

IMU Data Printing

IMU Data Printing

1. Connecting the Device
2. Binding Device ID
 - 2.1 Device Viewing Commands
 - 2.2 Establishing Port Mapping
 - Introduction to Rule File Syntax
 - 2.3 Verification and Viewing
3. Installing the Driver Library
 - 3.1 Downloading the File
 - 3.2 Installing the Driver Library
 - Unzip
 - Install Library
 - Install other necessary libraries
4. IMU ROS environment configuration
 - 4.1 Creating and compiling a workspace
 - 4.2 Printing IMU Data
5. Raspberry Pi 5, RDK X5 series, Jetson series
 - 5.1 Installing Docker
 - 5.2 Importing Docker Images
 - 5.3 Start the Docker Image
 - 5.4 Print IMU Data

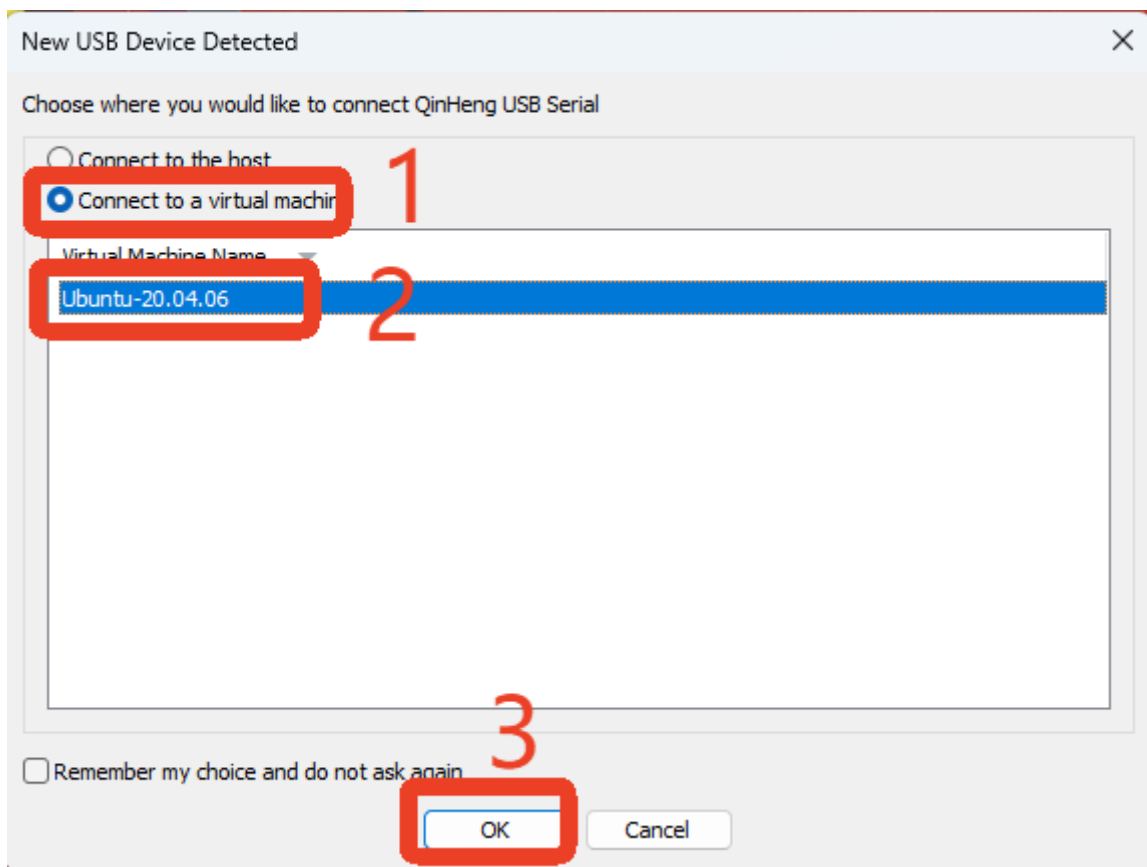
This tutorial uses Ubuntu 22.04 and ROS2 version Humble as an example.

