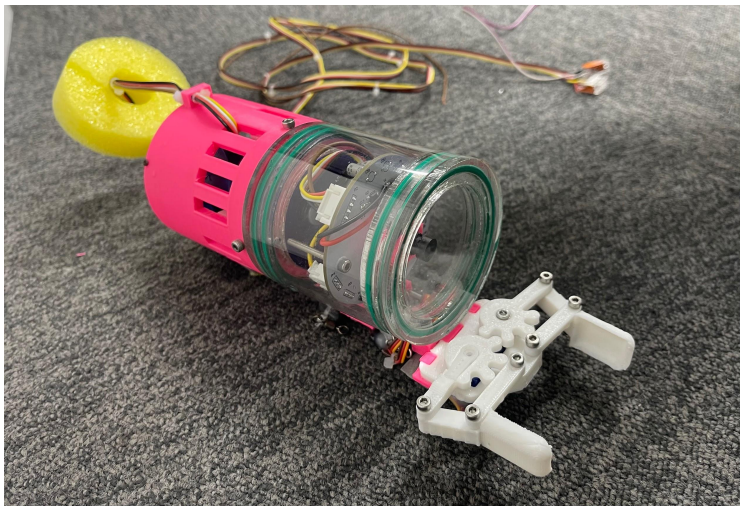
EE3070 Design Project
Final Report

Object Retrieval with Micro-ROV



AU YEUNG Hing Hui
55729793

Abstract

In this project, a Micro-ROV will be developed as a low-cost telerobotic submarine built with the goal of exploring small and unknown underwater areas. At this stage the initial prototype development was limited, starting from the design to the manufacturing, and testing the performance of the prototype which included tightness, stability and mobility. Expectations for the future to be further developed include the addition of camera features, improved performance, other sensor features and so on. The prototype in this project will be made using 3D-Printing technology because this is an alternative way for fabrication the Micro-Class underwater ROV so it can make various parts with shapes and sizes that are in accordance with the desired design.

Introduction

According to the data from National Geographic, the ocean makes up 71% of the Earth's surface. However, more than 80% of our ocean is unmapped, unobserved and unexplored. Engineers around the world are designing underwater robotics (aka Remotely Operated Vehicle or ROV) to support work to feed the needs for the future. Due to the complexity of the ocean environment, there are lots of small and unknown areas in the ocean where ROVs cannot arrive and then accomplish some related tasks. For example, retrieving the sample from the hydrothermal vents for research and monitoring seaweed and chitons in a tide seabed pool. Therefore, a Mini-Class ROV or Micro-Class ROV is needed in these scenarios.

In the 2019 MATE ROV competition, there is a task to deploy the Micro-ROV from the primary ROV to inspect the inside of the drainpipe for indicators of possible dam failure. For the task implementation, the primary ROV needs to deploy the Micro-ROV into a 13.5-inch drainpipe to retrieve the data sample, which the drainpipe is around 2 meters long and is not transparent. At the end of the drainpipe, a data sample is placed inside at the end of the drainpipe. The participant should design a Micro-ROV to retrieve the data sample without destroying the drainpipe. Plus, it is a similar task in 2020 MATE ROV Explorer-class competition. A Micro-ROV possibly will be required for other tasks in future MATE ROV competition. It is expected that there is a basic prototype of the Micro-Class ROV for further development.

Therefore, in this project, a Micro-ROV would be developed. And a gripper that would be mounted in the Micro-ROV to achieve the object retrieval. Moreover, to minimize the size of the Micro-ROV, only one thruster and one micro FPV camera would be installed in the Micro-ROV. More details would be explained in other sections.

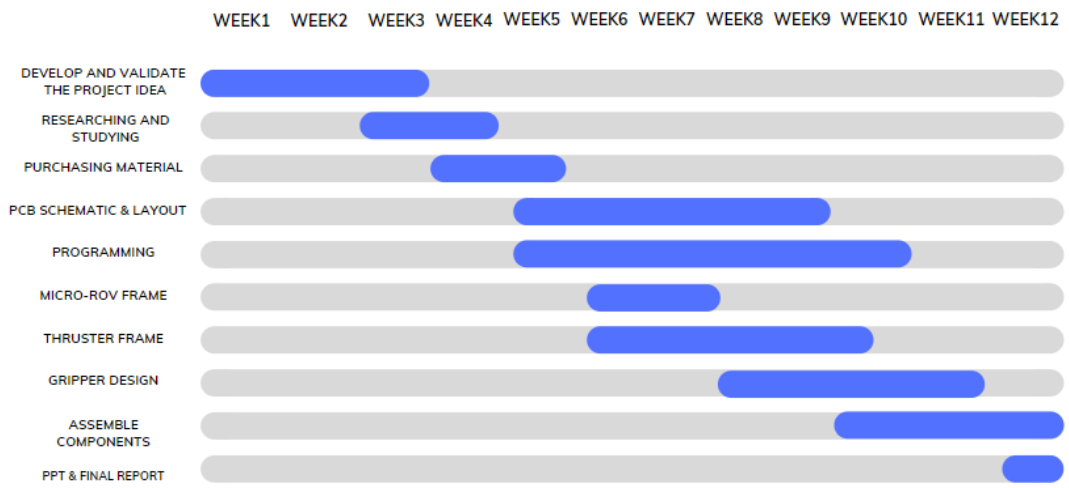
Objectives

- Make a PCB Board that integrates with the main system of the Primary ROV
- Customize the Micro-ROV frame
- Design a 3D-Print Servo Gripper and a thruster frame
- Use CAN Bus protocol for communication between Micro-ROV and main ROV

Success Criteria

- Able to send the view back to the ground station
- Able to retrieve the object by using the gripper while only very limited space is available
- Provide two communication modes:
    a) UART, or Universal Asynchronous Receiver-Transmitter
    b) CAN Bus Communication
- Provide flexibility of replacing assemblies

Project Schedule



Technical Background

1.1 SG90 Servo by Towerpro

SG90 Servo is a micro servo which is a tiny and lightweight servo with high output power as it commonly operates at 5V. And it consists of a small electric motor, a potentiometer, embedded control board and a gearbox. The position of the output shaft is constantly measured by the internal potentiometer and compared with the target position set by the controller. Plus, the torque of the SG90 Servo is 1.8kg/cm. That means the motor can pull a weight of 1.8kg when it is suspended at a distance of 1 cm.

Specifications:

- Weight: 9g
- Dimension: 22.2 x 11.8 x 31 mm approx
- Stall torque: 1.8 kgf cm
- Operating speed: 0.1 s/60 degree
- Operating voltage: 4.8 V (~5V)
- Dead band width: 10 us
- Temperature range: 0°C - 55 °C

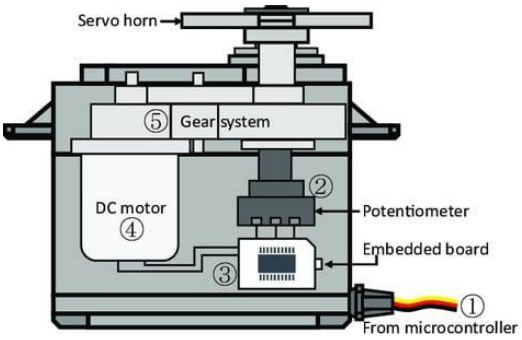


Figure1.1 The virtual view of a typical DC servo motor

Servo motors are controlled by sending a PWM (pulse-width modulation) signal to the signal line of the servo. The width of the pulses determines the position of the output shaft. With a pulse width of 1.5 milliseconds (ms), the servo will move to the neutral position (90 degrees). The min (0 degrees) and max (180 degrees) positions correspond to a pulse width of 1 ms and 2 ms respectively.

