GTSR Open Source ROV Instruction Manual

(Draft Version)

Georgia Tech System Research (GTSR)

Qiuyang Tao

Advisor: Dr. Fumin Zhang

Dec/28/2016

This document applies to the Open Source Underwater Vehicle developed during spring 2015.
To obtain latest version, please visit GT-MUR repository on GitHub.

Introduction

The Georgia Tech Miniature Underwater Robot (GT-MUR) is a small size underwater vehicle dually used as research and educational platform. Featured as miniature size, safe operation, and low cost, GT-MUR is a perfect platform for environmental sampling, human-robot interaction, and multi-robot sensing network.

GT-MUR started as part of Vertically Integrated Projects (VIP) Program around two years ago, and the team aimed to realize a thorough system which can be conveniently built using inexpensive, easily accessible materials and common tools that available in most engineering schools.

We have built a prototypical platform that consists of one underwater robot that is able to communicate with a hand-held controller. A pool test was conducted in summer 2016 with this early prototype. The system demonstrated the functionality of remotely controlling the vehicle using inferred (IR) communication.

We decided to make this ROV (Remotely Operated Vehicle) completely open source, both hardware and software, to the public. Researchers, educators, students, as well as hobbyists can make their own GT-MUR and modify the design based on their needs. We believe by sharing this easy-to-built prototype, more people will be interested and get involved in underwater robotics.

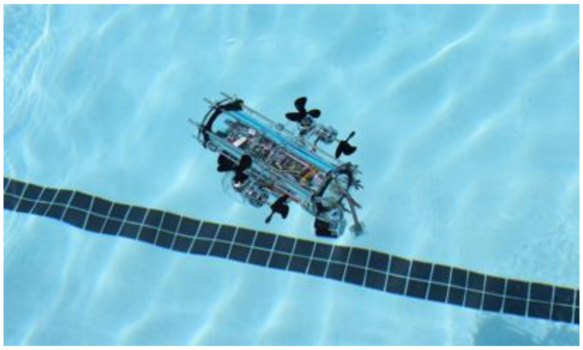

Figure 1. GT-MUR pool test in shallow fresh water.

Mechanical design

In an effort to build a precise and reliable mechanical structure with common tools, we minimized the number of parts and simplified the design to make the vehicle easier to build. All design files are created using AutoCAD and Autodesk Inventor, which provides accurate graphics and allows convenient modifications. Laser cutter and other tools were utilized to improve the precision during prototyping. All the CAD drawings can be found in the open source GT-MUR package available from our GitHub repository.

The overall structure of the vehicle is shown in Figure 2. Six threaded rods (three rods are shown in the plot) run parallel to the cylindrical housing and fasten two circular plates at both ends (only one plate is shown in the figure). The plate pushes a rubber pad (the black part in between) against the tube to make it waterproof. The threaded rods also provide support for the thruster-holding plates as well as the ballast weights.

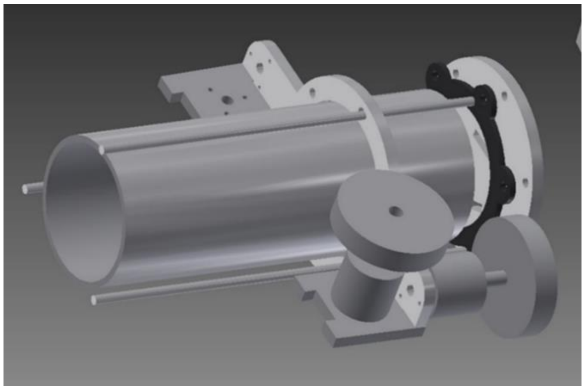

Figure 2. Overall structure of GT-MUR.

1. Waterproof housing

The watertight housing is greatly simplified without significantly sacrificing its waterproof performance. This simplification makes it possible to build the vehicle without professional tools or specialized components.

As discussed earlier, the waterproof housing consists of a cylindrical tube, a pair of rubber ring, two end caps and six threaded rods. The reason for choosing this configuration as well as detailed instructions on machining will be discussed in the following sections.

1.1 Transparent Acrylic / Polycarbonate tube

Most underwater vehicles use metallic or high-strength composite materials for their pressure hulls. However, since our vehicle operates in shallow fresh water, inexpensive consumer grade plastic tubes are the best choice. These materials are known for low price, light weight, easily accessible, but not precisely manufactured. To minimize this side effect, different water tight approach (use rubber pad instead of an O-ring) is developed and will be discussed in section 1.2.

