MOBILE ROBOTS (01416511)

Mobile Robot Chapter 3 : Introduction to Hardware.

Introduction to RAIBOCUP

In this chapter, use RAIBOCAP, which shows the 3D design in pictures, to learn the basics of hardware.

The robot diagram in this picture shows that an Intel NUC is used as the main processor for processing ROS and connecting to other devices such as the microcontroller, camera, and LiDAR. The microcontroller, camera, and LiDAR connect via USB cables and use ROS serial communication to send data from the modules to the main processor. The microcontroller sends data to the motor drive with a PWM signal for driving the motor. A 24VDC battery is used for the motor and interlock circuit and steps down to 19.3VDC for the Intel NUC.

ROS Connection

In this topic, use ROS for connecting the LiDAR, camera, and microcontroller. Therefore, you must set the permissions to read, write, and execute with the USB.

Setup USB permission

Use the command below to edit and add permissions for the USB.

```
$ sudo nano /etc/udev/rules.d/50-ired.rules
SUBSYSTEMS=="usb", ATTRS{idVendor}=="10c4", ATTRS{idProduct}=="ea60",
GROUP="users", MODE="0777"
```

The ***lsusb*** command is used to list the USB devices connected to your PC.

```
$ lsusb
```

```
ired@ired-parallels: ~
ired@ired-parallels: ~ 80x24
ired@ired-parallels:~$ lsusb
Bus 003 Device 005: ID 203a:fff9 Parallels Thitipong's iPhone Camera
Bus 003 Device 004: ID 203a:fff9 Parallels FaceTime HD Camera
Bus 003 Device 003: ID 203a:fffb Parallels Virtual Keyboard
Bus 003 Device 002: ID 203a:fffc Parallels Virtual Mouse
Bus 003 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 002 Device 005: ID 10c4:ea60 Silicon Labs CP210x UART Bridge
Bus 002 Device 004: ID 203a:fffe Parallels Virtual USB1.1 HUB
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 004: ID 203a:fffa Parallels Virtual Printer (EPSON L3150 Series)
Bus 001 Device 003: ID 203a:fffa Parallels Virtual Printer (EPSON L6270 Series)
Bus 001 Device 002: ID 203a:fffa Parallels Virtual Printer (Print to PDF (Mac De
sktop))
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

The ***ls -al*** command is used for checking permissions.

```
$ ls -al /dev/ttyUSB0
```

```
ired@ired-parallels:~$ ls -al /dev/ttyUSB0
crwxrwxrwx 1 root dialout 188, 0 Jul 16 21:32 /dev/ttyUSB0
```

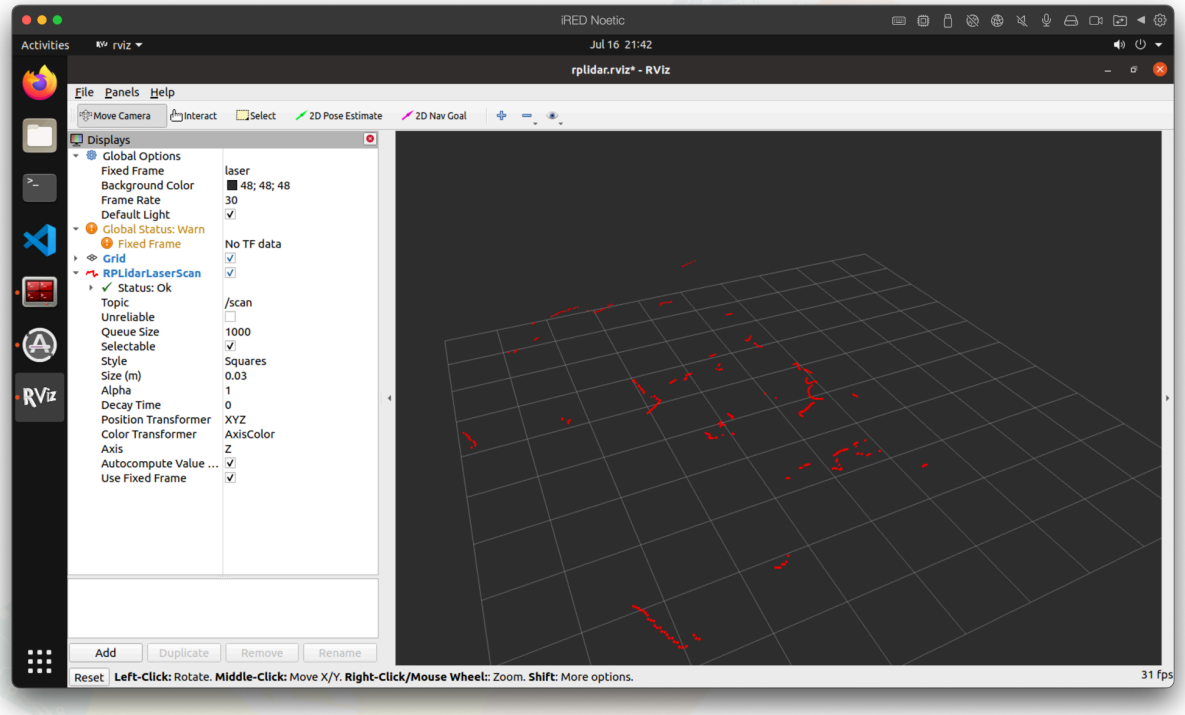
LiDAR

In this chapter, use the RPLiDAR A1 to see the data output from the LiDAR. Use the command below to clone and set up the environment for using the LiDAR package.