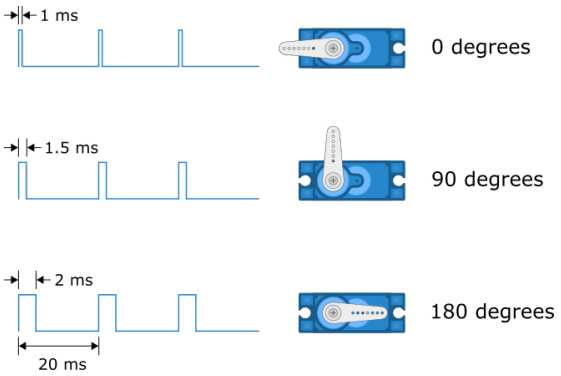


Figure 1.2 The Relationship between the Servo Position and the Length of the Pulse

1.2 Watertight Enclosure (Chemical Explanation)

Acrylic Liquid Adhesive is a special glue with a chemical substance called “Chloroform” and its molecular formula is $CHCl_3$. It appears as a colorless liquid and it is a corrosive and irritant chemical substance, which is soluble in water and is corrosive to metals and tissue. it is mainly used as a herbicide.

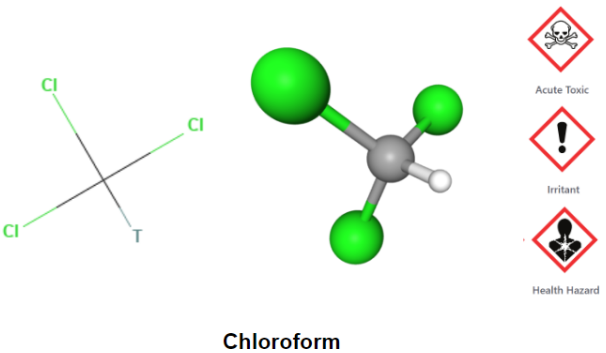


Figure1.3 The Chemical Structure of Chloroform(Trichloromethane)

Polymethyl methacrylate (PMMA) is also known as acrylic or acrylic glass and its chemical formula is $(C_5O_2H_8)n$. PMMA is a tough, highly transparent material with excellent resistance to ultraviolet radiation and weathering. PMMA is made by using the MMA Monomer (methyl methacrylate) and acrylic acid along with a catalyst, by the bulk polymerization process.

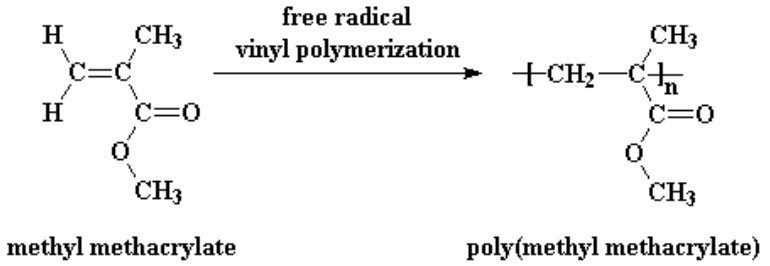


Figure1.4 The polymerization of MMA Monomer into PMMA Ploymer

Since Polymethyl methacrylate (PMMA) is soluble in chloroform, chloroform becomes a bonding chemical that bonds acrylic to acrylic surfaces in seconds.

1.3 Can Bus Communication

CAN stands for "Controller Area Network" and it is an serial communication protocol defined by the ISO 11898 standards. Those standards define how communication happens, how wiring is configured and how messages are constructed, among other things. Collectively, this system is referred to as a CAN bus. The CAN bus is a broadcast type of bus. This means that all nodes can receive all data packets in the transmission line. There is no way to send a message to just a specific node. All nodes will invariably pick up all traffic. The CAN hardware, however, provides local filtering so that each node may react only on the interesting messages.

And there are two CAN standards: Standard CAN or Extended CAN. The standard CAN, with the length of the Identifier in the Arbitration Field to eleven (11) bits, allows bit rates up to 1 Mbit/s and payloads up to 8 byte per frame, but a newly introduced format, the Extended CAN, allows higher bit rates and higher payloads with no more than twenty-nine (29) bits in the Identifier.

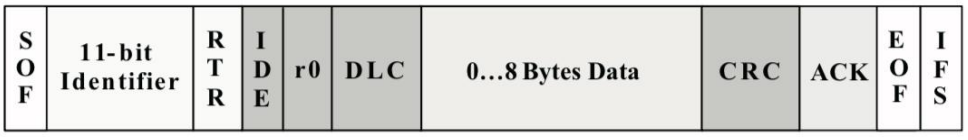


Figure 1.5 Standard CAN : 11-Bit Identifier

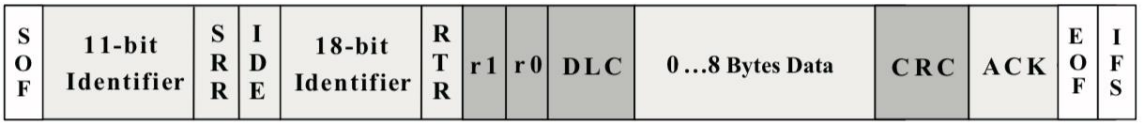


Figure 1.6 Extended CAN : 29-Bit Identifier

With the help of differential voltage, it would be determined how 0 and 1 are transmitted through the CAN bus. The above figure is the voltage graph that shows the voltage level of CAN low and CAN high. In CAN terminology, logic 1 is said to be recessive while logic 0 is dominant. When CAN high line and CAN low line are applied with 2.5 volts, then the actual differential voltage would be zero volt.

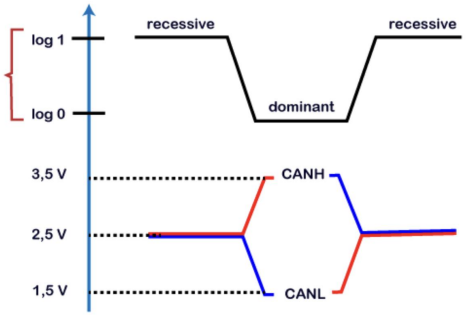


Figure 1.7 CAN Bus Signal Level

Implementation

2.1 System Block Diagram

There are two system block diagrams showing two communication methods between a Primary ROV and a Micro-ROV.

