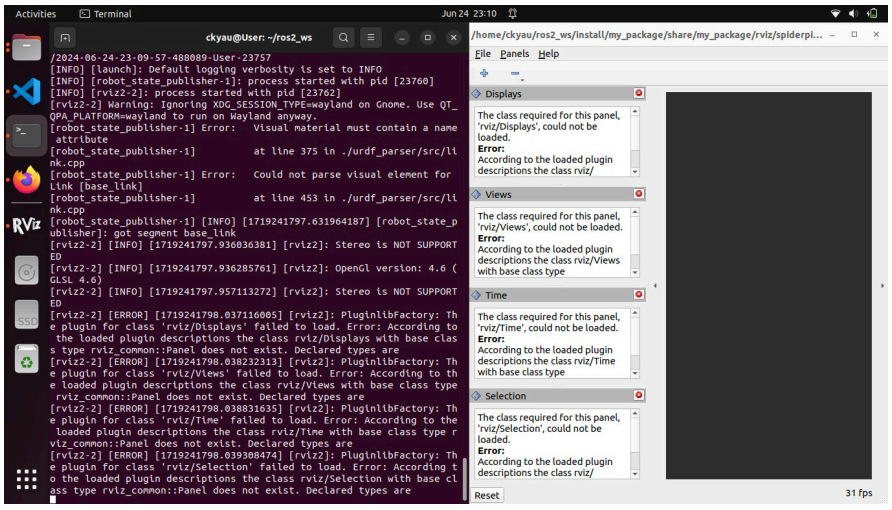
Week 1

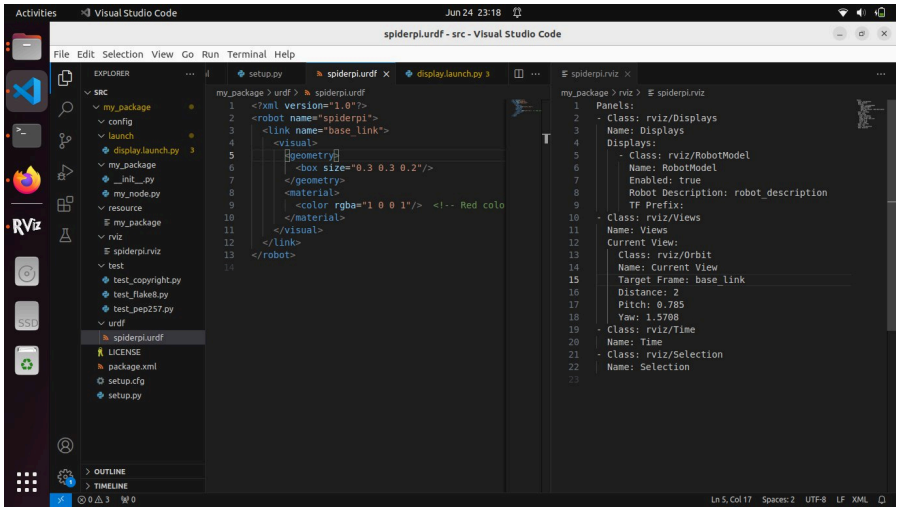
1. RTFM on ROS2 Humble

- Originally was packing the python files, but then wei sheng took over

2. Worked on URDF and packing up workspace

- Problem:
  Stuck on rviz
- Changed from rviz to rviz_common in rviz file





```
kage > rviz > spiderpi.rviz
                                                    rviz
Panels:
- Class: rviz_common/Displays
  Name: Displays
  Displays:
    - Class: rviz_common/RobotModel
      Name: RobotModel
      Enabled: true
      Robot Description: robot_description
      TF Prefix:
- Class: rviz_common/Views
  Name: Views
  Current View:
    Class: rviz_common/Orbit
    Name: Current View
    Target Frame: base_link
    Distance: 2
    Pitch: 0.785
    Yaw: 1.5708
- Class: rviz_common/Time
  Name: Time
- Class: rviz_common/Selection
  Name: Selection
```