This vehicle uses infrared communication, which needs to send/receive invisible light to/from the hand-hold controller. The overall design can be simplified if the IR emitter and receiver are placed inside a transparent hull instead of mounted outside. The transparent housing not only allows infrared light, which serves as the remote control information carrier, to penetrate through; but also makes its inner electronics easily observable, which helps to monitor the system and serves the educational purpose.

Acrylic and polycarbonate are two of the most popular materials for transparent tubes. Acrylic is typically cheaper and clearer while polycarbonate is more shock resistive. For this prototype, an 8-inch long polycarbonate tube with a 3-inch outer diameter and 0.22/0.25-inch wall thickness is preferred. Similar to the cylindrical tube, the end caps of the underwater housing are cut from an acrylic sheet with 0.22/0.25-inch thickness.

A flat and smooth cut of the tube is critical to the waterproof performance. Lathing is a typical way for metallic materials. Shown in Figure 3b, an Aluminum tube is machined nicely. However, this approach is very hard to be applied on plastic tubes. Plastic becomes “sticky” to the lathe tool during machining and Figure 3a shows an unsuccessful trial.

Figure 3. a) Unsuccessful lathe-cut plastic tube (left) b) lathe-cut Aluminum tube (right)

Instead, we use a miter saw to cut the tube and sand the cutting surface to make it flat. Before turning on the saw, it is highly suggested to use thick tape to protect the inner and outer walls of the tube from debris, as shown in Figure 4.

Figure 4. Inner and outer walls of the tube are protected by thick tape.

1.2 Why rubber pad instead of O-ring

O-rings are one of the most commonly used seals in underwater robotics and other applications. They are inexpensive, easy to buy and come with a variety of standardized sizes.

However, many easy-accessible plastic tubes, especially the ones with the size we need, don’t have precise cross-sectional shape. In other words, their cross profile might be an ellipse instead of a circle. Leakage may happen near the two vertexes of the ellipse. In addition, gland of the O-ring need to be precisely manufactured, which may potentially hinder the popularization of this vehicle.

As shown in Figure 5a, we use rubber pads for the waterproof housing. This approach has large tolerance to the imperfect shape of the plastic tubes. There are six holes on the edge of the pad, which avoid slippage and misalignment. Thickness of the rubber pad is empirically chosen to be 1/8 inch, and Figure 5b demonstrates how the tube, rubber pad and end cap stacked together. Figure 6 shows the assembled water tight housing. We tested this approach by pushing the assembly at the bottom of a swimming pool (~3m), and there is no leakage.

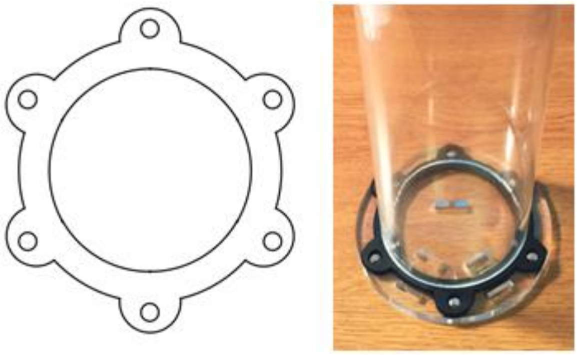

Figure 5. a) CAD drawing of the rubber pad (left) b) Rubber pad stacked with end cap and Acrylic tube (right)

Figure 6. Assembled water tight housing.

1.3 Wire penetrates through end caps

Motors, hydro pressure sensor, and other devices need to be installed outside the waterproof housing. Underwater connectors are convenient to use but their size is too big and not cost effective. To simplify the design and reduce cost, we epoxied the cables with the end caps.

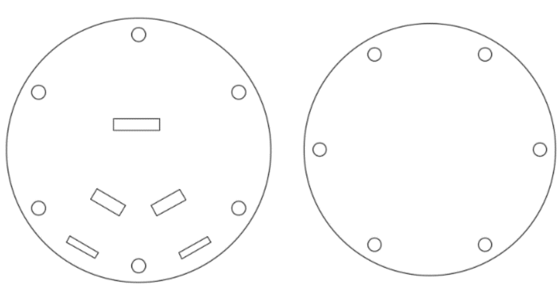

Figure 7. CAD drawing of the end caps.

