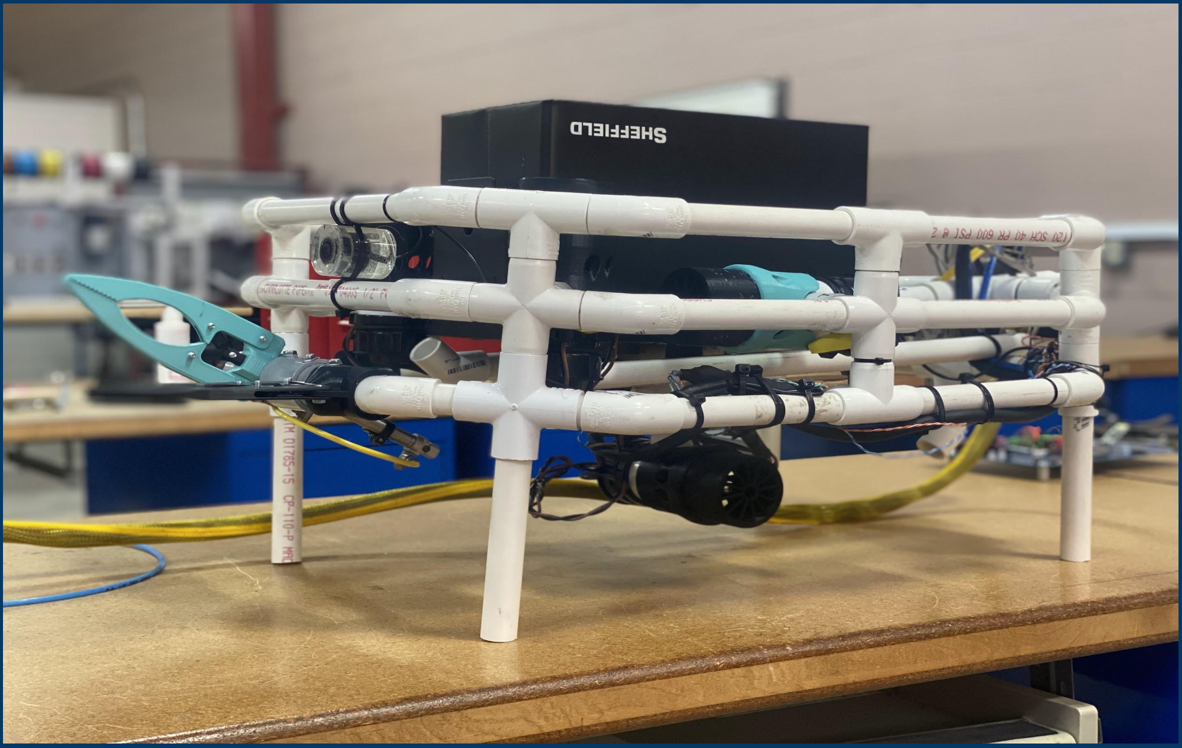
Savannah Tech Robotics

Technical Documentation

2022 Mate World Championship
UN Decade of the Ocean: MATE
Inspired ESG

Long Beach City College
E Carson St, Long Beach, CA 90808
June 2022

Distance Travelled 3,449.92 km



Meet The Savannah Tech Robotics Team

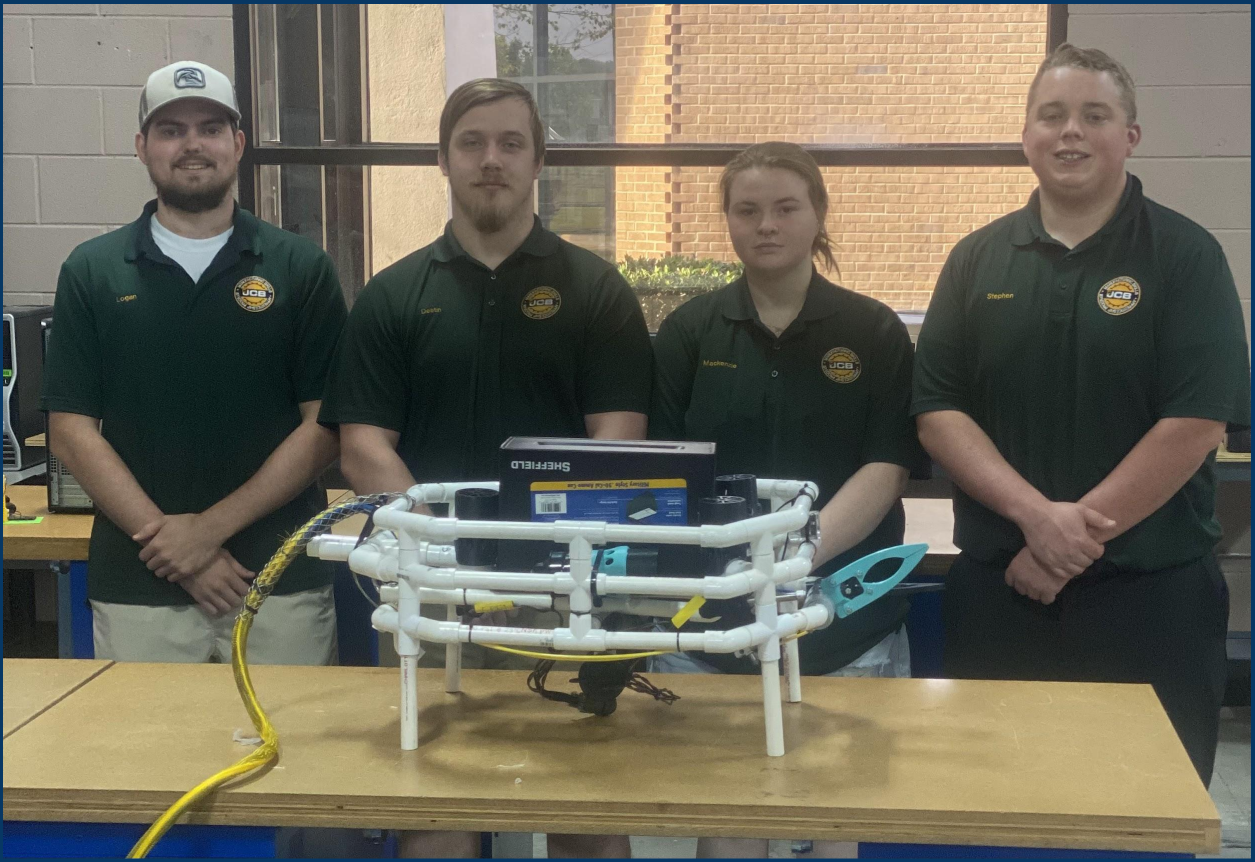
Destin Cramer, 22 Sophomore - *CEO, Pilot, and Lead Software engineer*

