



LIPSedge™ AE400

V0.9.0.7

Programming Guide

© Copyright LIPS Corporation 2020. All rights reserved.

Under the Intellectual Property Law, no part of this book may be copied in any form or used by any means without the written consent of LIPS Corporation. Violation of the said law results in consequences and those who failed to comply could be susceptible to penalties.

Although every effort has been made to ensure the accuracy of this manual, errors and inconsistencies may remain. The manufacturer assumes no liability resulting from errors or omission in this manual, even if damages arise from the use of the information. All contents are subject to constant revision to improve its reliability and may be changed without prior notice.

June 2020

Table of Contents

LIPSedge™ AE400 Programming Guide	5
Regulatory Compliance	6
FCC Compliance	6
FCC Label Notice	7
CE Compliance	8
RoHS Compliance	8
Notes for Programmers	9
I. Support for native Intel® RealSense™ SDK	9
II. Support for Linux Kernel	9
III. Gain Parameter Issues	9
IV. Configuring Multiple LIPSedge™ AE400	10
V. Intel® RealSense™ Viewer Automatic Update Issue	10
1. System Requirements	11
Windows	18
2. Prerequisite	20
2.1 Installing Microsoft Visual Studio Community 2015 / 2017	21
2.2 Installing Git Bash	24
2.3 Installing CMake	29
3. Building LIPSedge™ AE400 SDK from Source Code	33
3.1 Cloning LIPSedge™ AE400 SDK	33
3.2 Building LIPSedge™ AE400 SDK	34
3.3 Patching Intel® RealSense™ SDK	40
3.4 Programming with Wrappers	44

3.4.1	OpenNI2 Wrapper	44
3.4.1.1	Installing OpenNI2	44
3.4.1.2	Adjusting Network Configuration	47
3.4.1.3	Building OpenNI2 Wrapper.....	48
3.4.1.3	Starting Test Applications	52
3.4.2	OpenCV Wrapper	61
3.4.2.1	Installing OpenCV	61
3.4.2.2	Building OpenCV through Compiler	62
3.4.2.3	Starting Test Application	68
3.4.3	Python Wrapper.....	72
3.4.3.1	Installing Python.....	72
3.4.3.2	Configuring Environment Variables	74
3.4.3.3	Configuring Development Environment..	77
3.4.3.4	Building Python from Source Code	78
3.4.3.4	Starting Test Applications	81
4.	Programming with HALCON	87
4.1	LIPSedge™ AE400 SDK with GenICam Installation & Configuration	88
4.2	HALCON Registration & Installation	92
4.3	Accessing Camera Image with HALCON.....	103
4.4	Parameters Loading and Image Acquisition	108
Linux	62
5	Prerequisites.....	115
5.1	Installing APT Tool	116
5.2	Configuring Backend Environments.....	117
5.3	Installing Git Tool.....	119
5.4	Installing CMake.....	120
6	Building LIPSedge™ AE400 SDK from Source Code.....	121
8.1	Cloning LIPSedge™ AE400 SDK	121
8.2	Building LIPSedge™ AE400 SDK.....	123
8.3	Patching Intel® RealSense™ SDK.....	126

9. Programming with Wrapper	129
7.1 Python Wrapper	129
7.1.1 Prerequisites	130
7.1.2 Building Python from Source Code	135
7.1.3 Starting Test Applications	139
7.1.4 Editing Sample Codes	143
Appendix - System Tools Overview	145



LIPSedge™ AE400 Programming Guide

Welcome to **LIPSedge™ AE400 Programming Guide!** This document provides a step-by-step guide for configuring application developing environment on your PC / laptop for LIPSedge™ AE400 camera.

This document does **NOT** cover the basic installation / configuration process of LIPSedge™ AE400 camera. For the basic installation / configuration process of your camera, refer to *LIPSedge™ AE400 User's Guide*.

Regulatory Compliance



FCC Compliance

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Caution: Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.



FCC Label Notice



This device complies with part 15 of the FCC Rules. Operation is subject to the following conditions:

- (1) This device may not cause harmful interference, and
- (2) this device must accept any interference received, including interference that may cause undesired operation.



CE Compliance

This is a Class B product. In a domestic environment, this product may cause radio interference in which case the user may be required to take adequate measures.



RoHS Compliance

All lead-free products offered by the company comply with the requirements of the European law on the Restriction of Hazardous Substances (RoHS) directive, which means our manufacture processes and products are strictly “lead-free” and without the hazardous substances cited in the directive.

Notes for Programmers

I. Support for native Intel® RealSense™ SDK

LIPS Corp. provides a series of system tools with the LIPSedge™ AE400 SDK for software development. These system tools are modified based on Intel® RealSense™ System Tools. Modified Intel® RealSense™ System Tools can **NOT be used interchangeably** with native Intel® RealSense™ System Tools. We highly recommend using only modified Intel® RealSense™ System Tools provided for the software development process.

II. Support for Linux Kernel

LIPSedge™ AE400 SDK runs best on **Linux Kernel version 4**. Upgrading the Linux Kernel to **Version 5 or above** may cause the camera's framerate to decline by rare chances.

If the framerate decline issue occurs, downgrade your Linux Kernel to **Version 4**.

III. Gain Parameter Issues

Avoid dragging the Gain parameter in Intel® RealSense™ Viewer under Windows 10 at abrupt speed as it leads to a failure message occasionally.

IV. Configuring Multiple LIPSedge™ AE400

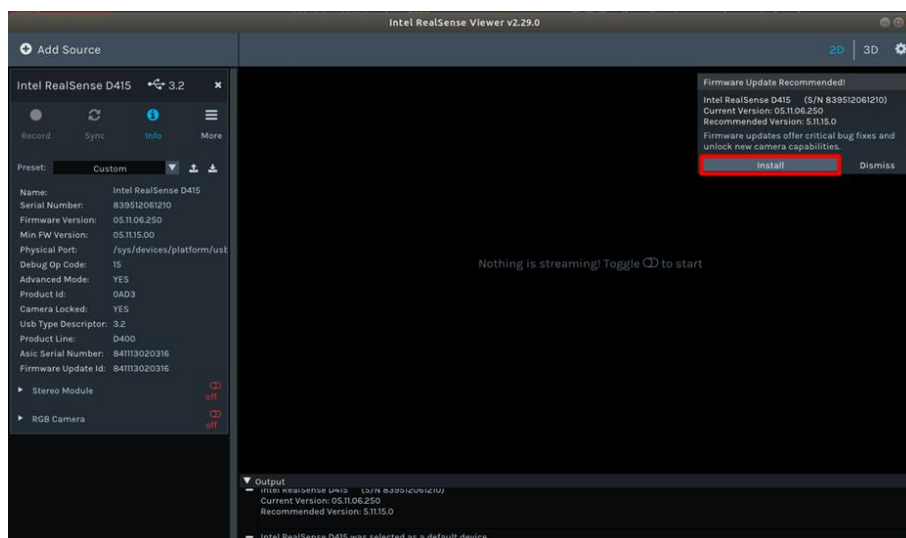
Multiple configurations of LIPSedge™ AE400 is **NOT** supported with **LIPSedge™ AE400 SDK V0.9.0.7** with **Firmware V1.4** as the **network.json** file, the file in which the camera's IP address is saved, only saves **1 IP address** at a time.

Developers may design a sub-application for capturing multiple LIPSedge™ AE400 configuration and concatenate the data captured by the sub-application. If you do so, each of the cameras must be assigned to a separate sub-network, with a separate physical Ethernet port assigned to each camera to ensure the data transmission efficiency.

V. Intel® RealSense™ Viewer Automatic Update Issue

When running **LIPSedge™ AE400 SDK V0.9.0.7** with Intel® RealSense™ SDK V2.17 or above, an update message prompts you to update the Intel® RealSense™ SDK to later versions. However, Intel® RealSense™ SDK compatibility are not fully tested from version to version on the LIPSedge™ AE400 camera. Currently, clicking the Install button does **NOT** lead to an upgrade, and may cause damages to your camera.

As the result, Do **NOT** click install on the following message.



1. System Requirements

We recommend using the following operating systems when accessing LIPSedge™ AE400:

Recommended Hardware

- Gigabit Ethernet Interface
- USB 3.0 Interface

Recommended Systems

- Windows 10 64-bit System
- Ubuntu 16.04 LTS System
- Internet Explorer 6.0 (or equivalent) or above

Windows

2. Prerequisite

LIPS Corp. provides the source code of the LIPSedge™ AE400 SDK on <https://github.com/lips-hci/ae400-realsense-sdk>. We recommend the following application / compiler for building LIPSedge™ AE400 SDK.

- Microsoft Visual Studio Community 2015 / 2017 or above
- Git Bash 2.25.0 or above
- CMake V3.8 or above

2.1 Installing Microsoft Visual Studio

Community 2015 / 2017

Microsoft Visual Studio Community 2015 / 2017 or above is the main compiler for LIPSegde™ AE400 SDK.

1. Go to <https://visualstudio.microsoft.com/vs/older-downloads/>, select **2017** and click **Download**.

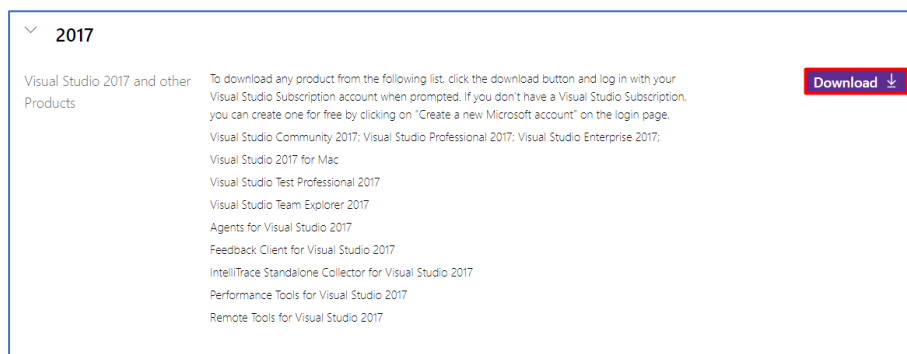


Figure 2-1

2. Log onto <https://my.visualstudio.com/Downloads>. On the **Visual Studio Community 2017 (version 15.0)** option, select your preferred system language and click **Download**.

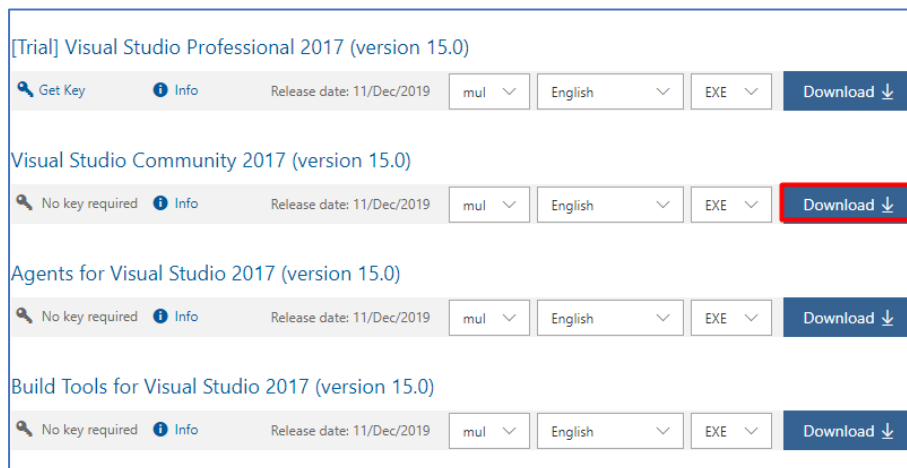


Figure 2-2

3. Start the downloaded file and click **Continue**.

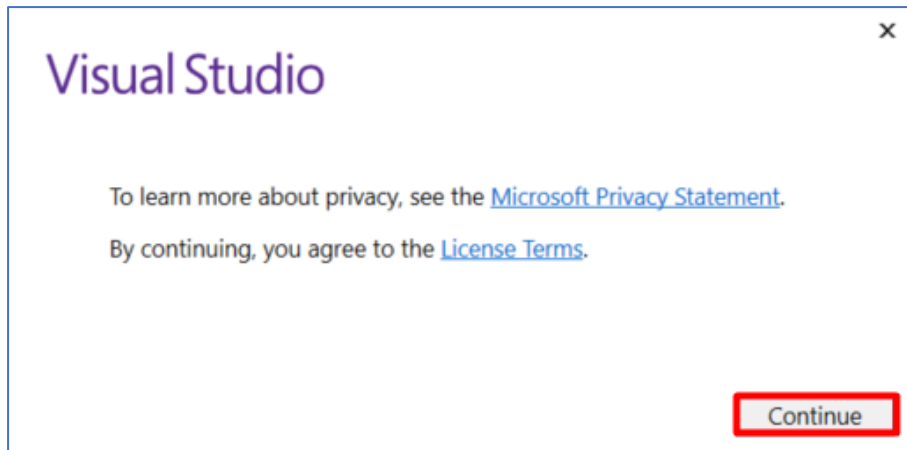


Figure 2-3

4. Click **Desktop development with C++** and select the following options:

- **VC++ 2017 v141 toolset (x86, x64)**
- **Visual C++ Tools for CMake**
- **Visual C++ ATL support**
- **C++ Profiling tools**
- **Windows 10 SDK (10.0.14393.0)**
- **Windows 10 SDK (10.0.10586)**

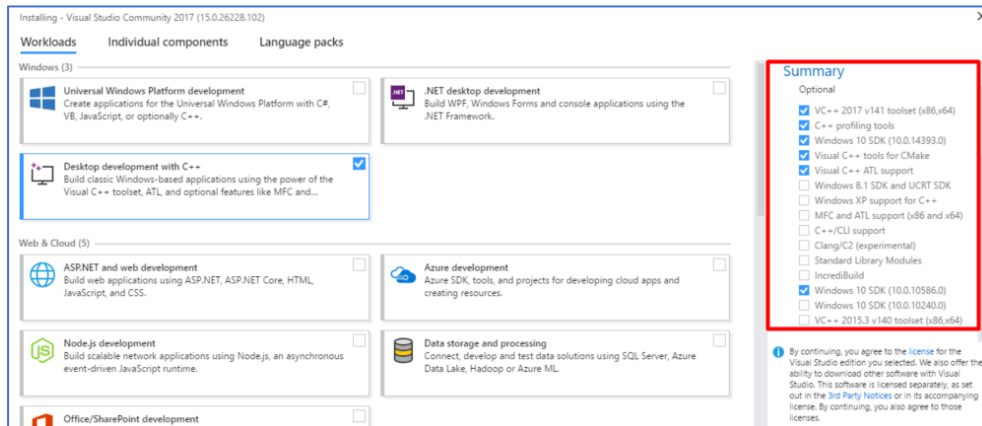


Figure 2-4

5. Click Install.

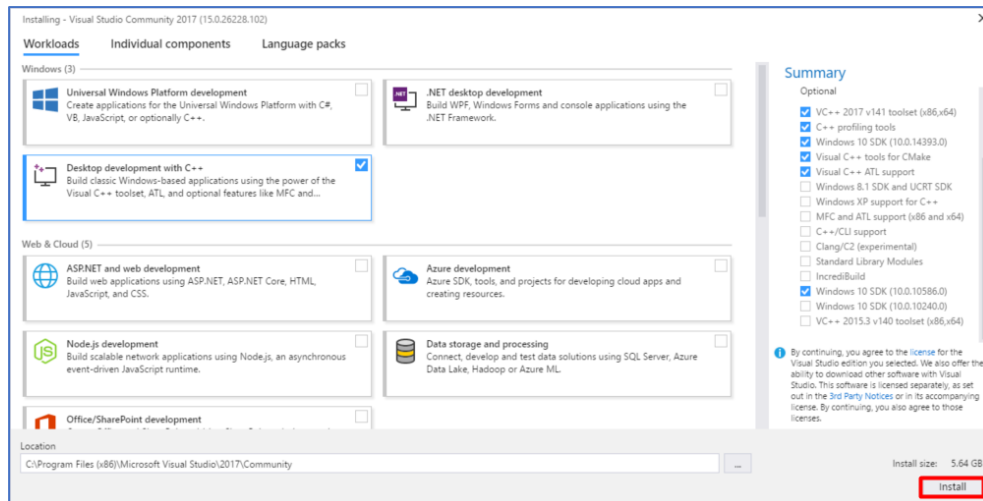


Figure 2-5

2.2 Installing Git Bash

With **Git Bash installed**, you have the convenience of configuring the backend environment for LIPSegde™ AE400 SDK through **command lines** with ease.

1. Go to <https://git-scm.com/downloads> and click **Download the latest source release for Windows**.

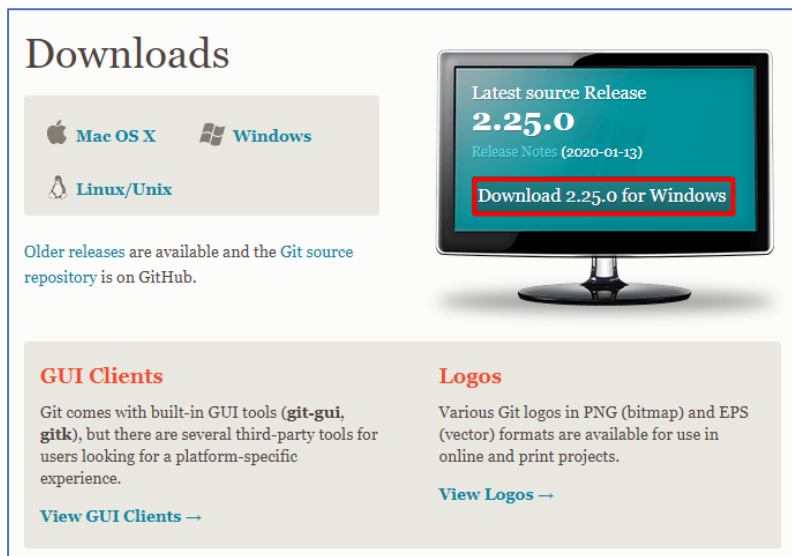


Figure 2-6

2. Start the downloaded file and click **Next**.



Figure 2-7

3. Click **Next**.

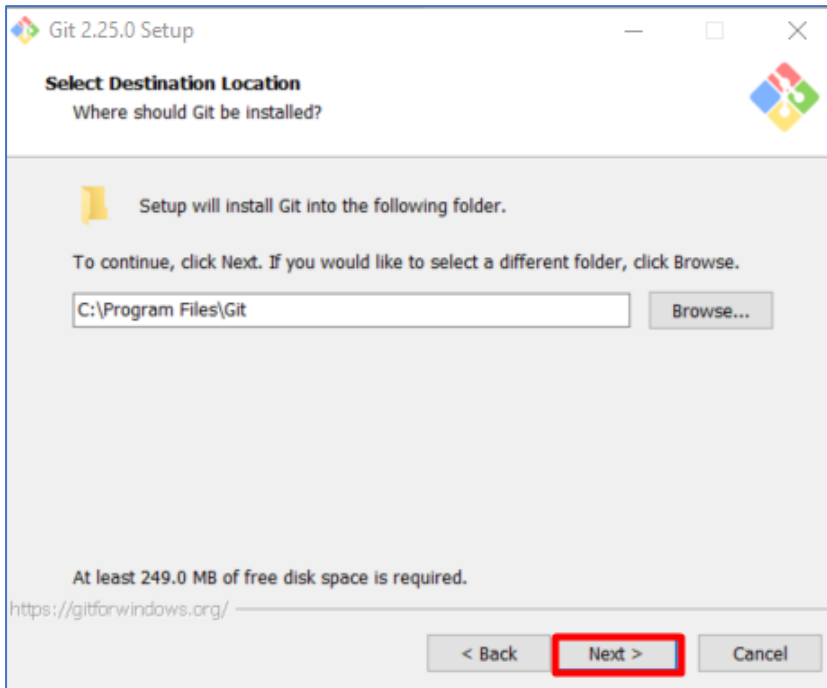


Figure 2-7

4. Click **Next**.

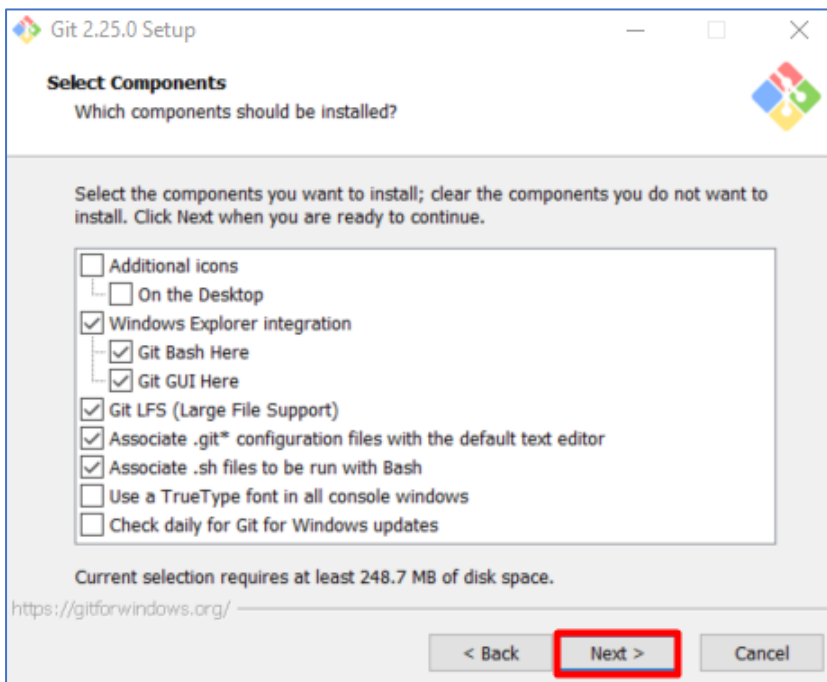


Figure 2-8

5. Click **Next**.

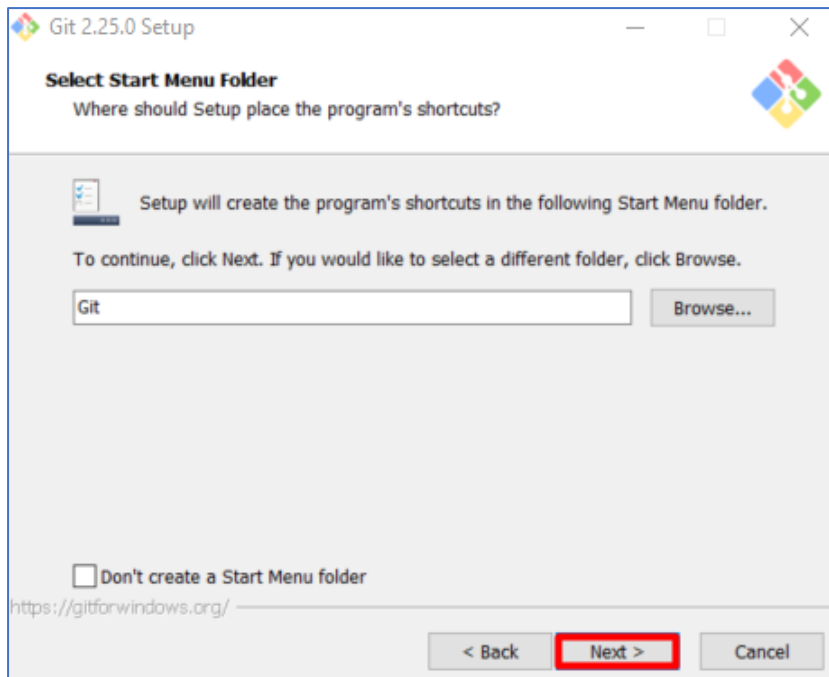


Figure 2-9

6. Click **Next**.

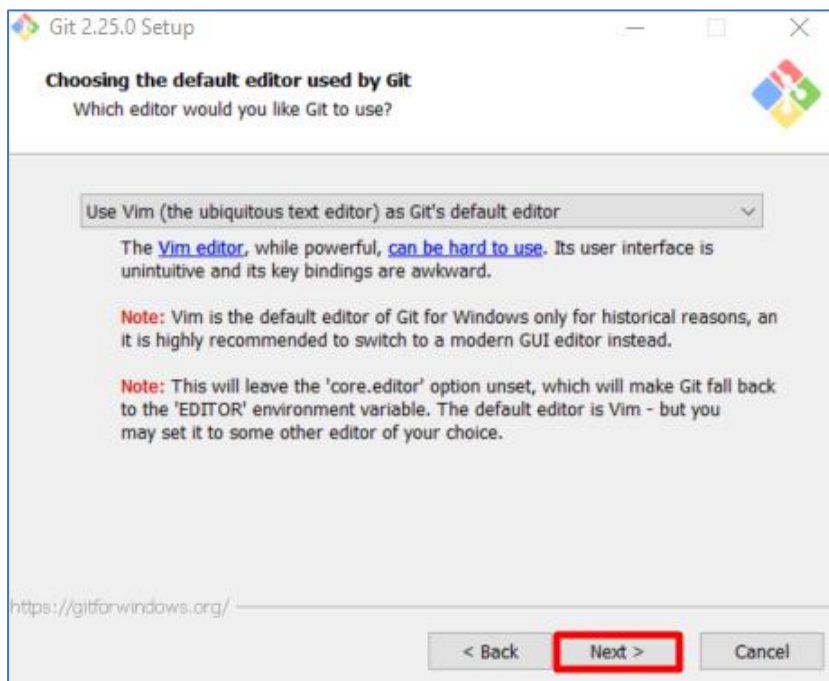


Figure 2-10

7. Click **Next**.

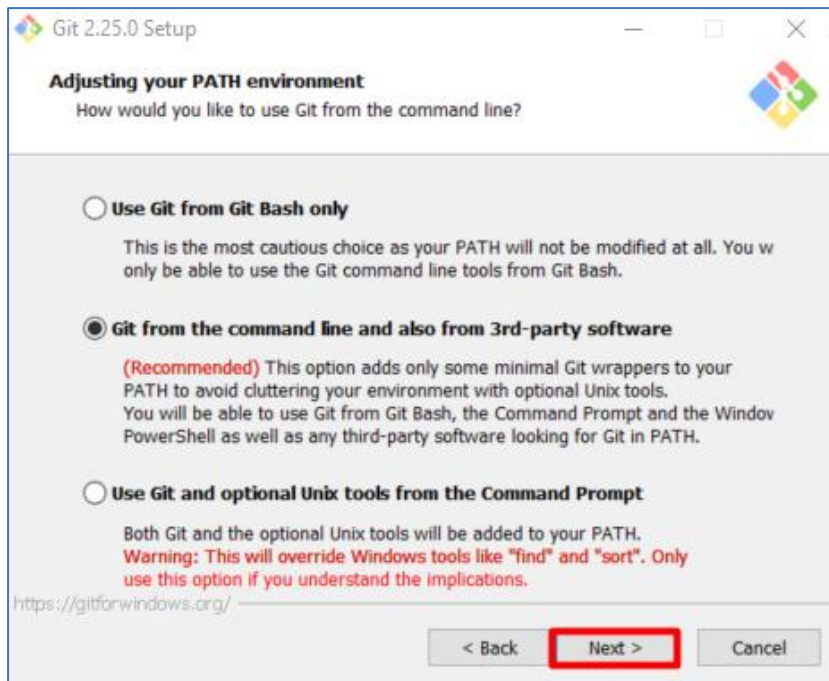


Figure 2-11

8. Click **Next**.

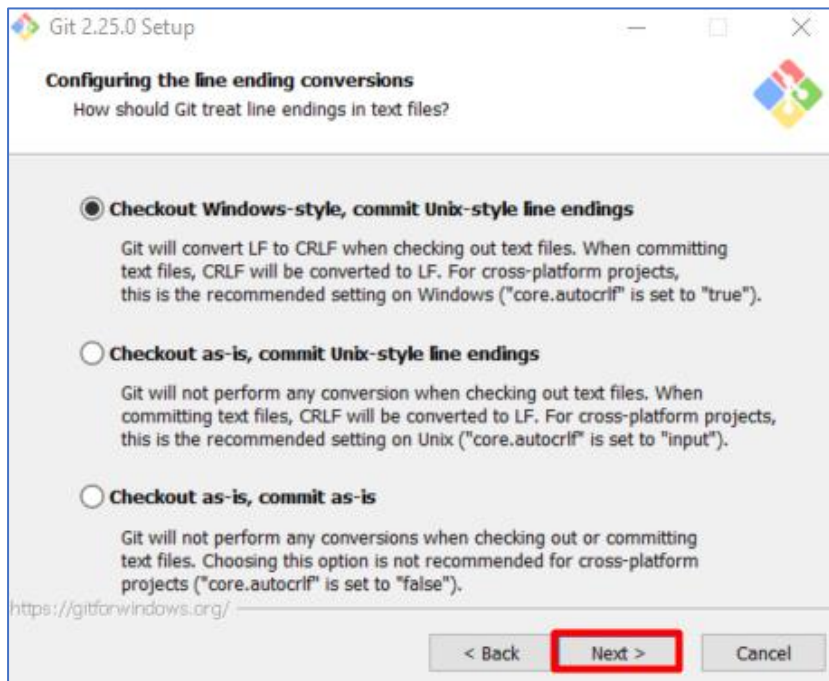


Figure 2-12

9. Click **Next**.

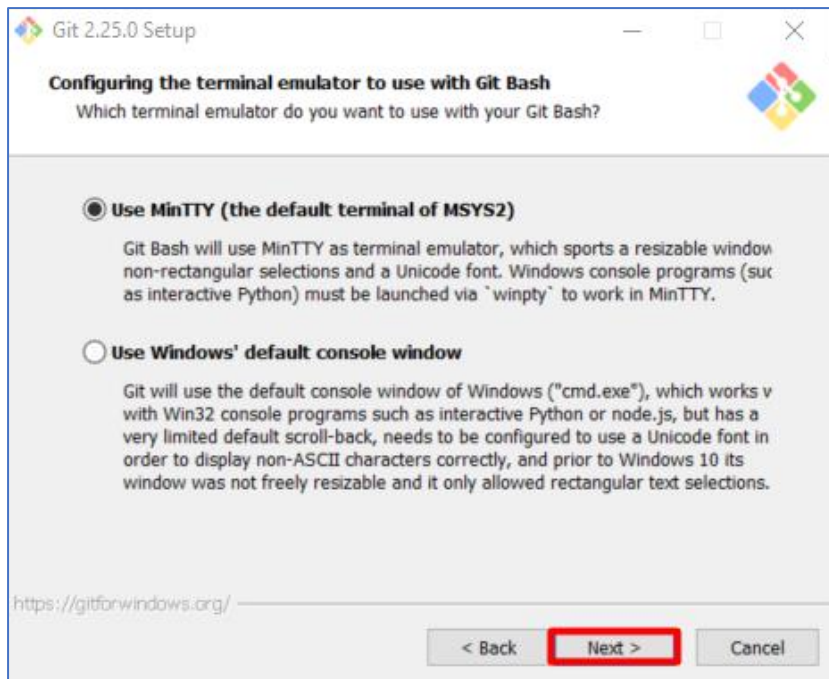


Figure 2-13

10. Click **Install**.

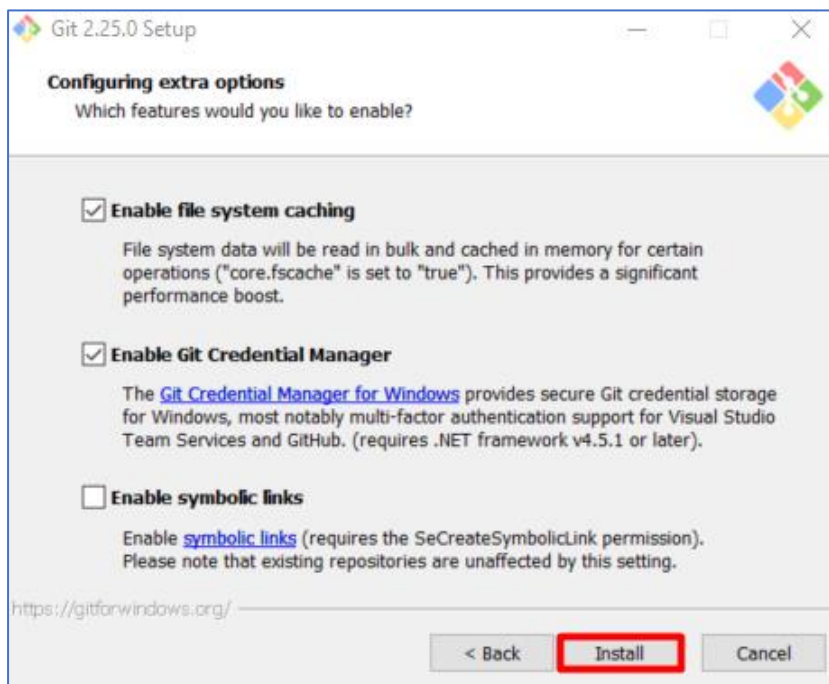


Figure 2-14

2.3 Installing CMake

LIPSedge™ AE400 SDK requires **CMake** to generate **Makefiles** necessary for building the application. We recommend using **CMake V3.8 or above** for the build.

1. Go to <https://cmake.org/download/> and click **Windows Installer**.

Platform	Files
Windows win64-x64 Installer: Installer tool has changed. Uninstall CMake 3.4 or lower first!	cmake-3.16.4-win64-x64.msi
Windows win64-x64 ZIP	cmake-3.16.4-win64-x64.zip
Windows win32-x86 Installer: Installer tool has changed. Uninstall CMake 3.4 or lower first!	cmake-3.16.4-win32-x86.msi
Windows win32-x86 ZIP	cmake-3.16.4-win32-x86.zip
Mac OS X 10.7 or later	cmake-3.16.4-Darwin-x86_64.dmg cmake-3.16.4-Darwin-x86_64.tar.gz
Linux x86_64	cmake-3.16.4-Linux-x86_64.sh cmake-3.16.4-Linux-x86_64.tar.gz

