UUV Testbed README

Fall 2022

Altium Schematic and PCB Files:

There are three separate PCB files: the Power, Main, and IMU Sensor board. The Power PCB file includes the Thruster schematic and Battery Management Schematic. The Main board PCB includes the Main board schematic and the IMU sensor board PCB includes the IMU sensor board schematic.

For these designs, the system is powered by a 22.2V Lithium ion battery, and only powers the IMU with the battery management IC disconnected. The IMU communicates with a Raspberry Pi 4+ (code is included for USB camera and IMU). With the current design, the Raspberry Pi 4+ only works when powered by a wall outlet.

What works with current hardware design:

* Power to IMU

What will be updated:

* Providing safe power to thruster(s) and Raspberry Pi 4+ with battery management IC (instead of just IMU)
* Add CANBUS communication between thruster(s) and IMU
* Hole sizes for the battery and thruster PCB connections
* Fuse selection
* Surface mount connections for the mezzanine connectors on the IMU board and Main board
* Method of attaching the Power board and Raspberry Pi 4+ onto the Main board

Software

Setting up Raspberry PI

* Do not use Ubuntu Core, Use Ubuntu Desktop. The current OS is Ubuntu 22.04.1 LTS with Desktop Image
* To install miniconda use Miniconda3-py39\_4.9.2-Linux-aarch64.sh . DO NOT UPDATE Conda version
* Install ROS2 HUMBLE on Ubuntu Deskop 22.0.4[. Instructions here](https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html)
  + Install Desktop, not BareBones or Development Tools

Install USB V4L Camera Driver for Dell Camera

**# Required Libraries**

* Colcon
* Github
* v4l2loopback
* ROS2 Desktop (Humble)
* [Instructions](https://index.ros.org/r/usb_cam/) to install USB V4L Camera Driver for Dell Camera
  + Before following instructions install the following:
    - Install Colcon.

**sudo apt install python3-colcon-common-extensions**

* + - Install git hub:

**sudo apt install git**

* + Prepare Termainal by Exceucting in the Command Line:

**source /opt/ros/humble/setup.bash to start ROS2 in Ubuntu terminal**

* + Use Quickstart:

**sudo apt install ros-humble-usb-cam**

* Install v4l2loopback 
  + This is to initialize /dev/video0, which is required to run the camera in an image viewer. If the raspberry pi does not recognize the camera through this input, it will not work in ROS2
* Restart Raspberry Pi4. Turn power off. Plug camera in to correct bus as shown in step below. Turn power on.
* Plug Camera USB to BUS 2 USB. Only recognizes video 0 on Bus 002 of the Raspberry Pi. Camera must be connected to BUS 2 and no other bus. Use the command **lsusb** to verify connectivity and that it is connected to the right bus. The camera will be labeled as DELL. Camera must be recongized for the next step to work
* Open terminal and execute:

**source /opt/ros/humble/setup.bash**

\*when opening new terminal window you must always execute **source /opt/ros/humble/setup.bash** to start ROS2 in Ubuntu terminal

* To view image use code below. A live stream of the camera will appear as well as show that the specifications of the image on the terminal. Press CTRL+C to exit

**ros2 launch usb\_cam demo\_launch.py**

## Install IMU DRIVER On ROS2

**# Bugs**

**Bugs: Currently there is a bug in the code where the SMBus is not being read in my\_python\_node.py I am not sure why this is happening if I2C is enabled and SMBus library is added. This happens when you execute the IMU ROS2 node**

**# Required Libraries**

* python 3.7 (the version is important!)
* raspi-config
* I2c-tools
* setuptools==58.2.0 (the version is important!)
* smbus

## install python 3.7 or less (setuptools will not work if running on higher python)

$ sudo apt install python3.7

## install raspi-config to configure I2C

$ sudo apt-get update

$ sudo apt-get install raspi-config

## Enable I2C by opening Raspberry Pi configuration tool using the code below to open this tool

$ sudo raspi-config

## Install I2C tools to see if IMU is detected

$ sudo apt install i2c-tools

## Install setup tools to run node

$ pip install setuptools==58.2.0

## Install smbus

### $ pip install [smbus](https://www.devmanuals.net/install/ubuntu/ubuntu-12-04-lts-precise-pangolin/install-python-smbus.html)

# The following instructions are to initialize a ros2 node. Code should be already properly zipped in the Github. The code in the GitHub file should be saved in Desktop. However, if you need to create a ros 2 node here are the instructions to do so. Use this [website](https://roboticsbackend.com/write-minimal-ros2-python-node/) for help. Reminder that

##The code below sets up the main folder for node and the src folder inside the main folder

$ cd ~/ros2\_ws/src

## Initialize ros 2 node package

$ ros2 pkg create my\_robot\_tutorials --build-type ament\_python

## Go to my\_robot\_tutorials folder which was created when you built the package. Create a new python file called my\_python\_node.py

$ cd my\_robot\_tutorials/my\_robot\_tutorials

$ touch my\_python\_node.py

## Open the setup.py file. Edit the python file at the line that says entry\_points. Edit the line so it to looks like the code below declaring the name of the node action.

entry\_points={

'console\_scripts': [

'minimal\_python\_node = my\_robot\_tutorials.my\_python\_node:main'

],

...

## Open the package.xml file. Edit the python file to include the license name shown below:

<license>Apache-2.0</license>

## In the package.xml file add the following two lines in the order shown here. They will be included above the other depend code lines.

<**depend**>rclpy</**depend**>

**<buildtool**\_depend>ament\_python</buildtool\_depend>

## Go back to terminal and run code below to compile package in ros2\_ws . You will need to do this anytime there are any changes to the ros2\_ws folder and the python codes inside

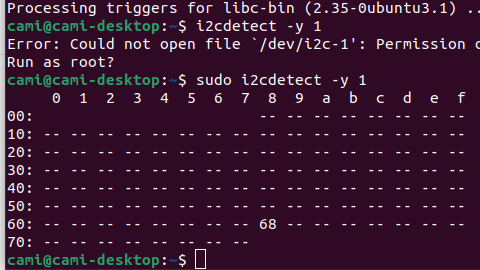
$ cd ros2\_ws

$ colcon build --packages-select my\_robot\_tutorials

# The remaining steps below are to implement the ROS2 node for the IMU. It will go over how to recompile the package if necessary.

## Run the following code to check IMU connection. The output should look like the image below or else there may be problems with your IMU connection:

$ sudo i2cdetect -y 1

****

## Recompile Code in terminal (if necessary)

$ cd ros2\_ws

$ colcon build --packages-select my\_robot\_tutorials

## Open new terminal window and execute ros2 node. :

$ source /opt/ros/humble/setup.bash

$. ~/ros2\_ws/install/setup.bash

$ ros2 run my\_robot\_tutorials minimal\_python\_node

**\*\*\*Currently there is a bug in the code where the SMBus is not being read in my\_python\_node.py I am not sure why this is happening if I2C is enabled and SMBus library is added.**