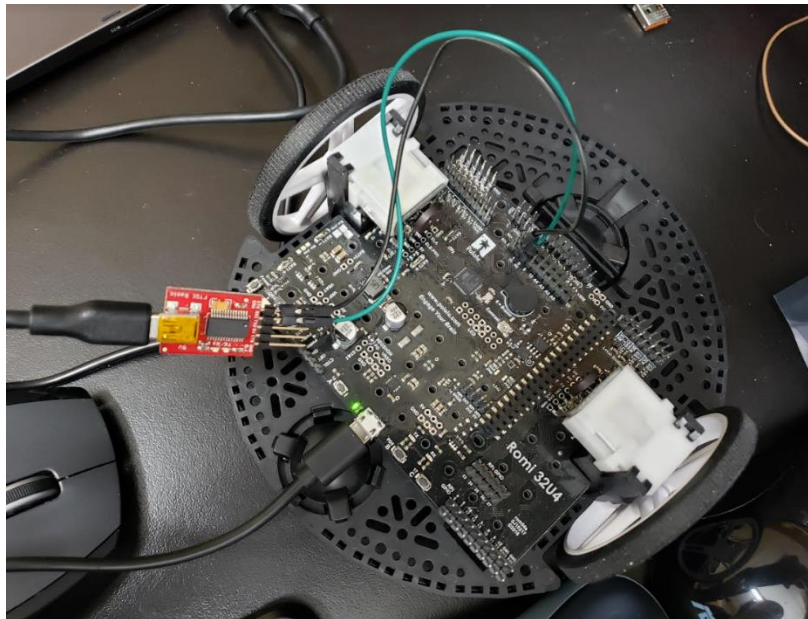


TASK 7

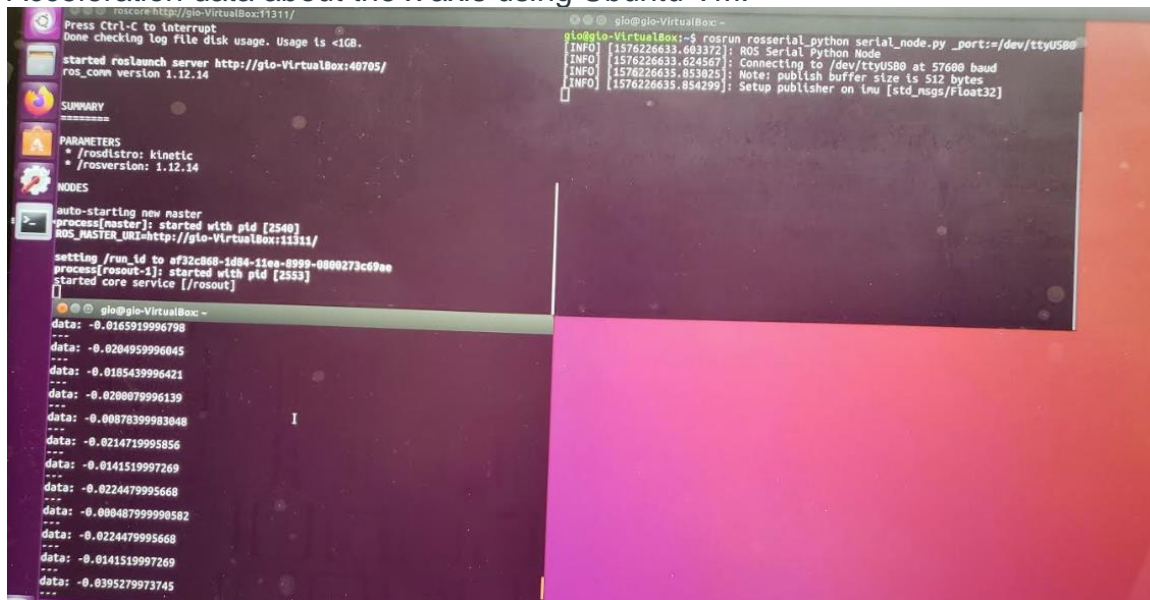
Interface IMU- Interface the IMU on the board to get acceleration and gyro values. Then create a message including odometry and IMU values and transmit the message to ROS.