Figure 2-15

2. Start the downloaded file and click **Next**.

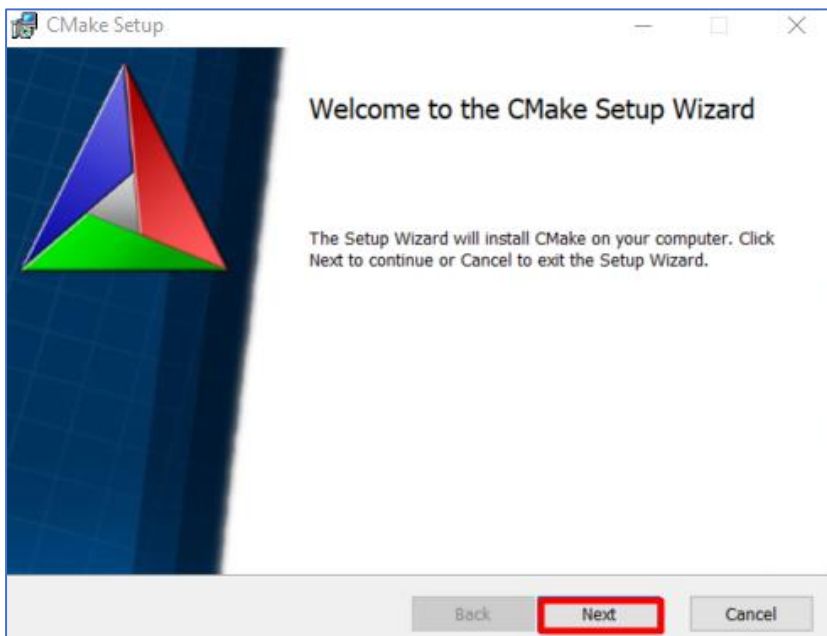


Figure 2-16

3. Agree to the user's agreement and click **Next**.



Figure 2-17

4. Click **Next**.

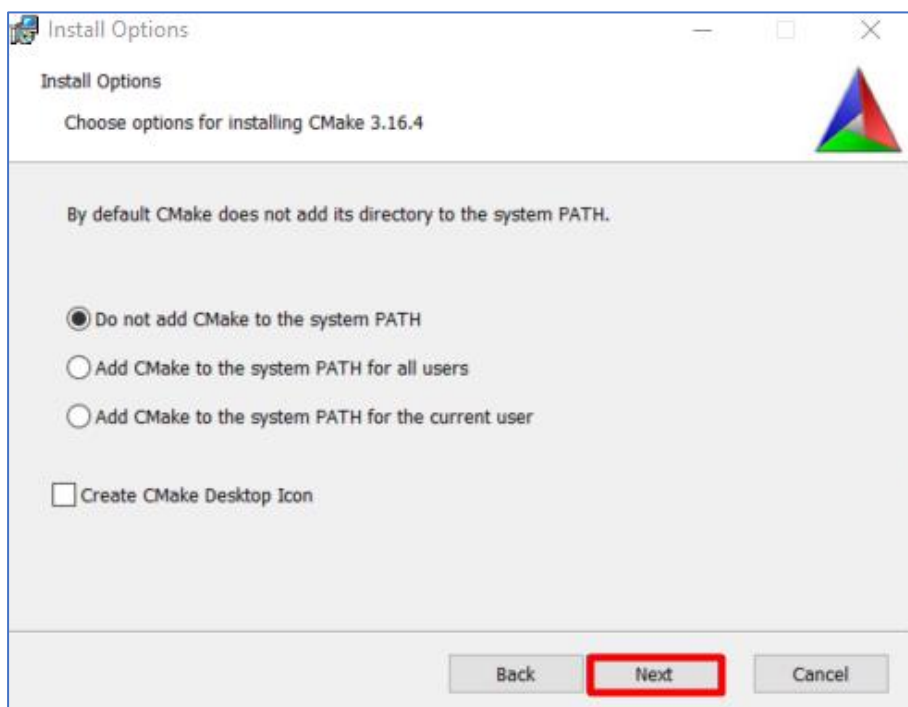


Figure 2-18

5. Click **Next**.

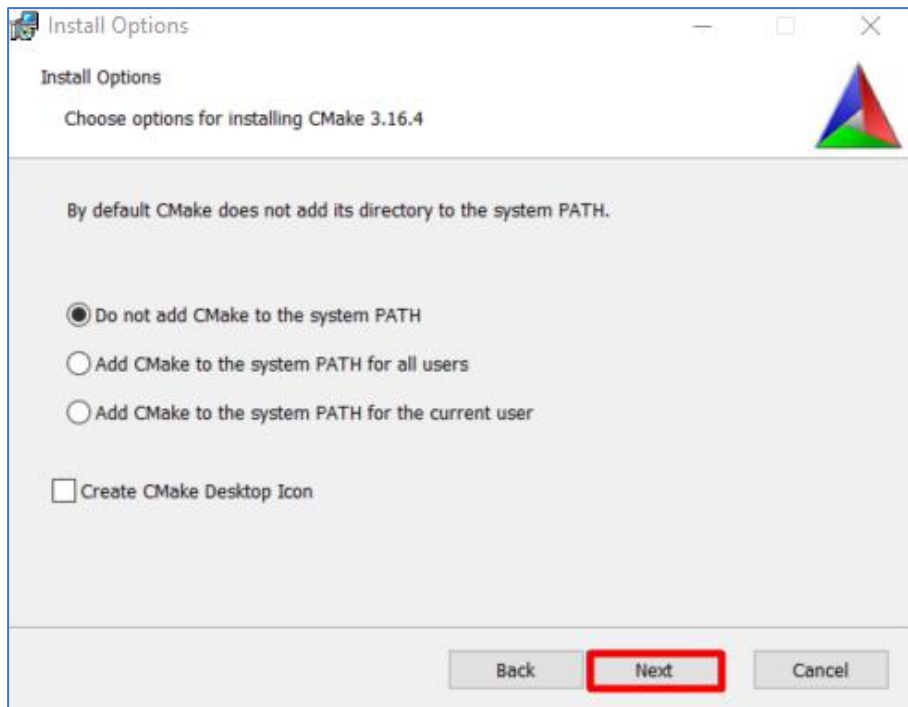


Figure 2-19

6. Click **Next**.

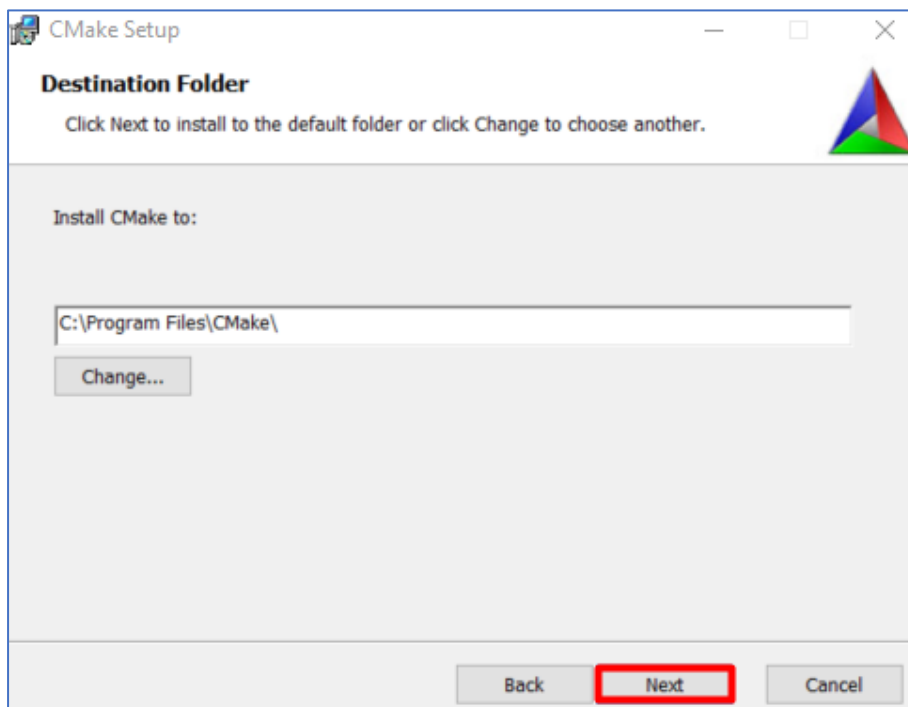


Figure 2-20

7. Click **Install**.

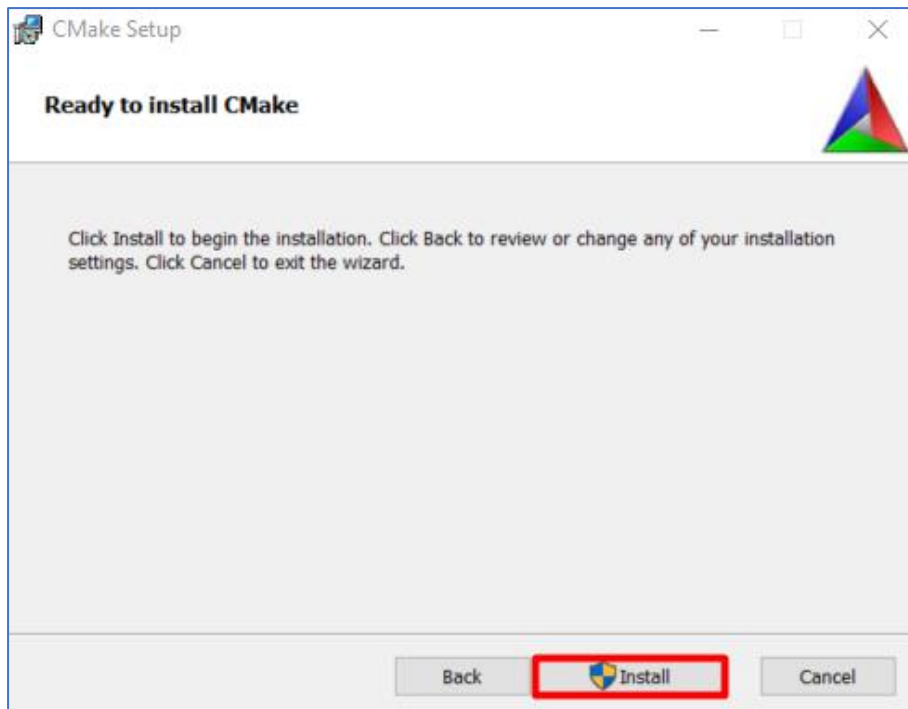


Figure 2-21

8. Click **Finish**.

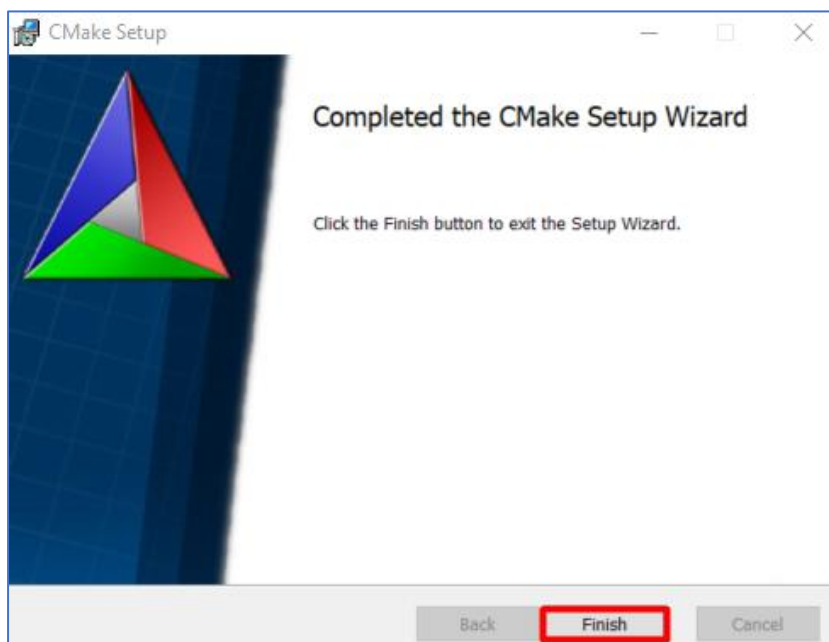


Figure 2-22

3. Building LIPSedge™ AE400 SDK from Source Code

LIPS Corp. provides the source code package on <https://github.com/lips-hci/ae400-realsense-sdk> based on Intel® RealSense™ SDK. The SDK is comprehensively tested to be compatible with Intel® RealSense™ SDK v2.17. We recommend utilizing Intel® RealSense™ SDK v2.17 for application development. Once the required application / compilers are ready, follow the instructions to clone and build the LIPSedge™ AE400 SDK through command lines.

3.1 Cloning LIPSedge™ AE400 SDK

1. Start **Git Bash** and clone the **LIPSedge™ AE400 SDK** to your PC / laptop.

```
git clone https://github.com/lips-hci/ae400-realsense-sdk.git ~/your destination
```

```
000200@200N378 MINGW32 /d
$ git clone https://github.com/lips-hci/ae400-realsense-sdk.git
Cloning into 'ae400-realsense-sdk'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 2531 (delta 3), reused 0 (delta 0), pack-reused 2522
Receiving objects: 100% (2531/2531), 50.70 MiB | 473.00 KiB/s, done.
Resolving deltas: 100% (605/605), done.
Updating files: 100% (2120/2120), done.
```

Figure 3-1

2. The cloning process is completed.

```
000200@200N378 MINGW32 /d
$ git clone https://github.com/lips-hci/ae400-realsense-sdk.git
Cloning into 'ae400-realsense-sdk'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 2531 (delta 3), reused 0 (delta 0), pack-reused 2522
Receiving objects: 100% (2531/2531), 50.70 MiB | 473.00 KiB/s, done.
Resolving deltas: 100% (605/605), done.
Updating files: 100% (2120/2120), done.
```

Figure 3-2

3.2 Building LIPSedge™ AE400 SDK

Once the LIPSedge™ AE400 SDK is cloned, configure the application development environment on Git Bash and build the application through the compiler, **Microsoft Visual Studio 2017**.

You can use the same method to build Intel® RealSense™ SDK, with a few details changed.

1. Go to **LIPSedge™ AE400 SDK source code folder**. In the case of Intel® RealSense™ SDK, go to the **Intel® RealSense™ SDK source code folder**.

```
cd LIPS SDK source code folder
```

```
000200@200N378 MINGW32 /d/ae400-realsense-sdk (master)
$ ls
AE400_WEB-BANNER.png  NOTICE
appveyor.yml          package.xml
AUTHORS               README.md
CMake/                screenshot-git-windows-run-patch-script.png
cmake_uninstall.cmake screenshot-rs-capture.png
CMakeLists.txt        screenshot-vs2017-project-install_network_config.png
common/               scripts/
config/                src/
CONTRIBUTING.md      third-party/
doc/                   tools/
examples/             unit-tests/
include/               wrappers/
LICENSE
```

Figure 3-3

2. In the LIPSedge™ AE400 SDK source code folder, create a folder to contain the LIPSedge™ AE400 SDK built. In the case of Intel® RealSense™ SDK, create the folders in the Intel® RealSense™ SDK source code folder.

```
mkdir build_Release  
cd build_Release
```

```
000200@200N378 MINGW32 /d/ae400-realsense-sdk (master)  
$ mkdir build_Release  
  
000200@200N378 MINGW32 /d/ae400-realsense-sdk (master)  
$ cd build_Release  
  
000200@200N378 MINGW32 /d/ae400-realsense-sdk/build_Release (master)
```

Figure 3-4

3. Start **CMake** and add the following location:

Field Name	Location
Where is the source code	Enter the location of the source code. E.g. D:\ae400-realsense-sdk D:\librealsense
Where to build the binaries	Enter the location in which the outcome applications are saved. E.g.: D:\ae400-realsense-sdk\build_Release D:\librealsense\build_Release

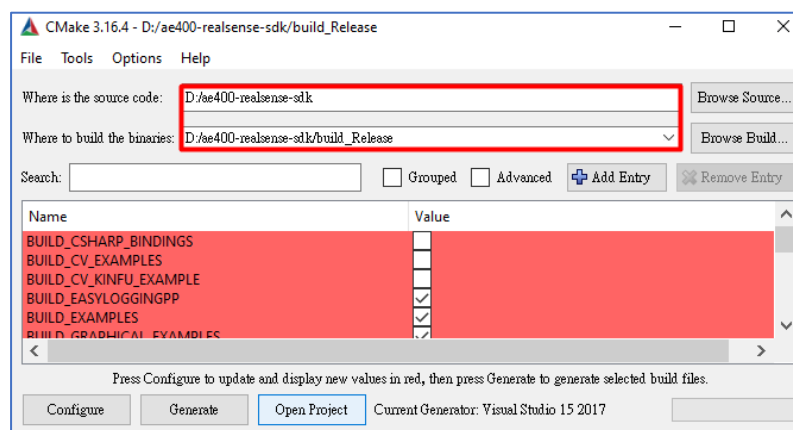


Figure 3-5

4. Click **Generate**.

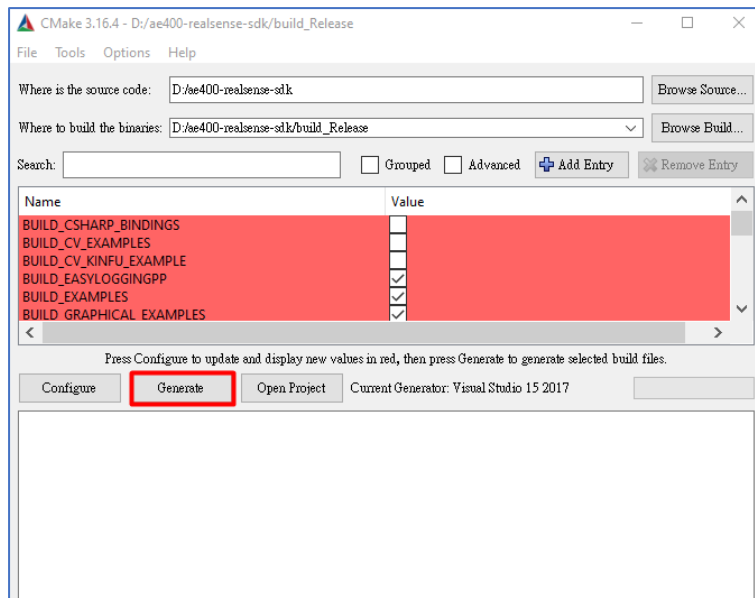


Figure 3-6

5. Specify the options for **generators**, **platforms**, and **compilers** necessary for this build and click **Finish**.

- **Specify the generator for this project:** Visual Studio 15 2017
- **Optional platform:** x64
- **Option for Compilers:** Use default native compilers

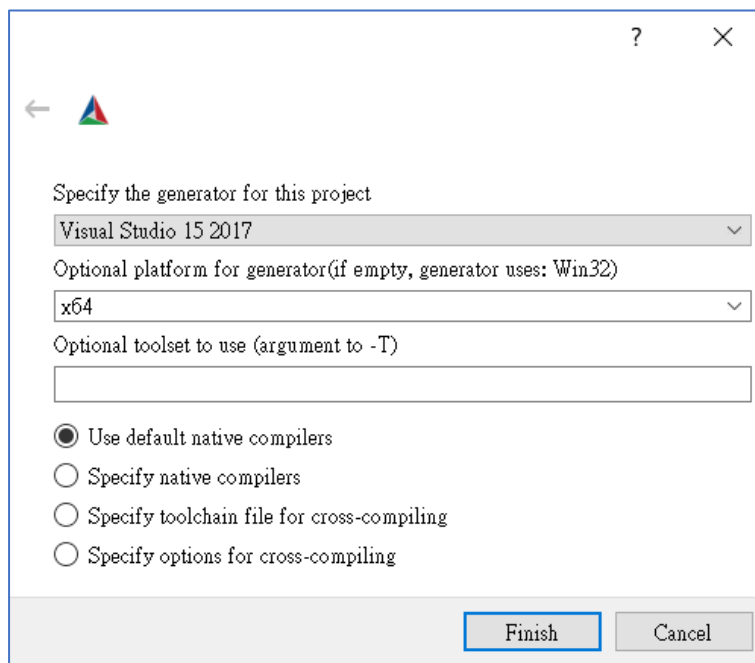


Figure 3-7

- Click **Generate**. If the Makefiles are not generated, make sure that the **Visual C++ Tools for CMake** are installed to your compiler and the **options** mentioned in *Step 5*. are selected. For details, refer to *2.1 Installing Microsoft Visual Studio Community*.

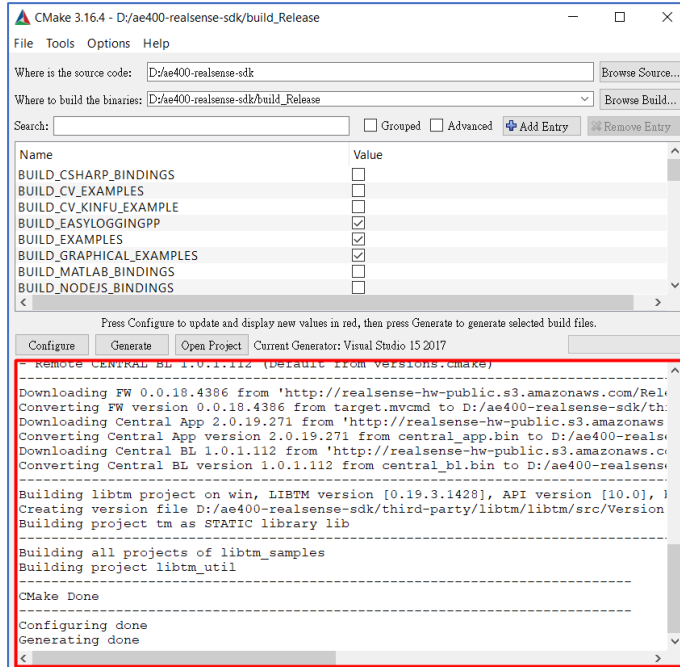


Figure 3-8

- Go to **LIPSEdge™ AE400 SDK > build_Release** and open **librealsense2.sln**. In the case of Intel® RealSense™ SDK, create the folders in the **Intel® RealSense™ SDK source code folder**, find the **.sln** file in **Intel® RealSense™ SDK > build_Release**.

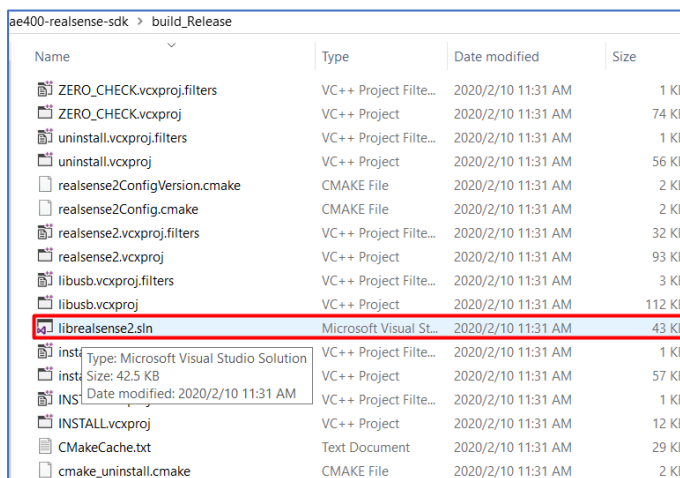


Figure 3-9

8. On the top menu, select **Build > Build Solutions**.

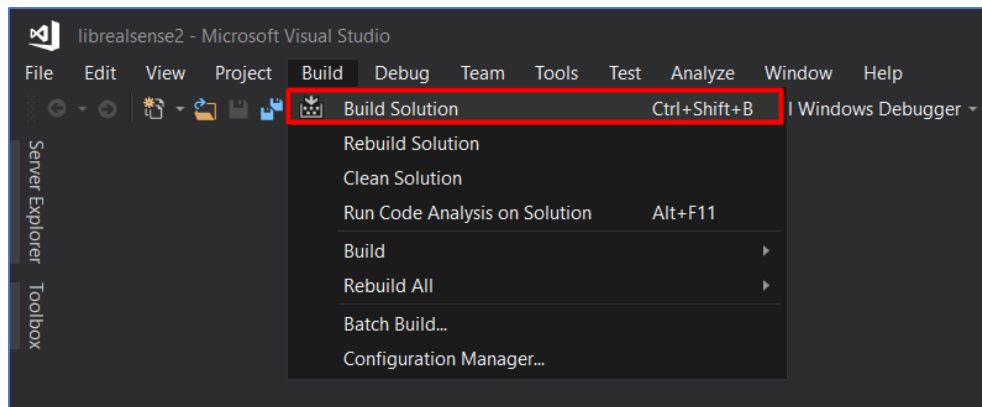


Figure 3-10

9. Under the **Output** window, confirm that the source codes are built.

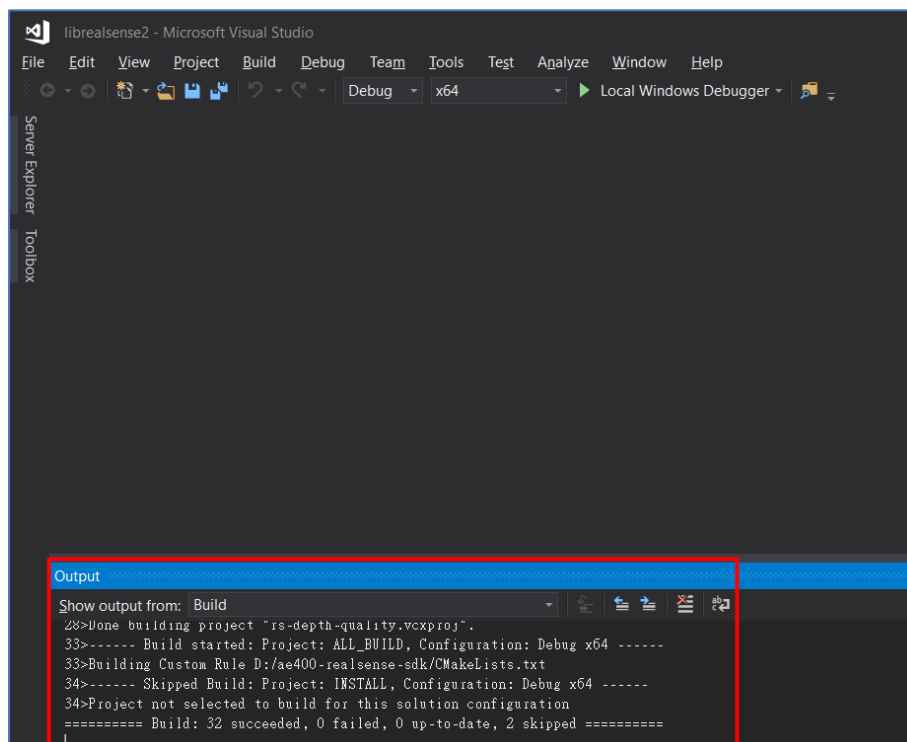


Figure 3-11

3.3 Patching Intel® RealSense™ SDK

By default, the LIPSedge™ AE400 camera connects to your applications through a Gigabit Ethernet interface, which is incompatible with native Intel® RealSense™ SDK applications.

LIPS Corp. provides a patch file to enable Ethernet connection with native Intel® RealSense™ SDK, thereby allowing developers to develop LIPSedge™ AE400-compatible applications on various Intel® RealSense™ SDK versions. However, Intel® RealSense™ SDK compatibility are not fully tested from version to version on the LIPSedge™ AE400 camera. Exercise the patches with caution. If error occurs, contact LIPS Corp. at info@lips-hci.com.

Follow the instructions below to apply patches for the Ethernet interface. Here, we use Intel® RealSense™ SDK with a **V2.29.0** Git branch as an example.

1. Start **Git Bash** and clone the **Intel® RealSense™ SDK** to your PC / laptop.

```
git clone https://github.com/IntelRealSense/librealsense.git ~/your  
destination
```

```
000200@200N378 MINGW64 /d  
$ git clone https://github.com/IntelRealSense/librealsense.git
```

Figure 3-12

2. Go to the **Intel® RealSense™ SDK** folder and select the Git branch for your local PC / laptop. Here, we use **v2.29.0** as an example.

```
git checkout -b rs2.29.0 v2.29.0
```

```
000200@200N378 MINGW64 /d/librealsense (master)
$ git checkout -b rs2.29.0 v2.29.0
```

Figure 3-13

Note: If your local Git branch is **v2.17.1**, error may occur in the subsequent building process as the USB library (libusb) is unstable in **v2.17.1**. To fix the libusb issue, commit the **89b8a02b** fix to the branch.

```
git checkout -b rs2.17.1 v2.17.1
```

```
git cherry-pick 89b8a02b
```

```
000200@200N378 MINGW64 /d/librealsense (master)
$ git checkout -b rs2.17.1 v2.17.1
Switched to a new branch 'rs2.17.1'

000200@200N378 MINGW64 /d/librealsense (rs2.17.1)
$ git cherry-pick 89b8a02b
[rs2.17.1 99b54c69c] Switch to use stable libusb version (#3049
Author: Sergey Dorodnicov <sergey.dorodnicov@intel.com>
Date: Sun Jan 13 18:37:53 2019 +0200
Committer: Alex Hon <000200@lips-hci.com>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accur
You can suppress this message by setting them explicitly. Run t
following command and follow the instructions in your editor to
your configuration file:

    git config --global --edit
```

3. Clone the **LIPSedge™ AE400 SDK** to your PC / laptop. For details, refer to 3.1 *Cloning LIPSedge™ AE400 SDK*.
4. Go to the **LIPSedge™ AE400 SDK** folder.

```
cd LIPSedge™ AE400 SDK
```

```
000200@200N378 MINGW64 /d/librealsense (master)
$ git checkout -b rs2.29.0 v2.29.0
Switched to a new branch 'rs2.29.0'

000200@200N378 MINGW64 /d/librealsense (rs2.29.0)
$ cd ..

000200@200N378 MINGW64 /d
$ cd ae400-realsense-sdk/
```

Figure 3-14

5. Execute the patch file.

```
./scripts/patch-realsense-to-lips-ae400-sdk.sh ~/Intel® RealSense™
SDK folder
```

```
000200@200N378 MINGW64 /d
$ cd ae400-realsense-sdk/

000200@200N378 MINGW64 /d/ae400-realsense-sdk (master)
$ ./scripts/patch-realsense-to-lips-ae400-sdk.sh ~/librealsense
AE400 SRC GIT = /d/ae400-realsense-sdk
RS2 SRC GIT = /c/Users/000200/librealsense
```

Figure 3-15



6. Go to the **Intel® RealSense™ SDK** folder and build the Intel® RealSense™ SDK. For details in building the Intel® RealSense™ SDK, refer to the similar steps in *3.2 Building LIPSedge™ AE400 SDK*.

3.4 Programming with Wrappers

LIPSedge™ AE400 is based on Intel® RealSense™ SDK, which supports various third-party technologies in wrappers of various programming languages, such as OpenNI2, OpenCV, or Python. When building the SDK, specify the **build configuration** to enable the wrapper you want and run the **CMake build** again to generate the binaries and libraries.

For detail on supported Intel® RealSense™ SDK wrappers, refer to <https://github.com/IntelRealSense/librealsense/tree/master/wrappers>.

3.4.1 OpenNI2 Wrapper

For developing python applications, refer to the following instructions to build up OpenCV Wrapper.

3.4.1.1 Installing OpenNI2

We recommend using **OpenNI2** for developing OpenNI2 applications for LIPSedge™ AE400.

1. Go to <https://structure.io/openni> or the QR code. Under **Windows**, download **OpenNI2 2.2.0.33 Beta (x86)**.

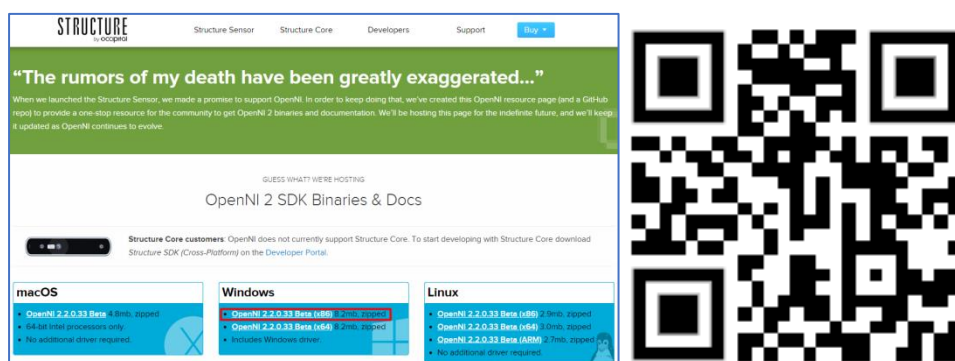


Figure 3-16

2. Extract the downloaded file and click **OpenNI-Windows-x86-2.2.msi**.

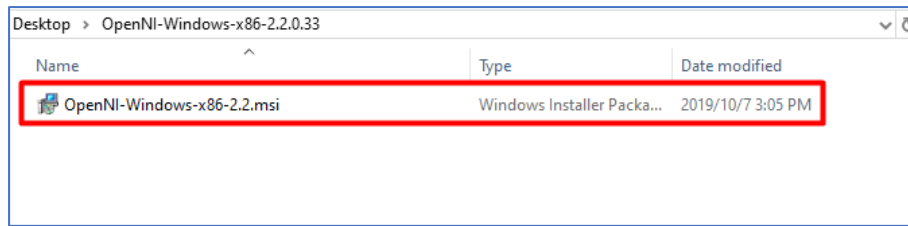


Figure 3-17

3. Click **Install**.

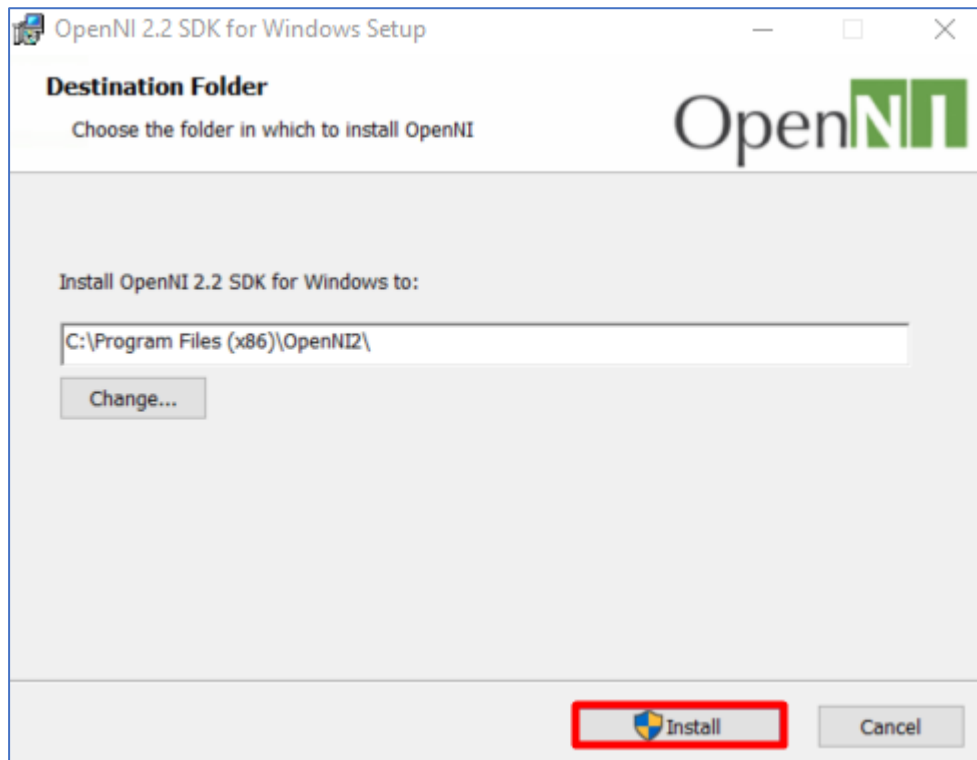


Figure 3-18

4. Click **Finish**.

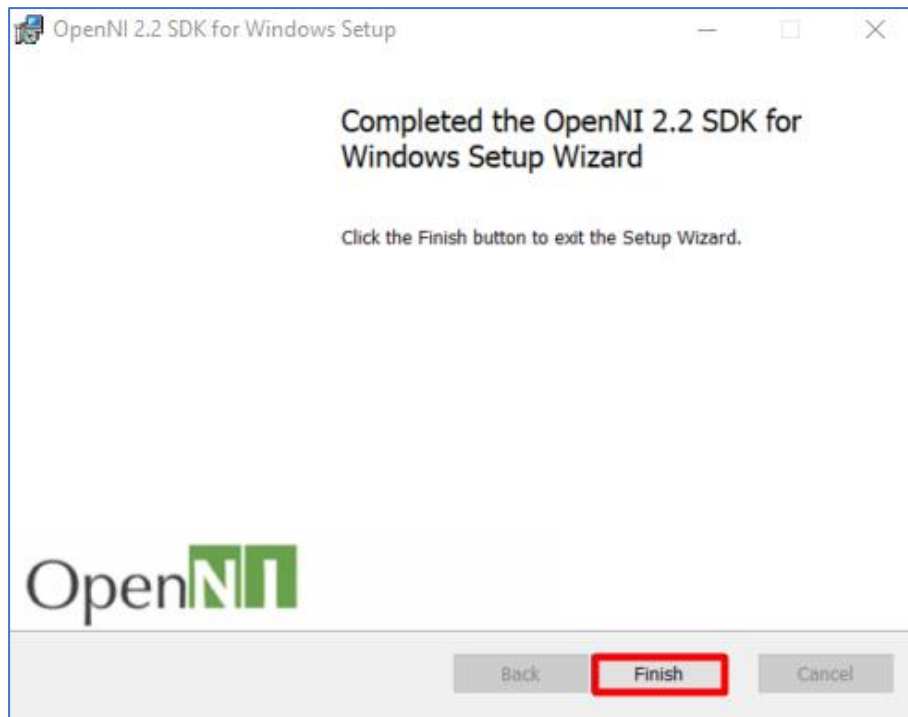


Figure 3-19

3.4.1.2 Adjusting Network Configuration

Before using OpenNI Wrapper, make sure the **network configuration** is properly adjusted.

1. On local host, go to **Program Files > OpenNI2 > Tools > OPENNI2 > Drivers** and open **network.json**.

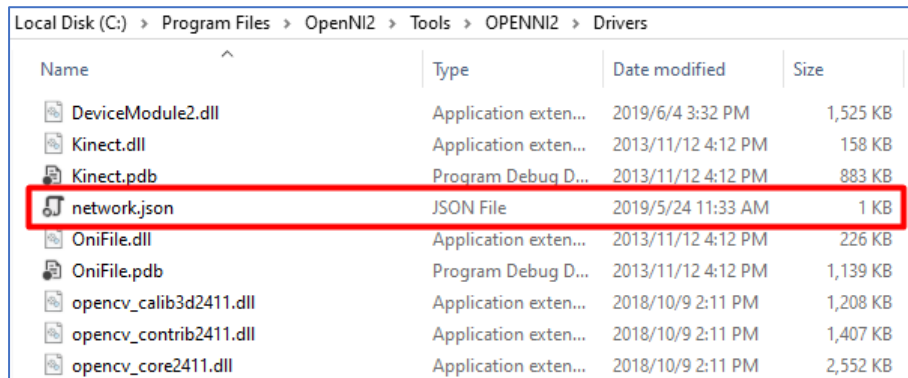


Figure 3-20

2. Type **your camera's IP address** as the value of "ip".

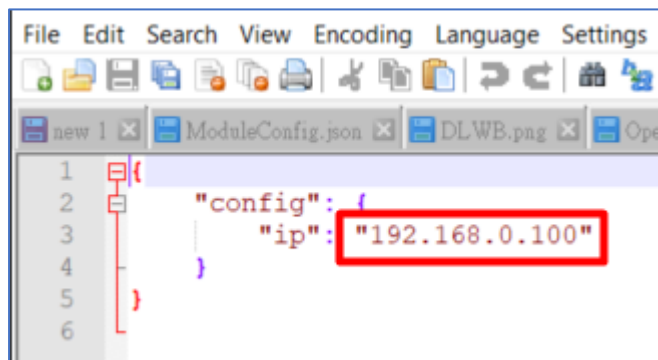


Figure 3-21

3.4.1.3 Building OpenNI2 Wrapper

Once OpenNI2 is installed and the network configurations are completed, build the libraries through the compiler of **Microsoft Visual Studio 2017**. Before the building process, make sure the prerequired items are installed.

For details on **relevant application / compiler**, refer to *2. Prerequisites*.
 For details on **OpenNI2 requirements**, refer to *3.4.1.1 Installing OpenNI2 and 3.4.1.2 Adjusting Network Configuration*.

1. Clone the **LIPSedge™ AE400 SDK source code package** to your **PC / laptop**. For details, refer to *3.1 Cloning LIPSedge™ AE400 SDK*.
2. Assign the **location on your PC / laptop** for the built outcome for **CMake**. For details, refer to *Step 1 – 3 in 3.2 Building LIPSedge™ AE400 SDK*.
3. On the **Entry List** below, select **BUILD_OPENNI2_BINDINGS**.