MacKenzie Cramer, 22 Sophomore - *CFO , Lead Electrical Engineer, Co Pilot*

Logan Lawson, 20 sophomore - *Director, Props Mgr, Mechanical Engineer*

Stephen Grant, 21 Sophomore - *Safety Officer, Tether Mgt, Mechanical Engineer*

George Shami, 19 Sophomore/ Dual enrollment - *3D Designer*



Abstract

This is Savannah Tech Robotics first year competing in MATE ROV and we will be competing in the Pioneer class. Savannah Tech Robotics consists of four teammates who are from the Savannah, Georgia area, all of which have invested over 300 hours into ensuring the completion of our ROV named “Plank”. While the team has faced multiple roadblocks and issues along the way, overcoming these were only one small step in the process.

Our ROV was built by a team of four members who worked around the clock to make sure plank was up to performance standards. Whether it was the tether, frame, or the manipulator, everything was done a certain way for a specific reason.

Savannah Tech Robotics engineered and manufactured Plank, a remote operated vehicle (ROV) with the capabilities to successfully complete each task given by the MATE Center. Plank was designed with the idea of simplicity, which the team kept true throughout the entire process of manufacturing this robot. Plank has the capabilities to inspect and troubleshoot items of interest, while also maneuvering exceptionally through the water to complete whatever task it is presented. Plank also boasts a lightweight and durable frame, allowing it to slide through the water with ease. All of these elements allow Plank to be the best in its class, making it the obvious choice for future underwater research and development.

Table Of Contents

Design Rationale

Engineer Design Rationale

Plank was designed with a lightweight frame consisting of PVC and 3D printed components. The design has an octagonal shape which consists of three rows of PVC that allows the team to add attachments and also provides stability. Holes are also put in specific locations along the frame to allow water to travel throughout to help with neutral buoyancy. Plank is also cost efficient as the whole frame is made up of PVC and 3D printed parts.

The tasks that were given to us by MATE were always the focal point of when we were designing Plank. Many ideas were brought up and dropped before we all came to a final design that would work best for all the tasks given. Safety was also on the forefront of everybody's minds. We shrouded all of the fans with our own 3D design so that nothing would be trapped or cut up by the blades. The Manipulators were 3D printed and are controlled by pneumatics.

Innovation

Started with the Barracuda Kit for $1260.00 and Redesigned the controller

with a custom board, arduino Mega, and rs485 to send speeds and directions to another custom board in the rover that then controlled the motors. This reduced the cost of the controller and tether by reducing the cost of the boards needed and allowing us to use just a 2 wire cable for power and an ethernet cable for communication. Since we only needed one wire for communication, it also allowed us to run power and data for the cameras through the same ethernet cable. In addition to cutting the cost of the boards and tether, using fewer cables in the tether also made it noticeably more flexible.

Control/Electrical System

Surface Control Box

The control system is housed inside a compact, Seahorse brand protective equipment case. Then power comes in through a modified backplane, then connectors route the power to the tether, display/ cameras, and buck converter for the controller boards.

The Buck converter reduces the 12 V into 9 V to provide an input voltage. The buck Convertor is connected to a 2 pin connector on the custom controller board to provide power to the arduino.

The custom controller board provides a simplified way to connect the 2 joysticks, RS485 transceiver, panic switch, and power to their pins on the arduino.

The right joystick controls the left and right thrust motors, which are indicated on the ROV by their blue shrouds. This joystick will allow both motors to travel at the same direction and speed, which in turn allows for smooth travel through the water. Moving this joystick left or right will modify allow the thrusters to operate at different speeds, which will cause the rover to turn in place or while moving forwards or backwards.

The left joystick controls the crawling motor and all three vertical motors. Up and down is vertical and left and right controls left and right crawling.

The panic button is a latching pushbutton and if activated, it will set all motor speeds to stop which will allow safe recovery of the rover. This also allows the team to perform safe in-water repairs, or remove foreign objects from being tangled or caught on the rover.

The arduino reads the values from the logistics and pushbuttons in the control box. After this, it calculates the speeds and directions for each motor. These calculated values are then stored in a structure used by the Easy Transfer Library that we chose. We chose this to facilitate reliable serial communication. This structure is then sent through the RS485 to the RS485 board in the rover. After pausing momentarily, the arduino then repeats this process.

Our display receives power from the intended header on the backplane board. The data and power for the cameras is also pulled from the backplane board. We reused the 12 pin connector from the Barracuda kit to run power and data from the controls and used RCA to 2 wire. We then used a barrel pin to 2 wire adapters to connect the cameras to the ethernet cable in the tether.

Picture on the left: Control box

Onboard Electronics

Our onboard electronics are housed inside of an ammo crate that has a waterproof seal when closed. When unplugged from the main power connectors it allows us to perform quick maintenance if necessary. After passing through the strain relief and entering the ROV, the power and data wires for the cameras are routed to each of the cameras. Then the RCA and barrel pin adapters are used to connect the cameras to the ethernet lines. These connections were sealed by applying a layer of silicone, then hot glue, and finally by covering it with epoxy lined heat shrink.

The power and data for the motor controllers is passed into the waterproof container through an eleven pin milspec amphenol connector. The backshells of the milspec connector were sealed with silicone and then hot glue.

