A
+3.3V
[2,3,11,12,13,14,15,16,17,20,23,25]
I2CSCL
[1]
I2CSDA
[1]
R32
1K
R33
1K
J31
HEADER-1X2
I2C
21


6
5
4
3
2
1
D
C
B
A
XIN
[1]
XOUT
[1]
XTAL1
4
C6
DNP
32.768 Hz
1
XT2IN
[1]
XT2OUT
[1]
XTAL2
2
C7
DNP
4.0 MHz
1
UCXTL
22


6
5
4
3
2
1
D
C
B
A
W2
SCREW_TERMINAL
F1
FUSE-NANO
1A
W3
SCREW_TERMINAL
F2
FUSE-NANO
1A
W4
SCREW_TERMINAL
F3
FUSE-NANO
6A
W5
SCREW_TERMINAL
F4
FUSE-NANO
6A
W6
SCREW_TERMINAL
F5
FUSE-NANO
5A
W7
SCREW_TERMINAL
F6
FUSE-NANO
4A
W1
SCREW_TERMINAL
GND
+3.3V
[2,3,11,12,13,14,15,16,17,20,21,25]
+5V
[8,24]
+12V
[4,6,7,19]
+12V2
[9,10]
+12V3
[5,6]
+12V4
[8]
R48
75
D21
LED-0603
D20
R46
510
D23
LED-0603
D18
R40
510
D24
LED-0603
D17
R47
160
D22
LED-0603
D19
R41
510
D26
LED-0603
D25
R42
510
D28
LED-0603
D27
POWERKNEX
23

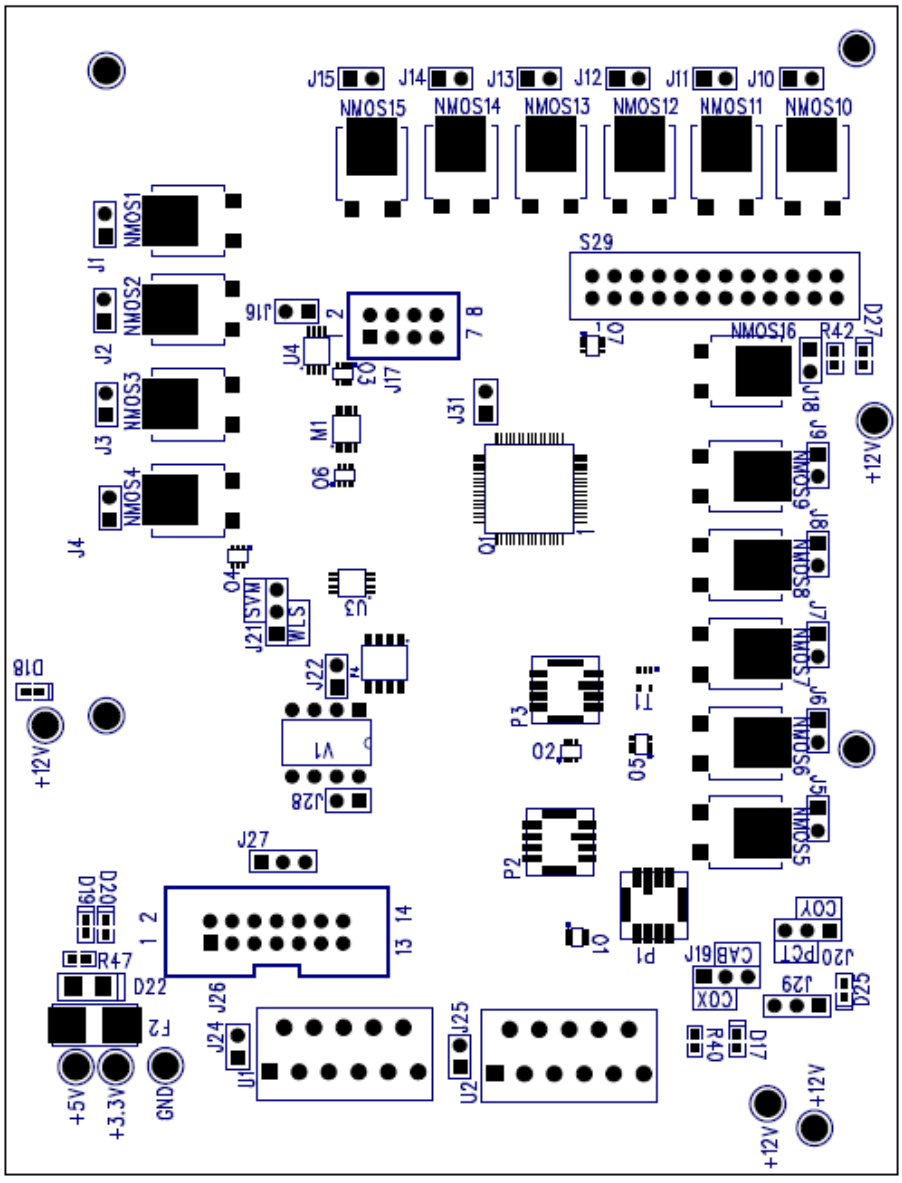

6
5
4
3
2
1
D
C
B
A
+5V
S29
CS3 [1]
CS2 [1]
SPISIMO [1,20]
SPICLK [1,20]
SPISTE [1]
SPISOMI [1]
HEADER2X12
HEADER-24
GND
CS1 [1]
SPICONNECT
24


6
5
4
3
2
1
D
C
B
A
+3.3V
[2,3,11,12,13,14,15,16,17,20,21,23]
J28
HEADER-1X2
AVCC [1]
C10 20%
C9 0.1uF
GND
J26
HEADER-7X2POL
J27
3PINSIPHEAD
TDO/TDI_JTAG [1]
TMS_JTAG [1]
TDI/TCLK_JTAG [1]
TCK_JTAG [1]
R38 47k
RST [1]
34-117
C8 10nF
JTAG
25


6
5
4
3
2
1
D
C
B
A
MOUNT1
PC104MOUNTINGPACKAGE
GND
MOUNTING
26

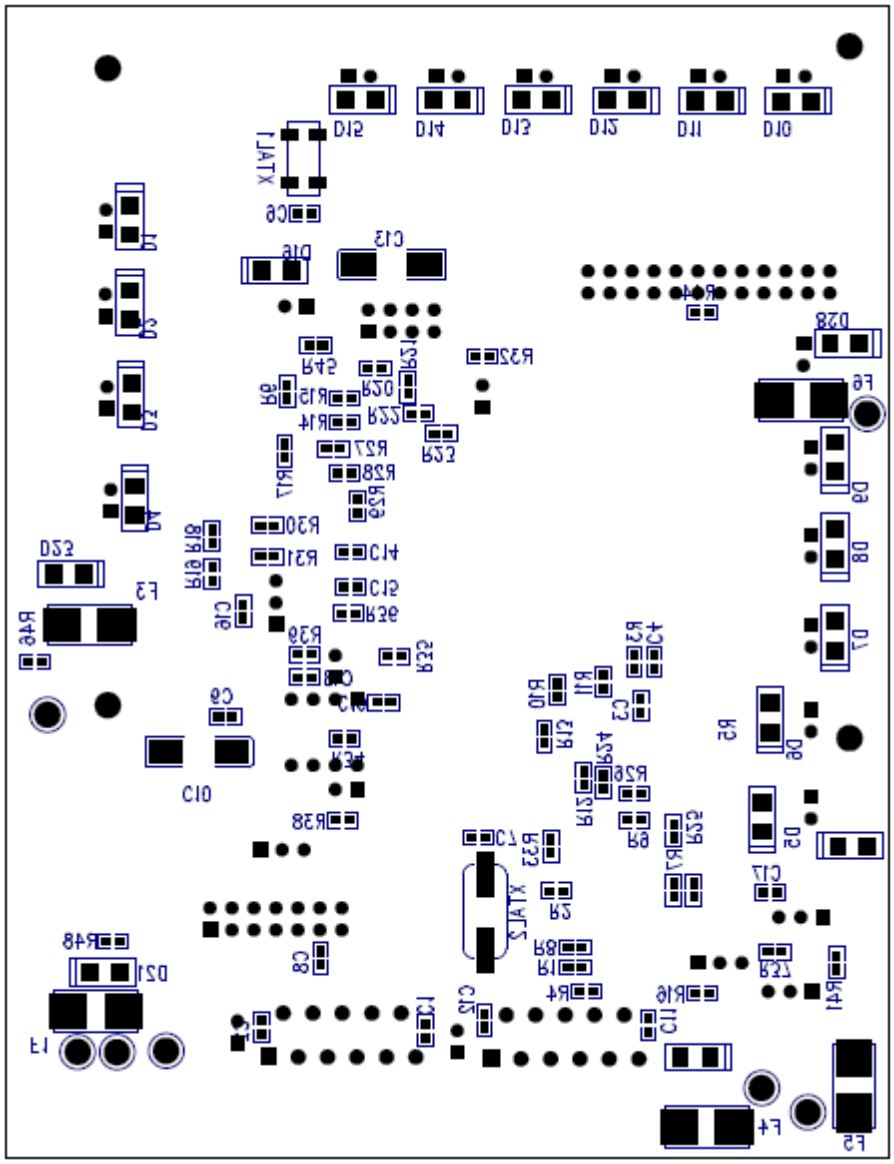
A.2 Sub-Hub PCB Design: Top View

This is a top view of the PCB showing the component pads and the silkscreen (which shows reference designators and component outlines).



A.3 Sub-Hub PCB Design: Bottom View

This is a bottom view of the PCB showing the component pads and the silkscreen (which shows reference designators and component outlines).



