

# Lekiwi for Nomad 使用文档

## 读前参考

可以先看看 [📖Nomad 代码部署说明](#) 以及 [📖Create3 底盘部署说明](#)，我是根据这个进行的适配  
所以我们需要三个硬件

- 操控 Lekiwi 的树莓派
- 运行 Nomad 的服务器
- 以及操控机器人的本机 PC

为了能够让三个Node互相通讯，需要配置 Fast DDS Discovery Server

[https://iroboteducation.github.io/create3\\_docs/setup/discovery-server/](https://iroboteducation.github.io/create3_docs/setup/discovery-server/)

## 硬件修改

在nomad 论文里面摄像头需要抬高获得更多信息，所以不能使用 lekiwi 原来的位置。我使用了2020的铝型材进行架高。



其中使用的零件有 2020 铝型材的stand <https://makerworld.com/en/models/799661-angles-for-aluminum-profiles-2020-3030-4040-4545#profileId-739497>

以及 camera mount



2020 camera mount v4.stl

110.04KB



## 查看camera

确认一下是 video 多少，查看支持的格式和分辨率。我们要使用 YUYV 这个格式

```
v4l2-ctl -d /dev/video2 --list-formats-ext
```

记住支持的分辨率，之后需要用到

Code block

```
1  [1]: 'YUYV' (YUYV 4:2:2)
2          Size: Discrete 1280x720
3          Interval: Discrete 0.100s (10.000 fps)
4          Interval: Discrete 0.200s (5.000 fps)
5          Size: Discrete 800x600
6          Interval: Discrete 0.067s (15.000 fps)
7          Interval: Discrete 0.100s (10.000 fps)
8          Interval: Discrete 0.200s (5.000 fps)
9          Size: Discrete 640x480
10         Interval: Discrete 0.033s (30.000 fps)
11         Interval: Discrete 0.040s (25.000 fps)
12         Interval: Discrete 0.050s (20.000 fps)
13         Interval: Discrete 0.067s (15.000 fps)
14         Interval: Discrete 0.100s (10.000 fps)
15         Interval: Discrete 0.200s (5.000 fps)
```

因为用到的分辨率不高，选择 640x480 10 FPS 就好。但是不是所有的摄像头都支持，选择一个合适的就好。

## 安装树莓派ros2环境

### 配置 NTP 服务器校准时间

⚠ 特殊说明：

1. ROS2 依赖时间同步，因此需要确保 NTP 服务器正常（清华校园网需要专用 NTP 服务器）

1. 确保 chrony 已经安装。如果未安装使用 `sudo apt install chrony`

之后运行 `sudo nano /etc/chrony/chrony.conf`，按如下编辑

#### Code block

```
1  #pool ntp.ubuntu.com          iburst maxsources 4
2  #pool 0.ubuntu.pool.ntp.org iburst maxsources 1
3  #pool 1.ubuntu.pool.ntp.org iburst maxsources 1
4  #pool 2.ubuntu.pool.ntp.org iburst maxsources 2
5  server ntp.tuna.tsinghua.edu.cn iburst
6  server 166.111.8.28 iburst
7  server 166.111.8.29 iburst
```

2. 保存后执行 `systemctl restart chrony`
3. 同步时间 `sudo chronyc makestep`，并使用 `date` 来确保时间正确（如果是UTC时间，则慢8小时）
4. 执行 `chronyc sources -v`，应当出现三行 IP 地址，并且数据不为空，IP 地址前不应出现 `^?` 的符号

## git clone ros2 workspace

在这里包括了操控 lekiwi 底盘和发布 odom 的节点，clone 在树莓派上用它操控机器人

[https://github.com/EmbodiedLLM/lekiwi\\_nomad\\_pi](https://github.com/EmbodiedLLM/lekiwi_nomad_pi)  
github.com

## 修改 launch 文件

使用的 launch 文件在

`src/lekiwi_controller_cmd/launch/lekiwi_complete.launch.py` 路径

根据上面的摄像头分辨率修改 launch 文件，根据自己摄像头支持的格式设置

#### Code block

```
1  Node(
2      package='v4l2_camera',
3      executable='v4l2_camera_node',
4      name='v4l2_camera_node',
5      parameters=[
6          {'time_per_frame': 0.1},
7          {'pixel_format': 'YUYV'},
8          {'image_width': 640}, # 设置图像宽度
9          {'image_height': 480} # 设置图像高度
10     ]
11 )
```