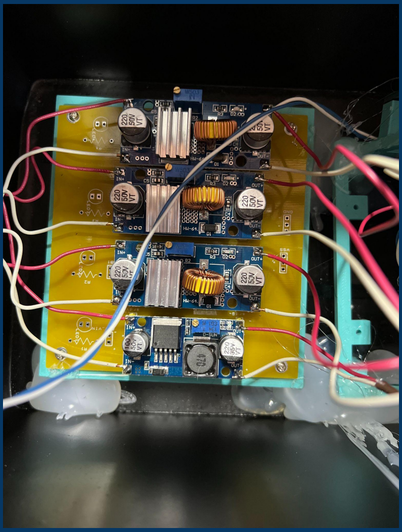
On the inside of the box, power is routed from the connector to a custom board with four buck converters. The first buck converter provides 9V output for the arduino on the custom motor controller board. The other three buck converters all provide between 7 to 12 volts to each of the three H-bridges. The data line from the RS485 is run from the connector to the RS485 on the custom motor controller board.

The custom motor controller board inter connects the Arduino mega and RS485. Initially we designed the board to use single chip H-bridges. Due to our higher current requirements we chose to use Driver boards. We used sockets to create interface cables to connect the driver boards to the original H-bridge sockets. This had the added benefit of being able to simply unplug a faulty driver and plug in a replacement without the need to desolder the pins.

The arduino receives the structure packet from the RS485 and the EasyTransfer Library converts it to the speeds and directions for each motor. The arduino then writes these values to each of the driven channels.The arduino pauses briefly then repeats the recieve, convert, write process.

The driver boards have two channels each. Board one controls vertical 1 and the crawling motor. Board two controls vertical two and left trust. Board three controls vert three and right thrust. We put the vertical motors on separate drivers to reduce simultaneous current draw from each board.

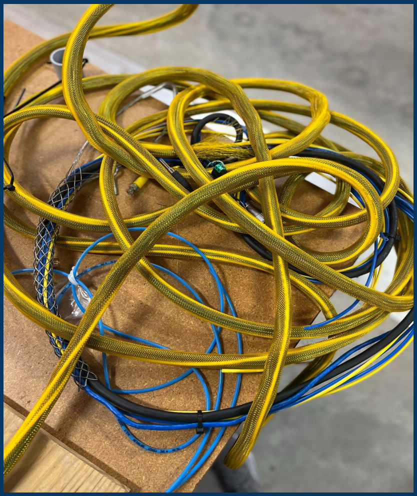
The wires from each driver channel to the motors are ran from the driver board outputs, through the milspec connectors, and then to the motors. The crawling motor is passed through the eleven pin connector used for power and data. The three vertical motors and two thrust motors are passed through a ten pin milspec connector.



Pictures on the left: four buck converters, one is for the arduino (12V>9V) the other three are for the H-Bridges (12V>7.5 - 12V)
Picture on the right: Shows the arduino board, the RS485, the custom controller board, and the three H-Bridges.

Tether

We have one power cable, one ethernet cable, and four pneumatic hoses running through our mesh sleeve. At the end of each part of the tether we have a strain relief to help prevent tugging on the cables and hoses.



On left and right: Main tether and connection to ROV

Software

Basic Controls

The right joystick controls the left and right thrust motors, which are indicated on the ROV by their blue shrouds. This joystick will allow both motors to travel at the same direction and speed, which in turn allows for smooth travel through the water. Moving this joystick left or right will modify allow the thrusters to operate at different speeds, which will cause the rover to turn in place or while moving forwards or backwards.

The left joystick controls the crawling motor and all three vertical motors. Up and down is vertical and left and right controls left and right crawling.

The panic button is a latching pushbutton and if activated, it will set all motor speeds to stop which will allow safe recovery of the rover. This also allows the team to perform safe in-water repairs, or remove foreign objects from being tangled or caught on the rover.

Frame

Our product's frame consists of a lightweight triple layered structure made up of PVC and connectors holding it all together. Every Side was measured and cut to a specific length to conform with the MATE ROV specifications. Our Frame also has holes put throughout specific locations along the PVC to allow water to travel through the frame and allow neutral buoyancy. In the middle of the frame we have two pieces of PVC laying horizontally so that our Motor Control Box can sit perfectly in the frame. For wire management we secured the loose wires to the sides of the frame.

Thrusters

We used six pump motors from the barracuda kits. Three of them are vertical for moving up and down. Currently they move at the same speed and direction. Then we have two thrust motors, one on the left and one on the right for forwards and backwards propulsion. The speed and direction of the thrust motors are independently variable allowing us to be able to turn it. Then we have one crawling motor underneath to be able to shift left or right. We 3D printed propellers and shrouds for our thrusters to protect the blades and the sea life. The shroud also help direct the flow of water for propulsion.

On the left we have all of our motors and how they are positioned.
On the right: we have a picture of just the motor itself.

Buoyancy & Ballast

To control our Buoyancy we are using pool noodles and Static Ballast in the form of added weights.

Payload and Tools

Manipulators

We 3D printed a vertical and horizontal grabber attached to our pneumatic cylinders. These are mounted on the front of our ROV.

Pneumatics

We decided to go with Pneumatic for the manipulators for a more reliable robust control mechanism that could easily stand up to water pressure and eliminate the chance of failure due to electrical shorts. So we first have air come in from a pressurized source into a quick connection that then goes into our pressure relief valve. Then it regulated to 40 PSI in our pressure regulator. The line then goes to a tee that supplies two manual DCV's. Each DCV has color coded lines that attach to the outputs allowing us to quickly identify which manipulator is attached. These lines are then run through the tether to their own respective cylinder on the manipulators. Forward closes each one and backwards opens each one.

On left:
3-D gripper
On right:
Pneumatic
connection

Cameras

We used the same camera system available with the barracuda system. This allowed standard connection from the existing backplane board. Outside the control box we used adapters to connect camera power and data through the ethernet cable in the tether that we used for controller/ ROV communication. We waterproofed it by backfilling an acrylic tube with epoxy and coating the wire/ epoxy junction with silicone.