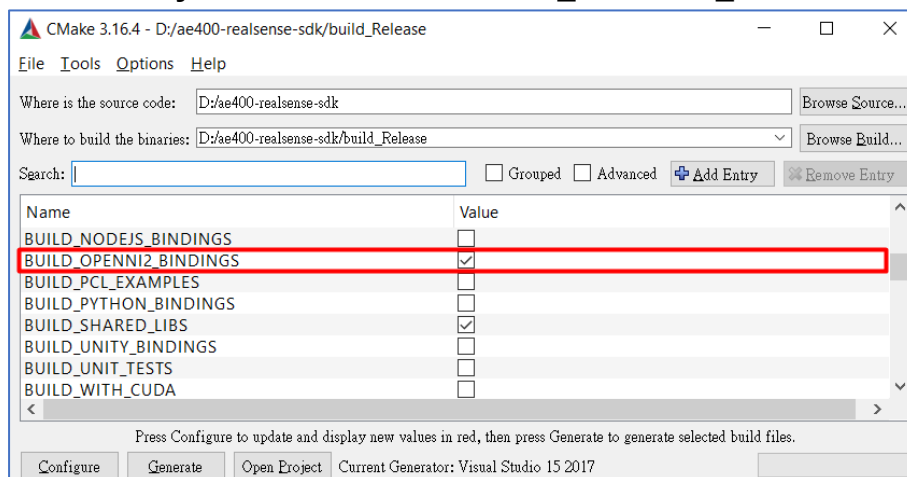


Figure 3-22

4. Click **Configure** and entries **OPENNI2_DIR** and **REALSENSE2_DIR** appear on the Entry List.

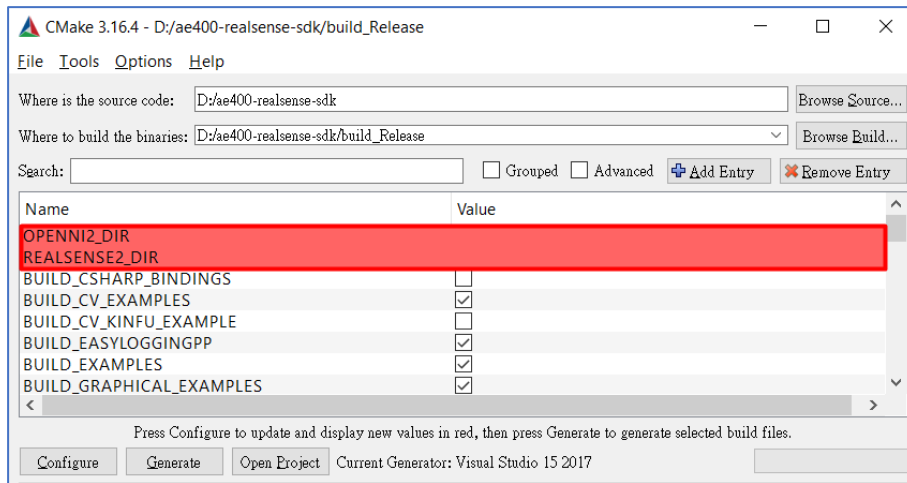


Figure 3-23

4. Assign the value to the following entries:
 - **OPENNI_DIR**: c:/Program Files/OpenNI2
 - **REALSENSE2_DIR**: c:/Program Files (x86)/Intel RealSense SDK 2.0

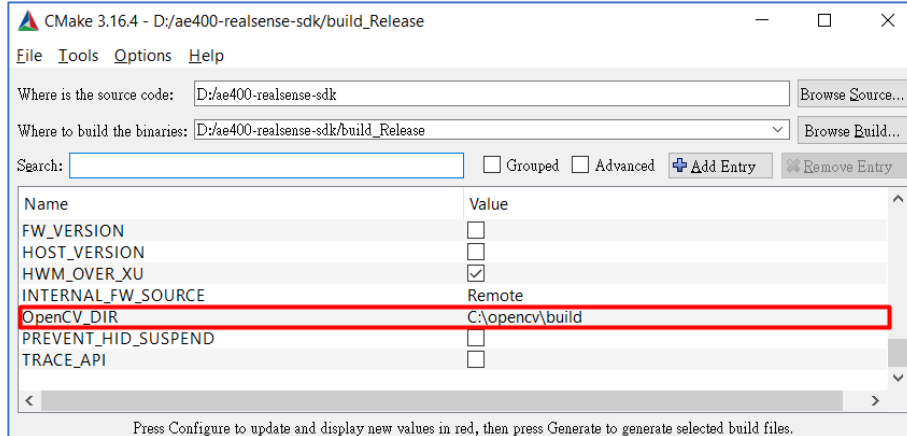


Figure 3-24

5. Click **Generate** and the Makefiles for OpenNI2 are generated.

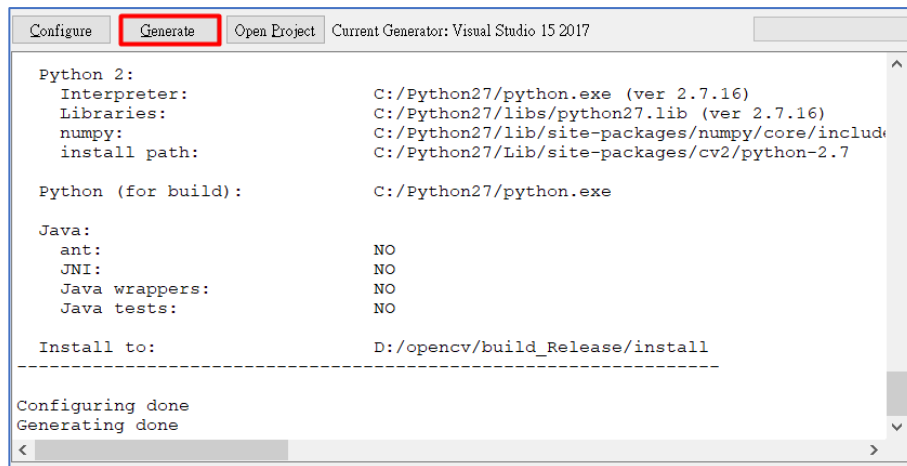


Figure 3-25

6. Click **Open Project** to launch **Microsoft Visual Studio Community 2017**.

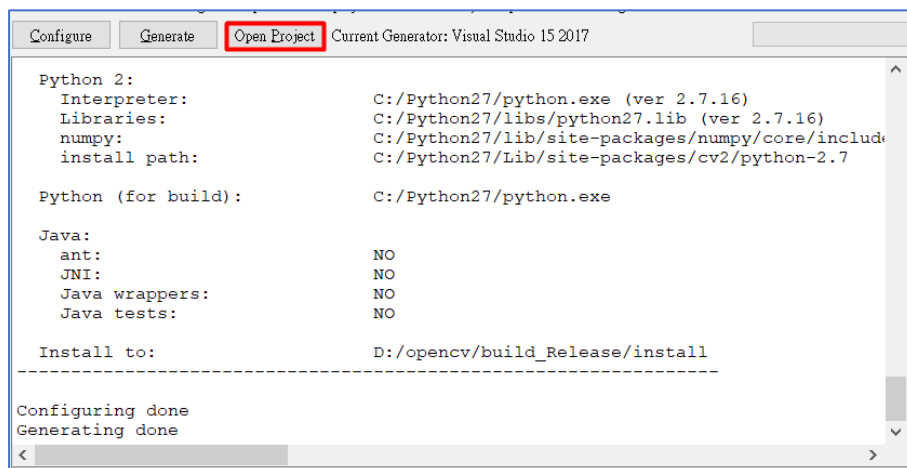


Figure 3-26

7. On the top menu, select **Build > Build Solutions**.

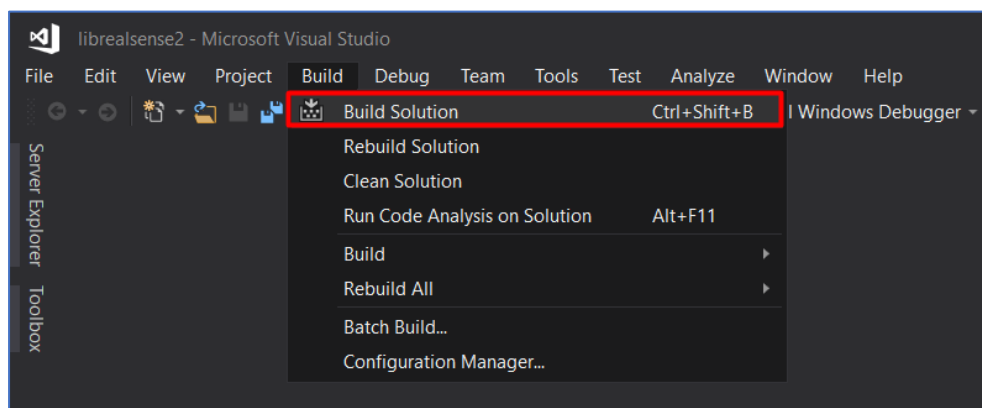
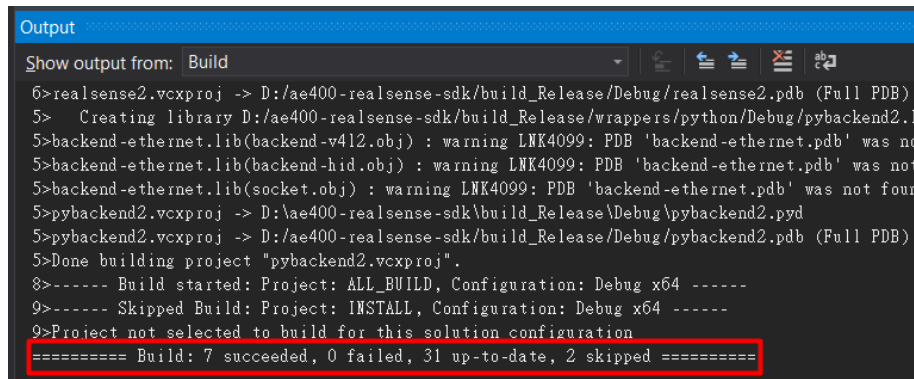


Figure 3-27

8. The wrapper is built.



```
Output
Show output from: Build
6>realsense2.vcxproj -> D:/ae400-realsense-sdk/build_Release/Debug/realsense2.pdb (Full PDB)
5> Creating library D:/ae400-realsense-sdk/build_Release/wrappers/python/Debug/pybackend2.lib
5>backend-ethernet.lib(backend-v4l2.obj) : warning LNK4099: PDB 'backend-ethernet.pdb' was not found
5>backend-ethernet.lib(backend-hid.obj) : warning LNK4099: PDB 'backend-ethernet.pdb' was not found
5>backend-ethernet.lib(socket.obj) : warning LNK4099: PDB 'backend-ethernet.pdb' was not found
5>pybackend2.vcxproj -> D:\ae400-realsense-sdk\build_Release\Debug\pybackend2.pdb
5>pybackend2.vcxproj -> D:/ae400-realsense-sdk/build_Release/Debug/pybackend2.pdb (Full PDB)
5>Done building project "pybackend2.vcxproj".
8>----- Build started: Project: ALL_BUILD, Configuration: Debug x64 -----
9>----- Skipped Build: Project: INSTALL, Configuration: Debug x64 -----
9>Project not selected to build for this solution configuration
===== Build: 7 succeeded, 0 failed, 31 up-to-date, 2 skipped =====
```

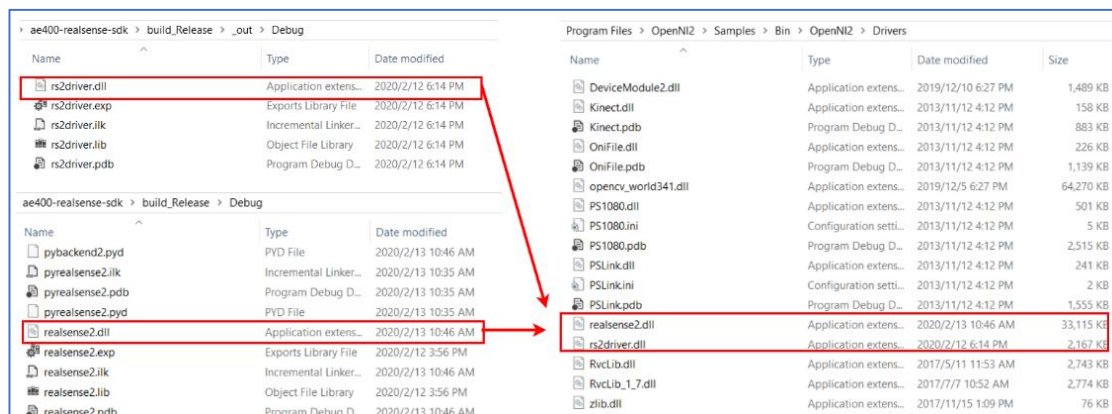
Figure 3-28

3.4.1.3 Starting Test Applications

Once the built process is completed, you can run a series of test applications for basic programming purposes available at **Program Files > OpenNI2 > Samples > Bin.**

Note: Prior to starting the test applications, copy the following files from the **source location** to the **location of OpenNI2**, e.g. C:\Program Files\OpenNI2\Samples\Bin\OpenNI2\Drivers

File Name	Source Location
rsdriver.dll	LIPSedge™ AE400 SDK > build_Release > _out > Debug D:\ae400-realsense-sdk\build_Release_out\Debug
realsense2.dll	LIPSedge™ AE400 SDK > build_Release > Debug e.g.: D:\ae400-realsense-sdk\build_Release\Debug



Viewing Live Image

1. On local host, go to **Program Files > OpenNI2 > Samples > Bin** and open **SimpleViewer.exe**.

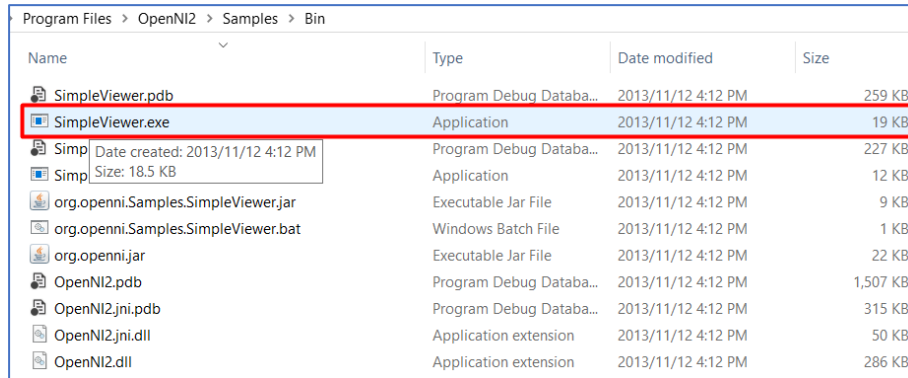


Figure 3-29

2. You can view depth / RGB / IR images on live.

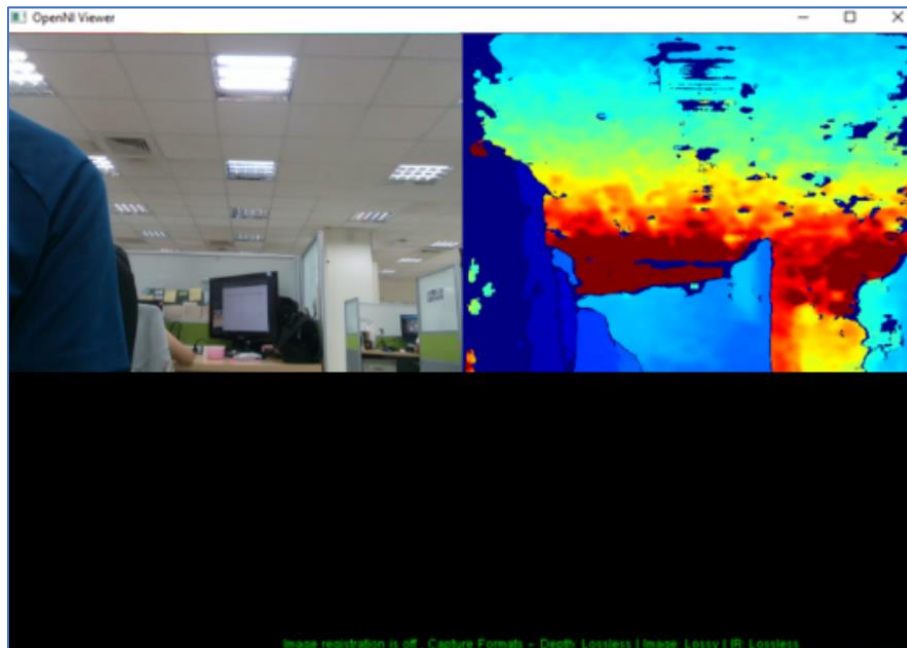


Figure 3-30

- Right-click on live image for the pop-up menu and select **Device > Streams**. Optionally turn **depth / RGB / IR images On / Off** from the pop-up menu.

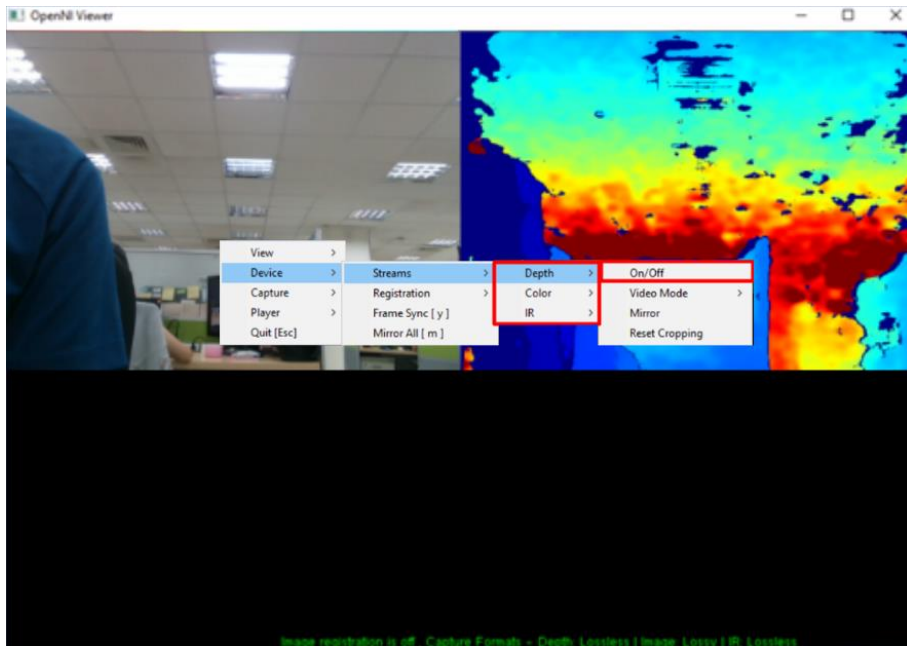


Figure 3-31

Recording Video Playback

1. Start **NiViewer**. Right-click on live image, select **Capture** and click **Start [s]**.

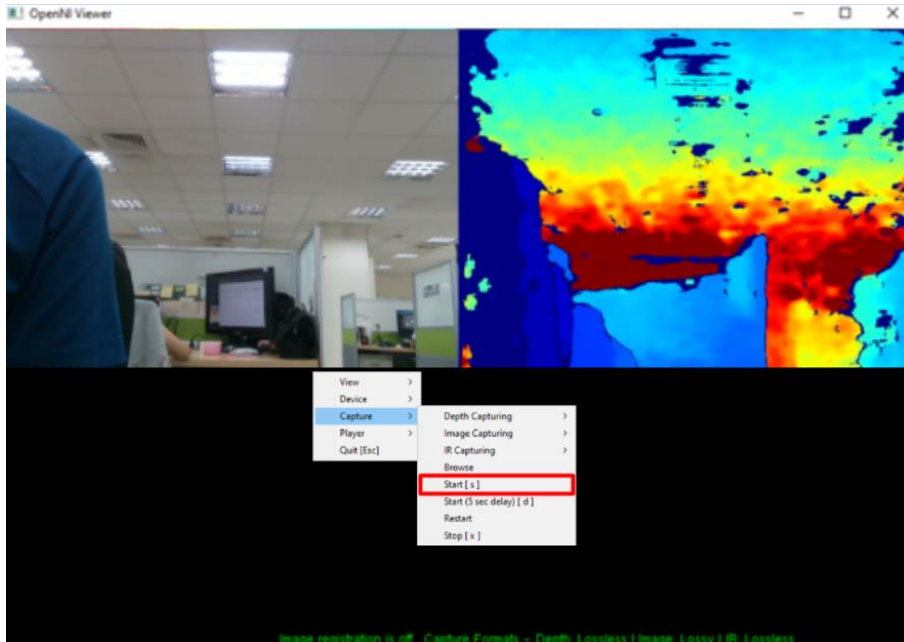


Figure 3-32

2. Click **Save** to save the recording to the desired location.

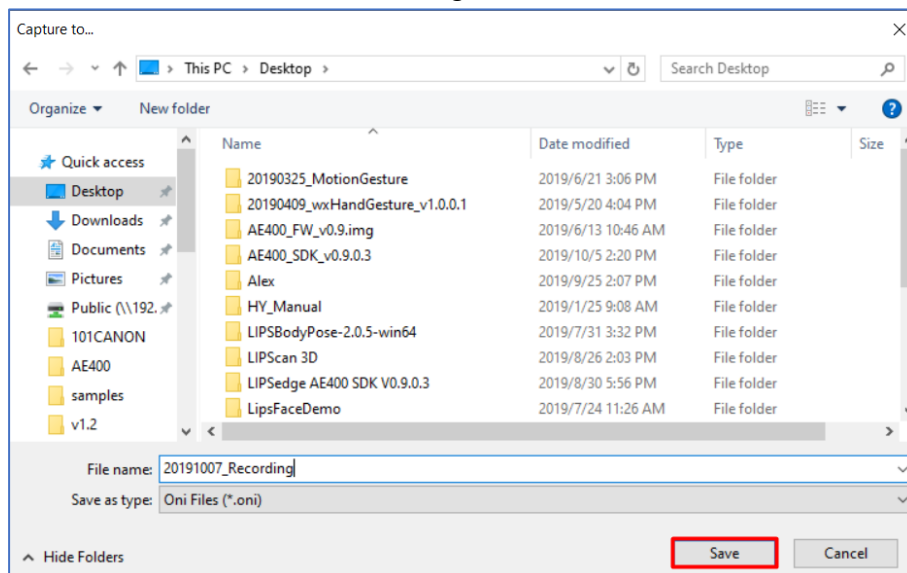


Figure 3-33

- When you are finished recording, right-click on live image, select **Capture** and click **Stop [x]**. The video clip will be stored at the location previously specified.

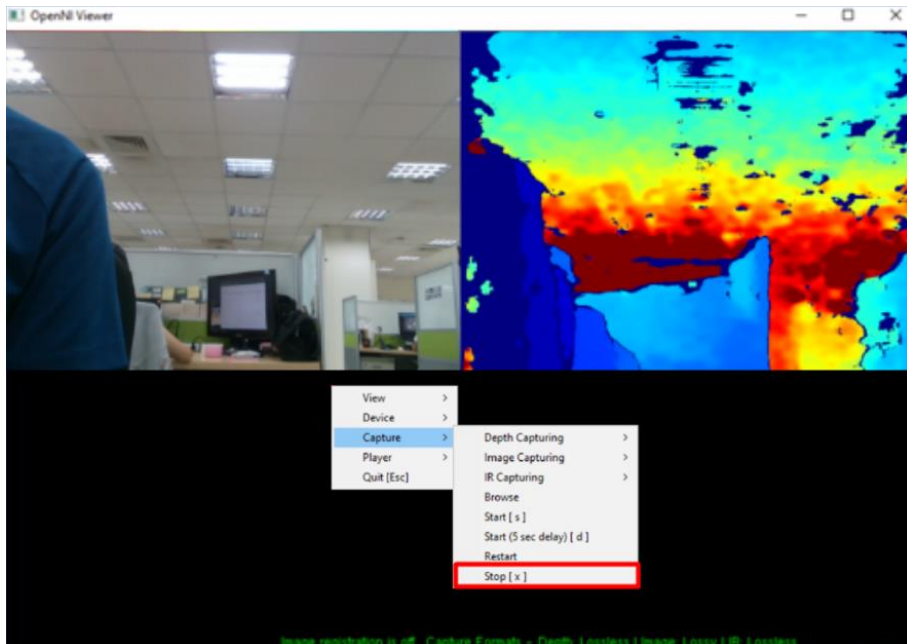


Figure 3-34

- To view video playback, right-click on live image, select **Capture** and click **Browse**.

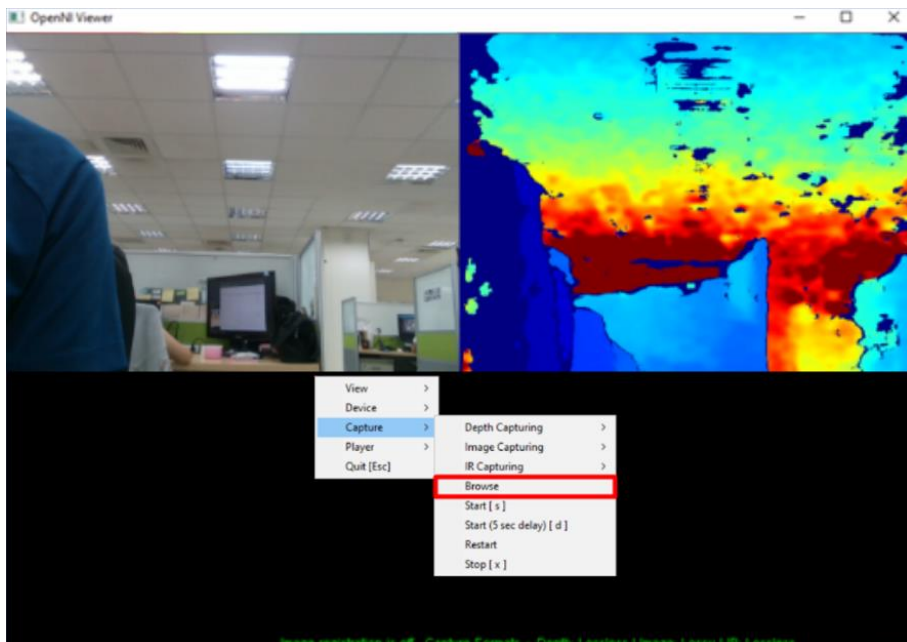


Figure 3-35

5. Select the previously recorded file. Upon opening the video playback, the NiViewer enters **Player Mode**.

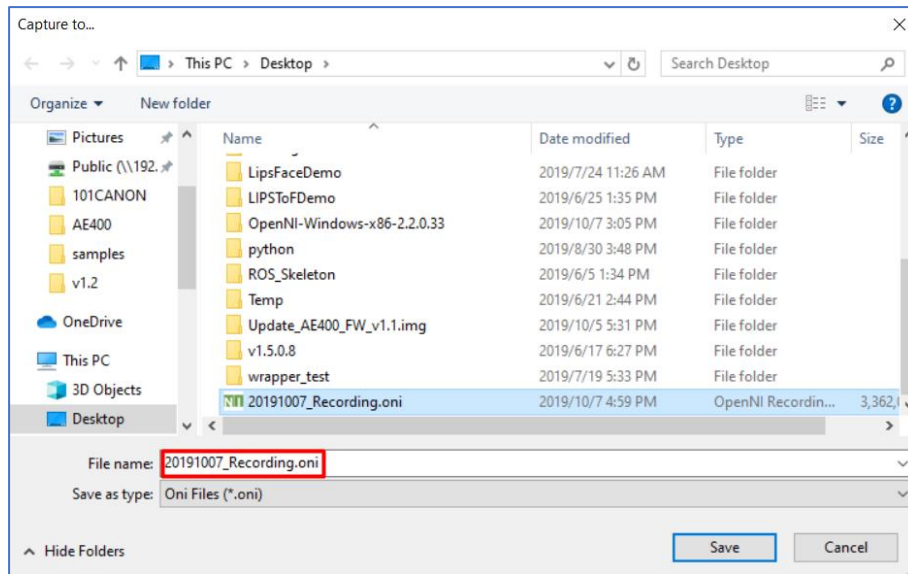


Figure 3-36

6. Under **Player Mode**, right-click on live image, select **Player** and click **Pause / Resume [Space]** or press **Space** to pause / resume the video.

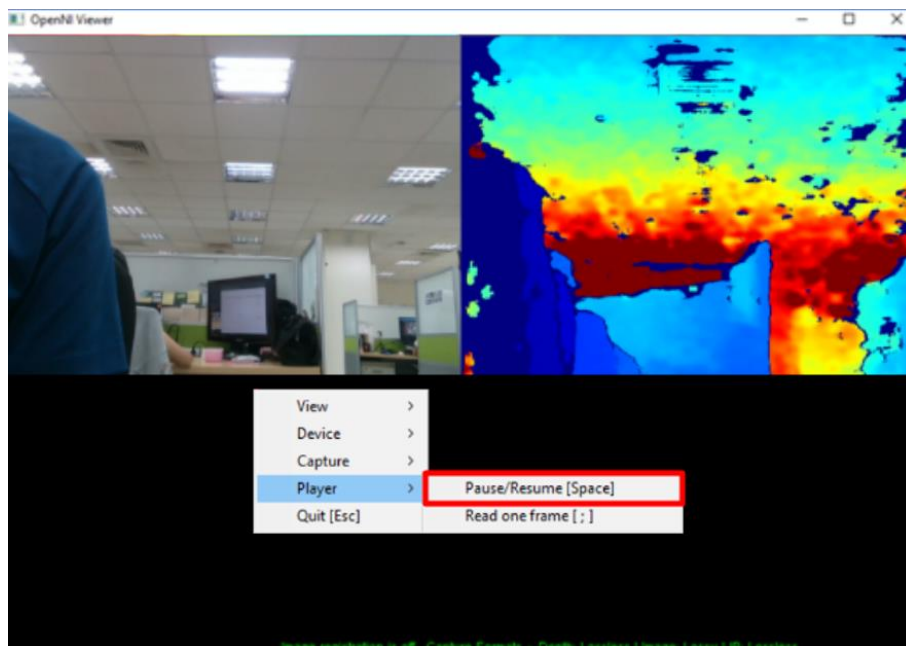


Figure 3-37

7. To view video playback frame by frame while pausing, select **Player** and click **Read one frame [;]** or press the ; key.

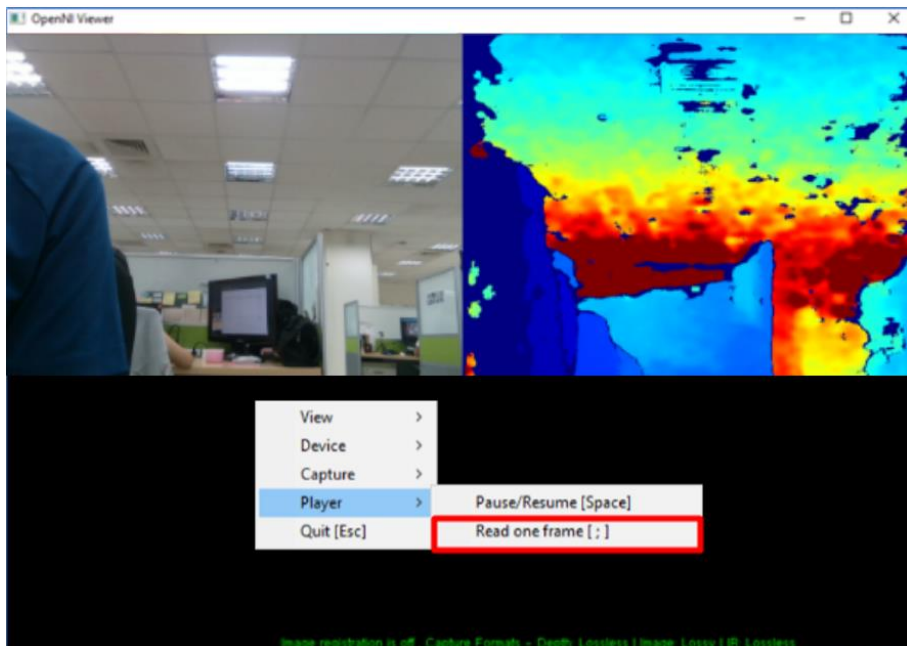


Figure 3-38

Map Depth Image to RGB Images

1. Start **NiViewer**. On the bottom **Status** bar, check the Image registration status. The image registration displays **Off** when the depth image is not mapped to RGB images.

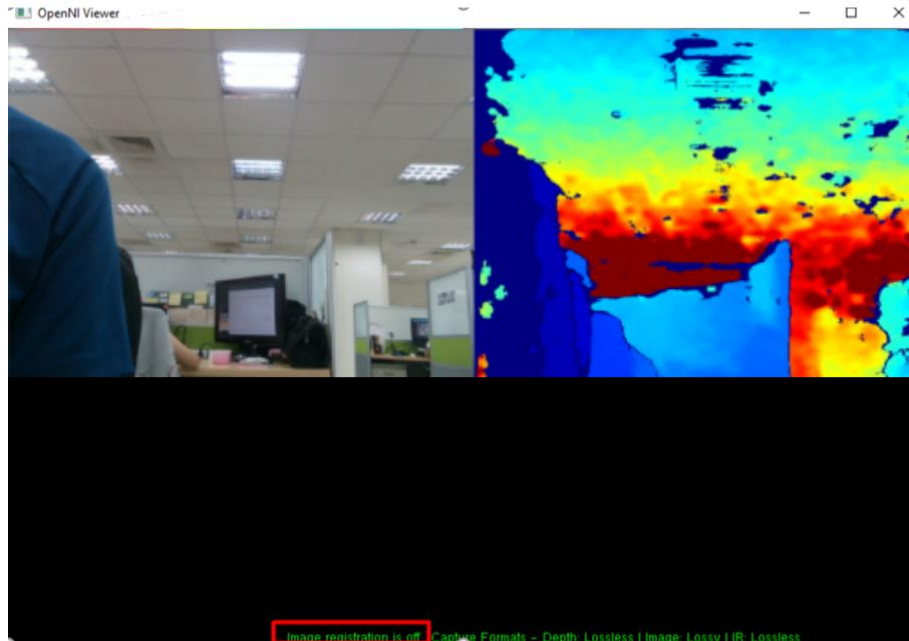


Figure 3-39

2. Right-click on live image, select **Registration** and click **Depth -> Image**.

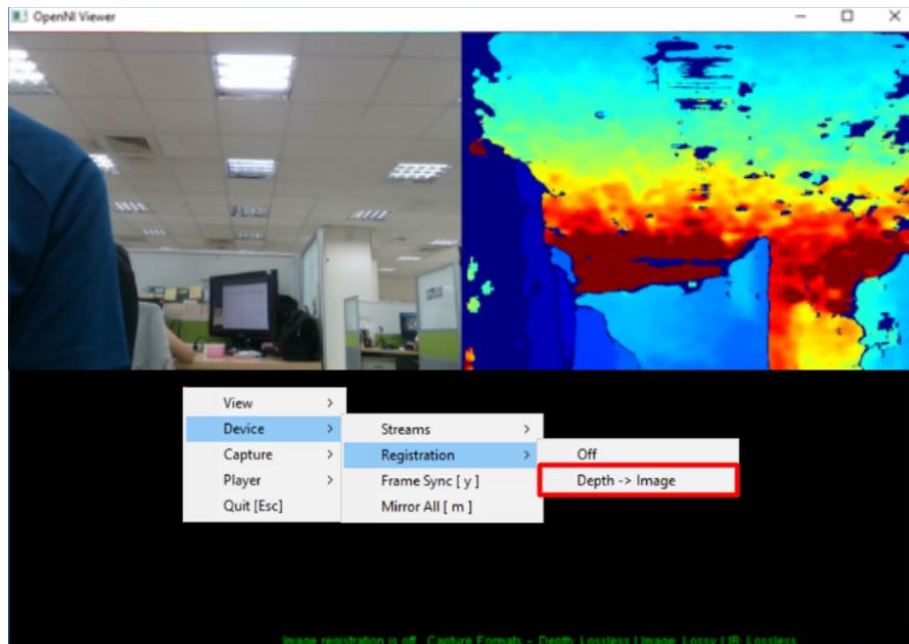


Figure 3-40

- On the bottom **Status** bar, the properly mapped images will be displayed as **Image Registration On**.

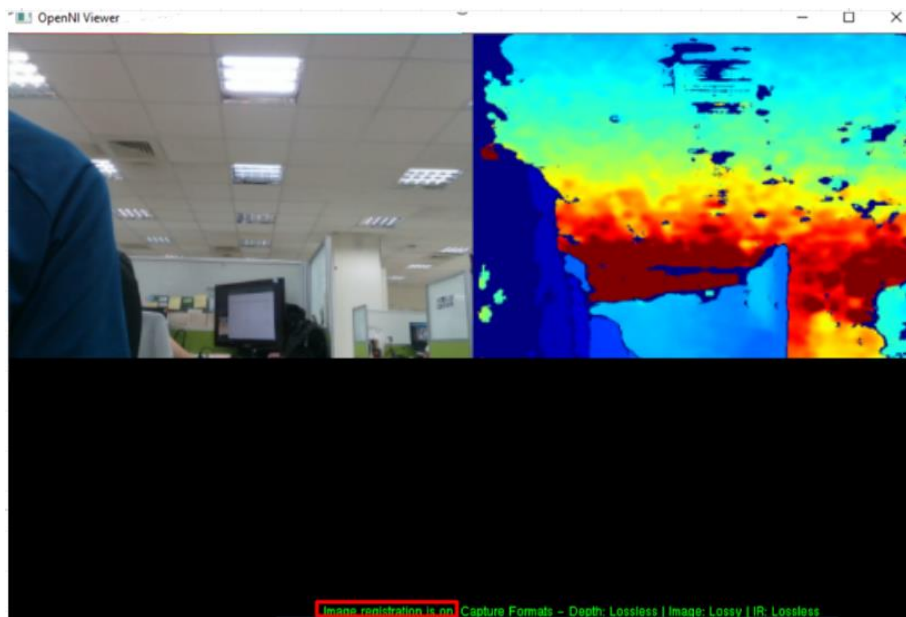


Figure 3-41

- On the system **Console**, the **registrationMode** changes to **1** as the images are properly mapped.

```
[RS2] ENABLE STREAM type=1 sensorId=0 streamId=0 640x480 @30
[RS2] ENABLE STREAM type=2 sensorId=1 streamId=0 640x480 @30
[RS2] ENABLE STREAM type=3 sensorId=0 streamId=1 640x480 @30
[RS2] rs2_create_pipeline
[RS2] rs2_pipeline_start_with_config
[RS2] oni::driver::Rs2Device::setProperty propertyId=5 dataSize=4
[RS2] registrationMode=0
[RS2] type=1 sensorId=0 streamId=0 fovX=56.682877 fovY=44.048698 depthScale=0.001000
[RS2] type=2 sensorId=1 streamId=0 fovX=54.949623 fovY=42.644009 depthScale=0.000000
[RS2] type=3 sensorId=0 streamId=1 fovX=56.682877 fovY=44.048698 depthScale=0.000000
[RS2] STARTED
[RS2] oni::driver::Rs2Device::setProperty propertyId=5 dataSize=4
[RS2] registrationMode=1
```

Figure 3-42

3.4.2 OpenCV Wrapper

For developing python applications, refer to the following instructions to build up OpenCV Wrapper.

3.4.2.1 Installing OpenCV

The configuration process of OpenCV wrapper requires **OpenCV 3.4**. Download OpenCV 3.4 and extract the downloaded file to your PC / laptop before building the wrappers.

1. Go to <https://opencv.org/releases/page/3/>. Click **OpenCV 3.4 – Windows** and the download process automatically starts.

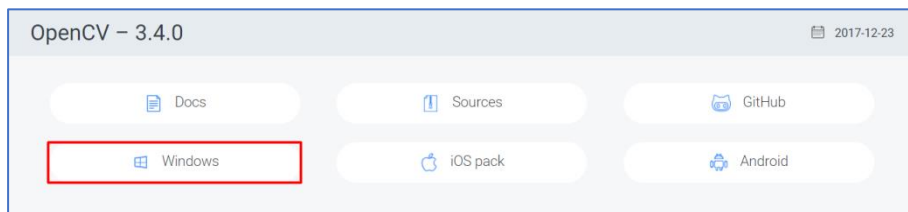


Figure 3-43

2. Open the downloaded file and click **Extract**. OpenCV 3.4 is extracted on your PC / laptop.

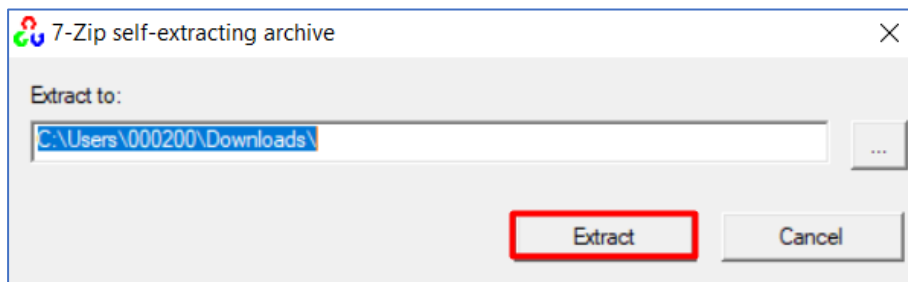


Figure 3-44

3.4.2.2 Building OpenCV through Compiler

Once OpenCV is installed, build the libraries through the compiler of **Microsoft Visual Studio 2017**. Before the building process, make sure the prerequisite items are installed.

For details on **relevant application / compiler**, refer to *2. Prerequisites*.
 For details on **OpenCV requirements**, refer to *3.4.2.1 Installing OpenCV*.

