How to use ROS

Document link: SLAMWARE ROS SDK

Aurora is a newly developed sensor for integrated positioning and mapping created by SLAMTEC that integrates laser, vision, inertial navigation and deep learning technologies. The sensor does not require external dependencies and can provide six-degree-of-freedom positioning capabilities for indoor and outdoor three-dimensional high-precision mapping machines as soon as it is turned on.

1. Development environment requirements

Based on Ubuntu 20.04 / 22.04 operating system, and installed with ROS2 software package.

2. Download radar SDK

Download link: SLAMTEC Product Documents Download and Technical Support

Four versions of SDK have been placed in the attachment. Use the corresponding version according to the gcc version of your environment.

Aurora ROS2 SDK contains resources and codes that you may use during development. Its directory structure is organized as follows:

Directory	Description
docs	Reference Documents
src	Source Code
--slamware_ros_sdk	ROS SDK Source Code Package
--slamware_sdk	SDK-related header files and library files
--aurora_remote_public	Aurora-related header files and library files

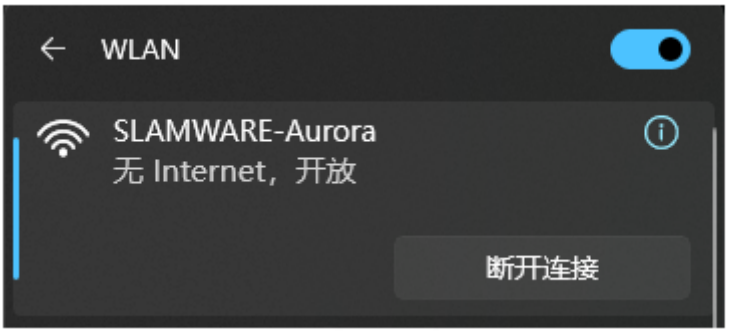
Terminal input, the following content can view the gcc version

```
gcc -v
```

```
yahboom@yahboom-VM:~$ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/11/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none:amdgcn-amdhsa
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu 11.4.0-1ubuntu1~22.04' --with-bug
url=file:///usr/share/doc/gcc-11/README.Bugs --enable-languages=c,ada,c++,go,brig,d,fortran,objc
,obj-c++,m2 --prefix=/usr --with-gcc-major-version-only --program-suffix=-11 --program-prefix=x8
6_64-linux-gnu- --enable-shared --enable-linker-build-id --libexecdir=/usr/lib --without-include
d-gettext --enable-threads=posix --libdir=/usr/lib --enable-nls --enable-bootstrap --enable-cloc
ale=gnu --enable-libstdcxx-debug --enable-libstdcxx-time=yes --with-default-libstdcxx-abi=new --
enable-gnu-unique-object --disable-vtable-verify --enable-plugin --enable-default-pie --with-sys
tem-zlib --enable-libphobos-checking=release --with-target-system-zlib=auto --enable-objc-gc=aut
o --enable-multiarch --disable-werror --enable-cet --with-arch-32=i686 --with-abi=m64 --with-mul
tilib-list=m32,m64,mx32 --enable-multilib --with-tune=generic --enable-offload-targets=nvptx-non
e=/build/gcc-11-XeT9lY/gcc-11-11.4.0/debian/tmp-nvptx/usr,amdgcn-amdhsa=/build/gcc-11-XeT9lY/gcc
-11-11.4.0/debian/tmp-gcn/usr --without-cuda-driver --enable-checking=release --build=x86_64-lin
ux-gnu --host=x86_64-linux-gnu --target=x86_64-linux-gnu --with-build-config=bootstrap-lto-lean
--enable-link-serialization=2
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 11.4.0 (Ubuntu 11.4.0-1ubuntu1~22.04)
yahboom@yahboom-VM:~$
```

3. Hardware Requirements

To use ROS2 SDK, you need an Auraro space mapping device, turn it on and configure the appropriate IP address. After the slamware_ros_sdk_server_node node is started, it will try to connect to the device. For factory devices, directly connect to the self-heating point of the Aurora radar.



4. Environment setup

If you use the factory virtual machine we provide, or other factory motherboard images, the following environment setup steps do not need to be rebuilt, and you can directly run the map building command.

