

# 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 the Docker Image
  - 5.3 Start the Docker Image
  - 5.4 Print IMU Data

This tutorial uses Ubuntu 20.04 and ROS1 version Noetic as an example.

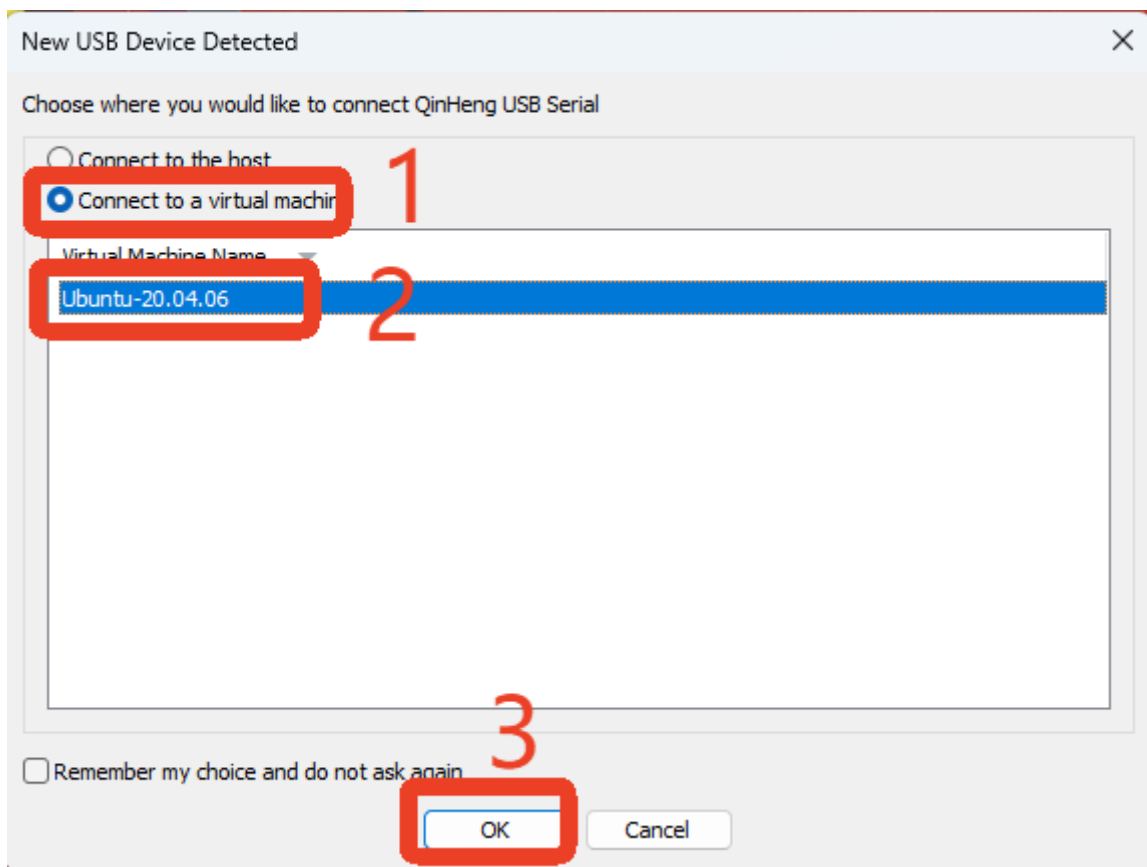
## 1. Connecting the Device

---

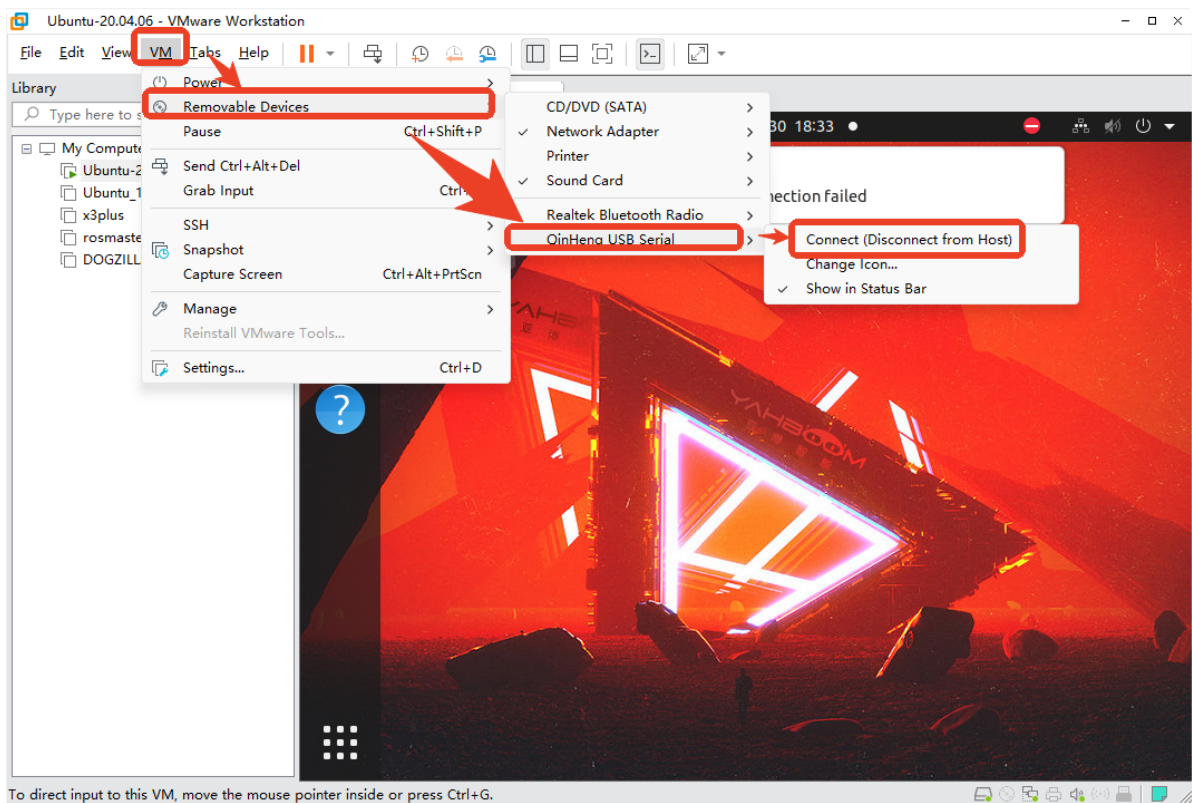
Connect the IMU attitude sensor to the host controller's USB port using a Type-C cable.

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@yahboom-virtual-machine: ~  
yahboom@yahboom-virtual-machine:~$ lsusb  
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub  
Bus 003 Device 007: ID 0e0f:0002 VMware, Inc. Virtual USB Hub  
Bus 003 Device 005: ID 0e0f:0002 VMware, Inc. Virtual USB Hub  
Bus 003 Device 003: ID 0e0f:0002 VMware, Inc. Virtual USB Hub  
Bus 003 Device 008: ID 1a86:7523 QinHeng Electronics HL-340 USB-Serial adapter  