1. Connecting the Device

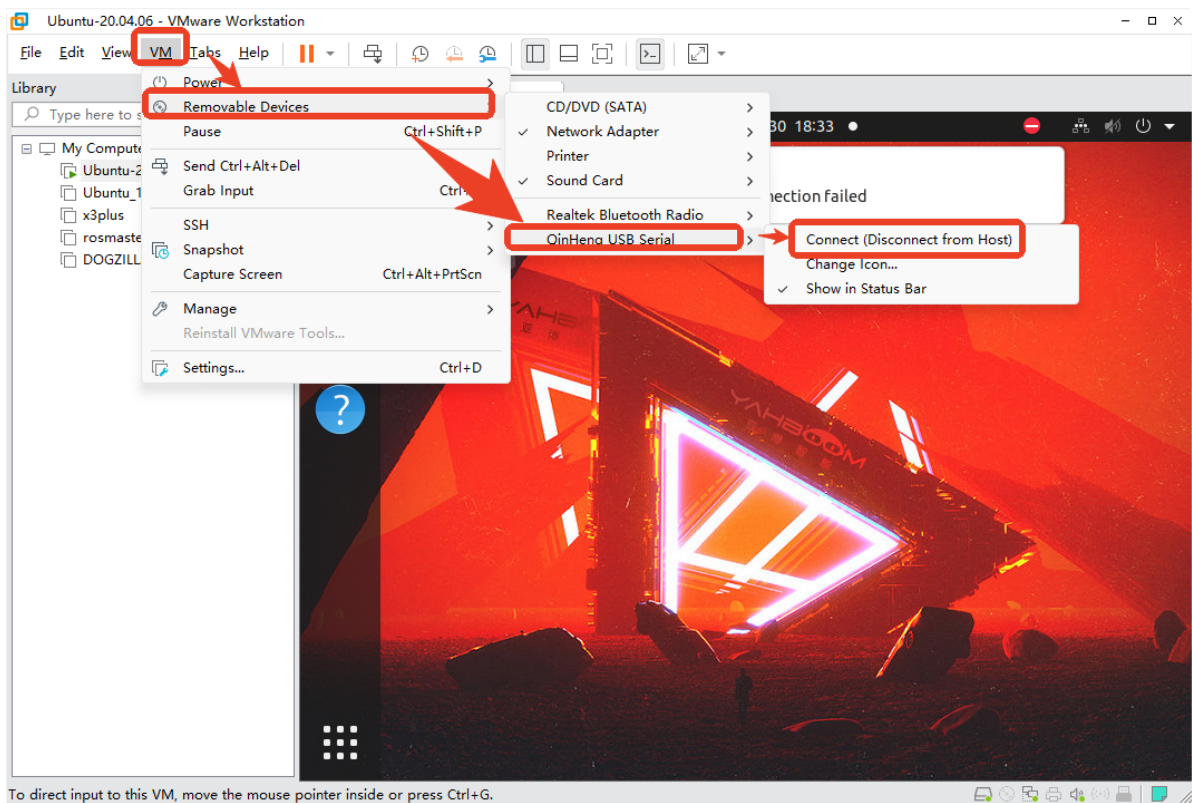
Connect the IMU attitude sensor to the host controller's USB port using a Type-C cable.

If using a virtual machine, refer to the following steps:

Virtual machine displays a pop-up window



Virtual machine does not display a pop-up window



2. Binding Device ID

2.1 Device Viewing Commands

View Device ID

```
lsusb
```

```
yahboom@VM: ~  
-----  
[System Information]  
IP_Address_1: 192.168.11.181  
-----  
ROS_DOMAIN_ID: 28 | ROS: humble  
-----  
yahboom@VM:~$ lsusb  
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub  
Bus 003 Device 004: ID 0e0f:0002 VMware, Inc. Virtual USB Hub  
Bus 003 Device 003: ID 0e0f:0002 VMware, Inc. Virtual USB Hub  
Bus 003 Device 005: ID 1a86:7523 QinHeng Electronics CH340 serial converter  
Bus 003 Device 002: ID 0e0f:0003 VMware, Inc. Virtual Mouse  
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub  
yahboom@VM:~$
```

View Device Number

```
ll /dev/ttyU*
```

```
yahboom@VM:~$ ll /dev/ttyU*  
crw-rw---- 1 root dialout 188, 0 10月 22 19:00 /dev/ttyUSB0  
yahboom@VM:~$
```

2.2 Establishing Port Mapping

Enter the rules.d directory

```
cd /etc/udev/rules.d/
```

Create a new file named myimu.rules and write the following content:

```
sudo gedit myimu.rules
```

If it prompts that the gedit command cannot be found, run:

```
sudo apt install gedit -y
```

Write the following content:

```
KERNEL=="ttyUSB*", ATTRS{idVendor}=="1a86", ATTRS{idProduct}=="7523",  
MODE:="0777", SYMLINK+="myimu"
```



Save and exit to apply the rule

```
sudo udevadm trigger  
sudo service udev reload  
sudo service udev restart
```

Replug and replug the USB device. Done!

