NASA Space Grant Robotics

Arizona State University, Tempe, Arizona

Mentor: Shea Ferring
Advisors: Robert Wagner, Zach Schimke, Candace Jackson

CEO: Emily McBryan
Andrew Palmowski – Business Relations
Ben Teoh – Business Relations
Chad Stearns – Financial
Kea Warren – Event Coordinator
Nick DiPalma – Systems
Tina Alam – Interpersonal Relations
Samantha Axtell – Secretary

Mechanical Engineer/ Designer: Jonathan Houda
Eric Tolley – Designer
Erick Yanez – Designer
Isaias Martinez – Mechanical/3D modeler
Jose Velazquez – Mechanical/ 3D modeler
Philip Burbank – Manufacturer
Tyler Achey - Mechanical

Electrical Engineer: Michael Przeslica
Alex Kafka- Technical Advisor
Cody Lopez – Electrical
Jan Christian Chavz-Kortlang – Electrical
Dominic Chen – Electrical Designer
Matt Plank – Electrical

Computer Engineer: Ben Prather
Juan Guzeman – Autonomous Programmer
Anthony Hallas - Programmer
Ben Mackowski - Programmer
Mike Kanaly - Programmer
Shelby Isa – Programmer

Abstract

This year, in 2012, the NASA Space Grant Robotics Corporation is revealing their new and streamline underwater vehicle Koi. Koi has an elegant design that integrates both remote operations and semi-autonomous controls for ease of use and precise movements. Koi moves smoothly through the water with powerful custom thrusters capable of five degrees of freedom including tilt and strafe. To complete the mission objectives, Koi utilizes stereovision, a Seabotix claw, and a custom vacuum system. Koi also comes equipped with a series of sensors for directional aid, a depth guide, and data from the surrounding environment all of which is relayed on the operator's heads up display. The team is composed of a variety of undergraduate students attending Arizona State University. Fueled by challenges from the MATE competition, their application and innovation makes the NASA Space Grant Robotics team a strong force at ASU and a proud representation of the Space Grant Consortium.

FIGURE 1: KOI DESIGN

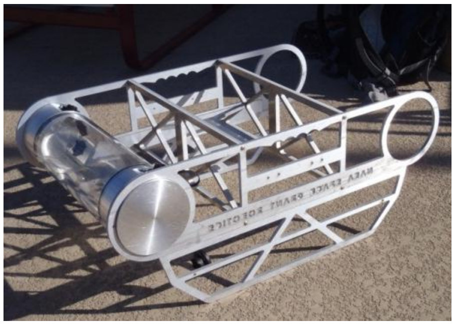


FIGURE 2: KOI

Table of Contents

Budget/Expense Sheet

Donations and Sponsors	**Company**	**Amount**	**Description**
	Space Grant	$2,129.60	Monetary donation
	Orbital	$1,000.00	Monetary donation
	McBryan Federation	$500.00	Monetary donation
	Alpha Wire	$6,500.00	Cable: 800 ft.
	Syntech	$200.00	Syntactic foam
	ASASU	$1,485.65	Components
	Southwest Water Jet	$300.00	Metal cutting service
	SolidWorks	$4,000.00	40 licenses 3D CAD
	Industrial Metal Supply	$100.00	Aluminum metal stock
Total: $16,215.25			
Expenses	**Company**	**Amount**	**Description**
	APASE	-$850.00	NURC Invoice
	MATE	-$50.00	MATE Invoice
	AUVSI	-$500.00	AUVSI Invoice
	SparkFun	-$18.59	Laser
	Adefruit	-$13.65	Temperature Probe
	Jk Devices	-$53.00	Mega Mini
	Vicor	-$238.00	Power Converters
	DIY Drones	-$84.56	9 axis IMU
	HobbyKing	-$24.84	Servo motors
	AndyMark	-$68.04	Auto reset fuses
	MDFLY electronics	-$59.45	Bd25 Breakout Panals
	Mouser	-$324.00	Bulgin Connectors
	Dimension Engineering	-$150.00	Speed Controllers
	McMaster-Carr	-$200.00	Polycarbonate Enclosures
	McMaster-Carr	-$50.00	3/4 in End mill
	McMaster-Carr	-$75.00	Fasteners
Total: -$2,759.13			
Previously owned components	**Company**	**Amount**	**Description**
	SeaBotix	$2,000.00	Seabotix Claw
	SeaBotix	$3,000.00	Seabotix Thrusters
	Scorpion	$550.00	Custom Brushless thrusters
Total: $5,550.00			

Borrowed components	Company	Amount	Description
ASU professor Dr. Saripali	Point Gray Research	$1,600.00	Stereo Camera
Total: $1,600			
Travel	**Company**	**Amount**	**Description**
	Wyndon Hotel	$1,500	3 hotel rooms
	Car rental	$500.00	Car rental
	Air fare	$5,000.00	Air fare for 10 team members
Total: $7,000			
Total value of ROV Koi:	**$6,921.63**		

Electrical Schematic

See appendix A - H

Design Rationale


FIGURE 1: KOI DESIGN

The design of Koi started with a look at what the customer, MATE, wanted to complete their mission of surveying the SS Gardner. The company brainstormed and created a comparison table of needs, wants, and wishes. While considering the impact of finances, time, and team member ability and performance this table of comparisons slowly became the design of our ROV Koi. Koi maintains the basic needs, or requirements, to accomplish the mission while possessing several advanced systems and designs that make it more than a basic ROV. The following section goes into detail of the design process from the different departments.

