

# Livox SDK

Livox SDK is the software development kit designed for all Livox products. It is developed based on C/C++ following Livox Communication Protocol, and provides easy-to-use C style API. With Livox SDK, users can quickly connect to Livox products and receive point cloud data.

[#lidar](#) [#livox](#)

---

Last update: 2021-06-29 15:27:33

## Table of Content

1. Prerequisites
2. Install Livox SDK
3. Install Livox ROS Driver
  - 3.1. Install ROS
  - Install Livox ROS driver

## 1. Prerequisites

- Ubuntu 14.04/ 16.04/ 18.04, support x64 x86 and ARM (Nvidia TX2)
- Windows 7/ 10, Visual Studio 2015 Update3/ 2017/ 2019
- C++11 compiler

Livox SDK needs to be built in the host machine, therefore, some tool-chain and build tools have to be installed.

```
sudo apt-get update && \
sudo apt-get install -y build-essential && \
sudo apt-get install -y curl && \
sudo apt-get install -y git && \
sudo apt-get install -y cmake
```

## 2. Install Livox SDK

 <https://github.com/Livox-SDK/Livox-SDK>

```
git clone https://github.com/Livox-SDK/Livox-SDK.git
cd Livox-SDK
cd build
cmake ..
make
sudo make install
```

### ARM-Linux Cross Compile

The following commands will install C and C++ cross compiler tool-chains for 32bit and 64bit ARM board. You need to install the correct tool-chain for your ARM board. For 64bit SoC ARM board, only install 64bit tool-chain, and for 32bit SoC ARM board, only install 32bit tool-chain.

Install **ARM 32 bits** cross compile tool-chain:

```
sudo apt-get install gcc-arm-linux-gnueabi g++-arm-linux-gnueabi
```

Install **ARM 64 bits** cross compile tool-chain:

```
sudo apt-get install gcc-aarch64-linux-gnu g++-aarch64-linux-gnu
```

Cross Compile Livox-SDK for ARM 32 bits tool-chain , In the Livox SDK directory , run the following commands to cross compile the project:

```
cd Livox-SDK
cd build
```

```
cmake .. -DCMAKE_SYSTEM_NAME=Linux -DCMAKE_C_COMPILER=arm-linux-gnueabi-gcc -
DCMAKE_CXX_COMPILER=arm-linux-gnueabi-g++
make
sudo make install
```

For ARM 64 bits tool-chain, in the Livox SDK directory, run the following commands to cross compile the project:

```
cd Livox-SDK
cd build
cmake .. -DCMAKE_SYSTEM_NAME=Linux -DCMAKE_C_COMPILER=aarch64-linux-gnu-gcc -
DCMAKE_CXX_COMPILER=aarch64-linux-gnu-g++
make
sudo make install
```

Note:

gcc/g++ cross compiler need to support C ++11 standard

Read more in the GitHub page: <https://github.com/Livox-SDK/Livox-SDK>.

