NAUTILUS

the rov of the future

California Academy of
Mathematics and Science
CARSON, CA
MATE 2019

TEAM MEMBERS

Jiajer Ho ’19 - CEO & Electrical Lead
Frank Chaqueco ’20 - Design Lead & Electrical
Isaac Addis ‘19 - Software Lead
Alexander Vazquez ‘21 - Software Engineer
Matias Vergara ‘22 - Mechanical Engineer
Jared Arbolario ’22 - Mechanical Engineer
Aarya Topiwala ’22 - Electrical Engineer
Kyle Garcia Takata ‘21 - Finances & Publications

Sandra James - Mentor

ABSTRACT

The latest addition to the 45C Robotics family, the Nautilus, is an aquatic Remotely Operated Vehicle (ROV), that specializes in locating, mapping, and manipulating different objects in scenarios such as repairing dams, observing underwater habitats, and locating historical artifacts. 45C Robotics is composed of an 8-person team, that has the dedication and expertise to make an ROV that is up for the task. 45C is separated into teams that specialize in a certain aspect of the ROV. These subteams frequently communicate with each other to ensure that the ROV system works smoothly.

The Nautilus features a custom frame manufactured with CNC mills, and 3D printers. It is constructed with precision instruments to ensure a high-quality product. Nautilus also showcases several custom Printed Circuit Boards (PCBs) for an advanced and reliable flight computer. Our PCBs provide the robot with multiple advanced sensors that allow the computer to constantly monitor the electronic housing.

To enable advanced maneuverability, Nautilus features six custom-designed thrusters providing the robot with 5 axes of motion. In addition to the wide range of motion, PID (Proportional-Derivative-Integral) was added to allow the pilot to control the robot normally even when weight distributions are not ideal.

These new features on the Nautilus make it the most advanced ROV ever created by 45C Robotics. Our team has continued push through difficulties with a positive team culture for nine months to create an ROV that exceeds expectations set by MATE.

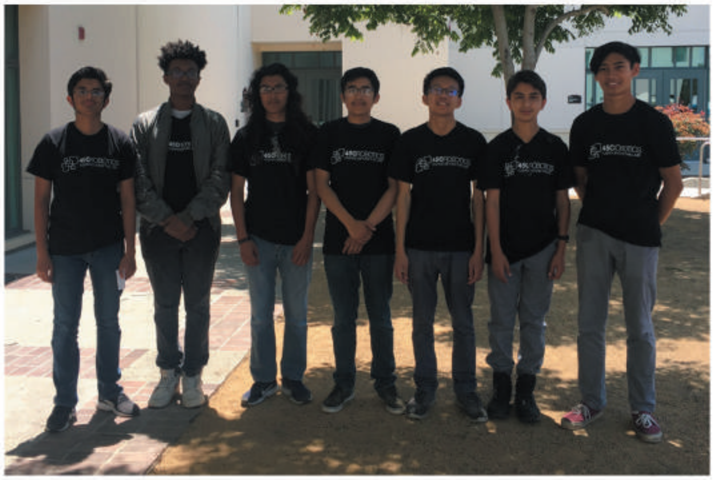


Figure 1A: Team Photo

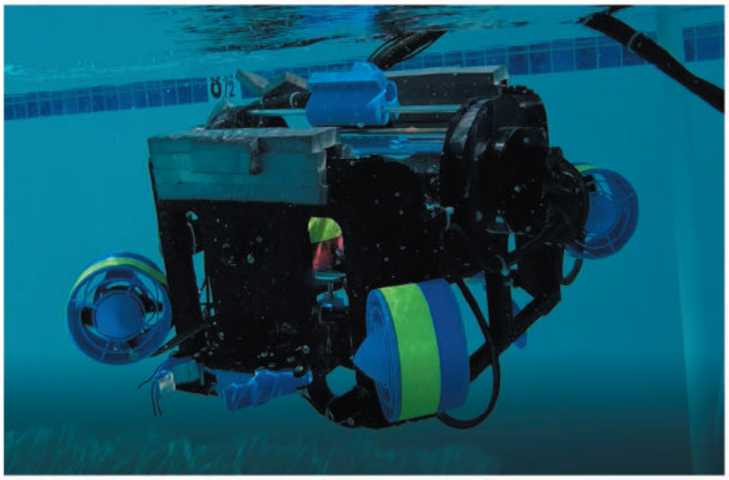


Figure 1B: Nautilus' Water Test

TABLE OF CONTENTS

DESIGN RATIONALE

DESIGN PHILOSOPHY

At 45C Robotics, our design philosophy is to push the limits of conventional engineering. We take risks and are not afraid of failure. During development, the team utilized a step-by-step design process for maximum efficiency. Members of 45C Robotics began this process by brainstorming ideas, with no limitations. These ideas were compared through a decision matrix and evaluated using the following criteria: cost, size, weight, functionality, and simplicity.

MECHANICAL

Figure 2: Unassembled Frame

FRAME

The frame is made from a CNC machined marine grade HDPP and seaboard HDPE. The basic structure of the frame is to support our new vector motor placements but also different kinds of manipulators that we have designed. The 45° angle on all four corners of the frame were intentionally designed to accommodate our custom motor mounts, and the whole frame is structurally supported with 175° brackets on each corner. Aluminum rods were also used to further strengthen the frame, while serving as a way to mount our cameras in convenient locations.

MOTOR MOUNTS

Nautilus' motors are guarded by two designs of 92 mm and 91.4 mm diameters in order to best match with the motor housing. One of the designs consists of a concentric cone with the outer circle. This cone allows for higher hydraulic efficiency when the vector motors are on. The other design mimics that of which it is attached to in order to have structural integrity and efficient hydrodynamics. During the design process of the motor guards, the IP20 requirements were considered in order to insure the safety of MATE divers. Incidentally, the Nautilus' motor guards were manufactured using 3D printing with PLA (Polylactic Acid).

Figure 3: IP20 Motor Mounts

MOTOR PLACEMENTS

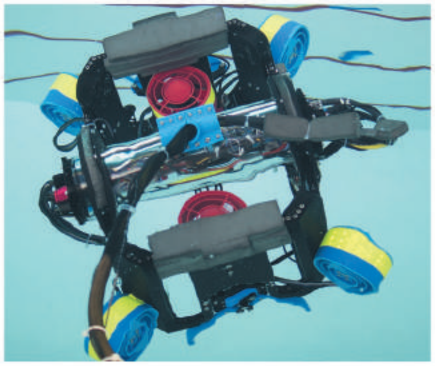


Figure 4: Nautilus Underwater

The placement of our Brushless motors (BLDC) enables the Nautilus to maneuver with endless possibilities. This is due to the fact that the motors are placed at the four corners of the Nautilus at 45 degree angles as shown in the diagram adjacent. The equality of the vectors created by the 45 degree placement enables the Nautilus to be stable while underwater and allows multiple axes of motion. Our robot can pivot left and right, forwards and back, and turn left and right. Our Up/Down motors are placed to enable pitch maneuverability.

MANIPULATORS

The rotable front claw of the ROV is is held using a bearing to reduce stress on the servo that does the turning. This same servo is mounted vertically on a 3D printed plate to reduce obstruction from the thrust produced by the motor above it. This whole mechanism allows us to pour grout and release trout fry into habitats similar to task 1 & 2 with the use of custom 3D printed cups.

Our back claw is elongated and thicker than last year's claw to be able to grip objects up to 16 centimeters in diameter. These features allow our claw to be extremely versatile for tasks such as recovering water samples or lifting heavy debris.

All the tools use our custom waterproofed servo motor which consists of pouring epoxy into the servo motor controller as well as using epoxy for the wires leaving the servo, similar to a cable penetrator.