Introduction to Rule File Syntax

```
KERNEL=="ttyUSB*", ATTRS{idVendor}=="1a86", ATTRS{idProduct}=="7523",  
MODE:="0777", SYMLINK+="myimu"
```

Analysis

```
KERNEL # The device name matching the event  
ATTR{filename} # The sysfs attribute of the device matching the event. ...  
idVendor # Manufacturer ID  
idProduct # Product ID  
SYMLINK # Creates a symbolic link for the device file under /dev/. This gives  
the device an alias.  
MODE # Sets permissions for the device.
```

2.3 Verification and Viewing

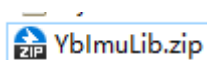
```
ll /dev/my*
```

```
yahboom@VM:/etc/udev/rules.d$ ll /dev/my*  
lrwxrwxrwx 1 root root 7 10月 22 19:02 /dev/myimu -> ttyUSB0  
yahboom@VM:/etc/udev/rules.d$
```

3. Installing the Driver Library

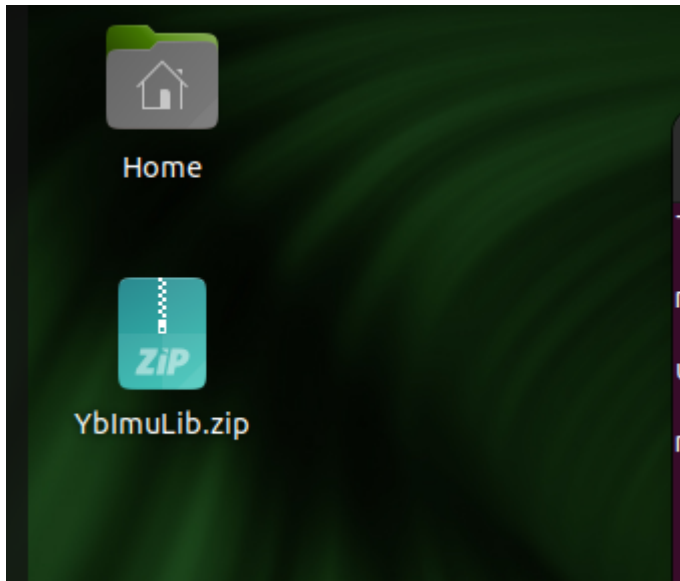
3.1 Downloading the File

The latest version of the driver library, named YbImuLib.zip, is provided in the data folder.



3.2 Installing the Driver Library

Copy the YbImuLib.zip file to the virtual machine.



Unzip

```
unzip YbImuLib.zip
```

```
yahboom@VM:~/Desktop$ unzip YbImuLib.zip
Archive:  YbImuLib.zip
  creating: YbImuLib/
  inflating: YbImuLib/.gitignore
  inflating: YbImuLib/README.md
  inflating: YbImuLib/setup.py
   creating: YbImuLib/YbImuLib/
  inflating: YbImuLib/YbImuLib/__init__.py
  inflating: YbImuLib/YbImuLib/YbImuI2cLib.py
  inflating: YbImuLib/YbImuLib/YbImuSerialLib.py
yahboom@VM:~/Desktop$
```

Install Library

```
cd YbImuLib
```

Run the installation command. If you see the version number displayed at the end, the installation was successful. This command will overwrite any previously installed Rosmaster_Lib driver library.

```
sudo python3 setup.py install
```

```

yahboom@VM:~/Desktop$ cd YbImuLib
yahboom@VM:~/Desktop/YbImuLib$ sudo python3 setup.py install
running install
/usr/lib/python3/dist-packages/setuptools/command/install.py:34: SetuptoolsDeprecationWarning: setup.py install is deprecated. Use build and pip and other standards-based tools.
  warnings.warn(
/usr/lib/python3/dist-packages/setuptools/command/easy_install.py:158: EasyInstallDeprecationWarning: easy_install command is deprecated. Use build and pip and other standards-based tools.
  warnings.warn(
running bdist_egg

```

Install other necessary libraries

```

sudo pip3 install pyserial
sudo pip3 install smbus2

```

4. IMU ROS environment configuration

4.1 Creating and compiling a workspace

Open the command terminal and run the following command:

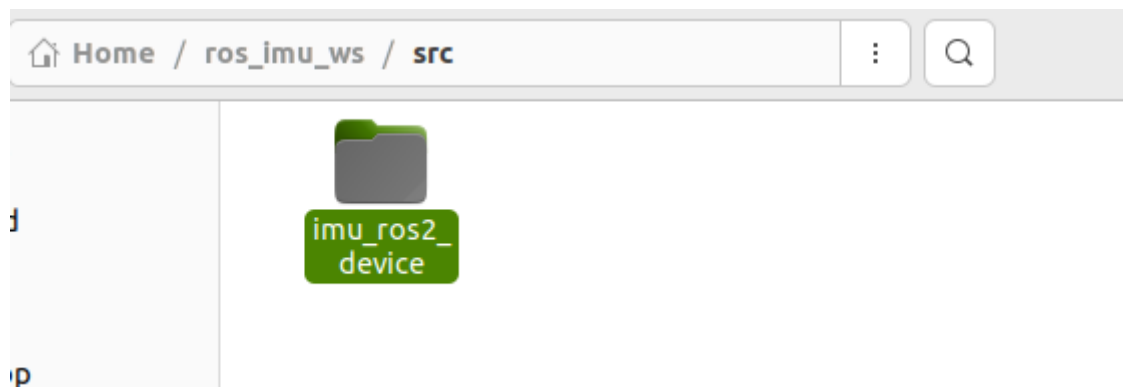
Return to the home directory and create a new workspace

```

mkdir ros_imu_ws
cd ros_imu_ws
mkdir src
colcon build --symlink-install

```

Copy the imu_ros2_device folder to the ~/ros_imu_ws/src directory.



Compile

```

cd ~/ros_imu_ws
colcon build

```

After successful compilation, add the workspace path to .bashrc

```

sudo gedit ~/.bashrc

```

Copy the following content to the end of the file:

```

source ~/ros_imu_ws/install/setup.bash

```

```

133
134 # Print IP address 1
135 echo -e "\033[33mIP_Address_1: $(hostname -I | awk '{print $1}')\033[0m"
136 # Print IP address 2
137 #echo -e "\033[33mIP_Address_2: $(hostname -I | awk '{print $2}')\033[0m"
138
139 export ROS_DOMAIN_ID=28
140
141 # ros2
142 source /opt/ros/humble/setup.bash
143
144 echo "-----"
145 echo -e "ROS_DOMAIN_ID: \033[32m$ROS_DOMAIN_ID\033[0m | \033[34mROS: $(printenv ROS_DISTRO)
\033[0m"
146 echo "-----"
147
148 source ~/ros_imu_ws/install/setup.bash

```

Save and exit, update environment variables

```
source ~/.bashrc
```

4.2 Printing IMU Data

Open a terminal and start the IMU node

```
ros2 run imu_ros2_device ybimu_driver
```

Open a second terminal and check the IMU topic

```
ros2 topic list
```

```

yahboom@VM:~$ ros2 topic list
/baro
/euler
/imu/data_raw
/imu/mag
/parameter_events
/rosout
yahboom@VM:~$

```

Print the data for the /imu/data_raw topic

```
ros2 topic echo /imu/data_raw
```

```
header:
  stamp:
    sec: 1761132965
    nanosec: 7432176
    frame_id: imu_link
orientation:
  x: -0.4606093764305115
  y: 0.08095275610685349
  z: -0.22871613502502441
  w: 0.8518229126930237
variance:
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
angular_velocity:
  x: 0.0
  y: 0.0
  z: 0.0
angular_velocity_covariance:
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
linear_acceleration:
  x: 0.07324442274239326
  y: -0.848170415356914
  z: 0.5727713858455153
linear_acceleration_covariance:
```

Open a third terminal and print /imu/mag Topic Data

```
ros2 topic echo /imu/mag
```



```
docker load -i ros2_imu.tar
```

Note: Importing images takes a long time, please be patient. It is recommended to use a TF card of 64GB or more with at least 30GB of free space to avoid insufficient space causing import failure.

