YUJIN ROBOT

YUJIN AMS

YUJIN 3D LiDAR

YRL3V2 Series

Driver Package Operations Guide

Revision: 1.0
Date: 2022/04

YUJIN ROBOT

All logos and trademarks indicated in this document are legally protected as part of Yujin Robot's intellectual property rights. No part of this document may be reproduced in any form without the explicit written consent of Yujin Robot. The appearance and specifications of the product are subject to change without notice for product improvement.

Copyright and Disclaimer

The information in this guide is subject to change without notice during the product’s life cycle. The printed version of the guide is updated periodically. Some details may be omitted or inaccurate when compared to current product information. Even in the hypothetical event that Yujin Robot was aware of the possibility of any damages ahead of time, Yujin Robot assumes no responsibility for errors, omissions, or damages that may arise from the information contained in this document.

Change History

The following table contains version information for this document and a history of significant changes.

Version	Date of Writing	Changes
V1.0	Jan 10, 2022	Draft

Table of Contents

1. Introduction

Yujin Robot provides a driver package for YUJIN LiDAR.

You can install and use the driver package in Ubuntu OS.

- Ubuntu is an open source OS.
- How to enter commands in Ubuntu OS: Run “**ctrl +alt + t**” on the desktop to enable the Terminal window and enter the desired command.

Note: This document is prepared for users of Ubuntu 20.04.

2. Installing the driver package

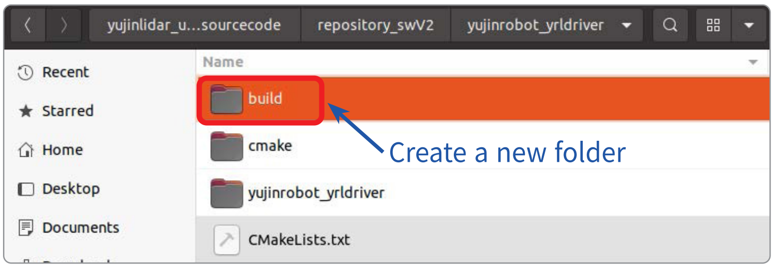
The driver package folder “yujinrobot_yrldriver” contains the following items:

- build (Create a new “build” folder.)
- cmake
- yujinrobot_yrldriver
- CMakeLists.txt

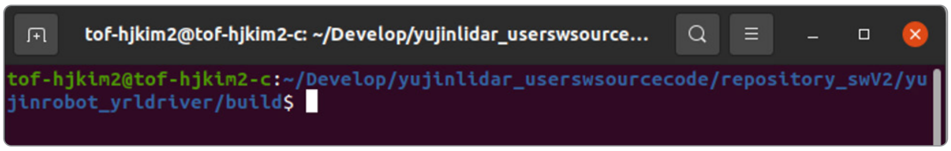
2.1 Creating a build folder

Create a new folder named “build”.

1. Create a new “build” folder under the “yujinrobot_yrldriver” folder.



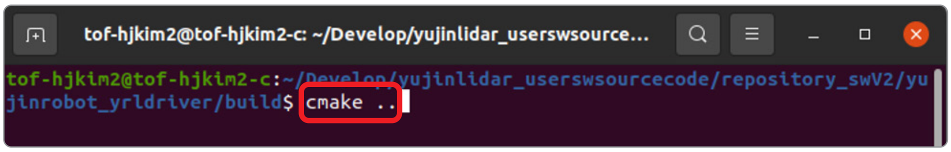
2. Launch the Terminal window in the newly created “build” folder.


tof-hjkim2@tof-hjkim2-c: ~/Develop/yujinlidar_userswsource...

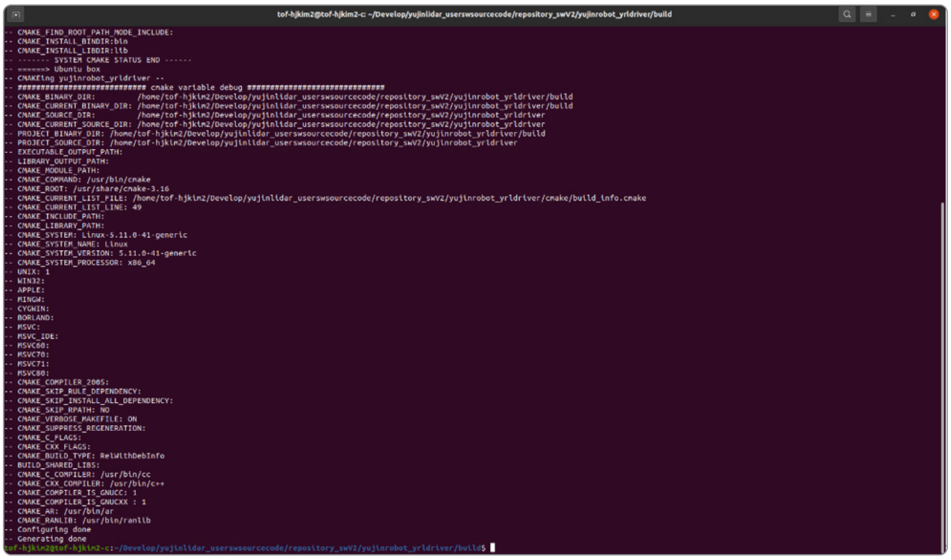
```
tof-hjkim2@tof-hjkim2-c:~/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build$ 
```