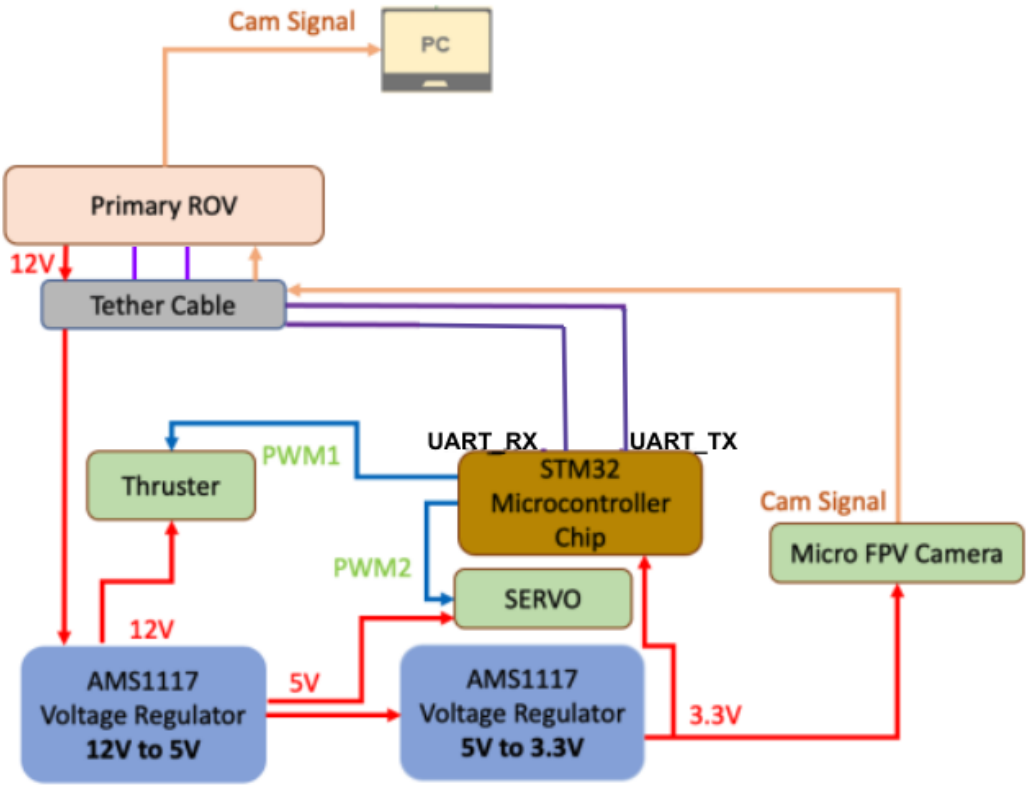


Figure 5.1 The System Block Diagram for UART Communication

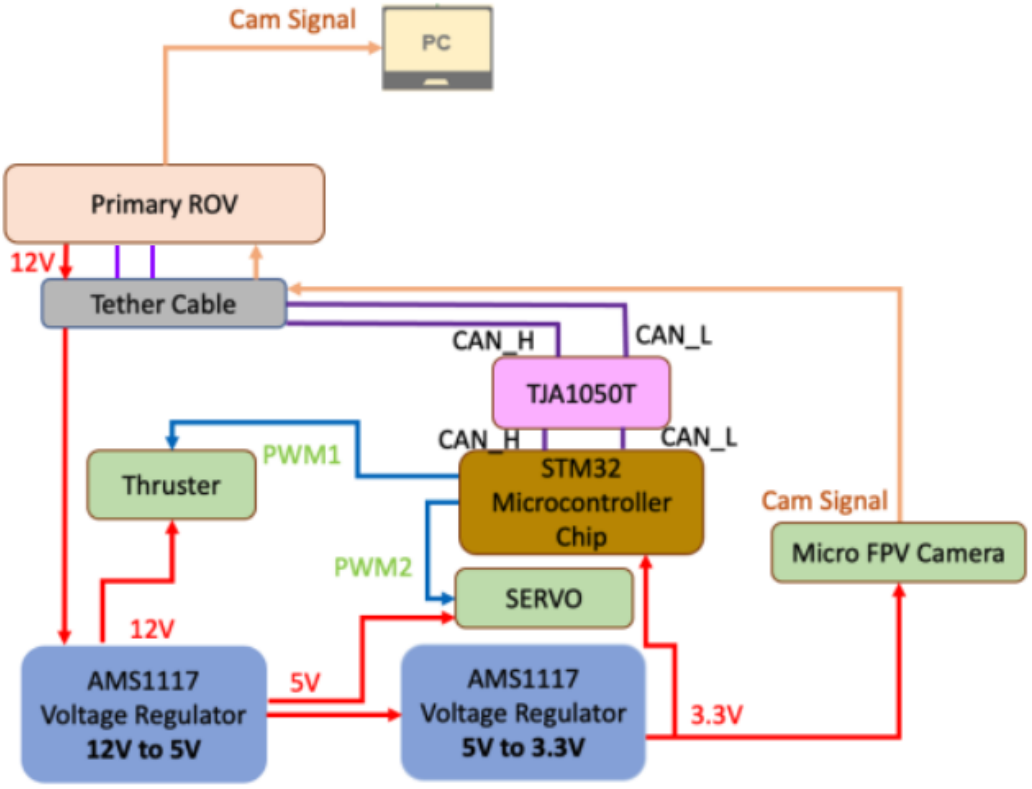


Figure 5.2 The System Block Diagram for CAN Bus Communication

2.2 Hardware design

2.21 Power

The Micro-ROV would take 12V input from the Primary ROV. As the other components like the Micro FPV Camera and SG90 servo require low voltage level, therefore, the AMS1117 series voltage regulators would be used in the Micro-ROV to convert 5V and 3.3V.

2.22 IC

STM32F103C8T6, a 32-bit microcontroller integrated circuits by STMicroelectronics, would be used and then all the actuators and the thruster would be controlled by the microcontroller. Plus, as mentioned previously, a CAN transceiver chip, TJA1050 Integrated Circuit, would provide an interface between the CAN protocol controller and the physical bus.

2.23 I/O

There are IO pins reserved on the STM32F103C8T6 microcontroller for the UART communication and the CAN bus communication. For the control part, the STM32 microcontroller would directly send the PWM signals to control the thruster and also servo motor through the GPIO Pins.