Week 2

1. URDF tf2 done, passed to David for imu thingy

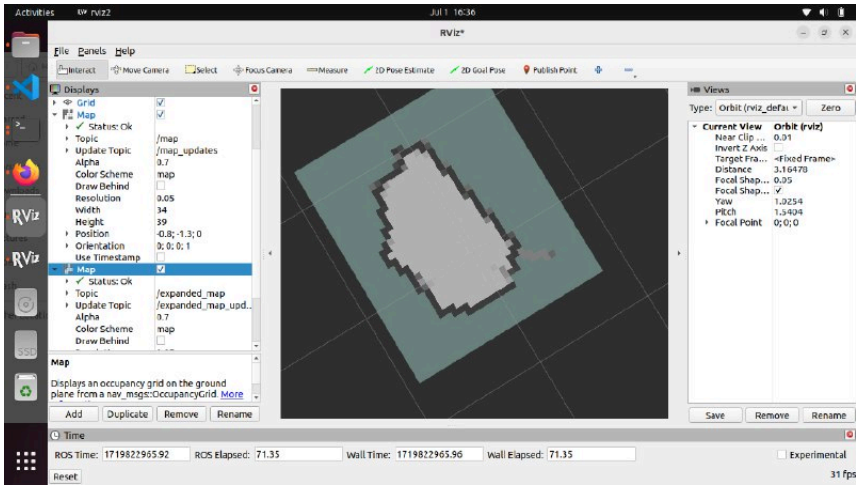
- ros2 run robot_state_publisher robot_state_publisher my_robot2.urdf
- ros2 run rviz2 rviz2
- ros2 run joint_state_publisher_gui joint_state_publisher_gui
  https://www.youtube.com/watch?v=V_C8Cmv4fgk

