SLAMKit

Mapping and Localization Solution

Datasheet

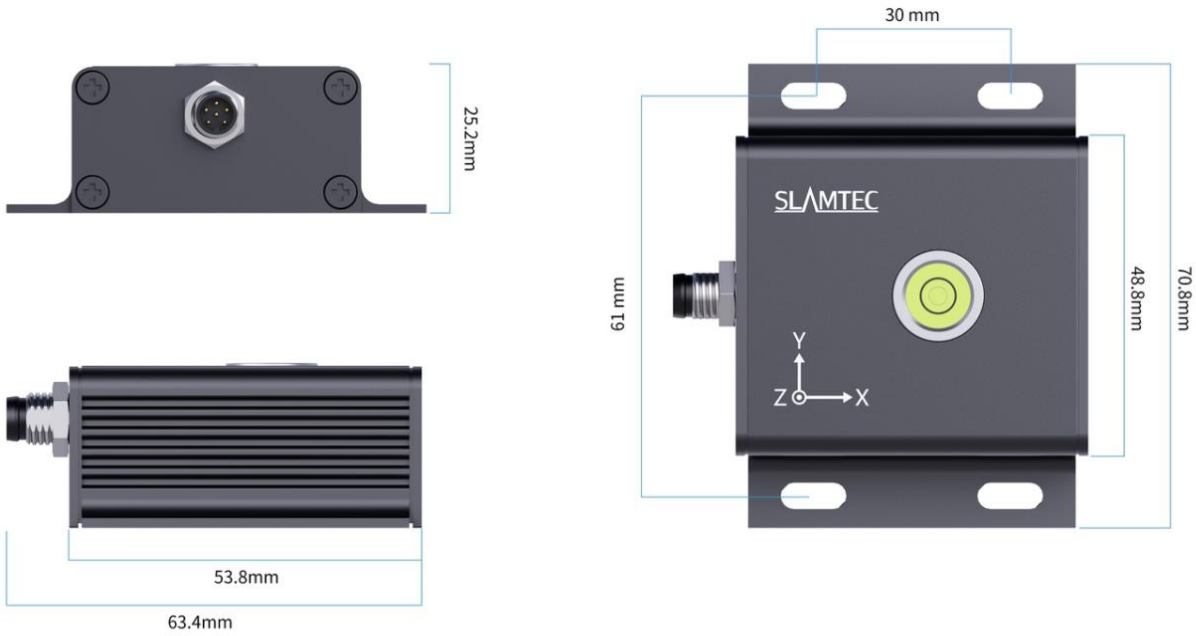
- Efficient
- Reliable
- Stable

Table of contents：

I. Introduction

SLAMKit is a self-developed software licensing product by SLAMTEC. It can run embedded in the main controller and achieve map building and real-time localization in scenarios through Lidar SLAM (Simultaneous Localization and Mapping). The system outputs (for example: map files and localization information) can be used for navigation planning of robots or other mobile devices, as well as planar modeling. Additionally, the product comes with a comprehensive toolchain, including the UI software Robostudio and a series of SDK toolkits for secondary development, enabling users to quickly build customized applications. SLAMKit has the following features:

- Low computing resource, aiding in cost reduction for customized products;
- The license card is a compact hardware device that is easy to integrate into customized products;
- Independently provide mapping and localization functions, supporting function decoupling on customer-side;
- Comes with a complete toolchain, supporting customer-side free expansion and flexible deployment;
- Industry-leading system stability.