3. Enter the command to set up the compile preferences.

- Command: cmake …

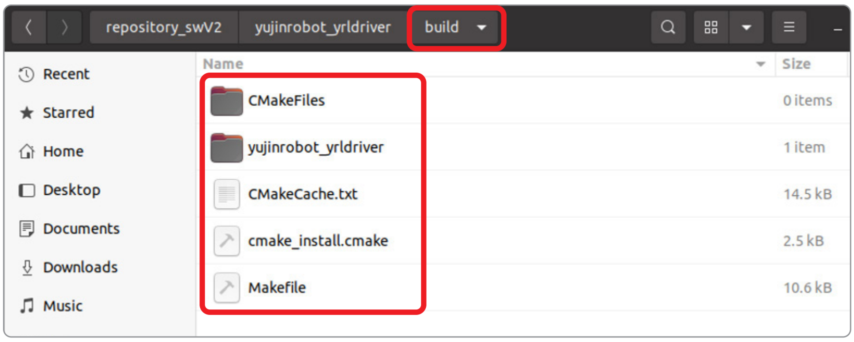

tof-hjkim2@tof-hjkim2-c: ~/Develop/yujinlidar_userswsource...

```
tof-hjkim2@tof-hjkim2-c:~/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build$ cmake ..
```



```
tof-hjkim2@tof-hjkim2-c: ~/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build
-- CMAKE_FIND_ROOT_PATH_MODE_INCLUDE:
-- CMAKE_INSTALL_BINDIR:bin
-- CMAKE_INSTALL_LIBDIR:lib
-- ------- SYSTEM CMAKE STATUS END ------
-- ======> Ubuntu box
-- CMAKEing yujinrobot_yrldriver --
-- ############################ cmake variable debug ##############################
-- CMAKE_BINARY_DIR:         /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build
-- CMAKE_CURRENT_BINARY_DIR: /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build
-- CMAKE_SOURCE_DIR:         /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver
-- CMAKE_CURRENT_SOURCE_DIR: /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver
-- PROJECT_BINARY_DIR: /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build
-- PROJECT_SOURCE_DIR: /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver
-- EXECUTABLE_OUTPUT_PATH:
-- LIBRARY_OUTPUT_PATH:
-- CMAKE_MODULE_PATH:
-- CMAKE_COMMAND: /usr/bin/cmake
-- CMAKE_ROOT: /usr/share/cmake-3.16
-- CMAKE_CURRENT_LIST_FILE: /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/cmake/build_info.cmake
-- CMAKE_CURRENT_LIST_LINE: 49
-- CMAKE_INCLUDE_PATH:
-- CMAKE_LIBRARY_PATH:
-- CMAKE_SYSTEM: Linux-5.11.0-41-generic
-- CMAKE_SYSTEM_NAME: Linux
-- CMAKE_SYSTEM_VERSION: 5.11.0-41-generic
-- CMAKE_SYSTEM_PROCESSOR: x86_64
-- UNIX: 1
-- WIN32:
-- APPLE:
-- MINGW:
-- CYGWIN:
-- BORLAND:
-- MSVC:
-- MSVC_IDE:
-- MSVC60:
-- MSVC70:
-- MSVC71:
-- MSVC80:
-- CMAKE_COMPILER_2005:
-- CMAKE_SKIP_RULE_DEPENDENCY:
-- CMAKE_SKIP_INSTALL_ALL_DEPENDENCY:
-- CMAKE_SKIP_RPATH: NO
-- CMAKE_VERBOSE_MAKEFILE: ON
-- CMAKE_SUPPRESS_REGENERATION:
-- CMAKE_C_FLAGS:
-- CMAKE_CXX_FLAGS:
-- CMAKE_BUILD_TYPE: RelWithDebInfo
-- BUILD_SHARED_LIBS:
-- CMAKE_C_COMPILER: /usr/bin/cc
-- CMAKE_CXX_COMPILER: /usr/bin/c++
-- CMAKE_COMPILER_IS_GNUCC: 1
-- CMAKE_COMPILER_IS_GNUCXX : 1
-- CMAKE_AR: /usr/bin/ar
-- CMAKE_RANLIB: /usr/bin/ranlib
-- Configuring done
-- Generating done
tof-hjkim2@tof-hjkim2-c:~/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build$
```

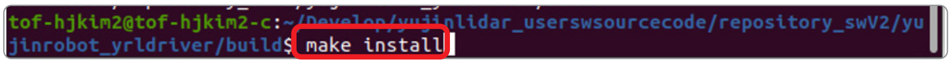
4. Check if all of the files below are created in the “build” folder.