## 3. Install Livox ROS Driver

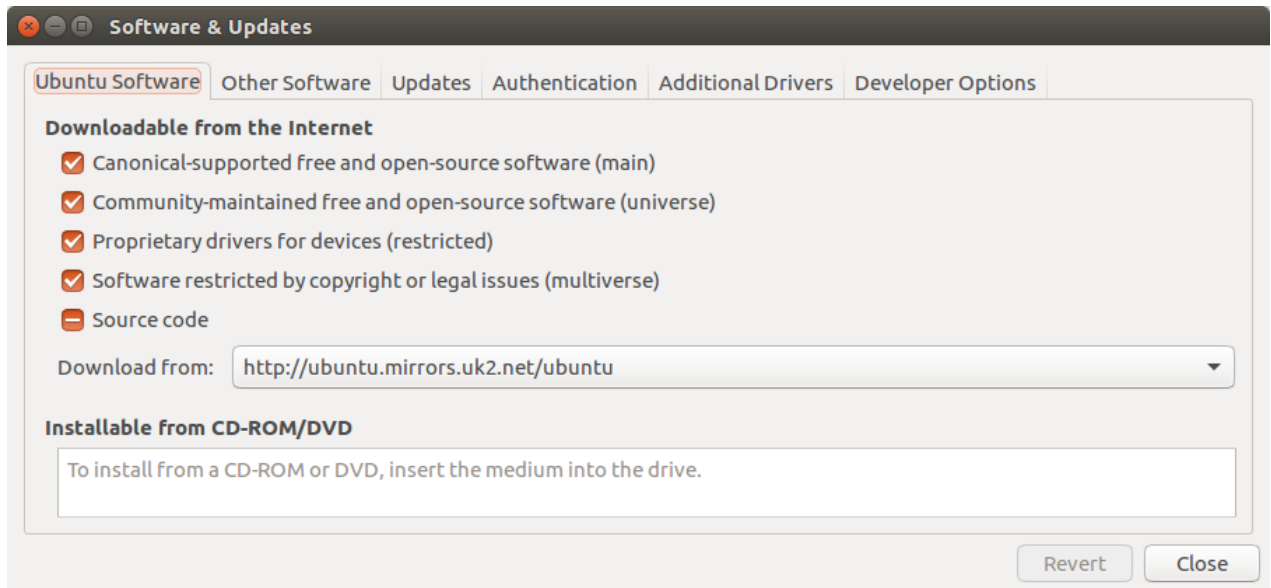
ROS driver can be run under Ubuntu 14.04/ 16.04/ 18.04 operating system with ROS environment (indigo, kinetic, melodic) installed.

### 3.1. Install ROS

ROS (Robot Operating System) provides libraries and tools to help software developers create robot applications. It provides hardware abstraction, device drivers, libraries, visualizers, message-passing, package management, and more. Be sure to install the full version of ROS (ros-distro-desktop-full).

#### 1. Configure Ubuntu repositories

Configure Ubuntu repositories to allow “restricted,” “universe,” and “multiverse.”



### *Ubuntu repositories*

## 2. Setup `sources.list`

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main"
> /etc/apt/sources.list.d/ros-latest.list'
```

## 3. Set up keys

```
sudo apt-get install -y curl
curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo
apt-key add -
```

## 4. Refresh repos

```
sudo apt-get update
```

## 5. Install ROS

ROS has different models, such as Melodic, Noetic, etc. See current models in <https://www.ros.org/install/>.

### **i What is the difference between ROS Melodic model and Noetic model?**

There aren't many differences at the base level. The ROS Noetic is recommended for Ubuntu 20.04 whereas ROS Melodic for Ubuntu 18.04:

Feature	ROS Noetic	ROS Melodic
Python	3.8	2.7
Gazebo	11.x	9.0
OpenCV	4.2	3.2

Detailed comparison is at [repositories.ros.org](https://repositories.ros.org).

Choose the ROS version based on the installed OS:

```
sudo apt install ros-melodic-desktop-full
```

or

```
sudo apt install ros-noetic-desktop-full
```

It's convenient if the ROS environment variables are automatically added to a bash session every time a new shell is launched:


```
echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc
source ~/.bashrc
```

If there are more than one ROS distribution installed, `~/.bashrc` must only source the `setup.bash` for the version which is currently being used.

## Install Livox ROS driver


Get `livox_ros_driver` from GitHub

```
git clone https://github.com/Livox-SDK/livox_ros_driver.git ws_livox/src
```

 Be sure to use the above command to clone the code to the local, otherwise it will compile error due to the file path problem.

Then build it:

```
cd ws_livox
catkin_make
```

 If running `catkin_make` gives error of command not found, it's probably that the ROS `setup.bash` is not executed and included in `~/.bashrc`. See above section to source it.

Read more in [https://github.com/Livox-SDK/livox\\_ros\\_driver](https://github.com/Livox-SDK/livox_ros_driver).