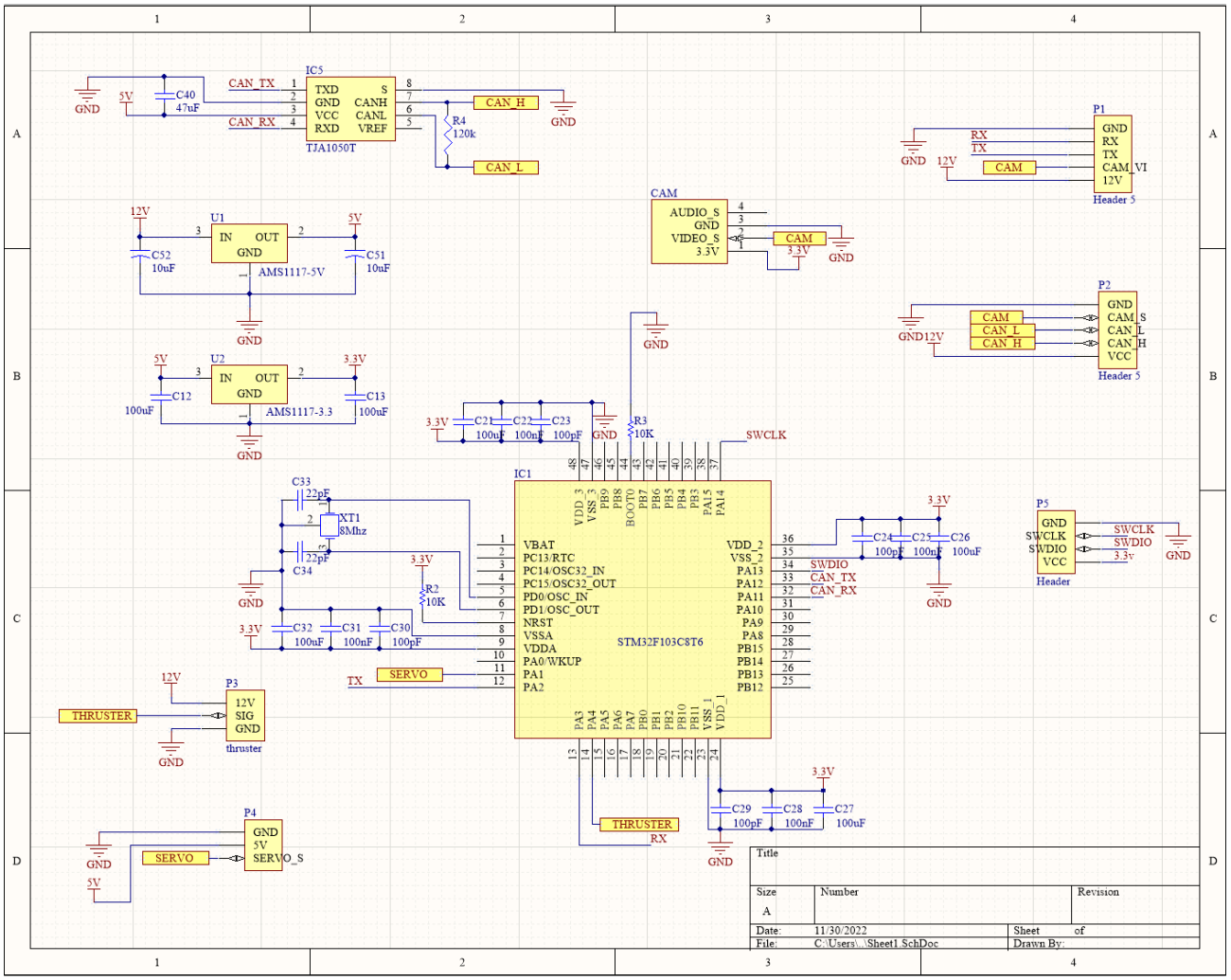


Figure 5.3 PCB Schematic diagram

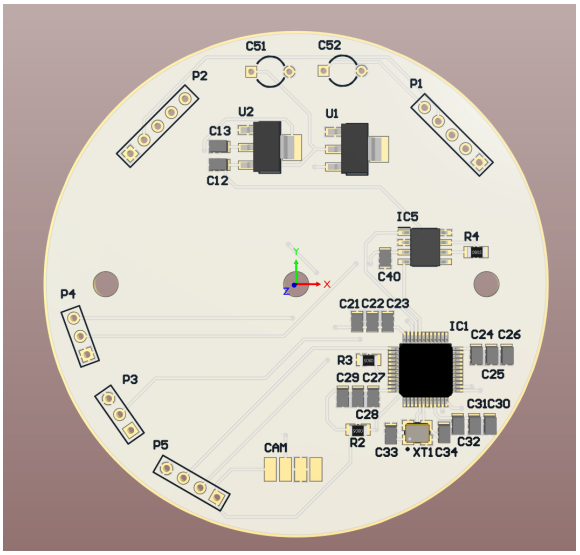


Figure 5.4 The front view of 3D PCB model

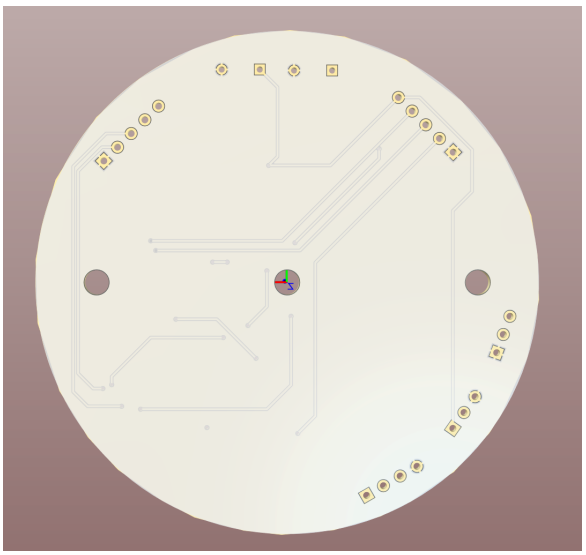


Figure 5.5 The back view of 3D PCB model

2.3 CAD design

2.31 The Mini Servo Gripper

The design of mini servo gripper design is based on the design of the servo gripper from thingiverse webside. And to install the mini servo gripper onto the Micro-ROV, two stainless steel bars would be used to hold the mini servo gripper while these two bars would thrust into the holes reserved in the gripper installer.

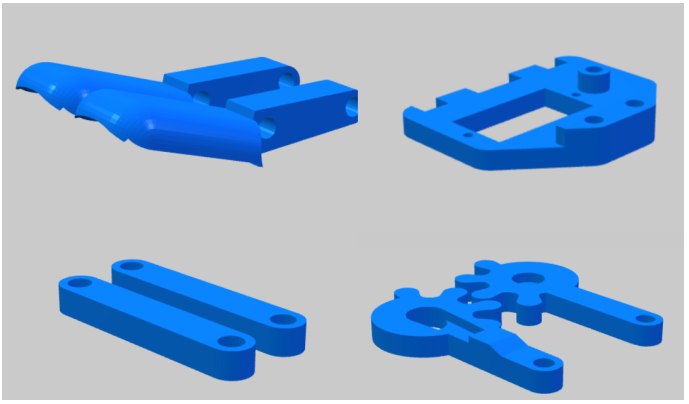


Figure 5.6 Mini Servo Gripper Design from www.thingiverse.com

The following design is for installing the Mini Servo Gripper on the Micro-ROV.

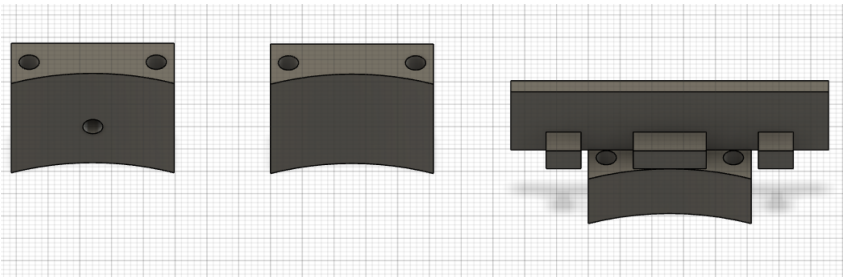


Figure 5.7 The Design of Gripper Installer

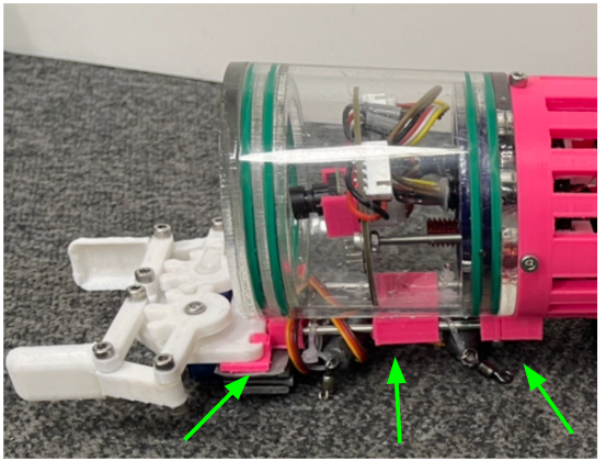


Figure 5.8 The Picture shows the Application of Gripper Installer

2.32 Thruster Frame

Regarding the thruster frame, it is a ‘pit’ on the top of the frame. So that the main frame holding the electronic part could be inserted into the thruster frame and then use the screws to connect these two parts. Plus, these are the space reserved for the thruster ESC(Electronic Speed Controller) on the frame.

And as shown in the 3D model design(on the bottom right hand side), the ‘L shape’ components could fix the positions of the thruster wires in case the wires hamper the thruster’s operation.