Hardware Design:

In order to create a robot which is both neutrally buoyant and hydrodynamic, our mechanical division opted for a simple symmetric design with a bow and stern container housing the electronics. This allows for easy assembly and disassembly of the robot and maintains balance. Also inside the electrical enclosures are cameras, this maximizes the use of the enclosure space and limits points of failure on the ROV regarding water leaks. The camera systems inside the enclosures are also mounted on a pivoting structure to provide a varied view point; this drastically increases the visibility from ROV Koi.

The frame is composed of two separate parts; the top part contains all the bare requirements for the ROV to run: computer, cameras, and thrusters, while the bottom half is for the payload. This design provides modularity to Koi since the payload can be switched out for any given mission with just a few bolts and wire connectors. The location for the payload and manipulators offers ample accessibility for Koi to observe and manipulate the wreck site or any other mission. Koi is controlled with five thrusters providing five degrees of freedom including tilt and strafe.

To maintain buoyancy of Koi, a block of syntactic foam is used along with the air inside the two large electronic enclosures. The design of Koi's structure puts all sources of buoyancy above the lines of the thrusters and symmetric about the frame and center of mass.

Design Rationale - Electrical

The electrical systems are divided between the two cylindrical enclosures. From its inception, the plan was to give this ROV extensive autonomous capability beyond what the company has attempted previously. Having two enclosures allow for easy replacement and upgradeability of the onboard computer and cameras without requiring the rearrangement and rewiring of more fundamental components.

The stern enclosure contains a reliable electrical system our team has perfected in previous ROVs (including thruster and motor control, sensors, and power management). The power tether brings fused

48VDC power from the surface into the stern enclosure. One Vicor DC-DC converter outputs 12VDC from supplied 48VDC. The 12VDC is used to power an ATMEGA2560 microcontroller, two Sabertooth 2X5 RC brushed DC motor controllers, and an IMU (inertial measurement unit) via a 12VDC to 5VDC linear regulator.

The ATMEGA2560 serves as the hub for electromechanical robot functions. It is an Arduino compatible microcontroller, ideal for our ROV for because of its robust hardware, extensive public codebase and documentation. It controls motors through digital PWM (pulse width modulation) output and receives data from the IMU and other sensors. The IMU used is an ArduIMU3+, designed originally to convert RC (remote-controlled) planes into UAVs (Unmanned Aerial Vehicle). It contains an ATMEGA328 microcontroller (Arduino compatible) to manage a 6-axis gyroscope/accelerometer sensor and a 3-axis magnetometer. These sensors provide information to the main ATMEGA2560 for semi-autonomous functions and transmitted to the surface for telemetry. The advantage of this IMU is that it offloads the sensor processing to the microcontroller on the IMU rather than sending raw data to the main microcontroller on-board the ROV. It is reprogrammed to provide data in a format most convenient to the programming team. The IMU sends data over a serial connection to the main ATMEGA2560 and requires 5VDC from a linear regulator.

For thrusters, our ROV utilizes five brushless “Scorpion” brand motors. There are significant advantages to using brushless motors as thrusters compared to brushed DC motors. First, the design of a brushless motor makes it inherently waterproof because there is no electrical contact between the rotating and stationary portions of the motor. This reduces the complexity of the thruster design because waterproof housings and shaft seals are not required, unless working in hazardous environments that would lead to the degradation of the enamel coating on the brushless stators. Additionally, brushless motors provide significantly higher thrust and torque compared to an equally sized brushed DC motor. A power link between the bow and stern provides 48VDC and 12VDC to the bow enclosure from the stern power tether connection. Three 48VDC to 5VDC Vicor DC-DC converters in the bow enclosure provide up to 40 amps of power through the wiring harness to supply the thrusters. Each brushless motor draws a maximum of 10 amps in the water.

Extra care must be taken, however, to keep the motor coils and windings of the brushless thruster clean as part of regular scheduled maintenance. For safety reasons, our company conducted extensive underwater testing prior to implementing brushless thrusters on the ROV. Our company concluded that even after repeated water use, the brushless motors do not conduct through the water because the wrapping of motor windings is enamel coated and insulated from the water.

Five TURNIGY brand motor controllers are used to run the brushless thrusters. Although the brushless motors are ‘sensor-less’ i.e. without a position sensor or encoder, the controllers must be within six inches of their respective thruster because they read feedback along the motor connections in order to provide proper motor timing based upon the load on the motor. Each controller is cast in 3M Scotchcast 4, 2-part plastic for waterproofing. Heatsinks required to cool the controllers are exposed outside of the Scotchcast to dissipate heat into the water. Each controller requires a 5V power and ground line, as well as, a PWM signal, signal ground, and a 2 wire jumper used to reprogram their onboard microcontrollers. These connections are all tied into a 25-conductor cable that interfaces with the stern electronics enclosure. Only the PWM signal lines and one ground are required for normal robot operation. The wires leading to the jumpers only need to be connected during initial robot testing.