1. Clone the **LIPSedge™ AE400 SDK source code package** to your **PC / laptop**. For details, refer to *3.1 Cloning LIPSedge™ AE400 SDK*.
2. Assign the **location on your PC / laptop** for the built outcome for **CMake**. For details, refer to *Step 1 – 3 in 3.2 Building LIPSedge™ AE400 SDK*.
3. Start **CMake** and add the following location:

Field Name	Location
Where is the source code	Enter the location of the source code. E.g. D:\opencv
Where to build the binaries	Enter the location in which the outcome applications are saved. E.g. D:\opencv\build_Release

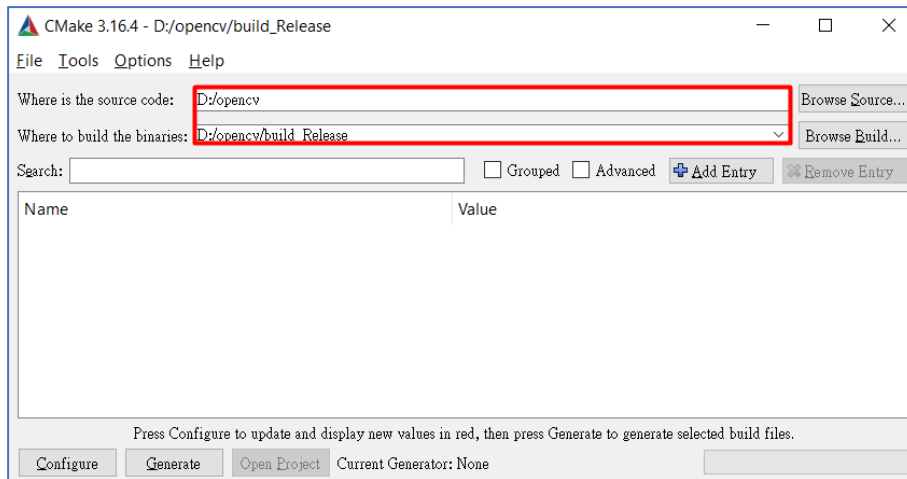


Figure 3-45

9. Click **Configure** and specify the options for **generators, platforms, and compilers** necessary for this build and click **Finish**.
 - **Specify the generator for this project:** Visual Studio 15 2017
 - **Optional platform:** x64
 - **Option for Compilers:** Use default native compilers

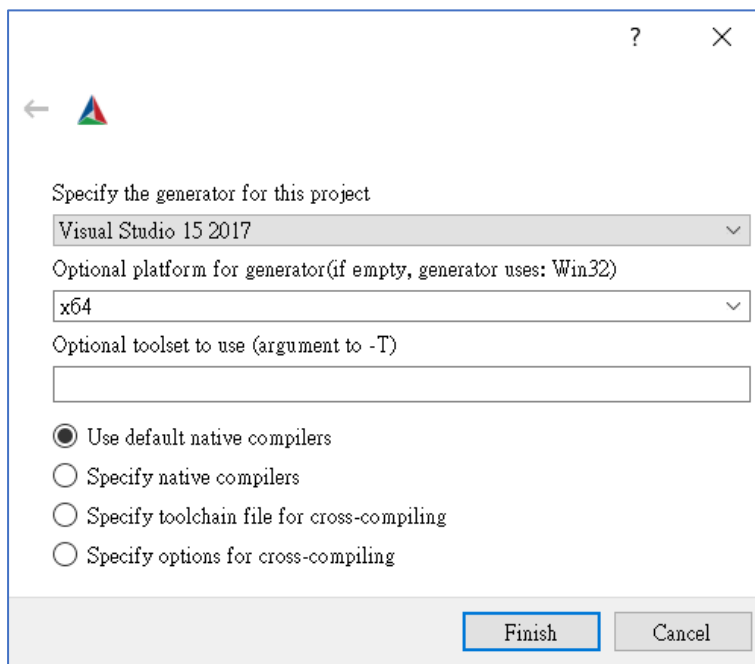


Figure 3-46

10. On the **Entry List** below, select **BUILD_CV_EXAMPLES**.

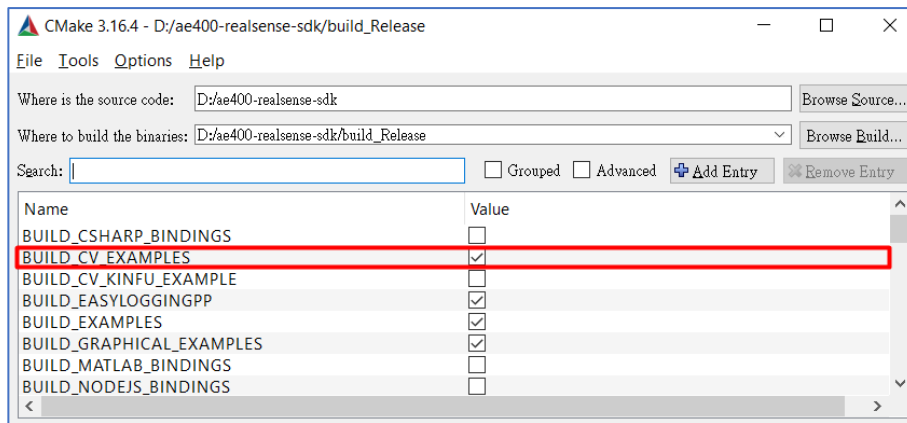


Figure 3-47

11. Click **Configure** and the entry **OpenCV_DIR** appears on the Entry List.

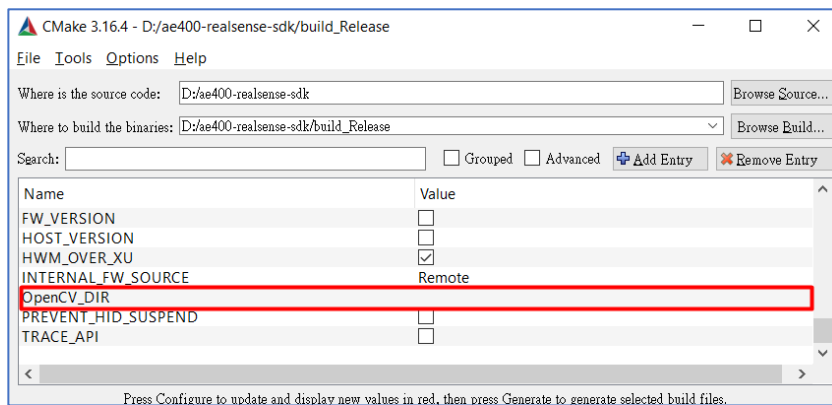


Figure 3-48

12. Assign the value to **OpenCV_DIR**.

- **OpenCV_DIR**: Extracted OpenCV Folder > build, e.g. C:\opencv\build.

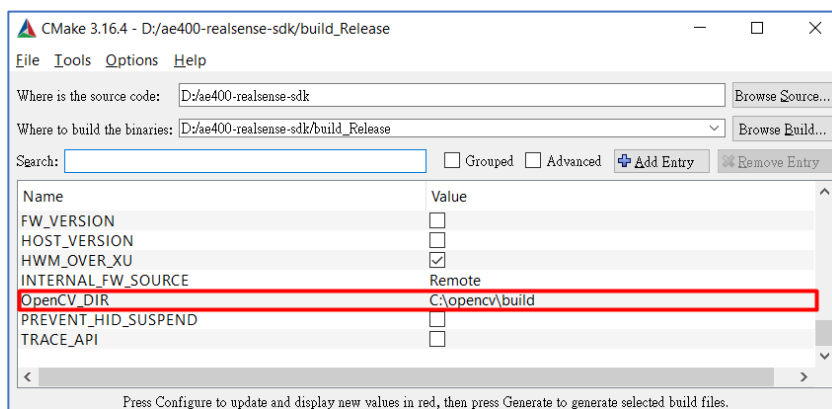


Figure 3-49

13. Click **Generate** and the Makefiles for OpenCV libraries are generated.

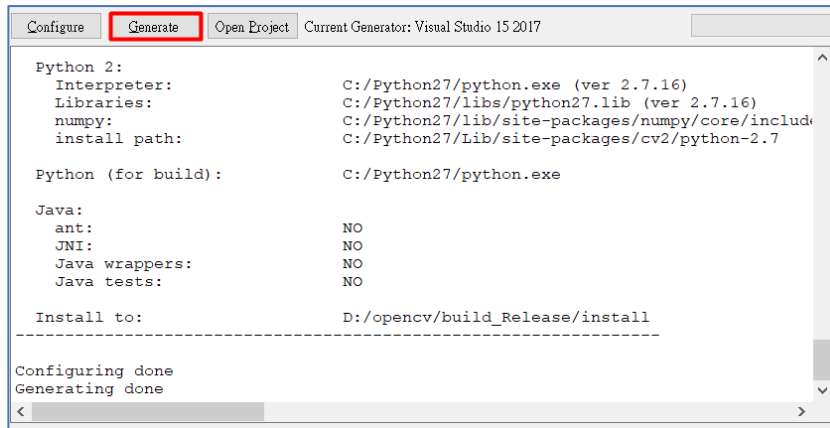


Figure 3-50

14. Click **Open Project** to launch **Microsoft Visual Studio Community 2017**.

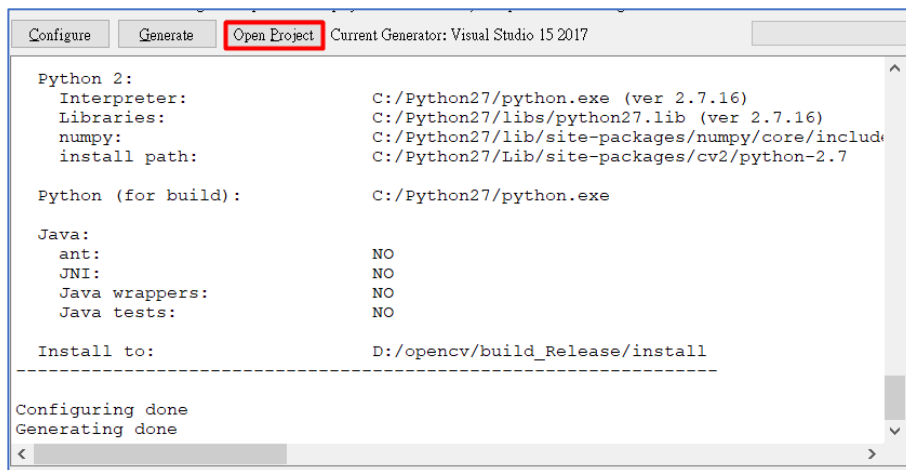


Figure 3-51

8. On the side menu, find **Examples > OpenCV**.

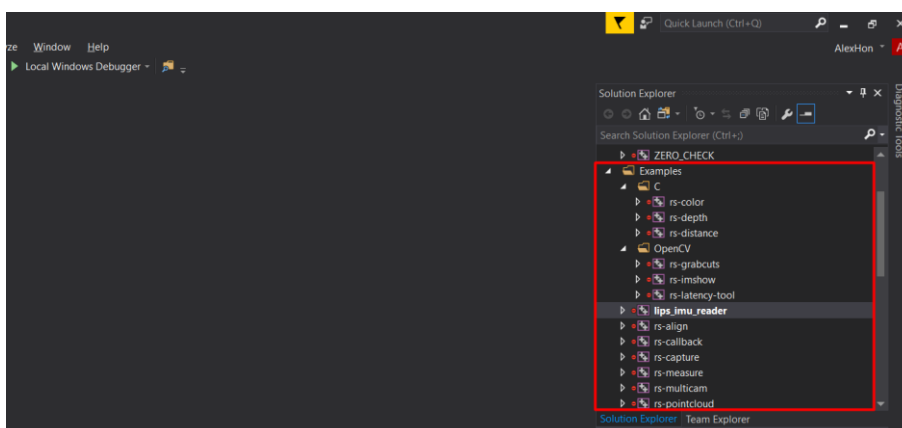


Figure 3-52

9. On the right menu, find the **source code project** of a **sample program**. Here, we use **rs-imshow** as an example.

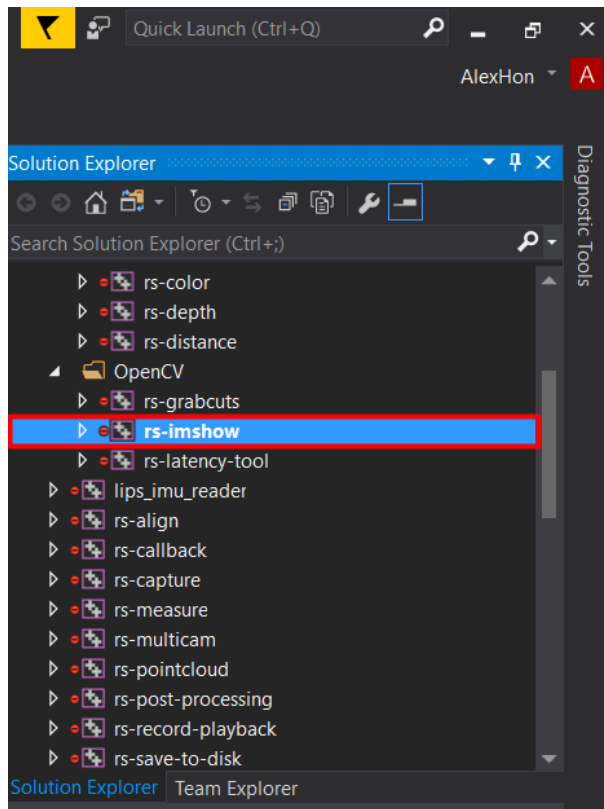


Figure 3-53

10. Right-click the project and select **Set as Startup Project**.

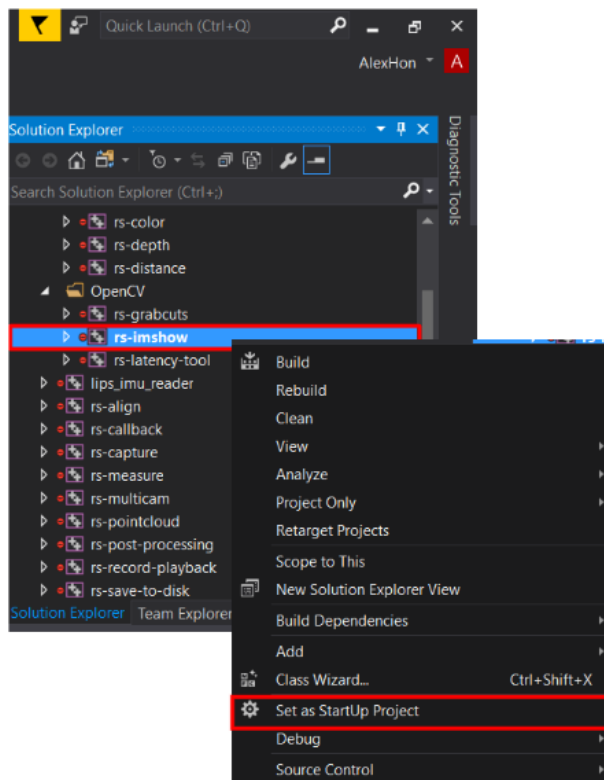


Figure 3-54

11. Press **F5** and the application is built.

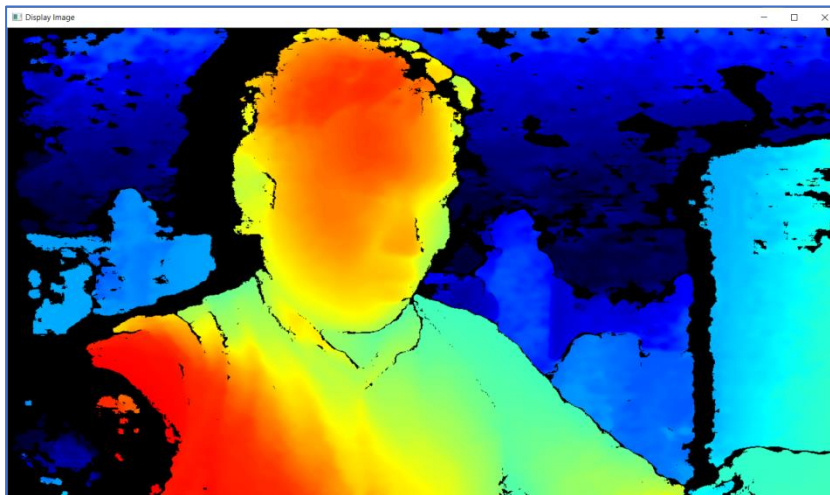


Figure 3-55

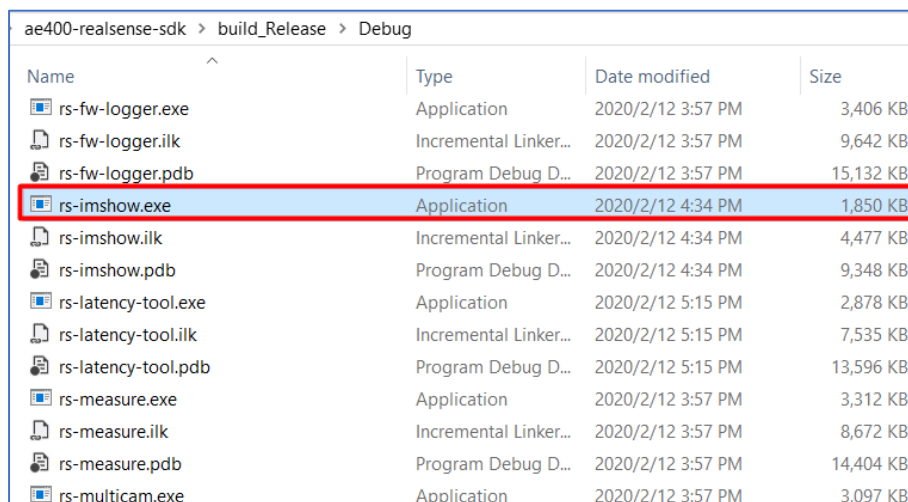
3.4.2.3 Starting Test Application

The OpenCV wrapper is bundled with a series of Intel® RealSense™, source codes which can be compiled into various applications.

Viewing Live Images

With the **rs-imshow.exe** program, you can view live depth / RGB image streaming.

1. Build the OpenCV wrapper from LIPSedge™ AE400 SDK. For details, refer to *3.4.2.2 Building OpenCV through Compiler*.
2. Once the LIPSedge™ AE400 SDK – OpenCV wrapper is built, go to **LIPSedge™ AE400 SDK > build_Release > Debug** and click **rs-imshow.exe**.



Name	Type	Date modified	Size
rs-fw-logger.exe	Application	2020/2/12 3:57 PM	3,406 KB
rs-fw-logger.ilc	Incremental Linker...	2020/2/12 3:57 PM	9,642 KB
rs-fw-logger.pdb	Program Debug D...	2020/2/12 3:57 PM	15,132 KB
rs-imshow.exe	Application	2020/2/12 4:34 PM	1,850 KB
rs-imshow.ilc	Incremental Linker...	2020/2/12 4:34 PM	4,477 KB
rs-imshow.pdb	Program Debug D...	2020/2/12 4:34 PM	9,348 KB
rs-latency-tool.exe	Application	2020/2/12 5:15 PM	2,878 KB
rs-latency-tool.ilc	Incremental Linker...	2020/2/12 5:15 PM	7,535 KB
rs-latency-tool.pdb	Program Debug D...	2020/2/12 5:15 PM	13,596 KB
rs-measure.exe	Application	2020/2/12 3:57 PM	3,312 KB
rs-measure.ilc	Incremental Linker...	2020/2/12 3:57 PM	8,672 KB
rs-measure.pdb	Program Debug D...	2020/2/12 3:57 PM	14,404 KB
rs-multicam.exe	Application	2020/2/12 3:57 PM	3,097 KB

Figure 3-56

3. You can view depth / RGB images live.

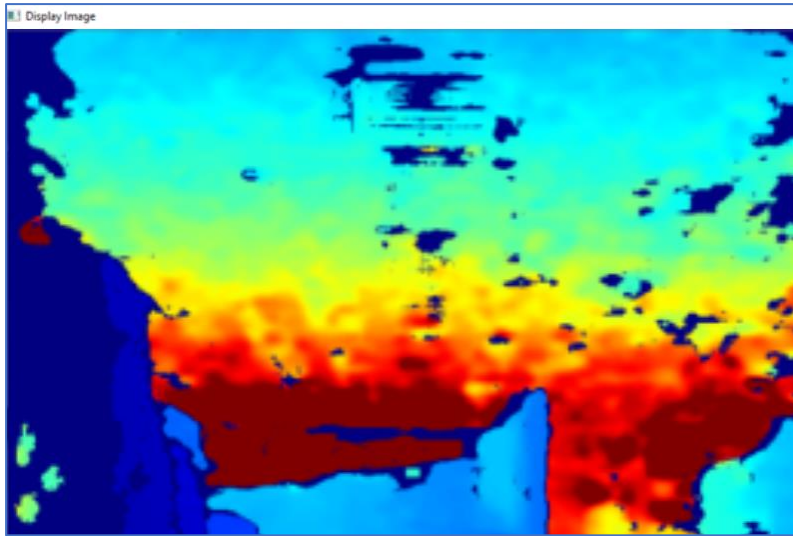


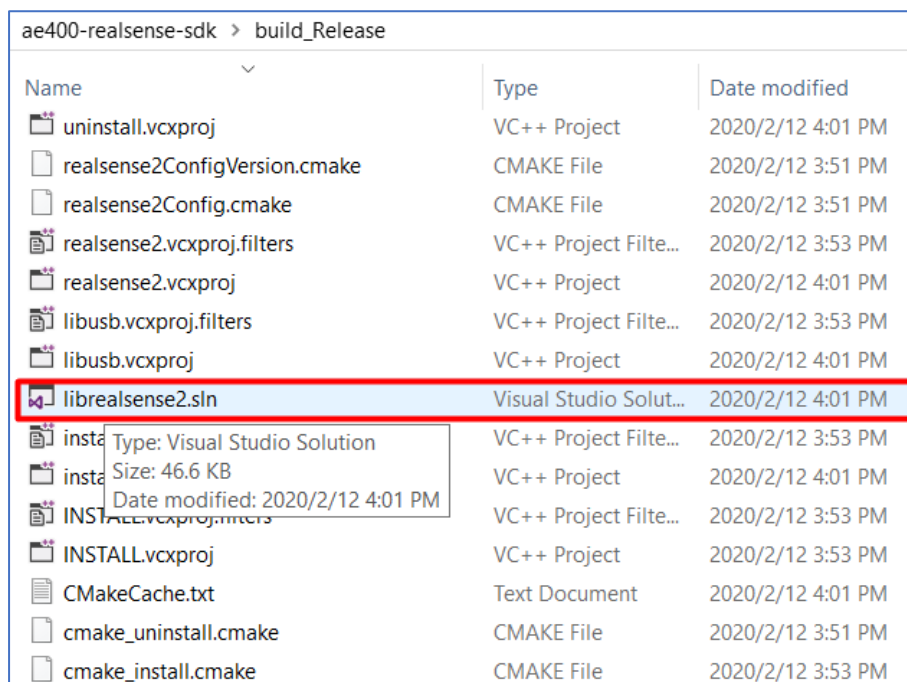
Figure 3-57

To exit the application, press any key.

View RGB Images without Background

The **rs-grabcuts.exe** program is a viewer capable of separating the subject in the foreground of RGB images from the background.

1. Build the OpenCV wrapper from LIPSedge™ AE400 SDK. For details, refer to *3.4.2.2 Building OpenCV through Compiler*.
2. Once the LIPSedge™ AE400 SDK – OpenCV wrapper is built, go to **LIPSedge™ AE400 SDK > build_Release** and click **librealsense2.sln**.



Name	Type	Date modified
uninstall.vcxproj	VC++ Project	2020/2/12 4:01 PM
realsense2ConfigVersion.cmake	CMAKE File	2020/2/12 3:51 PM
realsense2Config.cmake	CMAKE File	2020/2/12 3:51 PM
realsense2.vcxproj.filters	VC++ Project Filte...	2020/2/12 3:53 PM
realsense2.vcxproj	VC++ Project	2020/2/12 4:01 PM
libusb.vcxproj.filters	VC++ Project Filte...	2020/2/12 3:53 PM
libusb.vcxproj	VC++ Project	2020/2/12 4:01 PM
librealsense2.sln	Visual Studio Solut...	2020/2/12 4:01 PM
install.vcxproj.filters	VC++ Project Filte...	2020/2/12 3:53 PM
install.vcxproj	VC++ Project	2020/2/12 4:01 PM
INSTALL.vcxproj.filters	VC++ Project Filte...	2020/2/12 3:53 PM
INSTALL.vcxproj	VC++ Project	2020/2/12 3:53 PM
CMakeCache.txt	Text Document	2020/2/12 4:01 PM
cmake_uninstall.cmake	CMAKE File	2020/2/12 3:51 PM
cmake_install.cmake	CMAKE File	2020/2/12 3:53 PM

Figure 3-58

3. On the right menu, right click **rs-grabcuts** and select **Set as Startup Project**.

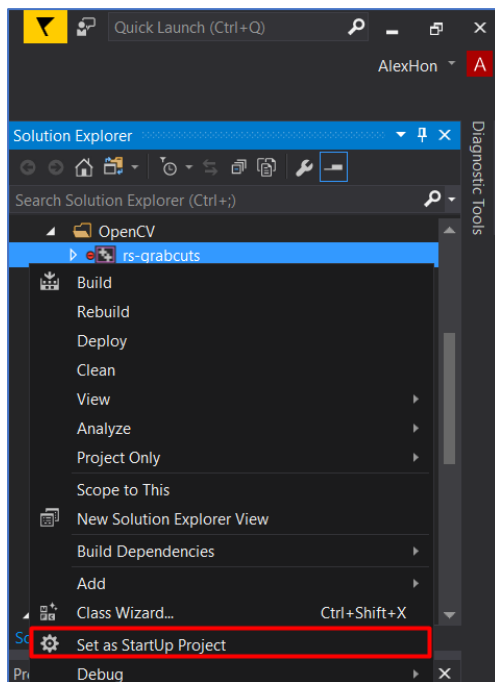


Figure 3-59

4. Press **F5** to build the sample application.
5. You can view color images without background.

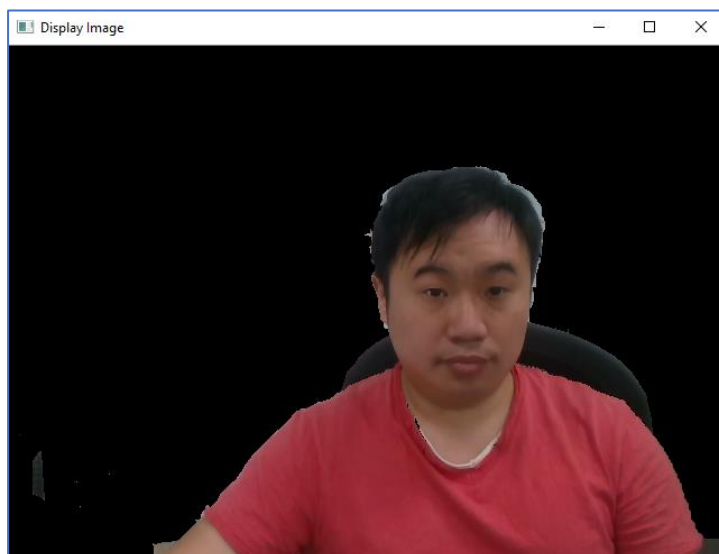


Figure 3-60

To exit the application, press any key.

3.4.3 Python Wrapper

For developing python applications, refer to the following instructions to build up Python Wrapper.

3.4.3.1 Installing Python

Intel® RealSense™, Python Wrapper requires **Python 2.7 / 3.0 or above** to be installed for the building process. Install **Python** prior to the wrapper building process. Here, we use **Python 2.7** for example.

1. Go to <https://www.python.org/downloads/release/python-2716/>. Click **Windows x86-64 MSI Installer** to download **Python 2.7 Windows (x64)**.

Version	Operating System	Description	MDS Sum	File Size	GPG
Gitipped source tarball	Source release		f1a2ace63106844833d01485466ece0	17431748	SIG
XZ compressed source tarball	Source release		30151d85a2c0479c09ea2cbe61f2aaf5	12752104	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	70b0f58eba7078b174056369b076c085	30252432	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	a3af70c13c654276d66c3c1cb1772dc7	23743901	SIG
Windows debug information files	Windows		f9490edbbf58b10b6f5718badc08b1f8	25088166	SIG
Windows debug information files for 64-bit binaries	Windows		4292c4db30c27fe4dbbee544967b6452	25899174	SIG
Windows help file	Windows		30b229b6712b231d2dff9211c7b21e2	6263118	SIG
Windows x86-64 MSI installer	Windows	for AMD64/EM64T/x64	2fe86194bb4027be75b2985202711a79	20361216	SIG
Windows x86 MSI installer	Windows		912428345b7e0428544ec4edcdff70286	19419136	SIG

Figure 3-61

2. Click the downloaded file. Select **Install for all users** and click **Next**.

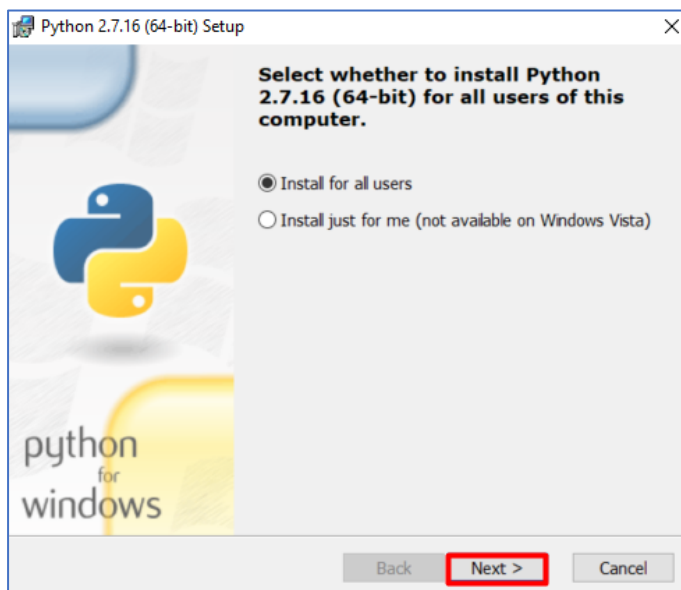


Figure 3-62

3. Click **Next**.

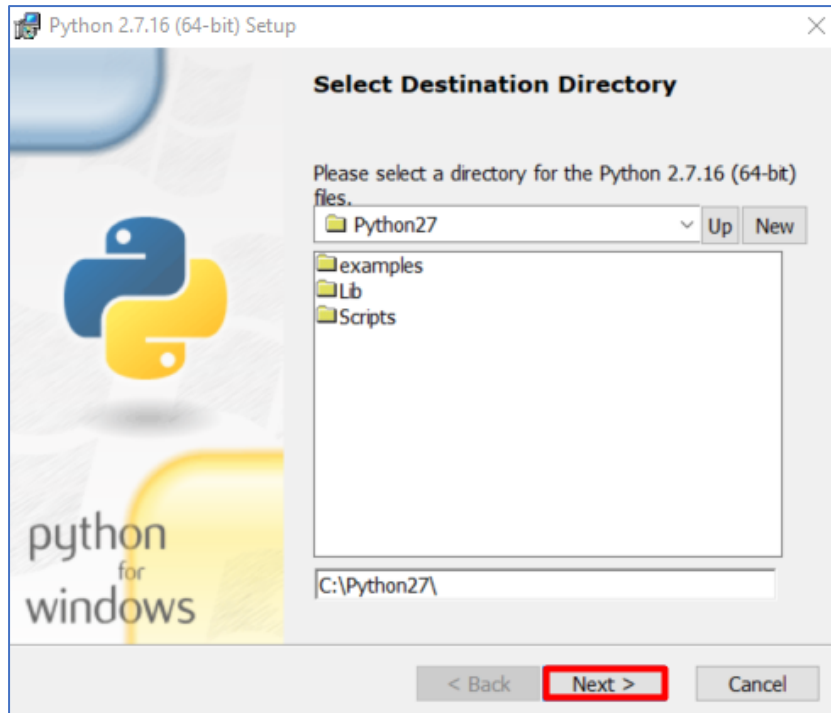


Figure 3-63

4. Click **Next**.

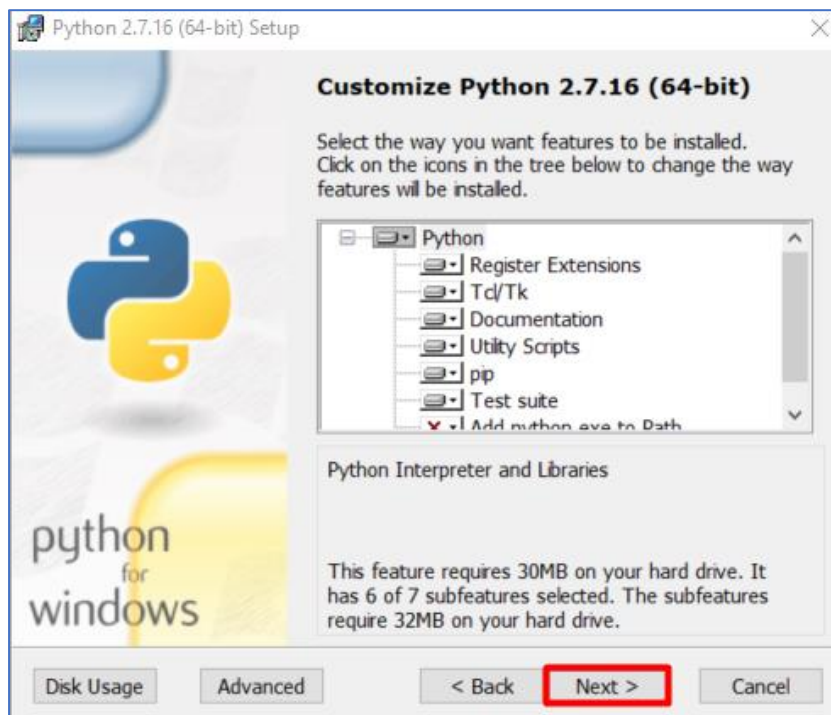


Figure 3-64

5. Click **Finish**.

3.4.3.2 Configuring Environment Variables

Upon installing the Python package, the environment variables are incorporated into Windows Environment Variables. If for any reason the application fails to incorporate the environment variables, you can incorporate the environment variables manually.

1. On the **Desktop** of your PC / laptop, right-click **This PC** and select **Properties**.

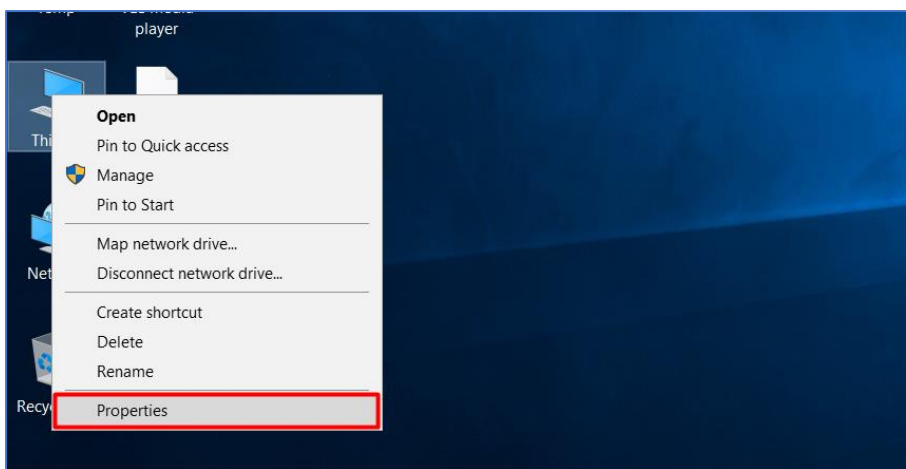


Figure 3-65

2. Click **Advanced system settings**.

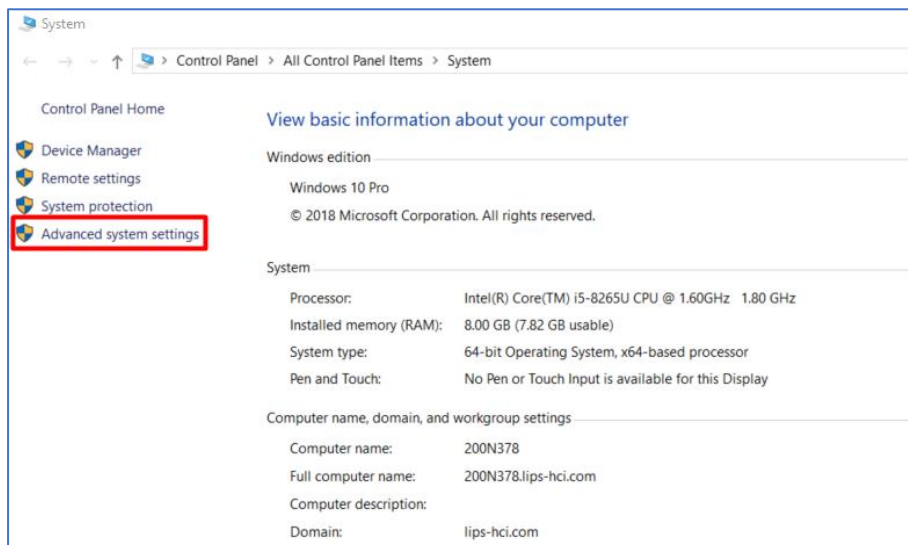


Figure 3-66

3. Click **Environment Variables**.

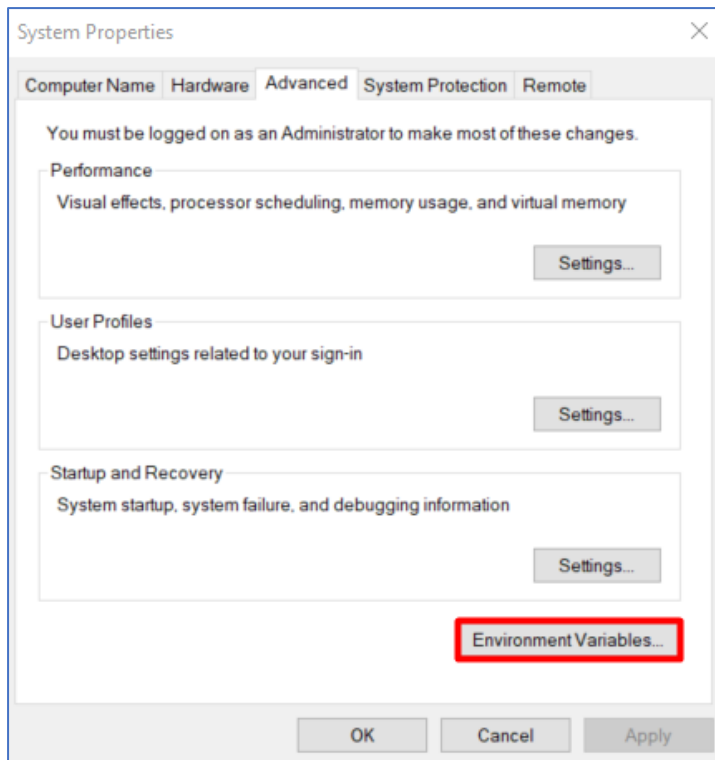


Figure 3-67

4. Under **System variables**, select **Path** and click **Edit**.

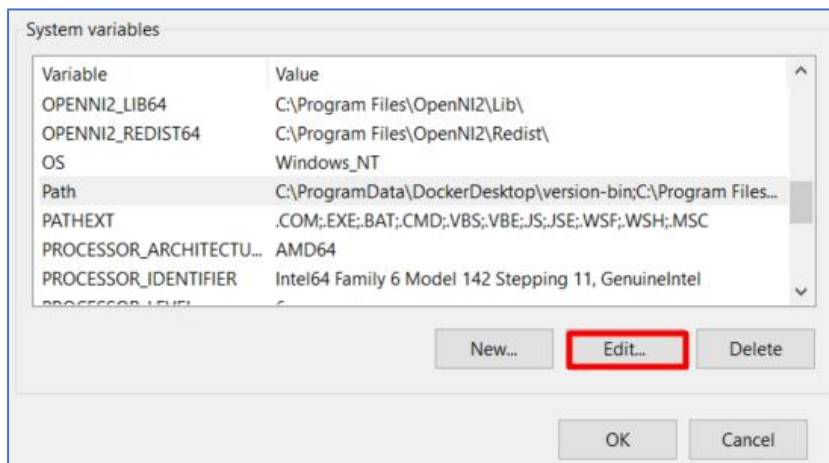


Figure 3-68

5. Type the following location and click **OK**.

Location	C:\Python27\Scripts
-----------------	---------------------

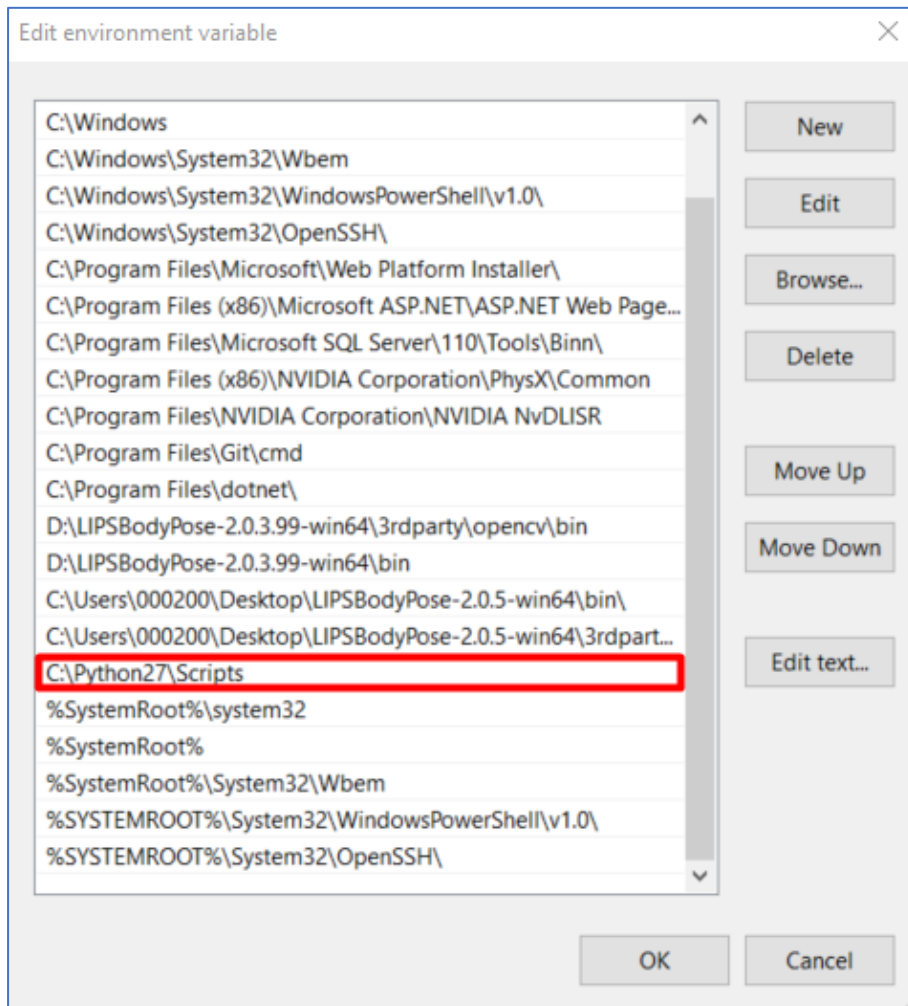


Figure 3-69

3.4.3.3 Configuring Development Environment

Install the **NumPy** library to enable Python with advanced mathematical functions necessary for computing dimensional data for image representation.

