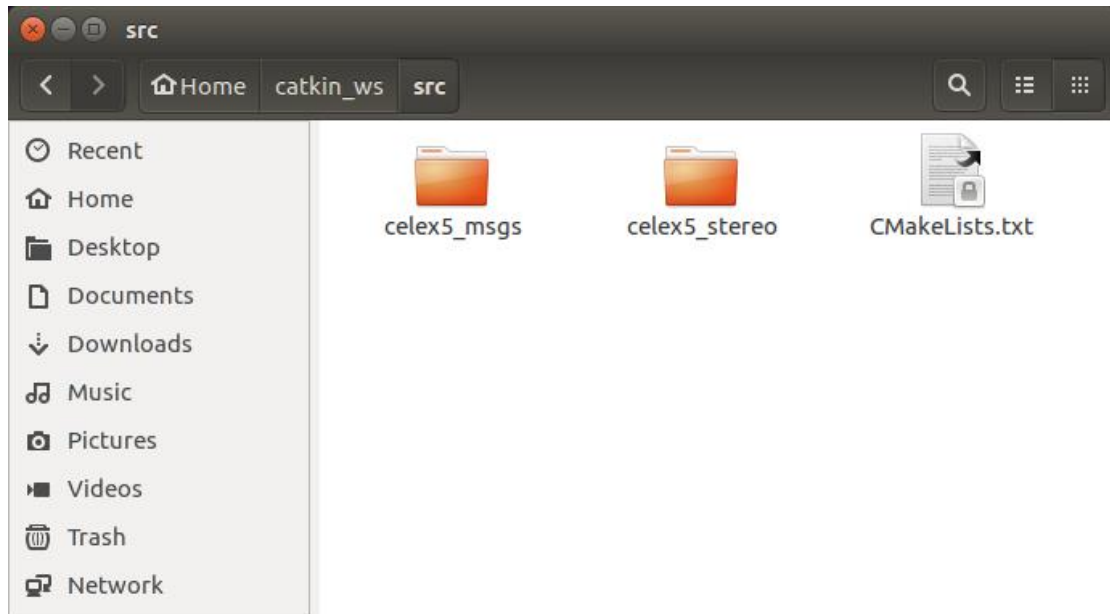
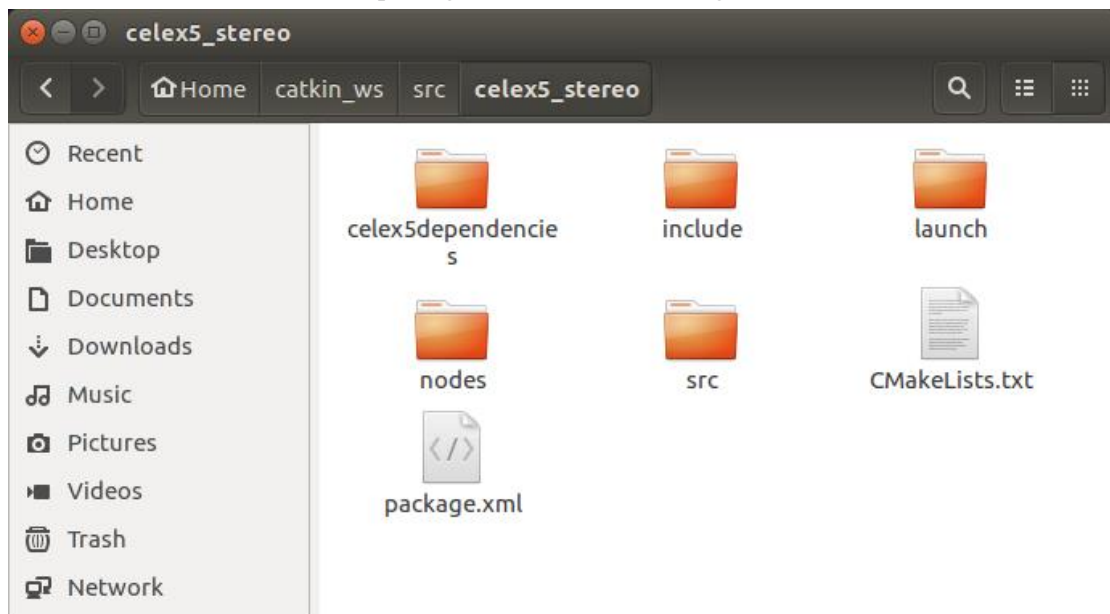


1 Introduction

The sample code file in the ROS environment under the release directory “**Sample-ROS**” mainly includes two packages (celex5_msgs and celex5_monocular), wherein the ***celex5_msgs*** is a custom ROS message package, and the ***celex5_monocular*** is a CeleX5 monocular function package. This example is compiled and run under Ubuntu 16.04 based on the Kinetic version of the ROS environment.