## 构造 Docker

在树莓派上使用 docker 运行 ros2 humble

首先安装docker 按照文档 <https://docs.ros.org/en/humble/How-To-Guides/Installing-on-Raspberry-Pi.html#>

如果要构建镜像，可以使用 ros humble perception

<https://hub.docker.com/layers/library/ros/humble-perception/images/sha256-aad34eb5ac95fbc95852d30f35da2138916...>  
**hub.docker.com**

（目前还没法把镜像推送过去）我的镜像名称是 ros2\_lekiwi

Git clone 之后挂载这个文件夹，比如 lekiwi\_nomad\_pi, 使用 pwd 找到路径，写在： 前面

镜像已经推送了 [https://hub.docker.com/repository/docker/acyanbird/ros2\\_lekiwi\\_pi/general](https://hub.docker.com/repository/docker/acyanbird/ros2_lekiwi_pi/general)

Code block

```
1 docker run -it --rm --entrypoint /bin/bash --net=host --privileged -v  
  /dev:/dev -v /home/acy/lekiwi_nomad_pi:/lekiwi_nomad_pi  
  acyanbird/ros2_lekiwi_pi
```

## 配置 DDS server

首先在 docker **之外** 查看本机的 ip `ifconfig`

在进入docker **之后** 修改bashrc `vim ~/.bashrc` 在下面加入如下几行

Code block

```
1 export ROS_DOMAIN_ID=42  
2 export RMW_IMPLEMENTATION=rmw_fastrtps_cpp  
3 export  
  FASTRTPS_DEFAULT_PROFILES_FILE=/lekiwi_ws/super_client_configuration_file.xml
```

DOMAIN\_ID 是唯一识别编号，有这个就不会和其他机器人网络搞混。Xml 文件在 workspace 里有，记得根据上文拿到的 IP 修改

Code block

```
1 <?xml version="1.0" encoding="UTF-8" ?>  
2 <dds>  
3   <profiles xmlns="http://www.eprosima.com/XMLSchemas/fastRTPS_Profiles">  
4     <participant profile_name="super_client_profile"  
      is_default_profile="true">
```

```

5         <rtps>
6             <builtin>
7                 <discovery_config>
8                     <discoveryProtocol>SUPER_CLIENT</discoveryProtocol>
9                     <discoveryServersList>
10                         <RemoteServer
11                             prefix="44.53.00.5f.45.50.52.4f.53.49.4d.41">
12                                 <metatrafficUnicastLocatorList>
13                                     <locator>
14                                         <udp4>
15                                             <address>把这里替换成IP! </address>
16                                             <port>11811</port>
17                                         </udp4>
18                                     </locator>
19                                 </metatrafficUnicastLocatorList>
20                             </RemoteServer>
21                         </discoveryServersList>
22                     </discovery_config>
23                 </builtin>
24             </rtps>
25         </participant>
26     </profiles>
27 </dds>

```

之后开启 server，在 Docker 里面输入

```
fastdds discovery -i 0 -l <SERVER_IP> -p 11811
```

就是上文获得的 IP 之后会得到这样的输出

Code block

```

1 root@acypi:/lekiwi_ws# fastdds discovery -i 0 -l 10.90.0.159 -p 11811
2 ### Server is running ###
3 Participant Type:  SERVER
4 Security:         NO
5 Server ID:        0
6 Server GUID prefix: 44.53.00.5f.45.50.52.4f.53.49.4d.41
7 Server Addresses:  UDPv4:[10.90.0.159]:11811

```

不要关掉这个窗口，这个需要在机器人运行期间一直开启，没有任何输出是正常的。

## 开启节点

之后需要另外开启一个终端，来运行节点。在其他连接到树莓派 ssh 的 terminal 里面使用 docker exec 进入。首先通过 `docker ps` 获得 id

Code block

```
1 docker ps
2 CONTAINER ID   IMAGE                COMMAND              CREATED          STATUS
   PORTS        NAMES
3 87399c85d407   ros2_lekiwi        "/bin/bash"         About an hour ago Up About an
   hour          upbeat_brown
4
```

把第一个的 ID 使用

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

进入容器

使用 launch 文件开启节点，如果没有安装 v4l2 的包请记得安装

Code block

```
1 cd lekiwi_nomad_pi
2 colcon build
3 source install/setup.bash
4 ros2 launch lekiwi_controller_cmd lekiwi_complete.launch.py
```