```
$ cd ~/catkin_ws/src
$ git clone https://github.com/aims-lab-kmitl/rplidar_ros.git
$ cd ~/catkin_ws
$ catkin_make
$ source ~/.bashrc
```

The command below starts a specific launch file that initializes and runs the RPLiDAR node along with RViz for visualization.

```
$ roslaunch rplidar_ros view_rplidar.launch
```

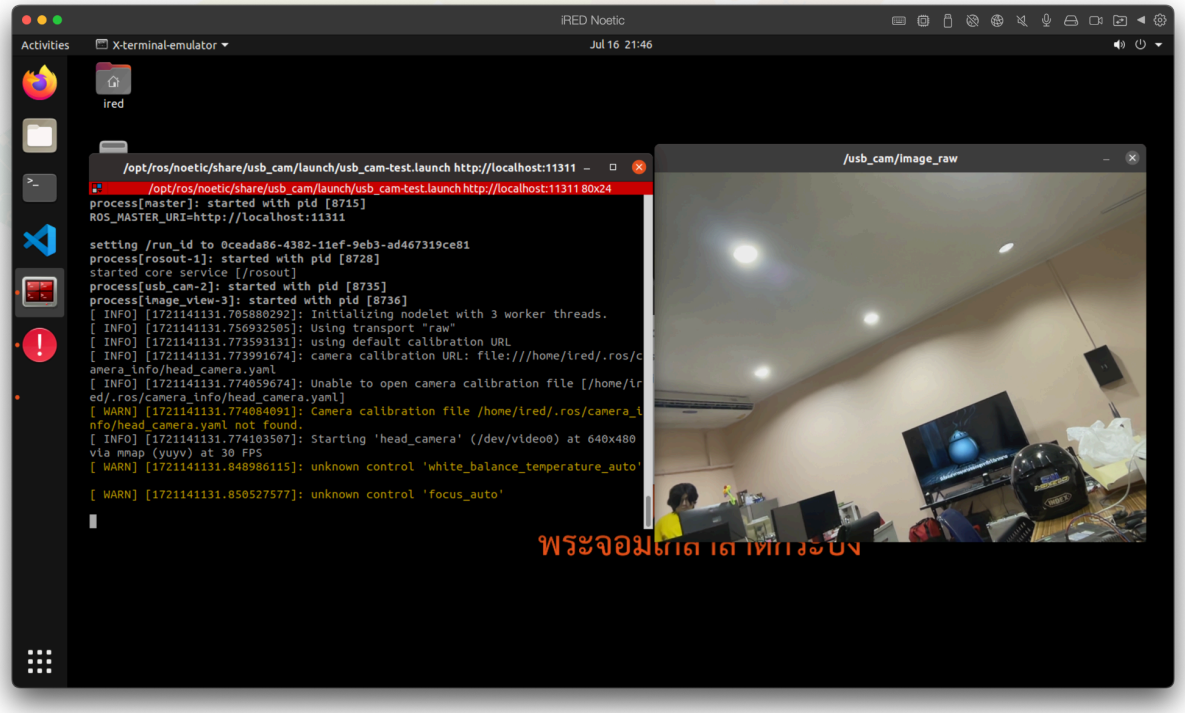


Camera

USB Camera

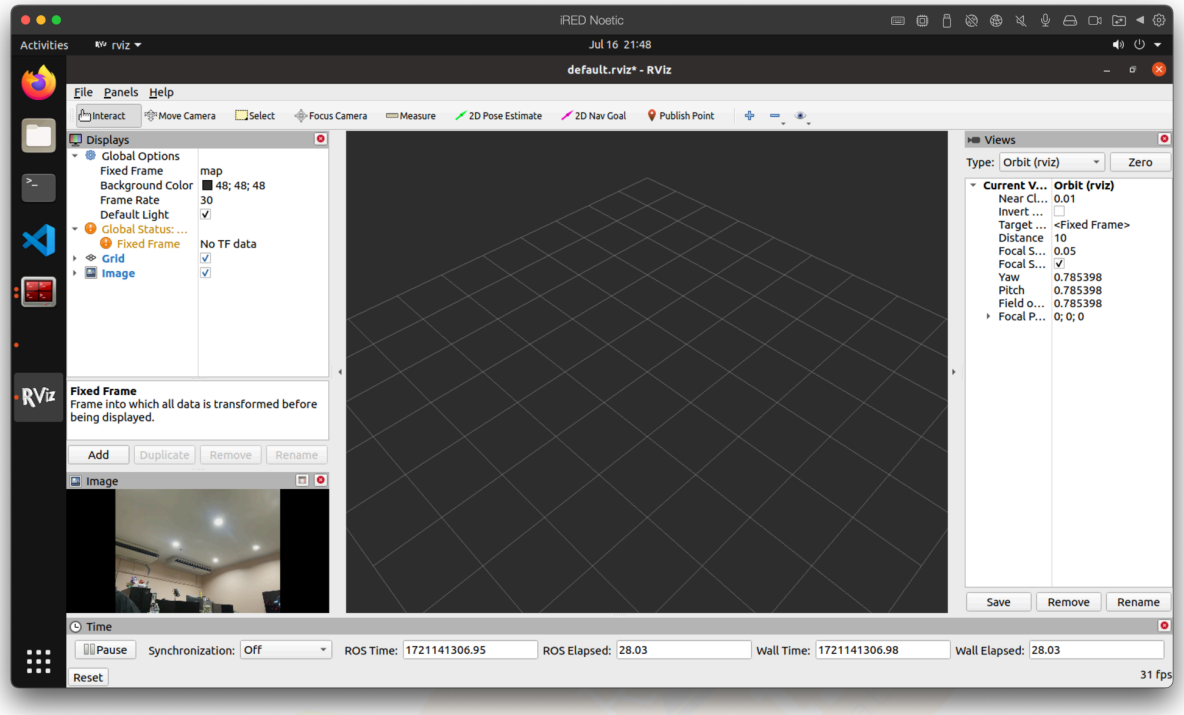
```
$ sudo apt install -y ros-noetic-usb-cam
```

```
$ roslaunch usb_cam usb_cam-test.launch
```

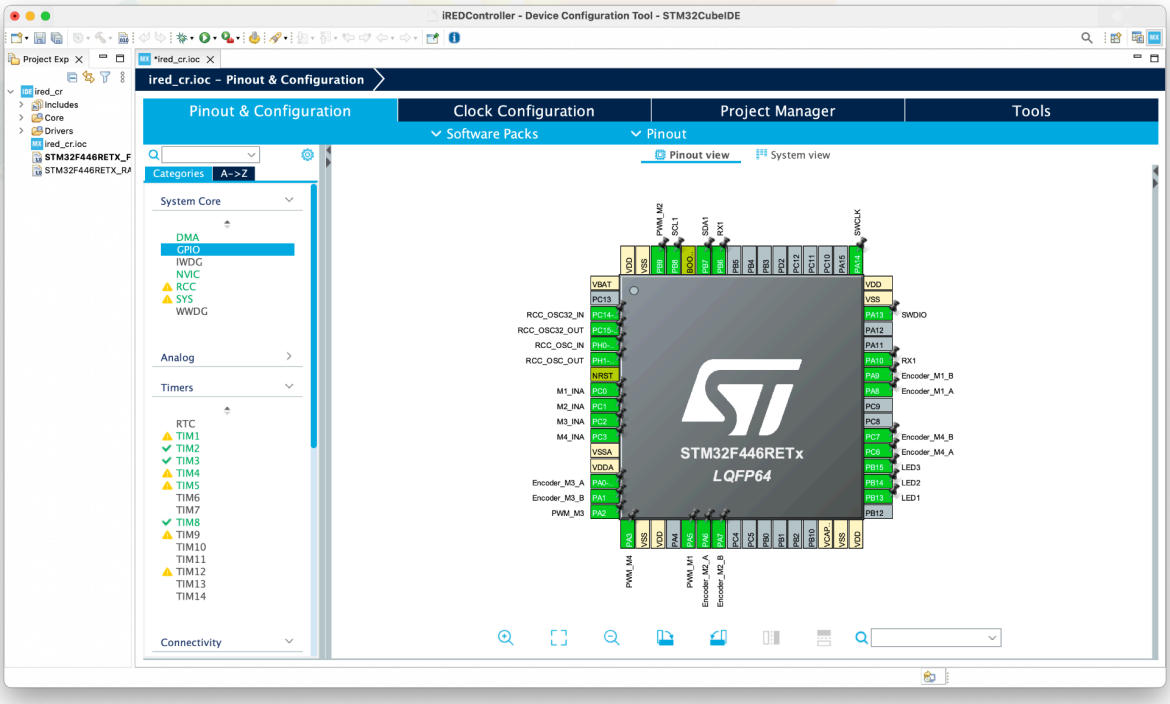


