SLAMKit Licensed Softeware

Reliable Mapping and Localization Solution

User Manual

Model: SLAMKit

Enabling robots to better understand their world

Apr. 2024 Version 0.1.0

SLAMTEC Inc.

www.slamtec.com

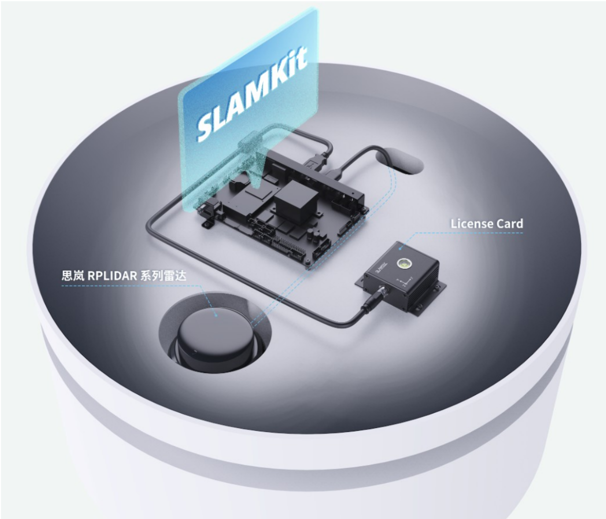
Contents

List of Tables

List of Figures

List of Listings

SLAMKit



1 Advantages

- High-performance IMU module with native USB interface
- Robust mapping for large-scale environment
- High-resolution map building for special scenarios
- Innovative SharpEdge™ map optimization technology
- Robust self-localization in complex and dynamic environments
- Comprehensive toolchain
- Full lifecycle support service

2 Introduction

SLAMKit is a software licensing product developed by SLAMTEC Inc. It is the first commercial solution designed for the robotics industry that can independently provide mapping and localization functionality. And the robotic applications can be deployed rapidly due to its industry-leading scalability and stability. SLAMKit can embedded run in a robot controller through software licensing, enabling the robot to map and self-localize in its operational scenario. Customers can build their high level applications easily through standardized software interfaces. As shown in the diagram below, SLAMKit consists of three components: RPLidar, license card, and licensed slamware.

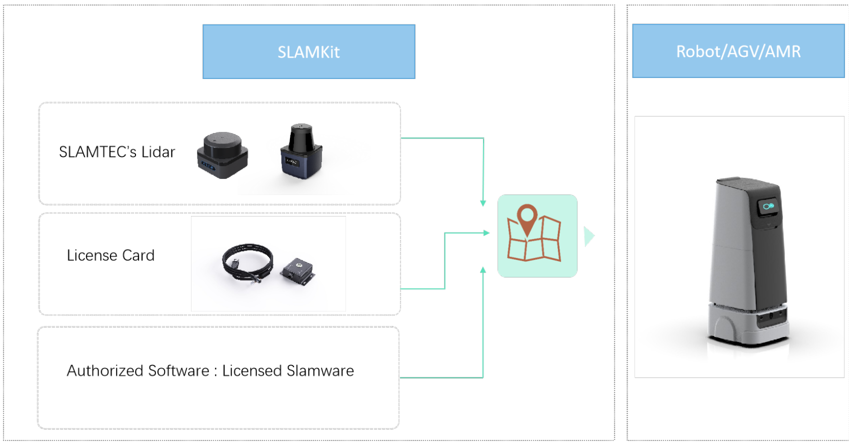


Figure 2.1: SLAMKit Composition Diagram

The licensed software known as Slamware is the core component of SLAMKit, whose system diagram is shown below. The system inputs include lidar frame, data sensed by license card and odometry. Among the above inputs, users only need to provide odometry through ROS topic. And the system outputs can be defined as a toolchain for secondary development, including the Robostudio, C++ SDK, Java SDK, Restful API SDK, ROS SDK, etc.

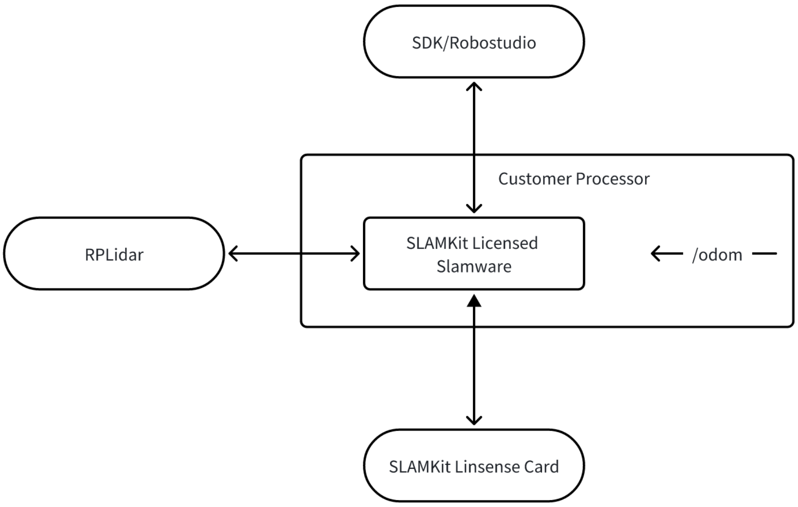


Figure 2.2: SLAMKit System Diagram

The SLAMKit license card can be used independently as an IMU module, it provides 9-axis sensor data continuously and stably. Otherwise, it serves as the authorization key of SLAMKit. The licensed slamware can be divided into two versions based on the authorization key: the developer edition and the pro edition. Here are the distinctions between them:

	Developer Edition	Pro Edition
Mapping area	150m * 150m	500m * 500m (Depending on the computational resources, can map larger area.)
Operating duration	1.5 hour repeatly	unlimited
Update service	no	iteration

Table 2.1: Functional Differences Between Versions

3 SLAMKit Deployment Tutorial

This section will introduce the deployment steps of SLAMKit to help users runing licensed software on their own system.

3.1 Computational Resources Evaluation

SLAMKit is an software application running on the client's processor. So users must confirm that the computing resources meet the requirements. The key parameters are shown in the following table:

Description	Illustrate
Supported Processors and Architectures	Intel x86, x64 series ARMV7/V8 series
Supported Operating Systems	Ubuntu 18.04, 20.04
Hardware Interface	1 * USB2.0 interface for licensing card. 1 * Ethernet port, 1 * Serial port, or 1 * USB port to connect Lidar + 1 * Power interface.(Please refer to the used LiDAR datasheet) Note: 100 Mbps Ethernet port is enough, and the USB Lidar is not recommended due to the signal interference.
CPU Usage	RK3399: max to 2 * A72 + 2 * A53 Intel Celeron J3455: max to 2.5 cores Note: Actual evaluation results may vary
Memory Usage	Mapping mode: 2GB on average for 5cm resolution map building. Localization mode: max to 2.5GB (map resolution is 5cm and map size is 250,000m^2) Note: Memory usage may increase with larger map areas and higher resolutions
Disk Usage	15GB (5GB for log, 10GB for core dump)
Software Interface Requirements	Subscribe to ROS topic /odom which is provided by user. Note: The publishing frequency of /odom must be greater than 50Hz, preferably 100Hz.

Table 3.1: Computational Resources Evaluation List

Specifically, SLAMKit pro edition users encountering performance bottleneck issues during usage or requiring comprehensive evaluation of product performance can contact SLAMTEC technical support team. The technical support team will assist users in evaluating whether the resource usage and the thread frequency meet the requirements for design specifications.

3.2 Lidar and License Card Installation

Users need to connect the RPLidar and the license card to the control unit. As shown in figure 3.1, the main control unit is the industrial personal computer (IPC) and IPC will be used to refer to the main control unit in the remainder of the document. Furthermore, the hardware interfaces follow the description in table 3.1.

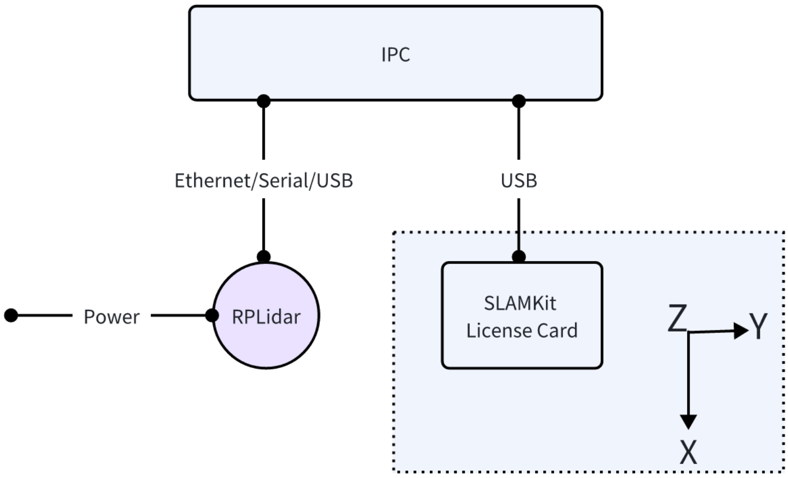


Figure 3.1: Concept Map of Hardware Connections

Figure 3.2: Physical Connection Example

When assembling the robot using the connected SLAMKit components and IPC, it is essential to inspect the following installation details.