Koi also contains two dual-channel Sabertooth speed controllers in the stern enclosure. Each module can control two brushed DC motors, saving a significant amount of space in the enclosure. The controller is capable of peak output current of 10 amps and a normal output current of 5 amps from 6 to 12 VDC. They also feature auto-centering code which automatically trims attached motors by assigning the zero output value to the first signal the microcontroller provides. Each Sabertooth speed controller is protected with a 10-amp auto-reset fuse. Two signal lines pass from each speed controller to the Arduino MEGA microcontroller. Two controllers allows us to have a robotic claw and three additional brushed DC motors for other mechanical actuators required by the mission.

To monitor pressure and temperature outside the enclosure, an external sensing module was built from an ATMEGA328 Arduino microcontroller. This Arduino consolidates the sensor data and sends it to the main ATMEGA2560 via a waterproof connection to the bow enclosure. This setup offloads the sensor processing to the external Arduino, freeing up processor time on the main ATMEGA2560. It also allows the entire sensor package to be interfaced with one 12-conductor cable to the bow electronics enclosure. The attached sensors are: temperature, water pressure sensor, and a temperature probe. The probe can withstand high temperatures and is mounted on the robot claw, so it can be inserted into tight spaces to measure hot objects or water currents.

Six waterproof Bulgin connectors on each enclosure allow data and power to pass in and out. The pin wiring in each connector is designed to be backwards compatible with the cable assemblies built previously for our other ROV models. The external lights, and external sensors can be universally mated to any of our ROV models. While most of the connectors have different numbers of pins to prevent connecting the wrong cables together, there are also labels on each interface to make it easy for any team member to assemble the robot without a wiring diagram.

The bow enclosure contains a forward-facing cameras and an ECX form-factor, Intel PC. The ROV can be teleoperated using only the electronics in the stern enclosure, but the addition of the PC and a Bumblebee stereo camera system allow for fully autonomous functionally through ROS (Robot Operating System) running on the PC. If autonomous operation is not desired on a mission, the ROV can be configured such that the computer in the bow enclosure can be shut down and the entire ROV operated through the ATMEGA2560 microcontroller via serial communication with the surface. The on-board PC sends instructions and receives feedback from the robot systems through a serial link with the stern enclosure's ATMEGA 2560 microcontroller. The PC is powered with 12VDC supplied via the power cable linked to the stern enclosure.

The Bumblebee 2 is a stereovision camera from Point Grey, which outputs high-contrast VGA signals via a firewire digital interface. This digital signal is processed by the on-board PC for autonomous functions and/or streamed to the surface along with communication signals over the PC's Ethernet connector. The stereo camera is attached to a 180-degree servo, which allows the camera to tilt vertically to look either above or below the ROV. Gimbaled on the same axis is a 5mW green laser, which can be switched on and off using the ATMEGA2560 in the stern enclosure. The laser is used to pinpoint targets and objects too far for the LED lighting to illuminate. Since it always points in the center of the camera's vision, precise aiming of the laser pointer is possible.

Inside each enclosure is a polycarbonate sheet on which the electronics are mounted. Servicing the electronics has been made easier by routing wires leaving the enclosure through a DB-25 connector. Removing an end cap of the enclosure will automatically disconnect this connector. Sliding the enclosure back together will reconnect the DB-25, similar to installing PCI cards into a desktop personal computer.

Design Rationale – Programming

Software

Koi is controlled in two different code blocks, a surface side code run from a computer and an onboard Arduino microcontroller. Both sides communicate through a serial port with data traveling between Koi's tether. Our surface side code is written in Java while the Arduino is coded in C/C++. Our programing team decided to use Java because of its cross-platform capabilities and the employees' expertise. We chose to use the Arduino microcontroller for similar reasons. It is cross-platform and cost efficient.

Surface Side

Our programing team built a system that was easily accessible and able to run on multiple machines. While designing the code from the ground up, one focus of the programming team was to build the code with many smaller classes that worked together instead of writing one long class that had all the functions. The goal was to create code that was easy to read and easy to modify. With small classes and packages, users can quickly locate where certain functions are, modify them, add to them, or delete unneeded code without the risk of causing errors in other functions.