4.1. Create a workspace first

```
mkdir -p ~/ros2_ws/src
cd ~/ros2_ws/src
```

Put the sdk source code you just downloaded in the src working directory where the source code is stored,

4.2 Compile

```
cd ..
colcon build
```

It is normal to have warnings when compiling slamware_sdk,

```
[13.419s] WARNING:colcon.colcon_cmake.task.cmake.build:Could not run installation step for packa
ge 'slamware_sdk' because it has no 'install' target
```

4.3 Configure the workspace system environment

```
source install/setup.bash
```

In order to automatically load the environment every time the terminal is started: add the following command to ~/.bashrc

```
echo "source ~/ros2_ws/install/setup.bash" >> ~/.bashrc
```

4.4 Configure the system environment for dynamic library for image building

Since aurora_remote_public_lib is a dynamic library, you need to add the platform path to LD_LIBRARY_PATH. For example, if you place slamware_ros2_sdk_linux-x86_64-gcc11 in the ~/ros2_ws/sec folder, you need to add the following command to ~/.bashrc

Note: If you download the SDK for aarch64 environment, the corresponding name should be changed to slamware_ros2_sdk_linux-aarch64-gcc11.

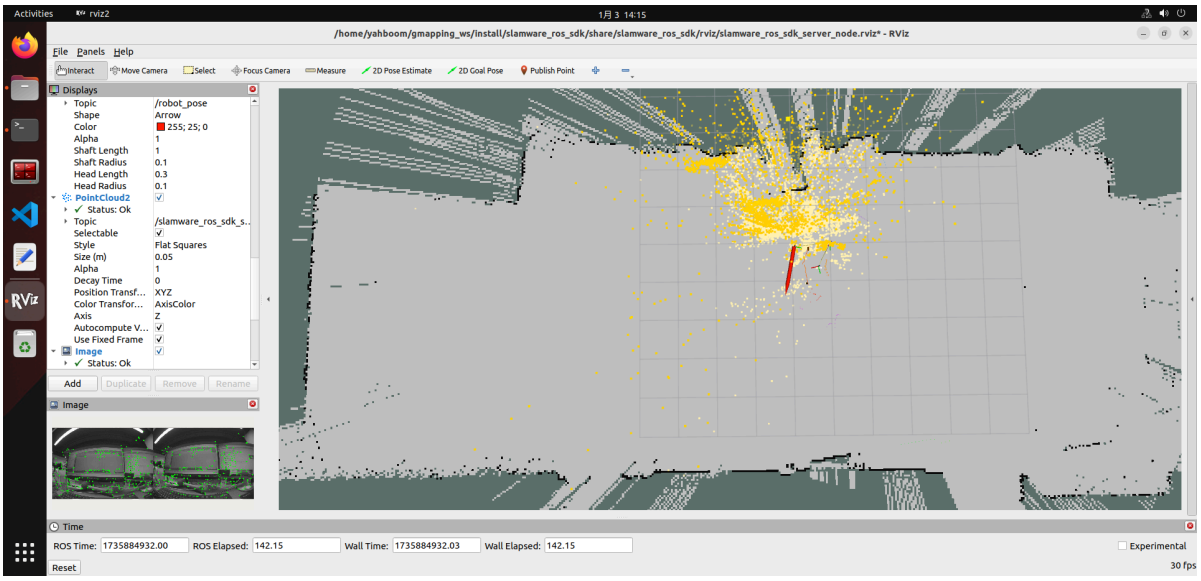
```
export LD_LIBRARY_PATH=~/ros2_ws/src/slamware_ros2_sdk_linux-x86_64-
gcc11/src/aurora_remote_public/lib/linux_x86_64:$LD_LIBRARY_PATH
```

4.5 Start the map node

Virtual machine version:

Open the virtual machine provided by the data. If the Aurora device is in AP mode, connect to the Aurora hotspot and run the following command in the terminal.