- The installation of license card follows the right-hand system rule, meaning the positive direction of the X-axis marked on the license card corresponds to the front direction of the robot, and the positive direction of the Y-axis marked on the license card aligns with the left side of the robot. Additionally, it is recommended to install the license card at the center or along the centerline of the robot, specifically on two perpendicular lines passing through the center of the wheelbase or the midpoint of the wheelbase, as illustrated in figure 3.3 below.

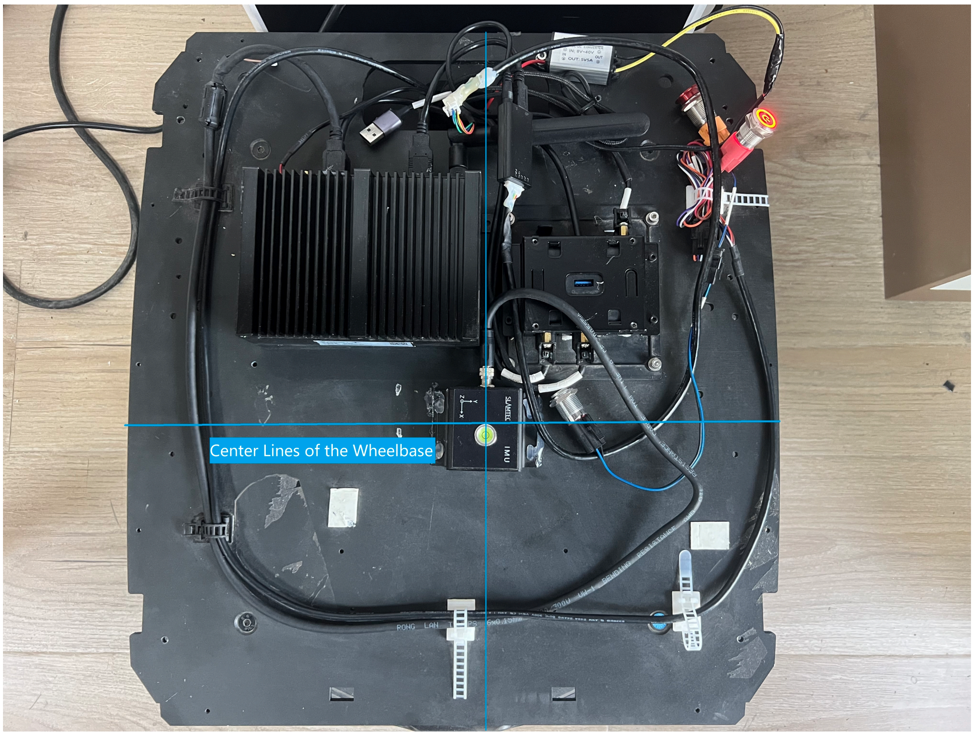


Figure 3.3: Installation of License Card

- The license card needs to be installed horizontally, ensuring that the small bubble of gradienter on the license card is within the black circle indicator as shown in figure 3.4 .

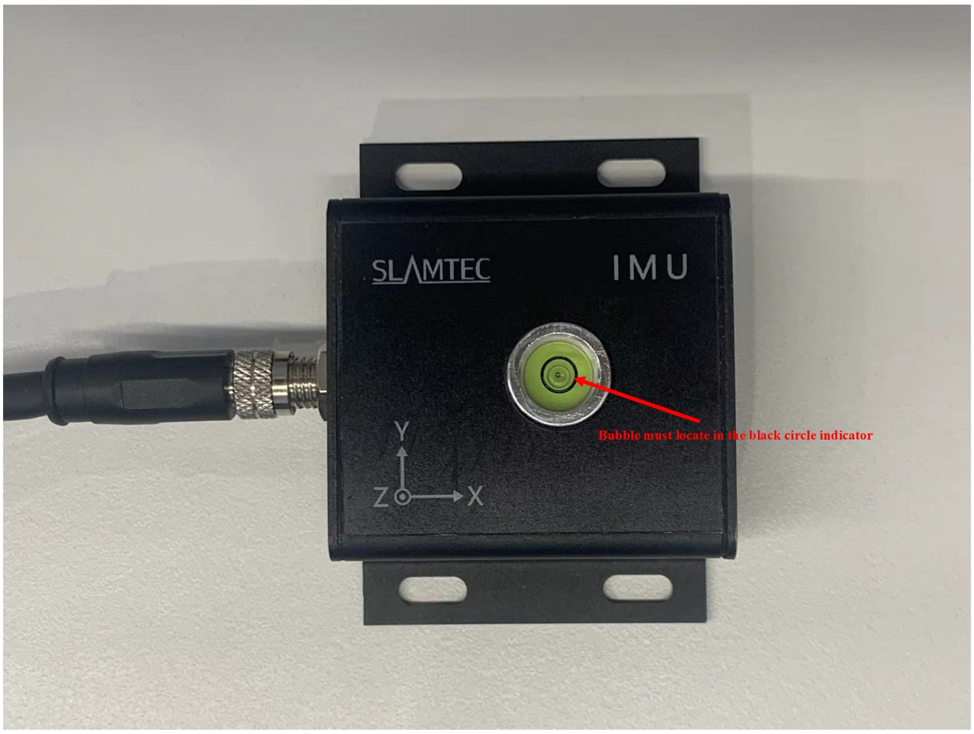


Figure 3.4: Horizontal Indication of License Card

- The license card should be installed away from the vibrating objects, which the objects broadly refer to components or structural parts that continue to vibrate even when the robot is stationary.

3.3 SLAMKit Software Installation

3.3.1 Preparation

Preparation of OS and ROS

- Before installing SLAMKit, users can use table 3.2 to confirm that the using operating system and ROS version are compatible with the versions supported by SLAMKit. (ROS installation tutorials)

OS version	ROS version
Ubuntu 18.04 64bit	Melodic
Ubuntu 20.04 64bit	Noetic

Table 3.2: Corresponding OS and ROS

- The following checks whether the odom topic meets the requirements of SLAMKit. Use the ROS commands to view the reported odom topic and its publishing frequency. It is necessary to ensure that there exists a ROS topic named /odom in the system, with data format nav_msgs/Odometry. SLAMKit needs to subscribe to this topic to get odometry data. Additionally, the publication frequency of /odom should be more than 50Hz, preferably 100Hz. It is important to note that the localization information output by SLAMKit is aligned with the coordinate system of the odom data provided by users. For example, in a differential-drive motion model, if the center of the odom is also the center of the wheelbase, then the localization result output by SLAMKit will be the pose of this wheelbase center in the map. Therefore, users should pay attention to this correspondence when providing odom data and getting localization pose.

```
rostopic list
rostopic hz /odom
```

Listing 1: Check ROS Topic Commands

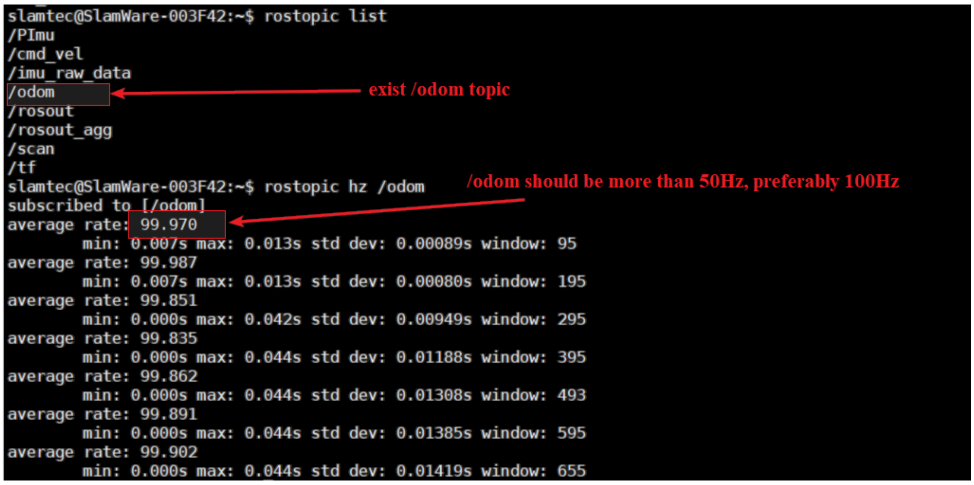


Figure 3.5: Details of /odom Topic

Download SLAMKit Installation Package and Robostudio

- Please download the SLAMKit ”deb” installation package from the official website of SLAMTEC. Both the Intel and ARM on-chip architectures are supported currently. Accordingly, users should download the package ”xxx_x86_64.deb” when Intel chip based development platform is used. And ”xxx_aarch64.deb” should be download when using ARM chip based development platform. The ”xxx” represents the version number of each ”deb” installation package.