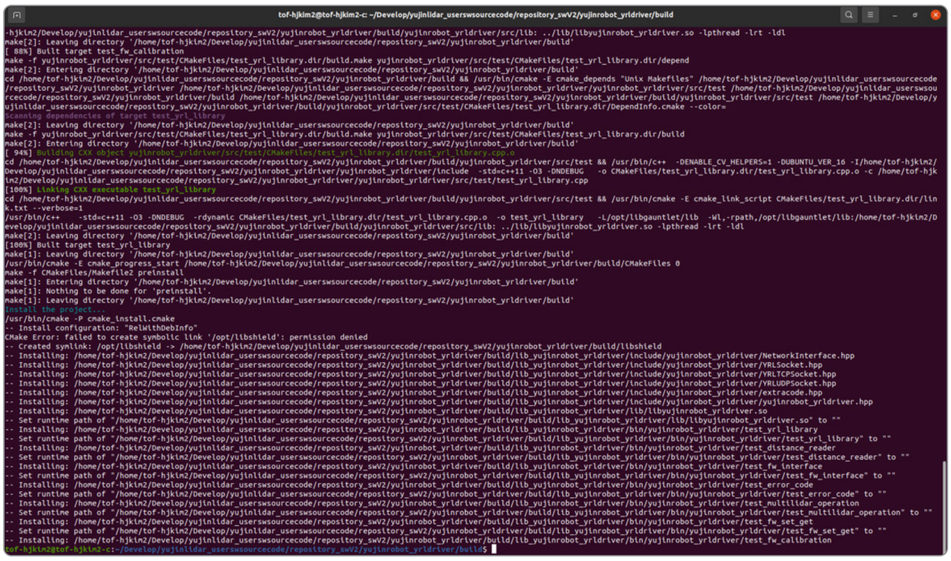
Name	Size
CMakeFiles	0 items
yujinrobot_yrldriver	1 item
CMakeCache.txt	14.5 kB
cmake_install.cmake	2.5 kB
Makefile	10.6 kB