Bus 003 Device 004: ID 0e0f:0003 VMware, Inc. Virtual Mouse  
Bus 003 Device 002: ID 0e0f:0002 VMware, Inc. Virtual USB Hub  
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
Bus 002 Device 002: ID 0e0f:0002 VMware, Inc. Virtual USB Hub  
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub  
yahboom@yahboom-virtual-machine:~$
```

View Device Number

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

```
yahboom@yahboom-virtual-machine:~$ ll /dev/ttyU*  
crwxrwxrwx 1 root dialout 188, 0 10月 17 17:28 /dev/ttyUSB0  
yahboom@yahboom-virtual-machine:~$
```

## 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:

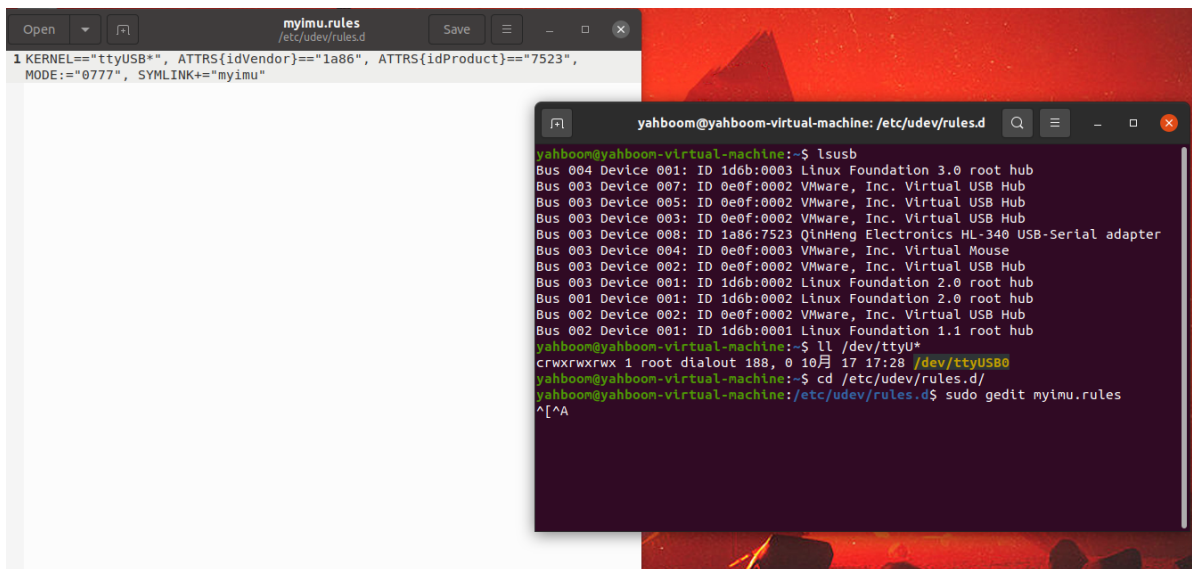
```
sudo gedit myimu.rules
```

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

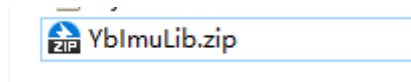
```
ll /dev/my*
```