A.4 Sub-Hub PCB Parts List

Item	Qty	Reference	Manufacturer	Description	Value	Part Name
1	5	J19-21 J27 J29	General	1 X 3 PIN HEADER, 0.1" CENTERS		3PINSIPHEAD
2	4	C3-4 C16-17	Murata	Monolithic Ceramic Chip Capacitor SMD 0603	.1u	CAP0603,.1u
3	2	C9 C14	Murata	Monolithic Ceramic Chip Capacitor SMD 0603	0.1uF	CAP0603,0.1uF
4	1	C8	Murata	Monolithic Ceramic Chip Capacitor SMD 0603	10nF	CAP0603,10nF
5	1	C15	Murata	Monolithic Ceramic Chip Capacitor SMD 0603	1u	CAP0603,1u
6	1	C18	Murata	Monolithic Ceramic Chip Capacitor SMD 0603	???	CAP0603,???
7	7	C1-2 C6-7 C11-12 C19	Murata	Monolithic Ceramic Chip Capacitor SMD 0603	DNP	CAP0603,DNP
8	1	C13	Xicon	Mini Alum Elect Cap	10000u	CAP-ELECTAA,10000u,20%
9	1	C10	NIchicon	SMT Aluminum Electrolytic Capacitor	10u	CAP-ELECTAA,10u,20%
10	23	D1-16 D21-24 D26 D28-	VISHAY	Surface Mount Fast Switching		DIO-VISHAY_RS1ATHRUK

		29		Rectifier		
11	2	F1-2	Littelfuse	FUSE, SURF MOUNT, 0603 FUSES	1A	FUSE-NANO,1A
12	**1**	**F6**	**Littelfuse**	**FUSE, SURF MOUNT, 0603 FUSES**	**4A**	**FUSE-NANO,4A**
13	1	F5	Littelfuse	FUSE, SURF MOUNT, 0603 FUSES	5A	FUSE-NANO,5A
14	2	F3-4	Littelfuse	FUSE, SURF MOUNT, 0603 FUSES	6A	FUSE-NANO,6A
15	22	J1-16 J18 J22 J24-25 J28 J31	General	1 X 2 PIN HEADER, 0.1" CENTERS		HEADER-1X2
16	1	J17	General	4 X 2 PIN POLARIZED HEADER, 0.1" CENTERS		HEADER-4X2POL
17	1	J26	3M	7 X 2 PIN POLARIZED HEADER, 0.1" CENTERS		HEADER-7X2POL
18	1	S29	General	12 X 2 PIN HEADER, 0.1" CENTERS		HEADER2X12
19	1	M1	Honeywell	2-Axis Magnetic Sensor		HMC1052
20	2	U3-4	International Rectifier	HEXFET Power CMosfet		IRF7509
21	6	D17-20 D25 D27	Kingbright	LIGHT EMITTING DIODE		LED-0603
22	1	T1	National Semiconductor	2.4V, 10uA, SC70 miscro SMD Temp Sens		LM20

23	2	U1-2	National Semiconductor	3A, 55V H-Bridge		LMD18200T
24	1	P4	Microchip	8 Bit Single SPI Digital Pot with Volatile Memory	10K	MCP41XXX,10K
25	3	P1-3	Measurement Specialties	PC Board Mountable Pressure Sensor		MSI-1451
26	1	Q1	Texas Instruments	IC MCU 16BIT 8K FLASH 64-LQFP		MSP430F23X
27	1	V1	SIGNETICS	Timer Block		NE555
28	7	O1-7	Texas Instruments	3V Single Supply High-Speed Op Amp SMD SC70		OPA358
29	1	MOUNT1	n/a	PC104 Form Mounting Hole Array	M-40 Screw Hole	PC104MOUNTINGPACKAGE,M-40 Screw Hole
30	4	R20-23	KOA Speer Electronics	Thick Film Chip Resistor SMD 0603	10K	RES0603,10K,0.5%
31	1	R47	KOA Speer Electronics	Thick Film Chip Resistor SMD 0603	160	RES0603,160,0.5%
32	4	R18-19 R32-33	KOA Speer Electronics	Thick Film Chip Resistor SMD 0603	1K	RES0603,1K,0.5%
33	3	R31 R44-45	KOA Speer Electronics	Thick Film Chip Resistor SMD 0603	1k	RES0603,1k,0.5%
34	1	R36	KOA Speer Electronics	Thick Film Chip Resistor SMD 0603	200	RES0603,200,1%

35	6	R7-8 R12-13 R25-26	KOA Speer Electronics	Thick Film Chip Resistor SMD 0603	200K	RES0603,200K,1%
36	2	R2 R4	KOA Speer Electronics	Thick Film Chip Resistor SMD 0603	3.4K	RES0603,3.4K,1%
37	2	R3 R37	KOA Speer Electronics	Thick Film Chip Resistor SMD 0603	470	RES0603,470,1%
38	1	R38	KOA Speer Electronics	Thick Film Chip Resistor SMD 0603	47k	RES0603,47k,0.5%
39	4	R15-16 R28-29	KOA Speer Electronics	Thick Film Chip Resistor SMD 0603	499K	RES0603,499K,1%
40	10	R1 R5-6 R9-11 R14 R17 R24 R27	KOA Speer Electronics	Thick Film Chip Resistor SMD 0603	5.1K	RES0603,5.1K,0.5%
41	4	R40-42 R46	KOA Speer Electronics	Thick Film Chip Resistor SMD 0603	510	RES0603,510,0.5%
42	1	R48	KOA Speer Electronics	Thick Film Chip Resistor SMD 0603	75	RES0603,75,0.5%
43	1	R30	KOA Speer Electronics	Thick Film Chip Resistor SMD 0603	9.1k	RES0603,9.1k,1%
44	3	R34-35 R39	KOA Speer Electronics	Thick Film Chip Resistor SMD 0603	???	RES0603,???,0.5%
45	7	W1-7	n/a	Screw Holes For Power Supply Connectoins	M-40 Screw Hole	SCREW_TERMINAL,M-40 Screw Hole
46	16	NMOS1-16	ST Microelectronics	N-channel 30V, 17A DPAK		STD17NF03

				POWER MOSFET		
47	1	XTAL2	Vishay	Low Profile SMD Type Crystal Unit	4.0 MHz	XTAL1,4.0 MHz
48	1	XTAL1	ECS Inc. International	SMD Tuning Fork Crystal	32768 Hz	XTALECX306X,32768 Hz

A.5 Sub-Hub Code Examples

A.5.1 Sub-Hub Header File, hub_defv1_0.h

```
#include "msp430x23x.h"      // Definitions, constants, etc for msp430F223
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <in430.h>
/* Stabilization thruster definitions
 * Numbered as follows:
 * 1 : Fore, Starboard
 * 2 : Fore, Port
 * 3 : Aft,  Starboard
 * 4 : Aft,  Port
 */
#define st_off  0x00 //F0000 0
#define st_1    0x01 //F0001 1
#define st_2    0x02 //F0010 2
#define st_3    0x04 //F0100 4
#define st_4    0x08 //F1000 8

#define st_12   0x03 //F0011 3
#define st_13   0x05 //F0101 5
#define st_14   0x09 //F1001 9
#define st_23   0x06 //F0110 6
#define st_24   0x0A //F1010 10
#define st_34   0x0C //F1100 12

#define st_123  0x07 //F0111 7
#define st_234  0x0E //F1110 14
#define st_134  0x0B //F1011 11
```

```
#define st_124  0x0D //F1101 13

#define st_1234 0x0F //F1111 15

/* Ballast Operation Definitions
 * Specified as follows:
 * b(ballast)_tank(fore, aft or all)_op(fill or purge)
 */
#define b_off         0x00 //F0000 0
#define b_fore_fill   0x01 //F0001 1
#define b_fore_purge  0x02 //F0010 2
#define b_aft_fill    0x04 //F0100 4
#define b_aft_purge   0x08 //F1000 8
#define b_all_fill    0x0A //F0101 10
#define b_all_purge   0x05 //F1010 5
/* Function Definitions */
//hub_setup: To be called at startup
//Sets up P3.6,7;P4.0,1; Returns jumper
//selected hub address (int 0-15) or error (int 16).
int hub_setup (void);
int i2c_setup (void);
void sol_control(char sol_call);
void solenoid_setup(void);
```

A.5.2 Sub-Hub Function File, hub_funcv1_0.c

```
#include "hub_defv1_0.h"

void solenoid_setup(void){
  P5SEL &= 0xF0; // Set P5.3:0 to I/O-Stab Sol
  P5DIR |= 0x0F; // Set P5.3:0 to out
  P5OUT &= 0xFF; // Set outputs high
  P1SEL &= 0xF0; //Set P1.3:0 to I/O - Ballast Sol
  P1DIR |= 0x0F; // Set P1.3:0 to out
  P1OUT &= 0xF0; //Set outputs low
}
int hub_setup (void) {
```

```
  int space;
  int address = 0x00;
  P3SEL  &= ~(BIT7|BIT6); // Hub Addr 0,1 (P3.6,7) = I/O
  P3DIR  &= ~(BIT7|BIT6); // Hub Addr 0,1 (P3.6,7) = Input
  P4SEL  &= ~(BIT1|BIT0); // Hub Addr 2,3 (P4.0,1) = I/O
  P4DIR  &= ~(BIT1|BIT0); // Hub Addr 2,3 (P4.0,1) = Input
  /* Doctor Input For Easier Future Expansion */
  space  = ~P3IN&(BIT7|BIT6);
  space |= ~P4IN&(BIT1|BIT0);
  address |= (space>>3)&(BIT3);//D3
  address |= (space>>5)&(BIT2);//D2
  address |= (space>>1)&(BIT0);//D0
  address |= (space<<1)&(BIT1);//D1
  address &= (BIT3|BIT2|BIT1|BIT0);//clean it up

  if (address >= 0x10) //if address is > F set to error flag
    address = 0x10;

  switch (address) { //returns correct hub address
    case 0x08:
      return 4;
    case 0x04:
      return 3;
    case 0x02:
      return 2;
    case 0x01:
      return 1;
    case 0x10:
      return 16;
    default:
      return 16;
  }
}
int i2c_setup (void) {
  UCB0CTL1 |= UCSWRST; //set RST bit
  UCB0CTL0 |= ~UCA10 || ~UCSLA10 || UCMM || ~UCMST ||
    UCMODE0 || UCMODE1 || UCSYNC;  //init all registers
```

```
  //7 bit addr, multi master, slave, I2C mode, Snchronous
 UCB0BR0 = 0x20; //Clock divided by 32
 UCB0BR1 = 0x00;
  //slave TX/RX, sit in RX
 UCB0CTL1 |= UCSSEL0 || UCSSEL1 || ~UCTR;
 return 1;
}
void sol_control(char sol_call) {
 switch (sol_call) {
    case '1':
      P5OUT |= st_1;
      break;
    case '2':
      P5OUT |= st_2;
      break;
    case '3':
      P5OUT |= st_3;
      break;
    case '4':
      P5OUT |= st_4;
      break;
    case 'a':
      P5OUT |= st_34;
      break;
    case 'f':
      P5OUT |= st_12;
      break;
    case 'd':
      P5OUT |= st_24;
      break;
    case 's':
      P5OUT |= st_13;
      break;
    case 'e':
      P5OUT |= st_1234;
      break;
    case 'j':
```

```
      P1OUT |= b_fore_fill;
      break;
    case 'k':
      P1OUT |= b_fore_purge;
      break;
    case ';':
      P1OUT |= b_aft_fill;
      break;
    case 'l':
      P1OUT |= b_aft_purge;
      break;
    case 'D':
      P1OUT |= b_all_fill;
      break;
    case 'P':
      P1OUT |= b_all_purge;
      break;
    default :
      P5OUT &= st_off;
      P1OUT &= b_off;
      break;
    }
}
```

