

# 1. Environment setup

This course takes **Ubuntu20.04+ros-foxy** as an example to explain how to build an environment and use a camera in a ros2 environment.

## 1. Install related dependencies

Terminal input,

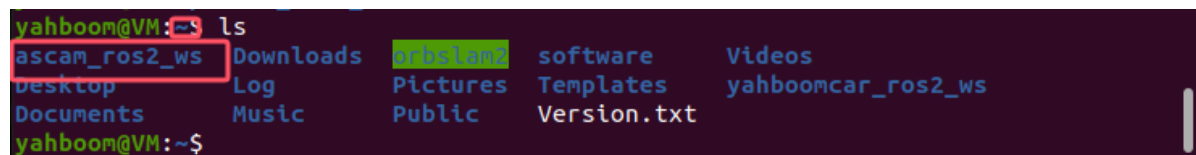
```
sudo apt install libgflags-dev nlohmann-json3-dev libgoogle-glog-dev ros-foxy-image-transport ros-foxy-image-publisher
```

Here, foxy\*\* is modified according to the actual ros2 version\*\*. If it is humble, just change it to humble.

## 2. Compile the function package

### 1), copy the workspace

Unzip the file and copy ascam\_ros2\_ws to the root directory

A terminal window showing the command 'ls' being executed. The output lists several directories: 'ascam\_ros2\_ws' (highlighted with a red box), 'Downloads', 'Log', 'Desktop', 'Documents', 'Pictures', 'Music', 'Public', 'software', 'Templates', 'Version.txt', 'Videos', and 'yahboomcar\_ros2\_ws'. The prompt is 'yahboom@VM:~\$'.

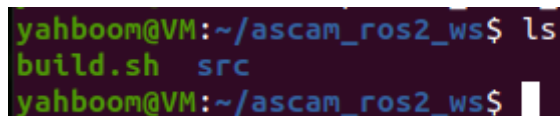
### 2), compile

Enter the ascam\_ros2\_ws workspace

```
cd ~/ascam_ros2_ws
```

Check the build.sh file

```
ls
```

A terminal window showing the command 'ls' being executed in the directory ~/ascam\_ros2\_ws. The output lists 'build.sh' and 'src'. The prompt is 'yahboom@VM:~/ascam\_ros2\_ws\$'.

Add running permissions

```
chmod a+x build.sh
```

Compile

```
./build.sh
```

When the prompt below appears, run the compilation command again until the prompt is successful

```

    from /usr/local/include/pangolin/display/viewport.h:31,
    from /usr/local/include/pangolin/gl/gl.h:30,
    from /usr/local/include/pangolin/pangolin.h:33,
    from /home/yahboom/orbslam2/ORB_SLAM2/include/MapDrawer.h:27,
    from /home/yahboom/orbslam2/ORB_SLAM2/include/Viewer.h:26,
    from /home/yahboom/orbslam2/ORB_SLAM2/include/Tracking.h:28,
    from /home/yahboom/orbslam2/ORB_SLAM2/include/System.h:29,
    from /home/yahboom/ascam_ros2_ws/src/ros2-ORB_SLAM2/src/stereo/
stereo-slam-node.hpp:13,
    from /home/yahboom/ascam_ros2_ws/src/ros2-ORB_SLAM2/src/stereo/
stereo-slam-node.cpp:1:
/usr/include/eigen3/Eigen/src/Core/util/Constants.h:162:37: note: declared here
 162 | EIGEN_DEPRECATED const unsigned int AlignedBit = 0x80;
    |                                     ~~~~~
---
Finished <<< ros2_orbslam [3min 2s]
Summary: 8 packages finished [3min 2s]
 5 packages had stderr output: ascamera ros2_orbslam yahboomcar_kcftracker yahb
oomcar_msgs yahboomcar_slam

```

```

yahboom@VM:~/ascam_ros2_ws$ ./build.sh
Starting >>> ascamera
Starting >>> ros2_orbslam
Starting >>> yahboomcar_astra
Starting >>> yahboomcar_kcftracker
Finished <<< ros2_orbslam [0.57s]
Starting >>> yahboomcar_mediapipe
Finished <<< ascamera [0.64s]
Starting >>> yahboomcar_msgs
Finished <<< yahboomcar_kcftracker [0.90s]
Starting >>> yahboomcar_slam
Finished <<< yahboomcar_slam [0.38s]
Starting >>> yahboomcar_visual
Finished <<< yahboomcar_astra [1.67s]
Finished <<< yahboomcar_msgs [1.20s]
Finished <<< yahboomcar_mediapipe [1.67s]
Finished <<< yahboomcar_visual [1.35s]

Summary: 8 packages finished [2.91s]
yahboom@VM:~/ascam_ros2_ws$