Installing NumPy

1. Start **Command Prompt** and type the following command.

```
pip install numpy
```

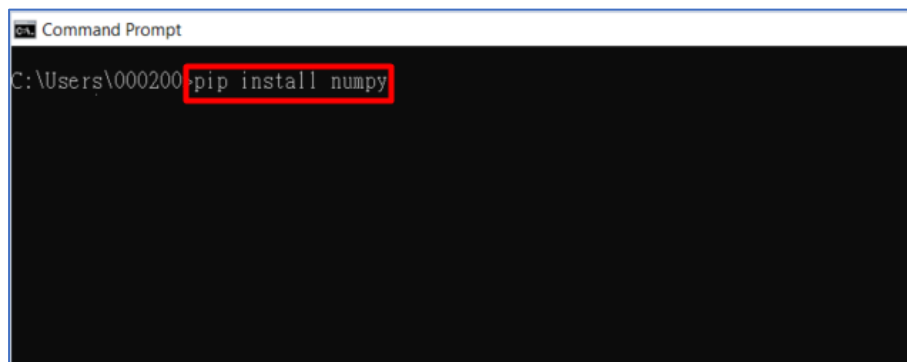


Figure 3-70

2. The library will be automatically installed.

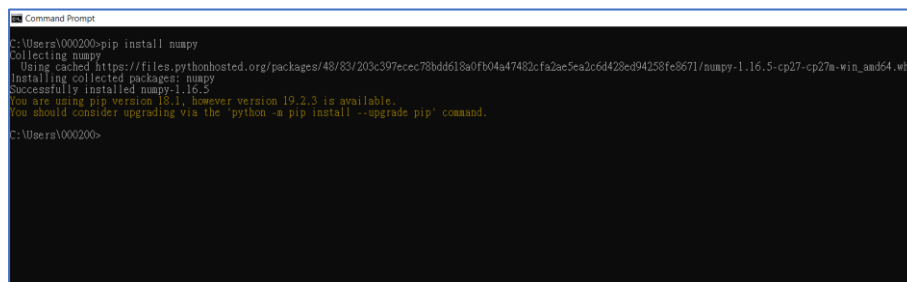


Figure 3-71

3.4.3.4 Building Python from Source Code

Once Python is installed, build the libraries through the compiler of **Microsoft Visual Studio 2017**. Before the building process, make sure the prereduced items are installed.

For details on **relevant application / compiler**, refer to *2. Prerequisites*.
 For details on **Python requirements**, refer to *3.4.31 Installing Python*.
 For details on **development environment configuration**, refer to *3.4.3.2 Configuring Environment Variables* and *3.4.3.3 Configuring Development Environment*.

1. Clone the **LIPSedge™ AE400 SDK source code package** to your **PC / laptop**. For details, refer to *3.1 Cloning LIPSedge™ AE400 SDK*.
2. Assign the **location on your PC / laptop** for the built outcome for **CMake**. For details, refer to *Step 1 – 3 in 3.2 Building LIPSedge™ AE400 SDK*.
3. On the **Entry List** below, select **BUILD_PYTHON_BINDINGS** and click **Configure**.

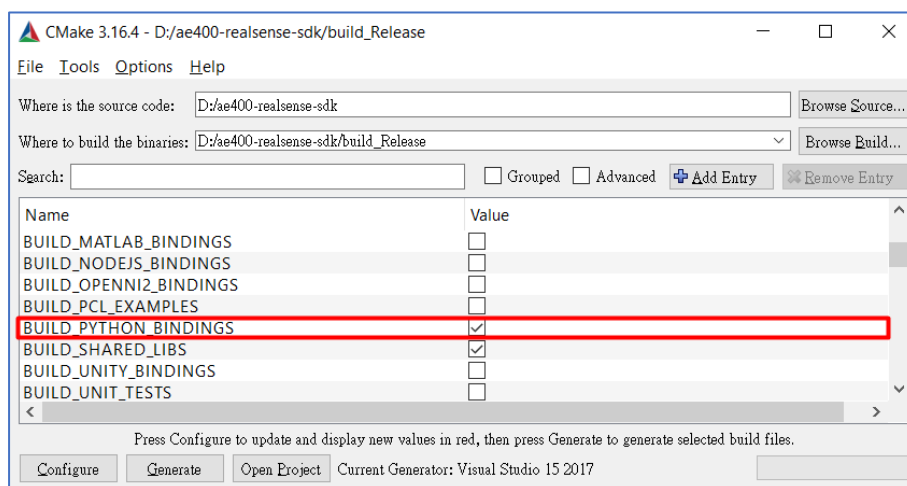


Figure 3-72

4. Generate a **Makefile** for the LIPSedge™ AE400 SDK containing the Python wrapper.

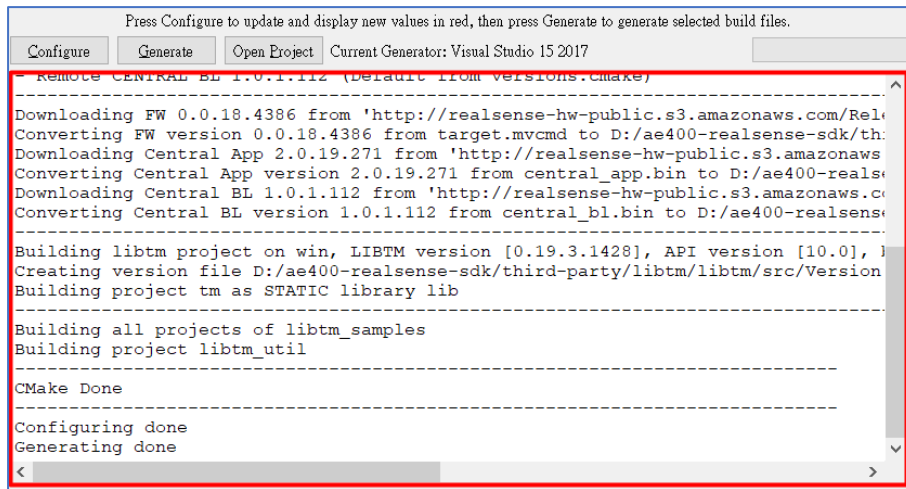


Figure 3-73

5. Go to LIPSedge™ AE400 > build_Release and open librealsense2.sln.

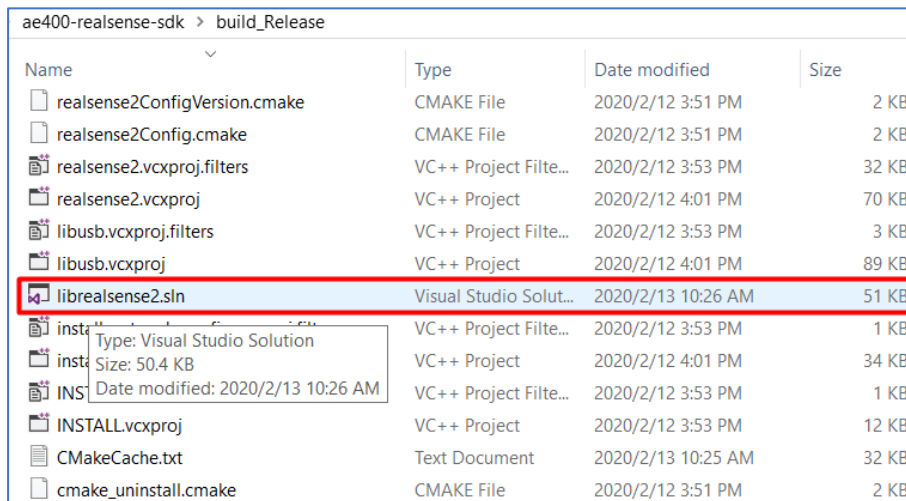


Figure 3-74

- On the right menu, right-click **pyrealsense2** and select **Build**.

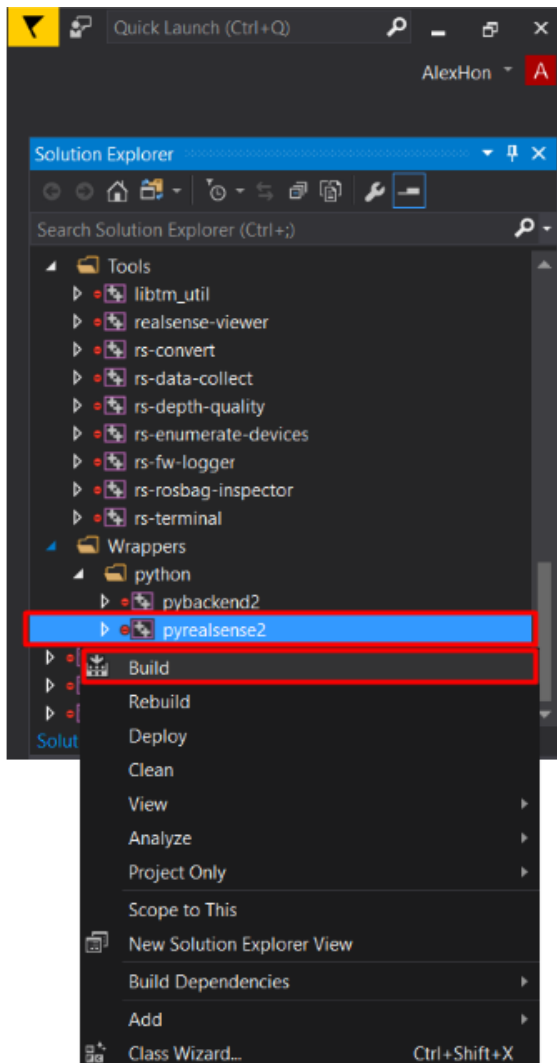


Figure 3-75

- The wrapper is built.

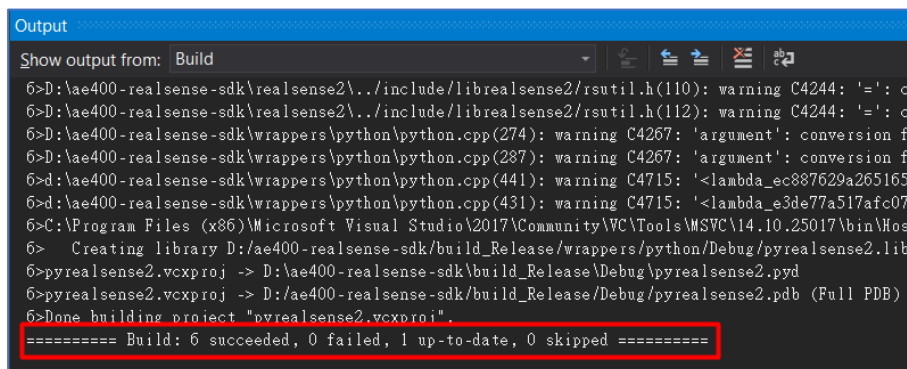


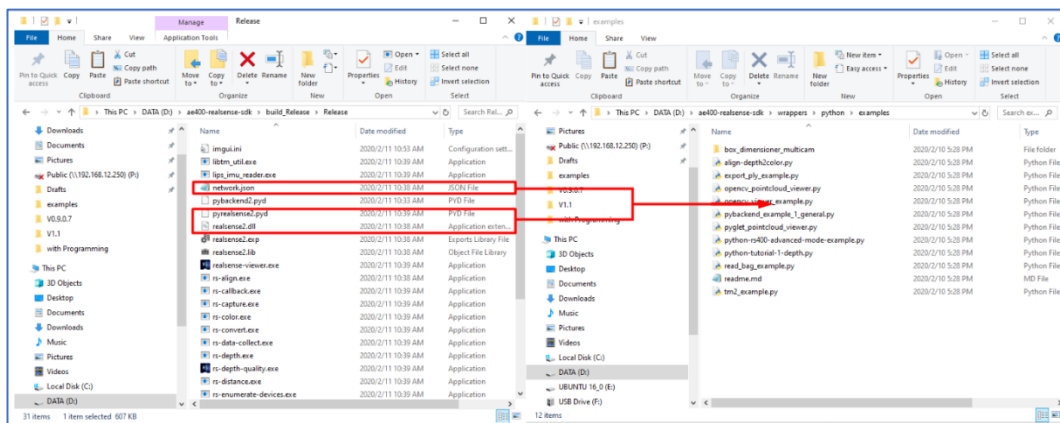
Figure 3-76

3.4.3.4 Starting Test Applications

The Python wrapper is bundled with a series of Intel® RealSense™, source codes which can be compiled into various applications.

Note: Prior to starting the test applications, copy the following files from **Output folder > Release** to the location of Python Wrapper, e.g. **LIPSedge™ AE400 SDK > wrappers > python > examples**.

- network.json
- realsense2.dll
- pyrealsense2.pyd



1. Go to **LIPSedge™ AE400 SDK > wrappers > python > examples**.

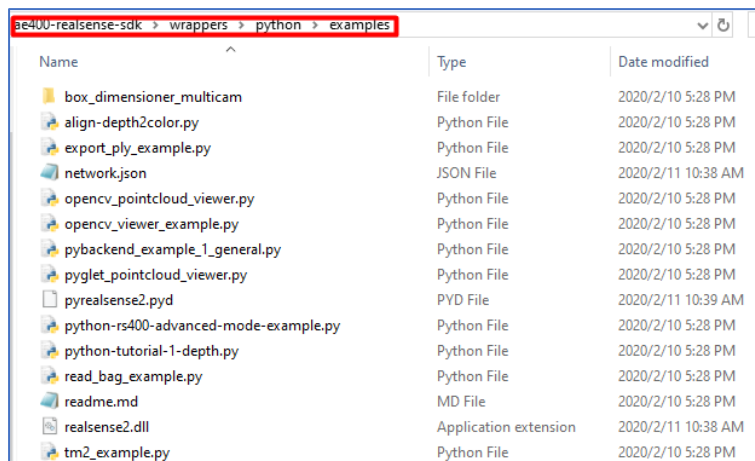


Figure 3-77

2. From there, you can access the following test applications:

No.	Name	Features
1.	opencv_viewer_example.py	Displays live image data.
2.	align-depth2color.py	Mapping depth / RGB images.
3.	opencv_pointcloud_viewer.py	Displays point cloud image data.

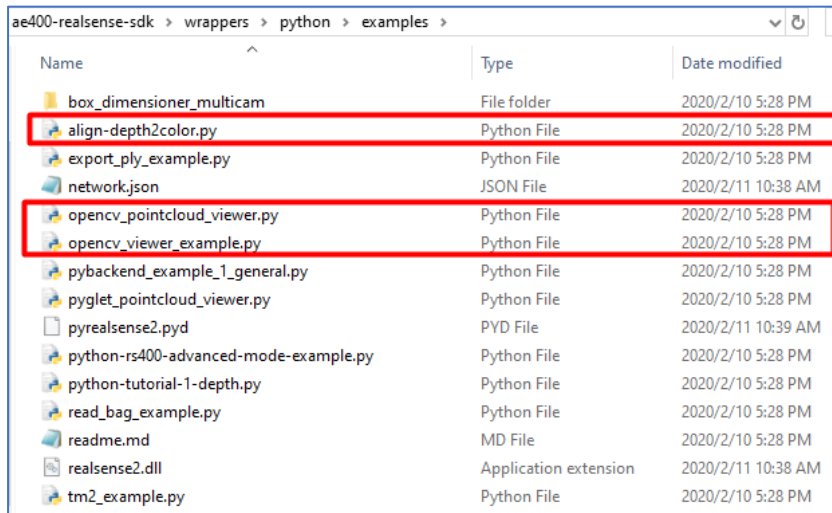


Figure 3-78

Viewing Live Image

Start `opencv_viewer_exmample.py` to view live image data. Depth and RGB images are supported. By default, the supported image resolution / fps is **640 x 480 @ 30**.

1. Click `opencv_viewer_example.py`.

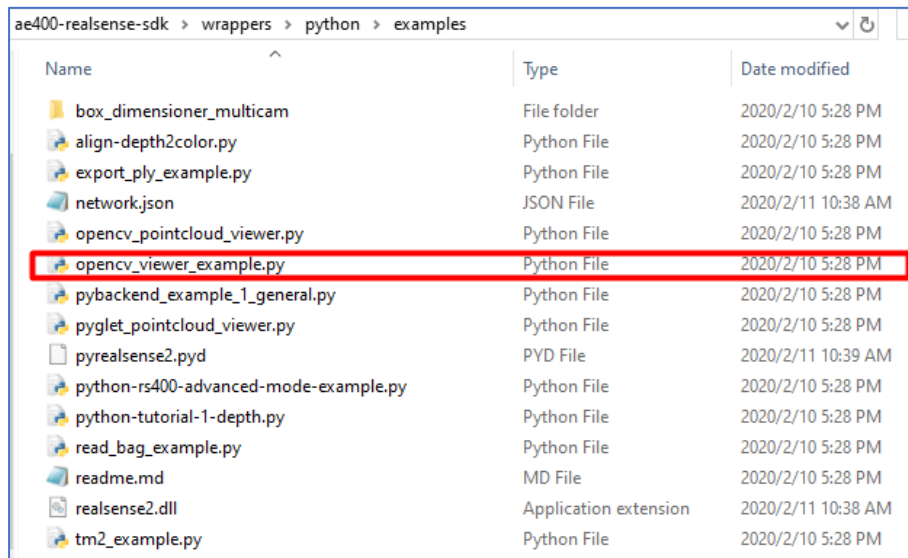


Figure 3-79

2. You can view depth / RGB images live.

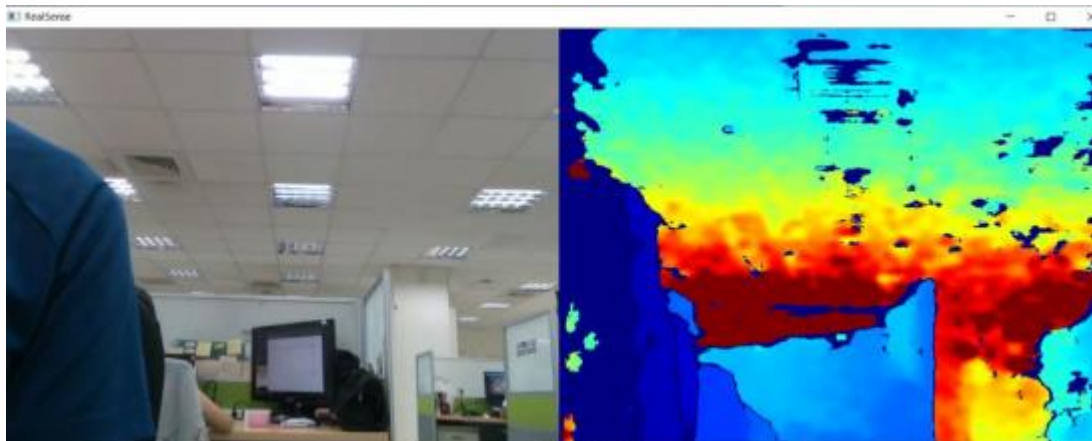


Figure 3-80

To exit the application, press **Ctrl + C**.

Mapping Depth / RGB Images

1. Click **align-depth2color.py**.

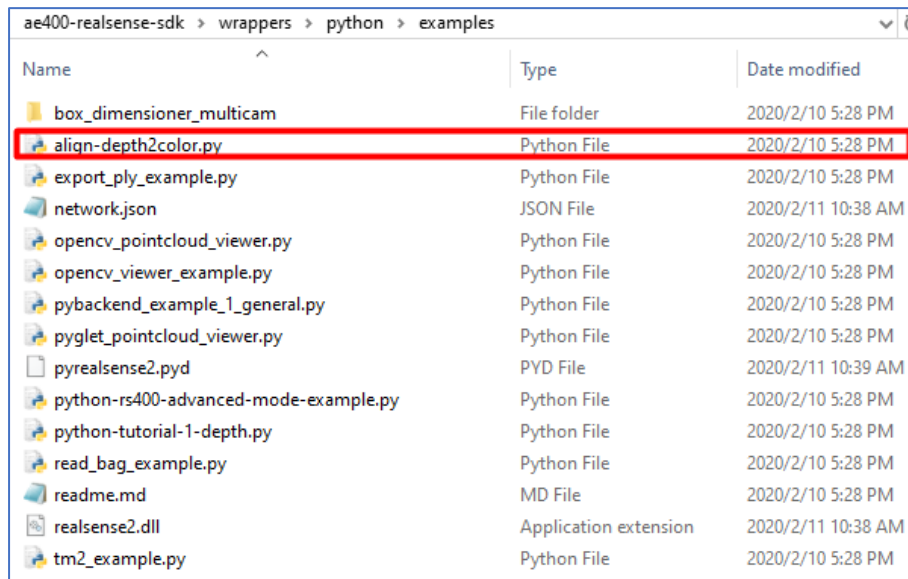


Figure 3-81

2. You can view depth / RGB images live.

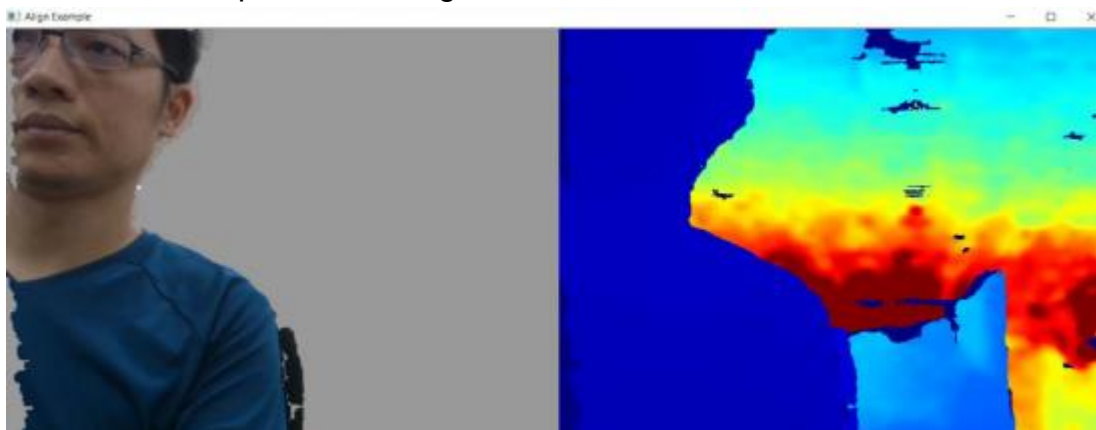


Figure 3-82

To exit the application, press **q** or **Esc**.

Viewing Point Cloud Images

1. Click `opencv_pointcloud_viewer.py`.

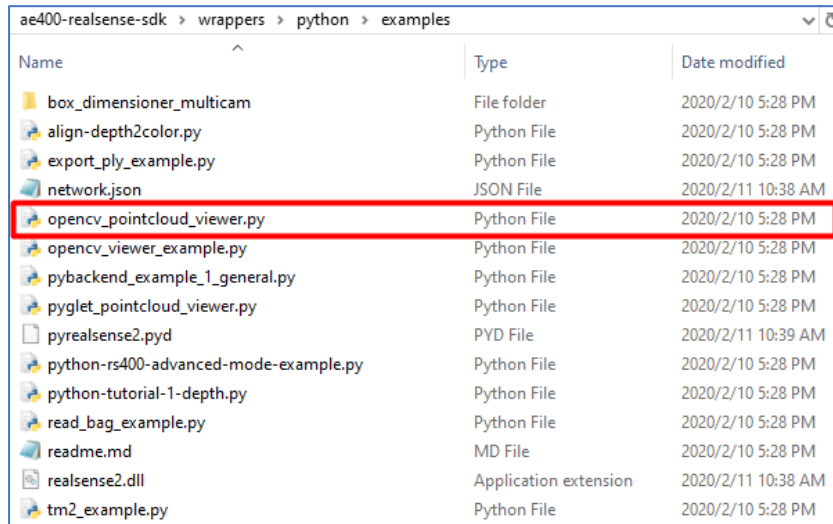


Figure 3-83

2. You can view the point cloud images live.

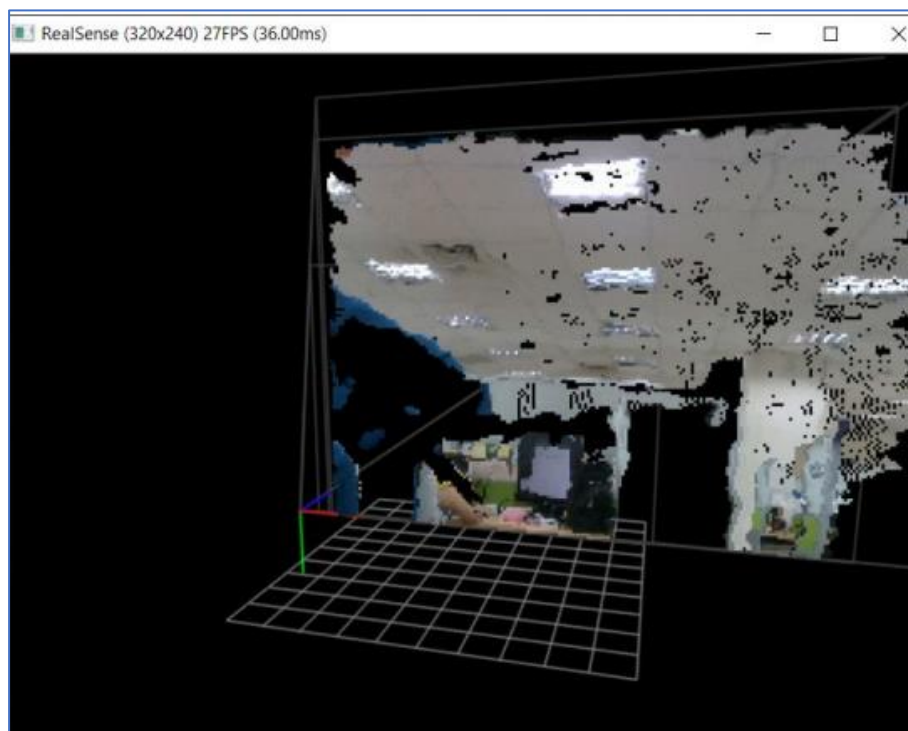


Figure 3-84

3. Use the hotkeys below for the following functions.

No.	Keys	Functions
1.	p	Pauses image display.
2.	r	Resets image.
3.	d	Cycles through decimation values.
4.	z	Toggles point scaling.
5.	c	Toggles color source.
6.	s	Save images as .OUT or .PNG.
7.	e	Exports points as .OUT or .PLY

To exit the application, press **q** or **Esc**.

4. Programming with HALCON

With the GenIcam interface wrapper, you can access the camera through **HALCON HDevelop**, the industrial standardized integrated development environment. HALCON HDevelop supports the display of imaging outcome alongside the related codes, thereby facilitating the 3D vision project development without compromising high-robustness and time-to-market.

Currently, LIPSedge™ AE400 SDK with GenICam only supports **LOADING** pre-defined parameters (.json file) into HALCON HDevelop. Individual code parameter modification is **NOT** supported. LIPSedge™ AE400 SDK with GenICam supports the Windows platform **ONLY**.

4.1 LIPSedge™ AE400 SDK with GenICam

Installation & Configuration

You can obtain LIPSedge™ AE400 SDK with GenICam at <https://filebox.lips-hci.com/index.php/s/QKbr5044amsnxTb> or by scanning the QR Code below:



Run the installation file contained in the LIPSedge™ AE400 SDK with GenICam for installation.

1. Extract the downloaded file and run **LIPS_AE400_SDK_GenTL-win64-0.9.0.7.exe**.

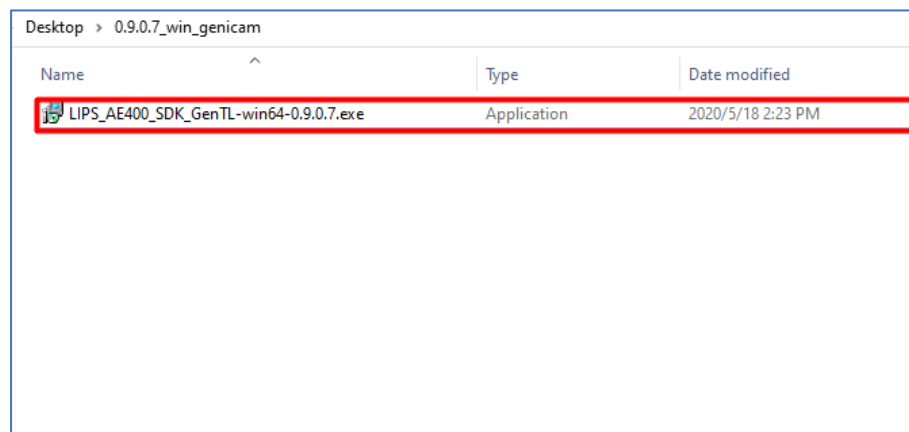


Figure 4-1

2. Click **Next**.



Figure 4-2

3. Grant your consent and click **Next**.

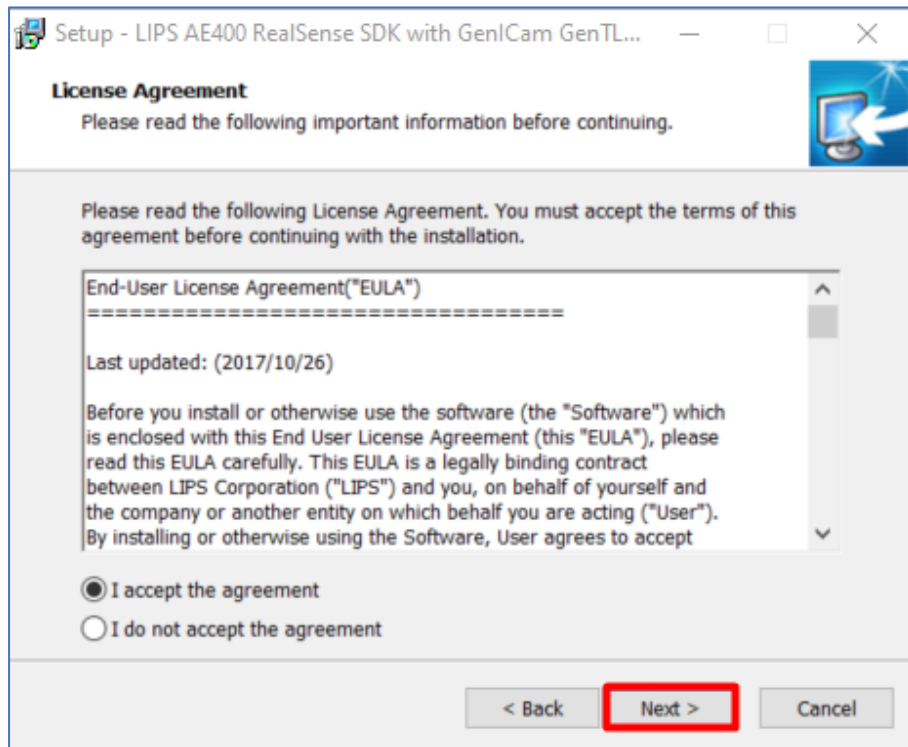


Figure 4-3

4. Click **Next**.

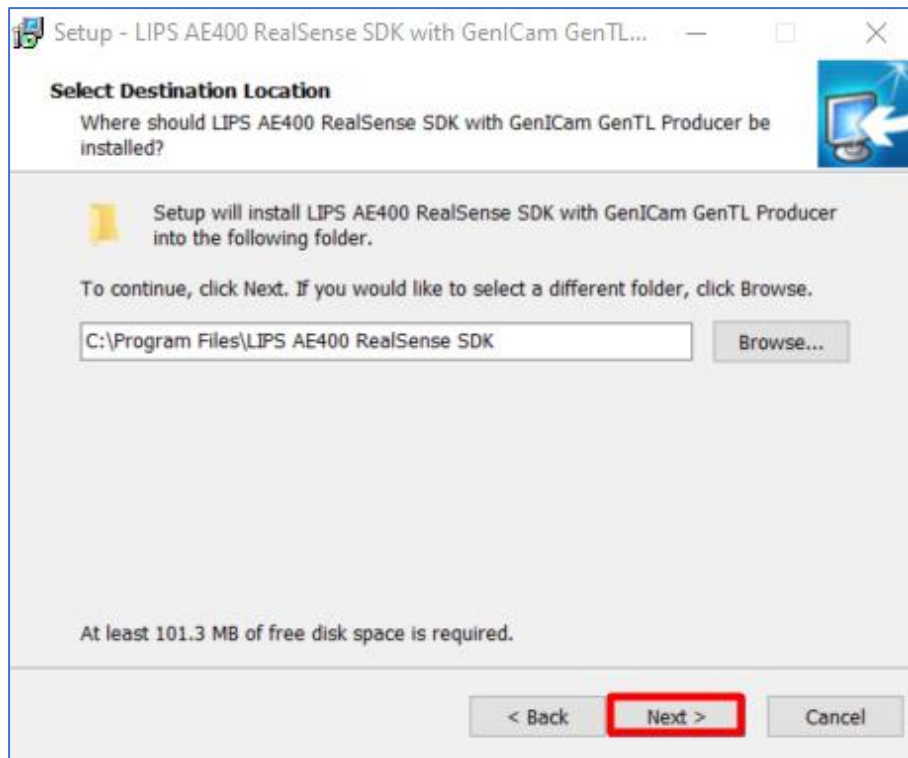


Figure 4-4

5. Click **Install**.

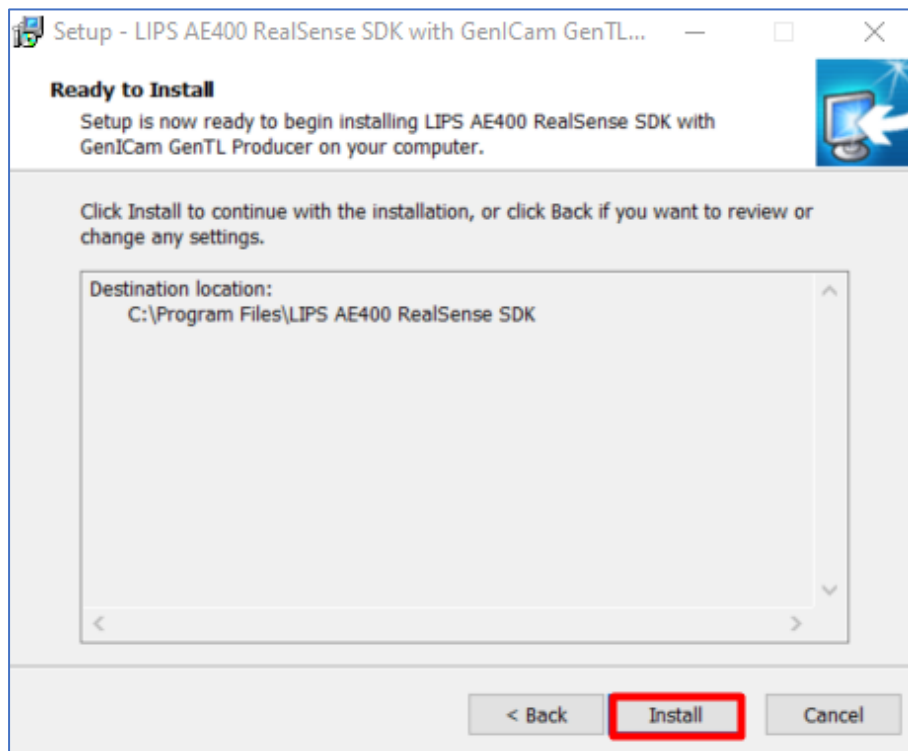


Figure 4-5

6. Click **Finish**.

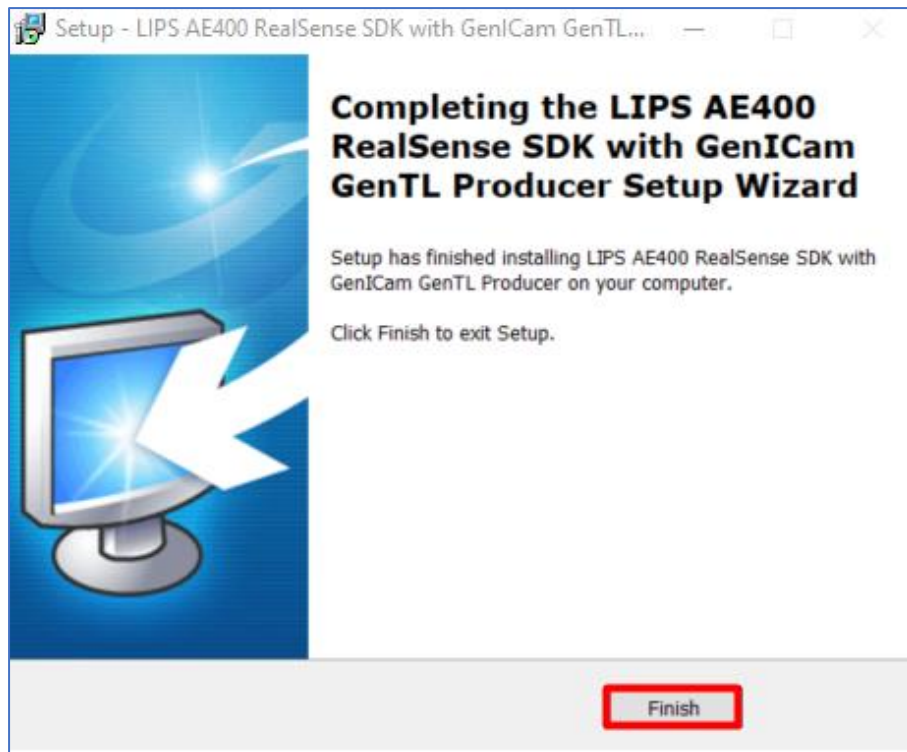


Figure 4-6

4.2 HALCON Registration & Installation

Before installing HALCON, you need to prepare the following items from your HALCON distributor:

- HALCON license file
- HALCON USB dongle for HDevelop

Once the license file and dongle are obtained, complete the registration and download HALCON at **MVTec official website**:

<https://www.mvtec.com/login/?referer=download%2Fhalcon%2F>.