A.5.3 Hub-Addressing Code

This code shows a very simple example which reads the hub address and returns it. This function is meant to be used in the main function. The returned hub address can then be used in a case statement to select and run a hub specific program.

```
#include "hub_funcv1_0.c"
void main() {
  int i;
  i = hub_setup(); //setup and return hub address
  __no_operation(); //Allows user to set breakpoint and read hub value i
}
```

A.5.4 Cabin Temperature Sensor Read Code

This code sets up the ADC to read the cabin temperature sensor and then reads the value. Ensure that pins 2 and 3 are jumpered on header J19 (to select cabin temperature).

```
#include "hub_funcv1_0.c"

/*---------------------- Main Function ----------------------*/
void main() {
 int i;
 float j;
 while (1) {
 //Temp sensor specific
 P6SEL = 0x01; //Configure Analog 0 for analog input

 //ADC Setup, make function to decide which channels to poll?
 while((ADC12CTL1) == ADC12BUSY); //check that ADC is not busy
 ADC12CTL0 &= !ENC; //disable ADC
 ADC12MCTL0 = 0x10; //USE REF2_5-AVss, A.0
 ADC12CTL0 |= REFON|REF2_5V|ADC12ON; //turn 2.5V reference on and turn ADC On

 while((ADC12CTL1) == ADC12BUSY); //check that ADC is not busy
 ADC12CTL0 |= (ENC|ADC12SC); // enable and start conversion
 ADC12CTL0 &= (!ADC12SC)|ENC; //clear conversion start bit
 while((ADC12CTL1) == ADC12BUSY); //check busy
 j = ((-100/1.1889)*((ADC12MEM0/4095)*2.5))+157.4; //calculate temperature
 }
}
```

A.6 Sub-Hub System V1.0 Device Errata Sheet

As is common with first revisions of such devices, a number of errors were found in the first revision of the Sub-Hub System device. This section outlines errors in both the circuit schematics and the PCB design as well as suggested methods to work around these errors. Suggested changes for better operation are also included.

A.6.1 Ballast, Depth and Emergency Charge Pressure Sensor: P1, P2, P3

The pressure sensor pins were mapped incorrectly to the package footprints. Pins 1 and 2 as well as 3 and 4 should be swapped.

The suggested workaround is to mount the package backwards (with pins 5-8 of the IC soldered to pins 1-4 of the footprint). As pins 5-8 are purely mechanical mounting pads and are not actually connected in the IC or on the PCB jumper wires can then be soldered across the chip to correctly wire the component.

A.6.2 1.2V Reference Generator: O4

Pins 1 and 3 on the op-amp O4 were labeled in the schematic. This causes a positive feedback amplifier rather than a unity gain buffer. The actual pin numbering scheme should start at pin 1 and advance counterclockwise around the package body. References are made with respect to the correct pin numbering (as seen on the IC itself).

The workaround for this problem is slightly more delicate. To use the 1.25V reference generator the trace between package pins 1 and 4 must be removed. The long trace connected to pin 1 must also be broken. This should be done after the first via leaving a trace and via connected to pin 1 but disconnected from any other circuitry. Finally, the short trace connected to pin 3 must be broken. The correct pin connection between pin 1 and the resistive voltage divider can be made by connecting the via just above the O4 footprint (formally connected to pin 3) and the via along the trace connected to pin 1. Pins 3 and 4 can be connected across the package body. Either pin 3 or pin 4 must then be connected to the next via along the trace formerly connected to pin 1.

A.6.3 Magnetic Compass: M1

The footprint provided for device M1, the Honeywell HMC1052 2-axis magnetic compass is for the 10-pin MSSOP package. This IC is currently only available in the 16-pin LLC package.

The suggested workaround for this error is to glue the top of the 16-pin LLC package chip to a free space on the PCB and then run jumper wires from the corresponding chip pins to the pad pins on the PCB.

A.6.4 3.3V and 5V Power Supply Connections: W1 and W2

The silkscreen designators for the 3.3V and 5V power supply connections were switched. This means that the 3.3V power supply should be connected to the board screw terminal labeled +5V and vice versa.

A.6.5 Vishay RS1B Diode Silkscreen: D1-16, D21-24, D26, D28-29

The silkscreen polarity identifier on diodes the Vishay RS1B diodes was reversed. Therefore the cathode pin of the diodes should be connected to the side of the footprint designated for anode connection.

A.6.6 Solder Mask Missing: M1, P4

The solder mask for components M1 (magnetic compass sensor) and P4 (digital potentiometer for the water leak sensor circuit) are missing.

The workaround for this is to simply scrape off the solder mask with a hobby knife and cleaning the area with isopropyl alcohol.

Appendix B. Power Consumption

Table 13 shows the individual contributions of each component to the overall power consumption. Some components such as the main thrusters are shown in groups including all components pertaining to those peripherals (with the individual component contributions outlined as well). The "PS E%" column shows the power supply rail efficiency for the voltage specified. "Max Power" shows the absolute maximum power consumption of each component. "ExpectedP" shows the reasonable expected maximum power consumption of each component. The percent contribution of individual components and groups is shown in the last column ("% Total").

The total maximum power that could be consumed by the vehicle is found to be 530 watts. This however is a completely unreasonable operating point. The expected maximum power consumption of the vehicle is found to be 175 watts.

Table 13. Power Consumption Analysis.

Group	Type	Part	Description	V	Max I	PS E%	Max Power	Duty Cycle	Expected P	% Total
	Central Computer	VL Cheetah		5	5	86	29.07	100.00	29.07	16.65
	ADC	VCMDAS1		5	0.51	86	2.97	100.00	2.97	1.70
	Frame Grabber	Soni M311		5	0.3	86	1.74	100.00	1.74	1.00
	Cabin Temp	National LM20		3.3	0.01	85	0.04	100.00	0.04	0.02
	PC104 Temp	National LM20		3.3	0.01	85	0.04	100.00	0.04	0.02
	Front Pressure	MSI Model 1451-030A-T		3.3	0.01	85	0.04	100.00	0.04	0.02
	Rear Pressure	MSI Model 1451-030A-T		3.3	0.01	85	0.04	100.00	0.04	0.02
	Ballast 1 Pressure	MSI Model 1451-030A-T		3.3	0.01	85	0.04	100.00	0.04	0.02
	Ballast 2 Pressure	MSI Model 1451-030A-T		3.3	0.01	85	0.04	100.00	0.04	0.02
	Ballast 1 Charge	MSI Model 1451-030A-T		3.3	0.01	85	0.04	100.00	0.04	0.02
	Ballast 2 Charge	MSI Model 1451-030A-T		3.3	0.01	85	0.04	100.00	0.04	0.02
Main	Main Thruster 1 Current	From LMD18200t thru Resistor	377uA/A	2.5	0.0008	100	0.001885	40.00	0.0007540	
Thrusters	Main Thruster 2 Current	From LMD18200t thru Resistor	377uA/A	2.5	0.0008	100	0.001885	40.00	0.0007540	
	Main Thruster 1 Motor	Anaheim BDPG-38-86-12V-3000-		12	3	100	36.00	40.00	14.40	
	Main Thruster 2 Motor	Anaheim BDPG-38-86-12V-3000-		12	3	100	36.00	40.00	14.40	
	Main Thruster 1 Driver	National LMD18200T		n/a	n/a	100	2.70	40.00	1.08	
	Main Thruster 2 Driver	National LMD18200T		n/a	n/a	100	2.70	40.00	1.08	17.73
Stab	Stab Thruster 1 Sol	STC		12	1	100	12.00	20.00	2.40	
	Stab Thruster 2 Sol	STC		12	1	100	12.00	20.00	2.40	
	Stab Thruster 3 Sol	STC		12	1	100	12.00	20.00	2.40	
	Stab Thruster 4 Sol	STC		12	1	100	12.00	20.00	2.40	
	Stab Thruster 1 Driver	In-House	Based on STD17NF03	n/a	n/a	100	0.02	20.00	0.00324	
	Stab Thruster 2 Driver	In-House	Based on STD17NF03	n/a	n/a	100	0.02	20.00	0.00324	
	Stab Thruster 3 Driver	In-House	Based on STD17NF03	n/a	n/a	100	0.02	20.00	0.00324	
	Stab Thruster 4 Driver	In-House	Based on STD17NF03	n/a	n/a	100	0.02	20.00	0.00324	5.50
Ballast	Ballast 1 Inlet Sol	STC		12	0.25	100	3.00	25.00	0.75	
	Ballast 2 Inlet Sol	STC		12	0.25	100	3.00	25.00	0.75	
	Ballast 1Inlet Driver	In-House	Based on STD17NF03	n/a	n/a	100	0.02	25.00	0.00405	
	Ballast 2 Inlet Driver	In-House	Based on STD17NF03	n/a	n/a	100	0.02	25.00	0.00405	
	Ballast 1 Outlet Sol	STC		12	0.25	100	3.00	25.00	0.75	
	Ballast 2 Outlet Sol	STC		12	0.25	100	3.00	25.00	0.75	
	Ballast 1 Outlet Driver	In-House	Based on STD17NF03	n/a	n/a	100	0.02	25.00	0.00405	
	Ballast 2 Outlet Driver	In-House	Based on STD17NF03	n/a	n/a	100	0.02	25.00	0.00405	1.73
Pump Dir	Pump Inlet 1 Solenoid	STC		12	0.25	100	3.00	50.00	1.50	
	Pump Inlet 2 Solenoid	STC		12	0.25	100	3.00	50.00	1.50	
	Pump Inlet 1 Driver	In-House	Based on STD17NF03	n/a	n/a	100	0.02	25.00	0.00405	
	Pump Inlet 2 Driver	In-House	Based on STD17NF03	n/a	n/a	100	0.02	25.00	0.00405	1.72
	Pump			12	13	100	156.00	50.00	78.00	44.66
	Pump Driver	In-House	Based on STD17NF03	n/a	n/a	100	0.02	50.00	0.01	0.03
	Gyro/Accelerometer	ADIS16354		5	0.06	86	0.35	100.00	0.35	0.20
	Magnetic Compass Circ	In-House		5	0.05	86	0.29	100.00	0.29	0.17
	Sonar Circ	In-House		24	8	85	225.88	5.00	11.29	6.47
	Drive Hub Processor	MSP530F233	at 16Mhz	3.3	0.006	100	0.02	10.00	0.00198	0.0011
	Fore Hub Processor		at 16Mhz	3.3	0.006	100	0.02	10.00	0.00198	0.0011
	Aft Hub Processor		at 16Mhz	3.3	0.006	100	0.02	10.00	0.00198	0.0011
	Watchdog Hub Processor		at 16Mhz	3.3	0.006	100	0.02	10.00	0.00198	0.0011
	Pressure Sensor Cond		x6			85	3.00	100.00	3.00	1.72
	Temperature Sensor Cond		x2			85	1.00	100.00	1.00	0.57
						Total Max Power:	560	**Total Expected Max Power:**	175	

