CVHS ROV Manual

Oregon State University
College of Engineering

Table of Contents

Table of Figures

Introduction

Primarily designed as a self-contained simplified low cost 4 thruster remotely operated vehicle (ROV) suitable for basic inspection tasks or as a STEAM learning platform the CVHS ROV project is an opensource vehicle developed by the Robotic Decision-Making Laboratory RDML at Oregon State University. https://research.engr.oregonstate.edu/rdml/home

Software Architecture

The system is comprised of two Raspberry Pi 4's running the Raspbian version of Debian, one on the ROV at a fixed IP address of 192.168.2.3 and the control unit which has a fixed IP address of 192.168.2.2. Any other computer networked into the system could also talk to either Pi if connected to the network and set to the 192.168.2.X and 255.255.255.0 subnet mask. Systemd is utilized to automatically start and monitor several scripts on both units. These scripts are started with the files that end in '"something".service'. The commands:

> *sudo systemd start* ***name****.servvice* (starts the associated script, if not already running)
>
> *sudo systemd enable* ***name****.service* (Sets the associated script to start on boot)
>
> *sudo systemd disable* ***name****.service (*Sets the associated script to not start automatically)

name should be replaced with the script name of the associated script as seen in Figure 1.

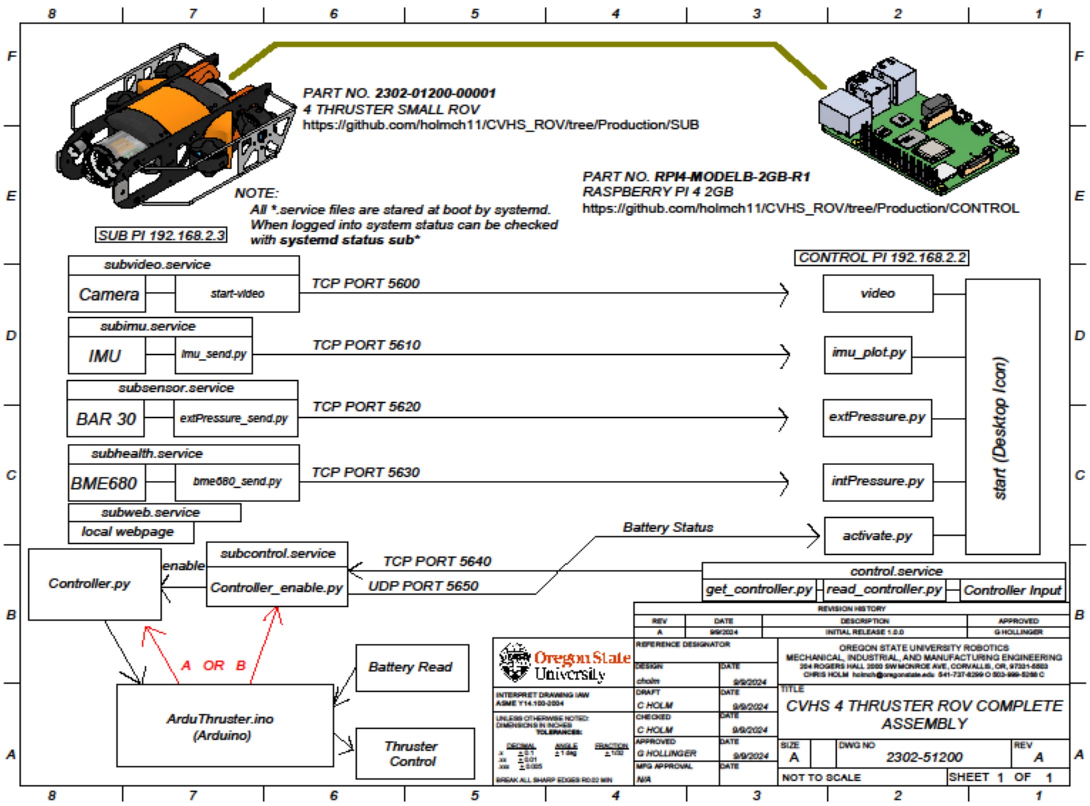
The general communication and flow of the software can be seen in 2302-51200 on Github:

https://github.com/holmch11/CVHS_ROV/tree/Production

or below in Figure 1.