```
yahboom@yahboom-virtual-machine:/etc/udev/rules.d$ ll /dev/my*
lrwxrwxrwx 1 root root 7 10月 17 17:28 /dev/myimu -> ttyUSB0
yahboom@yahboom-virtual-machine:/etc/udev/rules.d$
```

## 3. Installing the Driver Library

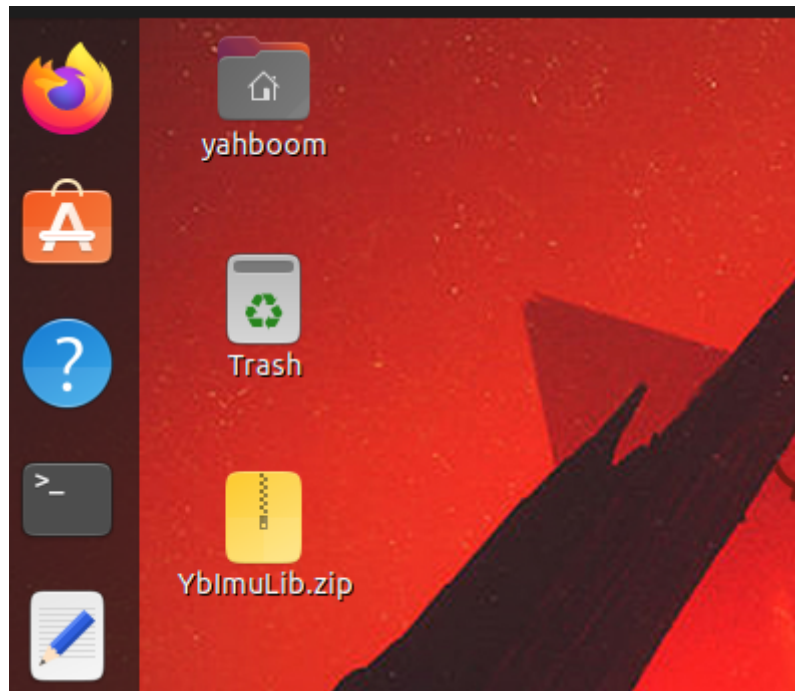
### 3.1 Downloading the File

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@yahboom-virtual-machine:~/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@yahboom-virtual-machine:~/Desktop$
```

#### Install Library

```
cd YbImuLib
```

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

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

```

yahboom@yahboom-virtual-machine:~/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@yahboom-virtual-machine:~/Desktop$ cd YbImuLib
yahboom@yahboom-virtual-machine:~/Desktop/YbImuLib$ sudo python3 setup.py install
running install
running bdist_egg
running egg_info
creating YbImuLib.egg-info
writing YbImuLib.egg-info/PKG-INFO
writing dependency_links to YbImuLib.egg-info/dependency_links.txt
writing top-level names to YbImuLib.egg-info/top_level.txt