LIPSedge™ AE400 camera supports the following **HALCON version**:

- **Product Name:** HALCON
- **Version:** 19.11 Progress
- **Platform:** Windows

1. On **MVTec official website**, click **Not registered yet?**

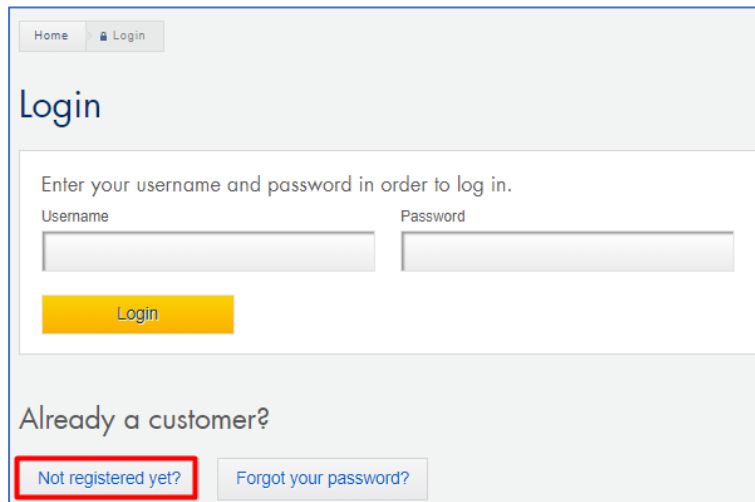


Figure 4-7

- Fill in the registration form. Select **HALCON** under the **Interested Product** section and click **Register**.

* Country: Taiwan Your country is not available? Please [contact us!](#)

* Phone: 0966153869

* E-mail: AlexHon@lips-hci.com

Which of our products are you interested in?

* Product(s): HALCON MERLIC
 Embedded Vision Deep Learning Tool

Further information:

Comment:

Your password:

* Password:

* Confirm password:

Please use letters and numbers / special characters (min. 6 characters)

Newsletter: I want to receive occasional information about MVTec and its products

* Agreement: I hereby confirm that my intended use of MVTec's standard software products complies with applicable law, e.g., export regulations
 I accept the MVTec [Privacy Policy](#)

Fields with * are mandatory.

Figure 4-8

3. Go to **your e-mail inbox** and activate your account.

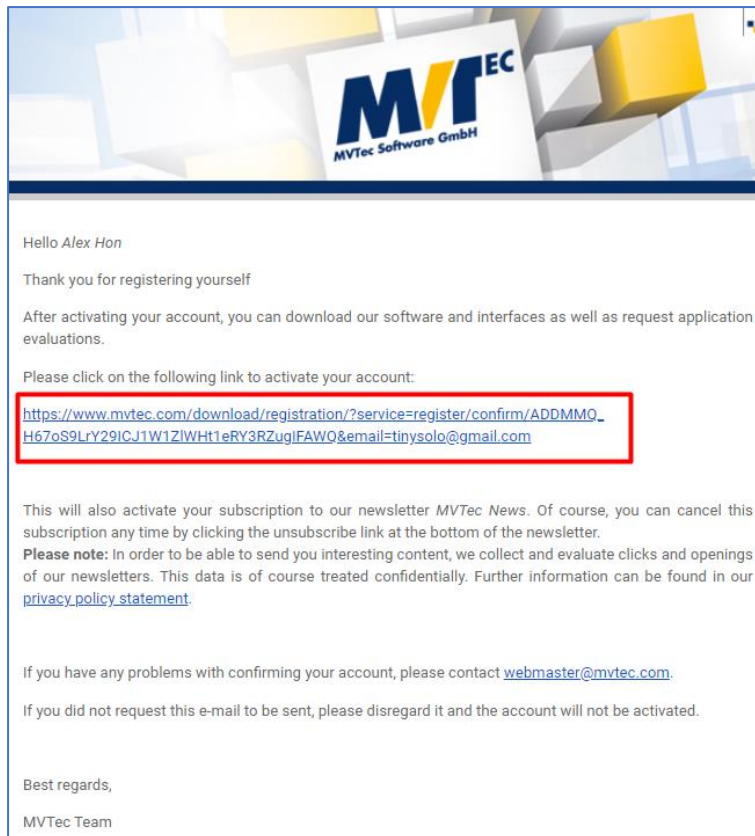


Figure 4-9

4. Log onto MVtec official website.

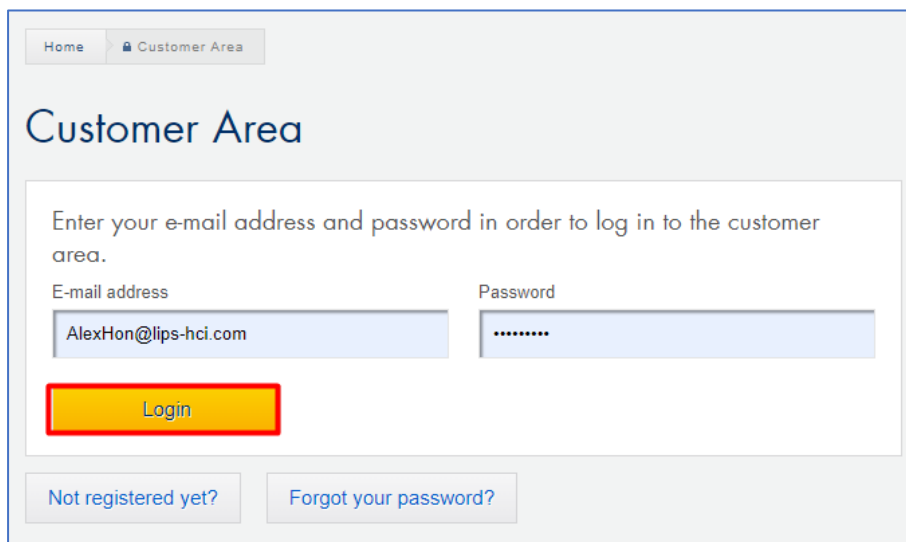
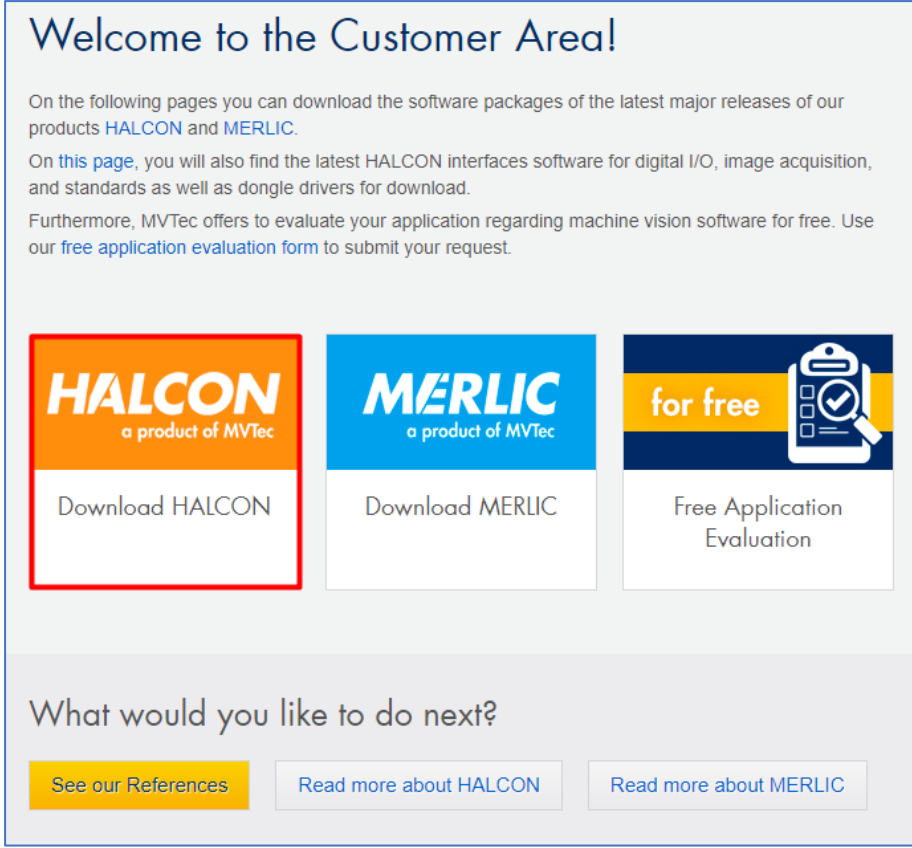


Figure 4-10

5. Click **Download HALCON**.



Welcome to the Customer Area!

On the following pages you can download the software packages of the latest major releases of our products **HALCON** and **MERLIC**.

On [this page](#), you will also find the latest HALCON interfaces software for digital I/O, image acquisition, and standards as well as dongle drivers for download.

Furthermore, MVTec offers to evaluate your application regarding machine vision software for free. Use our [free application evaluation form](#) to submit your request.

HALCON
a product of MVTec
Download HALCON

MERLIC
a product of MVTec
Download MERLIC

for free
Free Application Evaluation

What would you like to do next?

[See our References](#) [Read more about HALCON](#) [Read more about MERLIC](#)

Figure 4-11

- Select the HALCON version according to your platform and click **Download Full Version [2 GB]** to save the installation file to your PC / laptop.

Product
HALCON

Version
19.11 Progress

Operating System
Windows

To run HALCON you need a license. To get such a license, please contact your [local distributor](#).

If you are looking for other versions and security packs, your [local distributor](#) will help you as well.

If you are using Internet Explorer 11.0.9 please note additionally: this browser is preventing files with the file extension ".exe" from being downloaded correctly, as these file extensions are cut off. To install our products ideally, please add the file extension ".exe" manually to the respective product file, or use a different browser.

Please note:

- To use HALCON's *deep learning functionalities*, you need to download the Deep Learning Add-on(s) **and** the installer for HALCON Full Version resp. HALCON Runtime Version.
- Starting with HALCON 19.11, the Windows version requires two Deep Learning Add-ons:** 'core' (libraries and pretrained nets) and 'data' (images and nets used by the example programs).

Please choose

HALCON 19.11.0.0 Deep Learning Core - Additional Core Functionality Required for Deep Learning (x86sse2-win32 / x64-win64), Add-on to Full/Runtime Version Windows

Download Add-on to Full/Runtime Version [1 GB]

Read Release Notes

Installation guide (pdf)

HALCON 19.11.0.0 Deep Learning Data - Data Required for Deep Learning Example Programs (x86sse2-win32 / x64-win64), Add-on to Full/Runtime Version Windows

Download Add-on to Full/Runtime Version [1 GB]

Read Release Notes

Installation guide (pdf)

HALCON 19.11.0.2 Full Version (x86sse2-win32 / x64-win64) Windows

Download Full Version [2 GB]

Read Release Notes

Installation guide (pdf)

HALCON 19.11.0.2 Runtime Version (x86sse2-win32 / x64-win64) Windows

Download Runtime Version [194 MB]

Read Release Notes

Installation guide (pdf)

Figure 4-12

7. Start the downloaded file and click **Next**.

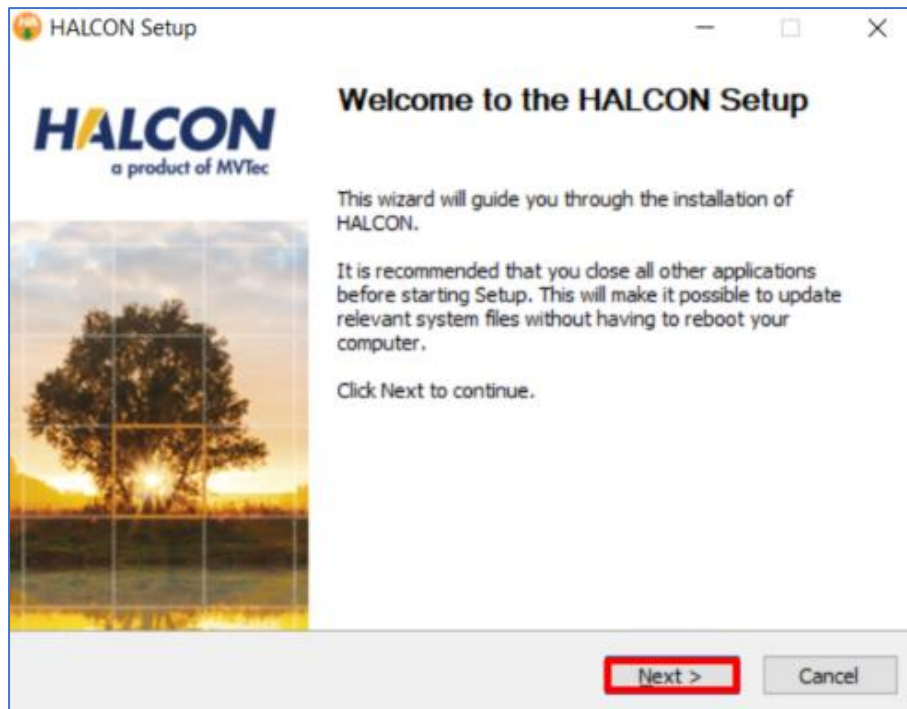


Figure 4-13

8. Consent to the license agreement and click **Next**.



Figure 4-14

9. Click **Next**.

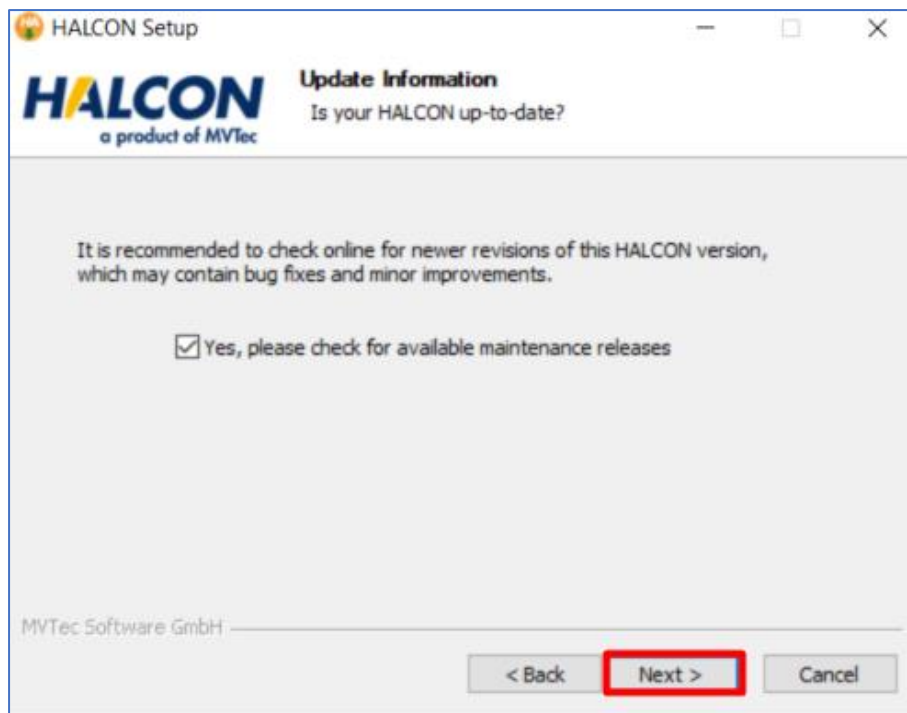


Figure 4-15

10. Click **Next**.

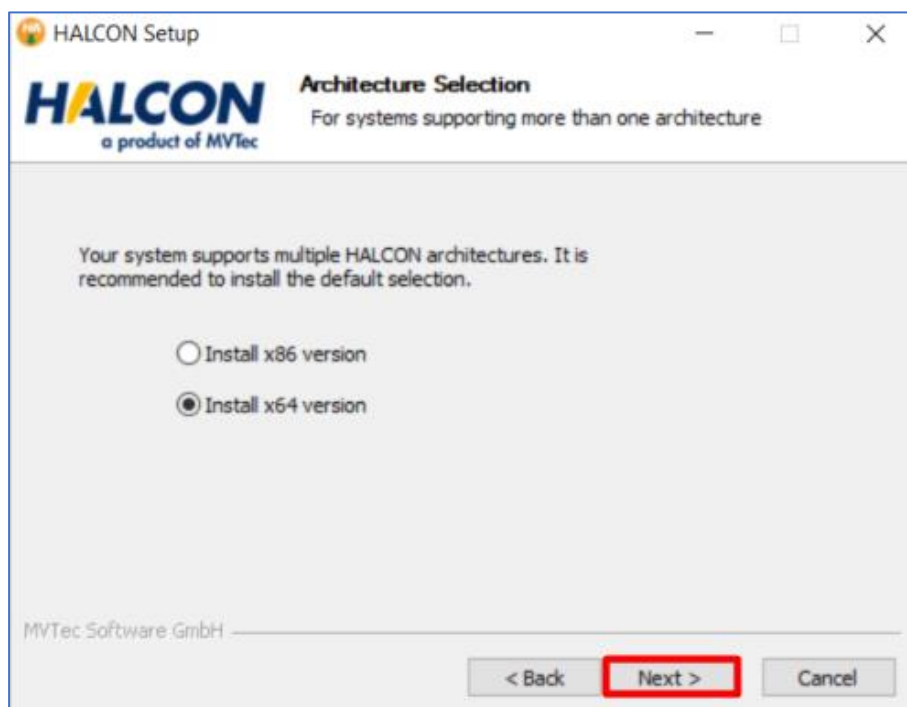


Figure 4-16

11. Click **Next**.

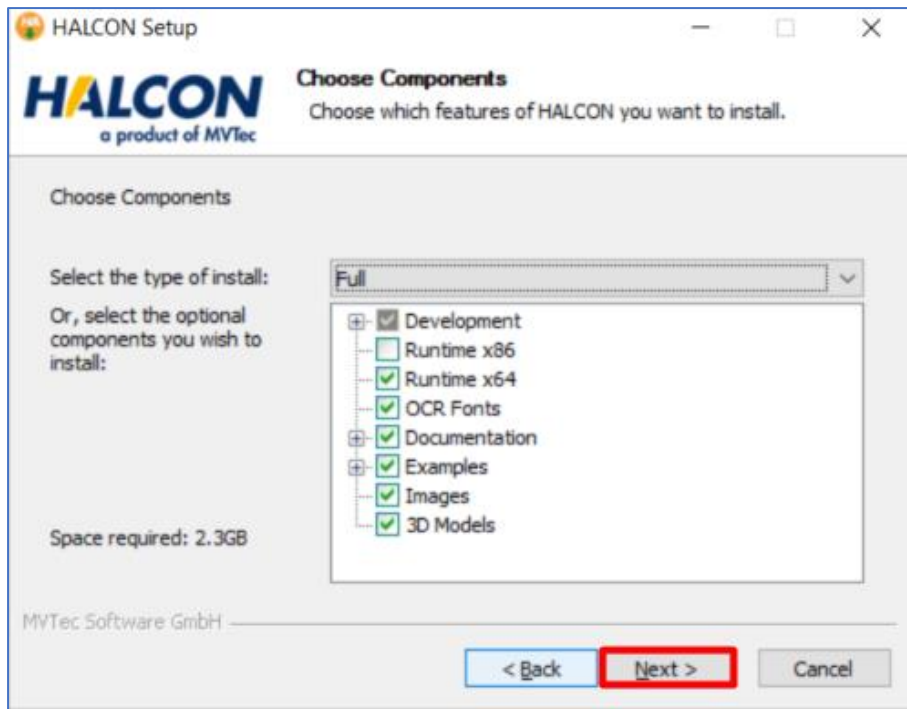


Figure 4-17

12. Click **Next**.

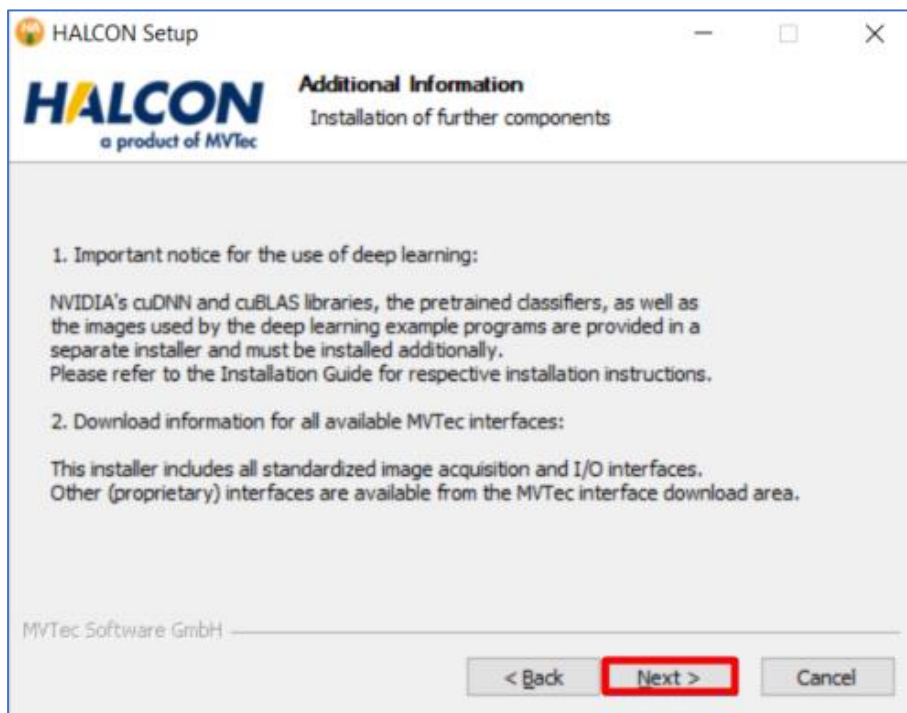


Figure 4-18

13. Click **Next**.

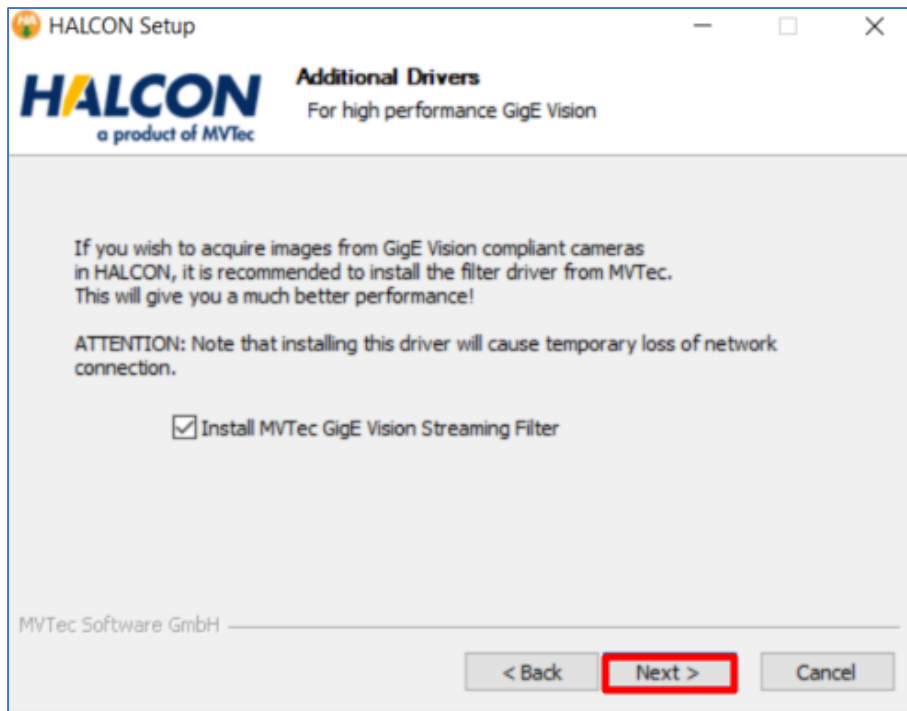


Figure 4-19

14. Select the preferred documentation language and click **Next**.

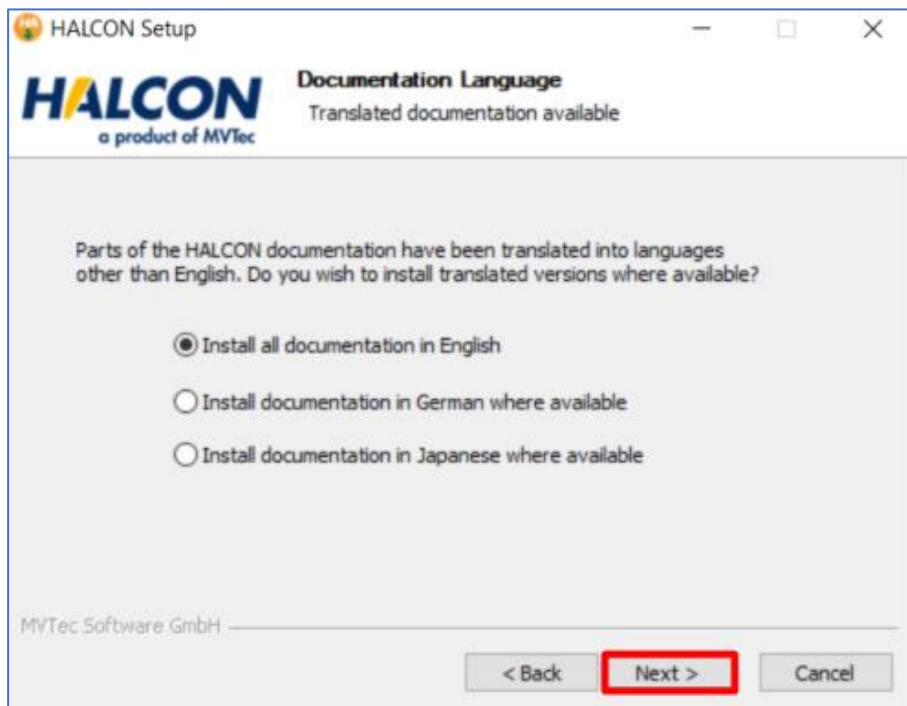


Figure 4-20

15. Click **Install**.

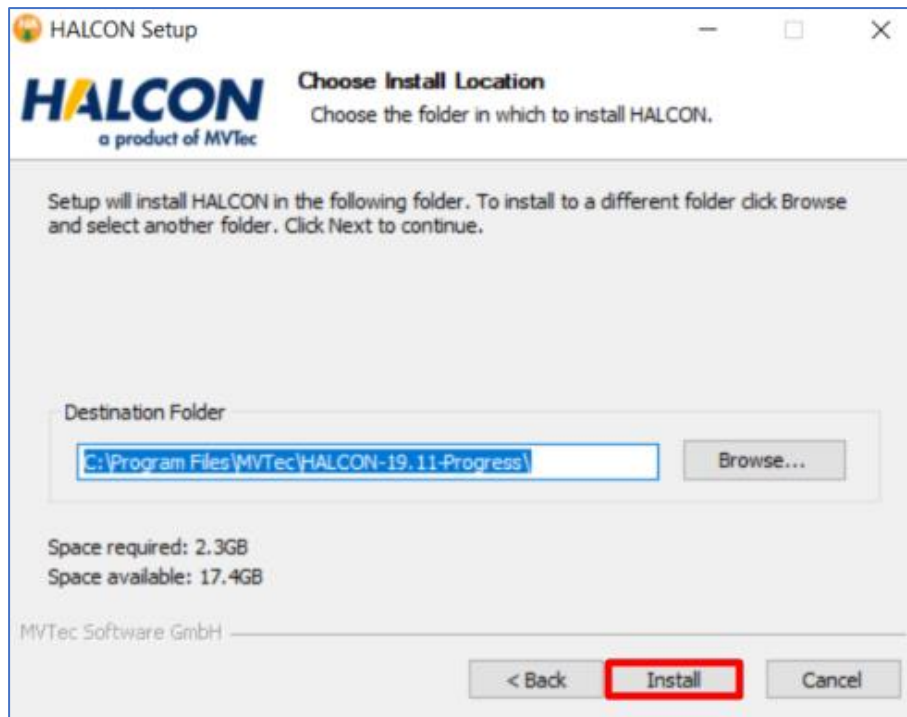


Figure 4-21

16. Specify the location of the license file and click **Next**.

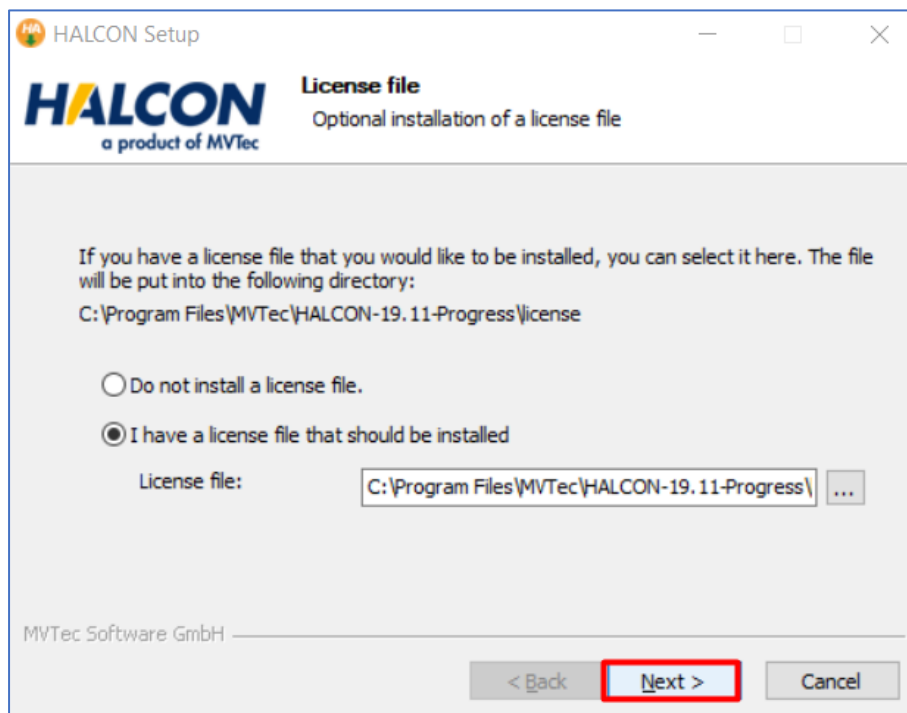


Figure 4-22

17. Click **Finish**.

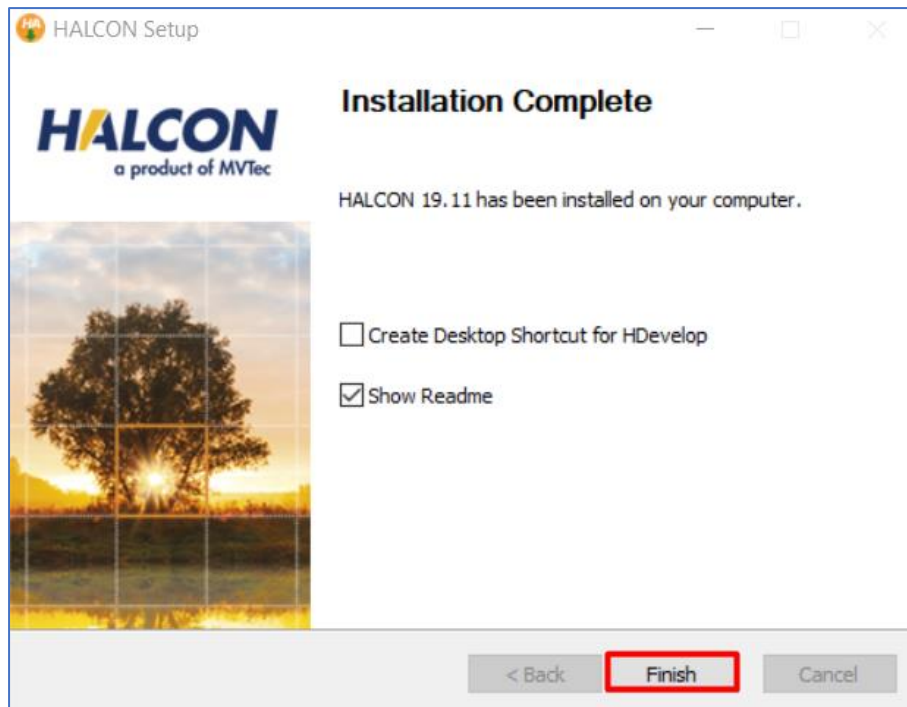


Figure 4-23

4.3 Accessing Camera Image with HALCON

Before activating **HALCON HDevelop**, make sure you have properly configured the environment variables for your PC/ laptop and obtained the **dongle** from HALCON. Otherwise, the application won't start.

Once the dongle is ready, follow the instructions below to access the LIPSedge™ AE400 camera image. HALCON HDevelop detects the camera **by its components separately**. For example, in HALCON HDevelop, the LIPSedge™ AE400 is listed as **Stereo Module / RGB camera** separately.

The following instructions focus on accessing the live view of LIPSedge™ AE400 camera. For other HALCON HDevelop functions, refer to https://www.mvtec.com/fileadmin/Redaktion/mvtec.com/documentation/manuals/hdevelop_users_guide.pdf.

1. Plug the dongle onto your PC/ laptop.



Figure 4-24

2. Start **HDevelop**. On the top bar, click **Assistant** and select **Open New Image Acquisition**.

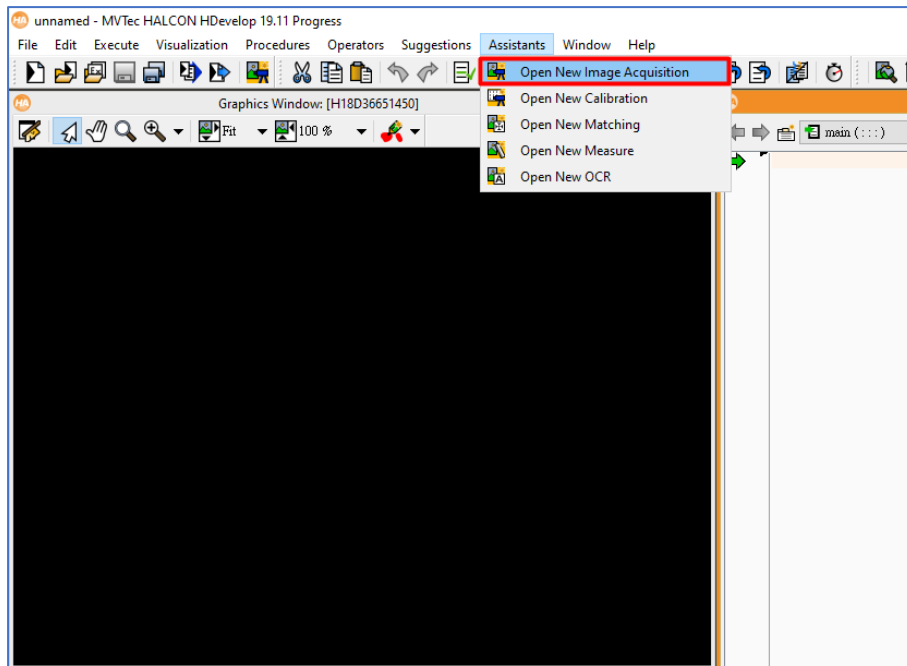


Figure 4-25

3. From the top tabs, click **Source** and select **GenIcamTL** under the **Image Acquisition Interface** drop-down list.

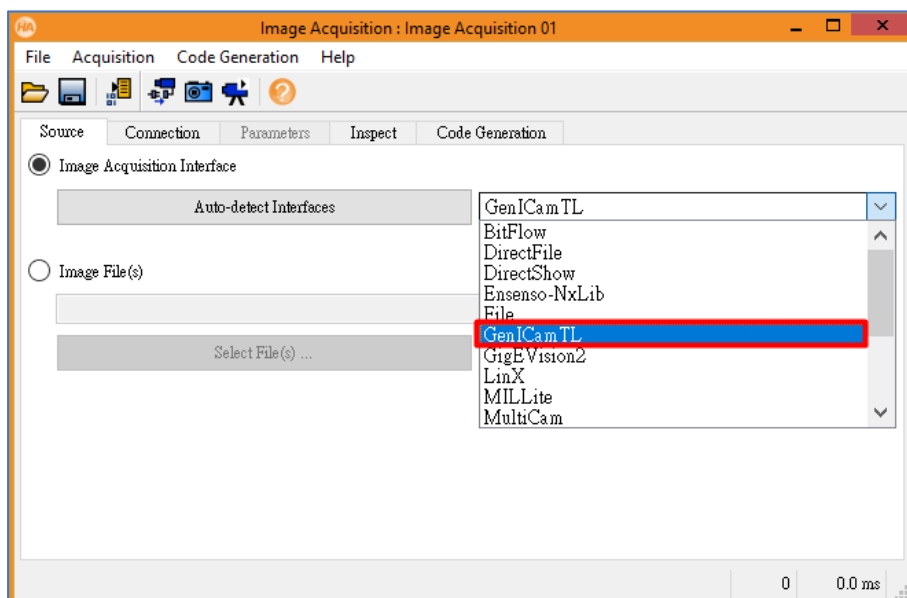


Figure 4-26

Note: HALCON HDevelop takes a while to detect supported interfaces. Do **NOT** make any changes or terminate the application while loading.