Generally, sensor and camera data is sent up on TCP ports to be read and displayed by the control pi and controller data is sent down to the ROV and once enabled can be enacted on using an Arduino to send pwm signals to the correct thrusters. The thrusters in paired up/down and forward/reverse. In each pair one spins clockwise and the other counterclockwise. You will hear the thrusters make beeping noises at startup and when an enable is registered which indicates the thruster has been correctly enabled and is ready to accept input. The Arduino currently also reads the battery voltage (through a voltage divider) and outputs that data to a serial port. Version B of the PI interface board also allows the battery to be read directly by the PI but no software has been written to support that yet. It would clean up the serial port opening/closing requirements if implemented.

Figure 1: Code Map Showing overall system architecture.



To run correctly the BME680 python scripts are run in a virtual environment which lives in the env folder and is initiated by a bash script (bme680_send). The external pressure sensor is run using a forked version of the Blue Robotics MS5837-python repository: https://github.com/holmch11/ms5837-python.

There is a simple webpage hosted on the sub side that was intended to show which processes are up and running correctly. There are errors here that need to be corrected before it achieves full functionality, however this is a nice to have feature and by no means is required for operation. If desired the entire webpage infrastructure could be removed. The page is hosted using nginx. Ssh keys are shared between the two Raspberry Pis so if on the control Pi one should be able to open a terminal window and issue the command:

ssh sub

This will allow you to navigate through the pi on the ROV and check things like what services are running, and restart them if crashed. It would also allow you to gracefully shutdown the ROV Pi as there are no automatic shutdown scripts/measures in place. Repeated power loss without issuing:

sudo shutdown now *#* command will result eventually in a corrupted SD card on either system.

Basic navigation in the terminal can be accomplished with standard linux commands like ***cd*** (change dir), ***ls*** (short listing of what is in the current directory) or ***ll*** (more detailed listing of directory contents).

Quick Start Guide

1. Remove 8 M4 screws from the Top Float Clamp Assembly 2302-01203
2. Remove Teflon cord from rear bottle end cap (figure 2).
3. Install red adapter into the pressure release valve which will pull the valve open.
4. Using Blue Robotics wedge tool only (metal tools will damage bottle), insert the wedge into the opening where the Teflon cord had been and rotate it
   wedging the end cap out from the bottle. Then use your hands to push evenly on both sides of the endcap to remove the electronics tray from the bottle.
5. Plug in the battery balance plug (5 pin JST XH) into the corresponding right angle plug on the Arduino Interface Board 2302-30302 (Figure 3).
6. Plug in the Main EC5 Blue Battery Plug in the corresponding plug on the vehicle terminal, the vehicle should turn on and start making noise at this point.
7. If necessary clean o-rings and o-ring surfaces with isopropyl alcohol (rubbing alcohol)
8. Apply a **thin** layer of o-ring silicone grease to the o-rings, and spread with a gloved hand, lightly stretching the o-ring as you work your way around will allow you to feel any cuts or issues with the o-ring.
9. Install o-rings and install end cap into the bottle, aligning bottle with endcap
10. Install Teflon cord into the end cap.
11. Attach handheld vacuum pump onto red adapter and pull vacuum to 10 inHg.
12. Monitor Pressure for ~15min and make sure the vacuum does not drop below 9inHg. If it does, you need to open the bottle and clean and or replace the o-rings, there is a leak in the bottle.
13. IF the pressure test is successful you can unscrew the red adapter with the pump still attached until it is loose and then remove red cap and vacuum pump, and then make sure the blue pressure relief valve is still tight. If done correctly the internal pressure should still be under vacuum.
14. Install the 8 M4 screws and tighten the Top Float Clamp Assembly and the vehicle is now ready for deployment.
15. Start the topside script (figure 5) and verify that all sensors are reporting normally (figure 6), internal pressure is under vacuum, and humidity is low. If the humidity is high before it goes into the water you need to replace the desiccant. Monitor this internal pressure and humidity as well as the battery voltage throughout the deployment and recover the vehicle if the battery voltage drops below 15.2V or if the internal humidity spikes or vacuum suddenly drops, a slow drop in vacuum is considered normal.