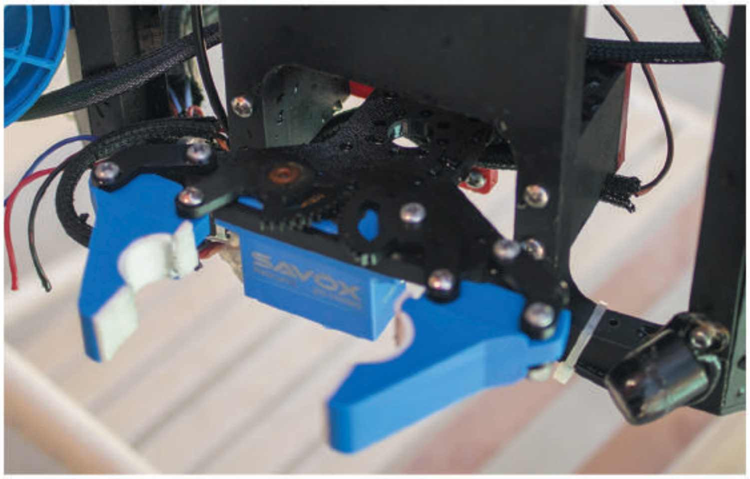


Figure 5A: Front Claw with rotatable design

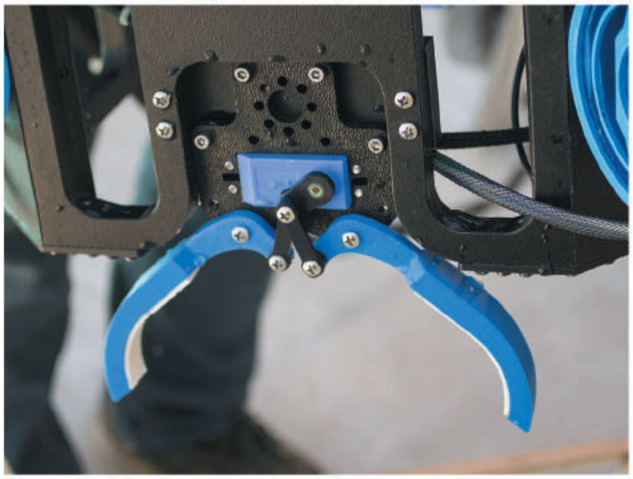


Figure 5B: Back Claw's Location

ELECTRICAL

TOP SIDE COMMUNICATIONS

To communicate from and to the control box, a custom PCB was created to allow the transmission and receiving of RS485 signals. (The PCB was designed using EasyEDA, a free online software for schematic and PCB design.) The board utilizes an ATMEGA2560 to manage all the communication and conversion processes. It does this by first receiving data from the sensor serial sender and converting it into TTL (Transistor–transistor logic) for the on-board computer in the control box via FTDI (Future Technology Devices International). It also converts all joystick and button inputs into digital signals to later then be converted into RS485 signals to be transmitted to the ROV for a full duplex communication system.

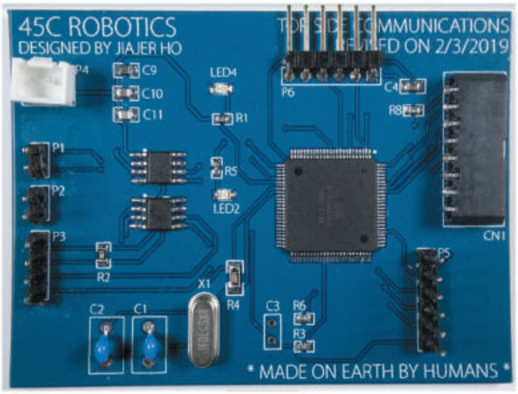


Figure 6A: Top Side Comm PCB

Figure 6B: Top Side Control Box

ELECTRONIC HOUSING

The Nautilus' on board electronics system consists of four different custom PCBs, an isolated DC-DC power supply, a step-down buck converter, and a custom made ESC panel. The housing itself is 76mm inner diameter with a length of 300mm. The flanges for the housing are designed to allow wires to enter and exit from both end. Similar to last year's design, Nautilus has the left end cap with all motor wires while the right one handles the tether and the connector for the micro-rov. In order to allow easy repairability of the onboard electronics, the motor wires are connected with Anderson power pole connectors that are color coded to reduce error when connecting. The decision to make custom circuit boards stemmed from the need for producing an electronic system that was reliable and in the smallest form factor. The PCBs were designed using EasyEDA, a free online software for schematic design and PCB routing. The five boards in the housing are named appropriately: ESC Vector Controller, Sensor Serial Sender, RS485 and Servo Controller, Voltage Spike Protection.

ESC VECTOR CONTROLLER BOARD

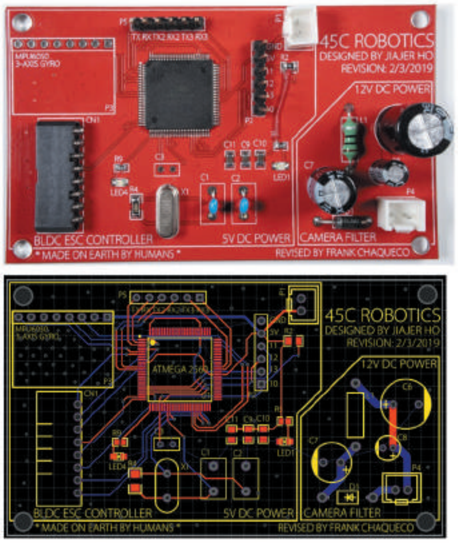


Figure 7A/B: ESC Controller PCB

The BLDC motor controller PCB utilizes an ATMEGA2560 microcontroller that runs on the open source Arduino IDE software. A 3-Axis MPU6050 gyroscope is implemented on the board for PID control. An ATMEGA2560 was chosen because it has multiple RX and TX I/O pins which are needed for the board to communicate with other boards. An 8 pin connector is used to connect the pins of the chip with the corresponding ESC on the back side of the board. The ESC's are sandwiched together with a custom cut aluminum plate. This allows reduction in space and also spreads out the heat generated from the ESCs.

The second part of this board consist of an LC Pi filter that reduces the power of the signal going into the camera. The LC Pi filter consists of a carefully chosen inductor, which reduces the sudden changes in current, and a capacitor, which reduces the sudden changes in voltage. However, because the capacitor draws a lot of power form the input and the input may not be able to handle the sudden voltage intake, a second capacitor was added to the input as a reservoir where energy is stored when a sudden voltage intake is experienced.

SENSOR SERIAL SENDER

The sensor PCB utilizes an ATMEGA328P micro-controller that runs on the Arduino IDE software. A DHT11 Temperature/Humidity sensor is added to ensure that the housing is kept at a safe temperature and condensation level. Moreover, a temperature probe is used for determining the temperature outside of the ROV. The probe connects to a 2 pin connector that goes through a resistor and then to an analog pin on the ATMEGA328P. The microcontroller converts the voltage drop within the thermistor to temperature values.

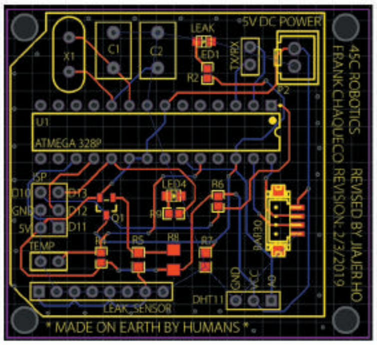


Figure 8: Serial Sensor Board

For our leak detection system, a transistor detects a small voltage difference from two probes which become connected when the sponge between it is wet. It then notifies the microcontroller of a potential leak. In addition, the board uses a BAR30 pressure sensor to determine the depth of the ROV. All these signals are sent to the communication board via TTL in order for the surface computers to monitor the sensors and provide the pilot with sensor data.

RS485 and Servo Controller

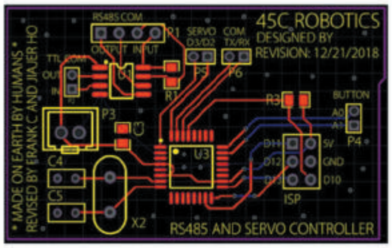


Figure 9: Servo Controller PCB

The RS485 and servo controller PCB consists of the SMD version of the ATMEGA328P and the MAX488CSA+ microcontroller. The ATMEGA328P takes incoming communication data from the tether and moves the servo accordingly via pulse width modulation (PWM). This allows the pilot to control the direction of the claw precisely using a potentiometer. The board also converts the RS485 signals from the tether to TTL and vise versa.

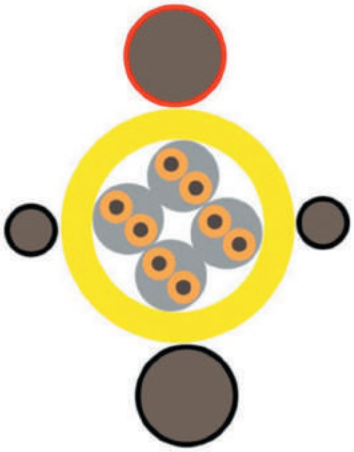


Figure 10: Tether Diagram

TETHER

The tether system consists of two 10 AWG wires for 14.5 volts and ground, eight 26 AWG wires for communication with coiled pairs, and two coaxial cables for the cameras. Four of the communication wires are used for full duplex RS485 communication and the rest are used for controlling the servo motor from a potentiometer. The team had originally tried using TTL signals for communication but that was unable to work in the noisy environment and the long tether distance. To solve this problem, coaxial cables were chosen for video transmission since they were well shielded to prevent ghosting effects and video flickering due to noise.