The CeleX5 monocular function package includes the following five folders and files.

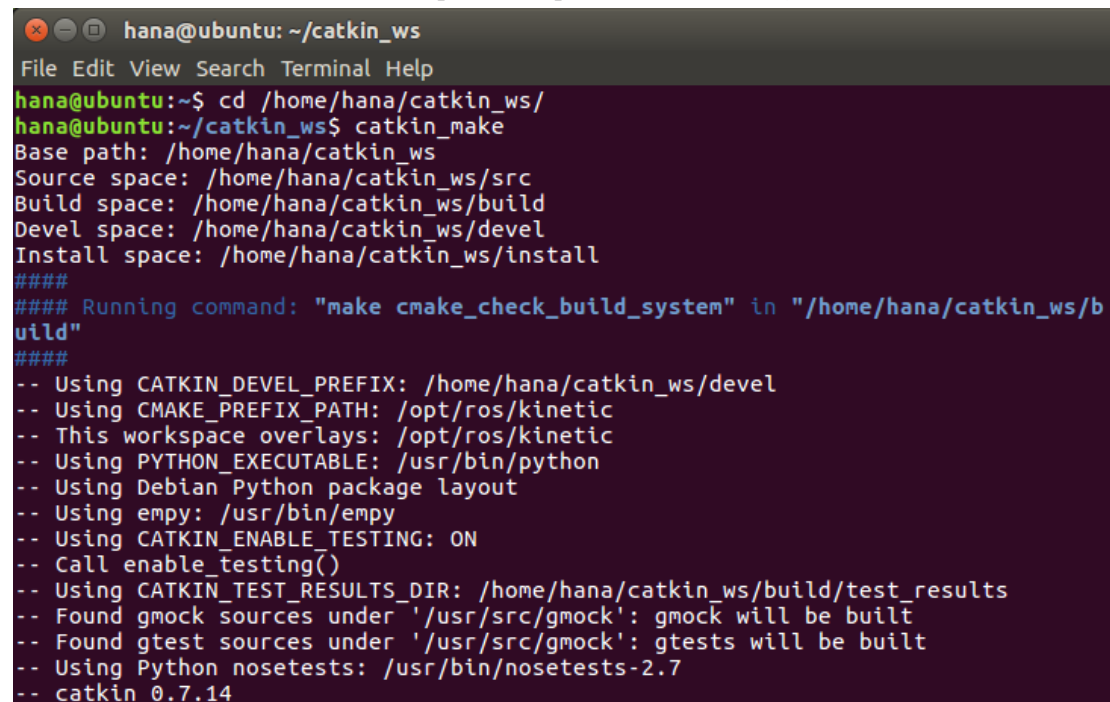


- ✧ **celex5dependencies:** This folder includes the API header files and library files. (for Ubuntu 16.04)
- ✧ **include:** This folder holds the header file.
- ✧ **launch:** The roslaunch startup file is stored in this file.
- ✧ **nodes:** This folder holds the startup node file for rosrn.

- ✧ **src:** The source file is stored in this folder.
- ✧ **CMakeLists.txt:** Used for compilation of executable files.
- ✧ **Package.xml:** Describe the properties of the package.