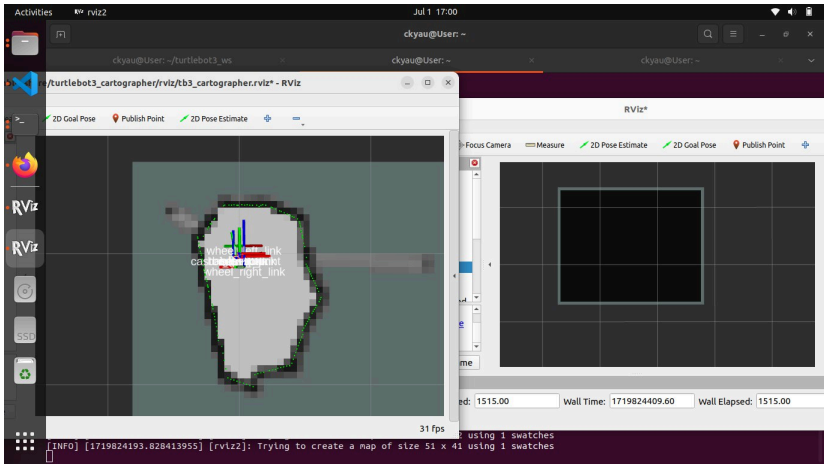
- Tf2 for 2 bots KIV

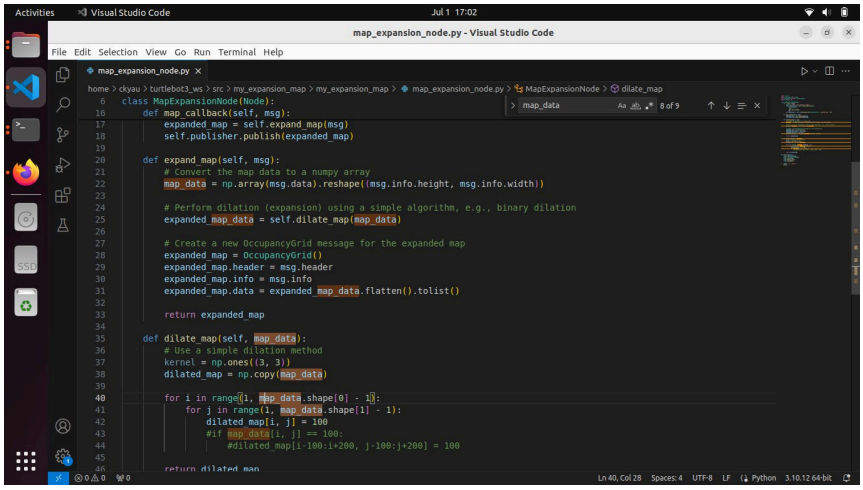
2. Map transformation

https://emanual.robotis.com/docs/en/platform/turtlebot3/slam/#run-slam-node

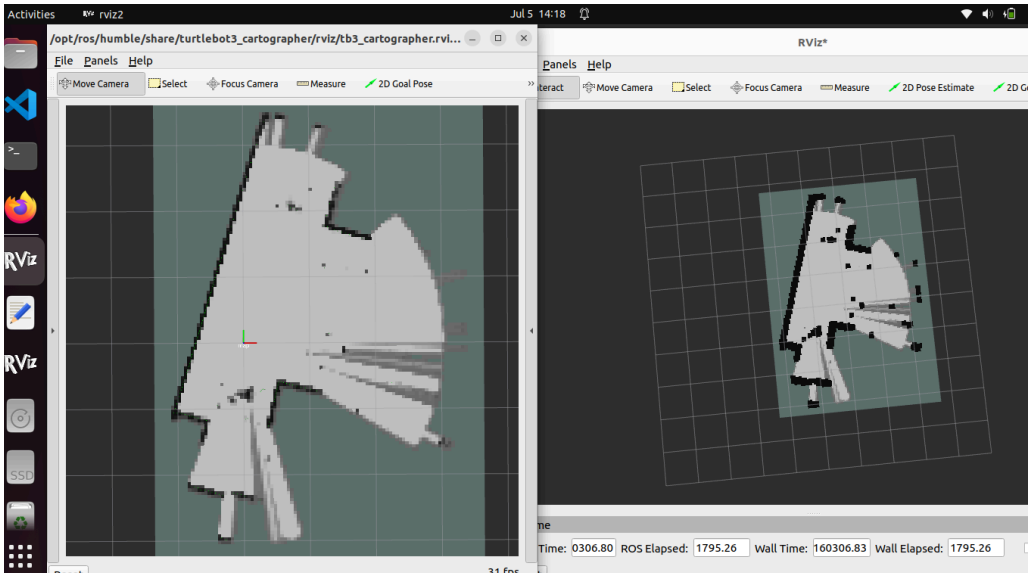
- colcon build --packages-select my_expansion_map
- source install/setup.bash
- ros2 run my_expansion_map map_expansion_node