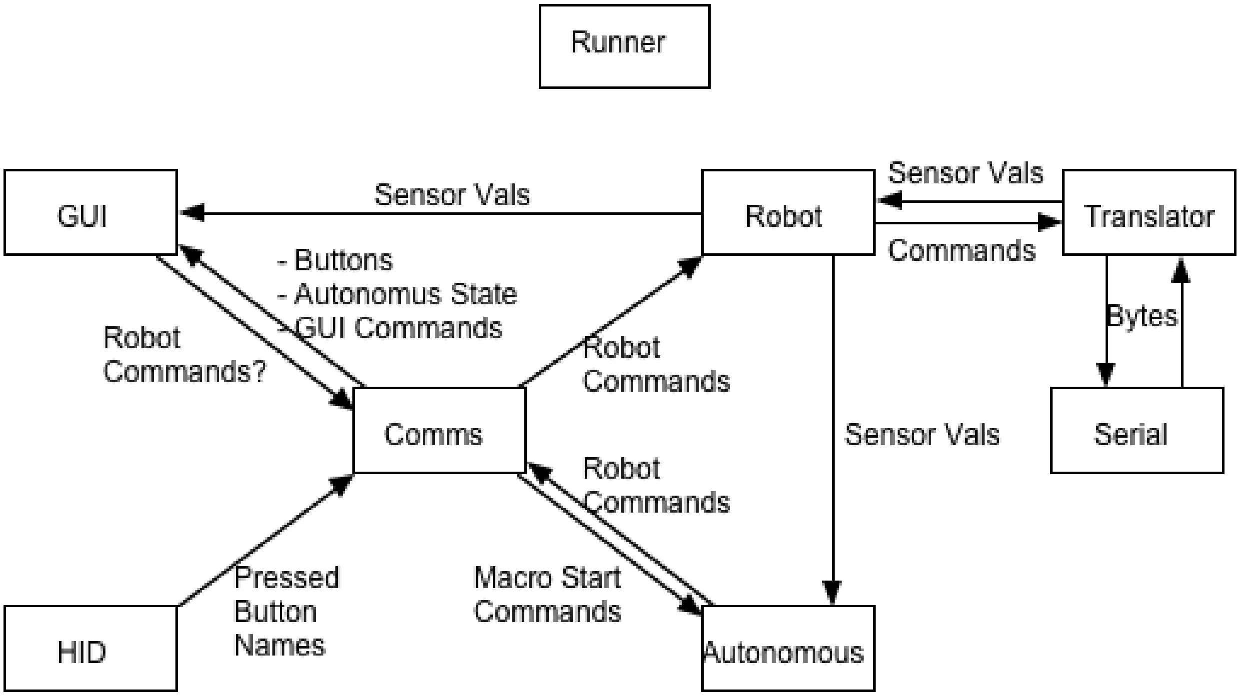


FIGURE 3: GENERAL DESIGN WHICH SERVED AS THE BASIS FOR OUR CODE

Each class would have its specific purpose and only communicate with classes that it needed to. The design expanded into three larger packages, HID, GUI, and CORE, which contained the rest of the classes. The HID package contains classes and libraries for the use of a keyboard and Xbox 360 controller which is compatible with Linux, Mac OS, and Windows. The GUI package contains the necessary functions to handle incoming sensor data and properly display them on the custom-built GUI screen. The CORE package is responsible for taking all human input and Koi's input and return the appropriate responses.

The entire system is controlled by an Xbox 360 controller. This was chosen because of the wide range of input mechanisms it offered combined with the ability to leverage existing operator experience. While the main system is controlled with the Xbox 360 controller, we also utilize keyboard functionality to activate less-used functions, such as semiautonomous commands.

Arduino

The programing team decided to use an Arduino microprocessor to handle all communication on Koi. The Arduino offers multiple connectors that handle data separate from the PC. Due to the Arduino’s open source nature it comes with well documented example code from other users. This means our programming team can use the Arduino for each and every situation and I/O device required by the application.

The Arduino is responsible for receiving all the high-level commands from the surface side code and interpreting them. It then sends appropriate commands to each individual motor to simulate what the driver wanted Koi to do. In return, the Arduino gathers all raw data from our sensors and passes them back to the surface side code to be displayed to the driver.

GUI

The Graphical User Interface, or GUI, is an important feature of the surface side code. The design was focuses on being simplistic and effective. All data passed from the Arduino is sent to the GUI to be displayed for the driver to see. Critical information such as depth and current direction are present to guide the driver through any environment. Other relevant information as the mission timer and data from sensor probes are presented in a clear manor that our drivers need for completing missing tasks in a timely manner. On the simulated screen are other useful notifications that the driver will encounter. In the top left corner notifications are displayed when semi-autonomous functions such as hold depth and tilt lock are activated. A notification will pop up in the middle of the screen if communication with the robot is broken during the run time. An message box at is also present to give any other relevant information to the driver

```
Temp: Probe: 0.0 ï¿
Temp: Aux: 0.0 ï¿½
Compass: 0.0
Acc_X: 0.0 m/s2
Acc_Y: 0.0 m/s2
Acc_Z: 0.0 m/s2
Gyro_X: 0.0 d/s
Gyro_Y: 0.0 d/s
Gyro_Z: 0.0 d/s
Pressure: 0.0 hPa
```

```
GUI Successfully Constructed
Using controller: Keyboard
Serial initialization failed: No such port
Simulating serial...
```

FIGURE 4:GUI INTERFACE WITH A SIMULATED BACKGROUND

Safety Features

Koi has several safety features to allow it to shut down in case of signal or power loss. The Arduino microcontrollers are programmed to shut down after 1.5 seconds without a signal from the surface. Both the TURNIGY brushless motor speed controllers and the Sabertooth brushed motor speed controllers shut down and reset, cutting power to the motors if the signal line does not receive any signal.