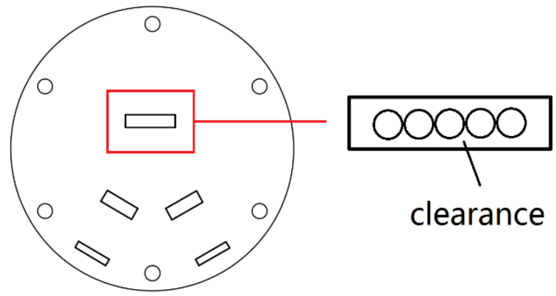

Figure 8. Clearance is needed between the wires and the square notch.

Special care is needed when epoxying the cables. Epoxy should be filled between the wires and the square cut-out on the end cap as shown in Figure 8. Before applying the epoxy, two preparation steps is recommended as demonstrated in Figure 9. First, limit the position of the wires by applying transparent tape on one side of the end cap. Next, hold the wires vertically to the plate, then apply hot melt glue at the same side of the tape. The glue will fill the gap between the two, and keep the wires in place. After these two steps, epoxy can be easily added from the other side of the end cap. Figure 10 shows the finished assembly.

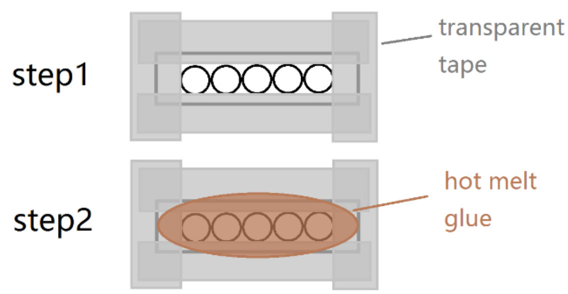

Figure 9. Preparation steps before apply epoxy.

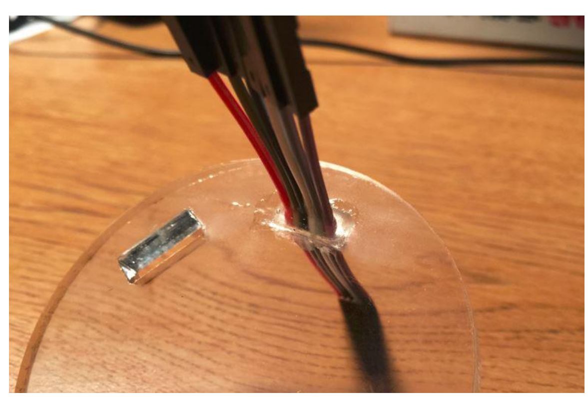

Figure 10. Wires are expoxied on the end cap.

2. Thrusting system

GT-MUR uses four motors to control its motion. Two of them are mounted upward to provide vertical thrust; while the other two motors are installed horizontally at each side of the vehicle to control horizontal movement, which is similar to differential wheeled robots. Both vertically and horizontally mounted motor pairs use counter-rotating propellers to minimize torque reaction, as shown in Figure 11.

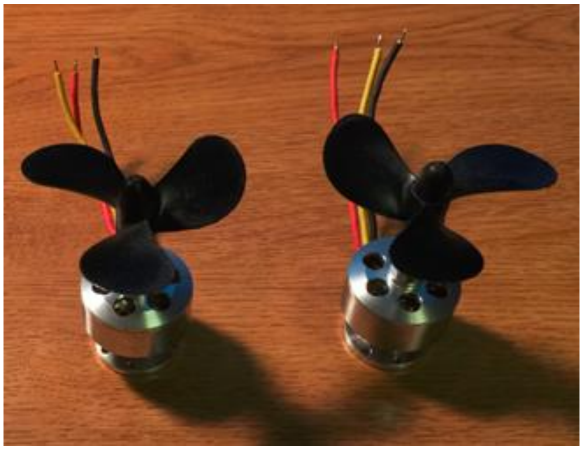

Figure 11. BLDC motors with counter-rotating propeller.

Professional underwater thrusters will significantly elevate the budget of the vehicle. Therefore we use inexpensive BLDC (brushless DC) motors considering our vehicle mainly operates in shallow fresh water. The out-runner BLDC motor we use has its windings on the stator, therefore permanent magnets are the major moving part. No brush, commutator or hall sensors are needed, which makes it possible to be submerged underwater without significant modification or protection.