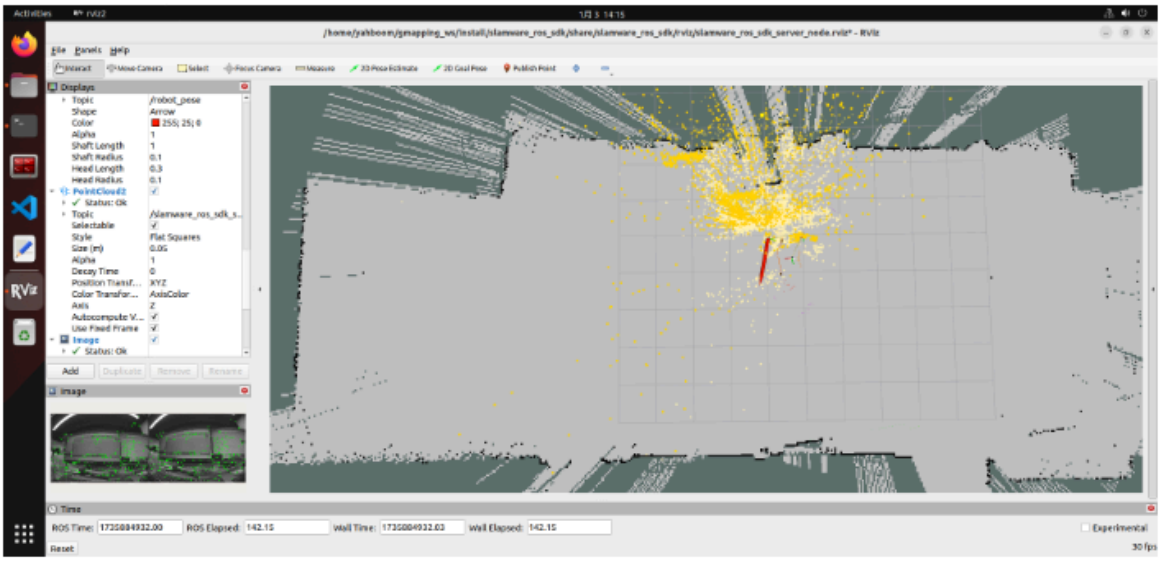
```
ros2 launch slamware_ros_sdk slamware_ros_sdk_server_and_view.xml
ip_address:=192.168.11.1
```



Orin nano version:

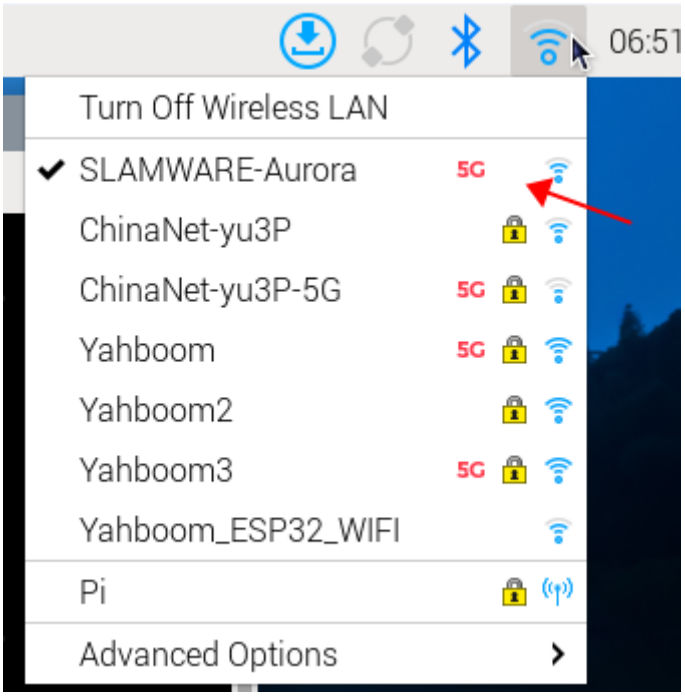
Use the monitor to connect the mainboard to the hotspot emitted by the radar, and enter the following command in the monitor terminal,

```
ros2 launch slamware_ros_sdk slamware_ros_sdk_server_and_view.xml
ip_address:=192.168.11.1
```