2 Compile ROS Sample Package

Place the *celex5_msgs* and *celex5_monocular* folders in the *src* directory of the created ROS workspace to compile. As shown in the following figure, the current ROS workspace is named *catkin_ws*. Enter the workspace and use the *catkin_make* command to compile all catkin projects in the *src* directory. (**Note:** Compilation depends on OpenCV, users need to configure the OpenCV environment under Ubuntu, this example uses OpenCV version 3.3.0)



```

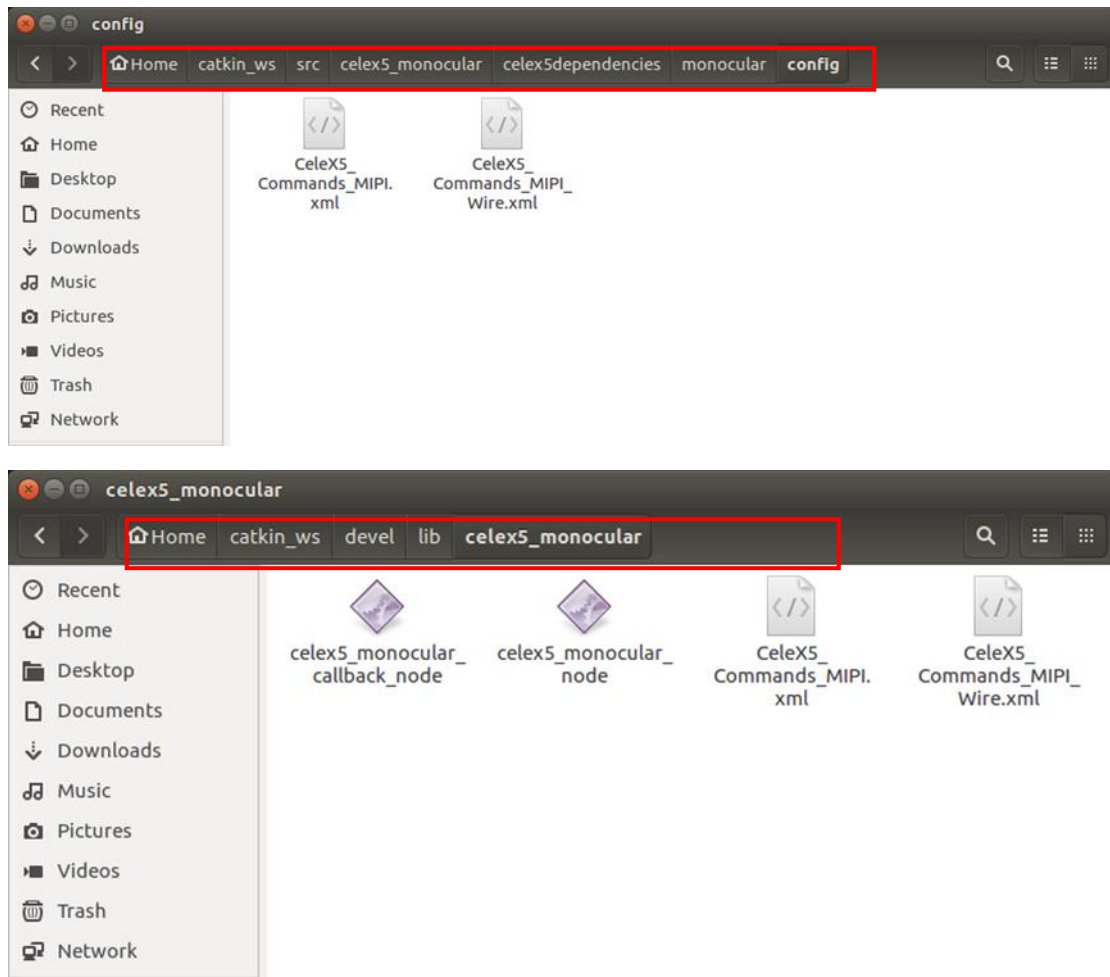
hana@ubuntu: ~/catkin_ws
File Edit View Search Terminal Help
hana@ubuntu:~$ cd /home/hana/catkin_ws/
hana@ubuntu:~/catkin_ws$ catkin_make
Base path: /home/hana/catkin_ws
Source space: /home/hana/catkin_ws/src
Build space: /home/hana/catkin_ws/build
Devel space: /home/hana/catkin_ws/devel
Install space: /home/hana/catkin_ws/install
####
#### Running command: "make cmake_check_build_system" in "/home/hana/catkin_ws/build"
####
-- Using CATKIN_DEVEL_PREFIX: /home/hana/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /opt/ros/kinetic
-- This workspace overlays: /opt/ros/kinetic
-- Using PYTHON_EXECUTABLE: /usr/bin/python
-- Using Debian Python package layout
-- Using empy: /usr/bin/empy
-- Using CATKIN_ENABLE_TESTING: ON
-- Call enable_testing()
-- Using CATKIN_TEST_RESULTS_DIR: /home/hana/catkin_ws/build/test_results
-- Found gmock sources under '/usr/src/gmock': gmock will be built
-- Found gtest sources under '/usr/src/gmock': gtests will be built
-- Using Python nosetests: /usr/bin/nosetests-2.7
-- catkin 0.7.14

```

3 Run ROS Sample Package

After compilation, the executable files *celex5_monocular_node* (direct mode) and *celex5_monocular_callback_node* (callback mode) will be generated in the */devel/lib/celex5_monocular/* directory of the workspace.

The users need to copy the necessary .xml configuration file (*/home/YOUR_WORKSPACE/src/celex5_monocular/celexdependencies/monocular/config/*.xml*) to the executable directory (*/home/YOUR_WORKSPACE/devel/lib/celex5_monocular/*).