Our frame also included a few safety features. The most obvious is the handles that are embedded into the frame, giving the people who carry the robot a safe and comfortable place to grab, which was also important so as the robot would not be dropped. All sharp edges of the robot have also been smoothed out so that no one would cut himself or herself. Also, plastic skids have been placed on the bottom of the robot so that when it comes in contact with the floor, nothing will be damaged.

Challenges

Technical

The biggest technical challenge faced on the mechanical department was how to manufacture the complex waterproof end caps for the electronics enclosure. To maintain a hydrodynamic shape, the waterproof connectors had to come out along the circumference of the end caps instead of the flat end in order for the wires to maintain in the ROV frame perimeter. This was especially difficult because the Bulgin connectors require a flat spot to be mounted to. As such, the manufacturing team had to machine flat spots along the inside and outside circumferences of the end cap at precise angles. Initially the use of a CNC machine was considered but due to availability and cost, we decided to machine the end caps ourselves. After talking to expert machinists on the best way to proceed, and being discouraged from the amount of time we were told they would take, our manufactures worked through the problems and finished the end caps in a timely fashion to be implemented on Koi.

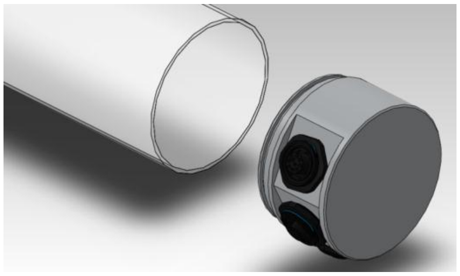


FIGURE 5: END CAP

Finances

Finances are always an issue for a nonprofit organization, but this year with the economy in its current state it was especially difficult at times to support the NASA Space Grant Robotics team and maintain production. The funding for the team was cut significantly in the spring, this created a big challenge for the CEO and the financial team to keep the Koi project a float. This challenge was met with determination and persistence as members approached major companies with a request for donated items and monetary donations. The technical departments focused on reusing old parts and recycling materials to keep building up the vehicle Koi. A number of proposals were written and a great deal of meetings where held with corporate educational offices in the attempts of receiving sponsorship. This year contains the highest number of market value donated items in the history of the company. Through the support of the sponsors and the persistence of the CEO, the team has never had more companies represented on the ROV!

Electrical

The greatest challenge faced by our company in the last year has come from our computer hardware suppliers. As we are a small specialized company, we often rely on parts that no longer have the demand required to keep them conveniently stocked. As such, we often encounter long delays in the delivery of our hardware which directly impacts our production and testing capabilities. For example, there was a recent occurrence in which we discovered that our Seabotix thrusters had reached their end-of-life and were not repairable. Being one of the most popular off-the-shelf thrusters on the market, the KOI robot was designed specifically to use this thruster. Due to the hardware's high price and lead time, we were forced to go back to step one in terms of thruster design. This not only crippled our ability to test and verify the manual controls of KOI, but also forced us to push back the development of the autonomous system that was promised to our customer. Because of the designed interoperability of our ROV systems, it took only a few days to convert KOI to use brushless thrusters.

Delays like the ones we faced in the past year are unacceptable and do not represent the service and product that NSGR is capable of. In order to overcome similar future inconveniences we will continue to design ROVs with exchangeable parts and interoperability of various electrical systems. In this way, we can deliver a capable final product.

Programming

One of the biggest challenges faced by our Programming Division was whether or not to use the existing legacy code. While that code was fully functional, it was neither properly documented nor organized. Moving forward, this legacy code was deemed unacceptable and it was decided to rewrite a new code from scratch. We aimed to create a smaller code that consisted of multiple classes that spoke to each other instead of one monolithic class containing all functions.

This was a large task which took up the first couple of months of our design process. Extensive research was done to find newer Java libraries to use with up-to date drivers that could be used. After the initial program was designed, the workload was divided so each employee was responsible for implementing one or two classes. In order to keep each employee up-to-date the company utilized a Subversion repository to synchronize the code between every employee's system.

Troubleshooting Techniques

Fixing a Seabotix Thruster:

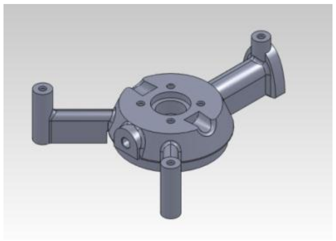


FIGURE 7: THRUSTER PART

FIGURE 8: ASSEMBLY

FIGURE 9: INTEGRATION

The company poses five first generation Seabotix thrusters that were donated to the team four years ago. These thrusters being over 7 years old with many miles on them started to falter in performance and during the build season a thruster broke. Instead of purchasing more Seabotix thrusters or acquiring thrusters from a different company, the Managerial department of the company chose instead to repair the existing thrusters. The problem, however, was that these thrusters were badly damaged from misuse by their previous owners. The Manufacturing department was then tasked with the reconstruction of the damaged thruster.