Appendix C. Vehicle Control Code

C.1 USB to I^2C using ez430-RF2500

This code is for the ez430-RF2500 to allow communication between the Sub-Hub system and the monitor station over an I^2C bus.

```
/* This code is meant for use with the MSP430 EZ430-RF2500 Development Tool.
/  It takes an ASCII character sent through the MSP430 UART, usually with Hyper
/  Terminal, and sends it over an I2C bus to an MSP430 slave.
*/

#include  "msp430x22x4.h"

unsigned char TXData;
unsigned char TXByteCtr;
unsigned short int timer=0; //count interrupts from TimerB0
unsigned short int min; // elapsed minutes
unsigned short int sec; // elapsed seconds
unsigned short int msec; // elapsed 1/100 seconds
unsigned short int tdir; // Direction of timer
void UART_Setup (void);
void I2C_TX_Setup (void);
void runtimerb(void);
void stoptimerb(int reset);

//  Echo back RXed character, confirm TX buffer is ready first
#pragma vector=USCIAB0RX_VECTOR
__interrupt void USCI0RX_ISR(void)
{
  TXData = UCA0RXBUF;                   // Store Received character
  TXByteCtr = 1;                        // Load TX byte counter
}

#pragma vector = USCIAB0TX_VECTOR
__interrupt void USCIAB0TX_ISR(void)
{
 // UART stuff
```

```
 if((IFG2 & UCA0TXIFG)){                 // USCI_A0 TX buffer ready?
 P1OUT ^= 0x01;
 UCA0TXBUF = TXData;                   // Transmit first character (SPI)
 }


//I2C Stuff
 if (TXByteCtr)  {                     // Check TX byte counter

   UCB0TXBUF = TXData;                   // Load TX buffer
   TXByteCtr--;                        // Decrement TX byte counter
   TXData = 0x00;
 }
 else {

   UCB0CTL1 |= UCTXSTP;                  // I2C stop condition
   IFG2 &= ~UCB0TXIFG;                   // Clear USCI_B0 TX int flag
   //__bic_SR_register_on_exit(CPUOFF+GIE);  // Exit LPM0
 }
}

#pragma vector=TIMERB0_VECTOR
__interrupt void Timer_B0(void)
{
 I2C_TX_Setup();
 __no_operation();
}

void main(void)
{
 WDTCTL = WDTPW + WDTHOLD;               // Stop WDT
 BCSCTL1 = CALBC1_1MHZ;                  // Set DCO
 DCOCTL = CALDCO_1MHZ;
 P3SEL &= ~BIT0; //Setup CC2500 CSn
 P3DIR |= BIT0; //CC2500 CSn OUTPUT
 P3OUT |= BIT0; //CC2500 CSn held high, CS2500 out of the question
```

```
  P1OUT = 0x00;                         // P1 setup for LED
  P1DIR |= 0x01;                        //
  TXData = 0x00;                        // Holds TX data
  _BIS_SR(GIE); // Global Interrupt enable
  while (1) {
    UART_Setup();
    I2C_TX_Setup();
    _EINT();

    while (UCB0CTL1 & UCTXSTP);             // Ensure stop condition got sent
    if (TXData){
      TXByteCtr = 1;
      while (UCB0CTL1 & UCTXSTP);           // Ensure stop condition got sent
      UCB0CTL1 |= UCTR + UCTXSTT;             // I2C TX, start condition
      runtimerb();
      while((UCB0STAT & UCBBUSY));
      stoptimerb(1);
      //__bis_SR_register(CPUOFF+GIE);          // CPU off, enable interrupts
    }
  }
}
void UART_Setup (void){
  P3SEL = 0x30;                         // P3.4,5 = USCI_A0 TXD/RXD
  UCA0CTL1 |= UCSSEL_2;                     // SMCLK
  UCA0BR0 =104;                          // 1MHz 9600
  UCA0BR1 = 0;                          // 1MHz 9600
  UCA0MCTL = UCBRS0;                        // Modulation UCBRSx = 1
  UCA0CTL1 &= ~UCSWRST;                     // **Initialize USCI state machine**
  IE2 |= UCA0RXIE;                        // Enable USCI_A0 RX interrupt
}
void I2C_TX_Setup (void){

  P3SEL |= 0x06;                        // Assign I2C pins to USCI_B0
  UCB0CTL1 |= UCSWRST;                      // Enable SW reset
  UCB0CTL0 = UCMST + UCMODE_3 + UCSYNC;     // I2C Master, synchronous mode
  UCB0CTL1 = UCSSEL_2 + UCSWRST;            // Use SMCLK, keep SW reset
  UCB0BR0 = 104;                        // fSCL = SMCLK/12 = ~100kHz
```

```
 UCB0BR1 = 0;
 UCB0I2CSA = 0x48;                    // Slave Address is 048h
 UCB0CTL1 &= ~UCSWRST;                // Clear SW reset, resume operation
 TXByteCtr = 1;                       // Load TX byte counter (I2C)
 IE2 |= UCB0TXIE;                     // Enable TX interrupt
 while (UCB0CTL1 & UCTXSTP);          // Ensure stop condition got sent
}

void runtimerb(void)
{
 TBCTL = TBSSEL_1 + CNTL_0 + MC_1 + ID_0; // ACLK, 16 Bit, up mode, div=1
 TBCCR0 = 0x8000; // 32768 ACLK tics = 1 second
 TBCCTL0 = CCIE; // TBCCR0 interrupt enabled
 tdir = 1;
}
/* This function stops Timer B */
void stoptimerb(int reset)
{
 TBCTL = MC_0; // stop timer
 TBCCTL0 &= ~CCIE; // TBCCR0 interrupt disabled
 if(reset)
 {
   timer=0;
   min=0;
   sec=0;
   msec=0;
 }
}
```

C.2 EZ430-RF2500 I^2C Note

The ez430-RF2500 target board's USCIB0 I^2C bus shares pins with the USCIB0 SPI bus. The target board includes a CC2500 wireless chip which communicates over the SPI bus. The CC2500's serial output pin (SO) is shared with the I^2C's SCL pin. If the CC2500's active low chip select (CSn) is not held high, the CC2500 will hold its SO pin low, which will pull SCL low. In this condition an I^2C module will assume that the bus is

busy and will not transmit. The code below must be inserted in the ez430-RF2500 target boards main function to change the CC2500 SO output to a high impedance input.

```
P3SEL &= ~BIT0; //Setup CC2500 CSn
P3DIR |= BIT0; //CC2500 CSn OUTPUT
P3OUT |= BIT0; //CC2500 CSn held high, CS2500 SO switched to high impedance
```

Appendix D. PC-104 Stack Operation

D.1 Versalogic Cheetah, Connections and Linux Installation

D.1.1 Connections

A PS/2 keyboard and mouse, SVGA monitor, ATX power supply and CD-ROM drive (with bootable CD) are required for the installation process. The VersaLogic CBR-8001, standard I/O breakout cable (to be connected to JS5) provides connections for the keyboard and mouse as well as a reset switch that can be used to restart the machine (along with a number of other connectors). The VersaLogic CBR-1008 ATX adapter is also needed when using a standard ATX connector (such as the connector found on the pico-PSU) for power. The micro ATX connector on the adapter is then connected to JS9. The SVGA monitor is to be connected to JN3. The CBR-4406 IDE cable should then be connected to JS4. These connections are shown below in a modified version of Figure 2 (pg. 8) from the VersaLogic Cheetah Manual. Before booting, ensure that the RAM and Compact Flash hard drive are installed.