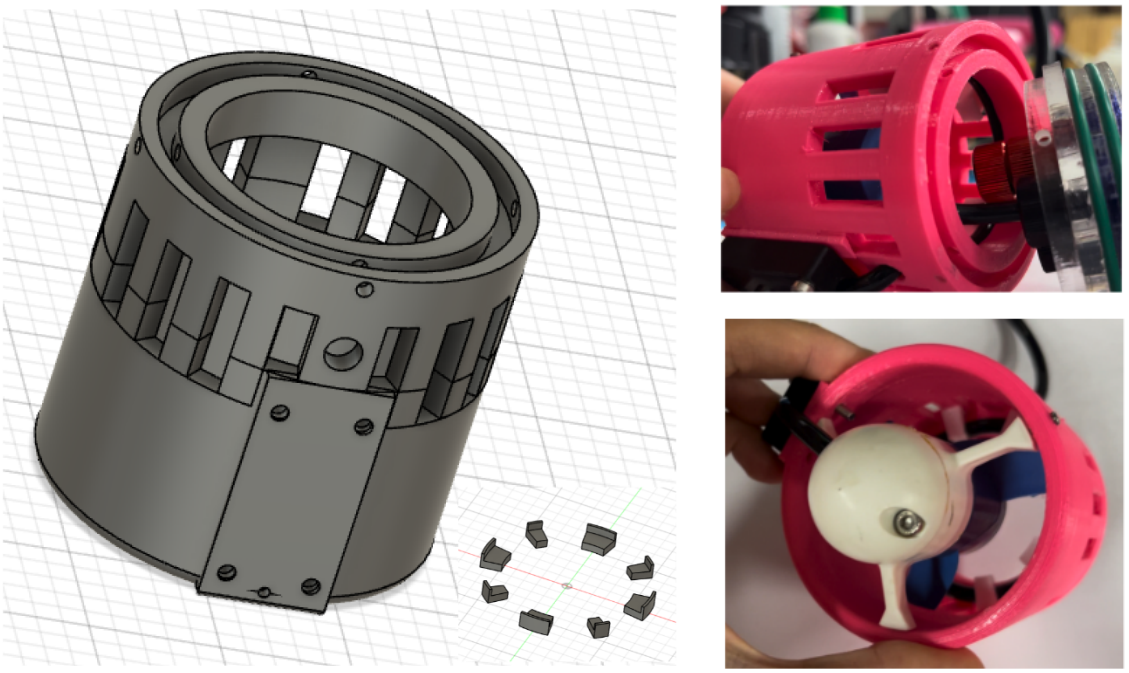


Figure 5.9 The Design of the Thruster Frame

2.33 Camera Mount & Cable Penetrator Locker

In the PCB, it is a hole (with diameter 3mm) reserved for mounting the camera. Therefore, for the camera mount, there is space for M3 screw to lock the camera on the PCB. And based on the design, the camera can slide into the camera mount and the camera wires would go through the cavity on the bottom of the camera mount.

And for the cable penetrator locker, it would fix the positions of the three cable penetrators installed the cover of the main frame. And the two holes with diameter 3mm are for hold the PCB board.

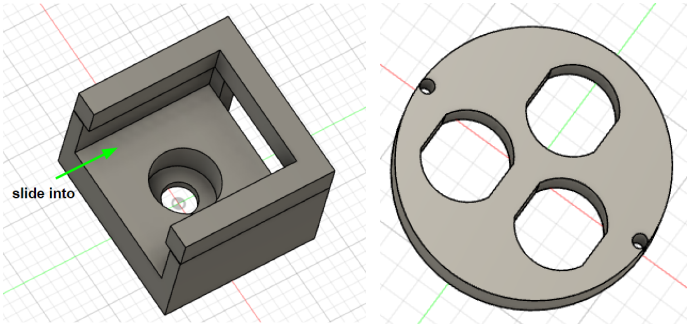


Figure 5.10 The Design of the Camera Mount and Cable Penetrator Locker

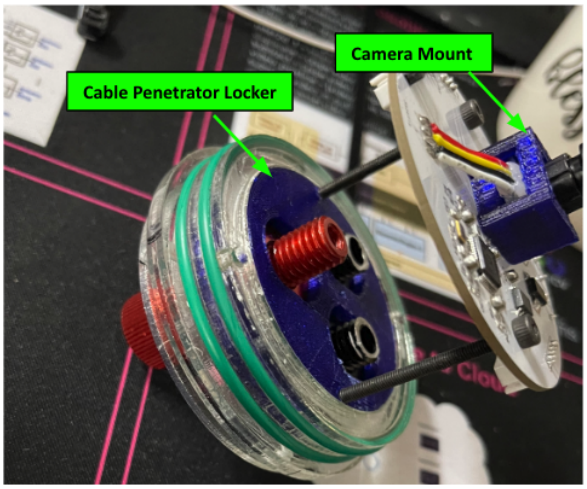


Figure 5.11 The Picture shows the Application of the Camera Mount and Cable Penetrator Locker

2.4 Software design

2.41 UART communication

The following pictures show the configuration for UART communication in STM32CubeMX IDE. Under the Configuration tab, there are some parameters needed to be set up. Word Length set to 8 Bits and the baud rate to 115200 Bits/s, Parity to None and Stop Bits to 1 as shown below.

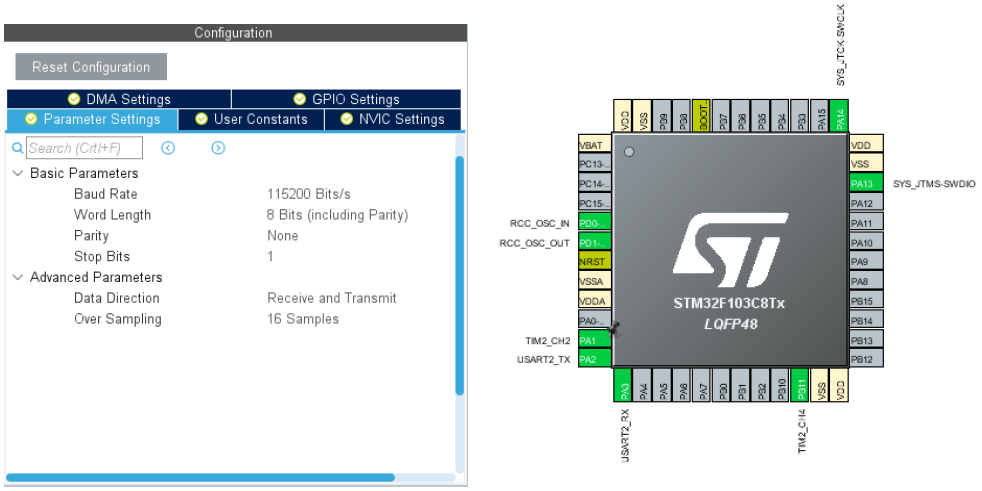


Figure 5.12 The Configuration for UART Communication

The output:

Based on the program(details on the Appendices section), while the STM32 microcontroller on the PCB receives the data sent from the another STM32 Black Pill module, the Micro-ROV would take the correspond action to move forward by changing the value of the 'htim2.Instance->CCR2', which means the channal 4 of the timer2. This channal was assigned to control the thruster.

In the following picture, it shows the the STM32 microcontroller on the PCB received a digit '2' and then assign to a variable 'RxBuffer'. Then, a PWM signal would be sent to the thruster to command move backward.

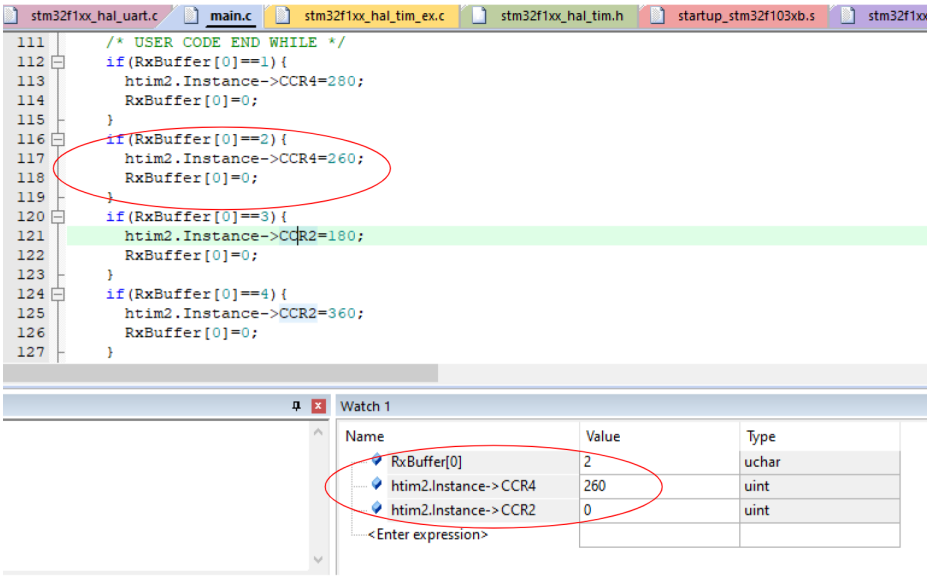


Figure 5.13 The Output for the UART Communication

2.42 Can Bus Communication

The following pictures show the configuration for UART communication in STM32CubeMX IDE. To configure for the CAN bus communication, the Prescaler is set to '4', Time Quanta to for Bit Segment1 to 15 times and Segment 2 to 2 times. Moreover, the Operating Mode is NORMAL Mode.