1. Install latest Arduino IDE, open “Sketch” menu, “Include Library”, then “Manage Libraries..”, and install “Romi32U4” library
2. Open “Updated_imu_data_read_1029.ino” found within the GitHub repository for Task 7.
3. Arduino code will be reading the internal imu readings of the Romi32U4 control board. It will read x, y, and z axis. Therefore, if the board is tilted the imu will read accordingly and display it on the serial monitor.
4. Open up Ubuntu VM or Raspberry pi (installation is described in Task 8). I will be using Ubuntu. Open arduino on the VM and run code “Updated_imu_data_read_1029.ino”. Upload code to the board.
5. Make sure the board is connected correctly. Use an FTDI to transmit and receive board connection serially.
 - FTDI RX pin is connected to Romi32U4 TX pin
 - FTDI TX pin is connected to Romi32U4 RX pin
6. Open a total of three terminals
 - a. First terminal run: `roscore`
 - b. Second terminal run: `roslaunch roserial_python serial_node.py _port:=/dev/ttyUSB0`
 - c. Third terminal run: `rostopic echo /imu`
7. On the third terminal, if ran successfully, it will publish acceleration data near the value of one when the x axis is in parallel with the gravity vector. Otherwise, when perpendicular to the gravity vector, the x-axis will read zero or values near it.
8. Test the code out by moving the rover at different angles and notice how the values change.

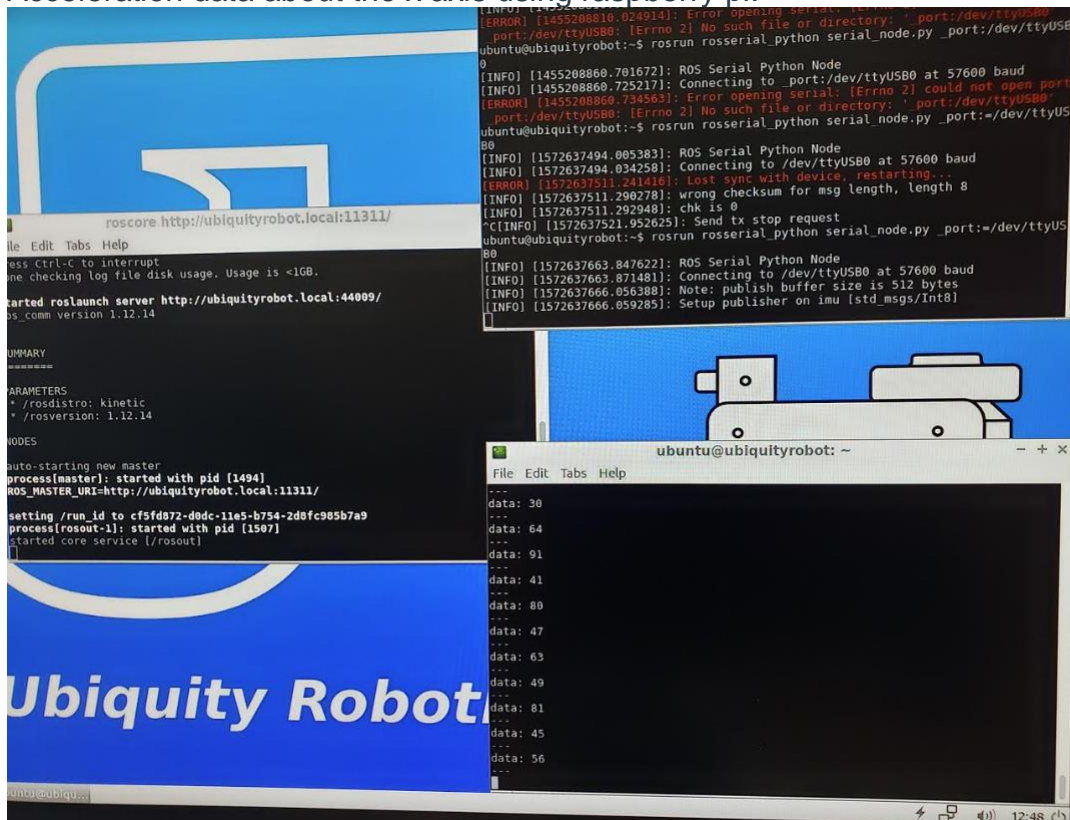
Connection of board with FTDI:



Acceleration data about the x axis using Ubuntu VM:



Acceleration data about the x axis using raspberry pi:



Video:

<https://www.youtube.com/watch?v=VKMsxUMJMcw>