- Similar to the description in last step, users should additionally download the appropriate ROS middleware ”slamware_ros_bridge” according to the used development platform and compiler version. Generally, for Ubuntu systems, the default compiler version corresponds to the following:

Ubuntu Version	GCC Version
Ubuntu 18.04 64bit	gcc7
Ubuntu 20.04 64bit	gcc9

Table 3.3: Default GCC Version for Ubuntu

- Please download RoboStudio from the official website of SLAMTEC. RoboStudio is a UI developed by SLAMTEC that is compatible with both Windows and Android. Users can use RoboStudio to connect with SLAMKit, enabling some operations such as mapping and self-localization monitoring, editing maps, and uploading configuration files.

3.3.2 Software Installation

1. Make a network connection between user's personal PC and IPC. Then, login to IPC using an SSH tool (MobaXterm is recommended). Or use the external peripherals such as a screen, keyboard, and mouse directly to login to the OS of IPC.

2. Copy the "deb" installation package and "slamware_ros_bridge" compressed file to the corresponding folder and change directory(use "cd" command) to that folder. Here, we'll use "/home/slamtec" as an example. If you encounter permission issues during copying, run the following command:

```
sudo chmod -R 777 /home/slamtec
```

Listing 2: Change Permissions Command

3. Replace "slamware_5.1.0_all.deb" in the following command with the actual name of the ".deb" file. And Run the following command to install the SLAMKit "deb" package.

```
sudo dpkg --force-all --refuse-confdef -i /home/slamtec/slamware_5.1.0_all.deb
```

Listing 3: Installation of SLAMKit Functional Software

4. Unzip the "slamware_ros_bridge" middleware to the floder. Then, use the following commands to compile and install the middleware.

```
tar zxvf /home/slamtec/slamware_ros_bridge-x86_64-gcc9.tar.gz
cd /home/slamtec/slamware_ros_bridge/
catkin_make
source devel/setup.bash
```

Listing 4: ROS Bridge Installation

3.3.3 Lidar Configuration

Users should download the default "cube config" file from the official website of SLAMTEC. The "cube config" can be divided into two kinds according to the interface type of the used RPLidar. One is released for the RPLidar with serial and USB interface named ”Slamkit_serial.cube_cfg_dat”. And the other is released for the ethernet RPLidar named ”Slamkit_udp.cube_cfg_dat”. Both cube config files are written in JSON format. Thus, users can use any kind of JSON editor to set private parameters(such as notepad). And the following steps will help users to update your own lidar configs onto SLAMKit.

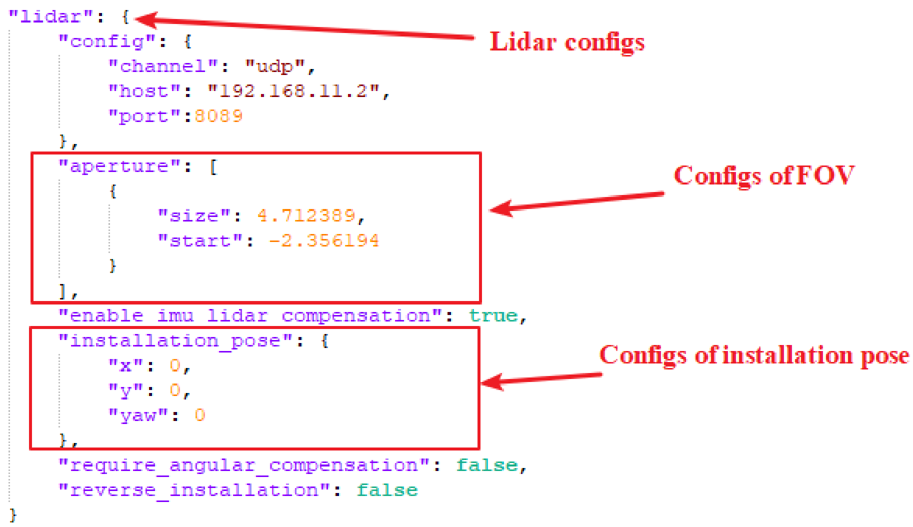


Figure 3.6: SLAMKit Cube Config

Baud Rate Configs of Serial Lidar When using a serial Lidar, users need to configure the connection baud rate according to the Lidar model and modify the ”baudrate” configuration item to the corresponding baud rate value. For example, when using the RPLidar C1, you need to set the ”baudrate” to 460800. The correspondence between SLAMTEC Lidar models and baud rates is shown in the table below.

Lidar	Baud rate(bps)
RPLidar A1	115200
RPLidar A2M7, A2M12, A3M1, S1	256000
RPLidar S2, S3	1000000
RPLidar C1	460800

Table 3.4: Correspondence Between SLAMTEC Lidar Models and Baud Rates

Installation Configs of Lidar The configuration items of Lidar installation pose indicates the coordinates and yaw(heading angle) of the Lidar in the coordinate system with the odometry center as the origin. As shown in the figure below, the odometry coordinate system follows the right-hand rule, where the positive x-axis direction is towards the front of the odometry center, and the positive y-axis direction is to the left of the odometry center (consistent with the coordinate indication on the license card); Yaw is the angle between the $0°$ mark direction of the Lidar and the x-axis, with a range of $[-\pi, \pi]$. The units for the x and y parameters in the lidar installation pose configuration are meters, and for yaw, it's in radians.

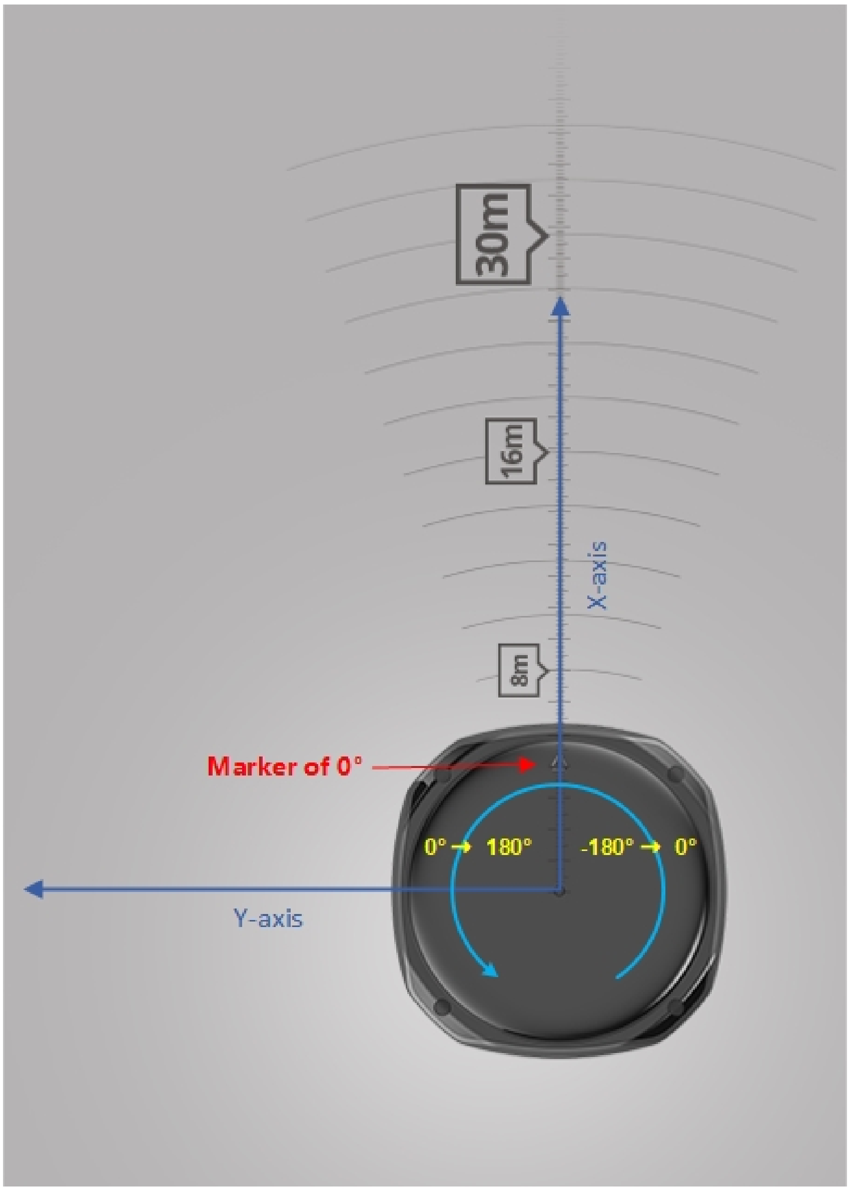


Figure 3.7: Illustration of Lidar Installation Pose

Example: If the lidar is installed 30cm in front of the odometry center and facing forward, the "installation_pose" should be configured as follows.

```
            "installation_pose":
            {
                "x": 0.3,
                "y": 0,
                "yaw": 0
            }
```