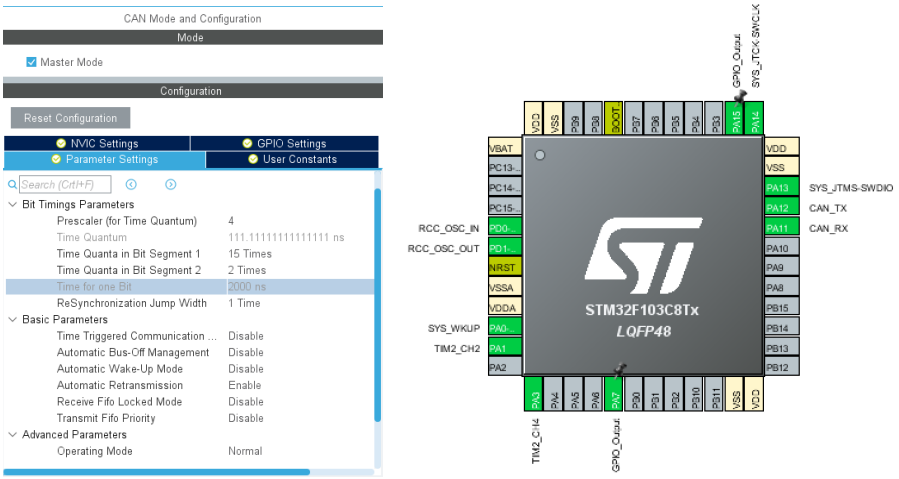


Figure 5.14 The Output for the CAN Bus Communication

The Output:
It is similar to the UART communication, the STM32 microcontroller on the PCB would control the gripper and thruster based on the received data. As the channal 2 of the timer2 is used to command the gripper to be closed, therefore, the following output shows the PWM signal was successfully sent to the 'htim2.Instance->CCR2', which is the channal 2 of the timer2.

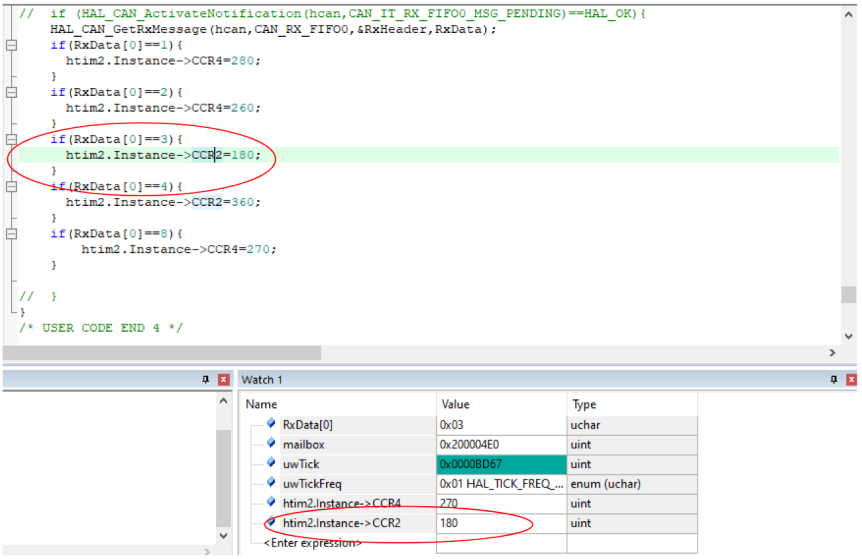


Figure 5.14 The Output for the CAN Bus Communication

Testing

For the testing, one extra STM32 Black Pill module was used and acted as the Primary ROV to send the signals to control the Micro-ROV.

At the beginning, the thruster was fully powered up and then it led to the dropped voltage. Once the whole system turn on or the thruster stopped immediately, the current at that moment would be extremely high. The current could be increased to 2.1 amperes. And this increased current burned the STM32 microcontroller chip and also the voltage regulator several times. So, the AMS1117 voltage regulator was replaced with a power converter module JW5015.

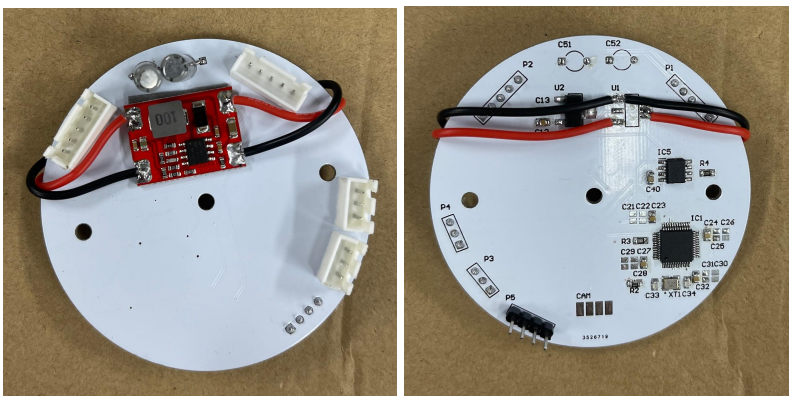


Figure 6.1 The PCB that replaced with a New Module

For the second testing, the power supply would provide 12V to the Micro-ROV. As it was not fully powered up the thruster, therefore, the whole system only took 0.11A to turn on all the components including the thruster, mini servo gripper and the micro FPV camera (as shown in the Figure 3.1). And all the components were under control successfully.

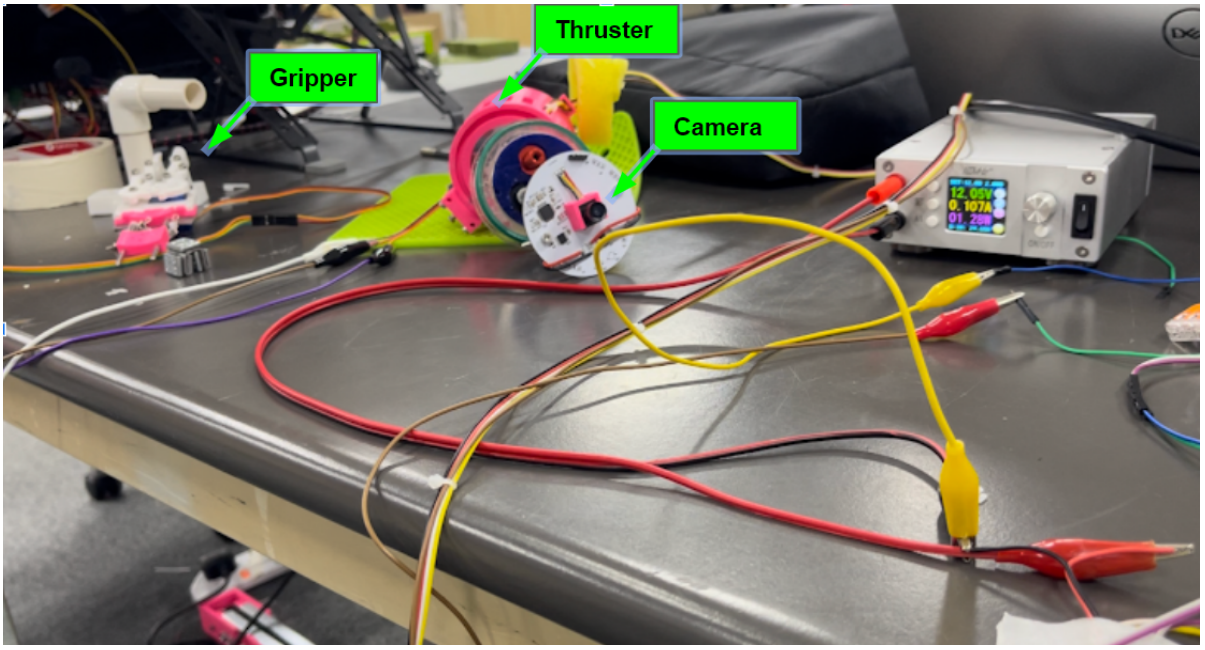


Figure 6.2 The Second Testing of the Micro-ROV

After assembling all the components, it was a buoyancy test to add weight and float into the Micro-ROV.