First, the team isolated the specific part which was causing the problem. The support piece that held both the motor in place as well as kept the protective housing for the fan blades had broken in the three critical support structures. We took the part, salvaging what was left and created a 3D model of the part in Solid Works as shown in figure 10. This piece was then sent it to a 3D printer in the design school of ASU. The first model had some small dimensional errors, which were corrected in the second print. This second print fit exactly in position. The repaired thruster was test on a previous ROV, Aquadevil. The thruster was able to move the mass of the ROV for a few hours before the seal was damaged and water flooded the repaired thruster shorting it and effectively destroying the brushes within the motor. The Manufacturing Department then manufactured new brushes. The team also learned about the built-in vacuum testing nozzle in the thruster where the integrity of the seal could be checked. We practiced the manual replication of a complex part. The Seabotix thruster is still being troubleshooted to fix the leak issue.

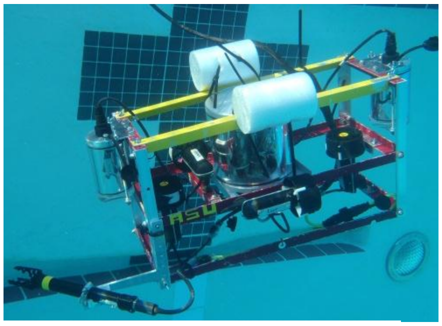


FIGURE 10: FULL ROV TEST

Payload Description

Claw

In order to maintain the delicate balance of our ecosystem, Koi has to relocate an endangered species of coral that is found on a section of the hull above the fuel tank on the *SS Gardner*. The overall task of transplanting the coral itself is not a difficult one, however as the coral is of an endangered species, precautions must be considered to ensure the safety of the coral. A Seabotix claw was utilized to complete this task but since the claw is not meant for delicate handling, it was lined with spongy material to both grip the coral better by enhancing the surface area contact and providing a softer touch to prevent damage.

FIGURE 11: CLAW

Magnetic debris pile markers

FIGURE 12: MARKER

Our company developed a simple yet effective method for finding and mapping which debris piles are ferrous. Five tubular manipulators are arrayed along the front of the ROV. Inside each is a powerful neodymium magnet attached to a folded, pink flag. A weak plastic seal covers the front of the tubes to keep the magnets from inadvertently deploying due to ROV movement. When the tube touches a ferrous debris pile, the magnet pushes through the plastic and attached to the pile. The bright pink flag makes the ferrous piles stand out from the rest of the sea floor. It allows future missions to quickly identify them for closer investigation.

FIGURE 13: PROCEDURE

Fuel Retrieval

A main goal of the mission was to remove the hazardous materials from the wreckage. The ROV charged with accomplishing this mission must have a system which can remove the fuel left inside the shipwreck's tank and to devise a way of maintaining the structure of the fuel tank in order to prevent the entire hull and consequently the vessel from collapsing. Our company devised a system for removing the hazardous fluids from the tank using a bilge pump that will force water into the tank which will in turn push out the 1.5-liters of hazardous fluid into a Platypus holding bag of 2-liter capacity. A camera mounted above the holding bag on the ROV monitors the purging process. When the bag is completely filled the pump is turned off and the ROV surfaces and delivers the holding bag to officials for inspection and disposal.

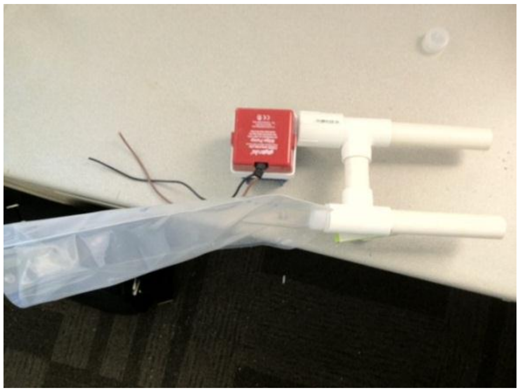
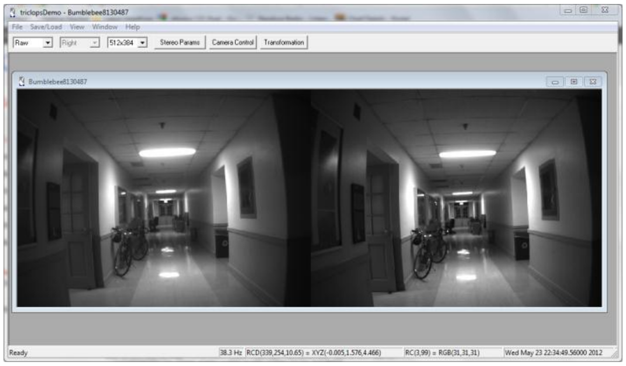


FIGURE 14: FUEL RETRIEVAL

Stereovision

Koi has a Stereo Vision module which adds another dimension to what the ROV perceives and extended functionality without impeding on current onboard systems. The left figure is a screen capture of the left and right images captured by the Bumblebee®2 camera from Point Grey Research. The figure to the right demonstrates a depth-view image indicated by the color of the surface – the darker the color the further that surface is from the camera and the brighter the color towards red indicates a closer surface. Now back to the left figure, the hallway extends approximately 26 meters according to rough measurements. The darkest blue pixel on the right image (near center) measured 25.837 meters long according to the vision software. Koi will use this same technology to measure the length of the shipwreck while surveying the ship on its side.