Dimensions

II. System Composition of SLAMKit

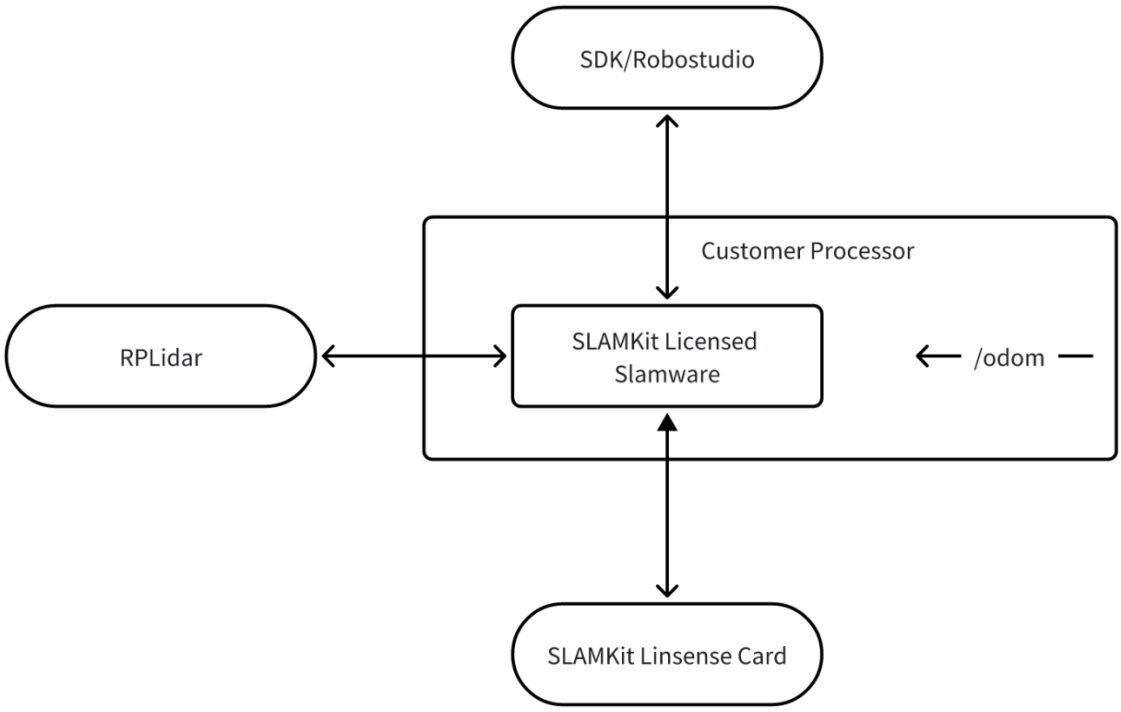
The current SLAMKit offering grants mapping and localization capabilities to customer robots through software licensing. The product architecture comprises three components: SLAMTEC Lidar, SLAMKit license card, and SLAMKit license software, as depicted below.

RPLidar A，S and T series　　SLAMKit License Card　　SLAMKit Licensed Software

Hardware Connection： Connect the aviation plug and aviation socket, then connect the license card to the controller using the USB interface.

III. Operation of SLAMKit

The core of SLAMKit is the license software component, whose system block diagram is shown below. Inputs include Lidar data, the license card, and odometry data. Among them, the lidar data and the license card are independently driven by the license software, while odometry data is sourced from the customer's ROS node. The system's outputs can be defined as the toolchain products like Robostudio, C++ SDK, JAVA SDK, Restful API SDK, ROS SDK etc.



IV. Performance Parameters of SLAMKit

Basic Parameters

Specifications	Details
Communication	Native USB interface and USB protocol
Output	On-chip timestamp, 3-axis accelerometer, 3-axis gyroscope, 3-axis compass, 3D Euler angles, Quaternions, High-precision yaw angle, Module stationary state, etc.
Output Frequency	200Hz
Startup Time	10000ms（including power-on self-test）
Operating Temperature	-40°C~+85°C
Storage Temperature	-40°C~+100°C
Supported Devices	Intel or ARM IPC + Ubuntu 18.04, 20.04 and 22.04
Operating Voltage	5V
Operating Current	12mA

Sleep Current	20uA
Product Dimensions	70.8mm * 63.4mm * 25.2mm
Development Tools	C++ SDK, ROS node

Parameters of IMU Module

Specifications		Conditions	Typical Value
Accelerometer	Range	/	±2g
	Sensitivity	/	16384 LSB/g
	Initial Tolerance	Board-level, all axes	±50mg
	Zero-G Level Change vs. Temperature	0°C~+85°C	±0.80mg/°C
	Noise Spectral Density	Based on Noise Bandwidth = 10Hz	230μg/√Hz
	Cross-Axis Sensitivity	/	±2%
Gyroscope	Range	/	±2000°/s
	Sensitivity	/	16.4 LSB/(°/s)
	Initial Tolerance	At 25°C, placed horizontally	±5°/s
	ZRO Variation Over Temperature	-40°C~+85°C	±0.05（°/s）/°C
	Noise Spectral Density	Based on Noise Bandwidth = 10Hz	0.015（°/s）/√Hz
	Cross-Axis Sensitivity	/	±2%
Compass	Range	/	±4900 μT
	Sensitivity	/	0.15 μT / LSB

	Initial Calibration Tolerance	/	-2000 ~ 2000 LSB
Pitch and Roll Angle	Range	/	X:±180° Y：±90°
Yaw	Range	/	Z:±180°
	Zero bias	Placed horizontally	<0.01°/hr