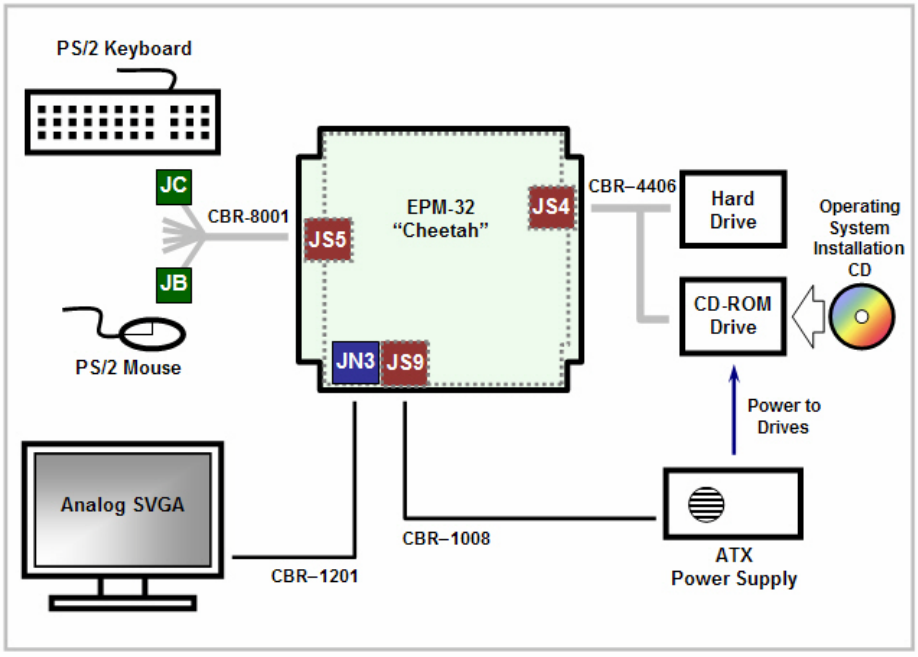


Figure 77. Cheetah Connections
© 2004-2009 Versalogic Corporation.[19]

As noted in the Cheetah Manual, the board should not be placed on a conductive surface (including electrostatic dissipative packaging) in order to avoid shorting the battery. It is suggested that standoffs be installed to create a stable platform.

D.1.2 CMOS Configuration

The PC104 manual outlines the installation procedure and CMOS settings to install a Windows platform on an external IDE hard drive (from a bootable CD-ROM). The current configuration, however, uses a compact flash drive as the main hard drive, and a Linux based OS. This requires slightly different CMOS settings.

The CMOS settings are accessed by pressing the DELETE key during the early boot process. Once the CMOS settings menu comes up, enter the "Basic CMOS Configuration" menu. Under "Drive Assignment Order" set Drive C to "IDE 2/Secondary Master". Under "Boot Order" CD-ROM should be set to boot first and Drive C should be set to boot second. Note that after the installation process is complete, Drive C should be set to boot first and CD-ROM can be removed from the boot order. If there are problems during installation, Drive C can be set to boot first and CD-ROM to boot second.

Both drives then need to be assigned under "ATA Drive Assignment". IDE 0 (the primary master) should be set to "IDE CDROM". It is important that the jumper on the CD-ROM drive be set to master. For the 4Gb Compact Flash drive, IDE 2 (the secondary master) should be set to "AUTOCONFIG, PHYSICAL". For hard drives larger than 4Gb, "AUTOCONFIG, LBA" (Large Block Addressing) may need to be used.

When finished exit CMOS setup and save changes.

D.1.3 Installation Disk

VersaLogic provides a quick start software package which includes a Debian Linux 3.1 installation image (.ISO) available at http://www.versalogic.com/private/cheetahsupport.asp. This version is tailored specifically to the Cheetah EPM-32 PC/104 and offers three stable images (Minimum, Medium and Full Install). The Full Install provides the GNOME graphical user interface. Before it can be used for installation, the ISO image must be burned to a bootable CD using software such as CDBurnerXP.

The release currently loaded on the Cheetah is DEV-CD-L2-R2.02. There is now a newer version of the package, DEV-CD-L2-3.00, which includes Debian Linux 5.0.

D.1.4 Installation

Once all of the above steps are completed, the CD can be inserted into the drive and the PC104 can be booted. The disk will boot to a desktop with the three installation options appearing as icons. The Full Install is suggested and will bring up an installation wizard. Once through all of the steps, the system will reboot to a log in screen. The default username is "root" and the default password is "password". It is suggested that all user's log in as root as this ensures access to I/O permissions for all programs (allowing use of the ADC and other additional boards and peripherals).

D.2 Versalogic Cheetah, Program Execution

D.2.1 GCC Compiler

The GCC compiler should install with the VersaLogic Debian Linux image. To check that the GCC compiler is installed, open a terminal and type "gcc". If the response "Comand gcc not found" is received the GCC compiler must be installed

The compiler will look in the current working directory by default (to find the current working directory open a terminal and type "pwd"). To avoid segmentation faults, or to compile a program outside of the working directory, it is suggested that the full file path be used (i.e. /root/new_folder/program.c). For best results, enable warning ("-Wall"). When using port I/O functions optimization level of 1 or higher must be used (i.e. "-o", "-o1", "-o2", etc.). This combination will result in a warning message when using port I/O functions, however this warning can be disregarded. GNU provides an in depth explanation of all of these commands and more at: http://gcc.gnu.org/onlinedocs/gcc/Option-Summary.html#Option-Summary.

The default executable will be placed in the working directory under the name "a.out". Following "-o" with a filename or file path and file name will change the executable name and file path.

Following the above, to compile "program.c", found in the current working directory and create an executable named program.out the following line should be entered in the terminal: "gcc –Wall –o program program.c" or "gcc –Wall –o

/root/new_folder/program program.c" to store the .out file in the directory "root/new_folder".

D.2.2 Executing a Program

Linux does not look in the current working directory when executing a file. Therefore, to execute a program, the full file path must be entered. For the example above, if the executable is stored in the directory "root/new_folder", in order to execute the file, the command "/root/new_folder/program.out" must be entered.

D.3 Versalogic VCM-DAS-1 Functionality

The following sections outline operation of the VCM-DAS-1. The DAS-1 should be attached to the PC104 stack as outlined in Section 0:

Central Processing.

D.3.1 Port I/O Access in Linux

The functions inb(address) and inw(address) allow a program to read from the registers (a byte and a word, respectively). Similarly, outb(value, address) and outw(value, address) allow a program to write a value to the registers.

These functions require Port I/O permissions. The function ioperm(address, number, on/off), where "address" is the starting address, number is the number of registers to be turned on or off and on/off is a "1" (to turn register access for those ports on) or "0" (to turn register access for those ports off). The use of this function requires root permissions (the user either needs to be logged in as root or the program needs to set the user ID as root).

All of the Port I/O functions are defined in "asm/io.h".

D.3.2 ADC Header File, vcm-das-1-WPI.h

The vcm-das-1-WPI.h header file includes function, constant and register definitions and addresses to be used with the VCM-DAS-1:

```
#include <stdio.h>

/*************************************
 ***    Base Address Definition    ***
 *************************************/
//Enter the base address (as specified by
//the jumper configuration of V4
//(See Versalogic VCM-DAS-1 manual, pg.8
//"Configuration/Module Addressing"

//////////////////////////////////////////////////
#define base_addr          0x300
//////////////////////////////////////////////////

/*************************************
 ***  Write Register Definitions   ***
 *************************************/
//Ref. Man. pg. 23
#define SPIWDAT         (base_addr + 9)
#define SPISEL          (base_addr + 8)
#define PARWHI          (base_addr + 7)
#define PARWLO          (base_addr + 6)
#define ADCPTR          (base_addr + 3)
#define ADCCVT          (base_addr + 2)
#define ADCSEL          (base_addr + 1)
#define CONTROL         (base_addr + 0)

/*************************************
 ***   Read Register Definitions   ***
 *************************************/
//Ref. Man. pg. 23
#define SPIRDAT         (base_addr + 9)
#define PARRHI          (base_addr + 7)
```

```
#define PARRLO          (base_addr + 6)
#define ADCHI           (base_addr + 5)
#define ADCLO           (base_addr + 4)
#define ADCSTAT         (base_addr + 0)

/*************************************
 ***   SPI Register Definitions   ***
 *************************************/
//Ref. Man. pg. 25
#define ADOFSET         0x00 (DPOT 1)
#define DA0GAIN         0x01 (DPOT 2)
#define ADGAIN          0x02 (DPOT 3)
//#define DA1GAIN       0x03 (DPOT 4)
//#define DA0DATA
//#define DA1DATA
//#define WDATA
//#define ADDRESS
//#define OPCODE

/*************************************
 ***      BIT-MASK Definitions     ***
 *************************************/
// register bits
#define D7              0x80  //10000000
#define D6              0x40  //01000000
#define D5              0x20  //00100000
#define D4              0x10  //00010000
#define D3              0x08  //00001000
#define D2              0x04  //00000100
#define D1              0x02  //00000010
#define D0              0x01  //00000001
#define Dn              0x00  //00000000

// For use with ADCCVT
#define CONVERT         0x01 //calls a conversion

// For use with ADCSTAT
```

```
#define ADCBUSY         D7   // If ADC is busy
#define ADCDONE         D6   // If a conversion is waiting to be read

// For use with CONTROL, Ref. Man. pg. 34
#define PARHIOUT        D7 //DI0:8-15  to Output (DIRHI)
#define PARHIIN         Dn //DI0:8-15  to Input  (DIRLO)
#define PARLOOUT        D6 //DIO:0-7   to Output (DIRHI)
#define PARLOIN         Dn //DI0:8-15  to Input  (DIRLO)
#define DMAEN           D5 //DMA Enable
#define AICEN           D4 //Auto-Increment Enable
#define ATRIG           D3 //Auto-Trigger Enable
#define PTRIG           D2 //Periodic-Trigger Enable
#define SHARE           D1 //Shared Interrupt Enable
#define INTEN           D0 //Interrupt Enable

// For Use with DACSR (DAC LOAD SHIFT REGISTER), Ref. Man. pg. 44
#define DetDACCh        0x8000//a and b bits determine dac channel to be loaded
#define LoadCh1         0x4000//Select Dac B (Ch 1) for load
#define LoadCh2         0x2000//Select Dac A (Ch 2) for load

// For use with SPISEL (Use Only One of The Following)
#define DACLOAD         D3 //Analog Output Data Load=DACSR,reset before reload
#define EESEL           D2 //EEPROM Chip Select
#define DPOTSEL         D1 //Digital Pot Chip Select
#define DACSEL          D0 //Analog Output Chip Select

//to be used with ADCSEL, scan ranges
#define scan0_15        0x00 //all channels
#define scan0_3         0x10 //chan 0-3 are scanned
#define scan0_7         0x20 //chan 0-7 are scanned
//channel select (OR with scan ranges)
#define ch0             0x00
#define ch1             0x01
#define ch2             0x02
#define ch3             0x03
#define ch4             0x04
#define ch5             0x05
```

```
#define ch6             0x06
#define ch7             0x07
#define ch8             0x08
#define ch9             0x09
#define ch10            0x0A
#define ch11            0x0B
#define ch12            0x0C
#define ch13            0x0D
#define ch14            0x0E
#define ch15            0x0F

//to be used with ADCPTR
#define TR128           0x00
#define TR256           0x01
#define TR512           0x02
#define TR1024          0x03
#define TR2048          0x04
#define TR4096          0x05
#define TR8192          0x06

/************************************
 ***      Constant Definitions     ***
 ************************************/
 //Conversion Factors for ADC Reads
#define step5           0.0001525878907
#define step10          0.0003051757813
//Conversion Factors for DAC Writes

/************************************
 ***      Function Definitions     ***
 ************************************/
//_____________________________________
//  Writes to Specified Dpot Register
//spiWrite: device: spi device select code
//spec:    special select (i.e. DPOT #3)
//instruction: instruction to be sent to selected SPI register
void spiWrite(short device, int inst);
```

```
//_____________________________________________
//_____________________________________________
//  Sign extends 2's complement short to int
//value: short value to be extended (i.e. ADC Reading)
int signExt(short value); //extends short to int, preserves 2's complement
//_____________________________________________
//_____________________________________________
void channelselect(short scanrange, short channel); //setup channel/scan range
//_____________________________________________
//_____________________________________________
// Takes scan_range and channel to be read,
// Returns 16 bit reading from ADC
short adcread (short scan_range, short channel);
//_____________________________________________
//_____________________________________________
//  Takes 0xXX from adcread and converts to voltage
float analog (short reading);
```