Ready | 38.3 Hz | RCD(339,254,10.65) = XYZ(-0.005,1.576,4.466) | RC(3,99) = RGB(31,31,31) | Wed May 23 22:34:49.56000 2012

FIGURE 15: TEST

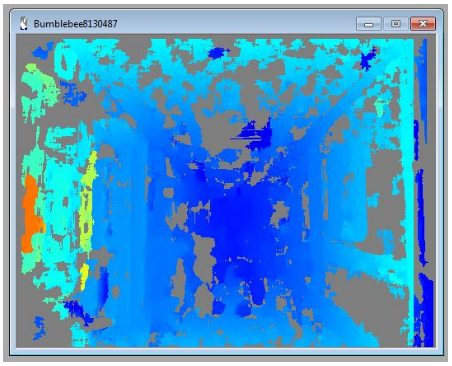


FIGURE 16: CAMERA DATA

Future Improvement

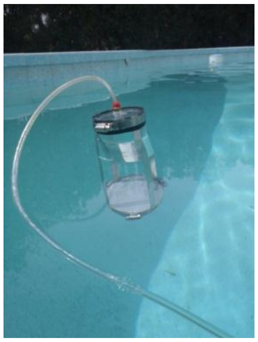


FIGURE 17: SCUBA

For future use on our ROVs, the company has begun research into variable buoyancy devices. One such device we are developing, named SCUBa (System for the Control of Underwater Buoyancy) uses rigid walled containers that allows air to be pumped in or let out, with one end open to the water. This will cause a change in volume of air inside the containers and thus change our buoyancy. We believe such a device that would allow us to control our buoyancy would be a great asset in the control of our robots. It will enable us to remain neutrally buoyant in environments of different densities, which would be helpful when switching between salt and freshwater environments. It will also allow the neutral buoyancy to adjust when mass changes, in cases where an object is being picked up. This consequently allows the lift of heavier objects underwater as well. Furthermore, it will provide the ability to ascend very rapidly by setting the system to maximum buoyancy, working in tandem with the already existing thrusters. All these things in combination will make having precise control over our buoyancy much more beneficial system than the traditional static buoyancy found in most UROVs.

Lessons Learned

I have been a member of this robotics team for four year now and this year the greatest lesson I have learned is not how to correctly seal a waterproof container or how to compile a complex code, rather it is how a team can come together and combine their talent s in order to fulfill a mission. Koi is a great representation of not just our engineering skills but our teamwork skills and leadership ability. As the CEO I learned more from the team members than I thought possible. The greatest prospective I gained was that all team members, no matter what they work on, deserve the same level of respect and consideration. Rather someone is designing an electronics enclosure or planning the next team building exercise, everyone deserves a respect and that keeps us professional. – Emily McBryan

Electrical-Tether Shielding

While we often use older, yet reliable, parts. This year we discovered the importance of verifying the integrity of our wires as well. The tether for one of our older model robots often transmitted an excess amount of noise. This impacted the programming department’s ability to read the data being transmitted from the robot and thus affected the development of the robot’s manual controls. In the past year we switched to a higher quality cable from *Alpha Wire* which was used in the development of the tether for the KOI robot. This higher quality cable led almost zero noise transmission**.**