Quick Start Figures

Figure 2: End-Cap Locking Teflon Cord

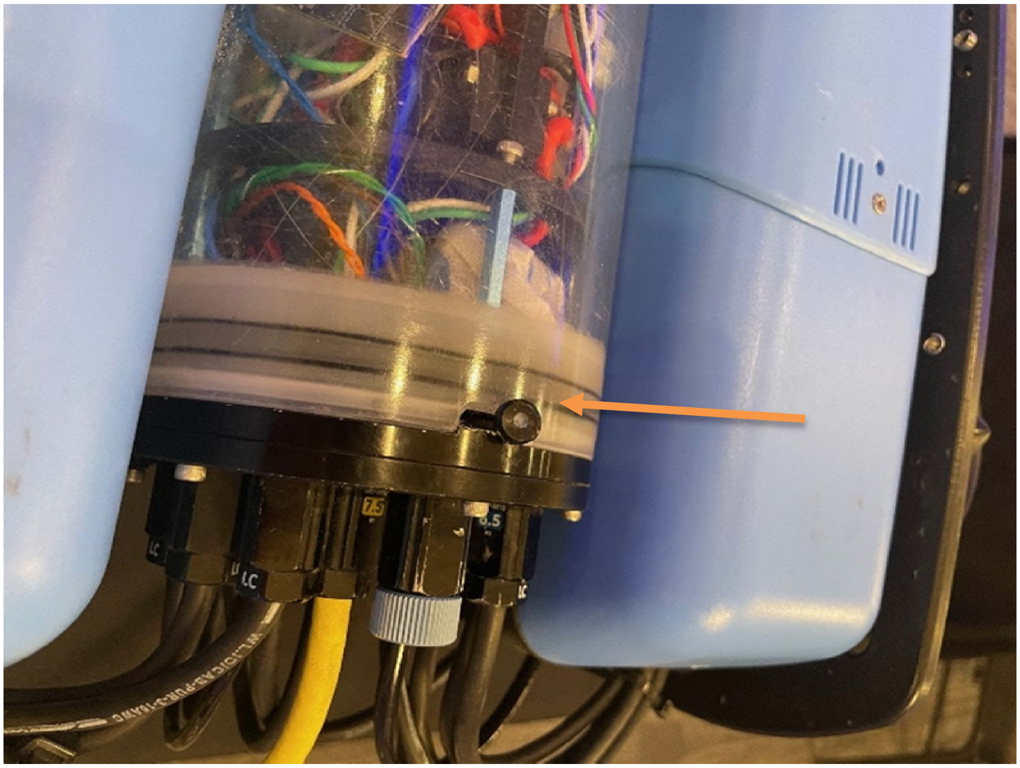
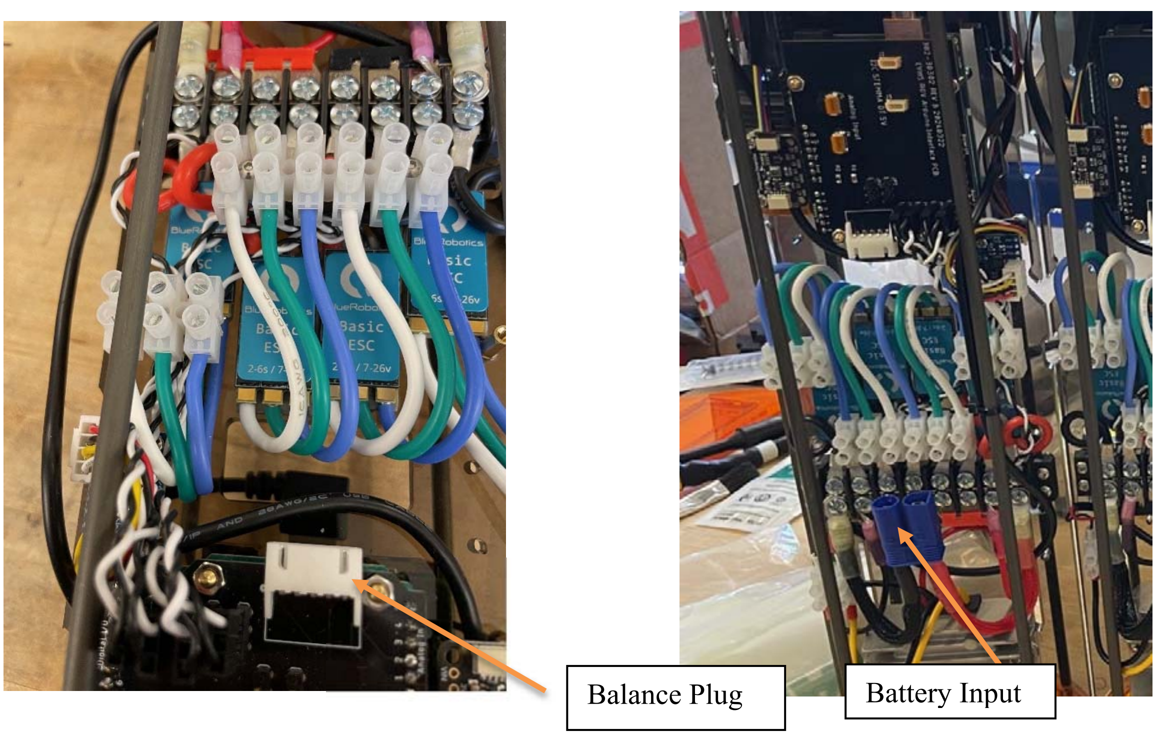