PROGRAMMING

INTER-BOARD COMMUNICATIONS

From the Control Box, the Top Side Communication (TSC) board sends data to the ESC control custom board through Serial communication at a baud rate of 57600 bits per second. In addition, the Top Side board receives temperature and gyro data from the sensor board. Since the data needed to be read as a whole on the receiving end, the data was transferred as a character array through the use of arduino's function Serial.write. However, in order to send and receive data as a character array all of our data needed to be converted into one single string. After being converted into one single string, the data was then converted to a character array. Both the TSC board and ESC control custom board needed a way to receive and interpret the data being received. The serial in both boards read the bytes of the character array with the use of the Serial.readBytes function.

Vector Control

Originally, the two axis analog joysticks that were used to control the six motors had raw potentiometer values from 0 to 1023 for both axes. To decrease the sensitivity of the joysticks, the raw potentiometer values were mapped in a different range between -25 to 25. Raw potentiometer values for each axis were retrieved using Arduino's analogRead() function `and the map() function was used to map the values. There were a total of five joystick values from -25 to 25, since there were five types of movements (Forward/Backward, Lateral Right/Left, Turning, Vertical movement, and Tilting). To have a 'smooth' transition between movements the following setup was used as shown in the diagram below. Blue and red represent two different types of movements. There was one joystick that only had Vertical movement, where above the blues dashed line was down and above the blue dashed line was up.

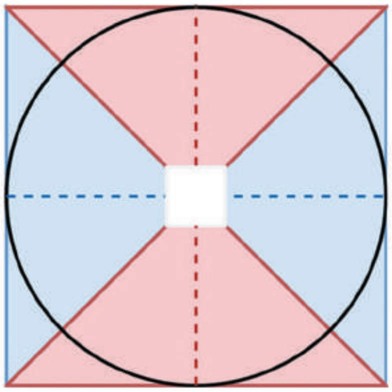


Figure 11: Joystick Pot

NOTE: The analog Joystick had two axes, which had values that could be represented within a square

After the joystick values were sent via serial communications as a character array, the ESC control board received the data. The whole character array was converted into a string and multiple substrings were retrieved using arduino's .substring() function. Each substring represented the mapped values of -25 to 25 from the joysting. These substrings were then converted into integers using the .toInt() function.The integer value was then mapped to values from 1000 to 1950 whenever only two motors were in use for vertical movement or tilting, 1150 to 1850 when four motors were in use, and 1300 to 1700 when six motors in use. The mapped values were then used as an argument for arduino's .writeMicroseconds() function to control the shaft of each of the motor's shaft, where 1500 stopped the motor.

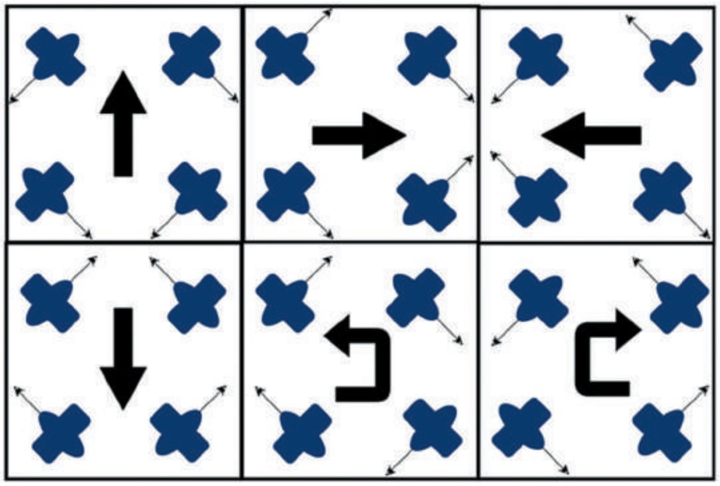


Figure 12: All possible Vector Movements

Video Input and Pre-Processing

Camera input is captured and processed using OpenCV Version 4.0 for Python. Thorough pre-processing of the images received from the camera at each iteration is imperative to ensure minimal noise and remove extraneous image features that may inhibit the performance of the Benthic Species algorithm.

The first three operations in the input image are Gaussian blurring, median filtering, and bilateral filtering, each serving to normalize the pixels within the frame for accurate contour (closed-edge shape) detection.

Initially applying Gaussian blurring reduces noise and detail. A kernel, a matrix of numbers adding together to 1 to prevent energy leaving or entering the image, is convolved to the input image to make the pixels follow a Gaussian, bell-curve distribution where the strongest pixel value appears at the center. Median filtering, on the other hand, iterates through each pixel, making its pixel intensity the median of its neighboring pixels.

Bilateral filtering aims to solve the strongest undesirable effect of the aforementioned processing algorithms: edge blurring. Contour detection relies heavily on noiseless, detailed-edged contours, and bilateral filtering replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels

.

Figure 13: (From top to bottom, left to right) Raw image, Gaussian Blurring, Median Filtering, and Bilateral Filtering applied sequentially on the frame. Each filter applied gradually blurs the image, carefully retaining important feature information such as vertical and horizontal edges, but removing unimportant intricacies that can lower the performance of contour detection.

Benthic Species

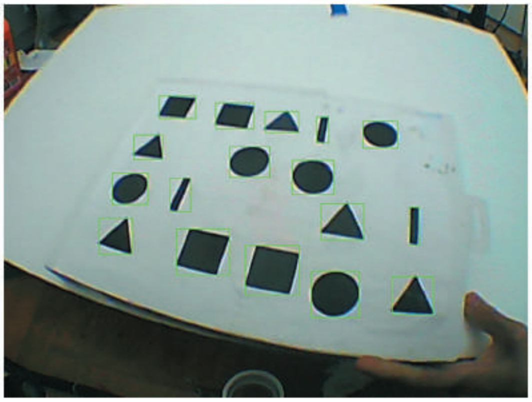


Figure 14: OpenCV bounding-box outlines

The shapes for the four benthic species are first cropped from the whole image before geometrically determining the type of the species. Initially, 45C Robotics used deep learning with a Convolutional Neural Network to pass the cropped contours into a trained model to determine the type of shape, but the model proved to be largely inaccurate when working in environments with poor lighting and angles.

45C Robotics found the best accuracy while using geometric properties of each contour to determine the type of shape. Using the cv2.arcLength() and cv2.approxPolyDP() functions, the vision program can outline the perimeter of each shape. The contours with three sides will always be triangles and the ratio between the width and height of the remainder of the contours can determine whether each is a square, line, or circle.

To prevent false positives -- the vision program identifying non-Benthic species as such -- the program takes an extra step to ensure that the average color within the contour's region in the image is black -- the color of the Benthic species.

Graphical Organization

The cameras on Nautilus are all interfaced through a central Graphical User Interface (GUI) built with the PyQt4 module in Python. Because PyQT4 cannot natively display Numpy arrays (the format for images used in OpenCV), images are casted from arrays into QImages, the format used for displaying frames using PyQT4.

Each video stream runs in a separate Python thread to avoid latency as a result of procedurally executing the tasks required for the vision and display programming. Each thread runs parallel to each other, allowing for the running of non-interfering tasks and greater speed and fluidity when using the GUI program.

PyQT4 provides an arsenal of tools and support for displaying a wide variety of information, which is why 45C Robotics chose to use the library instead of other popular frameworks in Python. In addition to providing native support for threading with its module QThread, PyQT4 offers tools for positioning, placing, and altering text, image, and window data.

Length of Dam Calculations

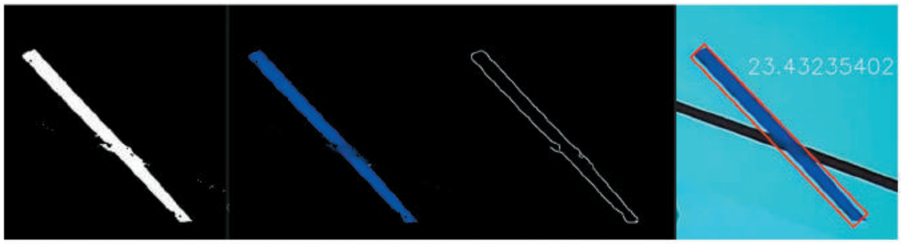


Figure 15: Computer Vision Software on Simulated Dam Crack

As camera input is captured and processed, each frame is converted into an HSV image. A mask is created to only detect blue objects using HSV values and the HSV frame.