```

**Note:** If you use other architecture platforms, please open build.sh and modify it to your own architecture platform. If orbslam errors occur, you need to install orbslam related function packages first. Or remove orbslam related function packages and compile first. The supporting virtual machine does not need to be installed

### 3), add environment variables

Terminal input,

```
echo "source ~/ascam_ros2_ws/install/setup.bash" >> ~/.bashrc
```

## 3, install udev rules

Terminal input,

```
cd ~/ascam_ros2_ws/src/ascamera/scripts
sudo bash create_udev_rules.sh
```

```

yahboom@VM:~/ascam_ros2_ws/src/ascamera/scripts$ sudo bash create_udev_rules.sh
copy /home/yahboom/ascam_ros2_ws/src/ascamera/scripts/angstrong-camera.rules to
/etc/udev/rules.d
reload udev rules
reload udev rules done

```

If the above picture appears, it means success.

## 4. Modify the configuration file path

Check the location of the configuration file

```

cd ~/ascam_ros2_ws/src/ascamera/configurationfiles
pwd

```

```

yahboom@VM:~$ cd ~/ascam_ros2_ws/src/ascamera/configurationfiles
yahboom@VM:~/ascam_ros2_ws/src/ascamera/configurationfiles$ pwd
/home/yahboom/ascam_ros2_ws/src/ascamera/configurationfiles
yahboom@VM:~/ascam_ros2_ws/src/ascamera/configurationfiles$

```

Modify the launch file

```

cd ~/ascam_ros2_ws/src/ascamera/launch
gedit hp60c.launch.py

```

```

1 from launch import LaunchDescription
2 from launch_ros.actions import Node
3 from launch.actions import DeclareLaunchArgument
4 from launch.substitutions import LaunchConfiguration
5
6 def generate_launch_description():
7     ld = LaunchDescription()
8     ascamera_node = Node(
9         namespace= "ascamera_hp60c",
10        package='ascamera',
11        executable='ascamera_node',
12        respawn=True,
13        output='both',
14        parameters=[
15            {"usb_bus_no": -1},
16            {"usb_path": "null"},
17            {"confiPath": "/home/yahboom/ascam_ros2_ws/src/ascamera/-
configurationfiles"},
18            {"color_pcl": True},
19            {"pub_tfTree": True},
20            {"depth_width": 640},
21            {"depth_height": 480},
22            {"rgb_width": 640},
23            {"rgb_height": 480},
24            {"fps": 25},
25        ],
26        remappings=[]
27    )
28
29    # ascamera_node2 = Node(
30    #     namespace= "ascamera_hp60c_2",
31    #     package='ascamera',
32    #     executable='ascamera_node',
33    #     respawn=True,
34    #     output='both',
35    #     parameters=[
36    #         {"usb_bus_no": 3},    # set your usb_bus_no

```

Recompile

```
cd ~/ascam_ros2_ws  
./build.sh
```

Close the terminal after compiling

## 5. Run the camera and view the image

Open a new terminal and enter,

```
#Camera launch  
ros2 launch ascamera hp60c.launch.py
```

Enter the following command to view the topic information,

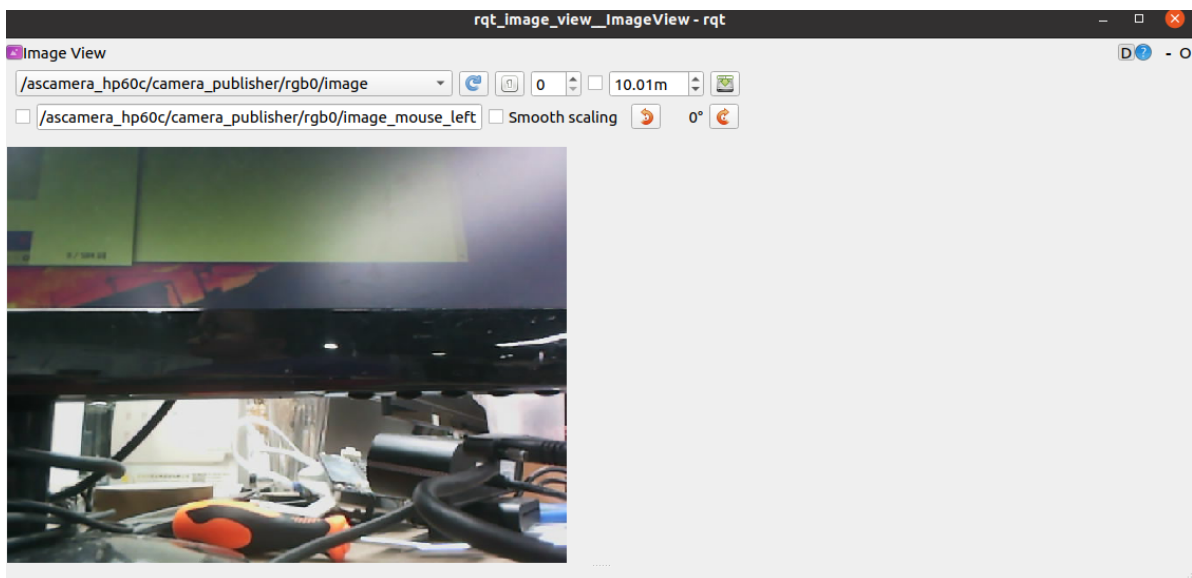
```
ros2 topic list
```

```
yahboom@VM:~$ ros2 topic list  
/ascamera_hp60c/camera_publisher/depth0/camera_info  
/ascamera_hp60c/camera_publisher/depth0/image_raw  
/ascamera_hp60c/camera_publisher/depth0/points  
/ascamera_hp60c/camera_publisher/rgb0/camera_info  
/ascamera_hp60c/camera_publisher/rgb0/image  
/parameter_events  
/rosout  
yahboom@VM:~$
```