Listing 5: Configuration Example of Lidar Installation Pose

View Range Configs of Lidar In figure 3.8, the field of view configuration is primarily used to set the effective scanning range of the Lidar. The $0°$ angle corresponds to the position marked on the lidar, and the angle range of one lidar frame is described using $[-\pi, \pi]$. The field of view can be configured as multiple non-overlapping sectors, each described by a starting angle "start" and an angular range "size" ("start" and "size" are both in radians).

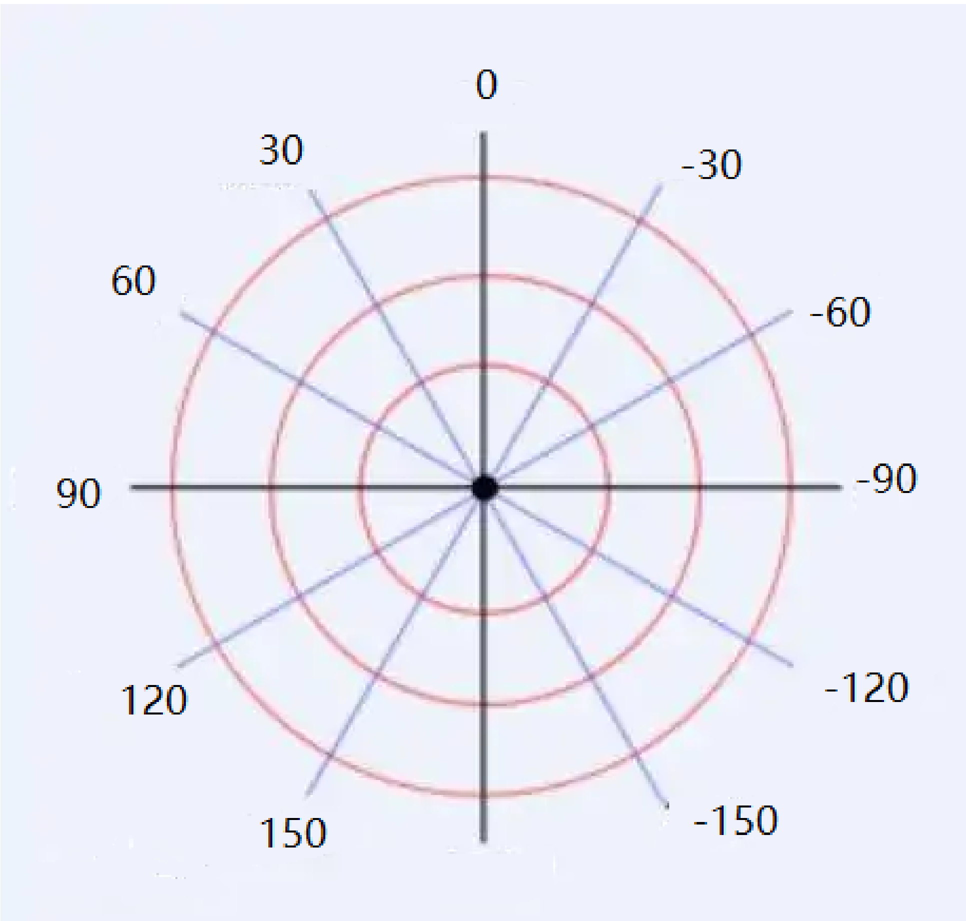


Figure 3.8: Angle Range of Lidar

Example 1: For configuring an angle range of $-135° \sim 135°$, "start" should be configured as -2.356 ($-135°$), and "size" should be configured as 4.712 ($270°$).

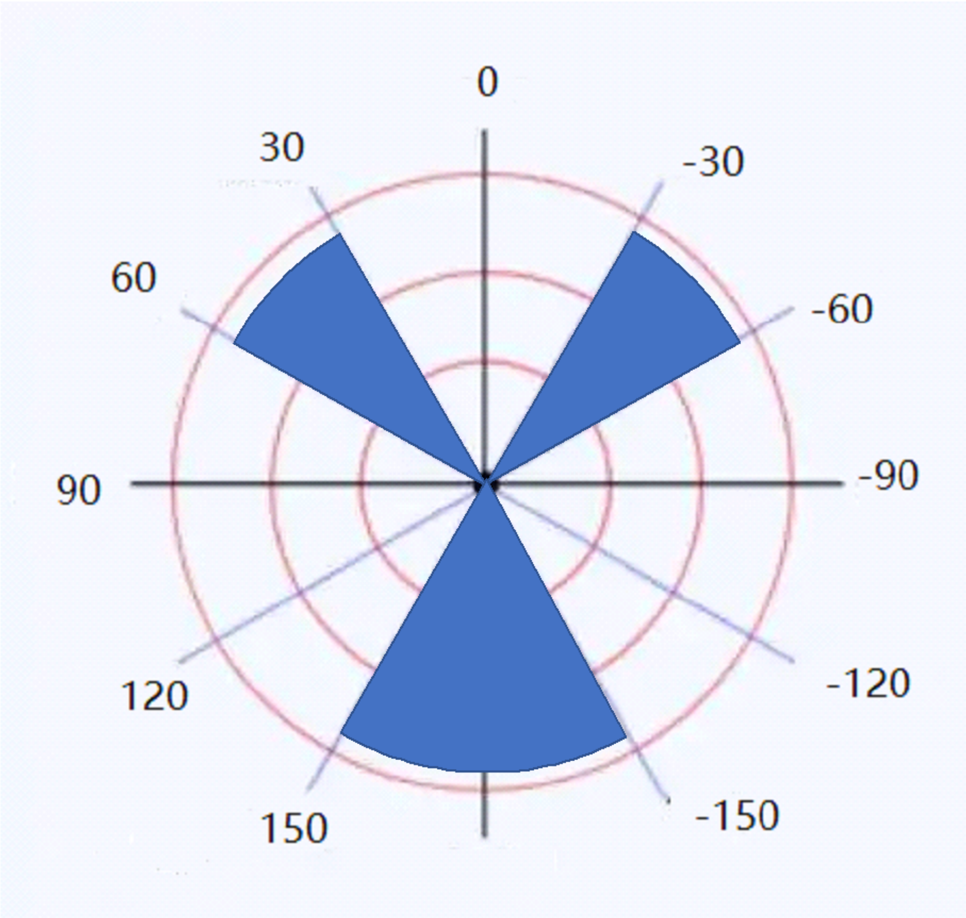


Figure 3.9: Example of Multiple View Ranges

Exmaple 2: Another example of multiple view ranges is illustrated. To filter the blue sector shown in figure 3.9, the corresponding configuration segments should be writtern as follows.

```
          "aperture": [
              {
                  "size": 1.570796,
                  "start": -2.617994
              },
              {
                  "size": 1.047198,
                  "start": -0.523599
              },
              {
                  "size": 1.570796,
                  "start": 1.047198
              }]
```

Listing 6: Configuration Example of Multiple Lidar View Ranges

Upload Configuration File Before uploading the cube config to IPC, make sure the ”update server” service is active. If the ”update server” service is not active, you need to start it manually. The following commmands are used to check the service status and start the service.

```
systemctl status update_server.service #check the service status
systemctl start update_server.service  #start the service
```

Listing 7: Service Check and Start Commands

Launch Robostudio and click ”SlamCube Config Tool”→”Upload Configuration”→”Config file”→”Upload Config” Shown in the next figure. Then, Wait until the cube config is successfully uploaded.