Parameters of Licensed Software

Specifications	Details
Dimension (RPLidar)	Refer to user manual for the selected model
Maximum Mapping Area	500,000m² ~1,200,000m²
Map Resolution	1cm/2.5cm/5cm selectable
Real-time Localization Error (Typical)	± 5mm,± 1°
Localization Stability:	Remains stable under 50% or more map environmental changes
Maximum Detection Range	50m (typical, varies depending on radar selection)
Scanning Frequency	10 ~ 20Hz
Localization Output Frequency	20 ~ 100Hz

V. Production List

Description	Quantity	Remark
RPLidar/LPX Lidar	1	A Series / S Series / LPX-T1 (radar purchased separately)
License Box	1	70.8mm*63.4mm*25.2mm
License Software	1	Runs on client processor
Controller	1	Optional

VI. Integration References for SLAMKit SDKs

SDK

We support the following versions:：

SDK (Android)：

Tool Download Link：https://www.slamtec.com/cn/support#slamware

This page provides API reference for developing with SLAMWARE SDK on the Android operating system:

Download Link:：

https://wiki.slamtec.com/pages/viewpage.action?pageId=1016579

SDK (Windows)：

1) Tool Download Link:：https://www.slamtec.com/cn/support#slamware

2）This page offers API reference for developing with SLAMWARE SDK on the Windows platform:

Download Link：

https://wiki.slamtec.com/pages/viewpage.action?pageId=1016252

SDK (ios)：

1）Tool Download Link：https://www.slamtec.com/cn/support#slamware

2）This page presents API reference for developing with SLAMWARE SDK on the iOS operating system：

Download Link:：

https://wiki.slamtec.com/pages/viewpage.action?pageId=1016581

ROS SDK：

1）Tool Download Link：https://www.slamtec.com/cn/support#slamware

2) This page provides reference for developing with the SLAMWARE SDK on the ROS platform：

Download Link：

https://wiki.slamtec.com/pages/viewpage.action?pageId=36208683

Linux SDK：

1）Tool Download Link：https://www.slamtec.com/cn/support#slamware

2) Slamtec offers Linux SDKs for different system architectures and compiler versions. These SDKs include sample code for robot motion, exporting map data, and makefiles for compiling executables. The following documentation outlines the process of compiling and executing code within the SDK for users' reference.

Download Link：

https://wiki.slamtec.com/pages/viewpage.action?pageId=22118401

Others：Robot Studio Application Software

Robot Studio is a scalable desktop application for managing and developing robots. It establishes communication with robots, retrieves sensor data, pose, status information, and map data via the robot's interface, processes this information, and presents it to users through an intuitive interface. Users can also send commands to mobile robots using Robot Studio for monitoring and control purposes.

This page provides the data manual for Robot Studio：

Download Link：http://www.slamtec.com/cn/RoboStudio

VII. Quick Start

1. **Preparation**
   a) Connect the license card to the used IPC.
   b) Connect the Lidar to the used IPC.
   c) Ensure the IPC has Docker installed and running correctly.
   d) Obtain the appropriate docker “tar” package based on the IPC's chip architecture and operating system.
2. **Steps**
   a) Import the tar package

   Copy the tar package to any directory on the IPC and run the following command, where “/home/slamtec” is the directory containing the tar file.

```
sudo docker load -i /home/slamtec/xxx.tar
```

b) Rename the image

Run the following command to rename the image, where “1970cd287a7d” is the image ID and “slamkit:x86_64” is the user defined image name. Note that the “:” in “slamkit:x86_64” cannot be omitted.

```
sudo docker tag 1970cd287a7d slamkit:x86_64
```

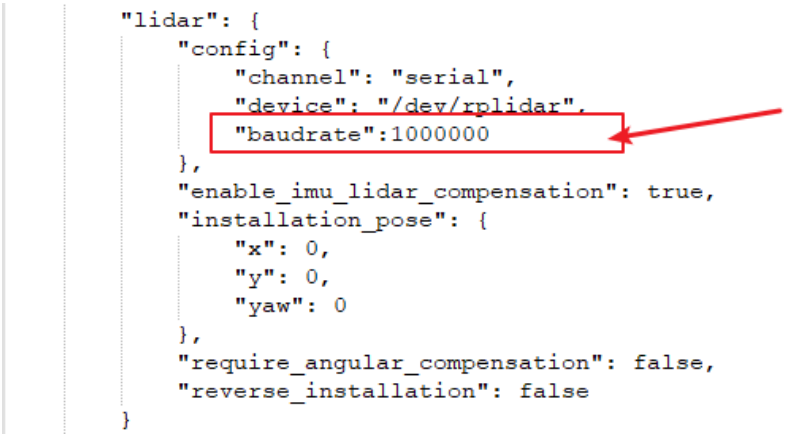
c) Create the container

Run the following command to create the container, where “u20” is the user defined container name and “slamkit:x86_64” is the image name defined in step 2.

```
sudo docker run -d --name u20 --privileged --network=host -v /dev:/dev slamkit:x86_64 /sbin/init
```