Figure 3: Battery Balance Plug Connector



Top Side Set-up

Before starting the ROV using the Quick Start, the Top Side Pi should be connected to a monitor using the micro-HDMI 0 (closest to USB-C), then connected to RJ-45 Ethernet Connecctor directly to ethernet switch (figure 4), and the usb-c power supply plugged in as well as usb accessories (keyboard, mouse, Logitech controller). The ROV Tether should also be plugged into the ethernet switch, technically the tether should be able to work when directly plugged into the topside pi RJ45, and for some this does work, but oddly for others it does not.

Figure 4: Ethernet Switch Connections: Top Side Pie and ROV Tether

Checking ROV Communications

If working correctly you should be able to open a terminal window and ssh into the sub (figure 5). If this does not work and ping 192.168.2.3 does not work, then for some reason there is no connection between the sub and topside pi. Check that both Pi's are on and your switch is plugged in, and all connections are tight.

Starting Top Side Communication

If the control pi is connected properly to both a Logitech controller and the ROV, you should be able to double click on the ROV Control Icon on the desktop (figure 6), a window will pop up that tells you this is executable and asks how you want to execute it (figure 6). You can choose execute or execute in terminal, the latter will open more windows but also give you more information if something goes wrong and doesn't start. You can also choose open and it will just show you what the icon script does in a text file.

Components of correctly started top side script

When correctly started what pops up is a collection of separate scripts. The most prominent is the video feed from the ROV camera, this can be restarted by closing the windows and then opening ~/CVHS_CONTROL/act/video. Video is a bash script that receives a gstreamer feed from the ROV. Several terminals and graphs also pop up, one for the imu data (imu_plot.py), one for the external pressure sensor(extPressure.py), and one for the internal pressure sensor(intPressure.py). The last script that is started creates a TKinter button for activating the control and also shows the batter output. This is activate.py This is shown in figure 7 below, however in this image internal pressure didn't start correctly so it is missing.