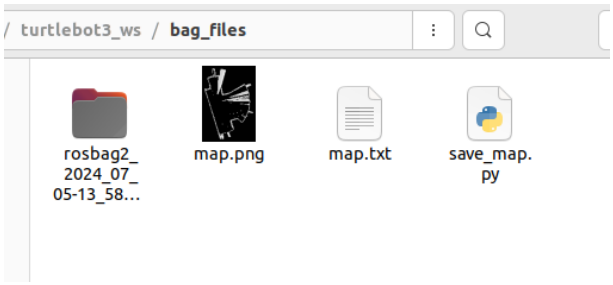




Week 3 Expansion Map



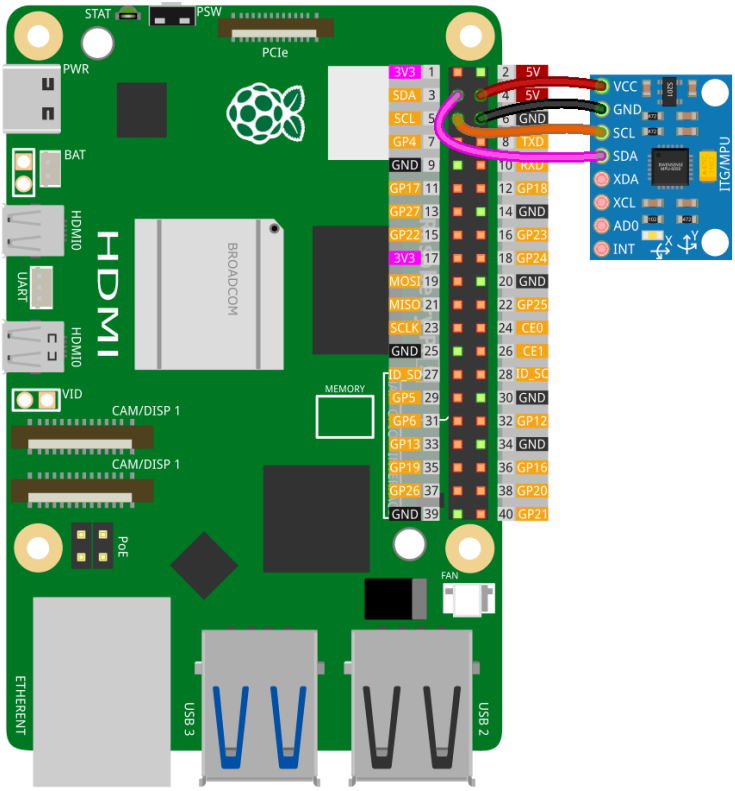
Original code doesn’t work cuz not all values are 100, need to set threshold



Ros2 map and smtg, map file is inside

MPU6050

SCL and SDA same, ground n 5V use rpi4



The first line is SCL, which is a serial clock primarily controlled by the controller device. SCL is used to synchronously clock data in or out of the target device. The second line is SDA, which is the **serial data line**.

https://github.com/MykhailoZhyhariev/mpu6050

https://www.youtube.com/watch?v=W0i-1Hjpqc4

Installation:

- Pip install mpu6050-raspberrypi
- sudo apt install python3-smbus

Troubleshoot errors:

- The PermissionError: [Errno 13] Permission denied indicates that the I2C bus access is restricted due to insufficient permissions. This typically happens when the script is not run with superuser privileges.

  Additionally, ensure that I2C is enabled on your Raspberry Pi:

  1. Open the Raspberry Pi configuration tool by running:

  ```sh
  sudo raspi-config
  ```

  2. Navigate to `Interfacing Options`.
  3. Select `I2C`.
  4. Enable I2C.
  5. Reboot your Raspberry Pi if prompted.

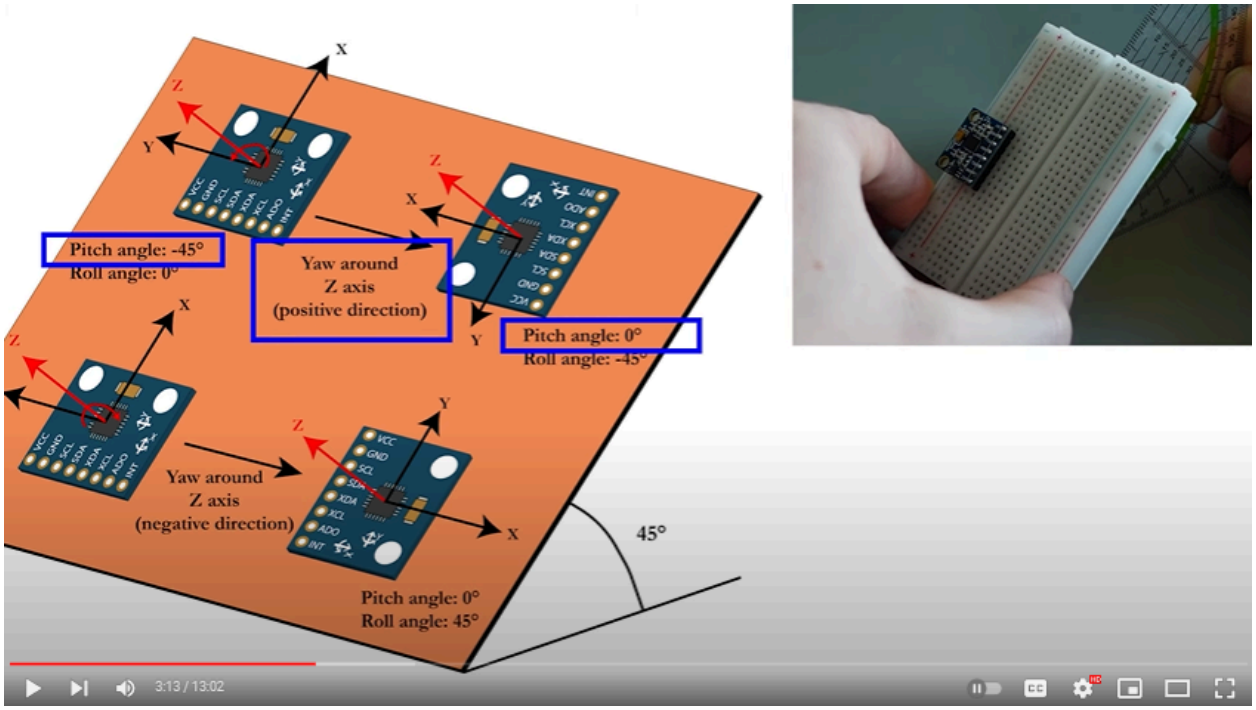
  This should resolve the permission issue. If you encounter any further issues, ensure that the `smbus` package is installed and that your user is part of the `i2c` group:

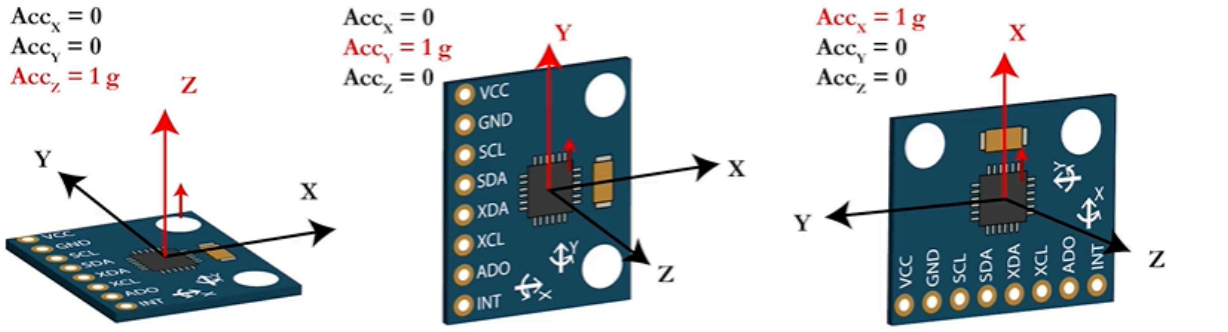
  ```sh
  sudo apt-get install -y python3-smbus
  sudo usermod -aG i2c pi
  ```

  Replace `pi` with your actual username if different. After adding the user to the `i2c` group, a reboot may be necessary for the changes to take effect.

- sudo python3 gyroscope_testing.py

[14 | Measure angles with the MPU6050 accelerometer (youtube.com)]


Pitch angle: -45°
Roll angle: 0°
Yaw around
Z axis
(positive direction)
Pitch angle: 0°
Roll angle: -45°
Yaw around
Z axis
(negative direction)
45°
Pitch angle: 0°
Roll angle: 45°
3:13 / 13:02


Acc$_X$ = 0
Acc$_Y$ = 0
Acc$_Z$ = 1 g
Acc$_X$ = 0
Acc$_Y$ = 1 g
Acc$_Z$ = 0
Acc$_X$ = 1 g
Acc$_Y$ = 0
Acc$_Z$ = 0
VCC
GND
SCL
SDA
XDA
XCL
ADO
INT