D.3.3 ADC Function File, vcm-das-1-WPI.c

The file vcm-das-1-WPI.c includes the functions necessary to gain basic control of the DAS-1 (these are defined in vcm-das-1-WPI.h):

```
#include <stdio.h>
#include <asm/io.h>
#include <unistd.h>
#include </root/PC104CODE/ADCCODE/vcm-das-1-WPI.h>

#ifdef step5
  #define conversion step5
#else
  #define conversion step10
#endif


void spiWrite(short device, int inst) {
   int i;
```

```
  switch (device) {
        case EESEL
          break;
        case DPOTSEL
          inst <<= 2;
          i = 10;
          break;
        case DACSEL
          i = 16;
          break;
        default
          break;
  }
  ioperm(SPISEL,1,1);
  outb(device,SPISEL);
  ioperm(SPISEL,1,0);

  ioperm(SPIWDAT,1,1);
  while( i-- > 0 ) {
    outb(((instruction>>i)&D0), SPIWDAT);
    printf(%x,%x, (instruction>>i),((instruction>>i)&D0));
  }

  ioperm(SPISEL,1,0);
}

int signExt(short value) {
  int extended;
  if ( ( value & 0x8000 ) == 0x8000 )
    extended = value | 0xFFFF0000;
  else
    extended = value & 0x00001111;
  return extended;
}

void channelselect(short scanrange, short channel) {
  ioperm(ADCSEL,1,1); //Gives port IO permission
```

```
  outb( (scanrange|channel) ,ADCSEL);
  ioperm(ADCSEL,1,0); //Removes port IO permission, allows settling
}

short adcread (short scan_range, short channel) {
  short reading;
  ioperm(ADCSTAT,1,1); //Gives Port IO permissions
  ioperm(ADCLO,2,1);

  //Check busy/done
  while ( (inb(ADCSTAT)&D7) == D7 ); //If busy, wait
  if ( (inb(ADCSTAT)&D6) != D6 ) {  //If no value yet
     ioperm(ADCCVT,1,1);
     outb(CONVERT,ADCCVT); //Tell ADCCVT to perform conversion
     ioperm(ADCCVT,1,0);
  }
  while ( (inb(ADCSTAT)&D6) != D6);  //Wait for DONE

  reading = inw(ADCLO); //Read ADCLO and ADCHI, reset DONE bit in ADCSTAT
  ioperm(ADCSTAT,1,0); //Remove Port IO permissions
  ioperm(ADCLO,2,0);
  return reading;
}

float analog (short reading) {
  return (reading * step10);
}
```

D.3.4 ADC Test Code

Provided below is sample test code to gain accessof the VCM-DAS-1 analog inputs. The program prompts the user for a file name (less than 9 characters long) then takes 10000 consecutive readings from ADC Channel 3 and writes them to the file:

```
#include </root/PC104CODE/ADCCODE/vcm-das-1-WPI.c>
#include <stdio.h>
```

```
#include <asm/io.h>
#include <unistd.h>

//This program gains control of the VCM-DAS_1 board
//Functions will be added later to allow full I/O
//It is assumed that only one DAS-1 board will be installed
//Functions will be added later to allow easy use of
//multiple boards

int main (void) {
  int i;
  short adcreading;
  char filename[10];
  char directory[] = "/root/";
  FILE *valueTest;


  printf("Enter File Name (less then 9 characters):\n");
  scanf("%9s", filename); //input 9 or less digit filename
  strcat(&s,".dat");  //append file extension
  strcat(&directory,&filename); //append extension to directory

  if ((valueTest = fopen(s,"a")) == NULL) { //open file to "append"
    printf("Can't open input file\n"); //tell user if file can't be opened
  }
  channelselect(ch0_3,ch0);
  for (i=0;i<10000;i++) {  //take 10000 readings and dump to file
   adcreading = adcread(); //full ADC read
   fprintf(valueTest,"%f ",analog(adcreading));
  }
  fclose(valueTest);  //close file before exit
  return 0;
}
```

Appendix E. Sonar Navigation System

E.1 Sonar Navigation System Initial Design Parts List

Parts List Autonomous Underwater Vehicle **Navigation System (Ultrasound Ranging Device)** **(09/22/2008)**

Supplier	Supplier Part No	Qty	Unit Price	Total Price	Description
DigiKey	497-3017-5-ND	1	2.73	2.73	RS232 voltage driver
	LMC6032IN-ND	1	1.46	1.46	Dual Op-Amp pack
	LM1458NNS-ND	1	1.2	1.2	Dual Op-Amp pack
	LM311NNS-ND	1	1.2	1.2	Op-Amp Comparator
	490-1888-2-ND	10	0.1	1	100 nf Capacitors
	445-2869-ND	5	0.522	2.61	22uf capacitors
	445-2851-ND	5	0.286	1.43	1uf capacitors
	36KH-ND	5	0.052	0.26	36k resistors
	10KQBK-ND	10	0.054	0.54	10k Resistors
	2.0KQBK-ND	5	0.054	0.27	2k Resistors
	120KH-ND	5	0.052	0.26	120k Resistors
	1.0KQBK-ND	5	0.054	0.27	1k Resistors
HobbyEngineering.com	H01324-01W	1	9.99	9.99	Ultrasonic Transducers (Transmitter/Receiver Set)
				Total 23.22	

E.2 Sonar Navigational System – Final Design Parts List

Autonomous Underwater Vehicle **Navigation System (Ultrasound Ranging Device)**

Parts List

Radu David (02/23/2009)

Supplier	Supplier Part No	Qty	Unit Price	Total Price	Description
DigiKey	497-3155-1-ND	5	1.08	5.4	STD17NF03 Power Mosfet
	296-17414-1-ND	5	1.25	6.25	Opa358 Single Rail OP-amp
	Z647-ND	2	13.26	26.52	Reed Relays DPDT relay
	ZVN3306FCT-ND	5	0.93	4.65	ZVN Mosfet
	2.0KQBK-ND	5	0.054	0.27	2k Resistors
	36KH-ND	5	0.052	0.26	36k resistors
	XC1243CT-ND	3	0.73	2.19	8MHz oscillator
Amidon Corp	FT-140-61	2	3.75	7.5	Material 61 Toroidal Ferroid 1.4"