```

## 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, return to the home directory, and run the following command:

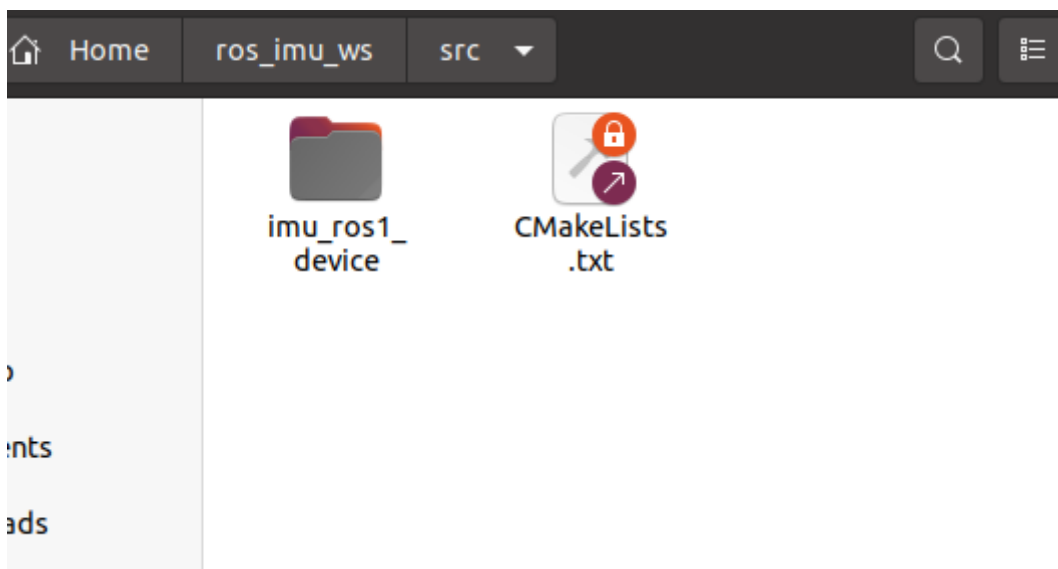
Create a new workspace

```

mkdir ros_imu_ws
cd ros_imu_ws
mkdir src
cd src
catkin_init_workspace

```

Copy the imu\_ros1\_device folder to the ~/ros\_imu\_ws/src directory



Compile

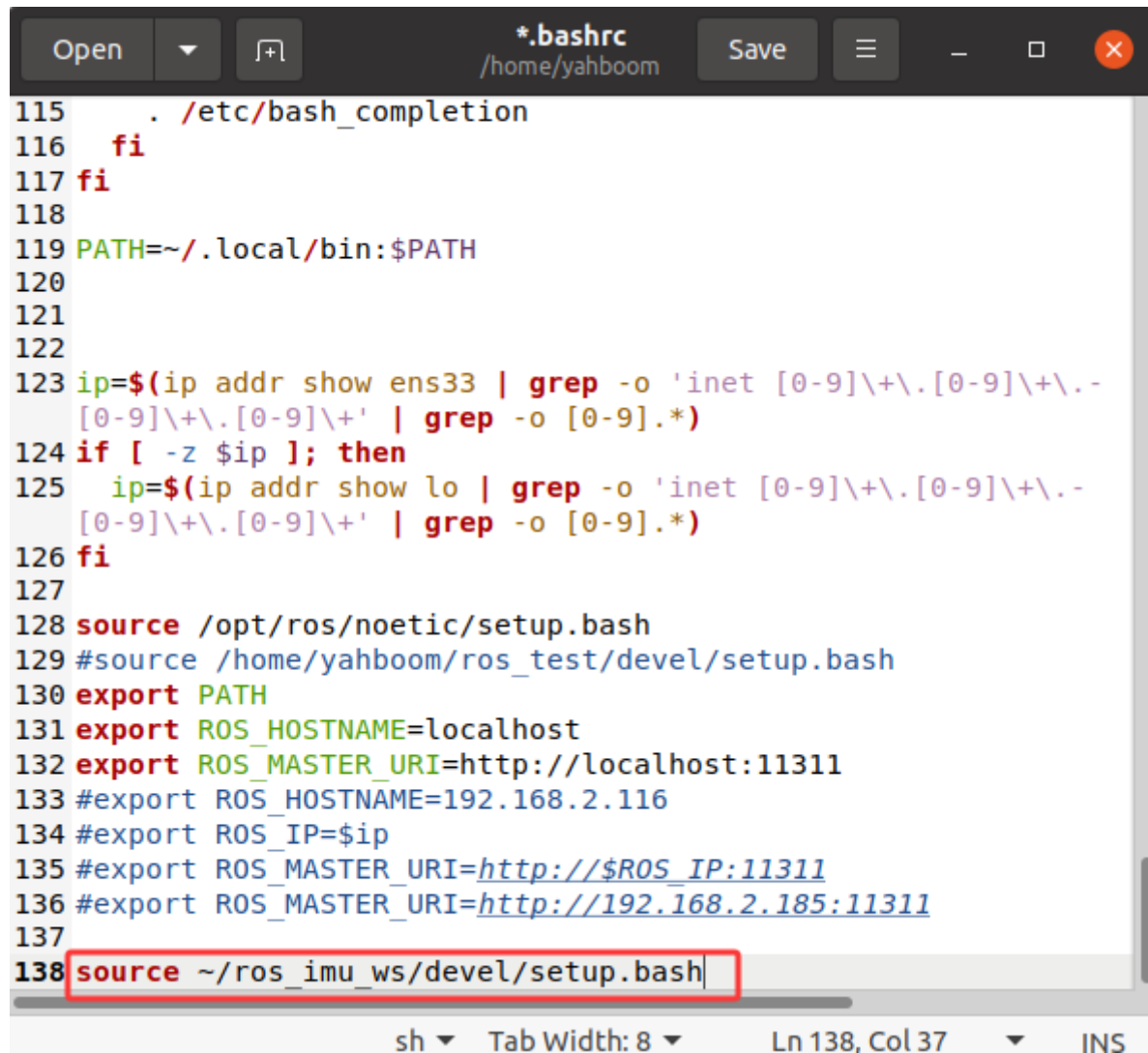
```
cd ~/ros_imu_ws
catkin_make
```

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/devel/setup.bash
```



The screenshot shows a text editor window titled `*.bashrc` with the path `/home/yahboom`. The file content is as follows:

```
115 . /etc/bash_completion
116 fi
117 fi
118
119 PATH=~/.local/bin:$PATH
120
121
122
123 ip=$(ip addr show ens33 | grep -o 'inet [0-9]\+\.[0-9]\+\.[0-9]\+\.[0-9]\+' | grep -o [0-9].*)
124 if [ -z $ip ]; then
125     ip=$(ip addr show lo | grep -o 'inet [0-9]\+\.[0-9]\+\.[0-9]\+\.[0-9]\+' | grep -o [0-9].*)
126 fi
127
128 source /opt/ros/noetic/setup.bash
129 #source /home/yahboom/ros_test/devel/setup.bash
130 export PATH
131 export ROS_HOSTNAME=localhost
132 export ROS_MASTER_URI=http://localhost:11311
133 #export ROS_HOSTNAME=192.168.2.116
134 #export ROS_IP=$ip
135 #export ROS_MASTER_URI=http://$ROS_IP:11311
136 #export ROS_MASTER_URI=http://192.168.2.185:11311
137
138 source ~/ros_imu_ws/devel/setup.bash
```

The last line, `source ~/ros_imu_ws/devel/setup.bash`, is highlighted with a red box. The editor status bar at the bottom shows `sh`, `Tab Width: 8`, `Ln 138, Col 37`, and `INS`.

Save and exit, update environment variables

```
source ~/.bashrc
```

## 4.2 Printing IMU Data

Open a terminal and start roscore

```
roscore
```

Open another terminal and start the IMU node



```
roslaunch imu_ros1_device ybimu_driver.py
```

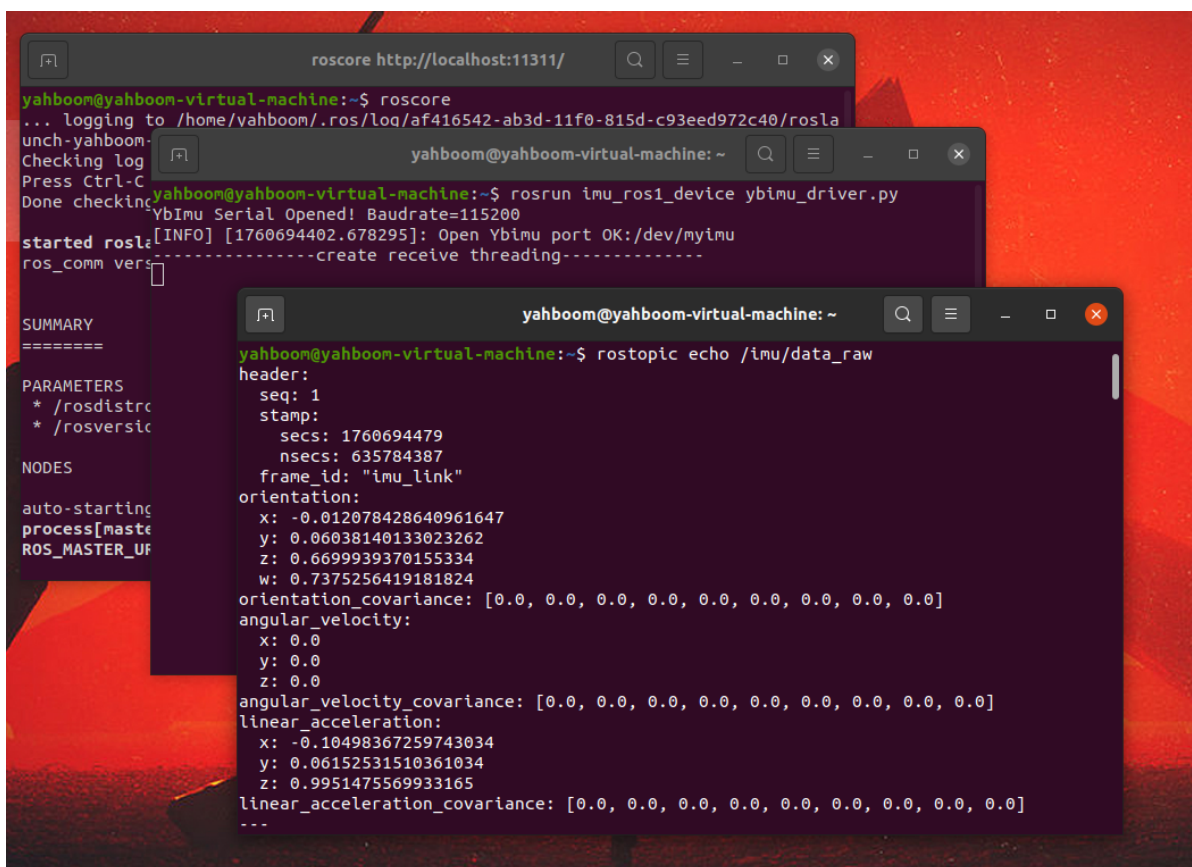
Open a second terminal and check the IMU topic