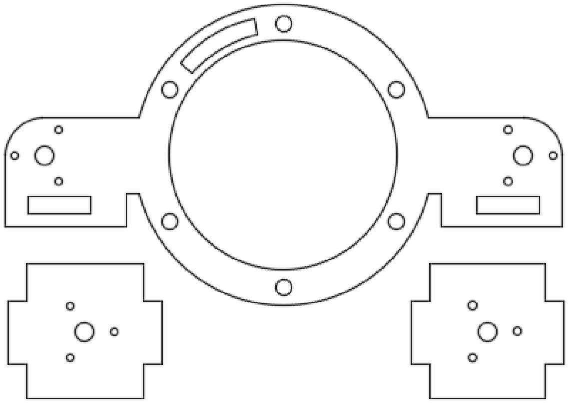

Figure 12. Plates used to mount the motors.

All the four motors are mounted on the laser-cut plastic plates shown in Figure 12. Each motor is mounted with three screws and the plates are installed on the threaded rods with nuts and washers, as shown in Figure 13.

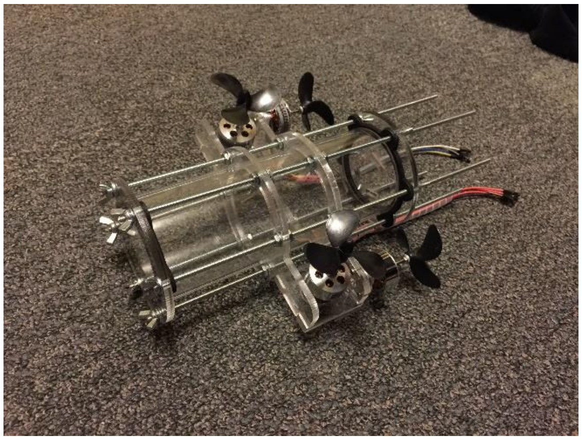

Figure 13. Motors mounted on GT-MUR.

To make sure the motors are aligned with the cylindrical tube, the plates need to be installed perpendicular to the threaded rods. Also, the vertical motors should be placed near the center of the vehicle to avoid undesired pitching moment. This can be adjusted by trimming the position of the nuts which fasten the motor-holding plates.

3. Chassis for electronics

A plastic chassis is placed inside the cylindrical housing to mount the electronic devices. As shown in Figure 14, the chassis is a rectangular plate with one wheel on each end. The diameter of the wheels is slightly smaller than that of the hull so that the chassis can fit in perfectly. To better utilize the limited space inside the hull, electronics are mainly placed on the upper side while the battery is installed beneath the chassis, as shown in Figure 15.

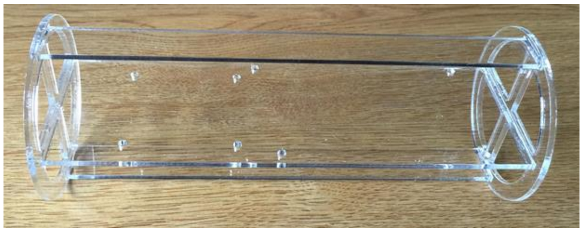

Figure 14. Chassis for electronics.

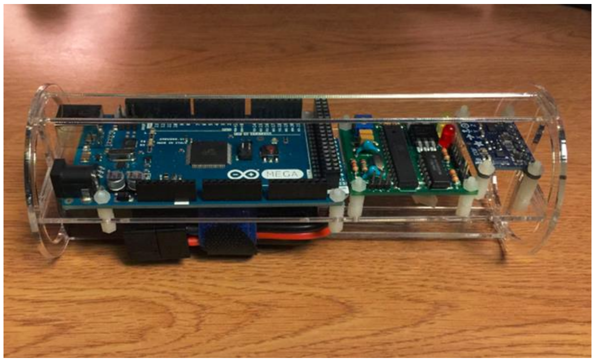

Figure 15. Electronics mounted on the chassis.

4. Ballasting

The purpose of adding balance weight is to ensure the ROV's density to be slightly less than water. So in cases of failure such as losing power or signal, the robot can automatically surface up. The ballast weight can be attached to the threaded rod at the bottom. This configuration lowers the center of mass and make it easier to trim the balancing of the vehicle by adjusting the position of the ballast weight along the rods.

ELECTRONICS

The entire electronic system consists two major parts, a hand-hold controller and an underwater vehicle. Figure 18 and 19 shows the assembled system and its block diagram respectively. To reduce the cost and simplify the design, we choose popular electronics that can be easily purchased from major distributors with low price. Part list and design files are available in the GT-MUR package on our GitHub repository.

