

---

## SynexensROS1 Instruction v4.0.1

---

Revision history version					
Date	ROS version	SDK version	Documentation version	Description	Author
202212114	v4.0.1	v4.0.3.0	v4.0.1	Initial version	YSY

## Catalog

Steps 1 Preface.....	33
2. Overview.....	33
Build and run.....	33
3.1. Compile.....	33
3.1.1. Package file directory knot.....	33
3.1.2. catkin compilation.....	55
3.1.3. Workspace uses synexens_ros1.....	55
3.1.4. Compile and run issues summary.....	56
3.1.5. Nodes provide topics.....	66
3.1.6. Server parameter function at startup.....	66
3.2 SDK replacement (e.g., needs to run on arm platform).....	7
3.2.1. armv8 platform replacement steps.....	7
4. Precautions.....	78
4.1. PointCloud size issues.....	78
Disclaimer.....	8

## 1. Preface

This ROS is developed based on SDK4.0+, please read the SDK instructions in advance before using. The program only realizes part of the function, please use according to your actual needs.

## 2. Overview

Supported devices: cs20 Single frequency cs20 dual frequency cs30 single frequency cs30 dual frequency CS20-P cs40

Supported system: ubuntu20.04\_x86 ubuntu18.04\_x86

Supported ROS version: Noetic Melodic

## 3. Compile and run

### 3.1. Compile

#### 3.1.1. Package file directory knot

synexens\_ros1

├─ CMakeLists.txt

├─ ext

- | └─ sdk
- |   └─ include
- |   └─ lib
- |   └─ opencv
- └─ include
- └─ synexens\_ros1
- └─ SYCalibrationTransformData.h
- └─ SYRosDevice.h
- └─ SYRosDeviceParmas.h
- └─ SYRosTypes.h
- └─ launch
- └─ driver.launch
- └─ viewer.launch
- └─ package.xml
- └─ README.md
- └─ rviz
- └─ view.rviz
- └─ script
- └─ setup.sh
- └─ synexens-usb.rules
- └─ src
- └─ SYCalibrationTransformData.cpp

| ┆ SYRosDevice.cpp  
| ┆ SYRosDeviceParams.cpp  
| ┆ SYRosNode.cpp

Core code files: include/synexens\_ros1/\*.hsrc /\*.cpp Main node functions

Core package file: CMakeLists.txt package.xml ROS package core file

SDK dependency: ext/sdk synexensSDK4.0 dependency library

Rviz visualization file: rviz/view.rviz Rviz configuration

USB Rules: scripts/synexens-usb.rules setup.sh USB permission file

launch file: launch/\*.launch ROSLaunch launch launch file

### 3.1.2. catkin compile

1. Decompress the zip file in your Linux system
2. Copy the synexens\_ros1 package to your workspace under catkin\_ws(name custom)/src folder
3. Run the compile command: `$cd catkin_ws && catkin_make`

### 3.1.3. Use synexens\_ros1 for the workspace

1. Run the ROS core roscore
2. `$ cd catkin_ws && source ./devel/setup.bash`
3. `$ roslaunch synexens_ros1 driver.launch/viewer.launch`

### 3.1.4. Compile run issue summary

1. Missing libraries: You need to copy the.so file from the ext/sdk/lib folder that the sdk depends on to the catkin\_ws/devel/lib directory and run it again
2. Compile time alert for missing libraries: **Be sure to unpack files under Linux when unpacking them**
3. Camera cannot be opened, no permissions: need to run script/setup.sh

### 3.1.5. Node provides topic

Since multiple devices can be connected to configure whether topics are displayed or not, topic communication is not fixed.

xxx/depth\_raw (' sensor\_msgs::Image ') depth image data

xxx/depth\_info (' sensor\_msgs::CameraInfo ') depth camera information

xxx/ir\_raw (' sensor\_msgs::Image ') IR image data

xxx/ir\_info (' sensor\_msgs::CameraInfo ') IR camera information

xxx/rgb\_raw (' sensor\_msgs::Image ') RGB image data

xxx/points2 (' sensor\_msgs::PointCloud2 ') point cloud image data

### 3.1.6. Server parameter function at startup

Parameter configuration details can be found in the driver.launch file.

Such as the need to add or modify the parameters, you can refer to include/synexens\_ros1 / SYRosDeviceParams h SRC/SYRosDeviceParams. The configuration parameters for the CPP file, src/SYRosDevice.cpp->SetOption to invoke SDK Settings.

## **SDK replacement (e.g., to run on arm)**

Different platforms rely on different SDKS. If we need to run on a different platform (for example, armv8), we need to find the SDK version for that platform and manually copy it into the ext/sdk directory, replacing the libraries and headers.

### **3.2.1. armv8 Platform replacement steps**

1. Find the SDK for the corresponding platform version and make sure it is working properly

Replace ext/sdk/include/\*.h

3. Replace ext/sdk/lib/\*.so

Replace ext/sdk/opencv/\*.so

5. Replace devel/lib/\*.so at runtime

**Note: Linux system SDK is best to use tar to package, decompress to be carried out in Linux. To ensure executable permissions and library files soft connection.**

## 4. Considerations

### 4.1. PointCloud size issues

Due to the rviz GUI tool display issues, the actual point cloud data is 1000 times larger than the data in ROS. The point relationship between the point cloud saved through the GUI and the ROS point cloud coincides, and the size difference is 1000 times.

## Disclaimer

The device application information and other similar content described in this publication is for your convenience only and may be superseded by updated information. It is your own responsibility to ensure that the application complies with the technical specifications. We make no representations or warranties, express or implied, written or oral, statutory or otherwise, including, but not limited to, representations or warranties regarding its use, quality, performance, merchantability or fitness for a particular purpose. The Company disclaims any liability for such information and for consequences arising out of its use. This product may not be used as a critical component in a life support system without written approval from the Company.