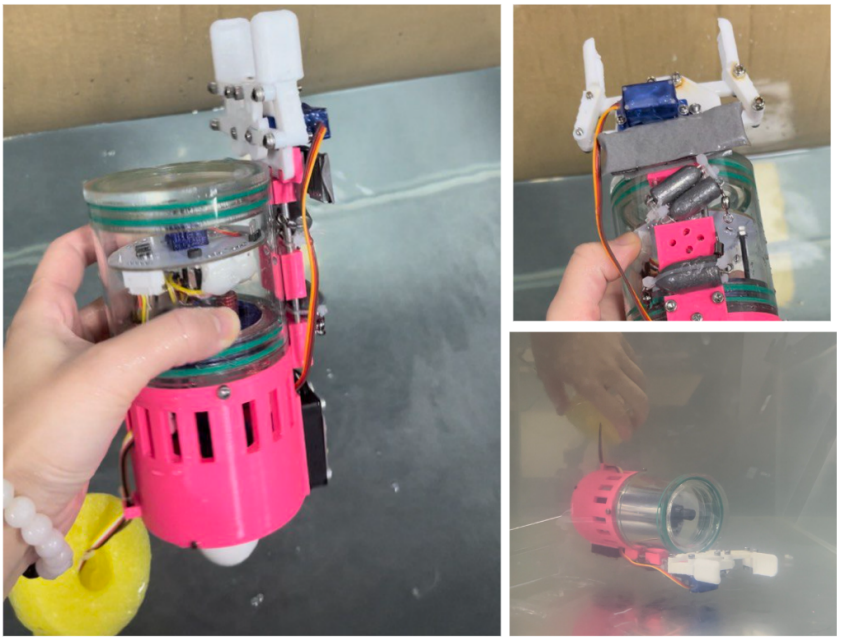


Figure 6.3 The Buoyancy Testing of the Micro-ROV

To test the functionalities of the Micro-ROV, the vehicle was placed into the water tank to test it.

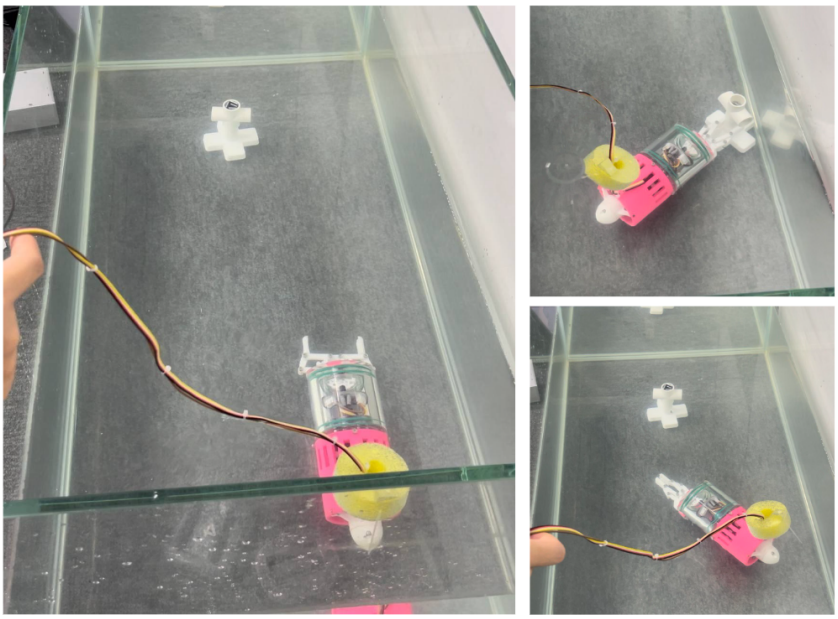


Figure 6.4 The Final Testing of the Micro-ROV

Result and Future Work

The first part that should be improved in the future is regarding the movement of the Micro-ROV. During the testing, the Micro-ROV only could move forward and backward. And it even could not move forward or backward straightly as the water was fluctuating. Therefore, to improve it, a small or mini motor with a propeller is needed to be installed into the Micro-ROV and then it could provide more movement like Surge and Sway.

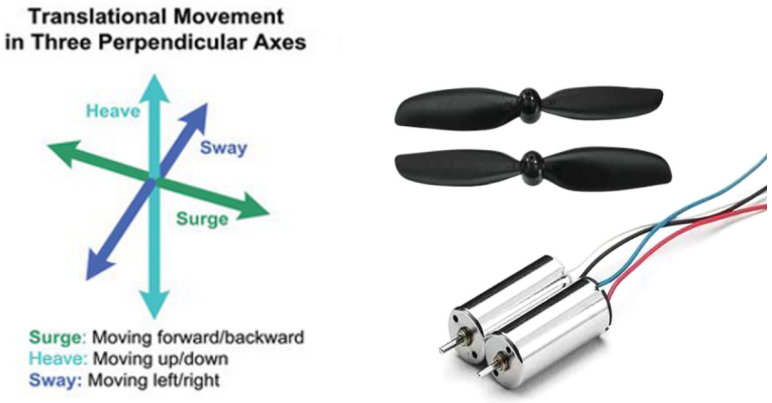


Figure 7.1 The ROV Movement and Mini Motor with Propeller

Meanwhile, as mentioned previously, for the power part, AMS1117 voltage regulator is not suitable to be used in this project as it is a series of low dropout three-terminal regulators with a dropout of 1.3V at 1A load current. Plus, its maximum output current is 1.4 amperes.

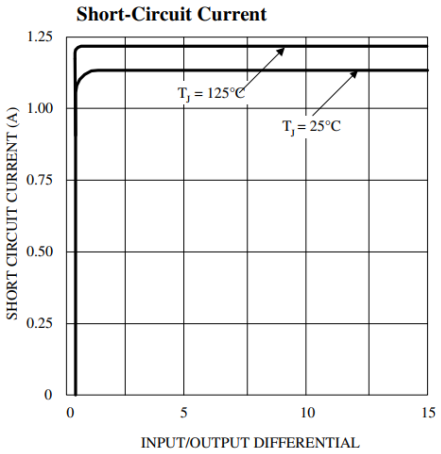


Figure 7.2 Short-Circuit Current for the AMS1117

That is the reason why if the thruster stopped immediately(fully powered up), the AMS1117 voltage regulator cannot handle this increased current. To solved this, a JW5015 power converter module was be used by replacing the AMS1177. A JW5015 power converter module is a switching regulator to delivers 2A of continuous output current with two integrated N-Channel MOSFETs. It guarantees robustness with short-circuit protection, start-up current run-away protection and input under voltage lockout.