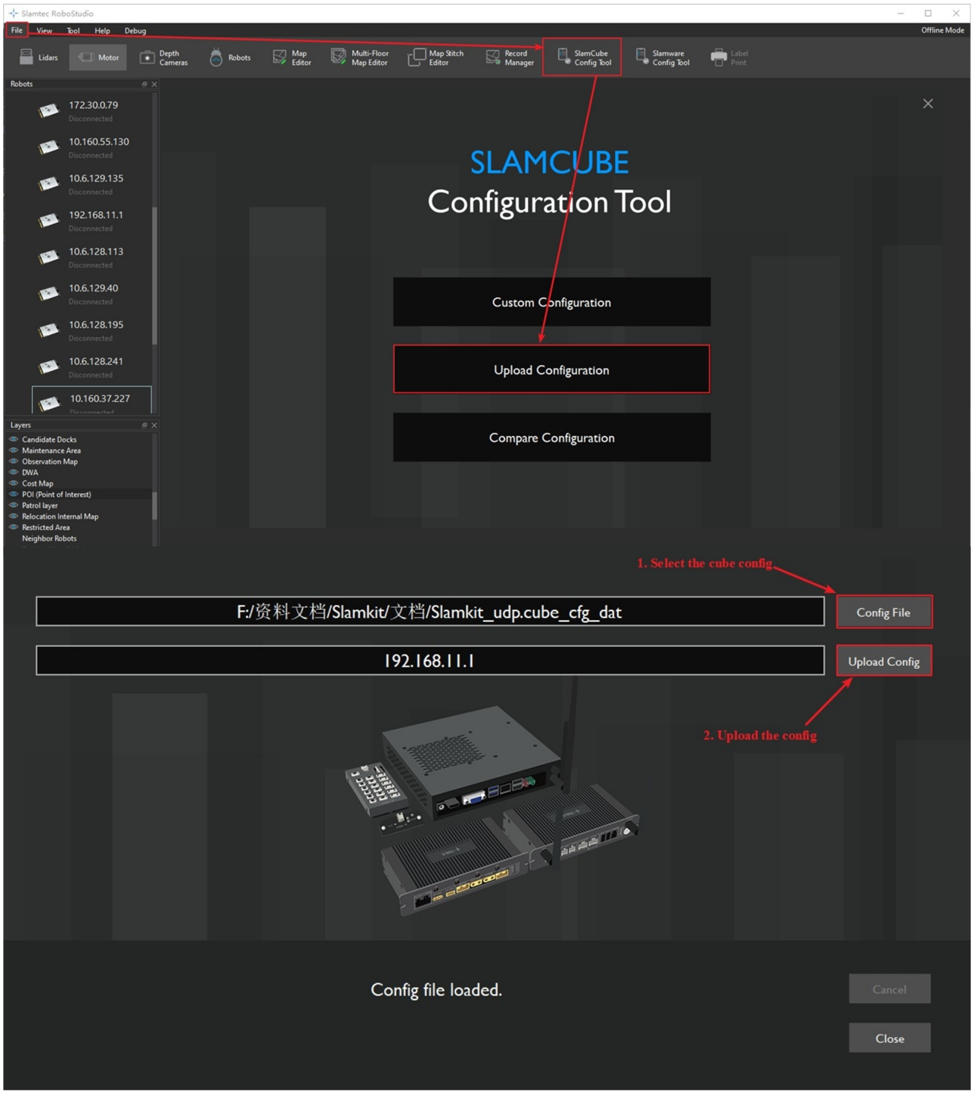


Figure 3.10: Using Robostudio to Upload Cube Config

3.3.4 Start Services

1. Run the following commands to launch "ROS bridge" middleware.

```
roslaunch slamware_ros_bridge slamware_ros_bridge.launch
```

Listing 8: Launch ROS Bridge

2. Run the following commands to restart the SLAMKit core services.

```
systemctl restart slamwared
systemctl restart agent
```

Listing 9: Restart Services

Now, the SLAMKit has already been installed successfully. Note that the licensed software services will start automatically on boot after installation. However, users need to manage and start the odometry middleware ”slamware_ros_bridge” yourself. And the odometry middleware must be started before the licensed software services, or use the command in Listing 9 to restart the corresponding services after starting the middleware.

4 Using SLAMKit

4.1 Connect SLAMKit Using Robostudio

1. Install Windows or Android RoboStudio on user's device.
2. RoboStudio can work in both online and offline mode. The difference between these two modes can be found in the user manual of RoboStudio.
3. When the WiFi module of IPC is set to AP mode, users can connect to its hotspot. Alternatively, when the WiFi module of IPC works in station mode, it can be connected to the same local network as the user's terminal device. Without WiFi, user can take a hardware connection with IPC via ethernet.
4. As shown in figure 4.1, users can follow the operation steps to connect to SLAMKit. (Attention: in the second step, "Manual Connect Robot" is appeared by right-clicking.) And the IP address is 192.168.11.1 in WiFi-AP and ethernet connection. Instead, the local network IP address of IPC can be used in WiFi-station mode.

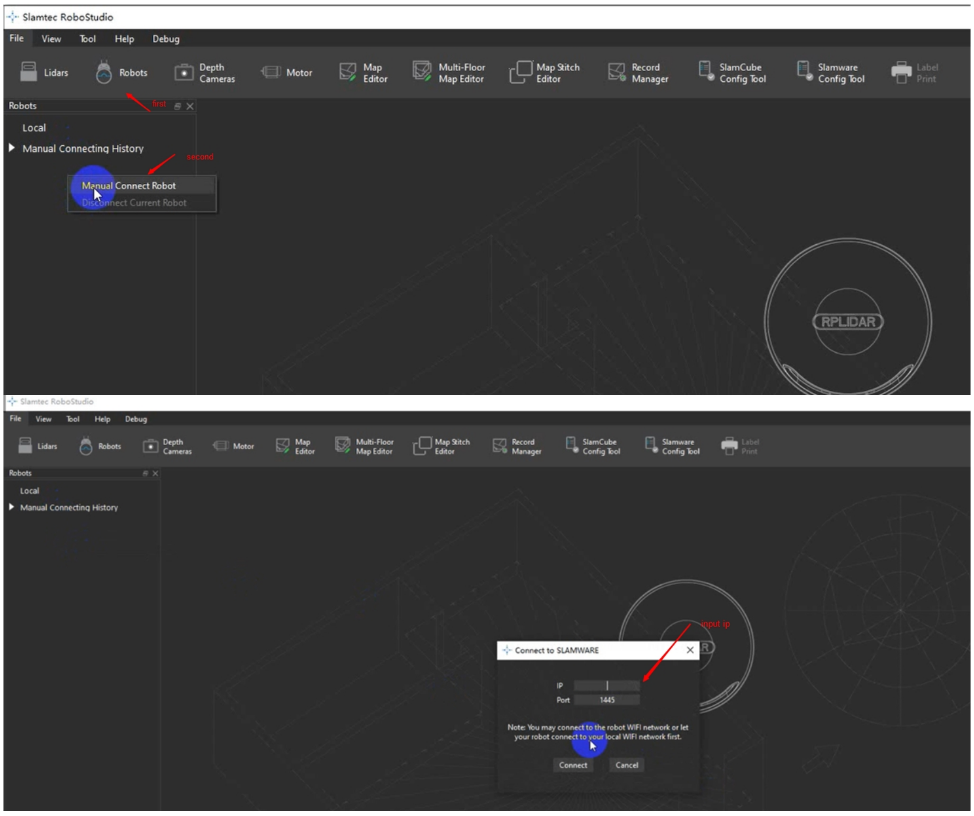


Figure 4.1: Connect to SLAMKit Using Robostudio

5. As soon as the SLAMKit is successfully connected, the main work area of Robostudio will display the pose, layered map, point cloud, and other status.

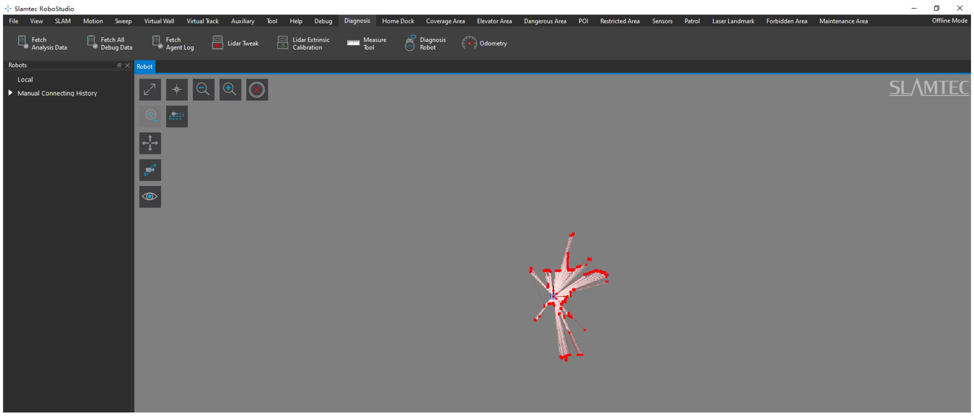


Figure 4.2: UI of Robostudio

6. For more operations, please refer to the user manual of Robostudio.

4.2 SDK for Custom Development

Different SDKs are provided by SLAMTEC to meet the development requirements on various platforms. Users can choose the one that best suits their preference.