On Left: Camera
that we used for
our ROV

BGC Float

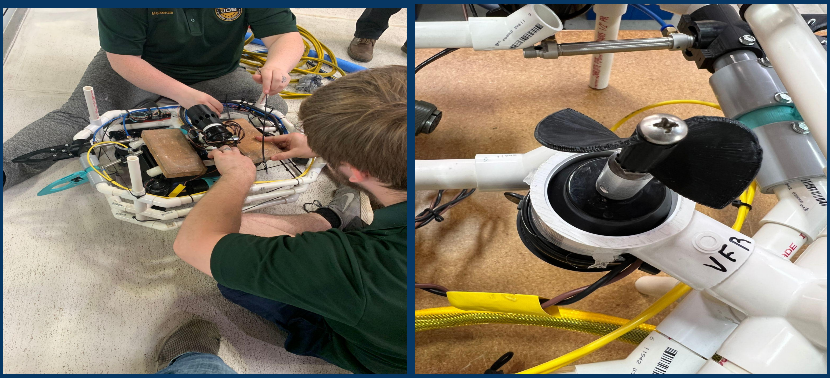
We decided to go with a simple design. We are using a 500cc syringe and a balloon attached to the syringe for the main buoyancy engine. To actuate it we are using a mechanical system consisting of 3-D printed latches and levers, rubber bands, and a mechanical kitchen timer. This will be enclosed in a 4 inch PVC with a sealable lid.

Initial setup will have the balloon full of water and the syringe depressed. The timer will trigger the first lever which will release the latches on the first stage pulling the syringe plunger to increase the density inside the float and causing it to sink. Once the timer hits the next stage it will move another lever releasing the latches on the second stage allowing the remaining rubber bands to depress the syringe plunger. This will force the water out into the balloon decreasing the density inside the float allowing it to float back up.

Critical Analysis

During our time testing the ROV, we ran into multiple roadblocks. As we put it in the water for the first time, we quickly realized that we did not have near enough weight to fully submerge. Knowing this the team slowly added weight, making sure to evenly distribute it across the ROV. We tried multiple strategies, including adding a substantial amount of weight with the use of pool weights, eventually even zip tying bricks to the bottom of it. After a period of time we finally got it to sink, but we still had issues with buoyancy. To correct this, we added small pieces of pool float around the sides, which eventually lead to us finally having a neutrally buoyant ROV.

As soon as we got the weight distribution and balance where we wanted it, we soon realized the majority of our props were not spinning at the rate we anticipated. After further inspection we quickly saw that our props were coming loose, which we fixed quickly and allowed us to move smoothly through the water. Even though we struggled at the beginning, the team was very pleased with our pneumatic manipulator system providing immediate success.



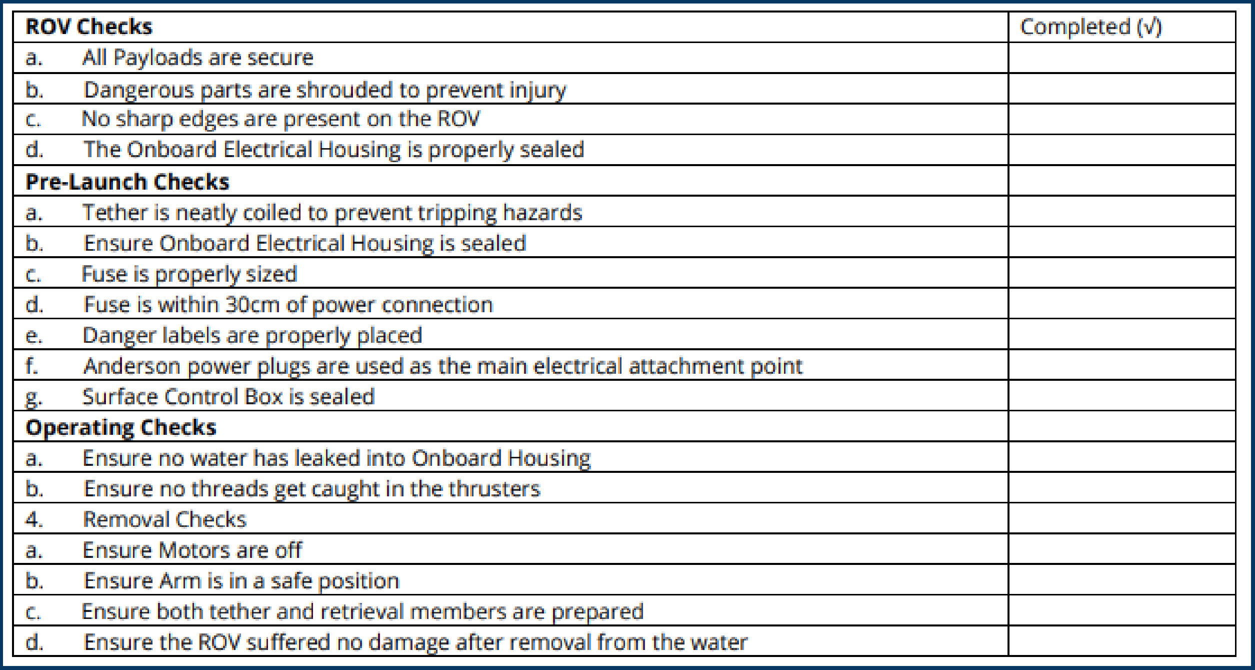
ON LEFT:
Showing the process of adding weight
ON RIGHT:
Showing one of the loose props

Project Development

Even though this is the team's first year competing in mate ROV, our expectations at Savannah Tech Robotics were still high. At first, all five team members would meet up at least twice a week to come up with new ideas and how to troubleshoot current problems. Each member was mostly responsible for his or her own part of the build but if they needed help or advice it was always given throughout the team. As we drew closer to our regional competition we began meeting much more frequently and working long nights, which allowed us to get the ROV done in a timely manner.