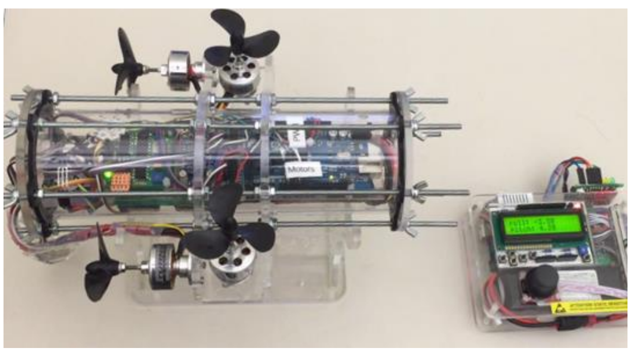

Figure 18. ROV (left) and hand-hold controller (right)

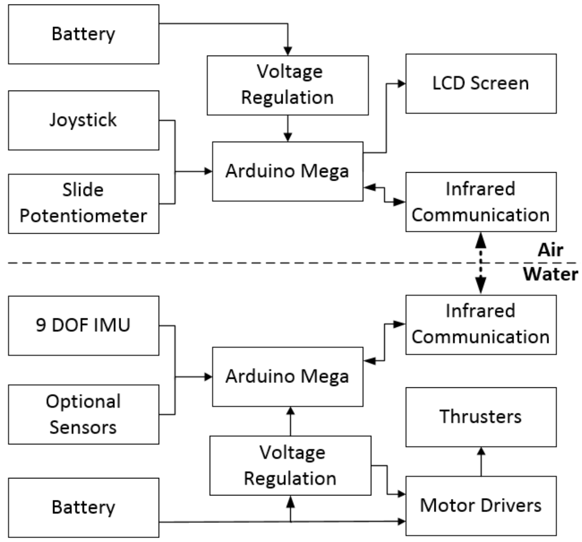

Figure 19. Block diagram of the entire system.

1. Handhold controller

The hand-hold controller is the main user interface for sending commands to the ROV and displaying the sensor measurement from the vehicle. Design and implementation of this hand-hold controller will be discussed in the following sections.

1.1 Micro Controller

Arduino Mega 2560 is chosen as the main platform for rapid prototyping. It reads user input from the joystick and push buttons, sends command and receives sensor data from the vehicle through IR communication, and displays information on the LCD screen. Mega 2560 has multiple serial ports, one is wired to the IR module and another one can be connected to the PC for programming and debugging.

1.2. LCD display

Buying a bare LCD panel and driving it directly would be most flexible and cost-effective, but the downside is the implementation may take too long. To reduce the complicity for hardware design and software programming, we chose a 16-by-2 character LCD module with built-in push buttons, as shown in Figure 20. This module comes with an Arduino library that significantly reduces the time needed for prototyping.

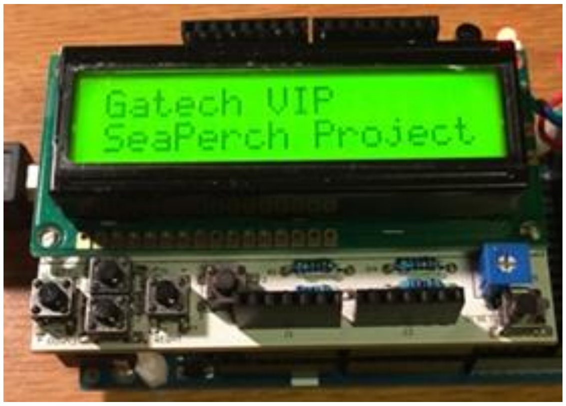

Figure 20. LCD module with push buttons.

1.3. Power supply

There are two voltages required for the hand-hold controller: 11.1V and 5V. Both voltages are needed for the IR communication module, and single 5V is used to power the Arduino board. A DC-DC converter is used to convert 11.1V from the battery to 5V with 3 Amp maximum current. This controller is powered by a rechargeable 11.1V 2200mah Lithium battery from Venom.

1.4. Joystick, push buttons and slide potentiometer

To improve the user experience of controlling the robot, we decided to use an analog joystick as shown in Figure 21b. The output of the joystick is proportional to the angle it is pushed to. Also, there is an integrated pushbutton that can be triggered when the joystick is pressed down.

To intuitively control the depth of the underwater robot, a slide potentiometer as demonstrated in Figure 21a is used. We designed two depth-keeping modes: automatic and manual. The automatic mode uses a PID controller to keep the depth of the vehicle. Under this mode, sliding the potentiometer upwards makes the ROV stay in a shallower depth while sliding down means greater depth. On the contrary, the manual mode allows the users to directly control the speed of the vertically mounted motors. When the slider is placed at the center, the vertical motors will stop spinning. The more the handle is pushed towards the upper half of the potentiometer, the greater thrust the motors will generate to bring the ROV towards the surface. Similarly, the more the handle is pushed towards the lower half, the greater speed the ROV will dive down with. The slide potentiometer we chose has a central detent feature, with which one can easily find the center of the slider. In addition, there is a built-in LED which lights up when the robot is in automatic depth-keeping mode.