```
$ rosrun rviz rviz
```

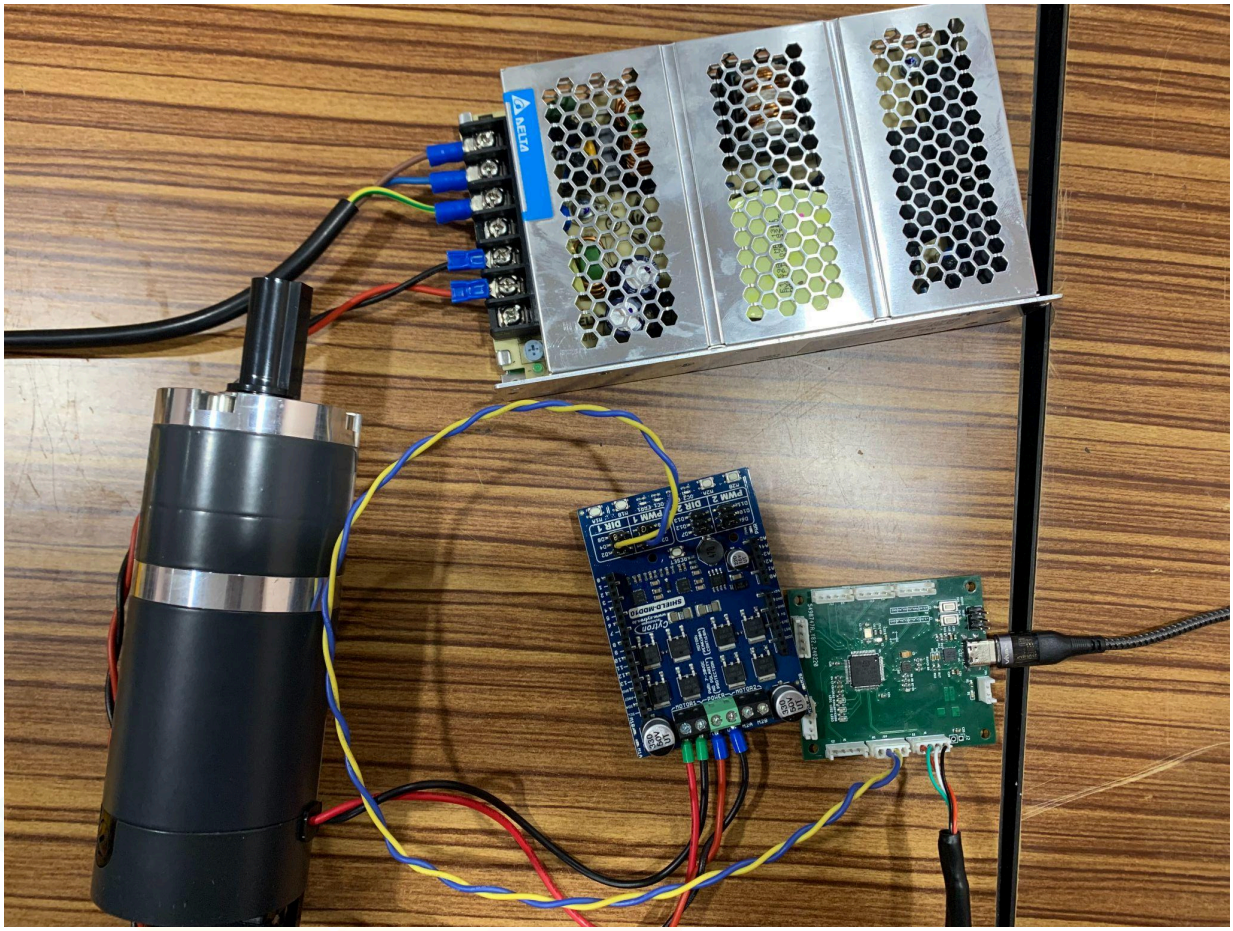
“Add >> By display type >> Image >> Image topic >> /usb_cam/image_raw”



Microcontroller