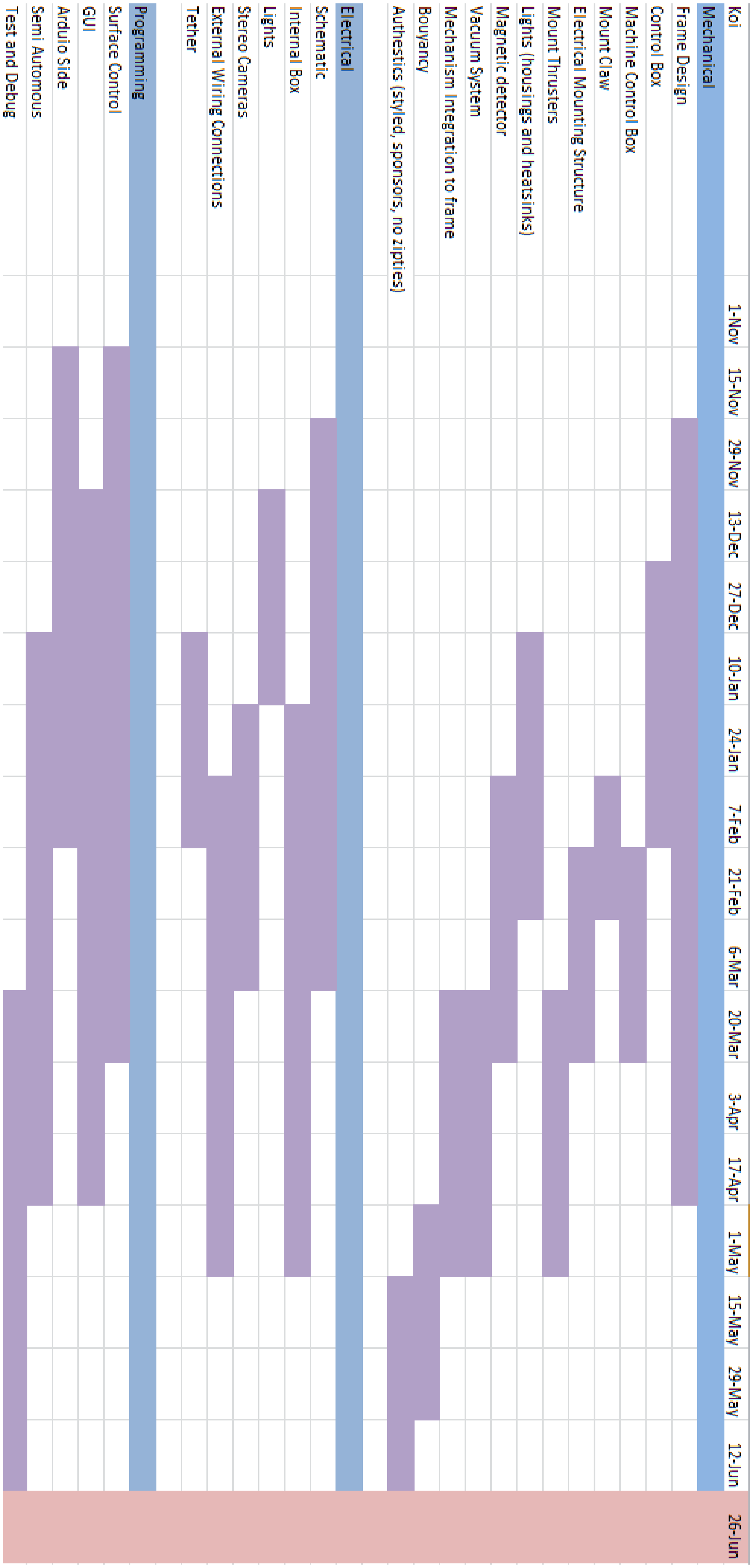
Programming

A major lesson learned was to split up programs into manageable parts. The file that was used last year that handled commands from the joysticks was split into only 2 classes with 2246 lines of code between them. The problem with this is that it is very hard to sift through when adding new functions and classes or fixing bugs. The programming division spent the first half of the development period going through the code, splitting it up into smaller files, and editing it. This effort diverted valuable programmer time away from adding new functionality, such as semiautonomous control.

Reflections

During my first year with ASU’s NASA Space Grant Robotics Team I learned the important skill of machining your own parts. During my previous high school experiences, we always relied on parts that were already made, and because of that, we were always very limited on the resources available and what we could accomplish with them. With the proper training we are free to machine anything we need. With resources like this, I am not limited to use object that we have, but now it is possible to make specific pieces that fit and work exactly how I need them to. -Anthony Hallas

Gantt chart



Koi	1-Nov	15-Nov	29-Nov	13-Dec	27-Dec	10-Jan	24-Jan	7-Feb	21-Feb	6-Mar	20-Mar	3-Apr	17-Apr	1-May	15-May	29-May	12-Jun	26-Jun
Mechanical																		▨
Frame Design			■	■	■	■	■	■	■	■	■	■	■					▨
Control Box					■	■	■	■										▨
Machine Control Box									■	■	■							▨
Mount Claw								■	■									▨
Electrical Mounting Structure									■	■	■							▨
Mount Thrusters											■	■	■	■				▨
Lights (housings and heatsinks)						■	■	■	■									▨
Magnetic detector								■	■	■	■							▨
Vacuum System											■	■	■	■				▨
Mechanism Integration to frame											■	■	■	■				▨
Bouyancy														■	■	■		▨
Authestics (styled, sponsors, no zipties)															■	■	■	▨
																		▨
Electrical																		▨
Schematic			■	■	■	■	■	■	■	■								▨
Internal Box							■	■	■	■	■	■	■	■				▨
Lights				■	■	■												▨
Stereo Cameras							■	■	■	■								▨
External Wiring Connections								■	■	■	■	■	■	■				▨
Tether						■	■	■										▨
																		▨
Programming																		▨
Surface Control		■	■	■	■	■	■	■	■	■	■							▨
GUI				■	■	■	■	■	■	■	■	■	■					▨
Arduio Side		■	■	■	■	■	■	■										▨
Semi Automous						■	■	■	■	■	■	■	■					▨
Test and Debug											■	■	■	■	■	■	■	▨