Figure 21. a) Slide potentiometer (left) b) analog joystick (right) [1] [2].

1.5. Adapter board

To minimize wiring effort and therefore improve reliability, a customized adapter board is designed as shown in Figure 22. It has the same mechanical layout as the “tail” of the Arduino board, on which it can be easily stacked. The board hubs the wires of the DC-DC converter, battery, IR module, joystick, and slide potentiometer. It also has a simple voltage divider, which enables the Arduino to monitor battery voltage using its analog pins.

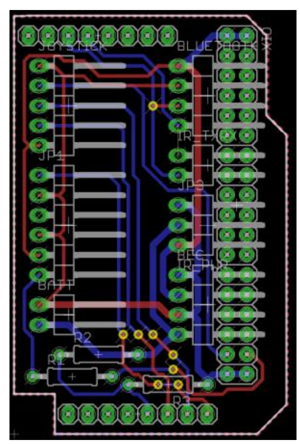

Figure 22. Adapter board for hand-hold controller.

2. Underwater vehicle

The ROV is an underwater platform which is able to execute command from the hand-hold controller and send sensor measurements back. Its electronic design will be discussed as follows.

2.1. Main platform

The main platform of the ROV is Arduino Mega 2560, the same model as the one used in the hand-hold controller. The board has abundant peripheral I/O support, which makes it easier to connect multiple sensors and actuators of the ROV. The board is relatively small in size and can be perfectly fitted on the electronic chassis.

2.2. ESC and ESC holder

The Electronic Speed Controller (ESC) is the device to drive and vary the speed of the motor. The ESC we choose is capable to handle up to 18Amp of current continuous and 50Amp peak.

However, calibration and configuration is needed to make this ESC work properly. Detailed instructions on throttle range calibration and changing programmable configurations can be find from “Motor_driver.pdf” in the GT-MUR package. Shown in Appendix 1, the recommended value of each programmable item is circled in red. This configuration is relatively conservative, users can change the setting to get higher power output or longer battery life.

There are 11 wires connected to each ESC and a customized PCB is made to hub the wires. As shown in Figure 23, all the four ESCs and their peripherals are mounted on the board.

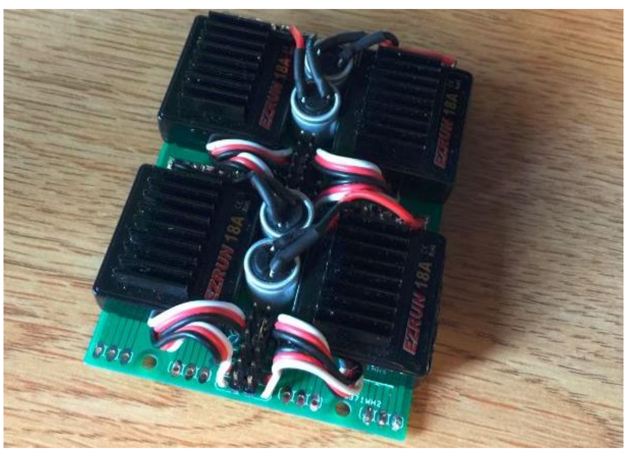

Figure 23. Four ESCs mounted on the customized PCB.

2.3. Power supply

Same as the case of the hand-hold controller, a DC-DC converter is needed to produce 5V voltage for the Arduino and the IR module. The ESCs are directly connected to the battery for higher efficiency. An 11.1V or 7.4V Lithium-Polymer battery with high discharge rate is recommended for GT-MUR. 11.1V 2200mah 20C and 7.4V 200mah 20C batteries from Venom are tested for this early prototype.

2.4. IMU and depth sensor

To get the attitude of the robot, we used a 10DOF (degree of freedom) MEMS (Microelectromechanical systems) IMU (Inertia Measurement Unit) as shown in Figure 24a. The integrated 3-axis gyroscope measures the angular velocity of the ROV, which is then fused with the readings from the 3-axis accelerometer and 3-axis compass. The manufacturer-provided sensor fusion algorithm runs on the Arduino board and the measured attitude is sent back to the hand-hold controller. Moreover, the integrated air pressure and the temperature sensor is able to monitor the environment inside the sealed hull.