Company Safety Review

Safety was imperative during the entire process of developing, manufacturing, and testing our ROV. We made sure at the very beginning that everyone involved was prepared and aware of any hazards that may occur during the evolution of our product. One example would be during the use of a bandsaw in the original construction of the ROV, everyone in the area wore proper PPE glasses and used caution while the bandsaw was being operated. Also to ensure the protection of the underwater environment that our product was going to be in we added custom 3-D printed shrouds to our thrusters, allowing everything in their vicinity easy protection from the props in use. Along with this we made sure to use rounded edges on all possible aspects of the ROV, which allowed us to once again reduce risk of harm to any of our teammates or underwater wildlife. Lastly we ensured that all of our electrical and pneumatic components were tightly sealed and were closely examined, which allowed us to easily avoid the risk of electrical hazard or one of our pneumatic lines coming out of place.



ROV Checks	Completed (√)
a. All Payloads are secure	
b. Dangerous parts are shrouded to prevent injury	
c. No sharp edges are present on the ROV	
d. The Onboard Electrical Housing is properly sealed	
Pre-Launch Checks	
a. Tether is neatly coiled to prevent tripping hazards	
b. Ensure Onboard Electrical Housing is sealed	
c. Fuse is properly sized	
d. Fuse is within 30cm of power connection	
e. Danger labels are properly placed	
f. Anderson power plugs are used as the main electrical attachment point	
g. Surface Control Box is sealed	
Operating Checks	
a. Ensure no water has leaked into Onboard Housing	
b. Ensure no threads get caught in the thrusters	
4. Removal Checks	
a. Ensure Motors are off	
b. Ensure Arm is in a safe position	
c. Ensure both tether and retrieval members are prepared	
d. Ensure the ROV suffered no damage after removal from the water	

Accounting

Quantity	Description	Total Price
3	L298 Dual H-Bridge Motor Speed Controller	$44.97
1	Sheffield 12643 .50 Cal Ammo Can	$26.47
1	Milton s-760 1/4" MNPT Quick Connect	$3.02
2	Tailonz Pneumatic Rotary Lever Hand Valve	$37.98
1	Dupont Connector Kit	$12.99
1	JST Connector Kit	$12.88
1	4mm Push Connectors Pneumatic	$12.99
1	M5 to 4mm push Connector Pneumatic	$7.99
1	Tailonz Pneumatic Pressure Regulator	$13.99
2	16p PDIP Sockets	$21.60
1	Epoxy	$23.06
1	Barrel Pin to 2Wire adapters	$7.99
1	RCA to 2wire adapters	$6.66
1	gorrilla 2 part Epoxy	$12.94
4	Buck Converters	$8.99
1	1x40 pin Headers	$7.98
2	RS485 Tranceiver Modules	$19.98
2	Mega 2560 R3	$41.98
1	5 oz IPS Weld-on #16	$9.99
1	IPS weld-on Micro tips	$3.00
1	blue 4mm OD Pneumatic Tube	$13.50
1	Yellow 4mm OD Pneumatic Tube	$13.50
1	55ft SJEOOW 12/2 Portable Cord	$70.95
1	Barracuda ROV with Thrusters and Tether (Rev 2)	$1,260.00
2	10ft 1/2 inch PVC	$7.96
10	1/4-in dia 90-Degree Tee PVC Fitting	$37.80
10	1-in dia 90-Degree Cross Tee PVC Fitting	$54.80
25	1-1/2-in dia 45-Degree Slip Elbow PVC Fitting	$74.00
1	Nitra Pressure Relief Valve	$26.50
	Total Cost	$1,896.46

Budgeting

Savannah Tech Robotics (STR) didn’t operate with a very big budget. We mostly relied on parts we had around the school. Any part that we needed we did intense research to find the cheapest price. A lot of the money we got for parts came from donations and grants from the school. As for travel and lodging, that was covered by the school as well. Also if it wasn’t something we absolutely needed we either didn’t use it or we would take a cheaper alternative route.

Build Vs. Buy

Because STR did not have a big budget we mostly relied on 3D printing parts like our manipulators and our fan shrouds. We also borrowed our pneumatic cylinders from our PLC program to help operate our manipulators. We also found some arduino boards that we knew we could use for our build. In total we spent less than $2,000 for our ROV build. Something that significantly helped our budget was that we were given the barracuda kit, which would have been worth more than our entire budget.

Refrencenes

2022 mate Rov Competition. (n.d.). Retrieved May 24, 2022, from https://files.materovcompetition.org/2022/2022_PIONEER_Manual_21_JAN_2022.pdf

Blum, J. (2020). *Exploring arduino: Tools and techniques for engineering wizardry*. Wiley.

Porter, B. (2016, November 30). *Arduino-EasyTransfer/EasyTransfer at master · MADSCI1016/Arduino-EasyTransfer*. GitHub. Retrieved May 24, 2022, from https://github.com/madsci1016/Arduino-EasyTransfer/tree/master/EasyTransfer

Arduino Mega 2560 REV3. Arduino Official Store. (n.d.). Retrieved May 24, 2022, from http://store.arduino.cc/products/arduino-mega-2560-rev3

Cartridge Pump Replacement Motors. SPX FLOW. (n.d.). Retrieved May 24, 2022, from https://www.spxflow.com/johnson-pump-marine/products/cartridge-pump-replacement-motors/

Arduino Mega 2560 datasheet - robotshop. (n.d.). Retrieved May 24, 2022, from https://www.robotshop.com/media/files/pdf/arduinomega2560datasheet.pdf

Acknowledgment

Savannah Technical Robotics would like to thank all of the organizations that have supported us and have helped us get this far in the MATE ROV Competition.