E.3 Sonar Navigational System – MSP430 Program

```
//*******************************************************************
*********

#include  <msp430x44x.h>
//#include  <io430x44x.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
//#include <math.h>
#include <in430.h>

#define          B1                P3IN & BIT4       //B1 - P4.4
#define          B2                P3IN & BIT5       //B2 - P4.5
#define          B3                P3IN & BIT6       //B3 - P4.6
#define          B4                P3IN & BIT7       //B4 - P4.7
#define           STATUS_LED          BIT3               //STATUS_LED -
P1.3
#define          time              200
#define          time_s            200
#define          BUZ1_ON           P1OUT |= BIT0     //P4.2
#define          BUZ1_OFF          P1OUT &= ~BIT0    //P4.2
#define          BUZ2_ON           P1OUT |= BIT2     //P4.3
#define          BUZ2_OFF          P1OUT &= ~BIT2    //P4.3
#define          DALLAS            P2IN & BIT7       //P2.7 - DALLAS
#define          CR                0x0d
#define          LF                0x0a
#define          LED_OFF           P1OUT |= BIT3
#define          LED_ON            P1OUT &= ~BIT3


//LCD digit segments


unsigned char i,time_segment,TXData;
unsigned int j;
  unsigned int Voltage;
```

```
const unsigned char UART_Message [] = "This is a test, UART works";

unsigned char message [];
unsigned int n = 123;
  int in_press;

void Delay (unsigned int a);
void DelayX(unsigned int b);
void Clear_LCD(void);
void LCD_all(void);
void BUZZER (void);
void UART_transmit (unsigned char Transmit_Data);
void get_ADC (void);
void  send_data  (int  n);   //function  that  sends  a  number  as  ASCII
characters

void runtimerb(void);
void stoptimerb(void);
void init_sys(void);

unsigned int timer = 0;
//unsigned int i = 0;
unsigned int data;

/******************Interrupt Subroutine for timer**************/

#pragma vector=TIMERB0_VECTOR
__interrupt void Timer_B(void)
{

 //  timer++; //increase the count everytime the interrupt occurs
  ADC12CTL0 |= ADC12SC + ENC;
  data= ADC12MEM0 & 0x0FFF;
  data=10;
  send_data(data);

   P1OUT ^= BIT1; //toggle a pin high-low to check the frequency rate
```

```
}



void main(void)

{
//+++++++++++++++++++++++++++++
  WDTCTL = WDTPW | WDTHOLD;                   // stop watchdog timer

  FLL_CTL0 &= ~XTS_FLL;                      // XT1 as low-frequency
  _BIC_SR(OSCOFF);                          // turn on XT1 oscillator

  do                          // wait in loop until crystal is stable
    IFG1 &= ~OFIFG;
  while (IFG1 & OFIFG);

  FLL_CTL1 &= ~FLL_DIV0;                     // ACLK = XT1
  FLL_CTL1 &= ~FLL_DIV1;



  IFG1 &= ~OFIFG;                     // clear osc. fault int. flag
  FLL_CTL1 &= ~SELM0;                 // set DCO as MCLK
  FLL_CTL1 &= ~SELM1;
  _BIS_SR(GIE);                       //Global Interrupts enable
  init_sys();                        //this initializes the USART module

  UART_transmit (CR);                //send message by RS232
  UART_transmit (LF);               //test at the beginning of program
  unsigned int j = 0;
  for (j=0; j<=10; j++)
  {
    j++;
  for (i=0; i!=26; i++)
    UART_transmit (UART_Message[i]);      //sends a message before the
controller starts collecting data
  UART_transmit (CR);
  UART_transmit (LF);
```

```
  }
   P1DIR |=  (BIT1);                //P1.1 to output
  Clear_LCD();                      //clear the LCD
//---------------enable timer interrupt--------------

//  runtimerb();  //the interrupt enable

//---------------MAIN LOOP----------------------------------
 get_ADC(); //initialize the ADC for single step single conversion

  while (1)
  {
      /*These  are  the  same  steps  that  are  done  in  the  interrupt
subroutine
      but for now I run them here to check for the maximum speed*/
    ADC12CTL0 |= ADC12SC + ENC;  //this enables an ADC conversion (no
need to call get_ADC()- time consuming)

  data= ADC12MEM0 & 0x0FFF;  //data is aquired
  data/=10; //just get the 3 highest decimals (last one is just noise)
  send_data(data);  //function that converts the data and sends it as 3
ASCII characters over RS232
  P1OUT ^= BIT1;
  }
}


/**************Run / stop timer functions*************/

void runtimerb(void)
{
  TBCTL = TBSSEL_2 + CNTL_0 + MC_1 + ID_0; // SMCLK (1.048576 MHz),
                                           //   16 Bit, up mode, div=1
  TBCCR0 = 0x0A; // 1048576 ticks each second, it can be set to any
frequency (now set around 10 KHz
  TBCCTL0 = CCIE; // TBCCR0 interrupt enabled

}
```

```
void stoptimerb(void)
{
  TBCTL = MC_0; // stop timer
  TBCCTL0 &= ~CCIE; // TBCCR0 interrupt disabled
}

void init_sys(void)
{
  //+++++++++++++++++++++++++++++

//UART ini (contains various USART implementations; the working one is
not commented)

  U1TCTL = 0x30;                          //UCLOCK = SMCLK (1MHz clock)
  U1BR0 = 0x09;                             //BAUD RATE = 115200
  U1BR1 = 0x00;                      //values  from  the  manual  (page  273
(14-16))  to set the USART at highest rate
  U1MCTL = 0x08;                               //with modulation
//  U1TCTL = SSEL0;
//UCLOCK = ACLK
//  U1BR0 = 0x03;                              //BAUD RATE = 9600
//  U1BR1 = 0x00;
//  U1MCTL = 0x4a;                                 //with modulation
  U1CTL = CHAR;                               //Start 8bit 1stop, N
  ME2 = UTXE1 | URXE1;                         //enable RX and TX
  P4SEL |= BIT0 | BIT1;                       //select UART pins
  P4DIR |= BIT0;                             //port direction for TXD0
  P4DIR &= ~BIT1;                           //port direction for RXD0
//  IE1 |= URXIE0;                         // Enable USART0 RX interrupt
//  _EINT();                              // Enable interrupts
//  U1CTL &= ~SWRST;                     //reset UART logic



//IO port ini

  P1SEL=BIT5;                                      //p1.5 is 32768Hz
```

```
  P1DIR=BIT5 | BIT3 | BIT0 | BIT2;               //BUZ,LED are outputs

  LCD_all();                                   //light all LCD segments

  for (i=0; i!=15; i++)
    {
      LED_ON;
      DelayX(60);
      LED_OFF;
      DelayX(60);
    }
}



void Delay (unsigned int a)                     //9+a*12 cycles
{
  unsigned int l;
  for (l=0 ; l != a; l++);
}


void DelayX(unsigned int b)
{
unsigned int m;
for(m=0;m!=b;m++) Delay(255);
}


void  UART_transmit  (unsigned  char  Transmit_Data)      //UART1  Transmit
Subroutine
{
  while ((IFG2 & UTXIFG1) == 0);              //Wait for ready U1TXBUF
  U1TXBUF = Transmit_Data;                    //send data
}

void send_data (int n)
{
```

```
  int j;
   if (n < 0)
   {
     n = 0-n;
     sign = '-';
    }
  UART_transmit (CR);                          //send message by RS232
(UART_transmit is an example function (fills the buffer with the
character and waits till it clears))
  UART_transmit (LF);                          //need     the     LF
character, and also return character (that increases the transmit time
considerably, but there might be ways around that)
 for (j=1; j<=3; j++) {                        //this     is     the
conversion from number to character (doesn't take very much time)
   message [4-j] = 0x30+(n%10);
  n/=10;
  }
  for (j=1; j<=3; j++) {
  UART_transmit(message[i]);
      }
}

//  ADC configuration
void get_ADC(void)
{
  //initialize control register ADC12CTL0 = 0110 0110  0111 0000
  //SHT1x and SHT0x = 66h (both 128 clks), MCS = 0 = no burst mode
  //REF2_5 = 1 (2.5 V), REFON = 1 = use internal reference voltage
  //ADC12ON = turn on ADC
  ADC12CTL0 =  SHT0_8 + SHT1_8 + REF2_5V + REFON + ADC12ON + MSC;
  // Initialize control register ADC12CTL1 = 0000 0010 0000 0000
  // CSTART ADDx = 0000 = start conversion with ADC12MEM0,
  // SHSx = 00 = use SW conversion initiation trigger, ADC12SC bits
  // SHP = 1 = SAMPCON signal sourced from sampling timer,
  // ISSH = 0 = sample input signal not inverted,
  // ADC12DIVx = 000= divide ADC12CLK by 1,
  // ADC12SSEL=00= ADC clock ADC12OSC (~5 MHz),
  // CONSEQx = 00 single channel, single conversion,
```

```
  // ADC12BUSY = 0 = no ADC operation active
  ADC12CTL1 = SHP + CONSEQ_0;

  // Set conversion memory control register ADC12MCTL0 = 0001 1010
  // EOS = 0, SREF =001 > Voltage refs = GND to 2.5V (Vref+)
  // INCHx = 0000 = analog input from A0
  ADC12MCTL0 = SREF_1 + INCH_1;

  P6SEL |= BIT0|BIT1|BIT2 ;  //SET port 6 pin 0 in function for ADC


  //Enable and start (single) conversion
  ADC12CTL0 |= ADC12SC + ENC;



  //in_press = ADC12MEM0 & 0x0FFF; // keep only low 12 bits
 // V_ADC = (long)in_press; // convert to long (voltage)
 // in_press = ADC12MEM3 & 0x0FFF;
 // I_ADC = (long)in_press;// convert to long (current);


}
```

E.4 Sonar Navigational System – Matlab Retrieve Data Program

```
% Create a serial port object.
obj1 = instrfind('Type', 'serial', 'Port', 'COM1', 'Tag', '');

% Create the serial port object if it does not exist
% otherwise use the object that was found.
 if isempty(obj1)
     obj1 = serial('COM1');
 else
     fclose(obj1);
     obj1 = obj1(1)
 end
```

```
% Connect to instrument object, obj1.
  fopen(obj1);

% Configure instrument object, obj1.
% set(obj1, 'BaudRate', 19200);

% Communicating with instrument object, obj1.
warning off;
fclose(obj1);
input = zeros(1,100);
delta = 0.1;
Length = 200;
array = [];
time = [];
fclose(obj1);
% get 10000 samples
a = [];
for cnt = 1:10000
    a(1,cnt) = fscanf(obj1, '%d');
    if a(1,cnt)> 409
        a(1,cnt) = 0;
    end
    a(1,cnt) = abs(2.5*a(1,cnt)/409);
end
```

Appendix F. Directional Thruster Testing

Table 14. 1100 GPH Optimum Thrust Calculations.