Use the rqt\_image\_view tool to view the image, terminal input,

```
ros2 run rqt_image_view rqt_image_view
```

Color image



Depth image



Select the topic in the upper left corner.

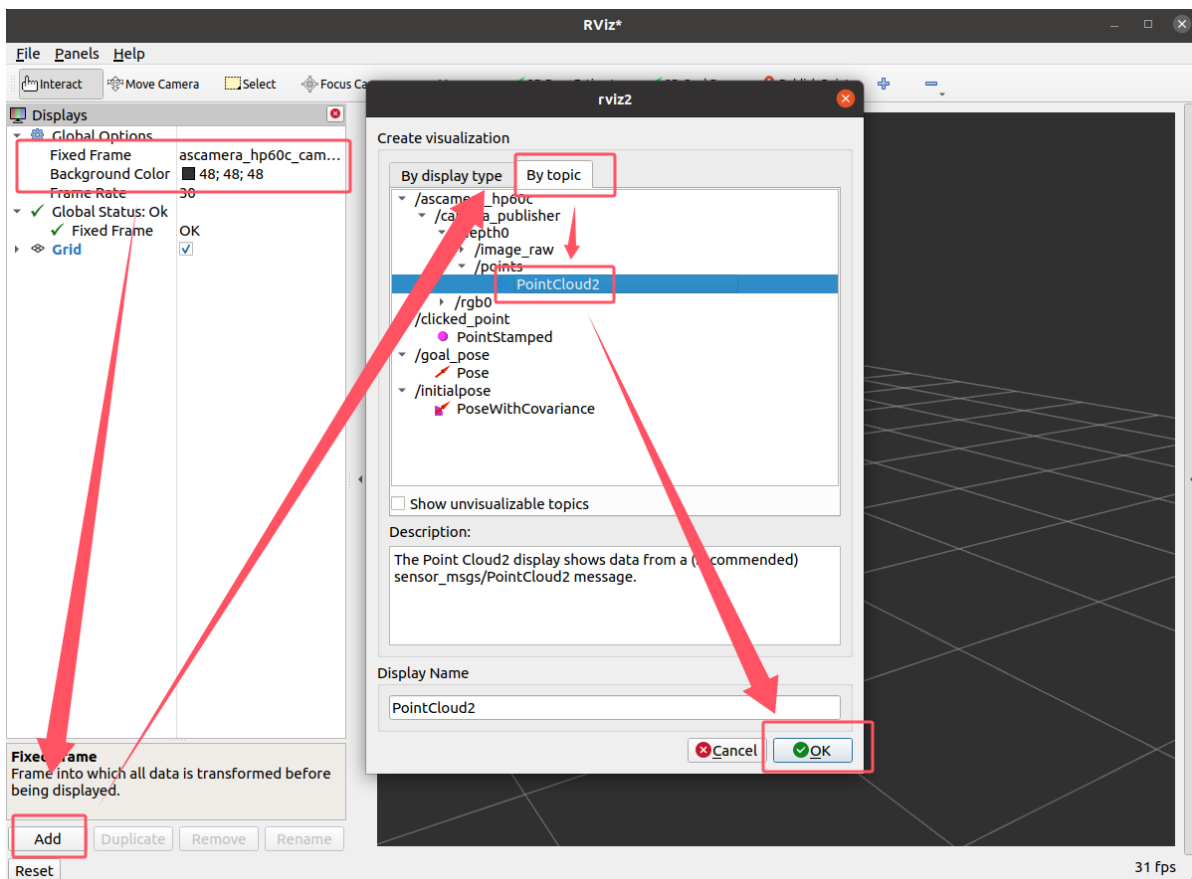
## 6. Run the camera and view the point cloud image

Terminal input,

```
ros2 launch ascamera hp60c.launch.py
```

You can see the point cloud data released by the camera in rviz, terminal input,

```
rviz2
```



After opening rviz, set the visualization point cloud data as shown in the figure above.