- From the top tabs, click **Connection** and select your LIPSedge™ AE400 camera from the **Device** drop-down list.

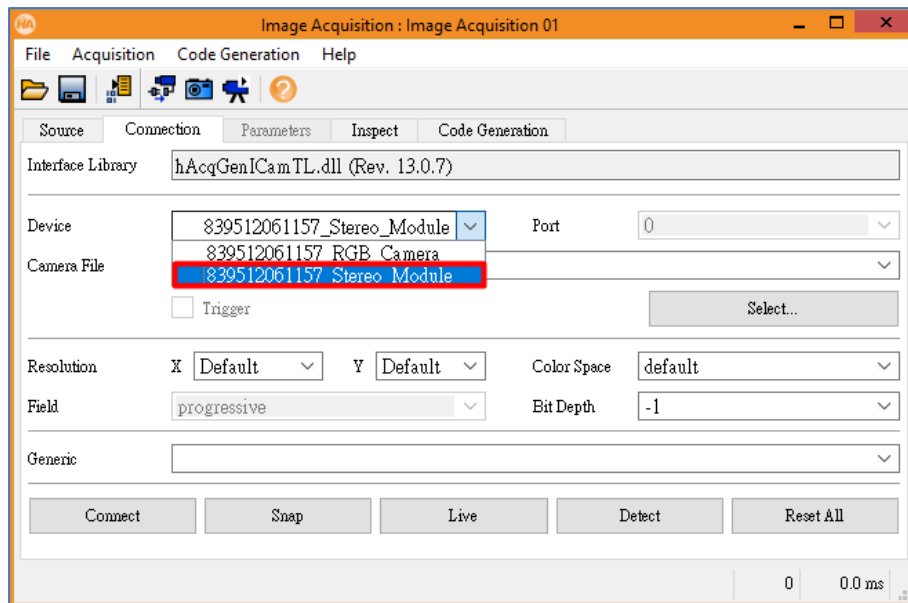


Figure 4-27

- Click **Live**.

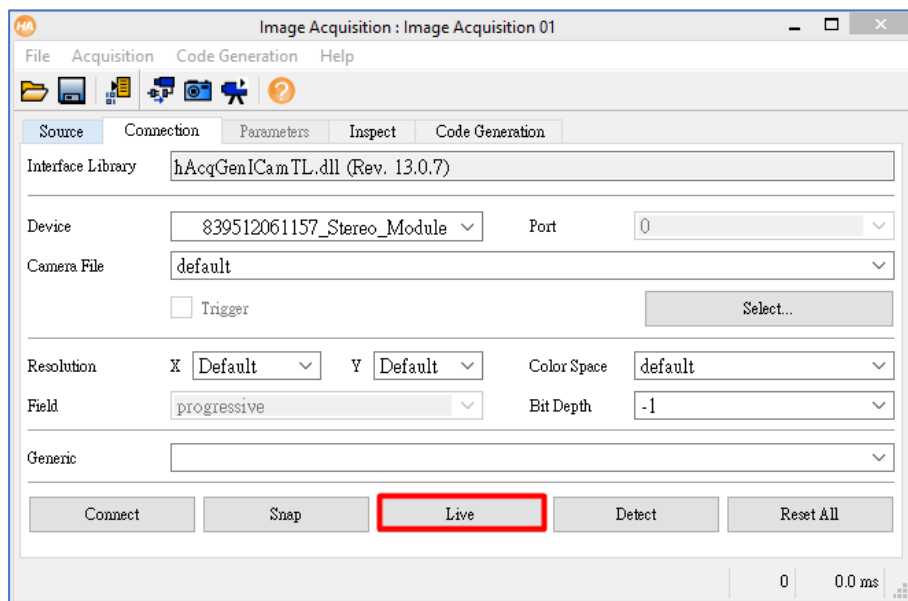


Figure 4-28

6. You can view the camera image on live.

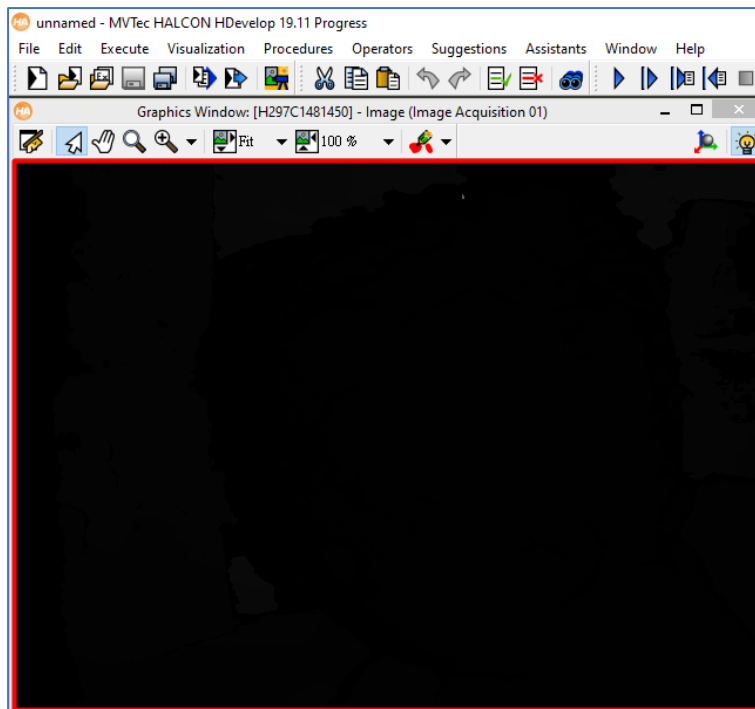


Figure 4-29

7. By default, HALCON HDevelop displays the camera image with low visibility. To improve the image visibility, right-click the **Graphics Window**, go to **Lut** and select **cubic_root**.

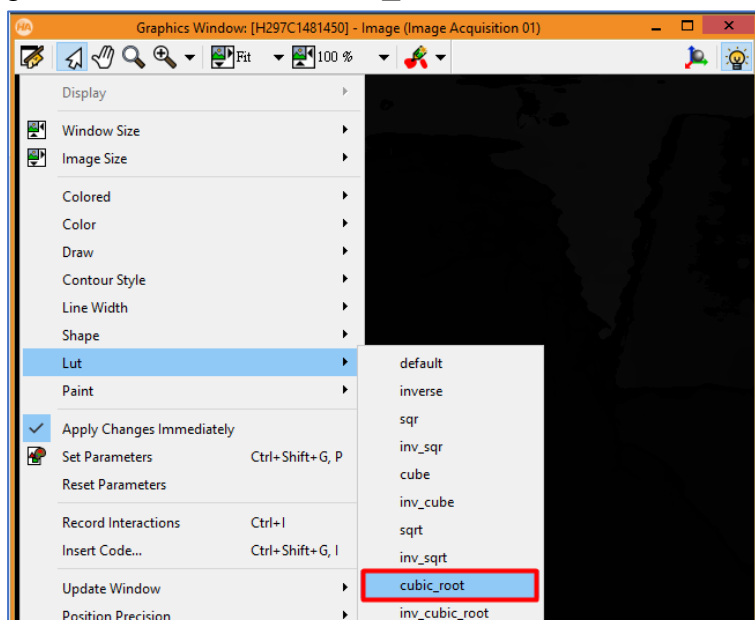


Figure 4-30

8. The image visibility is improved.

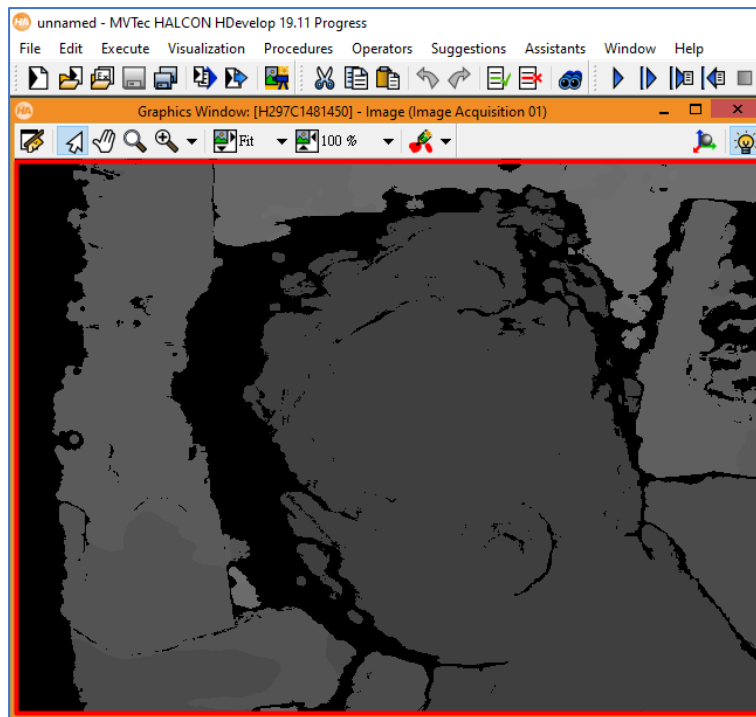


Figure 4-31

4.4 Parameters Loading and Image Acquisition

LIPSedge™ AE400 SDK with GenICam allows you to load predefined image parameters, recorded as the format of .json file, to HALCON HDevelop to have multiple display modes. You can load your own .json file or use predefined presets provided by Intel® RealSense™ through <https://github.com/IntelRealSense/librealsense/wiki/D400-Series-Visual-Presets>.

LIPSedge™ AE400 SDK with GenICam supports changing the parameters **ONLY** through loading .json files. Change parameters in the program window is **NOT** supported.

1. Insert the dongle and start **HDevelop**. On the top bar, click **Assistant** and select **Open New Image Acquisition**. For details, refer to *Step 1~2. In 4.3 Accessing Camera Image with HALCON*.
2. On the **Image Acquisition** window, click **Connect**.

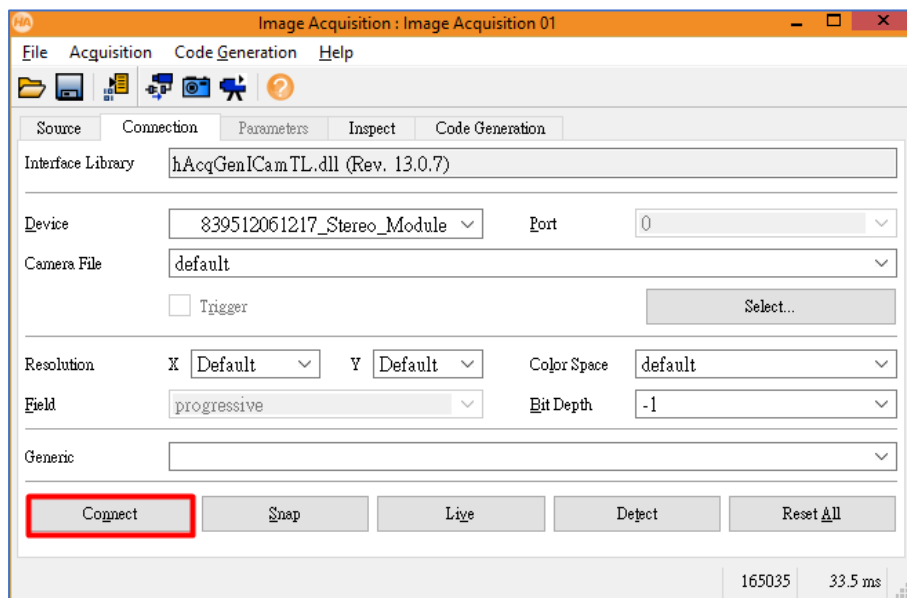


Figure 4-32

3. On the top Bar, select **Parameters**.

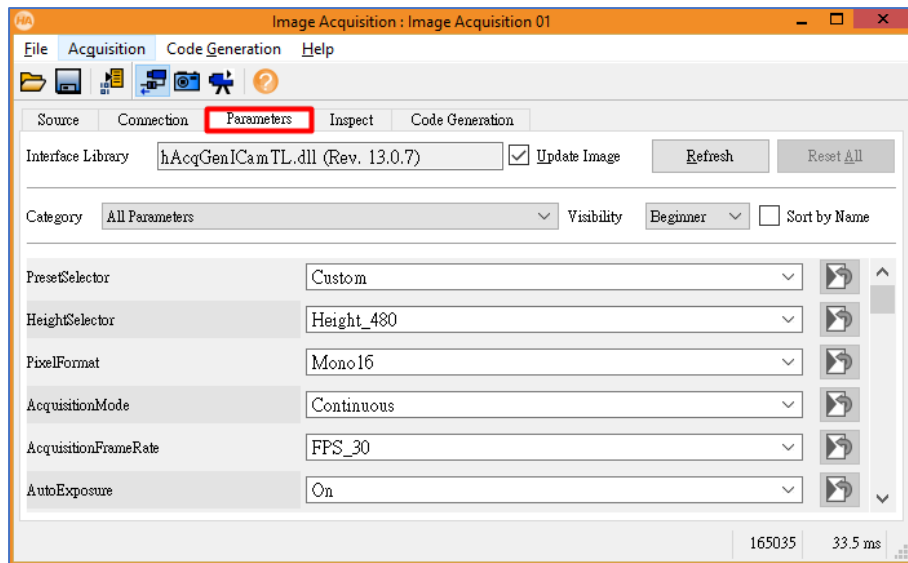


Figure 4-33

4. From the **Visibility** drop down list, select **Guru**.

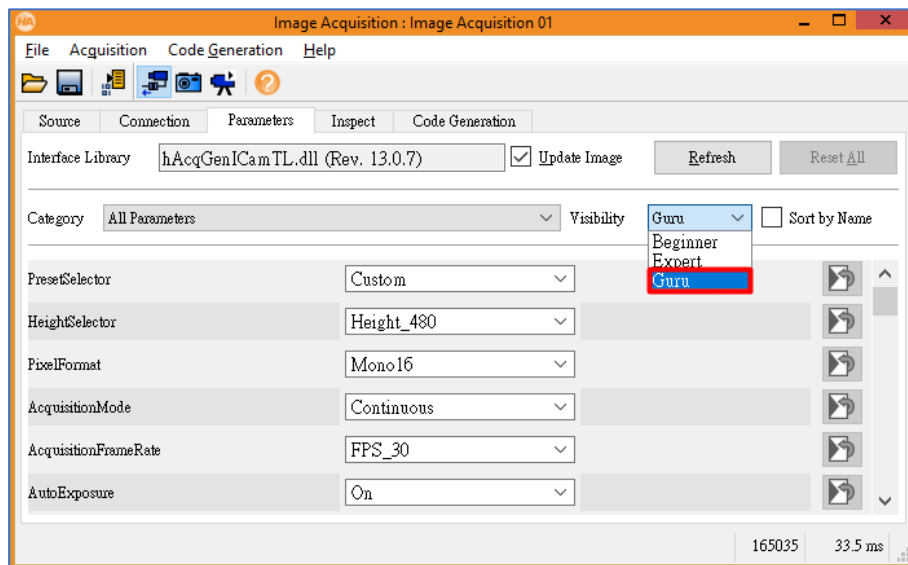


Figure 4-34

5. Type the location of the predefined parameters (.json file) into the **JSONFilepath** field.

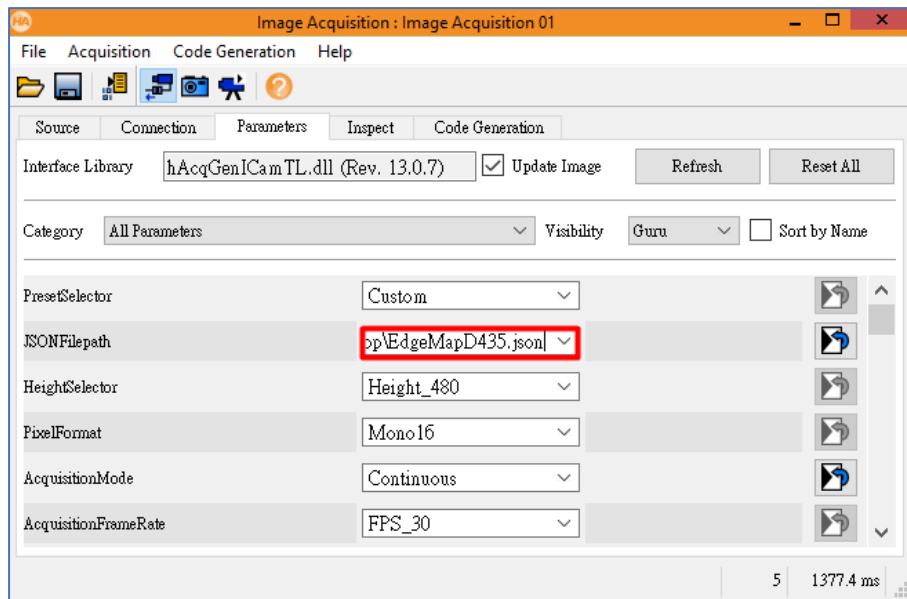


Figure 4-35

6. Scroll to the **Action parameters** section. Under **LoadJSON**, click **Apply**.

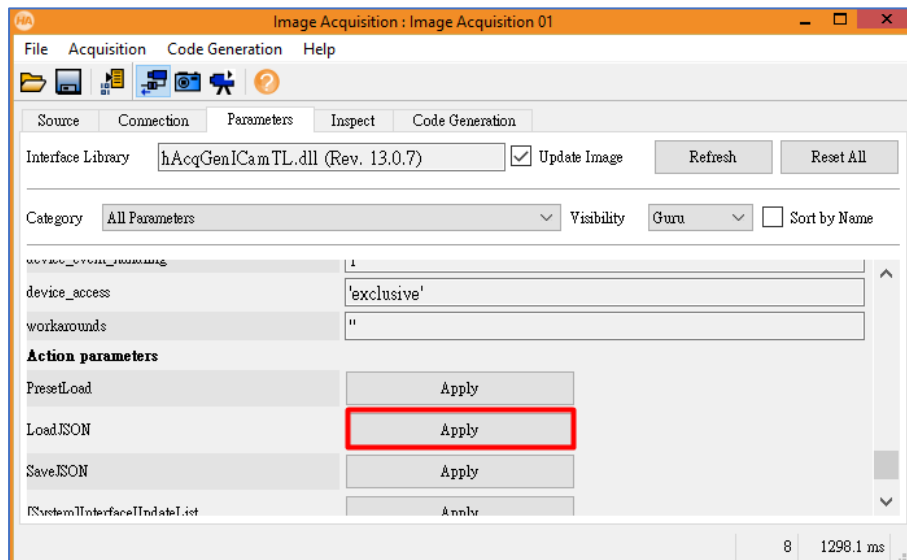


Figure 4-36

- On the top bar, go to **Connection** and click **Live** to view live image with the loaded parameters.

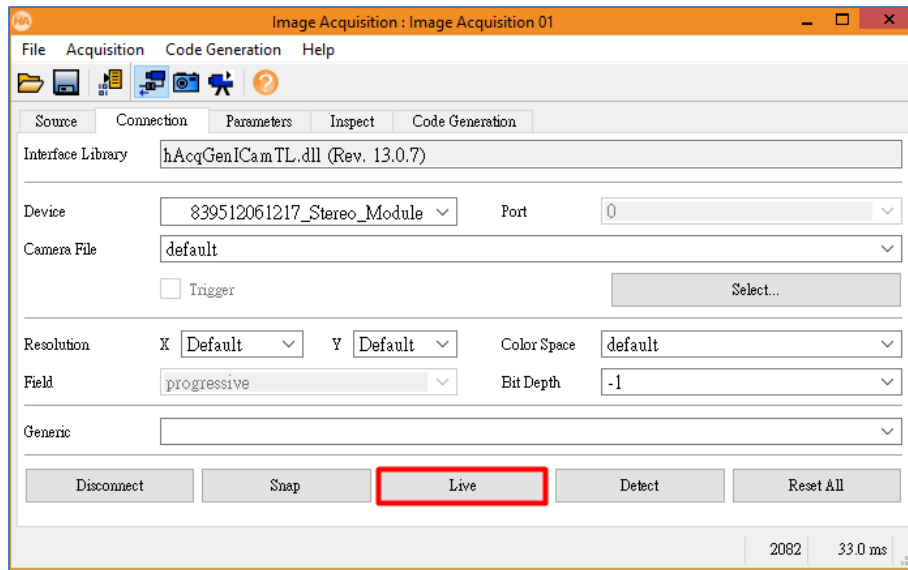


Figure 4-37

- Optionally right-click the **Graphics Window** and select **Lut > cubic_root** for better image visibility.

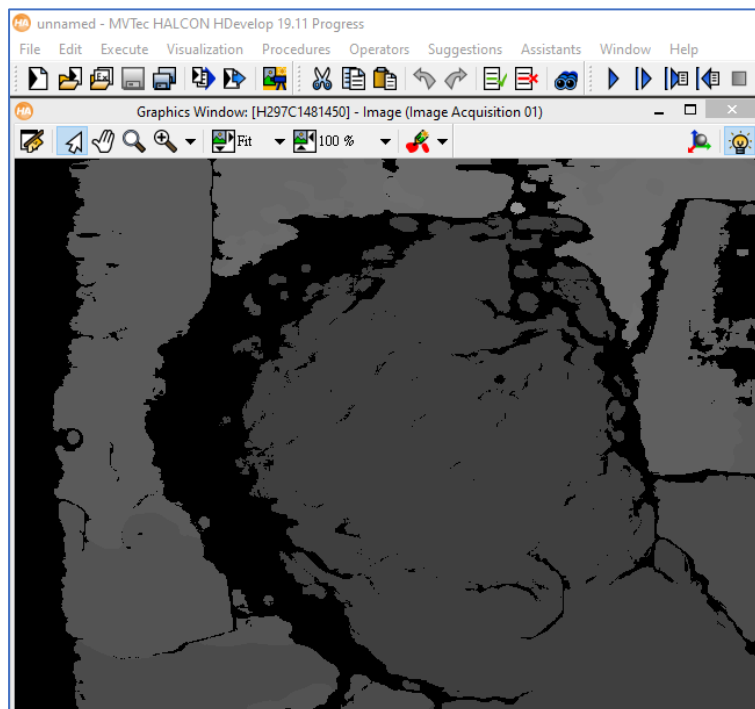


Figure 4-38

- To acquire the current image, go to Code Generation and click Insert Code.

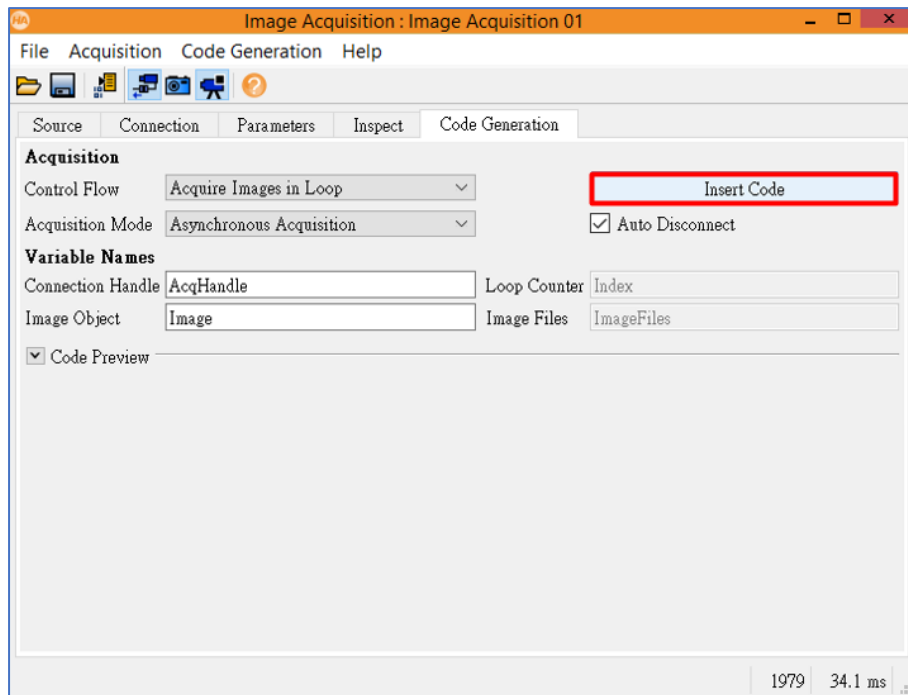


Figure 4-39

- The image acquisition code appears.

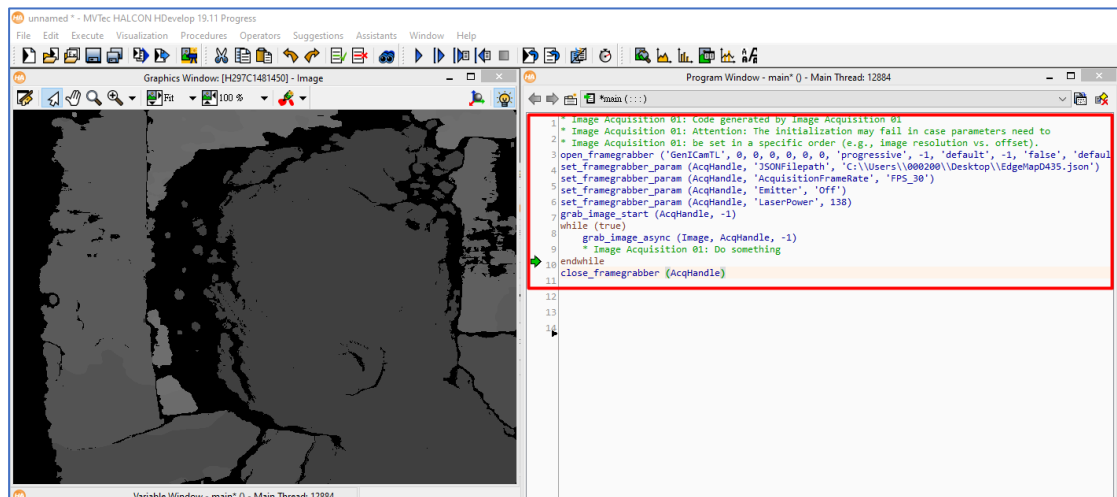



Figure 4-40

11. Click  to acquire the current image.

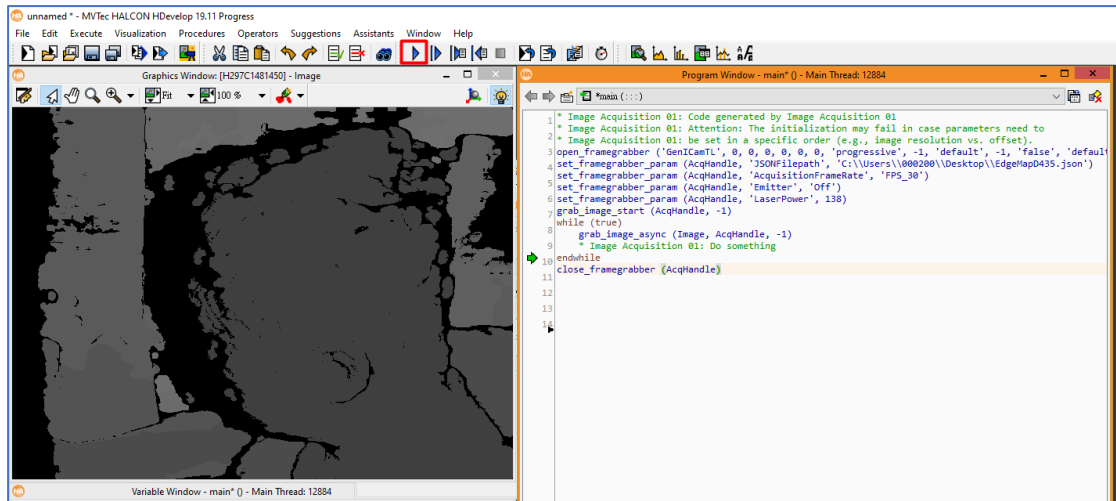


Figure 4-41

Linux

5 Prerequisites

LIPS Corp. provides the source code of the LIPSedge™ AE400 SDK on <https://github.com/lips-hci/ae400-realsense-sdk>. We recommend the following application / compiler for building LIPSedge™ AE400 SDK on Linux system

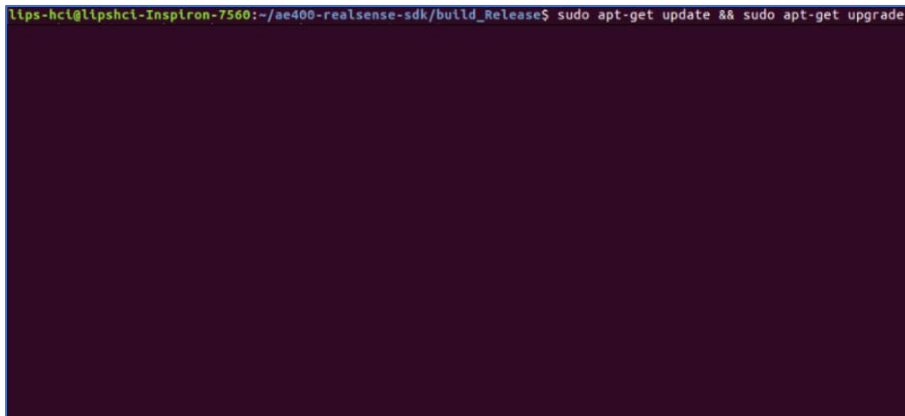
- **APT tool:** Install software packages such as Python and relevant libraries.
- **Core Backend Packages:** Enable secure internet / USB data transmission and GUI / graphics rendering.
- **Git Tools:** Download and manage source code packages on LIP Corp.'s Github or Intel® RealSense™'s Github.
- **CMake V3.8 or above:** Generate Makefiles necessary for building applications from source code.

5.1 Installing APT Tool

Ubuntu requires APT tools for efficient package installation / management. Make sure your APT tools are up to date before building LIPSedge™ AE400 SDK from source code.

1. Type the following command.

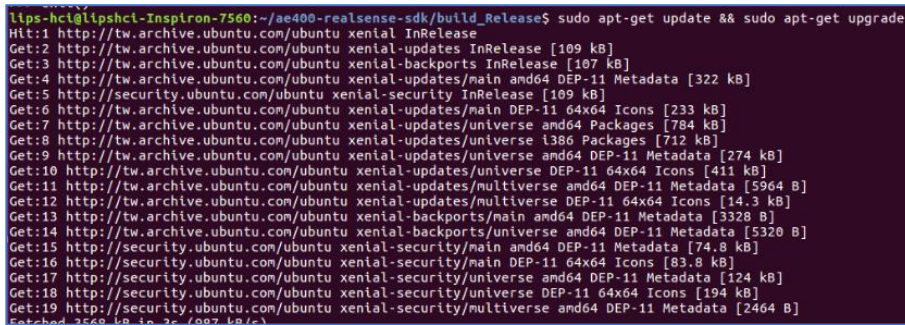
```
sudo apt-get update && sudo apt-get upgrade
```



```
l1ps-hcl@l1pshcl-Inspiron-7560:~/ae400-realsense-sdk/build_Release$ sudo apt-get update && sudo apt-get upgrade
```

Figure 5-1

2. The update is completed.



```
l1ps-hcl@l1pshcl-Inspiron-7560:~/ae400-realsense-sdk/build_Release$ sudo apt-get update && sudo apt-get upgrade
Hit:1 http://tw.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://tw.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:3 http://tw.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Get:4 http://tw.archive.ubuntu.com/ubuntu xenial-updates/main amd64 DEP-11 Metadata [322 kB]
Get:5 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Get:6 http://tw.archive.ubuntu.com/ubuntu xenial-updates/main DEP-11 64x64 Icons [233 kB]
Get:7 http://tw.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 Packages [784 kB]
Get:8 http://tw.archive.ubuntu.com/ubuntu xenial-updates/universe i386 Packages [712 kB]
Get:9 http://tw.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 DEP-11 Metadata [274 kB]
Get:10 http://tw.archive.ubuntu.com/ubuntu xenial-updates/universe DEP-11 64x64 Icons [411 kB]
Get:11 http://tw.archive.ubuntu.com/ubuntu xenial-updates/multiverse amd64 DEP-11 Metadata [5964 B]
Get:12 http://tw.archive.ubuntu.com/ubuntu xenial-updates/multiverse DEP-11 64x64 Icons [14.3 kB]
Get:13 http://tw.archive.ubuntu.com/ubuntu xenial-backports/main amd64 DEP-11 Metadata [3328 B]
Get:14 http://tw.archive.ubuntu.com/ubuntu xenial-backports/universe amd64 DEP-11 Metadata [5320 B]
Get:15 http://security.ubuntu.com/ubuntu xenial-security/main amd64 DEP-11 Metadata [74.8 kB]
Get:16 http://security.ubuntu.com/ubuntu xenial-security/main DEP-11 64x64 Icons [83.8 kB]
Get:17 http://security.ubuntu.com/ubuntu xenial-security/universe amd64 DEP-11 Metadata [124 kB]
Get:18 http://security.ubuntu.com/ubuntu xenial-security/universe DEP-11 64x64 Icons [194 kB]
Get:19 http://security.ubuntu.com/ubuntu xenial-security/multiverse amd64 DEP-11 Metadata [2464 B]
Fetched 3568 kB in 3s (997 kB/s)
```

Figure 5-2

5.2 Configuring Backend Environments

LIPSedge™ AE400 SDK requires several **librealsense** libraries in the build process to support internet / USB data transmission and GUI / graphics rendering. Make sure your PC / laptop is equipped with the following libraries:

- **libssl-dev**
- **libusb**
- **libgtk-3-dev**
- **libglfw3-dev (Ubuntu 16 LTS only)**
- **libglfw3-dev (Ubuntu 18 LTS only)**
- **libgl1-mesa-dev (Ubuntu 18 LTS only)**
- **libglu1-mesa-dev (Ubuntu 18 LTS only)**

1. Start the **Terminal** and type the following command.

```
sudo apt-get install libssl-dev libusb-1.0-0-dev pkg-config libgtk-3-dev
```

```
lips-hci@lipshci-Inspiron-7560:~$ sudo apt-get install libssl-dev libusb-1.0-0-dev pkg-config libgtk-3-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'libusb-1.0-0-dev' for regex 'libusb-1.0-0-dev'
pkg-config is already the newest version (0.29.1-0ubuntu1).
The following package was automatically installed and is no longer required:
  snapd-login-service
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  autotools-dev debhelper dh-strip-nondeterminism libatk-bridge2.0-dev
  libatk1.0-dev libatspi2.0-dev libcairo-script-interpreter2 libcairo2-dev
  libdbus-1-dev libdrm-dev libegl1-mesa-dev libepoxy-dev libexpat1-dev
  libfile-stripnondeterminism-perl libfontconfig1-dev libfreetype6-dev
  libgdk-pixbuf2.0-dev libgl1-mesa-dev libgmp-dev libgmpxx4ldbl libharfbuzz-dev
  libharfbuzz-gobject0 libice-dev libmail-sendmail-perl libmirclient-dev
  libmircommon-dev libmircookie-dev libmircookie2 libmircore-dev
  libpango1.0-dev libpcre3-dev libpcre3-2-3 libpcrecpp0v5 libpixman-1-dev
  libpng12-dev libprotobuf-dev libpthread-stubs0-dev libsm-dev libssl-doc
  libsys-hostname-long-perl libusb-1.0-doc libwayland-bin libwayland-dev
  libx11-dev libx11-doc libx11-xcb-dev libxau-dev libxcb-dri2-0-dev
  libxcb-dri3-dev libxcb-glx0-dev libxcb-present-dev libxcb-randr0-dev
  libxcb-render0-dev libxcb-shape0-dev libxcb-shm0-dev libxcb-sync-dev
  libxcb-xfixes0-dev libxcb1-dev libxcomposite-dev libxcursor-dev
  libxdamage-dev libxdmcp-dev libxext-dev libxfixes-dev libxft-dev libxi-dev
  libxinerama-dev libxkbcommon-dev libxrandr-dev libxrender-dev
  libxshmfence-dev libxtst-dev libxxf86vm-dev nettle-dev po-debconf
  x11proto-composite-dev x11proto-core-dev x11proto-damage-dev
  x11proto-dri2-dev x11proto-fixes-dev x11proto-gl-dev x11proto-input-dev
```

Figure 5-3

2. Install **OpenGL** libraries. Here, we use **Ubuntu 16.04 LTS** as an example

For **Ubuntu 16 LTS**:

```
sudo apt-get install libglfw3-dev
```

For **Ubuntu 18 LTS**:

```
sudo apt-get install libglfw3-dev libgl1-mesa-dev libglu1-mesa-dev
```

```
lips-hc@lipshc-insprion-7560:~$ sudo apt-get install libglfw3-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  snapd-login-service
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libgl1-mesa-dev libglfw3 libglu1-mesa-dev mesa-common-dev
The following NEW packages will be installed:
  libgl1-mesa-dev libglfw3 libglfw3-dev libglu1-mesa-dev mesa-common-dev
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 814 kB of archives.
After this operation, 5029 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://tw.archive.ubuntu.com/ubuntu xenial/universe amd64 libglfw3 amd64 3.1.2-3 [38.5 kB]
Get:2 http://tw.archive.ubuntu.com/ubuntu xenial-updates/main amd64 mesa-common-dev amd64 18.0.5-0ubuntu0-16.04.1 [543 kB]
Get:3 http://tw.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libgl1-mesa-dev amd64 18.0.5-0ubuntu0-16.04.1 [4456 B]
Get:4 http://tw.archive.ubuntu.com/ubuntu xenial/main amd64 libglu1-mesa-dev amd64 9.0.0-2.1 [202 kB]
Get:5 http://tw.archive.ubuntu.com/ubuntu xenial/universe amd64 libglfw3-dev amd64 3.1.2-3 [26.3 kB]
Fetched 814 kB in 0s (1839 kB/s)
Selecting previously unselected package libglfw3:amd64.
(Reading database ... 219234 files and directories currently installed.)
Preparing to unpack .../libglfw3_3.1.2-3_amd64.deb ...
Unpacking libglfw3:amd64 (3.1.2-3) ...
Selecting previously unselected package mesa-common-dev:amd64.
Preparing to unpack .../mesa-common-dev_18.0.5-0ubuntu0-16.04.1_amd64.deb ...
Unpacking mesa-common-dev:amd64 (18.0.5-0ubuntu0-16.04.1) ...
Selecting previously unselected package libgl1-mesa-dev:amd64.
Preparing to unpack .../libgl1-mesa-dev_18.0.5-0ubuntu0-16.04.1_amd64.deb ...
Unpacking libgl1-mesa-dev:amd64 (18.0.5-0ubuntu0-16.04.1) ...
```

Figure 5-4

5.3 Installing Git Tool

LIPSegde™ AE400 SDK requires **Git Tool** for building the SDK through **command lines**.

1. Start the **Terminal** and type the following command.

```
sudo apt install git
```

```
lips-hct@lipshci-Inspiron-7560:~$ sudo apt install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  snapd-login-service
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk
  gitweb git-arch git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 3917 kB of archives.
```

Figure 5-5

5.4 Installing CMake

LIPSedge™ AE400 SDK requires **CMake** to generate Makefiles necessary for building the application.