MATE ROV Competition judges, volunteers, and support staff MATE Inspiration for Innovation (MATE II)

MATE 2022 World Championship Sponsors

Long Beach City College

Jody Patterson and the Chatham Aquatic Center

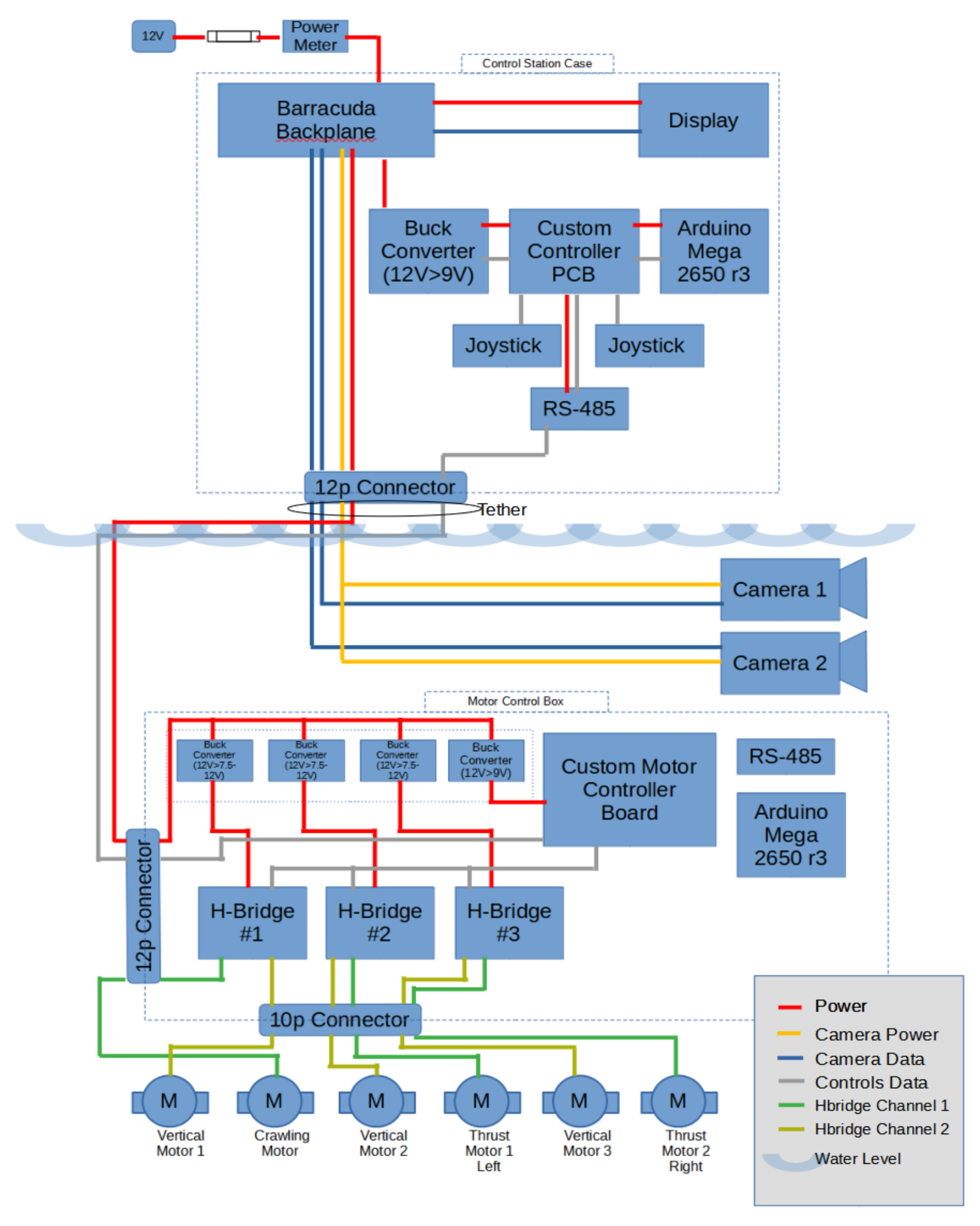
And our local support system including:

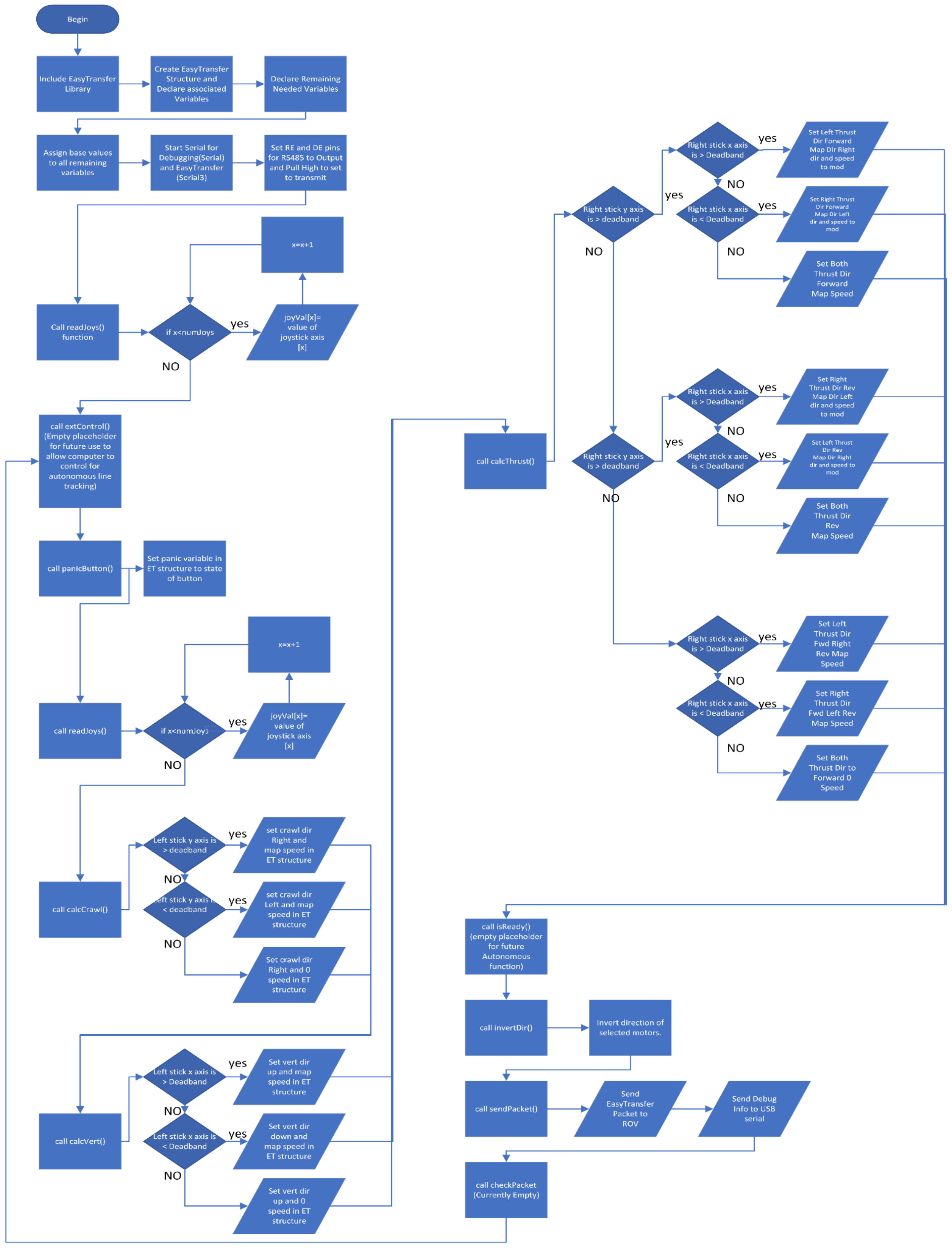
Savannah Technical College

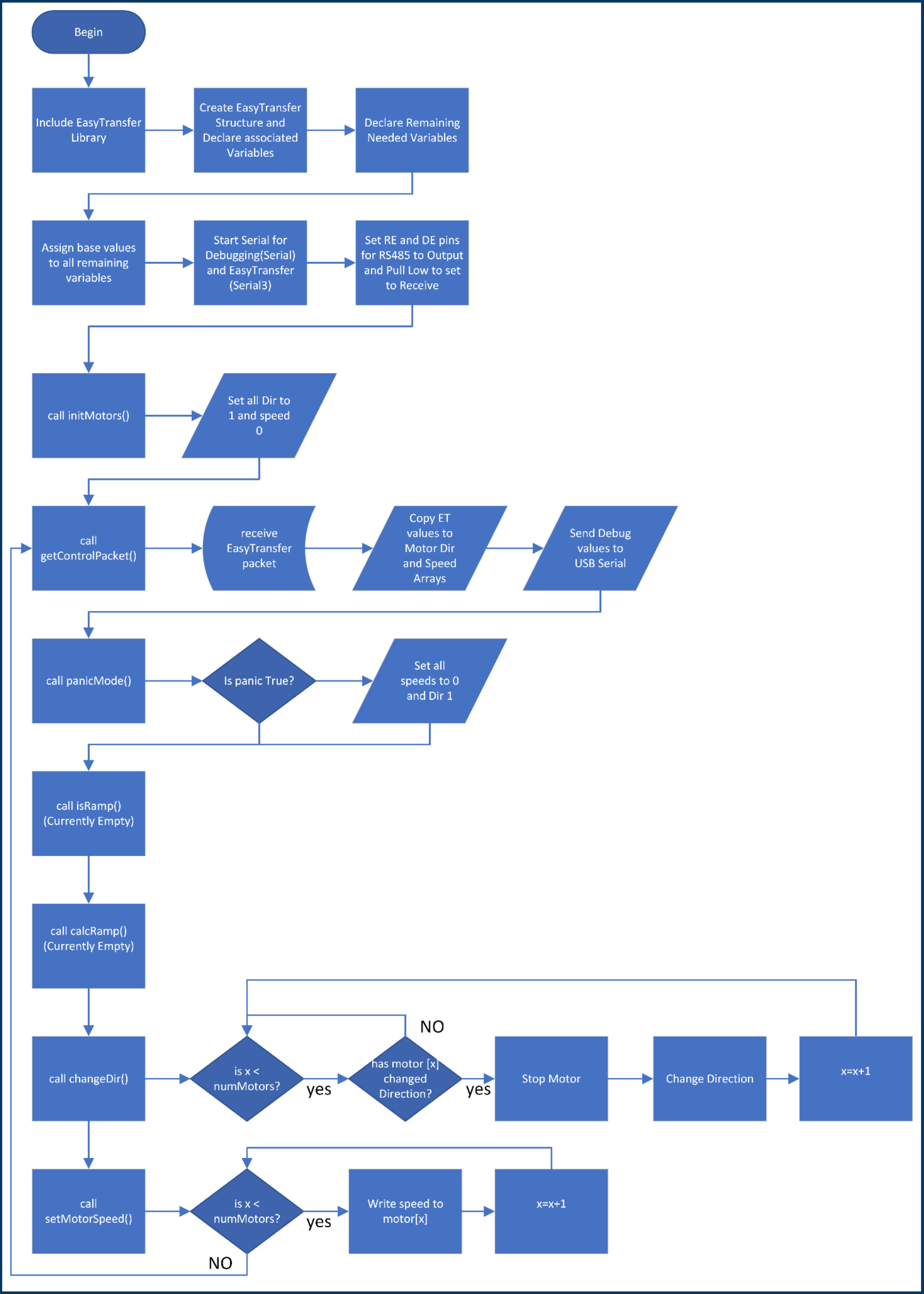
JCB

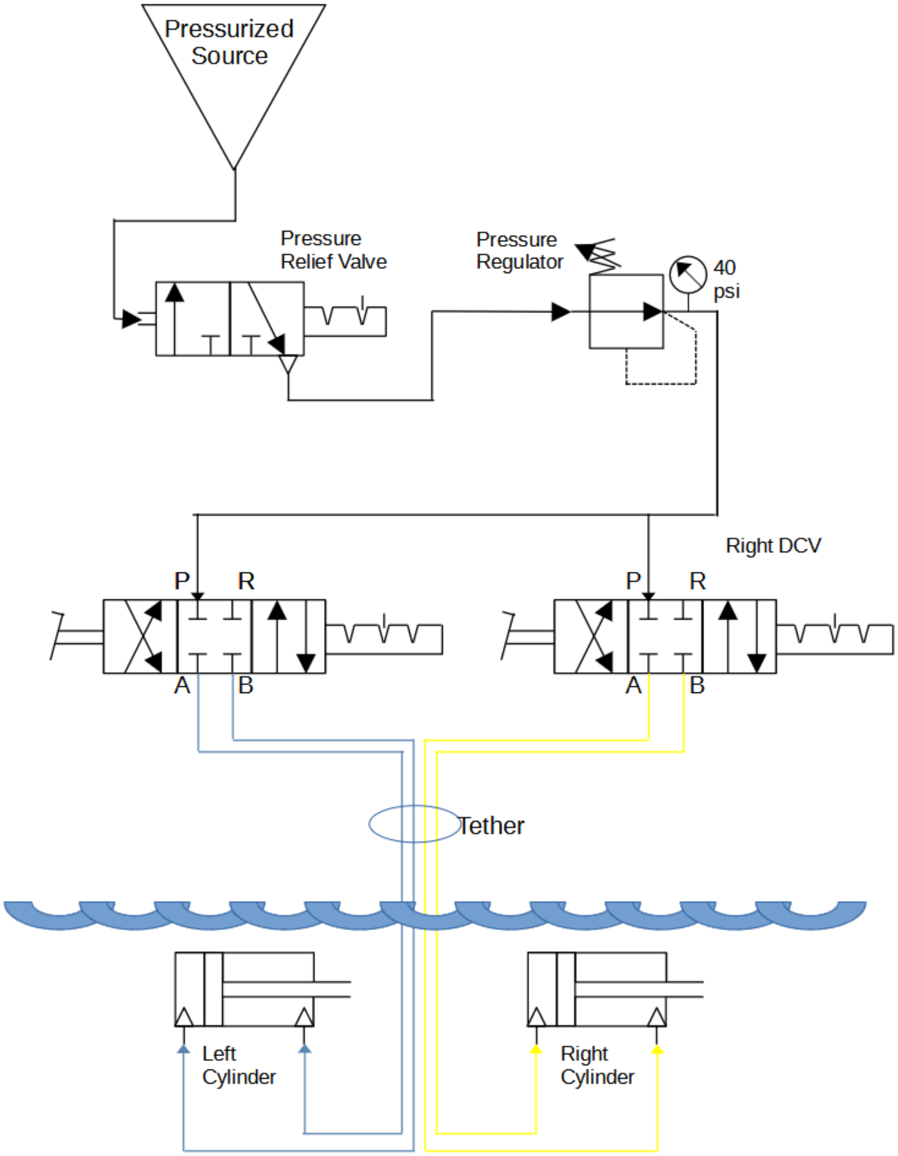
Our mentor David Ericsson and our parents.

SID’s/ Flowcharts




Begin
Include EasyTransfer Library
Create EasyTransfer Structure and Declare associated Variables
Declare Remaining Needed Variables
Assign base values to all remaining variables
Start Serial for Debugging(Serial) and EasyTransfer (Serial3)
Set RE and DE pins for RS485 to Output and Pull High to set to transmit
x=x+1
Call readJoys() function
if x<numJoys
yes
joyVal[x]= value of joystick axis [x]
NO
call extControl() (Empty placeholder for future use to allow computer to control for autonomous line tracking)
call panicButton()
Set panic variable in ET structure to state of button
x=x+1
call readJoys()
if x<numJoys
yes
joyVal[x]= value of joystick axis [x]
NO
call calcCrawl()
Left stick y axis is > deadband
yes
set crawl dir Right and map speed in ET structure
NO
Left stick y axis is < deadband
yes
set crawl dir Left and map speed in ET structure
NO
Set crawl dir Right and 0 speed in ET structure
call calcVert()
Left stick x axis is > Deadband
yes
Set vert dir up and map speed in ET structure
NO
Left stick x axis is < Deadband
yes
Set vert dir down and map speed in ET structure
NO
Set vert dir up and 0 speed in ET structure
call calcThrust()
Right stick y axis is > deadband
yes
Right stick x axis is > Deadband
yes
Set Left Thrust Dir Forward Map Dir Right dir and speed to mod
NO
Right stick x axis is < Deadband
yes
Set Right Thrust Dir Forward Map Dir Left dir and speed to mod
NO
Set Both Thrust Dir Forward Map Speed
