

# Environment build

This course explains how to build a ten-axis IMU module environment in ros2, including compiling function packages, binding serial ports, running programs and viewing data. This section takes ubuntu20.04+ros-foxy, the function package space name is WitImu\_ws as an example, and the default baud rate is 9600.

## 1、Compile feature pack

Create a workspace WitImu\_ws and create a new src folder in this directory to store the function package. Terminal input,

```
mkdir WitImu_ws
cd WitImu_ws
mkdir src
```

Then unzip the file to get the wit\_ros2\_imu folder, copy it to the src directory just created. Next go back to the workspace directory, use the colcon build command to compile, and enter in the terminal,

```
cd ~/WitImu_ws
colcon build
```

Finally, add the path of the workspace to .bashrc. Terminal input,

```
sudo vim ~/.bashrc
source ~/WitImu_ws/install/setup.bash #Add this sentence at the end, here my
workspace is in the ~ directory, modify it according to your own workspace directory
```

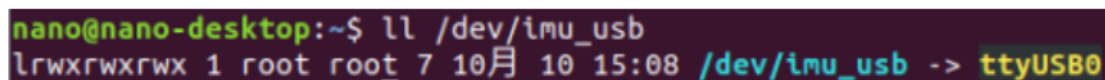
## 2、Bind serial port

In order to prevent the system from identifying errors when multiple usb devices are plugged in at the same time, we bind the serial port name of the module to /dev/imu\_usb. Enter the terminal,

```
cd ~/WitImu_ws/src/wit_ros_imu
sudo chmod 777 bind_usb.sh
sudo sh bind_usb.sh
```

Plug and unplug the USB data cable connected to the IMU module again. To effectively bind the port, enter the following command to check whether the binding port is successful,

```
ll /dev/imu_usb
```



```
nano@nano-desktop:~$ ll /dev/imu_usb
lrwxrwxrwx 1 root root 7 10月 10 15:08 /dev/imu_usb -> ttyUSB0
```

Note: It does not have to be ttyUSB0, as long as it is displayed as a USB device.

### 3、run test

Terminal input,

```
ros2 launch wit_ros2_imu rviz_and_imu.launch.py
```

```
pieyahboom:~/cartographer_ws2$ ros2 launch wit_ros2_imu rviz_and_imu.launch.py
[INFO] [launch]: All log files can be found below /home/pi/.ros/log/2023-03-15-15-44-44-100570-yahboom-310298
[INFO] [launch]: Default logging verbosity is set to INFO
/home/pi/cartographer_ws2/install/wit_ros2_imu/share/wit_ros2_imu/rviz_and_imu.launch.py:6: UserWarning: The parameter 'node_executable' is deprecated, use '
executable' instead
  rviz_and_imu_node = Node(
/home/pi/cartographer_ws2/install/wit_ros2_imu/share/wit_ros2_imu/rviz_and_imu.launch.py:6: UserWarning: The parameter 'node_name' is deprecated, use 'name'
instead
  rviz_and_imu_node = Node(
/home/pi/cartographer_ws2/install/wit_ros2_imu/share/wit_ros2_imu/rviz_and_imu.launch.py:17: UserWarning: The parameter 'node_executable' is deprecated, use
'executable' instead
  rviz_display_node = Node(
[INFO] [wit_ros2_imu-1]: process started with pid [310525]
[wit_ros2_imu-1] [INFO] [1678866290.233370396] [imu]: Serial port opened successfully...
```

Use the “ros2 topic echo” tool to see the specific content of the published data. Terminal input,

```
ros2 topic echo /imu/data
```

```
header:
  stamp:
    sec: 1678866329
    nanosec: 904145008
    frame_id: imu_link
orientation:
  x: -0.0063128977334605246
  y: 0.03806125383901889
  z: -0.028982260050985614
  w: 0.9988350799209775
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
```

### 4、Modify the baud rate

By default, the program uses a baud rate of 9600. If the baud rate is modified on the host computer, then the baud rate in the source code needs to be modified. The location of the source code to modify the baud rate is ~/WitImu\_ws/src/wit\_ros2\_imu/wit\_ros2\_imu/wit\_ros2\_imu.py

```
#line 149
def driver_loop(self, port_name):
    # open serial port
    try:
        wt_imu = serial.Serial(port="/dev/imu_usb", baudrate=9600, timeout=0.5)
```

Change 9600 to the baud rate modified on the host computer, then save and exit, and finally return to the workspace directory to compile.