Before running the ROS package, you must first run *roscore*. Then we can use *roslaunch* or *roslaunch* to run the node. Since the CeleX5 needs to read and write to USB devices, so you need to use root permissions. In order to ensure the successful running of the node program, we first enter the root permission directly.

```
root@ubuntu: /home/hana/catkin_ws
hana@ubuntu:~/catkin_ws$ sudo su
[sudo] password for hana:
root@ubuntu: /home/hana/catkin_ws# source ./devel/setup.bash
root@ubuntu: /home/hana/catkin_ws# rosrn celex5_monocular celex
celex5datamanager.h
celex5.h
celex5_monocular_callback.launch
celex5_monocular_callback_node
celex5_monocular.launch
celex5_monocular_node
root@ubuntu: /home/hana/catkin_ws# rosrn celex5_monocular celex5_monocular_node
XBase::getApplicationDirPath: readlink count = 69
XBase::getApplicationDirPath: readlink count = 69

***** HHXmlReader::importCommands_Celex5 Begin *****
***** HHXmlReader::importCommands_Celex5 End *****

--- Disable PLL ---

--- Load PLL Parameters ---
Celex5::writeCSRDefaults: PLL_Parameters
--- Enable PLL ---

--- Disable MIPI ---
```

If there is a package *** not found or rosrn and other commands is invalid, you can use *source ./devel/setup.bash* to refresh the environment.

```
/home/hana/catkin_ws/src/celex5_monocular/launch/celex5_monocular.launch http://lo
root@ubuntu: /home/hana/catkin_ws# roslaunch celex5_monocular celex5_monocular.la
unch
... logging to /root/.ros/log/5a1496fc-81be-11e9-b0a1-000c29876e43/roslaunch-ubu
ntu-8853.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

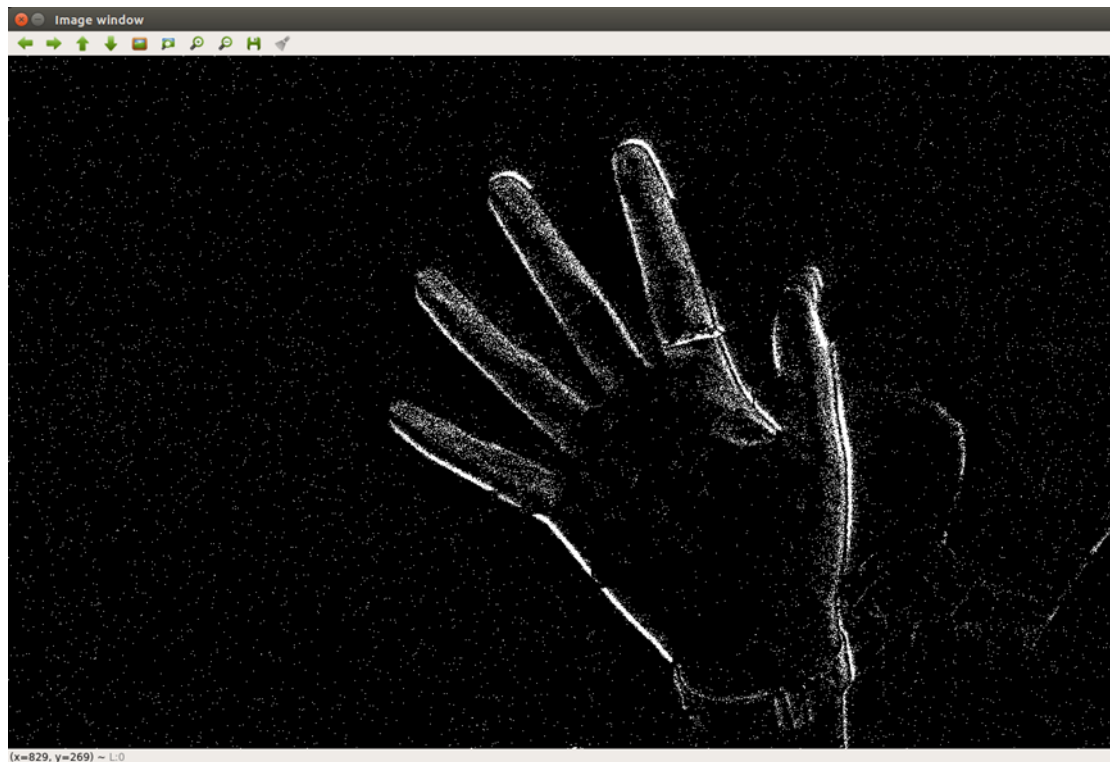
started roslaunch server http://ubuntu:37247/

SUMMARY
=====

PARAMETERS
* /celex_monocular/celex_mode: Event_Address_Onl...
* /celex_monocular/clock_rate: 100
* /celex_monocular/threshold: 170
* /rostdistro: kinetic
* /rosversion: 1.12.14

NODES
/
  celex_monocular (celex5_monocular/celex5_monocular_node)
```

You can see the image windows after rosrn or roslaunch the node.



You can also view image information published by `/imgshow` via rviz.