5.3 Start the Docker Image

```
sh ros2_humble.sh
```

5.4 Print IMU Data

Start the IMU Node

```
ros2 run imu_ros2_device ybimu_driver
```

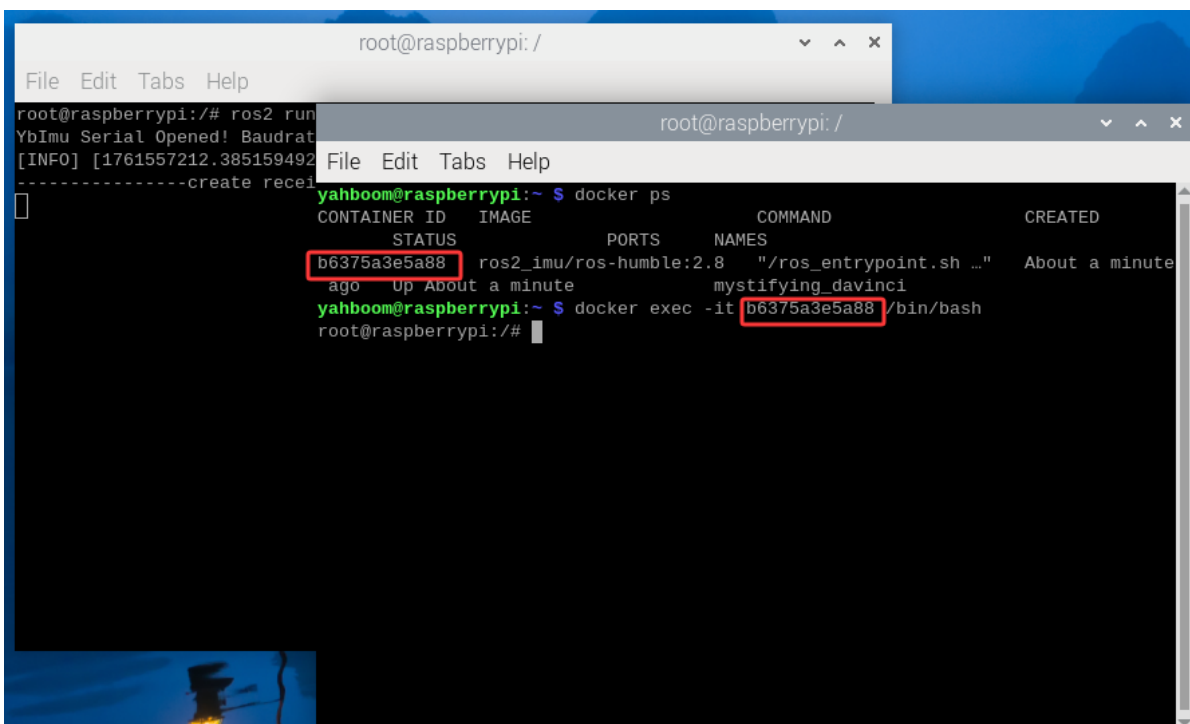
```
root@raspberrypi:/# ros2 run imu_ros2_device ybimu_driver
YbImu Serial Opened! Baudrate=115200
[INFO] [1761557212.385159492] [ybimu_node]: Open Ybimu Port OK:/dev/myimu
-----create receive threading-----
```

Open a second terminal and view the container ID

```
docker ps
```

Based on the container ID displayed above, change the container ID in the following command to the actual displayed ID, allowing multiple terminals to access the same Docker container

```
docker exec -it container_id /bin/bash
```



Check IMU topics

```
ros2 topic list
```

```
root@raspberrypi:/# ros2 topic list
/baro
/euler
/imu/data_raw
/imu/mag
/parameter_events
/rosout
root@raspberrypi:/#
```

Print /imu/data_raw topic data

```
ros2 topic echo /imu/data_raw
```

```
root@raspberrypi:/# ros2 topic echo /imu/data_raw
header:
  stamp:
    sec: 1761557303
    nanosec: 393602383
  frame_id: imu_link
orientation:
  x: 0.013812731951475143
  y: -0.02381979674100876
  z: 0.390880823135376
  w: 0.9181948900222778
orientation_covariance:
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
angular_velocity:
  x: 0.0
  y: 0.0
  z: 0.0
angular_velocity_covariance:
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
linear_acceleration:
```

Print /imu/mag topic data

```
ros2 topic echo /imu/mag
```

```
root@raspberrypi:/# ros2 topic echo /imu/mag
```

```
header:
```

```
  stamp:
```

```
    sec: 1761557377
```

```
    nanosec: 893474253
```

```
  frame_id: imu_link
```

```
magnetic_field:
```

```
  x: 17.432172612689595
```

```
  y: 19.14120914334544
```

```
  z: -15.991698965422529
```

```
magnetic_field_covariance:
```

```
- 0.0
```

```
- 0.0
```

```
- 0.0
```

```
- 0.0
```

```
- 0.0
```

```
- 0.0
```

```
- 0.0
```

```
- 0.0
```

```
- 0.0
```