With the mask, a new image is created that shows only the blue objects. Canny Edge detection is then used to detect the contours from blue objects. A bounding rectangle is created to surround the detect contours of the object.

From the bounding rectangle, the width and height of the cracks in pixels are extracted. The ratio between the width of the tape in centimeter to pixel length was used as a conversion factor. The conversion factor is used to calculate how many centimeters each pixel contains. The conversion factor is also used to calculate the length of the crack.

MICRO ROV

The MicroROV uses three DST-700s to move up/down, forward/backwards, and turn left and right. These basic movements are enough to survey a pipe with a camera. The MicroROV will be mounted onto the Nautilus using the Beluig waterproof connector. The tether for the MicroROV will be utilizing RS485 for noise immunity. Fiber Optics will not be used due to time constraints that were present in the making of the MicroROV. The frame of the MicroROV is minimalistic, because it only has to accommodate a camera. The PCBs controlling the MicroROV will be housed on the Nautilus' Control Box, further reducing the size of the MicroROV.

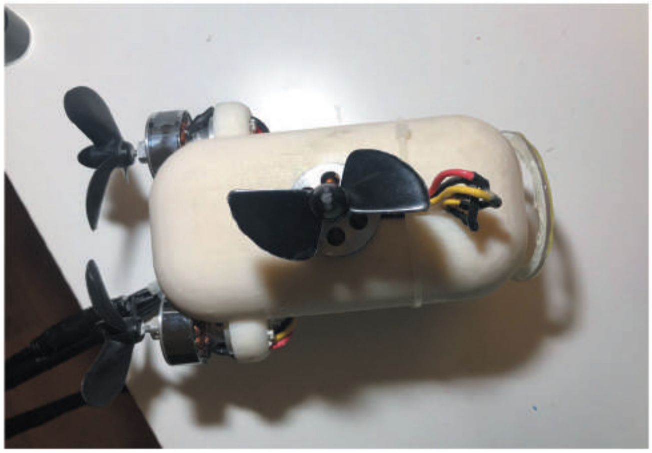


Figure 16A: Micro ROV Design Prototype

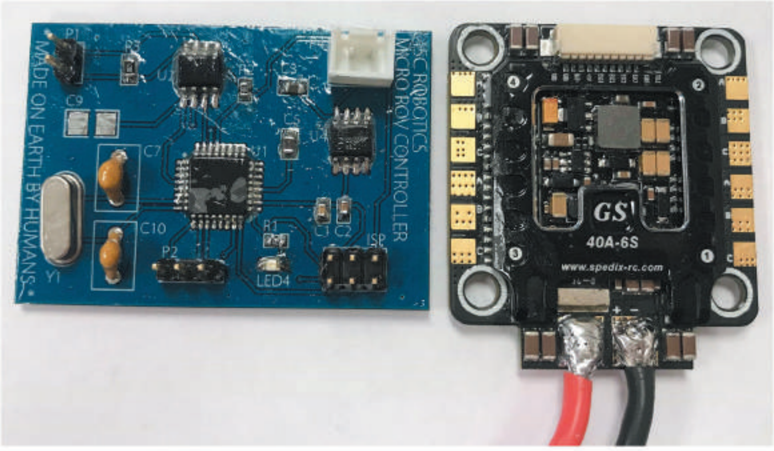


Figure 16B: Custom PCB and ESC for Micro ROV

TESTING PROCEDURES

To ensure that every element of the Nautilus functioned to the best of its ability, we conducted several tests on the robot. The individual components on the robot were tested, as well as the robot as a whole.

Housing

We tested our watertight housing using a vacuum pump and concluded that the enclosure was able to sustain 635 mmHg (approximately 12.28 psi) for at least 10 minutes. This test was performed to ensure that all seals on the housing were watertight. The aforementioned vacuum test was performed anytime the watertight enclosure was opened or closed, and before any test dive in the water.

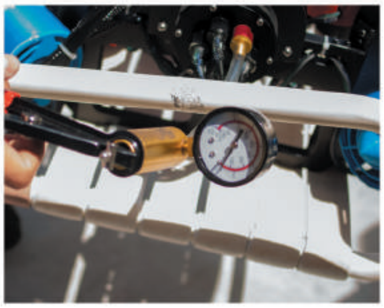


Figure 17A: Pressure Testing

Servo Motor

Like our watertight housing, we test our servo using a vacuum pump system. A broken servo was used to develop a testing rig for the O-Rings and gaskets on the servo motor. All servos on the Nautilus passed the vacuum test at 700mmHg (approx. 9 meters water depth) for 3 hours.

Figure 17B: Vacuum on Servo

Claw

The claw was tested by using a scale to determine the torque and grip of the claw. From this, we were also able to determine the strength of the claw to be 3kg. This is more than enough to pick up all the tasks for the competition.

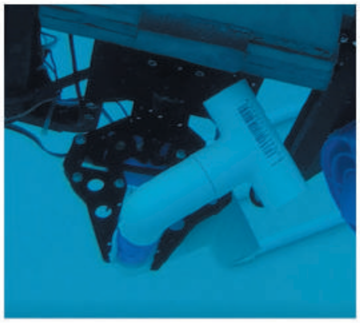


Figure 17C: Grip Test

Thrusters

Using a spring scale, we created a motor testing rig and tested the thrust of our six custom made thrusters. We did this in order to ensure the actual thrust of our motors, and whether it had enough thrust to maneuver the robot. The thrust of the up/down motors run together produce around 1.645 Kgf.

Figure 17D: Scale Test

SAFETY

ESD SAFETY

For this year’s robot we paid close attention to the consequences of electrostatic discharge on our electronics. Electrostatic discharge would prove deadly to our sensitive microcontrollers that play a crucial role in the manipulation of our robot. To definitively deal with this hazard, we designated ESD safe zones which required all electronics to be worked on an ESD safe mat, and any worker to wear an ESD wrist strap.

Figure 18: ESD Safe Area

SAFETY PROTOCOL

Nautilus was developed with an emphasis on safety. “Dry runs” were held periodically to ensure the mechanical and electronic components were working properly before testing the robot underwater. This year, we have integrated the safety checklist into the control box, so that the ROV will not start until all items have been checked. By doing so, any accidental usage of the ROV underwater while malfunctioning is greatly minimized. Conformal coating was also applied to our electronics in the case that water happened to enter the housing. Conformal coating would prevent any water from contacting our electronics since it creates a protective thin polymeric film which conforms to the contours of a printed circuit board to protect the board's components.

SAFETY CHECKLIST

1) Pre-Power

Area is clear of tripping hazards
Tether is layed out on deck
Electronics housing is sealed
All cable penetrators are not loose
Vacuum test electronics housing
Check vacuum port is secured

2) Vacuum Test Procedures

Connect vacuum hand pump to ROV
Pump to 350mmHg
Verify housing does not lose pressure
Remove vacuum pump and secure vacuum plug

3) Power-Up

Control box up and running
Ensure deck crew members are attentive
Shout, "Power On!"
Power on vehicle
Call out, "Thruster Test"
Perform thruster test with appropriate controls
Verify video cameras
Test manipulators

4) Launch

Call out, "Prepare for launch"
Deck crew respond "Affirmative"
Insert ROV in water and hold
Wait for release order

5) In Water

Check for bubbles
Check leak sensor
If leak sensor goes off, surface immediately
If bubbles are coming from housing, surface immediately

6) ROV Retrieval

Pilot announces "ROV Surfacing"
Deck crew responds, "ROV on surface"
Thrusters disabled
Flight engineer turns off vehicle power
Pilot calls out, "Safe to remove ROV"
After ROV is secured on the surface, deck crew shout "Secured"

Water Leak Protocol

Surface immediately
Power off ROV
Power off Control Box
Open housing and remove electronics for inspection

Communication Loss Protocol

Reboot all systems
If no communication power off ROV
Surface via tether
Resume operation when communication returns

Visual Inspection

Inspect motors for any obstructions
Check for loose end caps
Check for hanging wires
Inspect frame for any damage
Check for any loose securing system

COMPANY INFORMATION

COMPANY HISTORY

45C Robotics is a rising competitor in the MATE ROV competition. While the California Academy of Mathematics and Science (CAMS) ROV team operated as an elective course in previous years, two years ago we branched off as an individual robotics club. By becoming a separate entity from the school, as opposed to a required class, we became comprised of only the most passionate engineers, marketers, and programmers to ensure success. Although the CAMS ROV teams had continually placed nearly last at regionals every year, the 45C Robotics team not only advanced from the regional competition, but also ranked 10th out of 40 competitors in the 2018 MATE international competition within the second year of operation.