为了验证节点是否开启需要在 PC 端使用 keyboard 操控验证。以及如果更换了其他 camera，请确保它支持 10FPS 的输出。

## 安装本地 PC

在本地请使用 ubuntu 22.04 安装 ros2

<https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debs.html#install-ros-2-packages>

请记得按照上文修改 bashrc

Code block

```
1 export ROS_DOMAIN_ID=42
2 export RMW_IMPLEMENTATION=rmw_fastrtps_cpp
3 export
   FASTRTPS_DEFAULT_PROFILES_FILE=/lekiwi_ws/super_client_configuration_file.xml
```

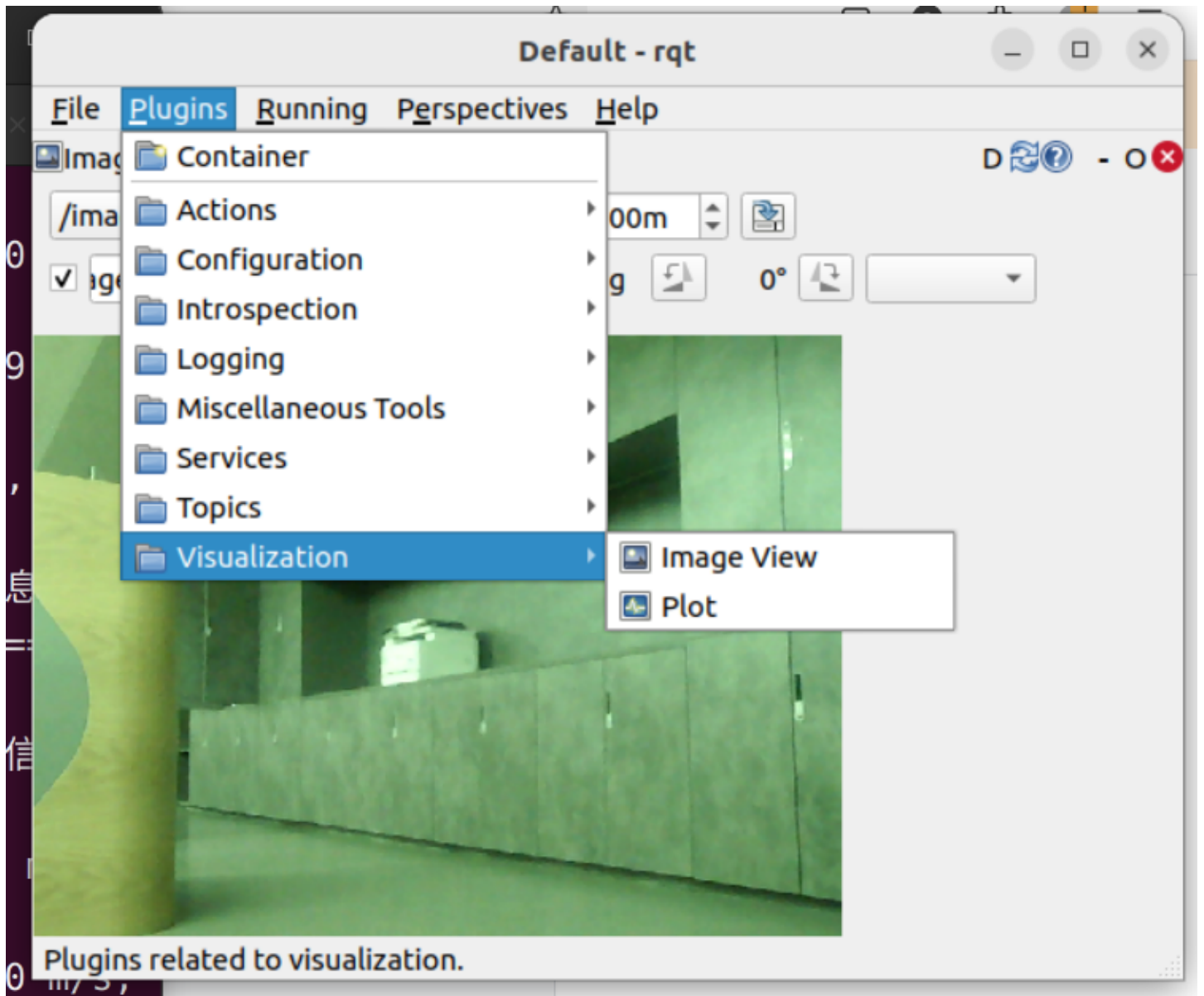
也在本地 clone [https://github.com/EmbodiedLLM/lekiwi\\_nomad\\_pi](https://github.com/EmbodiedLLM/lekiwi_nomad_pi)