```
rostopic list
```

```
yahboom@yahboom-virtual-machine:~$ rostopic list
/baro
/imu/data_raw
/imu/mag
/rosout
/rosout_agg
```

Print the data for the /imu/data\_raw topic

```
rostopic echo /imu/data_raw
```



The screenshot shows three terminal windows on a Linux system. The top window shows the output of `roscore` and `roslaunch` commands, indicating that the IMU driver is running. The middle window shows the output of `rostopic list`, listing the available topics. The bottom window shows the output of `rostopic echo /imu/data_raw`, displaying a large amount of sensor data including header information, orientation, angular velocity, linear acceleration, and covariance matrices.

```
roscore http://localhost:11311/
yahboom@yahboom-virtual-machine:~$ roscore
... logging to /home/yahboom/.ros/log/af416542-ab3d-11f0-815d-c93eed972c40/roscore
...
yahboom@yahboom-virtual-machine:~$ roslaunch imu_ros1_device ybimu_driver.py
Ybimu Serial Opened! Baudrate=115200
[INFO] [1760694402.678295]: Open Ybimu port OK:/dev/myimu
started roslaunch
ros_comm version: 1.12.0
-----create receive threading-----

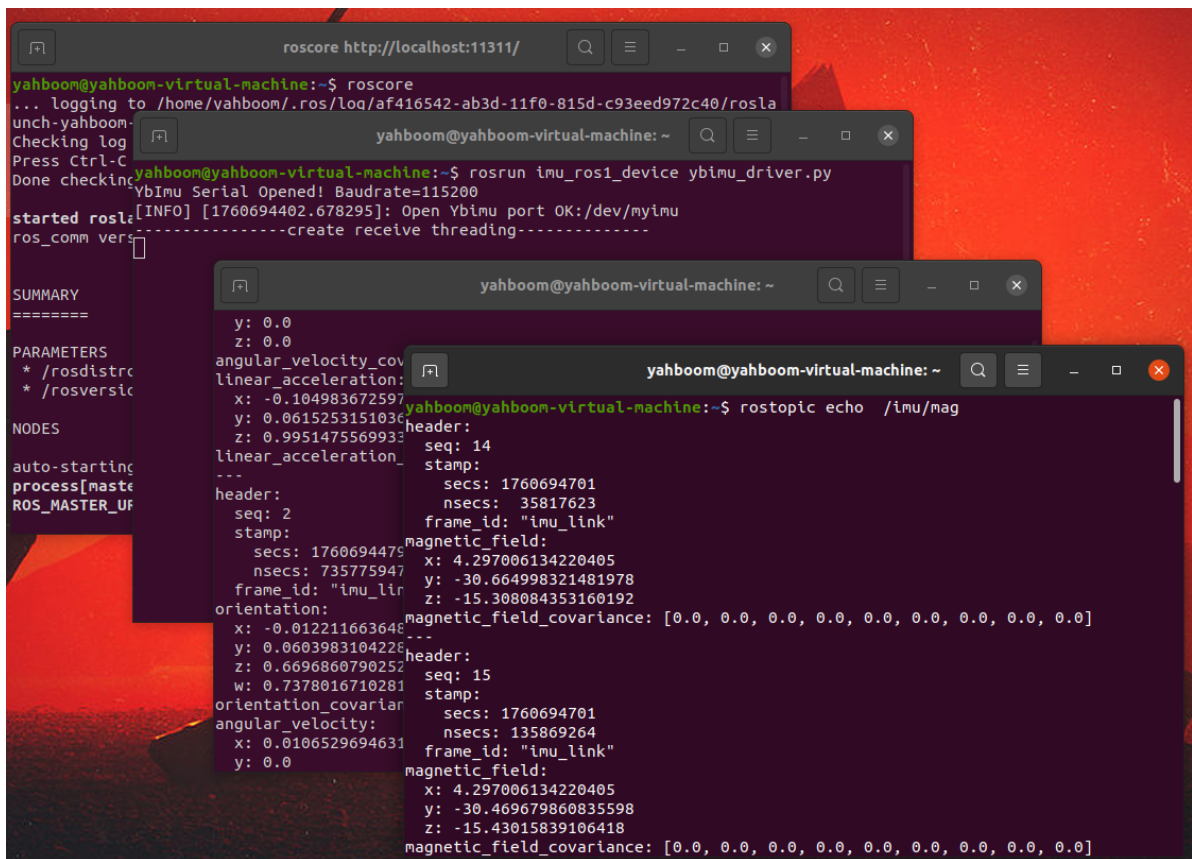
yahboom@yahboom-virtual-machine:~$ rostopic list
/baro
/imu/data_raw
/imu/mag
/rosout
/rosout_agg