Moreover, we were awarded the safety consciousness award in 2017 for our stellar safety measures, including a safety checklist that was followed before any procedure. Alongside 45C's rapid growth, we have experimented with numerous organizational structures, and are currently using the most effective system we've found. 45C Robotics utilizes a flat company structure to minimize bureaucracy and maximize efficiency. We have a Chief Executive Officer (CEO) and a leader in each subsection (e.g. Electrical, Mechanical, Programming, Finance, and Machining) to keep all team members focused and generally informed through the chain of command.

Instead of designating specific tasks for every team member, each person decides his or her own projects to work on and enlists the help of peers accordingly. Each subteam works independently, but retains close communication with the leaders and the rest of the members. This gives team members the opportunity to receive feedback without leadership micromanaging details. Each sub-leader takes charge of specific area of communication and development (e.g. creating a parts list, finding sponsor contacts, etc.) and helps guide each sub-team to complete major tasks (e.g. designing and fabricating the control box) in an organized and timely manner. This organizational structure promotes productivity by creating a safe space for the blooming of ideas assisted by constructive criticism, which has allowed us to create an efficient robot.

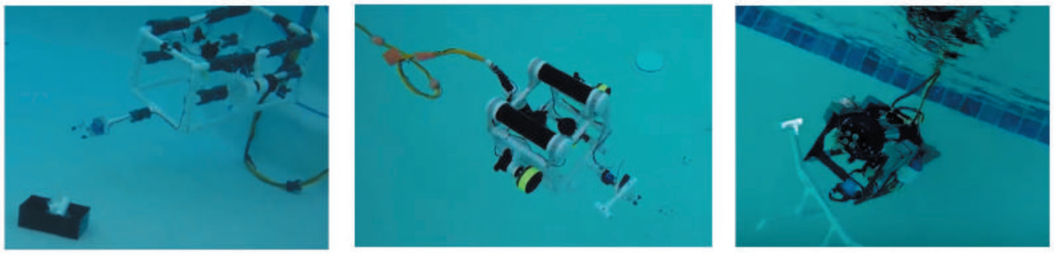


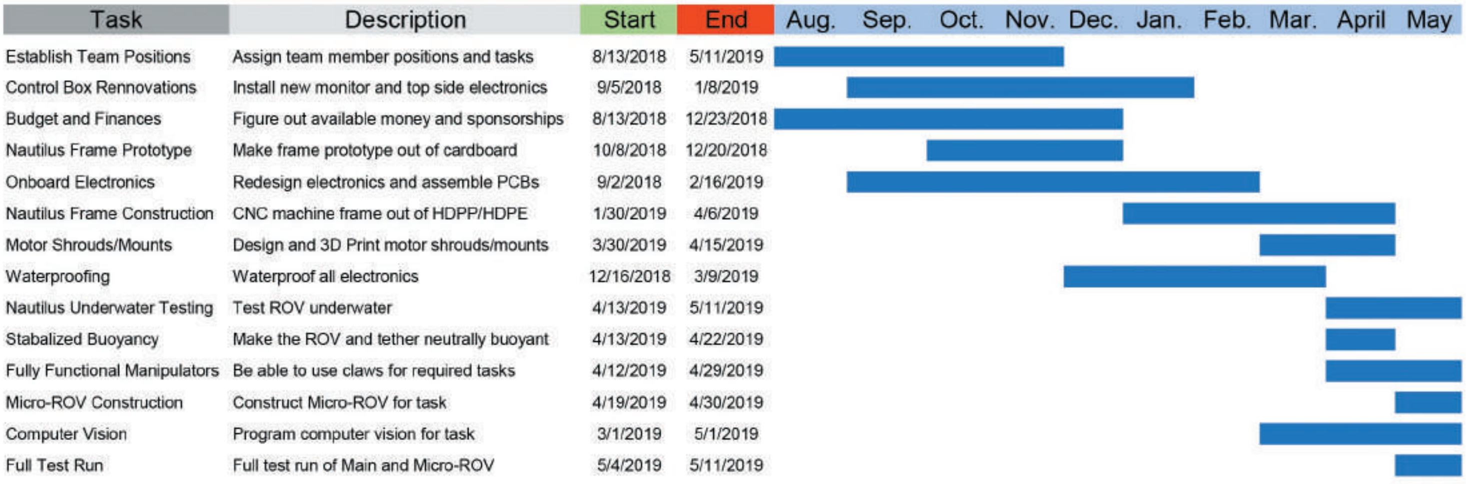
Figure 19A/B/C: 45C Robotics' ROV Evolution

SCHEDULING AND PROJECT MANAGEMENT

45C Robotics began meeting during the summer of 2018 and met every week for a total of 300 hours. During the school year we met every Monday and Friday for a total of 4 hours. As we approached the competition, extra meetings were scheduled to meet deadlines. In order to ensure our team had plenty of time to complete and practice with our robot, we implemented a strict attendance policy: every team member was required to attend at least 80% of all meetings. Additionally, to ensure every team member fully understood the ROV as it was updated, the CEO hosted info sessions to inform everyone about the latest upgrade of the Nautilus.

Prior to any meeting, each sub-leader would write down what needed to be accomplished on a to-do list to stay on task and complete projects on time. At the beginning of meetings, we would talk about what had been accomplished previously, and what was to be done next. Then, we would split into small groups (2-3 people) based on interest. These groups mingle and switched members throughout the year as every member became involved in multiple aspects of the ROV, helping every member understand every component.

After every meeting, sub-leaders would state what was accomplished, what needed to be done in the next meeting and updated the to-do list accordingly. The to-do list was shared with every member to ensure everyone knows the current pace of progress. As a team, we finished the robot by April to ensure an ample amount of time for pool practice. As we approached even closer to the competition, meetings became more frequent and longer to allow time for troubleshooting, pool practice, technical writing, and presentation practice.



Task	Description	Start	End	Aug.	Sep.	Oct.	Nov.	Dec.	Jan.	Feb.	Mar.	April	May
Establish Team Positions	Assign team member positions and tasks	8/13/2018	5/11/2019	■	■	■	■						
Control Box Rennovations	Install new monitor and top side electronics	9/5/2018	1/8/2019		■	■	■	■	■				
Budget and Finances	Figure out available money and sponsorships	8/13/2018	12/23/2018	■	■	■	■	■					
Nautilus Frame Prototype	Make frame prototype out of cardboard	10/8/2018	12/20/2018			■	■	■					
Onboard Electronics	Redesign electronics and assemble PCBs	9/2/2018	2/16/2019		■	■	■	■	■	■			
Nautilus Frame Construction	CNC machine frame out of HDPP/HDPE	1/30/2019	4/6/2019						■	■	■	■	
Motor Shrouds/Mounts	Design and 3D Print motor shrouds/mounts	3/30/2019	4/15/2019								■	■	
Waterproofing	Waterproof all electronics	12/16/2018	3/9/2019					■	■	■	■		
Nautilus Underwater Testing	Test ROV underwater	4/13/2019	5/11/2019									■	■
Stabalized Buoyancy	Make the ROV and tether neutrally buoyant	4/13/2019	4/22/2019									■	
Fully Functional Manipulators	Be able to use claws for required tasks	4/12/2019	4/29/2019									■	■
Micro-ROV Construction	Construct Micro-ROV for task	4/19/2019	4/30/2019										■
Computer Vision	Program computer vision for task	3/1/2019	5/1/2019								■	■	■
Full Test Run	Full test run of Main and Micro-ROV	5/4/2019	5/11/2019										■