```
$ cd ~/catkin_ws/src
$ git clone https://github.com/aims-lab-kmitl/rosserial.git
$ git clone -b omni-robot https://github.com/aims-lab-kmitl/ired_msgs.git
$ cd ~/catkin_ws
$ catkin_make
$ source ~/.bashrc
```



The picture shows that 24 VDC from the power supply connects to the motor driver's VB+ and VB-. The DC motor's red and black wires connect to M1A and M1B. The encoder wires connect to the microcontroller port E1. The motor driver's signal wires connect to M1.

```
$ roscore # terminal 1
$ rosrun rosserial_arduino serial_node.py _port:=/dev/ttyUSB0 _baud:=115200 # terminal 2
```

```
$ rostopic list
```

```
ired@ired-parallels:~$ rostopic list
/diagnostics
/ired/imu/raw
/ired/motor/speed_control
/ired/motor/speed_fb
/rosout
/rosout_agg
```

Drive motor with rostopic pub command

```
$ rostopic pub -r 10 /ired/motor/speed_control ired_msgs/MotorControl "speed: [50.0, 0.0, 0.0, 0.0]
pid_motor_front_left: [5.0, 0.0, 0.0]
pid_motor_front_right: [0.0, 0.0, 0.0]
pid_motor_rear_left: [0.0, 0.0, 0.0]
pid_motor_rear_right: [0.0, 0.0, 0.0]"
```

```
ired@ired-parallels:~$ rostopic pub -r 10 /ired/motor/speed_control ired_msgs/MotorControl "sp
eed: [50.0, 0.0, 0.0, 0.0]
pid_motor_front_left: [5.0, 0.0, 0.0]
pid_motor_front_right: [0.0, 0.0, 0.0]
pid_motor_rear_left: [0.0, 0.0, 0.0]
pid_motor_rear_right: [0.0, 0.0, 0.0]"
```

Data from encoder.

```
$ rostopic echo -n 5 /ired/motor/speed_fb
```

```
ired@ired-parallels:~$ rostopic echo /ired/motor/speed_fb -n 5
speed: [17.734833306442166, 0.0, -0.15288649402105314, 0.07644324701052657]
---
speed: [17.811276553452693, 0.0, -0.15288649402105314, -0.07644324701052657]
---
speed: [17.88771980046322, 0.0, -0.15288649402105314, 0.07644324701052657]
---
speed: [17.734833306442166, -0.07644324701052657, -0.15288649402105314, -0.07644
324701052657]
---
speed: [17.811276553452693, 0.07644324701052657, -0.15288649402105314, 0.0764432
4701052657]
---
ired@ired-parallels:~$
```

See the first data. That is motor that you connect

Raw data from IMU

```
$ rostopic echo -n 1 /ired/imu/raw
```

```
ired@ired-parallels:~$ rostopic echo -n 1 /ired/imu/raw
linear:
  x: 0.0010512695312499994
  y: 0.0024072265625
  z: 0.99526171875
angular:
  x: 0.011175572519085633
  y: 0.023053435114502422
  z: -0.14097709923664087
---
```

Assignment

Subscribe to the topic ***/ired/imu/raw*** and publish to the topic ***/ired/imu/data*** with the message type ***sensor_msgs/IMU***. Open RViz to see the transform axis. Submit your group assignment to the classroom and submit your code.