yahboom@yahboom-virtual-machine:~$ rostopic echo /imu/data_raw
header:
  seq: 1
  stamp:
    secs: 1760694479
    nsecs: 635784387
  frame_id: "imu_link"
orientation:
  x: -0.012078428640961647
  y: 0.06038140133023262
  z: 0.6699939370155334
  w: 0.7375256419181824
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:
  x: -0.10498367259743034
  y: 0.06152531510361034
  z: 0.9951475569933165
linear_acceleration_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
---
```

Open a new third terminal and print the topic data /imu/mag

```
rostopic echo /imu/mag
```





```
roscore http://localhost:11311/
yahboom@yahboom-virtual-machine:~$ roscore
... logging to /home/yahboom/.ros/log/af416542-ab3d-11f0-815d-c93eed972c40/ros-la
...
yahboom@yahboom-virtual-machine:~$ roslaunch imu_ros1_device yimu_driver.py
Yimu Serial Opened! Baudrate=115200
[INFO] [1760694402.678295]: Open Yimu port OK:/dev/myimu
-----create receive threading-----
ros_comm version: 1.0.0

SUMMARY
=====
PARAMETERS
* /roscore
* /roslaunch
* /rosversion
* /rostopic

NODES
auto-starting process[master]
ROS_MASTER_URI=http://localhost:11311

y: 0.0
z: 0.0
angular_velocity_covariance:
linear_acceleration:
x: -0.104983672597
y: 0.0615253151036
z: 0.995147556993
linear_acceleration_covariance:
header:
seq: 14
stamp:
secs: 1760694701
nsecs: 35817623
frame_id: "imu_link"
magnetic_field:
x: 4.297006134220405
y: -30.664998321481978
z: -15.308084353160192
magnetic_field_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
orientation:
x: -0.012211663648
y: 0.0603983104228
z: 0.6696860790252
w: 0.7378016710281
orientation_covariance:
angular_velocity:
x: 0.0106529694631
y: 0.0
z: 0.0
magnetic_field:
x: 4.297006134220405
y: -30.469679860835598
z: -15.43015839106418
magnetic_field_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```

## 5. Raspberry Pi 5, RDK X5 series, Jetson series

**Note:** Steps 2. Binding device ID and 3. Installing driver libraries need to be completed beforehand.

### 5.1 Installing Docker

1. Docker official website installation reference manual: <https://docs.docker.com/engine/install/ubuntu/>
2. Use the following command for one-click installation:

**Note:** Docker installation failure is normal. You can try multiple times, or install it using a proxy. You need to search online for instructions on using a proxy.

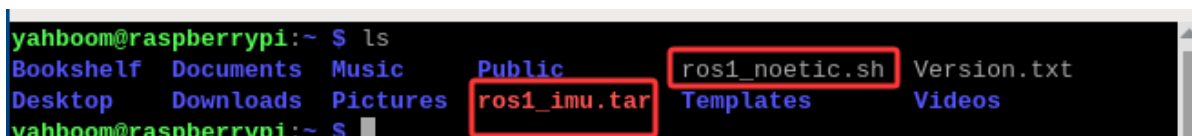
```
curl -fsSL https://get.docker.com | bash -s docker --mirror Aliyun
```

3. Check the Docker version

```
sudo docker version
```

### 5.2 Importing the Docker Image

Upload all files in the ROS1\_noetic folder within the Docker image folder in the attached folder to the main controller. Then run the following command to import it.



```
yahboom@raspberrypi:~$ ls
Bookshelf  Documents  Music      Public      ros1_noetic.sh  Version.txt
Desktop    Downloads  Pictures   ros1_imu.tar  Templates       Videos
yahboom@raspberrypi:~$
```

```
docker load -i ros1_imu.tar
```