- C++ SDK
- JAVA SDK
- ROS SDK
- RestfulAPI SDK

Here is an explanation of how to use the ROS SDK as an example.

1. After compiling the ROS SDK, modify the roslaunch file using the following command. Change the ”ip_address” parameter to the IP address of the IPC running SLAMKit. If the ROS SDK and SLAMKit are installed on the same IPC, change it to 127.0.0.1.

```
sudo vi src/slamware_ros_sdk/launch/slamware_ros_sdk_server_node.launch
```

Listing 10: Modify the IP Address

2. Use the following commands to launch ROS SDK.

```
roslaunch slamware_ros_sdk slamware_ros_sdk_server_node.launch
```

Listing 11: Launch ROS SDK

3. Then, users can confirm that the ROS SDK has been started successfully as follows.

```
rostopic echo /slamware_ros_sdk_server_node/scan
rostopic echo /slamware_ros_sdk_server_node/odom
```

Listing 12: Echo the Topics of ROS SDK

5 Get Start with IMU Module

The license card provided by SLAMKit can be used independently as an IMU module. Before using the IMU sensor data, users must confirm that ROS has been installed on your IPC. Then, users must finish the hardware installation described in section 3.2. Finally, the IMU module can be drived according to the following steps.

5.1 Dependencies Installation

The IMU driver depends on libusb-1.0-0-dev and ROS imu-tools. Run the following commands to install dependencies. Here, <dist> represents the ROS version name, such as ”noetic”, ”melodic”, etc., which should match the ROS version installed by the user.

```
sudo apt-get update
sudo apt-get install libusb-1.0-0-dev
sudo apt-get install ros-<dist>-imu-tools
```

Listing 13: Dependencies Installation Commands

5.2 Software Installation

1. Take a SSH connection with IPC as section 3.3.2, and run the following commands to create a workspace.

```
mkdir slamkit_ws
cd slamkit_ws
mkdir src
```

Listing 14: Create Workspace Commands

2. Download ”slamkit.tar.gz” file and copy it to the folder ”slamkit_ws/src” created in last step. Then, run the following commands to extract and install the driver package.

```
cd slamkit_ws/src
tar zxvf slamkit.tar.gz
cd ..
catkin_make
source devel/setup.bash
```

Listing 15: Driver Installation Commands

3. When the license card is successfully connected to the user's IPC. Use "lsusb" command to check if the corresponding device has been recognized by OS.

```
lsusb
#Device ID of SLAMKit license card
Bus 003 Device 011: ID fccf:f100 SLAMTEC SLAMWARELC
```

Listing 16: Device ID of License Card

4. Run the script to add the udev rule as follows.

```
cd slamkit_ws/src/slamkit/scripts
./add_udev.sh
```

Listing 17: Add udev Rule

5.3 Launch IMU Driver Node

SLAMKit provides two startup modes, allowing users to choose the appropriate one based on their applications.

Common Driver Mode In this mode, only the IMU data collection node is launched. Users can start and view data using the following commands.

- "imu/data_raw"(sensor_msgs/Imu): Messages containing raw IMU data, including angular velocities and linear accelerations. Linear accelerations are measured in m/s^2, and angular velocities are measured in rad/s.
- "imu/mag"(sensor_msgs/MagneticField): Messages containing raw magnetic data in $tesla$.

- ”imu/processed_yaw”(geometry_msgs::Vector3Stamped): Yaw data which processed in license card which measured in degree.

```
roslaunch slamkit_ros slamkit_usb.launch
rostopic list
#topic list as follows
/imu/data_raw
/imu/mag
/imu/processed_yaw
```

Listing 18: ROS Topics in Common Driver Mode

Filter Driver Mode In this mode, both the IMU data collection node and filter node are launched. And the data descriptions and operational commands are shown below.

- ”imu/data”(sensor_msgs/Imu): Messages containing the raw 3-dimensional acceleration and 3-dimensional angular velocity data, as well as quaternion data processed internally.
- ”imu/rpy/filtered”(geometry_msgs/Vector3): Messages containing the filtered 3-dimensional Euler angles (roll, pitch, and yaw), measured in radians. This item is only published when the ”publish_debug_topics” is set to true in the ROS launch file.
- ”imu/steady_state”(std_msgs/Bool): It publishes a boolean value obtained from processing IMU sensor data, indicating whether the IMU module is in a steady state (true for steady).

```
roslaunch slamkit_ros slamkit_usb_imu_filter.launch
rostopic list
#topic list as follows
/imu/data
/imu/data_raw
/imu/mag
/imu/processed_yaw
/imu/rpy/filtered
/imu/steady_state
```

Listing 19: ROS Topics in Filter Driver Mode

5.4 Test and View

SLAMKit also provides a way to test and view IMU data in RVIZ. Run the following "roslaunch" command to start it.

```
roslaunch slamkit_ros test_slamkit.launch
```

Listing 20: Run Test and View Mode

Then, RVIZ will visualize the data from the "imu/data" as shown in the next figure.

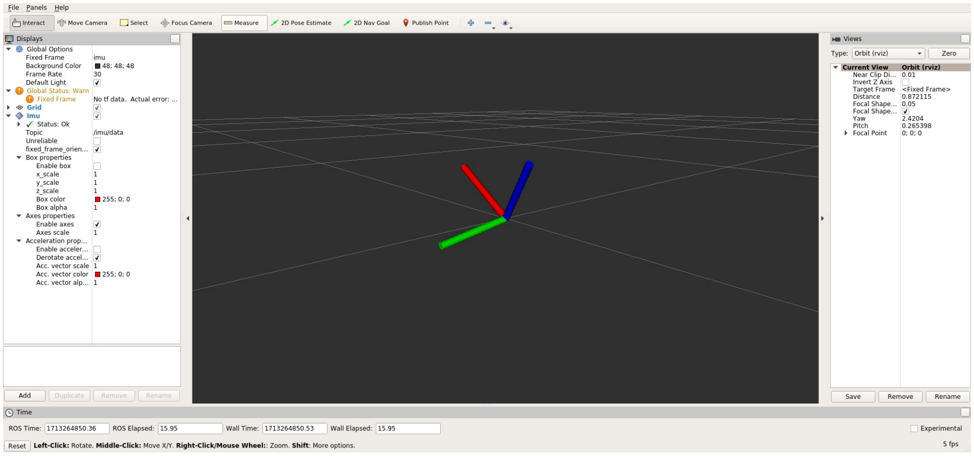


Figure 5.1: Data View in RVIZ

The data details of each topic can be viewed in the terminal as follows.

```
rostopic echo /imu/rpy/filtered
#/imu/rpy/filtered data details
header:
  seq: 116510
  stamp:
    secs: 1712653629
    nsecs: 982544281
  frame_id: "imu"
vector:
  x: -0.5118724746572548
  y: -0.851032829630307
  z: 0.24633383804987355
```

Listing 21: Data Details of Each Topic

```
rostopic echo /imu/data
#/imu/data data details
header:
  seq: 67355
  stamp:
    secs: 1712893929
    nsecs: 799465854
  frame_id: "imu"
orientation:
  x: 0.3294100186970593
  y: -0.01334335189830093
  z: 0.1402511003896779
  w: -0.9336169575268038
orientation_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
angular_velocity:
  x: -0.004261057671440972
  y: 0.02769687486436632
  z: -0.008522115342881944
angular_velocity_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
linear_acceleration:
  x: 0.74708251953125
  y: -5.19010009765625
  z: 6.66392822265625
linear_acceleration_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]

rostopic echo /imu/angles_degree
#/imu/angles_degree data details
header:
  seq: 134377
  stamp:
    secs: 1712653708
    nsecs: 150230549
  frame_id: "angle_degree"
vector:
  x: -29.359611349189777
  y: -48.705977758347366
  z: 14.224317662550938
```

Listing 22: Data Details of Each Topic (continued)

```
rostopic echo /imu/processed_yaw
#/imu/processed_yaw data details
header:
  seq: 8217
  stamp:
    secs: 1712893652
    nsecs: 718455979
  frame_id: "imu_processed"
vector:
  x: 0.0
  y: 0.0
  z: 1.6558480277118701
```

Listing 23: Data Details of Each Topic (continued)

6 Upgrade of SLAMKit Pro Edition

The pro edition of SLAMKit removes restrictions on mapping performance and usage time. Users can also enjoy the full lifecycle support services for robot products provided by SLAMTEC. The specific upgrade process is as follows.