Optimum Thrust Calculations Using 1100 GPH Pump No Loses					
Nozzle Diameter (in)	Nozzle Area (sq ft)	Exit Velocity (ft/s)	Thrust (ftlb/sq s)	Thrust (N)	Thrust (lbf)
0.375	0.000766602	52.23041794	99.22156574	13.72234254	3.087527072
0.3125	0.000532362	75.21180183	157.8240947	21.82707229	4.911091266
0.25	0.000340712	117.5184404	265.7060229	36.74714297	8.268107168
0.125	8.5178E-05	470.0737614	1164.722092	161.0810653	36.24323969
0.0625	2.12945E-05	1880.295046	4760.786367	658.4167545	148.1437698

Table 15. 5 GPM Optimum Thrust Calculations.

Optimum Thrust Calculations Using 5 GPM Pump No Loses					
Nozzle Diameter (in)	Nozzle Area (sq ft)	Exit Velocity (ft/s)	Thrust (ftlb/sq s)	Thrust (N)	Thrust (lbf)
0.375	0.000766602	14.21435924	27.00286605	3.734496375	0.840261684
0.3125	0.000532362	20.4686773	42.95137711	5.940175455	1.336539477
0.25	0.000340712	31.98230828	72.31113611	10.00063012	2.250141778
0.1875	0.00019165	56.85743694	135.7427142	18.77321737	4.223973909

Table 16. Single Nozzle Test 1.

Test 1 Single Nozzles		
Nozzle Diameter (in)	Reading on Spring Scale (N)	Thrust (lbf)
0.375	6	1.35
0.3125	5	1.125
0.25	4	0.9
0.125	2	0.45
0.0625	2	0.45

Table 17. Dual Nozzle Test 2.

Test 2 Two Nozzles		
Nozzle Diameter (in)	Reading on Spring Scale (N)	Thrust per nozzle (lbf
0.375	7	0.7875
0.3125	6	0.675
0.25	2	0.225
0.125	2	0.225
0.0625	2	0.225

Table 18. Machined Nozzle 2 GPM Test.

Test 2 Single Machined Nozzles 2 GPM Test			
Nozzle Diameter (in)	Reading on Spring Scale (N)	Back Pressure Reading (PSI)	Thrust (lbf)
0.375 (3/8)	2	0	0.45
0.3125 (5/16)	2	0	0.45
0.25 (1/4)	2	0	0.45
0.1875 (3/16)	4	0	0.9

Table 19. Machined Nozzle 5 GPM Test.

Test 1 Single Machined Nozzles 5 GPM Test			
Nozzle Diameter (in)	Reading on Spring Scale (N)	Back Pressure Reading (PSI)	Thrust (lbf)
0.375 (3/8)	2	0	0.45
0.3125 (5/16)	3	0	0.8
0.25 (1/4)	6	0	1.3
0.1875 (3/16)	8.5	5	1.9

Appendix G. Ballast Tank Sizing

	Name	Density	s_ult
		(kg/m^3)	(MPa)
Wall Material	**Lexan**	**1191**	**65.8**
Safety Factor	**2**		
Op. Pressure	**45**	psia	
	0.310264078	Mpa	
Mass Factor	**1.1**	(for Bladder, mounting components etc.)	
Cylindrical Tank with Flat Endcaps			
Internal Radius	**0.1524**	m	
	6	in	
Aspect Ratio (L/r)	**0.656167978**		
Surface Area(flat end-caps)	0.072965877	m^2	
Surface Area (cylinder)	0.095755744	m^2	
Internal Volume	0.007296588	m^3	
	7296.587714	cm^3	
	445.2651015	in^3	
Wall Thickness	Cylindar	Sides	
	0.001437211	0.002874	m
	0.056583115	0.113166	in
Optimal Tank Mass	0.455071041	kg	

References

[1] Iver Autonomous Underwater Vehicle. (2009). Retrieved March 20, 2009 from http://www.iver-auv.com/

[2] Serafina Autonomous Underwater Vehicle. (2009). Retrieved March 20, 2009 from http://users.rsise.anu.edu.au/~serafina/1-Research/Objectives.html

[3] Roy Burcher and Louiss Rydill; Cambridge University Press 1995; *"Concepts in Submarine Design"*.

[4] *GE Structured Products Thermoforming Lexan Sheet: General Guidelines.* (2008). Retrieved November 12, 2008 from http://www.piedmontplastics.com/static/literature/LEXAN%20Sheet%20General%20Guidelines.pdf

[5] *Buoyancy*. (2008). Retrieved 3 12, 2009, from Physics of Liquids and Gases: http://www.vectorsite.net/tpecp_08.html

[6] *How Submarines Work*. (2008). Retrieved March 12, 2009 from http://science.howstuffworks.com/submarine1.htm

[7] *AUV Design Info Page(2000), retrieved April 20, 2009 from* http://www.ise.bc.ca/WADEhydrocontrol.html

[8] Donald Cowling, IEEE 1996, *"Entire Operating Envelope H-Infinity Design for an Unmanned Underwater Vehicle"*

[9] *Magnetic Couplings.* (2009). Retrieved March 10, 2009 from http://www.dextermag.com/Magnetic-Couplings.aspx

[10] *Design of Rangefinders*. (2008). Retrieved October 2008 from www.superdroidrobots.com

[11] *Extruded Aluminum H-Bar*. (2009). Retrieved February 6, 2009 from www.crlaurance.com

[12] *General Electric Lighting*. (2008). Retrieved April 12, 2009 from http://www.gelighting.com/na/business_lighting/faqs/incandescent.htm

[13] *Thermal Conductivity of Various Materials*. (2008). Retrieved April 12, 2009 from http://www.engineeringtoolbox.com/thermal-conductivity-d_429.html

[14] *High Impact Polystyrene Properties*. (2008). Retrieved April 12, 2009 from http://www.matbase.com/material/polymers/commodity/ps-hips/properties

[15] *Lexan Data Sheet*. (2008). Retrieved April 12, 2009 from http://www.mousetrak.com/download/GIE/TDS/Lexan_500_British.pdf

[16] *PC/104 Embedded Consortium*. (2009). Retrieved April 13, 2009 from http://www.pc104.org/

[17] *Specifications-PC/104*. (2008). Retrieved November 28, 2008 from http://www.pc104.org/pc104_specs.php

[18] *Specifications-PC/104 plus*. (2009). Retrieved November 28, 2008 from http://www.pc104.org/pc104_plus_specs.php

[19] *VersaLogic Cheetah*. (2008). Retrieved September 4, 2008 from http://www.versalogic.com/products/ds.asp?productid=184

[20] *VersaLogic Cheetah Quick Start Package*. (2008). Retrieved April 4, 2009 from http://www.versalogic.com/software/Software_Search.asp?c=5&p=184

[21] *Sensoray Model 311 Frame Grabber Specifications*. (2008). Retrieved September 8, 2008 from http://www.sensoray.com/products/311_311ta.htm

[22] *Sensoray Model 311 Frame Grabber Images*. (2009). Retrieved April 14, 2009 from http://www.sensoray.com/products/311data.htm

[23] *VersaLogic VCM-DAS-1 Manual.* (2008). Retrieved September 5, 2008 from http://www.versalogic.com/Products/Manuals/mvcmdas1.pdf

[24] *VersaLogic VCM-DAS-1 Photo Page.* (2009). Retrieved April 14, 2009 from http://www.versalogic.com/Products/Manuals/mvcmdas1.pdf

[25] *Honeywell HMC205x Data Sheet*. (2008). Retrieved November 23, 2008 from http://www.ssec.honeywell.com/magnetic/datasheets/HMC105X.pdf

[26] *PCB Trace Width Calculator*. (2006). Retrieved October 28, 2008 from http://circuitcalculator.com/wordpress/2006/01/31/pcb-trace-width-calculator/

[27] *National Semiconductor LMD18200T Data Sheet*. (2008). Retrieved November 21, 2008 from http://www.national.com/mpf/LM/LMD18200.html

[28] *Texas Instruments MSP-FET430U Data Sheet*. (2009). Retrieved April 25, 2009 from http://focus.ti.com/docs/toolsw/folders/print/msp-fet430u64.html

[29] *King Pumps FloJet 4300 Specifications. (2008).* Retrieved April 15, 2009 from http://store.waterpumpsupply.com/flquiidepu5g.html

[30] *Anaheim Automations*. (2008). Retrieved March 24, 2008 from http://www.anaheimautomation.com/brush-dc-planetary-gearmotor-bdpg-38-86.aspx

[31] *STC Valve Specifications*. (2008). Retrieved March 12, 2009 from http://www.stcvalve.com/Solenoid_Valve_Specification_2V025.htm

[32] *PicoPSU Data Sheet*. (2008). Retrieved August 30, 2008 from http://resources.mini-box.com/online/PWR-PICOPSU-120/PWR-PICOPSU-120-manual-engl.pdf

[33] *SRF04 Rangefinder Schematics*. (2008). Retrieved October 2008 from http://www.superdroidrobots.com/product_info/SRF04.htm

[34] *Norcross Hawkeye DF2200PX Fishfinder*. (2008). Retrieved October 2008 from www.norcrossmarine.com

[35] *Airmar P28 Fishfinder*. (2008). Retrieved October 2008 from www.airmartechnology.com

[36] *Basic Push-Pull Amplifier Schematic*. (2008). Retrieved October 2008 from www.ktu.lt/ultra/journal/pdf_58_1/58-2006-Vol.1_06-L.Svilainis.pdf

[37] *Amidon Corporation Ferrite Toroids*. (2008). Retrieved October 2008 from www.amidoncorp.com

[38] *MSP430x23x Pinout*. (2008). Retrieved November 2008 from http://focus.ti.com/lit/ds/symlink/msp430f233.pdf

[39] Pinkuan Liu, Yulin Wang, and Jun Wu; Springer-Verlag Berlin Heidelberg 2008; *"Novel Design and Analysis of Magnetic Couplings Used for Vacuum Robot"*

[40] Olivier Chocron and Herv´e Mangel; IEEE 2008; *"Reconfigurable Magnetic-Coupling Thrusters for Agile AUVs"*