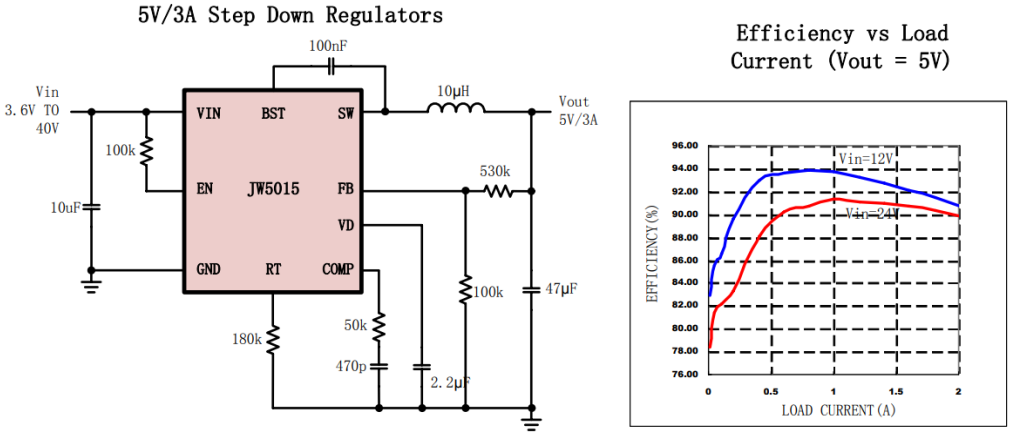
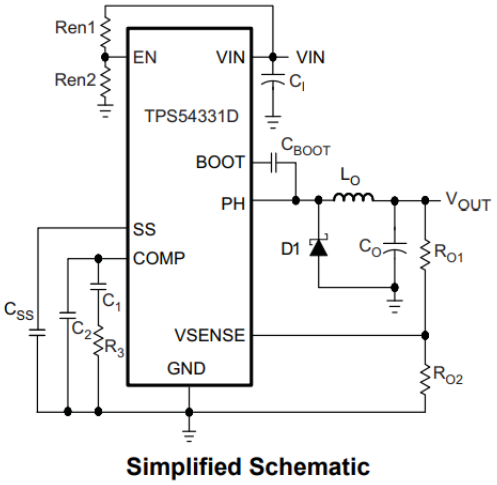


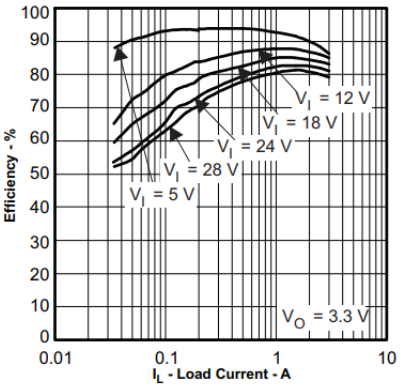
Figure 7.2 The Circuit Diagram for the JW5015 and its Relationship between the Load and Efficiency

However, the JW5015 power converter module is still not ideal for this project. So for the near future, we should consider using other voltage regulators, like TPS54331 Step Down Converter.

The TPS54331 integrated circuit is a converter that integrates a low RDS(on) high-side MOSFET. it provides an overvoltage transient protection circuit to limit voltage overshoots during startup and transient conditions.



Simplified Schematic



TPS54331 (D Package) Efficiency

Figure 7.3 The Circuit Diagram for the TPS54331 and its Relationship between the Load Current and Efficiency

References

C. M. Staff, “Everything you need to know about acrylic (PMMA),” *Everything You Need To Know About Acrylic (PMMA)*. [Online]. Available: https://www.creativemechanisms.com/blog/injection-mold-3d-print-cnc-acrylic-plastic-pmma. [Accessed: 01-Dec-2022].

“2,2-dichloropropionic acid,” *National Center for Biotechnology Information. PubChem Compound Database*. [Online]. Available: https://pubchem.ncbi.nlm.nih.gov/compound/2_2-Dichloropropionic-acid. [Accessed: 01-Dec-2022].

“Poly(methyl methacrylate): 9011-14-7,” *ChemicalBook*. [Online]. Available: https://www.chemicalbook.com/ChemicalProductProperty_EN_CB1221699.htm. [Accessed: 01-Dec-2022].

Benne, “How to control servo motors with Arduino (3 examples),” *Makerguides.com*, 02-Mar-2022. [Online]. Available: https://www.makerguides.com/servo-arduino-tutorial/. [Accessed: 01-Dec-2022].

“SG90 datasheet, equivalent, micro servo.,” *SG90 Servo Datasheet pdf - Micro Servo. Equivalent, Catalog*. [Online]. Available: https://datasheetspdf.com/pdf/791970/TowerPro/SG90/1. [Accessed: 01-Dec-2022].

Thingiverse.com, “Robotarm - mini servo gripper by hkucs-makerlab,” *Thingiverse*. [Online]. Available: https://www.thingiverse.com/thing:4394894. [Accessed: 01-Dec-2022].

“Introduction to the Controller Area Network (CAN) (rev. B).” [Online]. Available: https://www.ti.com/lit/an/sloa101b/sloa101b.pdf. [Accessed: 01-Dec-2022].

Appendix

The Program for UART Communication

```
void Thruster_int(void){
  htim2.Instance->CCR4=180;
  HAL_Delay(1000);
  htim2.Instance->CCR4=270;
  HAL_Delay(1000);
}

void HAL_UART_RXCpltCallback(UART_HandleTypeDef*huart){
  HAL_UART_Receive_IT(&huart2, (uint8_t *)&RxBuffer,1);
}
```

```
/* Private variables ---------- */
TIM_HandleTypeDef htim2;
UART_HandleTypeDef huart2;

/* USER CODE BEGIN PV */
uint8_t RxBuffer[1];
```

```
/* Private function prototypes ---------- */
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_TIM2_Init(void);
static void MX_USART2_UART_Init(void);
/* USER CODE BEGIN PFP */
void Thruster_int(void);
void HAL_UART_RXCpltCallback(UART_HandleTypeDef*huart);
/* USER CODE END PFP */
```

```
/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
  /* USER CODE END WHILE */
  if(RxBuffer[0]==1){
    htim2.Instance->CCR4=280;
    RxBuffer[0]=-1;
  }
  if(RxBuffer[0]==2){
    htim2.Instance->CCR4=260;
    RxBuffer[0]=-1;
  }
  if(RxBuffer[0]==3){
    htim2.Instance->CCR2=180;
    RxBuffer[0]=-1;
  }
  if(RxBuffer[0]==4){
    htim2.Instance->CCR2=360;
    RxBuffer[0]=-1;
  }
  if(RxBuffer[0]==8){
    htim2.Instance->CCR4=270;
    RxBuffer[0]=-1;
  }
```

The Program for CAN Bus Communication

```
/* Private variables ---------------------
CAN_HandleTypeDef hcan;

TIM_HandleTypeDef htim2;

/* USER CODE BEGIN PV */
CAN_FilterTypeDef Scanfilter;
CAN_RxHeaderTypeDef RxHeader;
CAN_TxHeaderTypeDef TxHeader;

uint32_t mailbox;
uint8_t TxData[1];
uint8_t RxData[1];
uint8_t TEMP;
/* USER CODE END PV */

/* Private function prototypes -----------
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_CAN_Init(void);
static void MX_TIM2_Init(void);
/* USER CODE BEGIN PFP */
void Thruster_int(void);
/* USER CODE END PFP */

 /* Initialize all configured peripherals */
 MX_GPIO_Init();
 MX_CAN_Init();
 MX_TIM2_Init();
 /* USER CODE BEGIN 2 */

 HAL_CAN_ActivateNotification(&hcan,CAN_IT_RX_FIFO0_MSG_PENDING);
 HAL_CAN_Start(&hcan);
 HAL_TIM_PWM_Start(&htim2,TIM_CHANNEL_2);
 HAL_TIM_PWM_Start(&htim2,TIM_CHANNEL_4);
 Thruster_int();

void HAL_CAN_RxFifo0MsgPendingCallback(CAN_HandleTypeDef *hcan){
//  if (HAL_CAN_ActivateNotification(hcan,CAN_IT_RX_FIFO0_MSG_PENDING)==HAL_OK){
    HAL_CAN_GetRxMessage(hcan,CAN_RX_FIFO0,&RxHeader,RxData);
    if(RxData[0]==1){
      htim2.Instance->CCR4=280;
    }
    if(RxData[0]==2){
      htim2.Instance->CCR4=260;
    }
    if(RxData[0]==3){
      htim2.Instance->CCR2=180;
    }
    if(RxData[0]==4){
      htim2.Instance->CCR2=360;
    }
    if(RxData[0]==8){
        htim2.Instance->CCR4=270;
    }

//  }
}
```