As shown in Figure 24b, a hydraulic pressure sensor is used to measure the depth of the ROV. This sensor is able to measure the pressure up to 14 Bar with resolution 0.2mbar. This pressure sensor has gel membrane and stainless steel cap, which protects it from high water pressure.

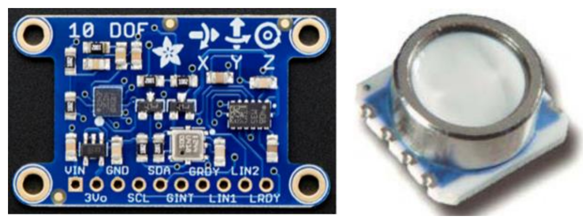

Figure 24. a) 10 DOF MEMS IMU (left)
b) hydraulic pressure sensor (right) [3] [4].

3. IR Communication

In addition to the conventional tethered control, GT-MUR is able to send/receive data to/from the hand-hold controller using infrared communication.

The IR module supports transparent transmission and is directly connected to the serial port of the Arduino. Since the serial pin stays at high voltage level when idle, the modulation activates only when the pin is asserted (at low voltage level). The first three waveforms in Figure 25 shows the original signal from serial port, 38khz carrier and modulated output. To simplify the hardware design, we chose a receiver module which is widely used in household appliances. This module has a photo detector, preamplifier and demodulator integrated into a single package and has immunity to ambient light. As shown in the fourth waveform in Figure 25, the output of the receiver becomes low when modulated signal is detected. It is identical to the original input signal (first waveform in yellow) in spite of time delay, which demonstrates transparent transmission feature.

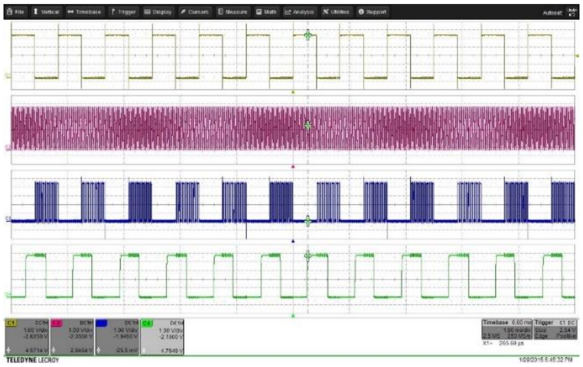

Figure 25. Baseband signal (yellow), carrier signal (pink), modulated signal (blue) and demodulated signal (green).

To achieve higher stability, we used a crystal oscillator to generate clock signal for the microcontroller and a logic gate to handle modulation process. The microcontroller is programmed to generate 38kHz carrier signal which is then fed into a NOR gate for modulation. A high-current MOSFET is adopted to amplify the modulated signal and drive the high-power IR emitter. In addition, users are able to adjust the output power by trimming the potentiometer shown in Figure 26. There are two indicator LEDs on the module. The yellow LED will blink when the module is sending data, while the green one lights up when signal comes in.

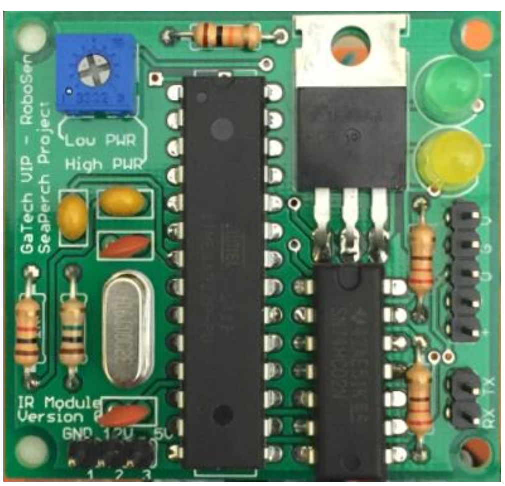

Figure 26. Infrared communication module.

To reduce the cost and simplify the programming. We use ATmega328P microcontroller for the IR module. This is a popular model commonly used in Arduinos and other devices. Without using an expensive programmer, we can plug this chip into an Arduino Uno (or similar models) and program it using Arduino board.

When using a high power IR LED, although it is not constantly turned on, a heatsink is still suggested to avoid potential damage.