d) Import cube config

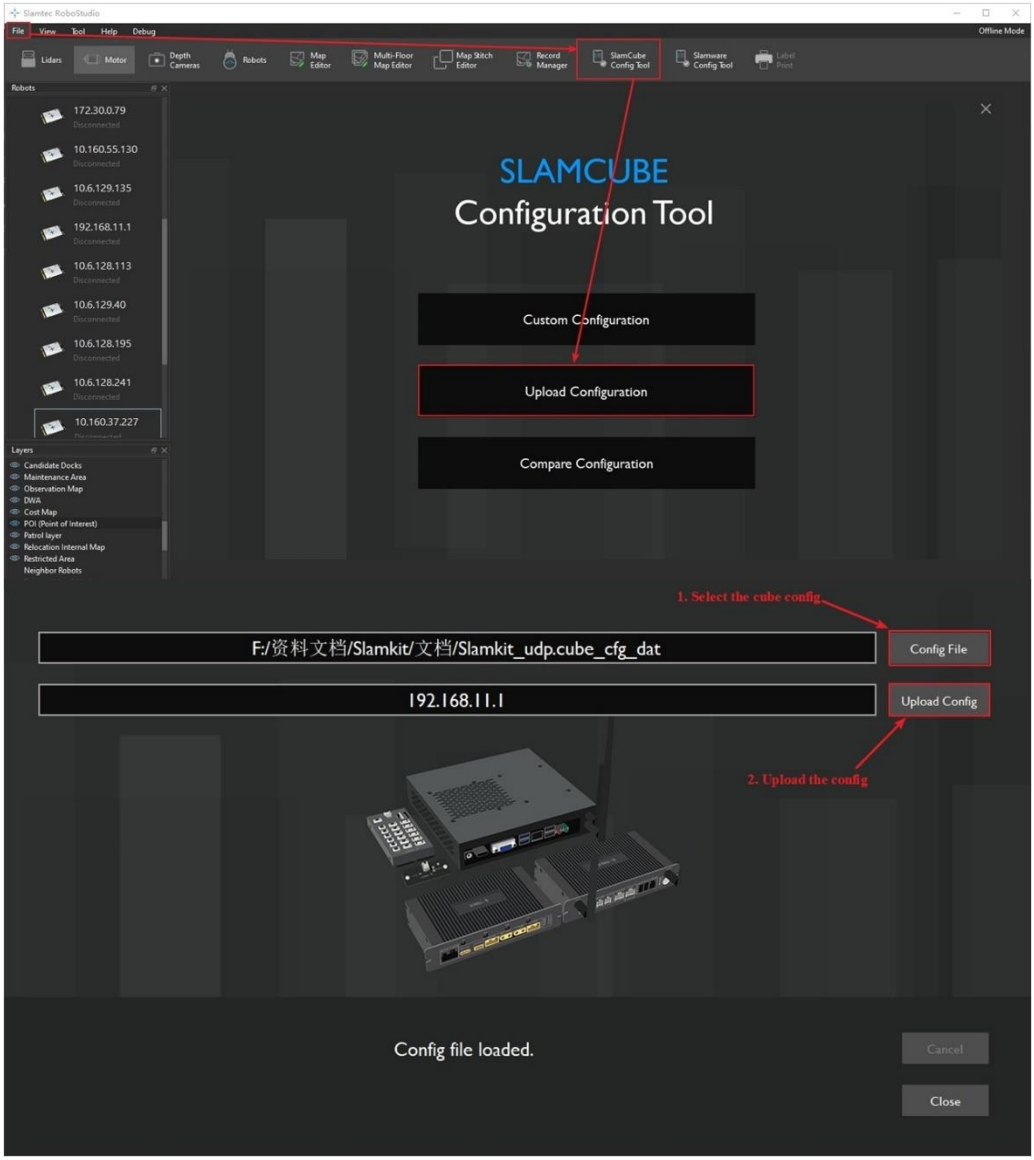
Download the corresponding cube config file based on the Lidar type being used. For example, “Slamkit_serial_quickstart.cube_cfg_dat” is for the Lidar with serial interface, and “Slamkit_udp_quickstart.cube_cfg_dat” is for the Lidar with ethernet interface. As shown in the figure below, when using serial Lidar, users need to modify the baud rate configuration in the cube config file accordingly.