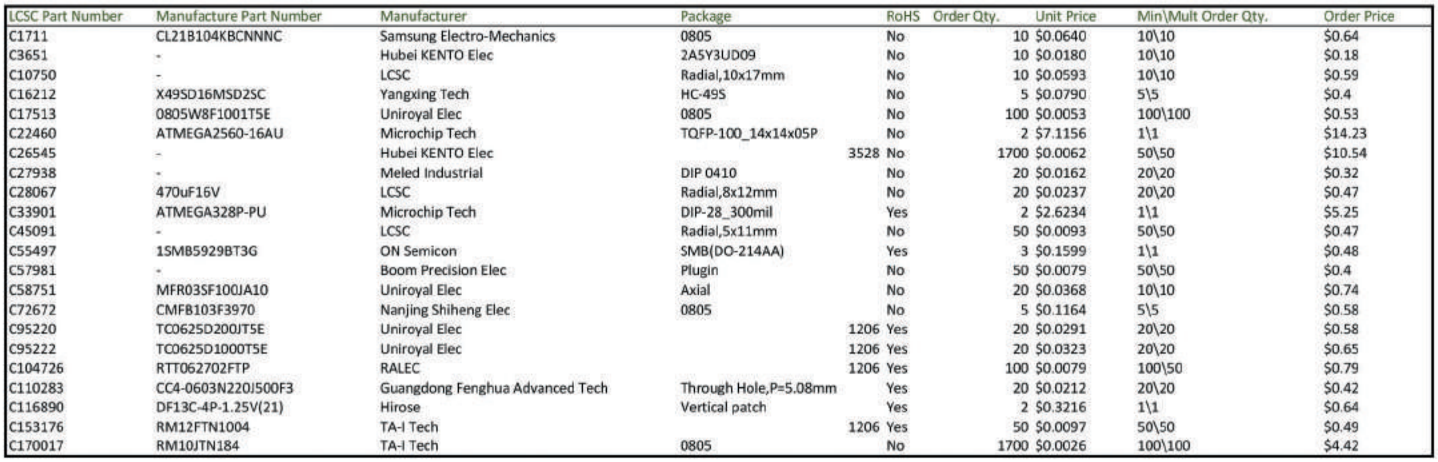
Figure 20: Schedule Chart

FINANCES

BUDGET

Our sponsors this year generously provided many components for the construction of our ROV. Maxim Integrated, Microchip, and Digikey provided many of the necessary electrical components in order to assemble our PCBs. SeeedStudio and PCBway heavily discounted the price needed to print our circuit boards. As a result, our electrical system was relatively cheap to make. Progressive RC donated all of our ESCs for our brushless motors. Dassault Systemes, generously provided all members of our team free student licenses of Solidworks. BlueRobotics provided a 10% off on our purchase of the M100 motors and other components. JetBrains provided the Pycharm software at no cost for our programming team. Loctite provided epoxy, which was used to waterproof all of our electronics. Hakko donated soldering equipment, including a battery powered soldering iron, and a fume extractor. ServoCity provided us with the claw mechanisms necessary for the tasks. Vicor sent us 2 isolated power supplies for free. Last but not least, Heyco provided a strain relief adapter, Techflex donated heatshrink and protective sleeves for our tether, and OurOwnStyle gave $200 to spend on any electronics or equipment we found necessary to build the Nautilus and the MicroROV.

The process of ordering parts starts with a Bill Of Materials (BOM). On the BOM we put the part name, its part number, quantity, vendor, the link (to access the part online), the use of the part, cost of each part, total cost (if there are more than 1), and any additional notes needed to order the part. This BOM is then approved by our mentor and then processed by the banker. In order to conserve some of the school's money,



LCSC Part Number	Manufacture Part Number	Manufacturer	Package	RoHS	Order Qty.	Unit Price	Min\Mult Order Qty.	Order Price
C1711	CL21B104KBCNNNC	Samsung Electro-Mechanics	0805	No	10	$0.0640	10\10	$0.64
C3651	-	Hubei KENTO Elec	2A5Y3UD09	No	10	$0.0180	10\10	$0.18
C10750	-	LCSC	Radial,10x17mm	No	10	$0.0593	10\10	$0.59
C16212	X49SD16MSD2SC	Yangxing Tech	HC-49S	No	5	$0.0790	5\5	$0.4
C17513	0805W8F1001T5E	Uniroyal Elec	0805	No	100	$0.0053	100\100	$0.53
C22460	ATMEGA2560-16AU	Microchip Tech	TQFP-100_14x14x05P	No	2	$7.1156	1\1	$14.23
C26545	-	Hubei KENTO Elec	3528	No	1700	$0.0062	50\50	$10.54
C27938	-	Meled Industrial	DIP 0410	No	20	$0.0162	20\20	$0.32
C28067	470uF16V	LCSC	Radial,8x12mm	No	20	$0.0237	20\20	$0.47
C33901	ATMEGA328P-PU	Microchip Tech	DIP-28_300mil	Yes	2	$2.6234	1\1	$5.25
C45091	-	LCSC	Radial,5x11mm	No	50	$0.0093	50\50	$0.47
C55497	1SMB5929BT3G	ON Semicon	SMB(DO-214AA)	Yes	3	$0.1599	1\1	$0.48
C57981	-	Boom Precision Elec	Plugin	No	50	$0.0079	50\50	$0.4
C58751	MFR03SF100JA10	Uniroyal Elec	Axial	No	20	$0.0368	10\10	$0.74
C72672	CMFB103F3970	Nanjing Shiheng Elec	0805	No	5	$0.1164	5\5	$0.58
C95220	TC0625D200JT5E	Uniroyal Elec	1206	Yes	20	$0.0291	20\20	$0.58
C95222	TC0625D1000T5E	Uniroyal Elec	1206	Yes	20	$0.0323	20\20	$0.65
C104726	RTT062702FTP	RALEC	1206	Yes	100	$0.0079	100\50	$0.79
C110283	CC4-0603N220J500F3	Guangdong Fenghua Advanced Tech	Through Hole,P=5.08mm	Yes	20	$0.0212	20\20	$0.42
C116890	DF13C-4P-1.25V(21)	Hirose	Vertical patch	Yes	2	$0.3216	1\1	$0.64
C153176	RM12FTN1004	TA-I Tech	1206	Yes	50	$0.0097	50\50	$0.49
C170017	RM10JTN184	TA-I Tech	0805	No	1700	$0.0026	100\100	$4.42

Figure 21: Example BOM Sheet

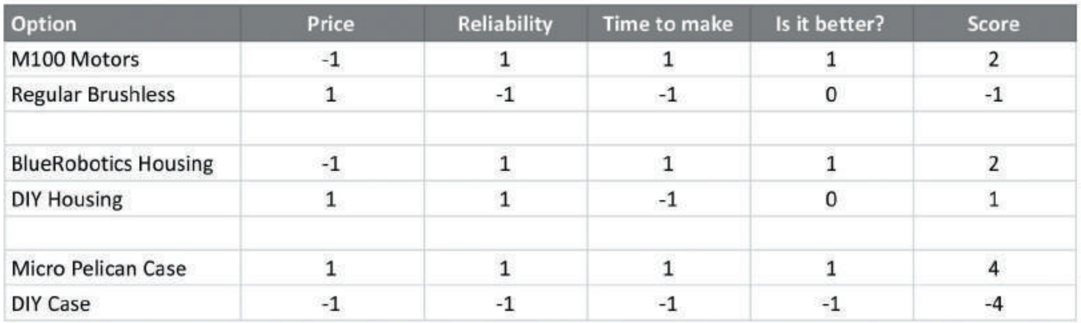
BUILD VS. BUY

At 45C Robotics, most of our mechanical, and electrical systems are self made. However, we choose to buy components for major systems (specifically propulsion and optical). To determine whether to buy something or not, we first ask ourselves, "Build or Buy?"If we reason that building the component ourselves will cost us more time and money than buying it, we buy the component.

The main components we bought this year were the on board electronics housing, the micro-rov housing, and our M100 motors. One major problem we faced last year was waterproofing. We often took hours applying waterproofing materials to each component to insure water-damage is prevented. Since the onboard electronics are complex and critical to the ROV, we deemed it'd take too much time, and risk to make our own housing.. Therefore, we decided to buy the Blue Robotics Onboard electronic housing, since the housing is extremely reliable in terms of waterproofing, and is easily attachable to the ROV. In addition, this year we decided to use brushless motors. Along with that decision came the problem of waterproofing them. We considered buying brushless motors and waterproofing them ourselves but that required a vacuum which we did not have and ensuring that no coils would come in contact with water would be risky since any exposure would cause corrosion and further complications. Because of this, we decided to buy six M100 motors from BlueRobotics which are waterproofed and ensured to prevent corrosion throughout its lifetime. For the micro-rov we decided to 3D print our own housing out of ABS since it was proven to be cheaper and more able to be worked with. Since it was custom built, this allowed us to implement components for the MicroROV more easily than if we had bought one. Moreover we were able to ensure that it met MATE competition standards since we had designed it ourselves.

NEW VS. REUSED

Since we were initially low on funding this year, we esteemed cost efficiency as paramount during our design process. In order to save money, we re-used as many components from our last ROV as possible. However, since components such as the camera and servos frequently had waterproofing problems, they had to be replaced often.