进入目录里

Code block

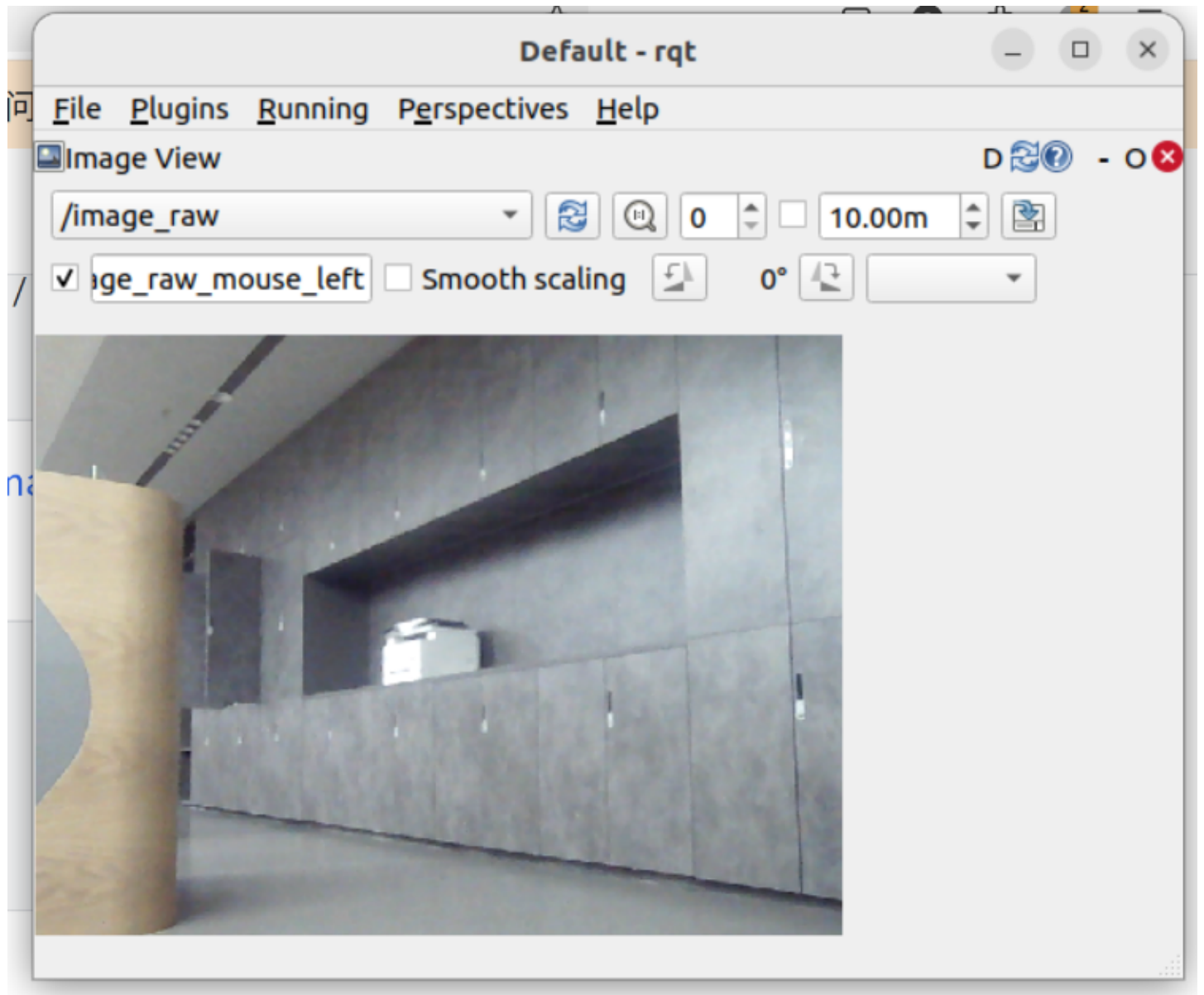
```
1 colcon build
2 source install/setup.bash
3 ros2 run lekiwi_controller_cmd keybroad
```