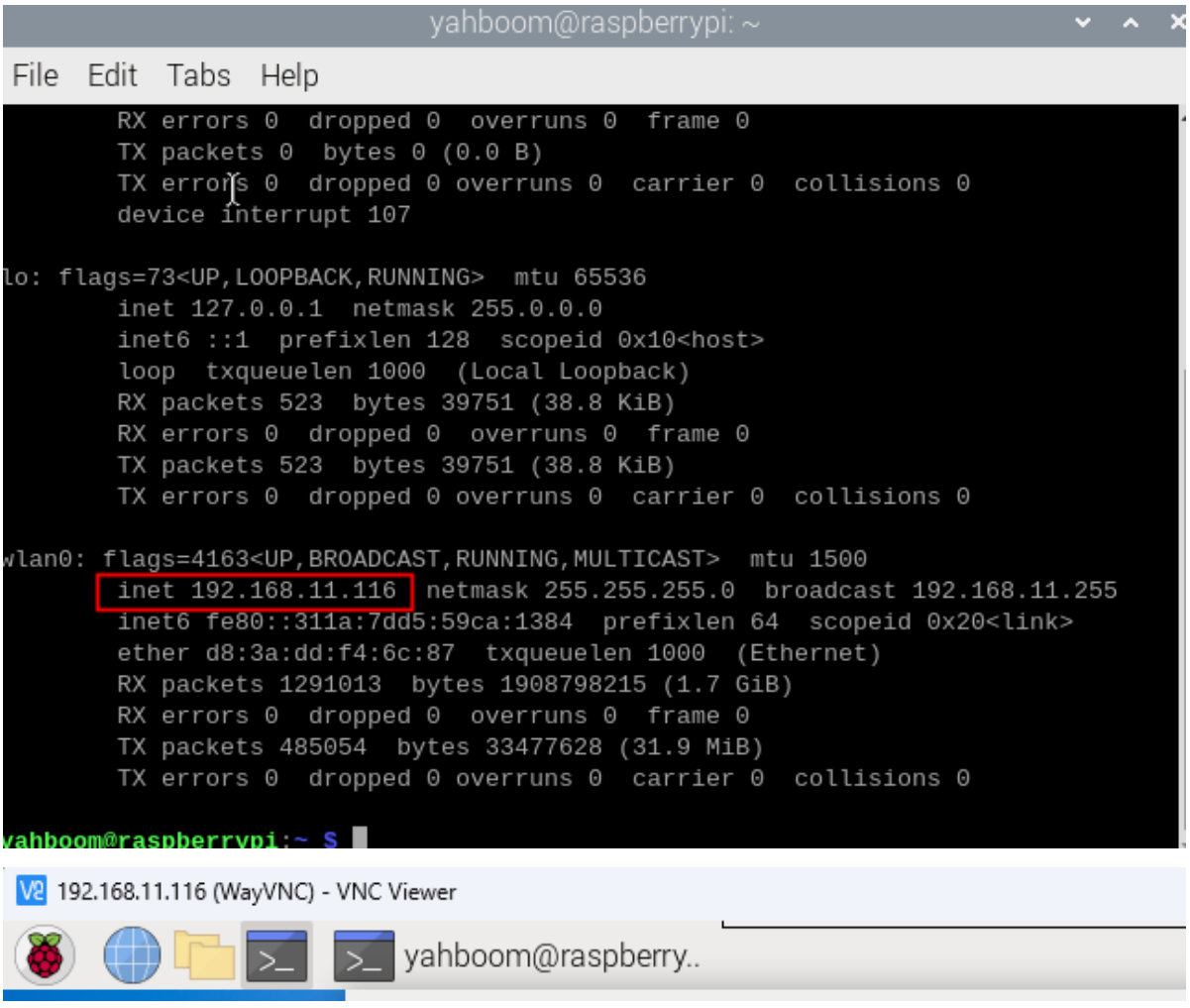


Raspberry Pi 5, X5, Jetson nano version:

If it is the Raspberry Pi desktop version and jetson nano, x5 desktop version, after successfully booting up, connect the external monitor to the hotspot of Aurora radar to the motherboard, as shown below,



Then enter ifconfig in the terminal to view the ip address under the current hotspot, and then use this address for vnc remote desktop connection, **(the premise is that the computer must also be connected to the hotspot emitted by Aurora radar)**



Enter docker in advance and enter in the terminal

```
sh ros2_humble.sh
```

If the following interface appears, it means that you have successfully entered docker.

```
yahboom@raspberrypi:~ $ sh ros2_humble.sh
access control disabled, clients can connect from any host
WARNING: Published ports are discarded when using host network mode
root@raspberrypi:/#
```

Then enter in the docker terminal,

```
ros2 launch slamware_ros_sdk slamware_ros_sdk_server_and_view.xml
ip_address:=192.168.11.1
```

If successful, you can view the map building screen.