The correspondence between Lidar types and baud rates is shown in the table below.

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

Launch the Robostudio and upload the cube config file as indicated in the figure below.



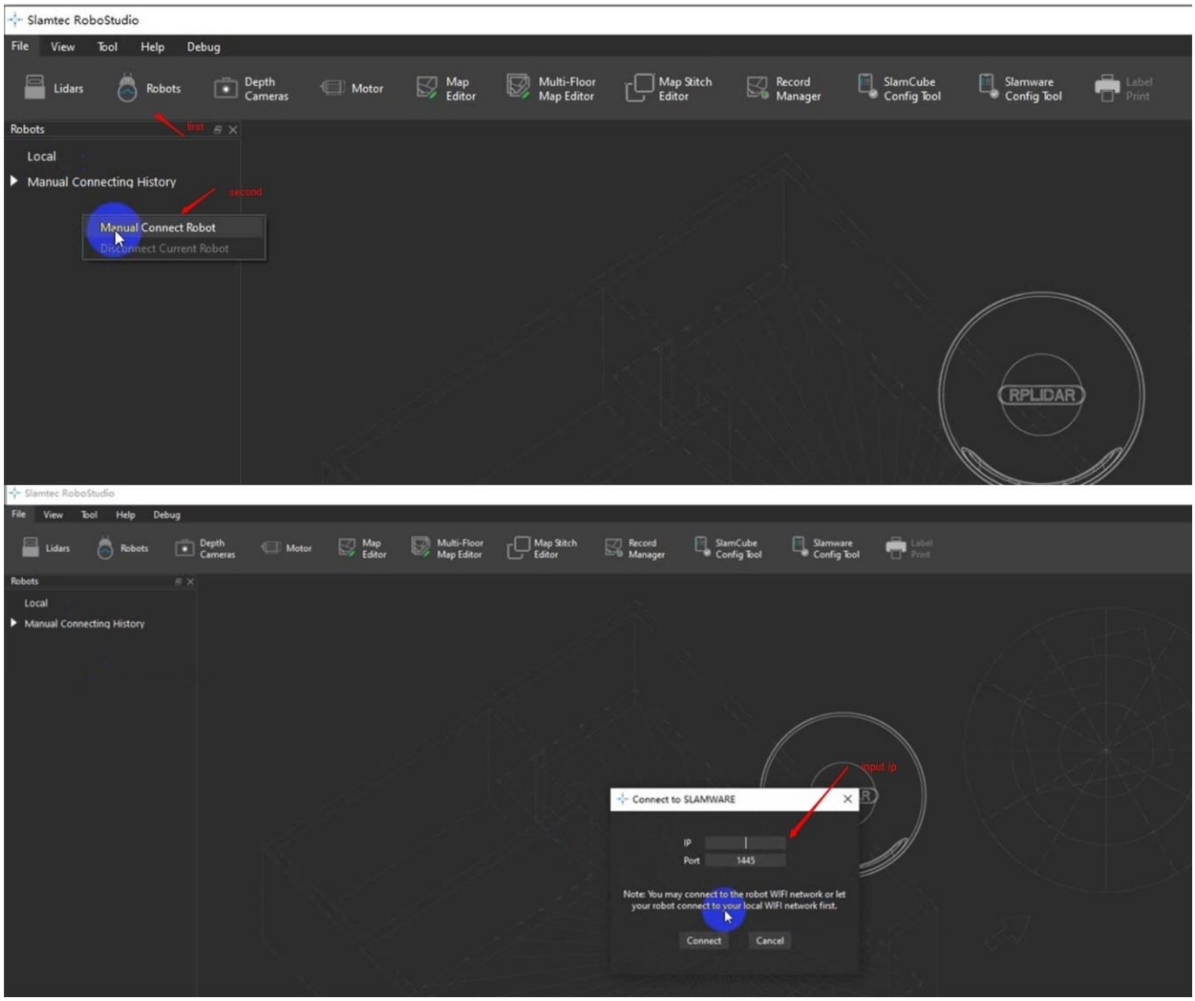
e) Restart the container

Run the following command to restart the container, where “u20” is the container name.

```
sudo docker restart u20
```

f) Connect SLAMKit

Follow the steps in the figure below to use the Robostudio for making a connection with SLAMKit .



g) Finish the installation

The quick start of SLAMKit is now complete. You can use Robostudio to experience mapping and localization applications. For more features and a better application experience with SLAMKit, please refer to the product user manual.