1. If the user has previously installed the developer edition of SLAMKit on the used IPC, this step can be skipped. However, if this is the first deployment of the pro edition, users need to follow the installation steps outlined in section 3 to install ".deb" package of developer edition first, and then proceed to the next step.

2. Use the IP address of IPC to access the update service page in browser. (WiFi AP mode and direct connection via Ethernet use IP 192.168.11.1, WiFi station mode uses the local network IP of IPC)

3. After finish the act of procurement, users need to get the device SN(serial number) from the update service page as shown in figure 6.1. Then, send it to the sales or support engineer, and they will provide a license file (sign.txt) in return.

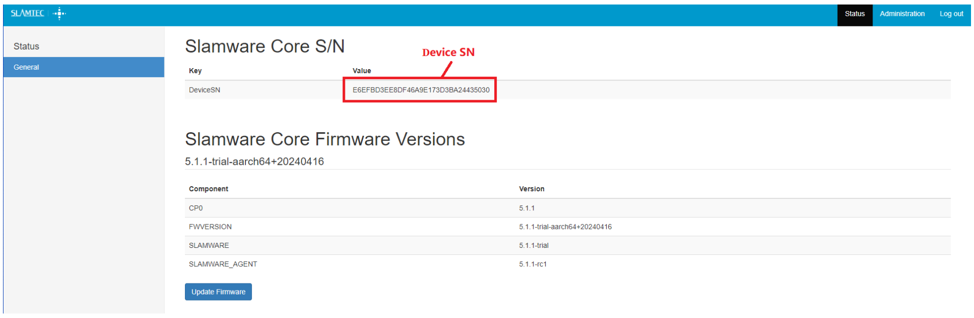


Figure 6.1: Get Device SN

4. Follow the instructions in figure 6.2 to login to the management page.

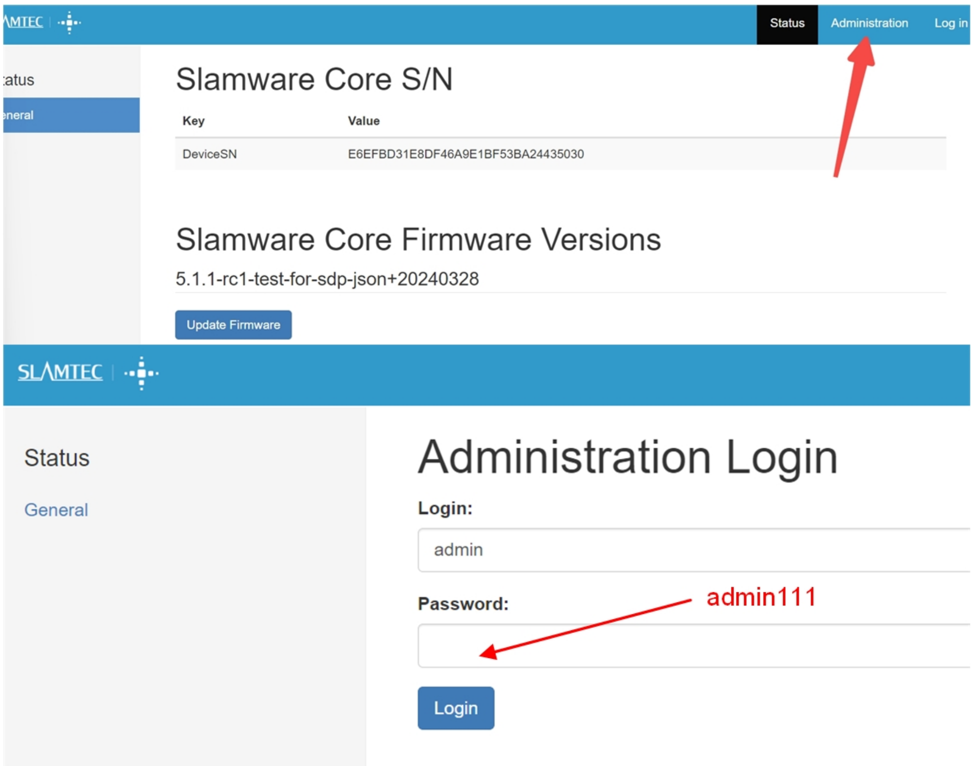


Figure 6.2: Login to the Management Page

5. After confirming that the SLAMKit license card is properly connected to IPC, follow the instructions below to complete the license update.

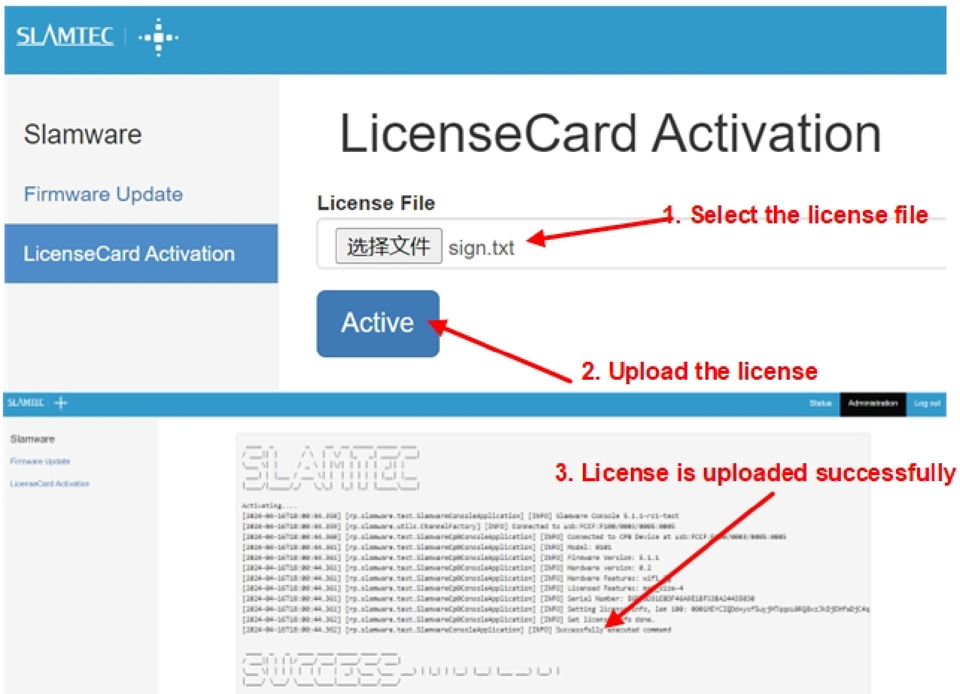


Figure 6.3: License Update Instructions

6. Users can obtain the pro edition firmware (firmware ending with .bin) from the technical support engineer of SLAMTEC. Then, follow the steps below to complete the firmware update.

