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 #ros

Last update: 2021-07-23 14:38:22

Table of Content

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. Dependencies

2.1. Point Cloud Library

The Point Cloud Library (PCL) is a large scale, open project[1] for point cloud processing. The PCL framework contains numerous state-of-the art algorithms including filtering, feature estimation, surface reconstruction, registration, model fitting and segmentation.

```
sudo apt install -y libpcl-dev
```

2.2. Eigen

Eigen is a C++ template library for linear algebra: matrices, vectors, numerical solvers, and related algorithms.

```
sudo apt install -y libeigen3-dev
```

2.3. OpenCV

OpenCV (Open Source Computer Vision Library) is an open-source computer vision library and has bindings for C++, Python, and Java. It is used for a very wide range of applications, including medical image analysis, stitching street view images, surveillance video, detecting and recognizing faces, tracking moving objects, extracting 3D models, and much more. OpenCV can take advantage of multi-core processing and features GPU acceleration for real-time operation.

```
sudo apt install -y python-opencv python3-opencv
```

2.4. Re-link libraries

```
sudo ln -s /usr/bin/vtk6 /usr/bin/vtk
sudo ln -s /usr/lib/python2.7/dist-
packages/vtk/libvtkRenderingPythonTkWidgets.x86_64-linux-gnu.so /usr/lib/x86_64-
linux-gnu/libvtkRenderingPythonTkWidgets.so
```

3. Install Livox SDK

The official guide is at 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 that `gcc` / `g++` cross compiler need to support C ++11 standard.

4. 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.

4.1. 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).

Refer to Install ROS guide to complete the installation.

Main steps include:

1. Setup source list to get ROS packages:

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

2. Add keys:

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

3. Then pull the package list:

```
sudo apt update
```

4. Finally, install a desktop-full package as recommended to start learning:

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

4.2. 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.

4.3. Missing features

Even the build is completed, there are still some warning during the compilation due to missing definitions.

```
-- Could NOT find ensenso (missing: ENSENSO_LIBRARY ENSENSO_INCLUDE_DIR)
** WARNING ** io features related to ensenso will be disabled
-- Could NOT find DAVIDSDK (missing: DAVIDSDK_LIBRARY DAVIDSDK_INCLUDE_DIR)
** WARNING ** io features related to davidSDK will be disabled
-- Could NOT find DSSDK (missing: _DSSDK_LIBRARIES)
** WARNING ** io features related to dssdk will be disabled
** WARNING ** io features related to pcap will be disabled
** WARNING ** io features related to png will be disabled
-- Found libusb-1.0: /usr/include
** WARNING ** io features related to libusb-1.0 will be disabled
-- Checking for module 'flann'
--   Found flann, version 1.9.1
-- Found Flann: /usr/lib/x86_64-linux-gnu/libflann_cpp_s.a
-- Could NOT find ensenso (missing: ENSENSO_LIBRARY ENSENSO_INCLUDE_DIR)
** WARNING ** visualization features related to ensenso will be disabled
-- Could NOT find DAVIDSDK (missing: DAVIDSDK_LIBRARY DAVIDSDK_INCLUDE_DIR)
** WARNING ** visualization features related to davidSDK will be disabled
-- Could NOT find DSSDK (missing: _DSSDK_LIBRARIES)
** WARNING ** visualization features related to dssdk will be disabled
-- Could NOT find RSSDK (missing: _RSSDK_LIBRARIES)
** WARNING ** visualization features related to rssdk will be disabled
```

At this time, they do not harm, then these warning may be solved later.