Option	Price	Reliability	Time to make	Is it better?	Score
M100 Motors	-1	1	1	1	2
Regular Brushless	1	-1	-1	0	-1
BlueRobotics Housing	-1	1	1	1	2
DIY Housing	1	1	-1	0	1
Micro Pelican Case	1	1	1	1	4
DIY Case	-1	-1	-1	-1	-4

Figure 22: Decision Matrix for Build VS. Buy

BUDGET MANAGEMENT

At the beginning of the season, our finance lead laid out a budget plan to estimate expenses, based on last year. With a target in mind, the finance lead led we in raising funds in various methods as aforementioned. Before making any purchases, we held a voting poll on messenger to ensure everyone agrees that the component being bought is worth the cost. Whenever a purchase was made, the receipts were collected and saved for future reimbursement by CAMS. Each receipt was then recorded into the Budget sheet attached below. Additionally, any funding gained was added into the Budget sheet to keep track of excess funding which can be utilized for future ROV projects or transportation costs.

CONCLUSION

TECHNICAL CHALLENGES

A major issue we encountered was the extreme high EMF from the brushless motors due to regenerative braking. These EMF noise can interfere with camera lines and cause major flickering. In order to reduce the EMF, we designed a custom PCB with a zener diode and a Heat Dissipating resistor. This allows any excess voltage spike to be converted into heat thus reducing the spike. However, even with this circuit, the camera still experienced flickering when the motors are moved. This was finally resolved by using an Isolated DC-DC power supply, which can protect the output from voltage surges of up to 50V for 100ms.

We originally tried making our custom motor mounts out of two pieces of acrylic: Tube and base plate. The original idea was to have the base plate and the tube welded together with acrylic glue. However, due to time constraints and the fragility of acrylic, we decided to redesign the motor mounting system and 3D printed the design.

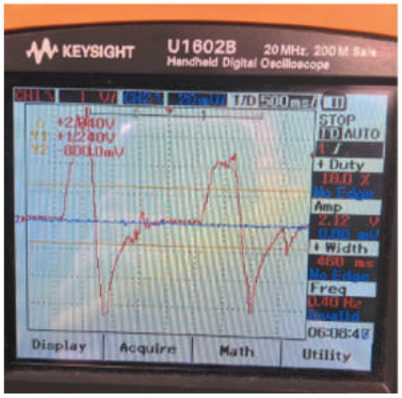


Figure 23A: Voltage Spike

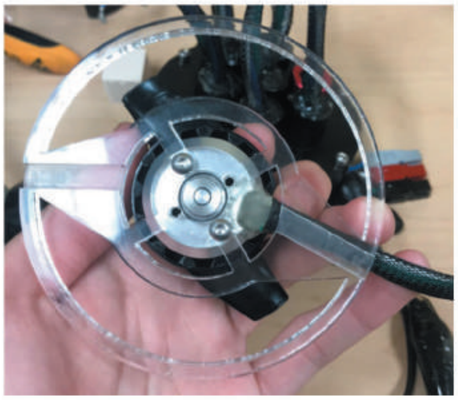


Figure 23B: Acrylic Motor Mounts

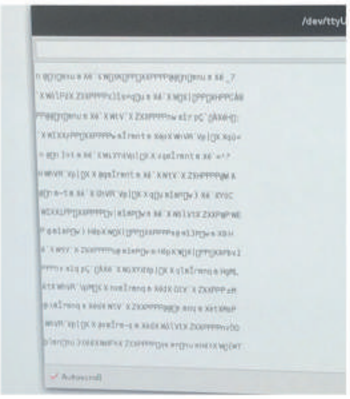


Figure 23C: Serial Corrupted Data

NON-TECHNICAL CHALLENGES

A major challenge we had with the team this year was the seniors' lack of dedication towards the team. As college season ended, many of the senior leads lacked the motivation to continue with the project. This proved to be a major setback as many of the seniors were picked as leads. To resolve this issue, the seniors lacking the dedication were demoted to regular positions while new members took their place. This not only provided less responsibilities to the seniors but trained the new members to lead the ROV team after the seniors graduate.

LESSONS LEARNED

The PCB software we use to design the PCBs is EasyEDA. This software allowed us to create our own PCB design. The software has very little limitations, which means we can create PCBs of different shapes and sizes. We were able to orientate and position our components to minimize the size of entire PCB. EasyEDA also has BOM tools, which allows us to calculate the price of all the parts, without individually recording, and potentially miscounting, each one.

Since we started making our own PCBs, we had to learn how to properly solder SMD and through-hole components. Each component has their own advantages and disadvantages. SMD components are smaller, which allows us to make more complex PCBs in smaller sizes. However, due to their size, they are harder to solder on, and they be misplaced, easier. Through-hole components are easier to solder, and they go through the board, allowing us to make traces on the other side of the board. There is a trade-off, through-hole components are inherently larger.

When programming for Serial communications, delays needed to be implemented as a means to have stable transfers of data between the boards. Adding the delays to the code was of no difficulty. The difficult part was getting the "right" delays.

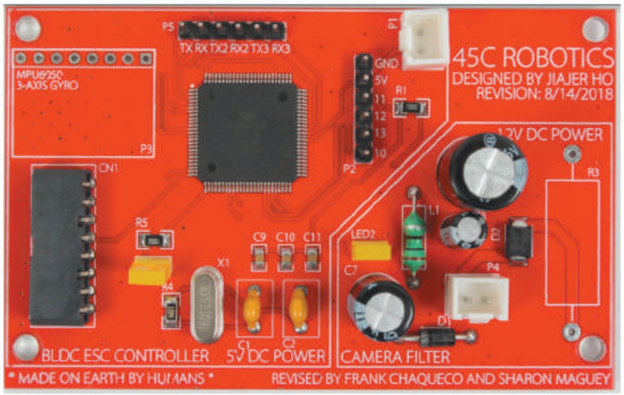


Figure 24A: PCB from August of 2018

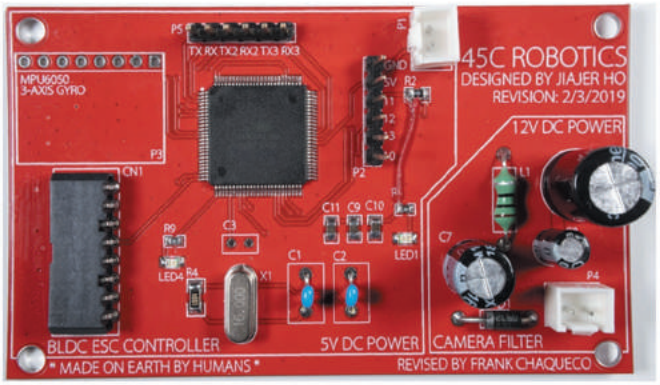


Figure 24B: PCB from February of 2019

Throughout this year, it was emphasized how we shouldn't have done many of the concepts at the last minute, yet it still proved a great problem. We were able to complete our robot in time, but, due to time constraints, were not able to implement concepts that we believed could improve our robot. Therefore, we have learned that it is of vital importance to manage our time better.

FUTURE IMPROVEMENTS

To more efficiently make use of space, we plan to use circular PCB. By using EasyEDA we will be able to redesign our PCBs to fit the diameter of the onboard housing. Another improvement, would be to use polycarbonate which is lighter, stronger, and more flexible for our motor mounts rather than acrylic. An issue that we will work on is the overflow of epoxy when epoxying the cable penetrators. By making sure that the epoxy is laid as flat as possible and applying the right amount we will hopefully be able to solve this problem. Something that would have made the programming team more effective would have been the use of a library for the serial communication code. This will make it easier to edit and faster to access. Moreover, creating more efficient code will decrease the delay from the data being sent and received between the boards. This will result in better control of the ROV and more accurate sensors data. In addition, we plan to apply control theory to the ROV which will increase its stability. Finally we plan to use a filter on the sensors used will give more accurate and precise data by reducing the noise to a minimum.

The MicroROV was not given much attention, due to the time constraints that were present. If our time was used efficiently, we could've made the MicroROV more efficient, and possibly implement Fiber Optics into its tether. When the Nautilus' frame was CADed, there was little consideration for how it was going to be assembled. We used brackets to connect the pieces together. However, the frame would've been stronger and more stable if assembly was considered in the beginning.