5. Enter the command to compile the source code.

- Command: make install

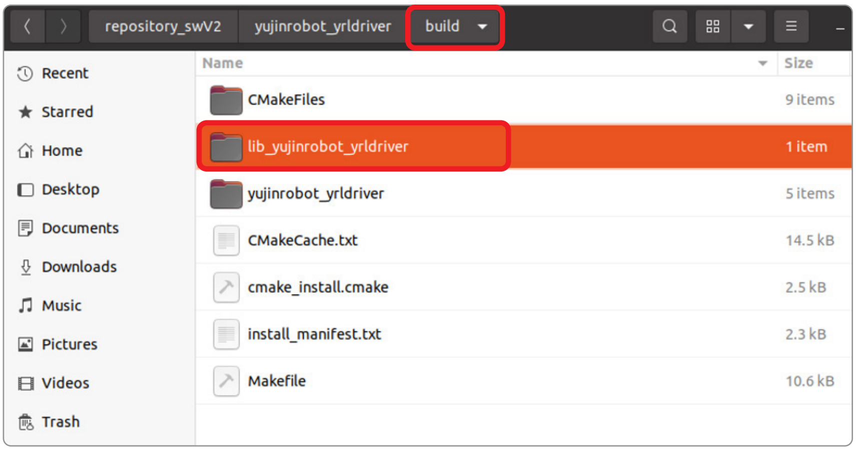


```
tof-hjkim2@tof-hjkim2-c:~/Develop/yujinlidar_userswsourcecode/repository_swV2/yu
jinrobot_yrldriver/build$ make install
```



```
tof-hjkim2@tof-hjkim2-c: ~/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build
-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/yujinrobot_yrldriver/src/lib: ../lib/libyujinrobot_yrldriver.so -lpthread -lrt -ldl
make[2]: Leaving directory '/home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build'
[ 88%] Built target test_fw_calibration
make -f yujinrobot_yrldriver/src/test/CMakeFiles/test_yrl_library.dir/build.make yujinrobot_yrldriver/src/test/CMakeFiles/test_yrl_library.dir/depend
make[2]: Entering directory '/home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build'
cd /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build && /usr/bin/cmake -E cmake_depends "Unix Makefiles" /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode
/repository_swV2/yujinrobot_yrldriver /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/yujinrobot_yrldriver/src/test /home/tof-hjkim2/Develop/yujinlidar_userswsou
rcecode/repository_swV2/yujinrobot_yrldriver/build /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/yujinrobot_yrldriver/src/test /home/tof-hjkim2/Develop/y
ujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/yujinrobot_yrldriver/src/test/CMakeFiles/test_yrl_library.dir/DependInfo.cmake --color=
Scanning dependencies of target test_yrl_library
make[2]: Leaving directory '/home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build'
make -f yujinrobot_yrldriver/src/test/CMakeFiles/test_yrl_library.dir/build.make yujinrobot_yrldriver/src/test/CMakeFiles/test_yrl_library.dir/build
make[2]: Entering directory '/home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build'
[ 94%] Building CXX object yujinrobot_yrldriver/src/test/CMakeFiles/test_yrl_library.dir/test_yrl_library.cpp.o
cd /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/yujinrobot_yrldriver/src/test && /usr/bin/c++  -DENABLE_CV_HELPERS=1 -DUBUNTU_VER_16 -I/home/tof-hjkim2/
Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/yujinrobot_yrldriver/include  -std=c++11 -O3 -DNDEBUG   -o CMakeFiles/test_yrl_library.dir/test_yrl_library.cpp.o -c /home/tof-hjk
im2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/yujinrobot_yrldriver/src/test/test_yrl_library.cpp
[100%] Linking CXX executable test_yrl_library
cd /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/yujinrobot_yrldriver/src/test && /usr/bin/cmake -E cmake_link_script CMakeFiles/test_yrl_library.dir/lin
k.txt --verbose=1
/usr/bin/c++    -std=c++11 -O3 -DNDEBUG  -rdynamic CMakeFiles/test_yrl_library.dir/test_yrl_library.cpp.o  -o test_yrl_library   -L/opt/libgauntlet/lib  -Wl,-rpath,/opt/libgauntlet/lib:/home/tof-hjkim2/D
evelop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/yujinrobot_yrldriver/src/lib: ../lib/libyujinrobot_yrldriver.so -lpthread -lrt -ldl
make[2]: Leaving directory '/home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build'
[100%] Built target test_yrl_library
make[1]: Leaving directory '/home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build'
/usr/bin/cmake -E cmake_progress_start /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/CMakeFiles 0
make -f CMakeFiles/Makefile2 preinstall
make[1]: Entering directory '/home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build'
make[1]: Nothing to be done for 'preinstall'.
make[1]: Leaving directory '/home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build'
Install the project...
/usr/bin/cmake -P cmake_install.cmake
-- Install configuration: "RelWithDebInfo"
CMake Error: failed to create symbolic link '/opt/libshield': permission denied
-- Created symlink: /opt/libshield -> /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/libshield
-- Installing: /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/lib_yujinrobot_yrldriver/include/yujinrobot_yrldriver/NetworkInterface.hpp
-- Installing: /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/lib_yujinrobot_yrldriver/include/yujinrobot_yrldriver/YRLSocket.hpp
-- Installing: /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/lib_yujinrobot_yrldriver/include/yujinrobot_yrldriver/YRLTCPSocket.hpp
-- Installing: /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/lib_yujinrobot_yrldriver/include/yujinrobot_yrldriver/YRLUDPSocket.hpp
-- Installing: /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/lib_yujinrobot_yrldriver/include/yujinrobot_yrldriver/extracode.hpp
-- Installing: /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/lib_yujinrobot_yrldriver/include/yujinrobot_yrldriver/yujinrobot_yrldriver.hpp
-- Installing: /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/lib_yujinrobot_yrldriver/lib/libyujinrobot_yrldriver.so
-- Set runtime path of "/home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/lib_yujinrobot_yrldriver/lib/libyujinrobot_yrldriver.so" to ""
-- Installing: /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/lib_yujinrobot_yrldriver/bin/yujinrobot_yrldriver/test_yrl_library
-- Set runtime path of "/home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/lib_yujinrobot_yrldriver/bin/yujinrobot_yrldriver/test_yrl_library" to ""
-- Installing: /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/lib_yujinrobot_yrldriver/bin/yujinrobot_yrldriver/test_distance_reader
-- Set runtime path of "/home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/lib_yujinrobot_yrldriver/bin/yujinrobot_yrldriver/test_distance_reader" to ""
-- Installing: /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/lib_yujinrobot_yrldriver/bin/yujinrobot_yrldriver/test_fw_interface
-- Set runtime path of "/home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/lib_yujinrobot_yrldriver/bin/yujinrobot_yrldriver/test_fw_interface" to ""
-- Installing: /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/lib_yujinrobot_yrldriver/bin/yujinrobot_yrldriver/test_error_code
-- Set runtime path of "/home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/lib_yujinrobot_yrldriver/bin/yujinrobot_yrldriver/test_error_code" to ""
-- Installing: /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/lib_yujinrobot_yrldriver/bin/yujinrobot_yrldriver/test_multilidar_operation
-- Set runtime path of "/home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/lib_yujinrobot_yrldriver/bin/yujinrobot_yrldriver/test_multilidar_operation" to ""
-- Installing: /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/lib_yujinrobot_yrldriver/bin/yujinrobot_yrldriver/test_fw_set_get
-- Set runtime path of "/home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/lib_yujinrobot_yrldriver/bin/yujinrobot_yrldriver/test_fw_set_get" to ""
-- Installing: /home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/lib_yujinrobot_yrldriver/bin/yujinrobot_yrldriver/test_fw_calibration
tof-hjkim2@tof-hjkim2-c:~/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build$
```

6. Check if an additional file has been created in the “build” folder as shown below. The ”lib_yujinrobot_yrldriver” folder contains a test execution file and a driver library file.