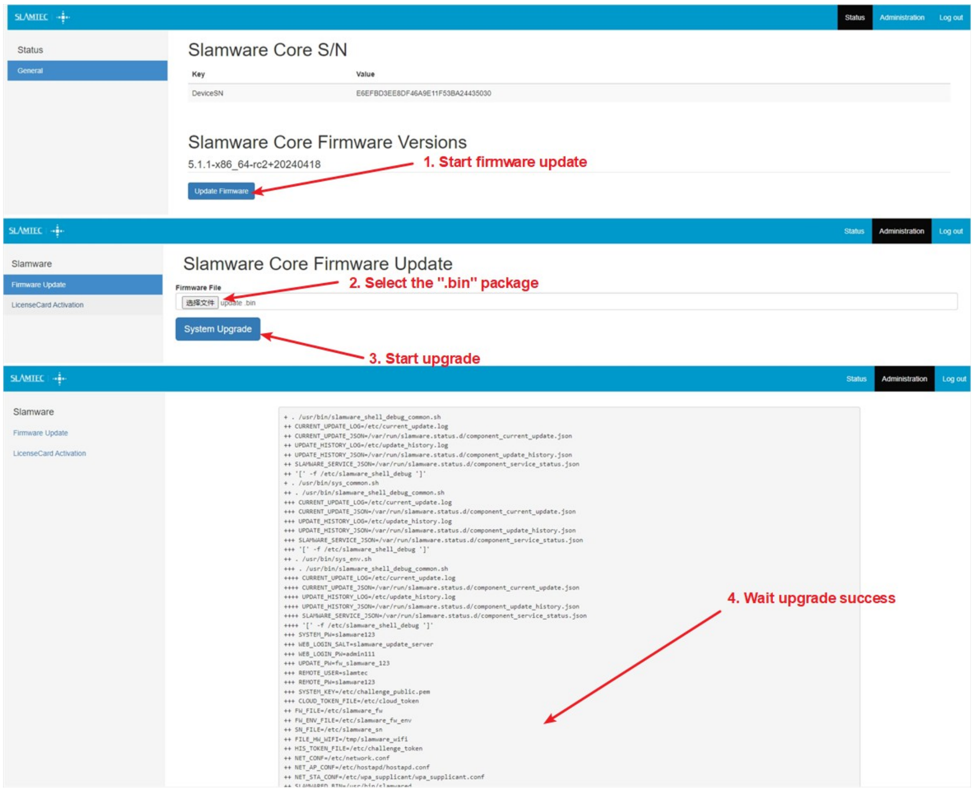


Figure 6.4: Pro Edition Firmware Update

7. After completing the pro edition firmware update, use the following commands to enable the services of SLAMKit.

```
systemctl restart slamwared
systemctl restart agent
```

Listing 24: Enable SLAMKit Commands

Now, the pro edition upgrade is complete. If users want to update the firmware for subsequent iterative versions, the above steps from 6 to 7 can be repeated to complete the firmware update.

7 FAQ

1. Error occurred during installation of the deb package.

Users can ignore the occurred error at the begining. Instead, use the following command to get the version of SLAMKit. If the version number matches the one you intended to install, it indicates a successful installation. If not, you may need to reinstall.

```
cat /etc/slamware_release
```

Listing 25: Get Version of SLAMKit

2. Failure to start the Ethernet RPLidar.

The default IP address of the Ethernet RPLidar is 192.168.11.2. Therefore, confirm that IPC is configured in the same network segment as the Lidar. You can use the following command to change the IP address of IPC.

```
#eno1 can be replaced with the actual port ID
ip addr add 192.168.11.1/24 dev eno1
```

Listing 26: Change the IP Address of IPC

Once, the RPLidar can be successfully ping from IPC, it means IPC can receive data from lidar.

```
ping 192.168.11.2
```

Listing 27: Check Ethernet Lidar Connection Command

3. How to check the connection between IPC and license card?

The connection status can be confirmed by using the ”lsusb” command to see if the fccf:f100 device is listed.

4. Why the output localization pose of SLAMKit is inaccuracy?

If there’s a fixed deviation between the localization pose output by SLAMKit and the actual location, causing inaccuracies in navigation. Firstly, users need to check if the odometry center coincides with the center of the robot. Then, users should confirm that the configured installation pose of the lidar matches its actual installation position.

Furthermore, a calibration tool embedded in Robostudio can be used, users can contact SLAMTEC support team to get help.

5. Failure to connect to IPC using SSH.
Only when the SSH service has been started on IPC, a new SSH connection can be created successfully. As a result, users should check the installation status of the SSH service firstly, and install it when the service does not exist. The following commands can be used for checking and installing the SSH service.

```
service ssh status #check SSH service status
sudo apt install openssh-server #install SSH service
```

Listing 28: Check and Install SSH Service

6. Why need to configure the FOV of lidar, and how to accurately set it?
When designing structures of robot, there may be issues where supporting structures obstruct the lidar observation at certain angles. Consequently, it's necessary to filter out the lidar points within the obstructed angular ranges by editing the corresponding configs. In order to accurately set these configs, users can follow the operation steps shown in figure 7.1 to directly read out the angular ranges by using Robostudio.

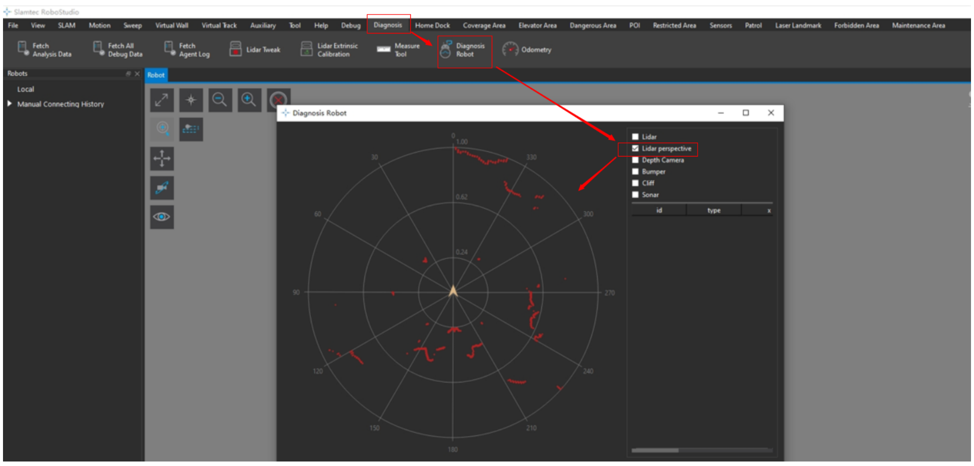


Figure 7.1: FOV of Lidar in Robostudio

7. The /cmd_vel topic received by ROS SDK conflicts with other topics.
Open the ”slamware_ros_sdk_server_node.launch” file, and use a different name instead of ”cmd_vel” in ”<param name=”vel_control_topic” value=”cmd_vel”/>”.

8. When independently use the IMU module, an error is raised as figure 7.2.

```
... logging to /root/.ros/log/7fda4b7a-fc7f-11ee-9f4a-cc4b73822bb0/roslaunch-NanoPC-T4-10918.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://NanoPC-T4:41845/

SUMMARY
========

PARAMETERS
 * /complementary_filter_node/publish_debug_topics: True
 * /rosdistro: melodic
 * /rosversion: 1.14.13
 * /slamkitNode/channel_type: usb
 * /slamkitNode/frame_id: imu
 * /slamkitNode/usb_interfaceId_slamkit: 3
 * /slamkitNode/usb_productId_slamkit: 61696
 * /slamkitNode/usb_rxEndpoint_slamkit: 5
 * /slamkitNode/usb_txEndpoint_slamkit: 5
 * /slamkitNode/usb_venderId_slamkit: 64719

NODES
  /
    complementary_filter_node (imu_complementary_filter/complementary_filter_node)
    slamkitNode (slamkit_ros/slamkitNode)

ROS_MASTER_URI=http://localhost:11311

process[slamkitNode-1]: started with pid [10937]
process[complementary_filter_node-2]: started with pid [10938]
[ INFO] [1713337618.157751673]: Starting ComplementaryFilterROS
[ INFO] [1713337618.199307466]: slamkit running on ROS package slamkit_ros, SDK Version:0.1.0
open my_device fccf:f100 ok, handle: 97c5a9e0
libusb_claim_interface wrong fccf:f100->3 , handle: 97c5a9e0 error msg:  Resource busy
[ERROR] [1713337618.231603783]: Error, cannot connect to slamkit.
[slamkitNode-1] process has died [pid 10937, exit code -11, cmd /home/amy/slamkit_ws/devel/lib/slamkit_ros/slamkitNode __name:=slamkitNode __log:=/root/.ros/log/7fda4b7a-fc7f-11ee-9f4a-cc4b73822bb0/slamkitNode-1.log].
log file: /root/.ros/log/7fda4b7a-fc7f-11ee-9f4a-cc4b73822bb0/slamkitNode-1*.log
```

Figure 7.2: Error Raised in Launching IMU Driver

This is due to the USB port conflict, simply stopping the licensed software service will resolve it.

```
systemctl stop slamwared
systemctl stop agent
```

Listing 29: Stop Licensed Software Commands

9. The RPLidar with serial port can not be normally driven.

Check if there is a soft link to RPLidar under dev. If not, you can run the following command to add it.

```
ln -s /dev/ttyUSB0 /dev/rplidar
```

Listing 30: Add RPLidar Soft Link Command

8 Revision History

Date	Version	Description
2024-05-29	0.1.0	Initial version

Shanghai Slamtec Co., Ltd.
Address□10F, Building E, Shengyin Tower, 666 Shengxia Rd., Shanghai, China
5F, Building K, Poly Green Square, 188 Liaoyang Rd., Shanghai, China
Made in China