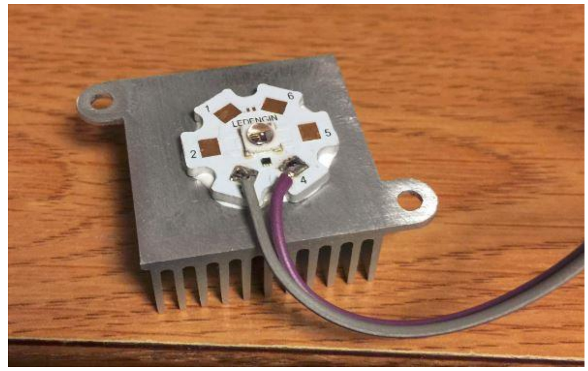

Figure 27. IR LED with heatsink.

Arduino Code Structure

C++ programs were written to be run on the Arduino boards for both hand-hold controller and the ROV. An Object-Oriented design is implemented and its block diagram is shown in Appendix 2. This architecture enables bi-directional communication between the controller and the robot. Sending and receiving data between the hand-hold controller and the robot is described as follows:

1. Sending control data to ROV

The ControlReader class reads control signal from the controller hardware. Then the information is passed to ControlSideByteCoder which converts the control data into a byte stream to be sent over IR. On the robot side, RobotSideByteCoder is responsible for receiving the byte stream from the controller and parse the bytes back into recognizable control data packets. After that, the control data is sent to MotorExecutor class which uses the information to control the motors.

2. Sending ROV data to controller

The RobotDataReader reads robot data (such as orientation, temperature, pressure, etc.) from various sensors carried by the robot. The data is then converted to a stream of bytes by the RobotSideByteCoder and forwarded to the controller side via IR. Upon receiving the robot data, ControllerSideByteCoder converts the bytes into readable packets and sends them to the LCDDisplayer for display.

3. Arduino libraries

As a convenience to the users, we packaged the Arduino libraries of GT-MUR into three folders. "PID_v1" is used to hold the depth of the vehicle, "SeaPerch" contains the supportive files we wrote for GT-MUR, and "vendor" has all the manufacturer-provided libraries.

To use the above libraries, simply copy the three folders to the libraries directory of Arduino, thus "Arduino_installation_directory/libraries/".

GT-MUR open source package

All the design files can be find in the GT-MUR open source package available at our GitHub Repository:

https://github.com/GeorgiaTechSystemResearch/GT_MUR_Open_Source_Package

The mechanical design files are in dwg and dxf format which can be edited using AutoCAD. The PCBs are designed and can be modified with EAGLE CAD. Moreover, GERBER files are also provided, users can directly send them to PCB manufacturer.

In addition to the CAD drawings, Arduino codes and this instruction manual, other resources are also provided in this package. Part list contains the components selected for the underwater vehicle and the hand-hold controller. For user's reference, datasheets and manuals of the major components are also attached.

References

[1] Bourns, "PTL Series Slide Potentiometer w/LED," PTL60-10G1-103B2 datasheet, [Revised Sept. 2011].

[2] "Thumb Joystick", Sparkfun.com, 2016. [Online]. Available: https://www.sparkfun.com/products/9032. [Accessed: 28- Dec- 2016].

[3] Kevin Townsend, *Adafruit 10-DOF IMU Breakout*, Adafruit Industries, 2016.

[4] Measurement Specialties, "MS5803-14BA Miniature 14 bar Module," MS5803-14BA datasheet, Sept 7. 2012.

Appendix 1: Recommended Programmable Value for the ESCs

Programmab le	Programmable Value								
	1	2	3	4	5	6	7	8	9
Basic Items									
1.Running Mode	Forward with Brake	*Forward/Reverse with Brake*	Rock Crawler						
2.Drag Brake Force	0%	5%	*10%*	20%	40%	60%	80%	100%	
3.Low Voltage Cut-Off Threshold	Non-Protection	2.6V /Cell	2.8V /Cell	*3.0V /Cell*	3.2V /Cell	3.4V /Cell			
4.Start Mode (Punch)	Level1 (Soft)	Level2	L3	L4	L5	L6	*L7*	L8	L9 (Very Aggresive)
Advanced Items									
5.Maximum Brake Force	25%	50%	75%	*100%*					
6.Maximum Reverse Force	25%	*50%*	75%	100%					
7.Initial Brake Force	*= Drag Brake Force*	0%	20%	40%					
8.Neutral Range	6% (Narrow)	*9% (Normal)*	12% (Wide)						
9.Timing	0.00 °	3.75 °	7.50 °	11.25 °	*15.00 °*	18.75 °	22.50°	26.25°	
10.Over-heat Protection	*Enable*	Disable							

Appendix 2: Block Diagram of the Arduino Codes