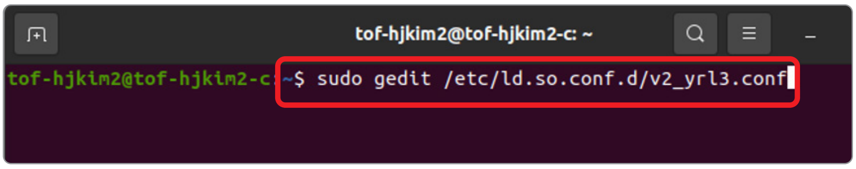
Name	Size
CMakeFiles	9 items
lib_yujinrobot_yrldriver	1 item
yujinrobot_yrldriver	5 items
CMakeCache.txt	14.5 kB
cmake_install.cmake	2.5 kB
install_manifest.txt	2.3 kB
Makefile	10.6 kB

3. Running the driver package

3.1 Setting up the library path

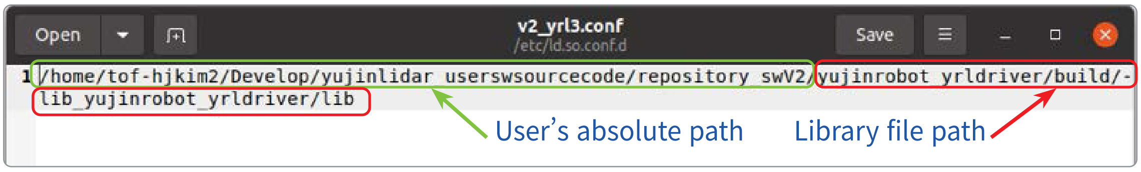
1. Launch the Terminal window and create a configuration file to set up the library path. Enter the command below.

- Command: ~$ sudo gedit /etc/ld.so.conf.d/**[File name]**.conf
    - **[File name]**: File name set by the user



```
tof-hjkim2@tof-hjkim2-c:~$ sudo gedit /etc/ld.so.conf.d/v2_yrl3.conf
```

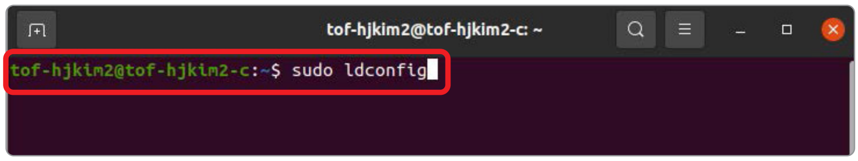
2. In the file created, enter the **[User’s absolute path]** and **[Library file path]** as shown below.
    - **[User’s absolute path]**: File path set by the user
    - **[Library file path]**: yujinrobot_yrldriver/build/lib_yujinrobot_yrldriver/lib



```
/home/tof-hjkim2/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/-
lib_yujinrobot_yrldriver/lib
```

3. Configure the shared library cache again. Enter the command below.

- Command: sudo ldconfig



```
tof-hjkim2@tof-hjkim2-c:~$ sudo ldconfig
```

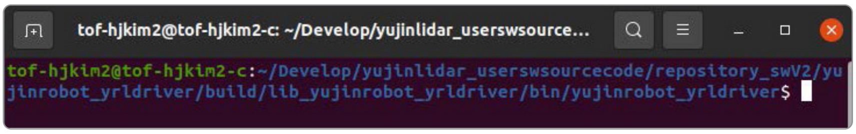
3.2 Running the test code

- Provided test code:
  - test_yrl_library
  - test_IP_change
  - test_mode_change
  - test_recover_network

- test_yrl_library run command: [executable file name] [IP address]
  - example) ./test_yrl_library 192.168.1.250
  - test_IP_change run command: [executable file name] [current IP address] [new IP address]
  - example) ./test_IP_change 192.168.1.250 192.168.11
  - test_mode_change run command: [executable file name] [IP address] [mode]
  - example) ./test_mode_change 192.168.1.250 1
  - test_recover_network run command: [executable file name] [IP address]
  - example) ./test_recover_network 192.168.1.250

1. Go to the folder that contains the "test code execution file".

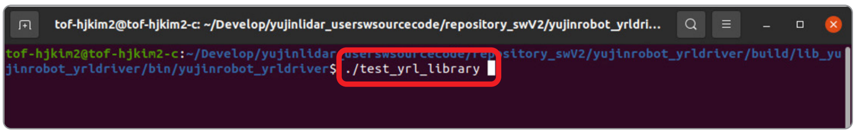
- yujinrobot_yrldriver/build/lib_yujinrobot_yrldriver/bin/yujinrobot_yrldriver


tof-hjkim2@tof-hjkim2-c: ~/Develop/yujinlidar_userswsource...

```
tof-hjkim2@tof-hjkim2-c:~/Develop/yujinlidar_userswsourcecode/repository_swV2/yu
jinrobot_yrldriver/build/lib_yujinrobot_yrldriver/bin/yujinrobot_yrldriver$ 
```