NO
Right stick y axis is > deadband
yes
Right stick x axis is > Deadband
yes
Set Right Thrust Dir Rev Map Dir Left dir and speed to mod
NO
Right stick x axis is < Deadband
yes
Set Left Thrust Dir Rev Map Dir Right dir and speed to mod
NO
Set Both Thrust Dir Rev Map Speed
NO
Right stick x axis is > Deadband
yes
Set Left Thrust Dir Fwd Right Rev Map Speed
NO
Right stick x axis is < Deadband
yes
Set Right Thrust Dir Fwd Left Rev Map Speed
NO
Set Both Thrust Dir to Forward 0 Speed
call isReady() (empty placeholder for future Autonomous function)
call invertDir()
Invert direction of selected motors.
call sendPacket()
Send EasyTransfer Packet to ROV
Send Debug Info to USB serial
call checkPacket (Currently Empty)


Begin
Include EasyTransfer Library
Create EasyTransfer Structure and Declare associated Variables
Declare Remaining Needed Variables
Assign base values to all remaining variables
Start Serial for Debugging(Serial) and EasyTransfer (Serial3)
Set RE and DE pins for RS485 to Output and Pull Low to set to Receive
call initMotors()
Set all Dir to 1 and speed 0
call getControlPacket()
receive EasyTransfer packet
Copy ET values to Motor Dir and Speed Arrays
Send Debug values to USB Serial
call panicMode()
Is panic True?
Set all speeds to 0 and Dir 1
call isRamp() (Currently Empty)
call calcRamp() (Currently Empty)
call changeDir()
is x < numMotors?
yes
has motor [x] changed Direction?
NO
yes
Stop Motor
Change Direction
x=x+1
call setMotorSpeed()
is x < numMotors?
yes
NO
Write speed to motor[x]
x=x+1


Pressurized Source
Pressure Relief Valve
Pressure Regulator
40 psi
Right DCV
P
R
A
B
P
R
A
B
Tether
Left Cylinder
Right Cylinder