如果可以移动证明通讯成功。也可以新开个终端输入 rqt



查看 rqt 是否有图像输出，顺便记住 topic。





这个 /image\_raw 会在 服务器端使用到

## 配置 GPU 服务器

在这里使用 <https://github.com/RobotecAI/visualnav-transformer-ros2> 代码。记得修改 super\_client\_configuration\_file.xml 中的 IP field。进入 docker(到时候要推一下) dev -v 之后的路径是你下载了 vint 仓库的路径

Code block

```
1 docker run -it --rm --entrypoint /bin/bash --net=host --gpus=all --privileged -v /dev:/dev -v /nvme/qwen/visualnav-transformer-ros2:/visualnav-transformer-ros2 acyanbird/nomad_lekiwi
```

进入 docker 之后记得也修改bashrc 文件,检查一下最后的路径

Code block

```
1 export ROS_DOMAIN_ID=42
2 export RMW_IMPLEMENTATION=rmw_fastrtps_cpp
3 export
  FASTRTPS_DEFAULT_PROFILES_FILE=/lekiwi_ws/super_client_configuration_file.xml
```

## 机器相关配置

在 config 文件夹里修改 camera\_front 和 robot

Code block

```
1 # camera parameters for src/gnm_locobot.launch
2 video_device: "/dev/video0" # change this to your video device path
3 image_width: 480
4 image_height: 640
5 pixel_format: yuyv
6 camera_frame_id: "image_raw"
7 io_method: "mmap"
8 framerate: 30
```

里面的 camera\_frame\_id 是摄像头广播的 frame name

之后在 robot 这里修改最高的线性速度和最高角速度

Code block

```
1 # linear and angular speed limits for the robot
2 max_v: 0.2 # m/s
3 max_w: 0.1 # rad/s
4 # observation rate fo the robot
5 frame_rate: 6 # Hz
6 graph_rate: 0.3333 # Hz
7
8 # topic names (modify for different robots/nodes)
9 vel_teleop_topic: /cmd_vel_mux/input/teleop
10 vel_navi_topic: /cmd_vel_mux/input/navi
11 vel_recovery_topic: /cmd_vel_mux/input/recovery
```

## 修改 create topomap

在 `src/visualnav_transformer/deployment/src/create_topomap.py` 里面添加变量 `IMAGE_TOPIC = "/image_raw"` 这个是camera publish图片的 topic，根据你的 topic 修改

## 训练数据

进入镜像之后应该要切换 `conda activate nomad_train`

## 修改 docker

Docker 在每次退出之后不会保留你的修改，如果想要保存镜像请使用 docker commit 功能。另外开启一个终端，在 docker 外面进行 commit

Code block

```
1 docker commit ID <名称>:tag
```

## 开始使用

### 树莓派

首先**连接上树莓派**，获得树莓派的 ip。进入 docker

```
docker run -it --rm --entrypoint /bin/bash --net=host --privileged -v /dev:/dev -v /home/acy/lekiwi_ws/:/lekiwi_ws ros2_lekiwi
```

开启 DDS server

```
fastdds discovery -i 0 -l <SERVER_IP> -p 11811
```

另外开一个终端进入 docker，用 launch 文件开启节点

```
ros2 launch lekiwi_controller_cmd lekiwi_complete.launch.py
```

### 本机

在**本机**开启 keyboard 节点

```
ros2 run lekiwi_controller_cmd keybroad
```

## GPU 服务器

进入 **GPU 服务器 docker**

Code block

```
1 docker run -it --rm --entrypoint /bin/bash --net=host --gpus=all --privileged -v /dev:/dev -v /nvme/d/qwen/visualnav-transformer-ros2/:/visualnav-transformer-ros2 ros2_lekiwi
```

如果要进行导航需要开启广播/cmd\_vel 的节点，保持开启，拓扑地图不需要

```
python scripts/publish_cmd.py
```

## 创建拓扑地图

Code block

```
1 python src/visualnav_transformer/deployment/src/create_topomap.py --dir  
  <map_name> --dt <1.0>
```

开始导航之后使用你的 keyboard node 来操控机器人行走，他会按照一秒一张的速率拍照（可以dt这里修改）

## 开始导航

我们假设我上一个map name 是 9a1 ,我希望机器人走到第24张图的地方。如果不给参数默认走到最后一个图

Code block

```
1 python src/visualnav_transformer/deployment/src/navigate.py --model nomad --  
  dir 9a1 --goal-node 24
```