**Note:** Importing the image 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 ros1_noetic.sh
```

## 5.4 Print IMU Data

Start roscore

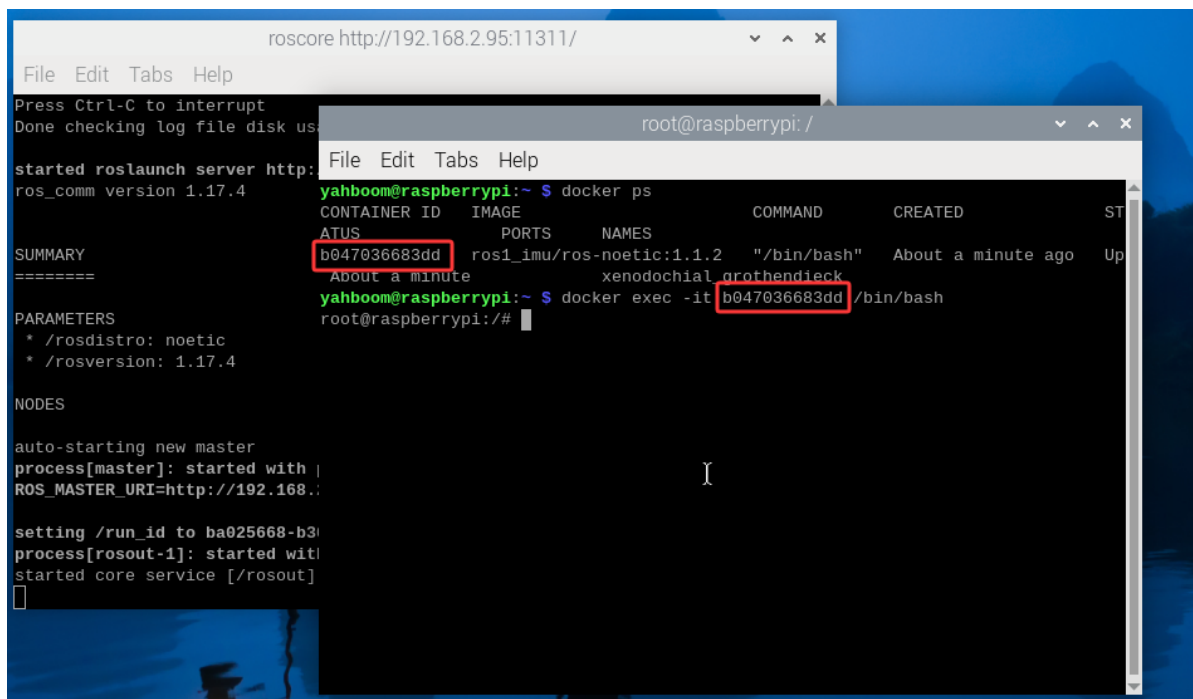
```
roscore
```

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
```



Start the IMU node

```
roslaunch imu_ros1_device ybimu_driver.py
```

```
File Edit Tabs Help
yahboom@raspberrypi:~ $ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS
b047036683dd   ros1_imu/ros-noetic:1.1.2          "/bin/bash"             About a minute Up
About a minute   xenodochial_grothendieck
yahboom@raspberrypi:~ $ docker exec -it b047036683dd /bin/bash
root@raspberrypi:/# rosrn imu_ros1_device ybimu_driver.py
YbImu Serial Opened! Baudrate=115200
[INFO] [1761552877.065925]: Open Ybimu Port OK:/dev/myimu
-----create receive threading-----
```

Open a third terminal, refer to the second terminal for multi-terminal access to the same Docker container.

```
docker ps
```

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

Check IMU topics

```
rostopic list
```

```
bash: rosrn: command not found
yahboom@raspberrypi:~ $ docker exec -it b047036683dd /bin/bash
root@raspberrypi:/# rostopic list
/baro
/euler
/imu/data_raw
/imu/mag
/rosout
/rosout_agg
root@raspberrypi:/#
```

Print /imu/data\_raw topic data

```
rostopic echo /imu/data_raw
```

```

root@raspberrypi:/# rostopic echo /imu/data_raw
header:
  seq: 1
  stamp:
    secs: 1761552994
    nsecs: 298794507
  frame_id: "imu_link"
orientation:
  x: -0.45180225372314453
  y: 0.022018495947122574
  z: -0.35680705308914185
  w: 0.8152952194213867
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:
  x: 0.29639576403088475
  y: -0.78225043488876
  z: 0.6040223395489365
linear_acceleration_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]

```

Print /imu/mag topic data

```
rostopic echo /imu/mag
```

```

root@raspberrypi:/# rostopic echo /imu/mag
header:
  seq: 1
  stamp:
    secs: 1761553077
    nsecs: 598796129
  frame_id: "imu_link"
magnetic_field:
  x: 7.5197607348857085
  y: -16.577654347361676
  z: -1.6113773003326517
magnetic_field_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]

```