Figure 5: Using a Terminal Window to access the

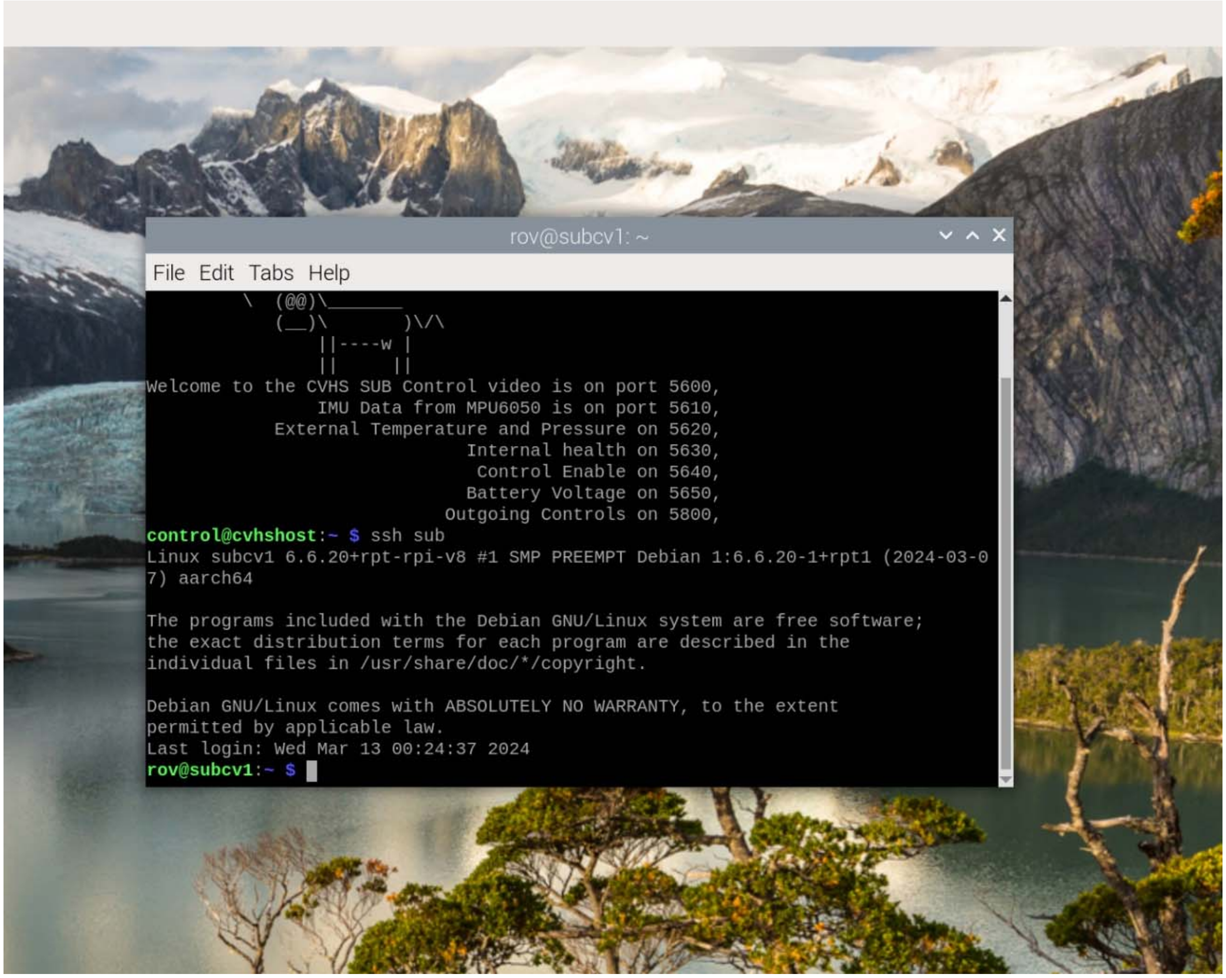


Figure 6: Using Desktop icon to run start script

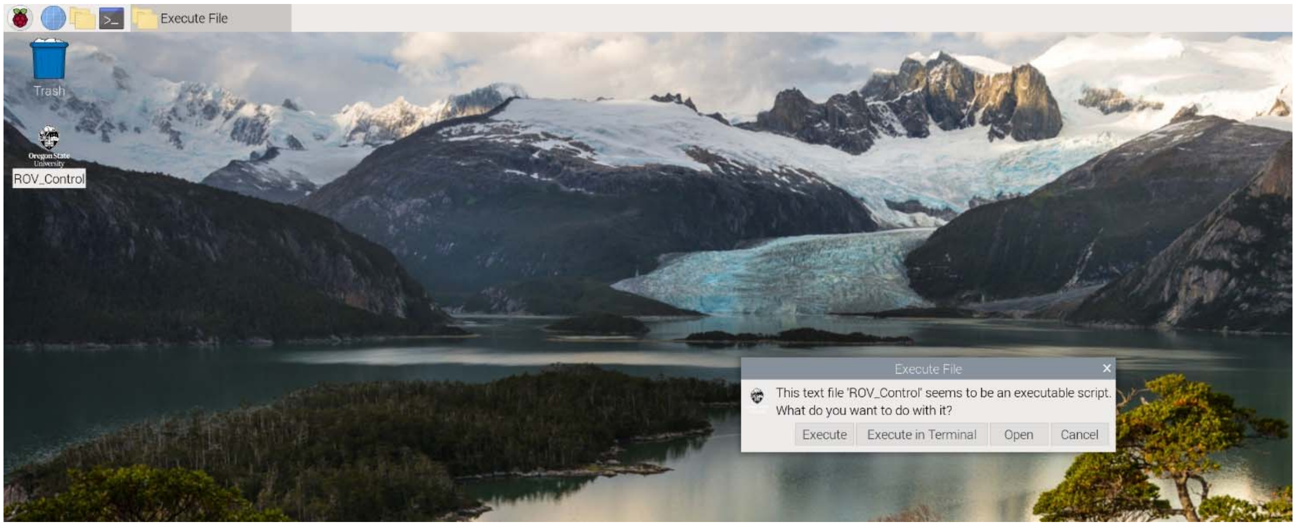
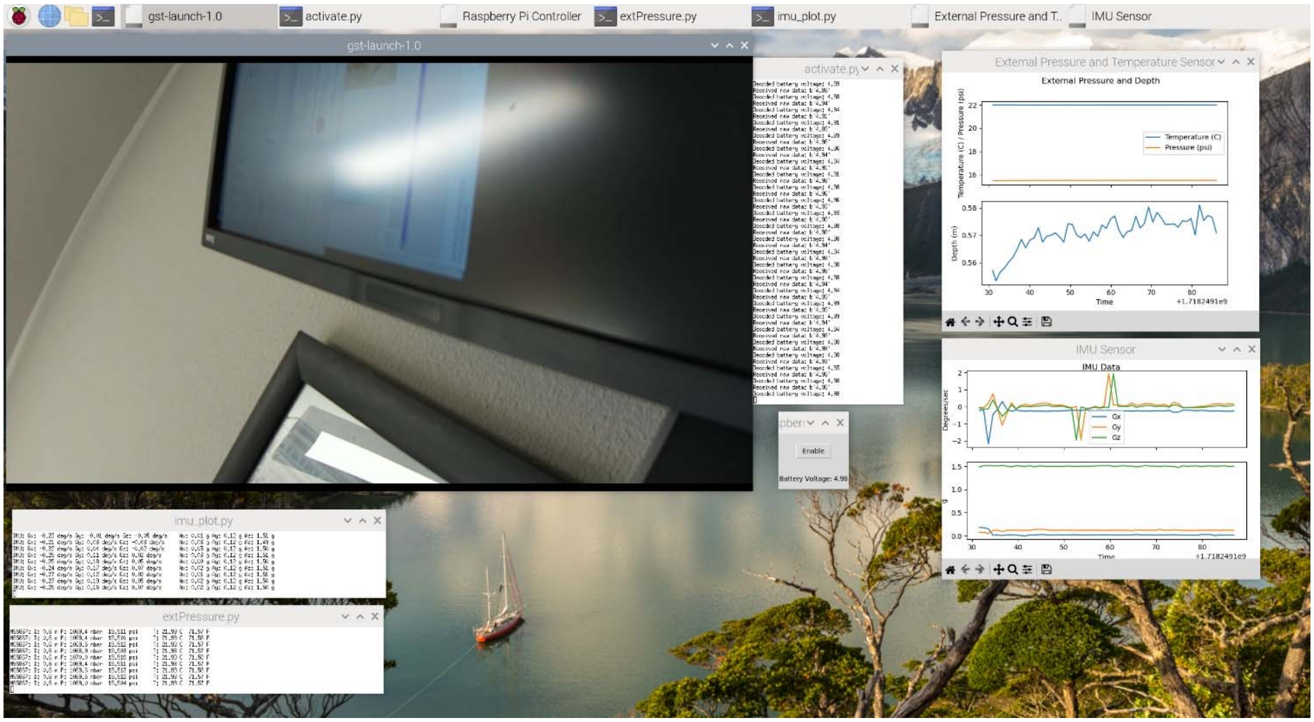


Figure 7: Correctly Started ROV Software



Helpful Blue Robotics Guides

https://bluerobotics.com/learn/wetlink-penetrator-installation-guide/
https://bluerobotics.com/learn/pressure-relief-valve-installation-and-usage/
https://bluerobotics.com/learn/battery-info/
https://bluerobotics.com/learn/watertight-enclosure-wte-assembly-new/

College of Engineering
Oregon State University
101 Covell Hall
Corvallis, OR 97331

engineering.oregonstate.edu