2. Launch the Terminal window and enter the command to run the test code.

- test_yrl_library


tof-hjkim2@tof-hjkim2-c: ~/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldri...

```
tof-hjkim2@tof-hjkim2-c:~/Develop/yujinlidar_userswsourcecode/repository_swV2/yujinrobot_yrldriver/build/lib_yu
jinrobot_yrldriver/bin/yujinrobot_yrldriver$ ./test_yrl_library 
```

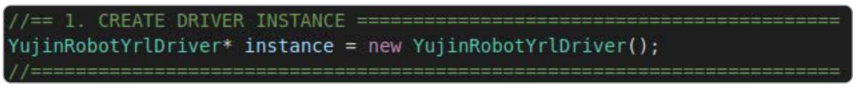
4. Using the test code

Yujin Robot provides a sample test code “test_yrl_library.cpp” for using the driver.

- File path: yujinrobot_yrldriver/yujinrobot_yrldriver/src/test/ test_yrl_library.cpp

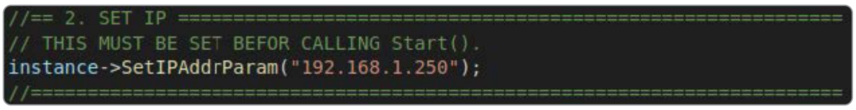
The main () in the “test_yrl_library.cpp” provides the function call order as an annotation.

1. Create the driver object.



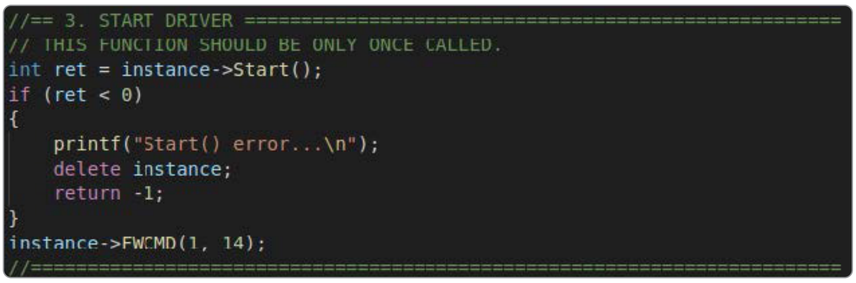
```
//== 1. CREATE DRIVER INSTANCE ==========================================
YujinRobotYrlDriver* instance = new YujinRobotYrlDriver();
//=====================================================================
```

2. Enter LiDAR’s IP address.



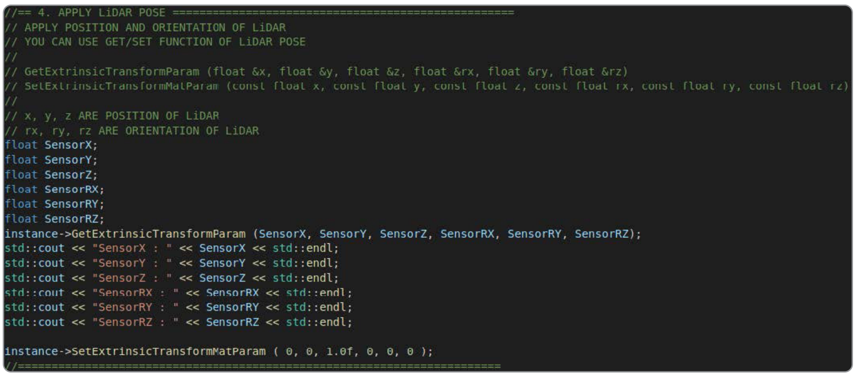
```
//== 2. SET IP ==========================================================
// THIS MUST BE SET BEFOR CALLING Start().
instance->SetIPAddrParam("192.168.1.250");
//=====================================================================
```

3. Establish TCP connection with LiDAR using the IP address.



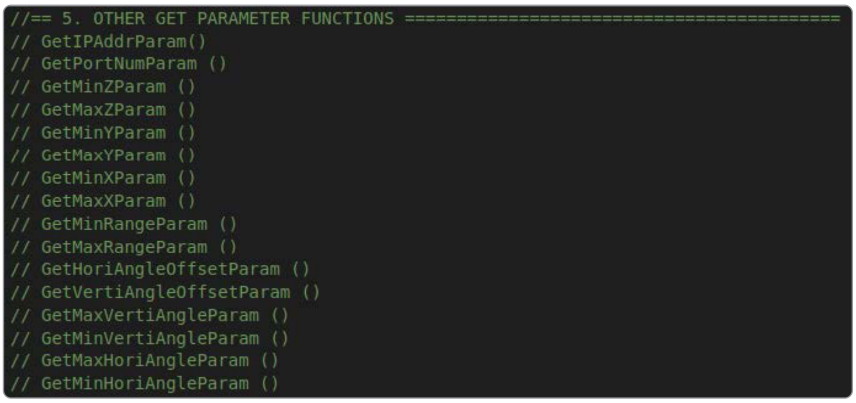
```
//== 3. START DRIVER ====================================================
// THIS FUNCTION SHOULD BE ONLY ONCE CALLED.
int ret = instance->Start();
if (ret < 0)
{
    printf("Start() error...\n");
    delete instance;
    return -1;
}
instance->FWCMD(1, 14);
//=====================================================================
```