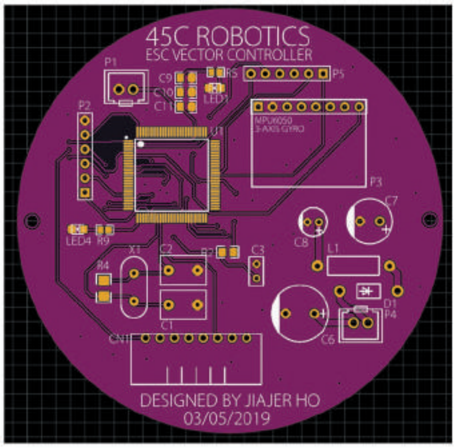


Figure 25: Potential Circular PCB design

ACKNOWLEDGEMENTS

We would like to thank the following individuals and organizations for their support:

Sandra James, Joseph Carpenter, Erika Dreyfus
Karina Garcia Takata, Pankaj Topiwala, Vibhuti Bhagwati, Alberto Chaqueco
James Masukawa, Jeyem Ortiz
Martin Klein, Jill Zande, Timmie Sinclair, Scott Fraser
All the MATE Competition officials and volunteers

REFERENCES

Substring(). (2019, April 06). Retrieved April 22, 2019, from https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/substring/

IndexOf(). (2019, April 06). Retrieved from https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/indexof/

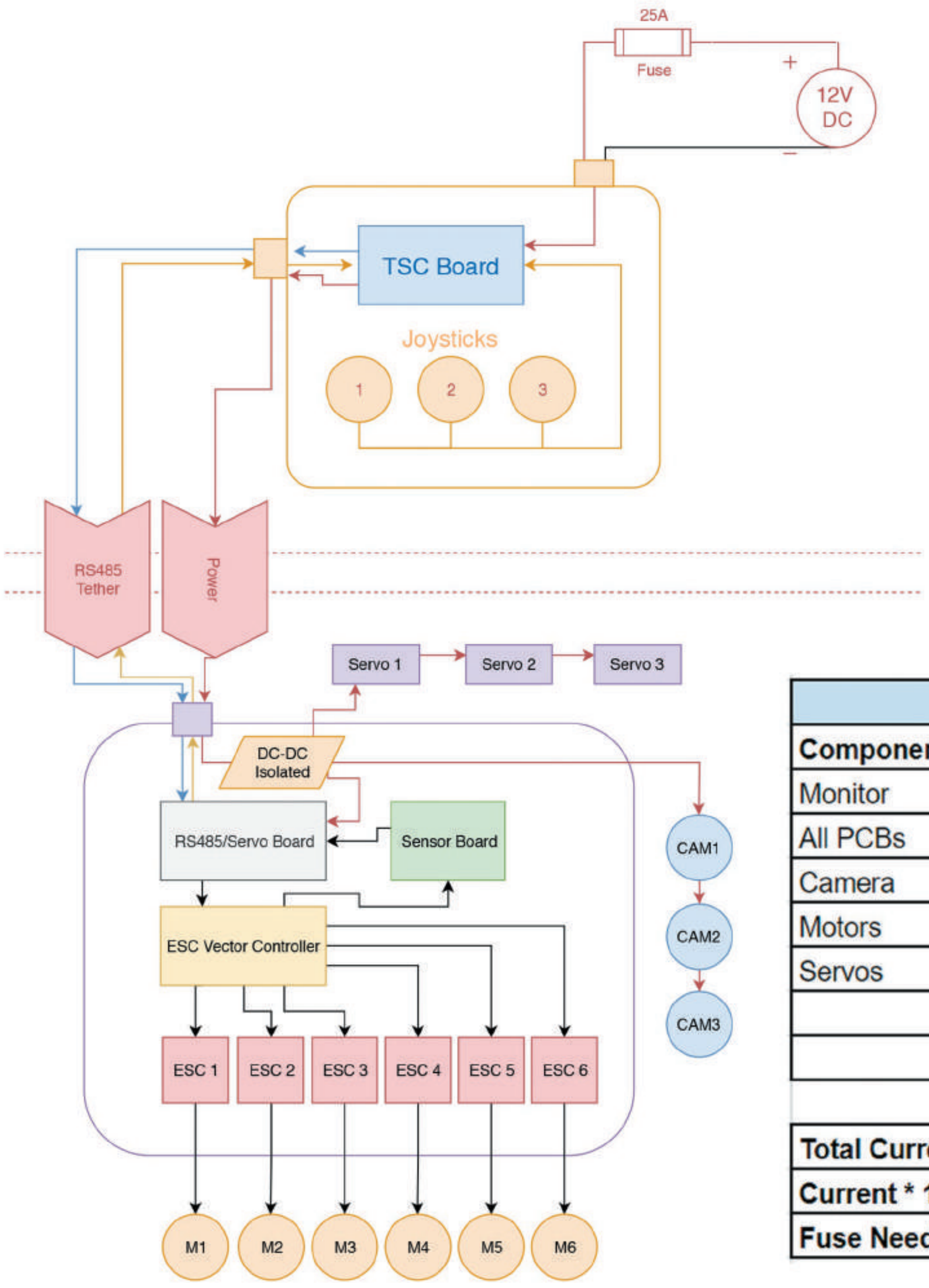
DHT11 & DHT22 Sensor Temperature and Humidity Tutorial. (2018, October 21). Retrieved April 22, 2019, from https://howtomechatronics.com/tutorials/arduino/dht11-dht22-sensors-temperature-and-humidity-tutorial-using-arduino/

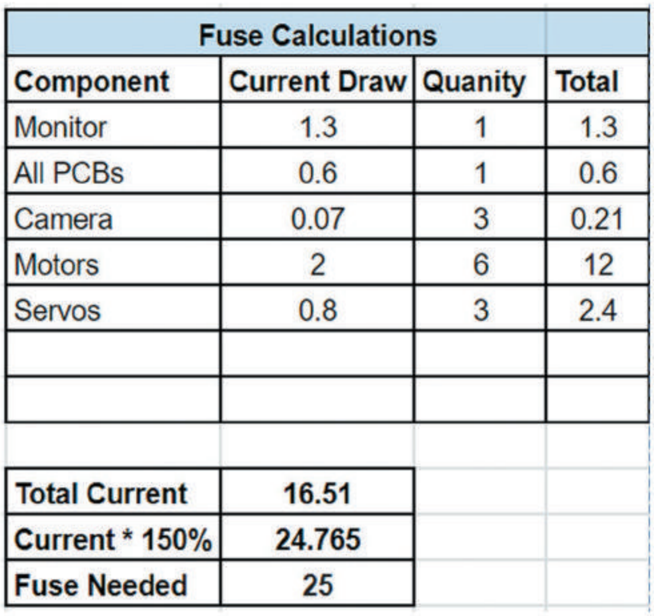
Instructables. (2017, October 19). Arduino Servo Potentiometer. Retrieved April 22, 2019, from https://www.instructables.com/id/ArduinoServoPotentiometer/

Instructables. (2017, September 25). Simple DHT11 Temperature Monitor. Retrieved April 22, 2019, from https://www.instructables.com/id/Simple-DHT11-Temperature-Moniter/

APPENDICES

SID AND FUSE CALCULATIONS





Fuse Calculations			
Component	**Current Draw**	**Quanity**	**Total**
Monitor	1.3	1	1.3
All PCBs	0.6	1	0.6
Camera	0.07	3	0.21
Motors	2	6	12
Servos	0.8	3	2.4
Total Current	**16.51**		
Current * 150%	**24.765**		
Fuse Needed	**25**		

BUDGET SHEET

Category	Amount Spent	Donated/Discounted	Fair Market Value
Frame/Flotation			
Seaboard HDPE	0	Dontated	20
HDPP	24	N/A	24
CNC Machining	60	Discounted	120
L200 Bouyancy Foam	10.15	N/A	10.15
On Board Housing			
Tube	32.42	N/A	32.42
Cable Glands	0	Donated	20.53
TechFlex Wrap	0	Donated	43.11
Thursters			
Motors	378	Discounted	420
ESCs	0	Donated	99.43
Electronics			
Printed Circuit Boards	0	Donated	120
Electrical Componenets	0	Donated	200
Control Box	0	Donated	120
Cables	73.12	N/A	73.12
Power Converters	0	Donated	250
Flight Computers	0	Donated	470
Cameras			
Cameras	53.12	N/A	53.12
Coaxial Video Cables	0	Donated	45.32
Tether			
Flexiable Power wires	52.43	N/A	52.43
Neutral Bouyant Tether	109.66	Discounted	121.84
Tools			
Savox Servo Motor	27.12	Discounted	120.4
Manipulators	0	Donated	48.32
Misceallaneous			
Hardware screws	21.24	N/A	21.24
3D Printed Materials	0	Donated	42.66
Misc.	57.12	N/A	57.12

Amount Spent	898.38
Amount Donated	1686.83
Total ROV Cost	2585.21

PROGRAMMING AND COM DIAGRAM