4. You can read and modify the processing parameters for data input/output.

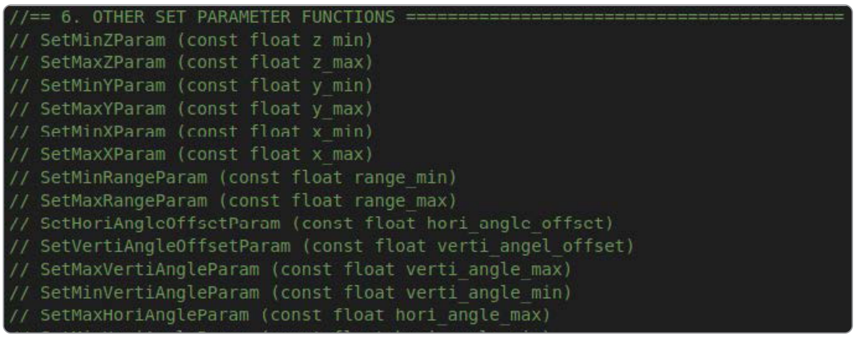


```
//== 4. APPLY LiDAR POSE ================================================
// APPLY POSITION AND ORIENTATION OF LiDAR
// YOU CAN USE GET/SET FUNCTION OF LiDAR POSE
//
// GetExtrinsicTransformParam (float &x, float &y, float &z, float &rx, float &ry, float &rz)
// SetExtrinsicTransformMatParam (const float x, const float y, const float z, const float rx, const float ry, const float rz)
//
// x, y, z ARE POSITION OF LiDAR
// rx, ry, rz ARE ORIENTATION OF LiDAR
float SensorX;
float SensorY;
float SensorZ;
float SensorRX;
float SensorRY;
float SensorRZ;
instance->GetExtrinsicTransformParam (SensorX, SensorY, SensorZ, SensorRX, SensorRY, SensorRZ);
std::cout << "SensorX : " << SensorX << std::endl;
std::cout << "SensorY : " << SensorY << std::endl;
std::cout << "SensorZ : " << SensorZ << std::endl;
std::cout << "SensorRX : " << SensorRX << std::endl;
std::cout << "SensorRY : " << SensorRY << std::endl;
std::cout << "SensorRZ : " << SensorRZ << std::endl;

instance->SetExtrinsicTransformMatParam ( 0, 0, 1.0f, 0, 0, 0 );
//=====================================================================
```



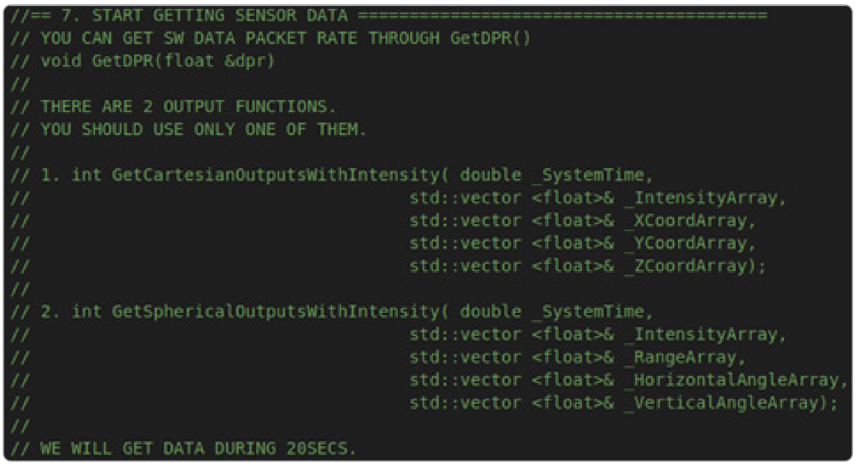
```
//== 5. OTHER GET PARAMETER FUNCTIONS ===================================
// GetIPAddrParam()
// GetPortNumParam ()
// GetMinZParam ()
// GetMaxZParam ()
// GetMinYParam ()
// GetMaxYParam ()
// GetMinXParam ()
// GetMaxXParam ()
// GetMinRangeParam ()
// GetMaxRangeParam ()
// GetHoriAngleOffsetParam ()
// GetVertiAngleOffsetParam ()
// GetMaxVertiAngleParam ()
// GetMinVertiAngleParam ()
// GetMaxHoriAngleParam ()
// GetMinHoriAngleParam ()
```



```
//== 6. OTHER SET PARAMETER FUNCTIONS ==========================================
// SetMinZParam (const float z_min)
// SetMaxZParam (const float z_max)
// SetMinYParam (const float y_min)
// SetMaxYParam (const float y_max)
// SetMinXParam (const float x_min)
// SetMaxXParam (const float x_max)
// SetMinRangeParam (const float range_min)
// SetMaxRangeParam (const float range_max)
// SetHoriAngleOffsetParam (const float hori_angle_offset)
// SetVertiAngleOffsetParam (const float verti_angel_offset)
// SetMaxVertiAngleParam (const float verti_angle_max)
// SetMinVertiAngleParam (const float verti_angle_min)
// SetMaxHoriAngleParam (const float hori_angle_max)
```

5. Functions that output sensor values are specified as annotations. One of the following two functions will be used:

- void GetCartesianOutputsWithIntensity(): Get the intensity and coordinates of the point cloud.
- void GetSphericalOutputsWithIntensity(): Get the intensity, range, horizontal angle, and vertical angle of the point cloud.

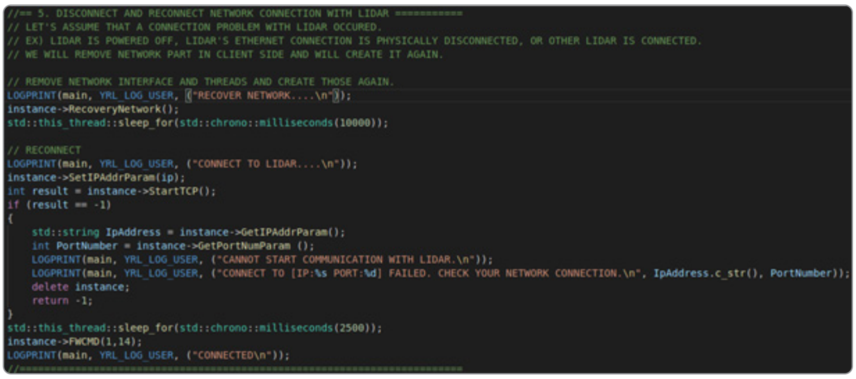


```
//== 7. START GETTING SENSOR DATA ========================================
// YOU CAN GET SW DATA PACKET RATE THROUGH GetDPR()
// void GetDPR(float &dpr)
//
// THERE ARE 2 OUTPUT FUNCTIONS.
// YOU SHOULD USE ONLY ONE OF THEM.
//
// 1. int GetCartesianOutputsWithIntensity( double _SystemTime,
//                                        std::vector <float>& _IntensityArray,
//                                        std::vector <float>& _XCoordArray,
//                                        std::vector <float>& _YCoordArray,
//                                        std::vector <float>& _ZCoordArray);
//
// 2. int GetSphericalOutputsWithIntensity( double _SystemTime,
//                                        std::vector <float>& _IntensityArray,
//                                        std::vector <float>& _RangeArray,
//                                        std::vector <float>& _HorizontalAngleArray,
//                                        std::vector <float>& _VerticalAngleArray);
//
// WE WILL GET DATA DURING 20SECS.
```

6. While running the code, the Network Recovery function is provided in the following cases:

- When LiDAR is turned off
- When LiDAR is disconnected from Ethernet
- When connecting another LiDAR

You can use the Recovery Network function of the sample test code “test_recover_network.cpp”.



```
//== 5. DISCONNECT AND RECONNECT NETWORK CONNECTION WITH LIDAR ==========
// LET'S ASSUME THAT A CONNECTION PROBLEM WITH LIDAR OCCURED.
// EX) LIDAR IS POWERED OFF, LIDAR'S ETHERNET CONNECTION IS PHYSICALLY DISCONNECTED, OR OTHER LIDAR IS CONNECTED.
// WE WILL REMOVE NETWORK PART IN CLIENT SIDE AND WILL CREATE IT AGAIN.

// REMOVE NETWORK INTERFACE AND THREADS AND CREATE THOSE AGAIN.
LOGPRINT(main, YRL_LOG_USER, ("RECOVER NETWORK....\n"));
instance->RecoveryNetwork();
std::this_thread::sleep_for(std::chrono::milliseconds(10000));

// RECONNECT
LOGPRINT(main, YRL_LOG_USER, ("CONNECT TO LIDAR....\n"));
instance->SetIPAddrParam(ip);
int result = instance->StartTCP();
if (result == -1)
{
    std::string IpAddress = instance->GetIPAddrParam();
    int PortNumber = instance->GetPortNumParam ();
    LOGPRINT(main, YRL_LOG_USER, ("CANNOT START COMMUNICATION WITH LIDAR.\n"));
    LOGPRINT(main, YRL_LOG_USER, ("CONNECT TO [IP:%s PORT:%d] FAILED. CHECK YOUR NETWORK CONNECTION.\n", IpAddress.c_str(), PortNumber));
    delete instance;
    return -1;
}
std::this_thread::sleep_for(std::chrono::milliseconds(2500));
instance->FWCMD(1,14);
LOGPRINT(main, YRL_LOG_USER, ("CONNECTED\n"));
//=================================================================
```

Yujin Robot Co., Ltd.
Harmony-ro 187 beon-gil 33, Yeonsu-gu, Incheon (Songdo-dong)
Phone +82 32 550 2300 Fax +82 32 550 2301

www.yujinrobot.com