1. Start the **Terminal** and type the following command.

```
sudo apt install cmake
```

```
lips-hci@lipshci-Inspiron-7560:~$ sudo apt install cmake
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  snapd-login-service
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  cmake-data libjsoncpp1
Suggested packages:
  codeblocks eclipse ninja-build
The following NEW packages will be installed:
  cmake cmake-data libjsoncpp1
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 3817 kB of archives.
After this operation, 29.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://tw.archive.ubuntu.com/ubuntu xenial-updates/main amd64 cmake-data all 3.5.1-1ubuntu3 [1121 kB]
Get:2 http://tw.archive.ubuntu.com/ubuntu xenial/main amd64 libjsoncpp1 amd64 1.7.2-1 [73.0 kB]
Get:3 http://tw.archive.ubuntu.com/ubuntu xenial-updates/main amd64 cmake amd64 3.5.1-1ubuntu3 [2623 kB]
Fetched 3817 kB in 0s (13.4 MB/s)
Selecting previously unselected package cmake-data.
```

Figure 5-6

6 Building LIPSedge™ AE400 SDK from Source Code

LIPS Corp. provides the source code package on <https://github.com/lips-hci/ae400-realsense-sdk> based on Intel® RealSense™ SDK. The SDK is comprehensively tested to be compatible with Intel® RealSense™ SDK v2.17. We recommend utilizing Intel® RealSense™ SDK v2.17 for application development. Once the required application / compilers are ready, follow the instructions to build the LIPSedge™ AE400 SDK.

8.1 Cloning LIPSedge™ AE400 SDK

1. Start **Terminal** and clone the source code of LIPSedge™ AE400 SDK to your local PC / laptop.

```
Git clone https://github.com/lips-hci/ae400-realsense-sdk.git ~/your  
destination
```



```
lips-hci@lipshci-Inspiron-7560:~$ git clone https://github.com/lips-hci/ae400-realsense-sdk.git
```

Figure 6-1

2. The cloning process is completed.

```
lips-hci@lips-hci-Inspiron-7560:~$ git clone https://github.com/lips-hci/ae400-realsense-sdk.git
Cloning into 'ae400-realsense-sdk'...
remote: Enumerating objects: 25, done.
remote: Counting objects: 100% (25/25), done.
remote: Compressing objects: 100% (24/24), done.
remote: Total 2547 (delta 12), reused 3 (delta 1), pack-reused 2522
Receiving objects: 100% (2547/2547), 50.71 MiB | 959.00 KiB/s, done.
Resolving deltas: 100% (614/614), done.
Checking connectivity... done.
```

Figure 6-2

8.2 Building LIPSedge™ AE400 SDK

Once the source code is cloned to the desired location in your PC /laptop, generate a Makefile through **CMake** and compile LIPSedge™ AE400 SDK into software packages according to the directory hierarchy assigned by the Makefile.

1. Go to LIPSedge™ AE400 SDK source code folder.

```
cd LIPS SDK source code folder
```



Figure 6-3

2. Create and go to the folder for building LIPSedge™ AE400 SDK.

```
mkdir build_Release
cd build_Release
```

```
lips-hci@lipshci-Inspiron-7560:~$ cd ae400-realsense-sdk/
lips-hci@lipshci-Inspiron-7560:~/ae400-realsense-sdk$ mkdir build_Release && cd build_Release
```

Figure 6-4

3. Generate the Makefile through CMake.

```
cmake .. -DCMAKE_BUILD_TYPE=Release
```

```
lips-hci@lipshci-Inspiron-7560:~$ cd ae400-realsense-sdk/
lips-hci@lipshci-Inspiron-7560:~/ae400-realsense-sdk$ mkdir build_Release && cd build_Release
lips-hci@lipshci-Inspiron-7560:~/ae400-realsense-sdk/build_Release$ cmake .. -DCMAKE_BUILD_TYPE=Release
-- The CXX compiler identification is GNU 5.4.0
-- The C compiler identification is GNU 5.4.0
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Info: REALSENSE_VERSION_STRING=2.17.1
-- Performing Test COMPILER_SUPPORTS_CXX11
-- Performing Test COMPILER_SUPPORTS_CXX11 - Success
-- Performing Test COMPILER_SUPPORTS_CXX0X
-- Performing Test COMPILER_SUPPORTS_CXX0X - Success
-- Setting Unix configurations
-- Checking internet connection...
-- Internet connection identified, enabling BUILD_WITH_TM2
```

Figure 6-5

4. Build the LIPSedge™ AE400 SDK.

make -j4

```
lips-hcl@lips-hcl-Inspiron-7560:~/ae400-realsense-sdk/build_Release$ make -j4
Scanning dependencies of target tm
Scanning dependencies of target realsense-file
[ 0%] Building CXX object third-party/libtm/libtm/src/CMakeFiles/tm.dir/UsbPlugListener.cpp.o
[ 1%] Building CXX object third-party/libtm/libtm/src/CMakeFiles/tm.dir/Device.cpp.o
[ 0%] Building CXX object third-party/libtm/libtm/src/CMakeFiles/tm.dir/Manager.cpp.o
[ 1%] Building C object third-party/realsense-file/CMakeFiles/realsense-file.dir/lz4/lz4.c.o
[ 2%] Building CXX object third-party/realsense-file/CMakeFiles/realsense-file.dir/rosbag/rosbag_storage/src/chunked_file.cpp.o
[ 3%] Building CXX object third-party/libtm/libtm/src/CMakeFiles/tm.dir/Common.cpp.o
[ 4%] Building CXX object third-party/libtm/libtm/src/CMakeFiles/tm.dir/infra/Log.cpp.o
[ 4%] Building CXX object third-party/realsense-file/CMakeFiles/realsense-file.dir/rosbag/rosbag_storage/src/lz4_stream.cpp.o
[ 5%] Building CXX object third-party/realsense-file/CMakeFiles/realsense-file.dir/rosbag/rosbag_storage/src/message_instance.cpp.o
[ 6%] Building CXX object third-party/realsense-file/CMakeFiles/realsense-file.dir/rosbag/rosbag_storage/src/bag_player.cpp.o
[ 6%] Building CXX object third-party/realsense-file/CMakeFiles/realsense-file.dir/rosbag/rosbag_storage/src/query.cpp.o
[ 7%] Building CXX object third-party/realsense-file/CMakeFiles/realsense-file.dir/rosbag/rosbag_storage/src/stream.cpp.o
[ 7%] Building CXX object third-party/realsense-file/CMakeFiles/realsense-file.dir/rosbag/rosbag_storage/src/uncompressed_stream.cpp.o
[ 8%] Building CXX object third-party/libtm/libtm/src/CMakeFiles/tm.dir/infra/Dispatcher.cpp.o
[ 9%] Building CXX object third-party/realsense-file/CMakeFiles/realsense-file.dir/rosbag/rosbag_storage/src/view.cpp.o
[ 9%] Building CXX object third-party/libtm/libtm/src/CMakeFiles/tm.dir/infra/Sm.cpp.o
[ 10%] Building CXX object third-party/libtm/libtm/src/CMakeFiles/tm.dir/infra/Utils.cpp.o
[ 10%] Building CXX object third-party/realsense-file/CMakeFiles/realsense-file.dir/rosbag/rosbag_storage/src/buffer.cpp.o
[ 10%] Building CXX object third-party/libtm/libtm/src/CMakeFiles/tm.dir/infra/Semaphore_ltn.cpp.o
[ 11%] Building CXX object third-party/libtm/libtm/src/CMakeFiles/tm.dir/infra/Event_ltn.cpp.o
[ 12%] Building CXX object third-party/realsense-file/CMakeFiles/realsense-file.dir/rosbag/rosbag_storage/src/bag.cpp.o
[ 12%]
```

Figure 6-6

5. Install the LIPSedge™ AE400 SDK built.

sudo make install

```
lips-hcl@lips-hcl-Inspiron-7560:~/ae400-realsense-sdk/build_Release$ sudo make install
[ 7%] Built target tm
[ 18%] Built target realsense-file
[ 55%] Built target realsense2
[ 56%] Built target rs-software-device
[ 57%] Built target rs-capture
[ 58%] Built target rs-callback
[ 59%] Built target rs-save-to-disk
[ 60%] Built target rs-multicam
[ 61%] Built target rs-pointcloud
[ 64%] Built target rs-align
[ 65%] Built target rs-sensor-control
[ 66%] Built target rs-measure
[ 67%] Built target rs-depth
[ 69%] Built target rs-color
[ 70%] Built target rs-distance
[ 73%] Built target rs-post-processing
[ 76%] Built target rs-record-playback
[ 77%] Built target lips_imu_reader
[ 79%] Built target rs-terminal
[ 83%] Built target rs-fw-logger
[ 84%] Built target rs-enumerate-devices
[ 89%] Built target realsense-viewer
[ 90%] Built target rs-data-collect
[ 95%] Built target rs-depth-quality
[ 98%] Built target rs-roscap-inspector
[ 99%] Built target rs-convert
[100%] Built target libtm_util
```

Figure 6-7

8.3 Patching Intel® RealSense™ SDK

By default, the LIPSedge™ AE400 camera connects to your applications through a Gigabit Ethernet interface, which is incompatible with native Intel® RealSense™ SDK applications.

LIPS Corp. provides a patch file to enable Ethernet connection with native Intel® RealSense™ SDK, thereby allowing developers to develop LIPSedge™ AE400-compatible applications on various Intel® RealSense™ SDK versions. However, Intel® RealSense™ SDK compatibility are not fully tested from version to version on the LIPSedge™ AE400 camera. Exercise the patches with caution. If error occurs, contact LIPS Corp. at info@lips-hci.com.

Follow the instructions below to apply patches for the Ethernet interface. Here, we use Intel® RealSense™ SDK with a **V2.29.0** Git branch as an example.

The patch is designed for developers who prefer native Intel® RealSense™ SDK. Prior to running the patch, make sure the Intel® RealSense™ SDK is installed on your PC / laptop. For details on Intel® RealSense™ SDK compatibility issues, refer to *IV, Support for native Intel® RealSense™ SDK, Notes for Programmers*.

1. Start **Terminal** and go to the **Intel® RealSense™ SDK** folder.

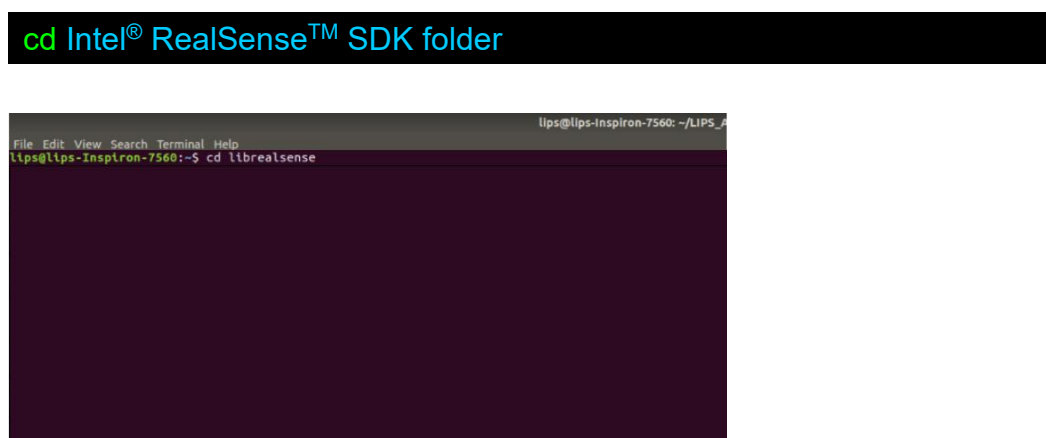
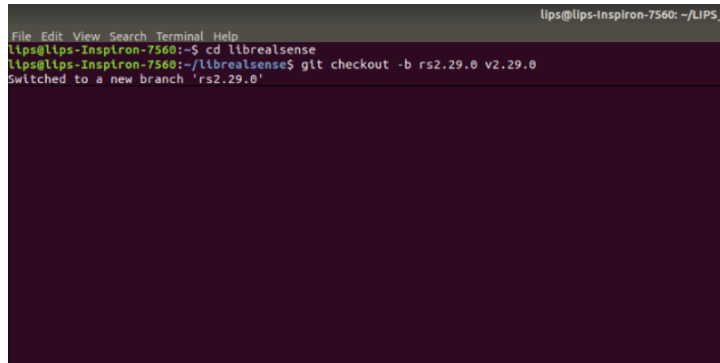


Figure 6-8

2. Update the Git branch on your local PC / laptop. Here, we use **v2.29.0** as an example.

```
git checkout -b rs2.29.0 v2.29.0
```

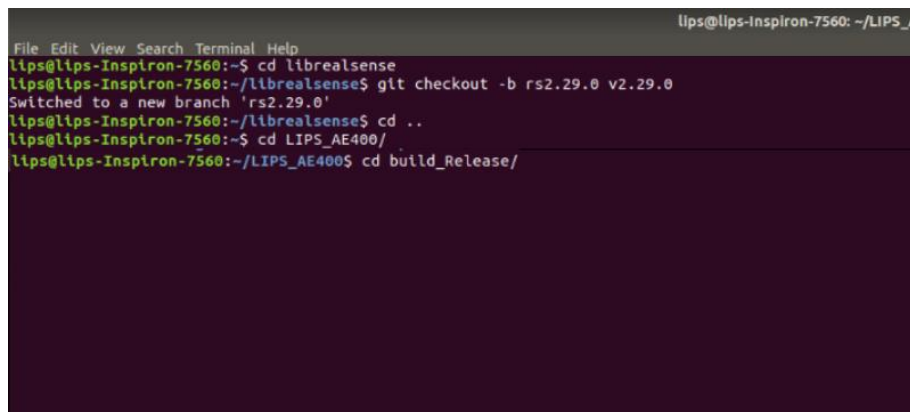


```
lips@lips-Inspiron-7560: ~/LIPS_
File Edit View Search Terminal Help
lips@lips-Inspiron-7560:~$ cd librealsense
lips@lips-Inspiron-7560:~/librealsense$ git checkout -b rs2.29.0 v2.29.0
Switched to a new branch 'rs2.29.0'
```

Figure 6-9

3. Go to the **LIPSedge™ AE400 SDK** folder

```
cd LIPSedge™ AE400 SDK
```

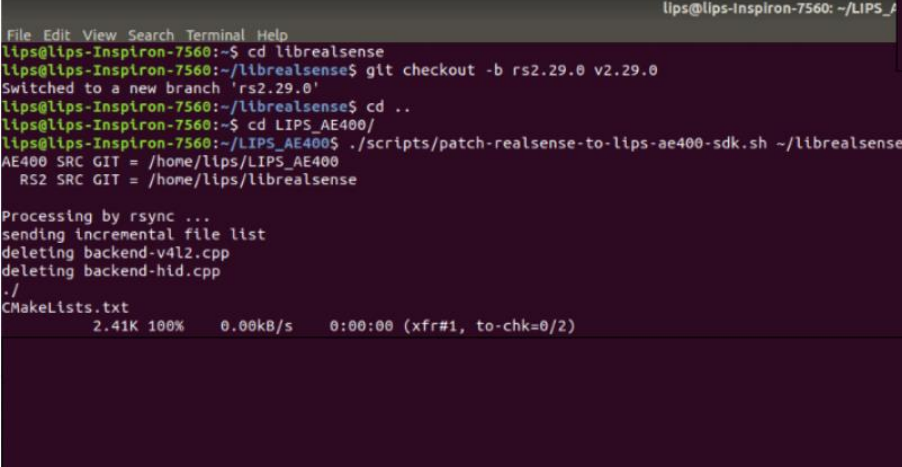


```
lips@lips-Inspiron-7560: ~/LIPS_
File Edit View Search Terminal Help
lips@lips-Inspiron-7560:~$ cd librealsense
lips@lips-Inspiron-7560:~/librealsense$ git checkout -b rs2.29.0 v2.29.0
Switched to a new branch 'rs2.29.0'
lips@lips-Inspiron-7560:~/librealsense$ cd ..
lips@lips-Inspiron-7560:~$ cd LIPS_AE400/
lips@lips-Inspiron-7560:~/LIPS_AE400$ cd build_release/
```

Figure 6-10

- Execute the patch file.

```
./scripts/patch-realsense-to-lips-ae400-sdk.sh ~/Intel® RealSense™
SDK folder
```



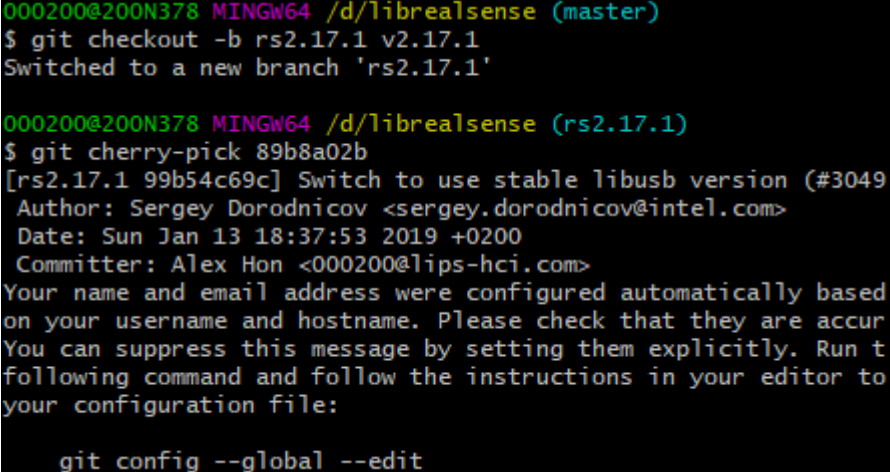
```
lips@lips-Inspiron-7560: ~/LIPS_#
File Edit View Search Terminal Help
lips@lips-Inspiron-7560:~$ cd librealsense
lips@lips-Inspiron-7560:~/librealsense$ git checkout -b rs2.29.0 v2.29.0
Switched to a new branch 'rs2.29.0'
lips@lips-Inspiron-7560:~/librealsense$ cd ..
lips@lips-Inspiron-7560:~$ cd LIPS_AE400/
lips@lips-Inspiron-7560:~/LIPS_AE400$ ./scripts/patch-realsense-to-lips-ae400-sdk.sh ~/librealsense
AE400 SRC GIT = /home/lips/LIPS_AE400
RS2 SRC GIT = /home/lips/librealsense

Processing by rsync ...
sending incremental file list
deleting backend-v4l2.cpp
deleting backend-hid.cpp
./
CMakeLists.txt
2.41K 100% 0.00kB/s 0:00:00 (xfr#1, to-chk=0/2)
```

Figure 6-11

Note: If you updated your local Git branch to **v2.17.1**, error may occur in the subsequent building process as the USB library (libusb) is unstable in **v2.17.1**. To fix the libusb issue, commit the **89b8a02b** fix to the branch.

```
git checkout -b rs2.17.1 v2.17.1
git cherry-pick 89b8a02b
```



```
000200@200N378 MINGW64 /d/librealsense (master)
$ git checkout -b rs2.17.1 v2.17.1
Switched to a new branch 'rs2.17.1'

000200@200N378 MINGW64 /d/librealsense (rs2.17.1)
$ git cherry-pick 89b8a02b
[rs2.17.1 99b54c69c] Switch to use stable libusb version (#3049
Author: Sergey Dorodnicov <sergey.dorodnicov@intel.com>
Date: Sun Jan 13 18:37:53 2019 +0200
Committer: Alex Hon <000200@lips-hci.com>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accur
You can suppress this message by setting them explicitly. Run t
following command and follow the instructions in your editor to
your configuration file:

git config --global --edit
```

9. Programming with Wrapper

LIPSedge™ AE400 is based on Intel® RealSense™ SDK, which supports various third-party technologies, such as Python wrapper. When building the SDK, specify the **build configuration** to enable the wrapper and run the **CMake build** again to generate the binaries and libraries.

For detail on supported Intel® RealSense™ SDK wrappers, refer to <https://github.com/IntelRealSense/librealsense/tree/master/wrappers>.

7.1 Python Wrapper

For developing python applications, refer to the following instructions to build up Python Wrapper.

7.1.1 Prerequisites

Ubuntu 16.04 LTS relies on several external software / packages when building LIPSedge™ AE400 SDK with Python wrapper, including:

- **Python:** Installs the Python language.
- **PIP tool:** Installs Python software packages.
- **NumPy Library:** Enables Python with advanced mathematical functions critical to the display of 3D images.
- **OpenCV-Python:** Enables computer vision for Python binding.

Note: Make sure your APT Tools are up to date before you start building LIPSedge™ AE400 SDK with Python wrapper. For details, refer to *4.1 Installing APT Tool*.

Installing Python

LIPSEdge™ AE400 SDK with Python wrapper is supported by **Python 2.7 / 3.0 or above**. Install Python prior to the wrapper building process.

1. Start **Terminal** and install **Python** and **Python-dev** package. Here, we use **Python 2.7** as an example.

For **Python 2.7**:

```
sudo apt-get install python python-dev
```

For **Python 3.5**:

```
sudo apt-get install python3.5 python3.5-dev
```

```
lips-hci@lips-hci-Inspiron-7560:~/ae400-realsense-sdk/build_Release$ sudo apt-get install python python-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
python is already the newest version (2.7.12-1-16.04).
The following package was automatically installed and is no longer required:
  snapd-login-service
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libpython-dev libpython2.7-dev python2.7-dev
The following NEW packages will be installed:
  libpython-dev libpython2.7-dev python-dev python2.7-dev
0 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.
```

Figure 7-1

Installing PIP Tool

PIP Tool allows efficient package installation / management for Python. Install PIP Tool in your Python environment before building LIPSedge™ AE400 SDK with Python wrapper.

1. Start **Terminal** and install the **PIP Tool**. Here, we use **Python 2.7 as an example**.

For **Python 2.7**:

```
sudo apt-get install python-pip
```

For **Python 3.5**:

```
sudo apt-get install python3-pip
```

```
lips-hci@lipshci-Inspiron-7560:~/ae400-realsense-sdk/build_Release$ sudo apt install python-pip
[sudo] password for lips-hci:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  snapd-login-service
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libpython-all-dev python-all python-all-dev python-pip-whl
  python-pkg-resources python-setuptools python-wheel
Suggested packages:
  python-setuptools-doc
The following NEW packages will be installed:
  libpython-all-dev python-all python-all-dev python-pip python-pip-whl
  python-pkg-resources python-setuptools python-wheel
0 upgraded, 8 newly installed, 0 to remove and 0 not upgraded.
Need to get 1583 kB of archives.
After this operation, 3066 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Figure 7-2

Installing NumPy library

Once the PIP Tool is setup, install the **NumPy** library to enable Python with advanced mathematical functions necessary for computing dimensional data for image representation.

1. Start Terminal and type the following command. Here, we use **Python 2.7** as an example.

For **Python2.7**:

```
pip2 install numpy
```

For **Python 3.5**:

```
pip3 install numpy
```

```
ltps-hcl@lps-hcl-Inspiron-7560:~/ae400-realsense-sdk/build_Release$ pip2 install numpy
Collecting numpy
  Downloading https://files.pythonhosted.org/packages/40/de/0ea5092b8bf2e3aa6fbb2e499a9f9adf810992884d414defc1573dca3f/numpy-1.18.1.zip (5.4MB)
    100% |#####| 5.4MB 266KB/s
Complete output from command python setup.py egg_info:
Traceback (most recent call last):
  File "<string>", line 1, in <module>
  File "/tmp/pip-build-RetVtv/numpy/setup.py", line 32, in <module>
```

Figure 7-3

Installing OpenCV-Python

LIPSedge™ AE400 SDK with Python wrappers requires [OpenCV 3.4.0](#) and its [contributed functionality module](#) when running applications such as `opencv_viewer_example.py`.

1. Start Terminal and type the following command.

```
sudo apt-get install python-opencv
```

```
lips-hcl@lips-hcl-Inspiron-7560:~/ae400-realsense-sdk/wrappers/python/examples$ sudo apt-get install python-opencv
[sudo] password for lips-hcl:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  snapsd-logind-service
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  lib965-va-driver libaac0 libavcodec-ffmpeg56 libavformat-ffmpeg56 libavutil-ffmpeg54 libbdplus0 libblas-common libblas3 libbluray1 libcr
  liblapack3 libmodplug1 libmp3lame0 libopencv-calib3d2.4v5 libopencv-contrib2.4v5 libopencv-core2.4v5 libopencv-features2d2.4v5 libopenc
  libopencv-legacy2.4v5 libopencv-nl2.4v5 libopencv-objdetect2.4v5 libopencv-photo2.4v5 libopencv-video2.4v5 libopenjpeg5 libschroedinger
  libswresample-ffmpeg1 libswscale-ffmpeg3 libtbb2 libtvolame0 libva1 libx264-148 libx265-79 libxvidcore4 libzvt1-common libzvt6 mesa-va
Suggested packages:
  libbluray-bdj firmware-crystalhd gfortran python-nose python-numpy-dbg python-numpy-doc libfglrx-andxvba1
The following NEW packages will be installed:
  lib965-va-driver libaac0 libavcodec-ffmpeg56 libavformat-ffmpeg56 libavutil-ffmpeg54 libbdplus0 libblas-common libblas3 libbluray1 libcr
  liblapack3 libmodplug1 libmp3lame0 libopencv-calib3d2.4v5 libopencv-contrib2.4v5 libopencv-core2.4v5 libopencv-features2d2.4v5 libopenc
  libopencv-legacy2.4v5 libopencv-nl2.4v5 libopencv-objdetect2.4v5 libopencv-photo2.4v5 libopencv-video2.4v5 libopenjpeg5 libschroedinger
  libswresample-ffmpeg1 libswscale-ffmpeg3 libtbb2 libtvolame0 libva1 libx264-148 libx265-79 libxvidcore4 libzvt1-common libzvt6 mesa-va
0 upgraded, 58 newly installed, 0 to remove and 0 not upgraded.
Need to get 18.7 MB of archives.
After this operation, 84.3 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://tw.archive.ubuntu.com/ubuntu xenial/universe amd64 libcrystalhd3 amd64 1:0.0-git20110715.fdd2f19-1build1 [46.1 kB]
Get:2 http://tw.archive.ubuntu.com/ubuntu xenial/universe amd64 libgs91 amd64 1:0.19-4 [27.1 kB]
Get:3 http://tw.archive.ubuntu.com/ubuntu xenial/universe amd64 libschroedinger-1.0-0 amd64 1.0.11-2.1build1 [293 kB]
Get:4 http://tw.archive.ubuntu.com/ubuntu xenial/universe amd64 libaac0 amd64 0.8.1-1 [47.4 kB]
Get:5 http://tw.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 libavutil-ffmpeg54 amd64 7:2.8.15-0ubuntu0.16.04.1 [166 kB]
Get:6 http://tw.archive.ubuntu.com/ubuntu xenial/universe amd64 libmp3lame0 amd64 3.99.5+repack1-9build1 [127 kB]
Get:7 http://tw.archive.ubuntu.com/ubuntu xenial/universe amd64 libopenjpeg5 amd64 1:1.5-2-3.1 [184 kB]
Get:8 http://tw.archive.ubuntu.com/ubuntu xenial/universe amd64 libshine3 amd64 3.1.0-4 [25.5 kB]
Get:9 http://tw.archive.ubuntu.com/ubuntu xenial/main amd64 libsnappy1v5 amd64 1.1.3-2 [16.0 kB]
Get:10 http://tw.archive.ubuntu.com/ubuntu xenial/universe amd64 libsoxr0 amd64 0.1.2-1 [70.2 kB]
Get:11 http://tw.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 libswresample-ffmpeg1 amd64 7:2.8.15-0ubuntu0.16.04.1 [51.7 kB]
Get:12 http://tw.archive.ubuntu.com/ubuntu xenial/universe amd64 libtvolame0 amd64 0.3.13-1.2 [49.2 kB]
Get:13 http://tw.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 libva1 amd64 1.7.0-1ubuntu0.1 [45.4 kB]
```

Figure 7-4

7.1.2 Building Python from Source Code

Once the build environment is configured, build the LIPSedge™ AE400 SDK with Python wrapper through **Terminal** command lines. Before the building process, make sure the prerequired items are installed.

For details on **relevant application / compiler**, refer to *4. Prerequisites*.
For details on **Python requirements**, refer to *6.1.1.1 Prerequisites*.

After the build process completes, update your local environment variable and import them to the **pyrealsense library**, the Python wrapper for Intel RealSense SDK 2.0.

1. Upgrade APT Tools to the latest version. For details, refer to *4.1 Installing APT Tools*.
2. Install Python and relevant developer tools. For details, refer to *Installing Python, 6.1.1.1 Prerequisites*.
3. Clone the **LIPSedge™ AE400 SDK source code package** to your **PC / laptop**. For details, refer to *5.1 Cloning LIPSedge™ AE400*.

```
git clone https://github.com/lips-hci/ae400-realsense-sdk.git
```

4. Assign the **location on your PC / laptop** for the built outcome for **CMake** and go to the folder. For details, refer to *5.1 Cloning LIPSedge™ AE400*.

```
mkdir build_Release && cd build_Release
```

5. Generate the **Makefile** required for building Python wrapper with the following values. Here, we built Python sample programs in **Python 2.7** as an example.

```

cmake .. -DCMAKE_BUILD_TYPE=Release
-DBUILD_PYTHON_BINDINGS=true
-DPYTHON_EXECUTABLE=/usr/bin/python2.7

```

```

ltps-hcl:/public/realtime/2660-ae400-realsense-sdk/build_release$ cmake .. -DCMAKE_BUILD_TYPE=Release -DBUILD_PYTHON_BINDINGS=true -DPYTHON_EXECUTABLE=/usr/bin/python2.7
-- Info: REALSENSE_VERSION_STRING=2.17.1
-- Setting linker configurations
-- Checking internet connection...
-- Internet connection identified, enabling BUILD_WITH_TM2
-- Found backend-ethernet library: /home/ltps-hcl/ae400-realsense-sdk/third-party/ltps/lib/v0.9/linux/amd64/libbackend-ethernet.a
-- Found PythonInterp: /usr/bin/python2.7 (found version "2.7.12")
-- Found PythonLibs: /usr/lib/x86_64-linux-gnu/libpython2.7.so
-- pybind11 v2.2.1
-- Performing Test HAS_FLTO
-- Performing Test HAS_FLTO - Success
-- LTO enabled
-- Building with TM2
-----
-- T265 Product versions:
-- - HOST 0.19.3.1428 (Default from versions.cmake)
-- - Remote FW 0.0.18.4386 (Default from versions.cmake)
-- - Remote CENTRAL APP 2.0.19.271 (Default from versions.cmake)
-- - Remote CENTRAL BL 1.0.1.112 (Default from versions.cmake)
-----
-- Downloading FW 0.0.18.4386 from "http://realsense-hw-public.s3.amazonaws.com/Releases/TM2/FW/target/0.0.18.4386/target-0.0.18.4386.mvncd"
-- Converting FW version 0.0.18.4386 from target.mvncd to /home/ltps-hcl/ae400-realsense-sdk/third-party/libtn/libtn/src/fw.h
-- Downloading Central App 2.0.19.271 from "http://realsense-hw-public.s3.amazonaws.com/Releases/TM2/FW/app/2.0.19.271/central_app-2.0.19.271.bin"
-- Converting Central App version 2.0.19.271 from central_app.bin to /home/ltps-hcl/ae400-realsense-sdk/third-party/libtn/libtn/src/CentralAppFW.h
-- Downloading Central BL 1.0.1.112 from "http://realsense-hw-public.s3.amazonaws.com/Releases/TM2/FW/bl/1.0.1.112/central_bl-1.0.1.112.bin"
-- Converting Central BL version 1.0.1.112 from central_bl.bin to /home/ltps-hcl/ae400-realsense-sdk/third-party/libtn/libtn/src/CentralBLFW.h
-----
-- Building libtn project on Linux, LIBTN version [0.19.3.1428], API version [10.0], branch [master], FW [0.0.18.4386], Central APP [2.0.19.271], Central BL [1.0.1.112]
-- Creating version file /home/ltps-hcl/ae400-realsense-sdk/third-party/libtn/libtn/src/Version.h
-- Building project to as static library lib
-----
-- Building all projects of libtn_samples
-- Building project libtn_util
-----
-- Cmake Done
-- Configuring done
-- Generating done
-- Build files have been written to: /home/ltps-hcl/ae400-realsense-sdk/build_release

```

Figure 7-5

Note: To build Python samples programs for different Python versions, change the value for Python directory location in - **DPYTHON_EXECUTABLE**. For example, for building Python sample programs for Python 3.5, type - **DPYTHON_EXECUTABLE=/usr/bin/python3.5**.

- Build the LIPSedge™ AE400 SDK with Python wrapper.

```
make -j4
```

```
lips-hci@lipshci-inspiron-7560:~/ae400-realsense-sdk/build_Release$ make -j4
Scanning dependencies of target pybackend2
Scanning dependencies of target tm
[ 0%] Building CXX object wrappers/python/CMakeFiles/pybackend2.dir/__/__/src/backend.cpp.o
[ 1%] Building CXX object wrappers/python/CMakeFiles/pybackend2.dir/pybackend_extras.cpp.o
[ 12%] Built target realsense-file
[ 13%] Building CXX object third-party/libtm/libtm/src/CMakeFiles/tm.dir/Manager.cpp.o
[ 13%] Building CXX object wrappers/python/CMakeFiles/pybackend2.dir/pybackend.cpp.o
[ 13%] Building CXX object third-party/libtm/libtm/src/CMakeFiles/tm.dir/device.cpp.o
[ 14%] Building CXX object wrappers/python/CMakeFiles/pybackend2.dir/__/__/src/types.cpp.o
[ 14%] Building CXX object wrappers/python/CMakeFiles/pybackend2.dir/__/__/src/archive.cpp.o
[ 15%] Building CXX object wrappers/python/CMakeFiles/pybackend2.dir/__/__/src/log.cpp.o
[ 16%] Linking CXX static library libtm.a
```

Figure 7-6

- Install the LIPSedge™ AE400 SDK with Python wrapper onto your PC / laptop.


```
sudo make install
```

```
lips-hci@lipshci-Inspiron-7560:~/ae400-realsense-sdk/build_Release$ sudo make install
[ 7%] Built target tm
[ 18%] Built target realsense-file
[ 54%] Built target realsense2
[ 55%] Built target pyrealsense2
[ 59%] Built target pybackend2
[ 60%] Built target rs-software-device
[ 61%] Built target rs-capture
[ 62%] Built target rs-callback
[ 63%] Built target rs-save-to-disk
[ 64%] Built target rs-multicam
[ 65%] Built target rs-pointcloud
[ 67%] Built target rs-align
[ 69%] Built target rs-sensor-control
[ 70%] Built target rs-measure
[ 71%] Built target rs-depth
[ 73%] Built target rs-color
[ 75%] Built target rs-distance
[ 78%] Built target rs-post-processing
[ 81%] Built target rs-record-playback
[ 82%] Built target lips_imu_reader
[ 83%] Built target rs-terminal
[ 86%] Built target rs-fw-logger
[ 87%] Built target rs-enumerate-devices
[ 91%] Built target realsense-viewer
[ 92%] Built target rs-data-collect
[ 96%] Built target rs-depth-quality
[ 98%] Built target rs-rosbag-inspector
[ 99%] Built target rs-convert
[100%] Built target libtm_util
```

Figure 7-7

8. Update the PYTHONPATH environment variable and import them to the pyrealsense library. The building process is completed.

```
export PYTHONPATH=$PYTHONPATH:/usr/local/lib
```



```
lips-hcl@lipshcl-Inspiron-7560:~/ae400-realsense-sdk/build_Release$ export PYTHONPATH=$PYTHONPATH:/usr/local/lib
```

Figure 7-8

7.1.3 Starting Test Applications

LIPS Corp. provides a series of test applications for verifying the connection between Python and RealSense library. Refer to **LIPSedge™ AE400 SDK > wrappers > python > examples** for the sample applications.

No.	Name	Features
1.	python-tutorial-1-depth.py	Displays depth image in binary format.
2.	opencv_viewer_example.py	Displays live image data.
3.	align-depth2color.py	Mapping depth / RGB images.
4.	opencv_pointcloud_viewer.py	Displays point cloud image data.

In the SDK building process, LIPSedge™ AE400 SDK generates the 3rd party libraries necessary for application development. These 3rd party libraries are bundled in the following location:

Name	Location
librealsense2 pyrealsense2	LIPSedge™ AE400 SDK > build_Release e.g. /home/lips-hci/ae400-realsense-sdk/build_Release
	LIPSedge™ AE400 SDK > build_Release > wrappers > python > examples e.g. /home/lips-hci/ae400-realsense-sdk/build_Release

Before starting the test applications, copy these 3rd party libraries to the test applications' location.

Viewing Depth Images

Start `python-tutorial-1-depth.py` to view live depth image data on **Terminal** in binary format.

1. Start **Terminal** and go to **LIPSedge™ AE400 SDK > wrappers > python > examples.**



Figure 7-9

2. Start `python-tutorial-1-depth.py`.

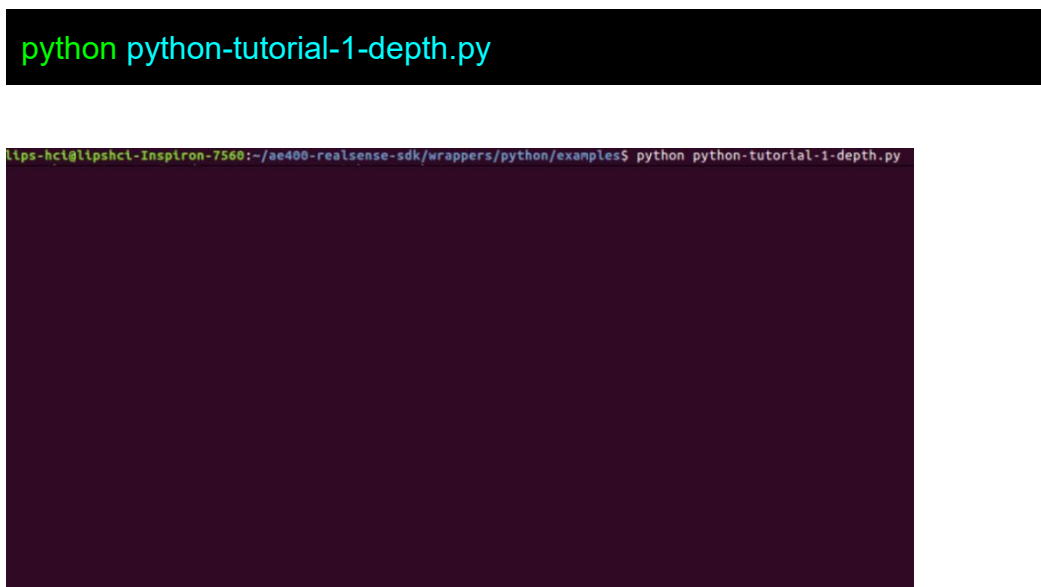


Figure 7-10

Starting Other Test Applications

For other test applications, type the following command. Here, we use **opencv_viewer_example.py** as an example. For details of application features, refer to similar content in *7.6.3.4 Starting Test Application, Windows*.

```
python [python version] [Test Application's Name]
```

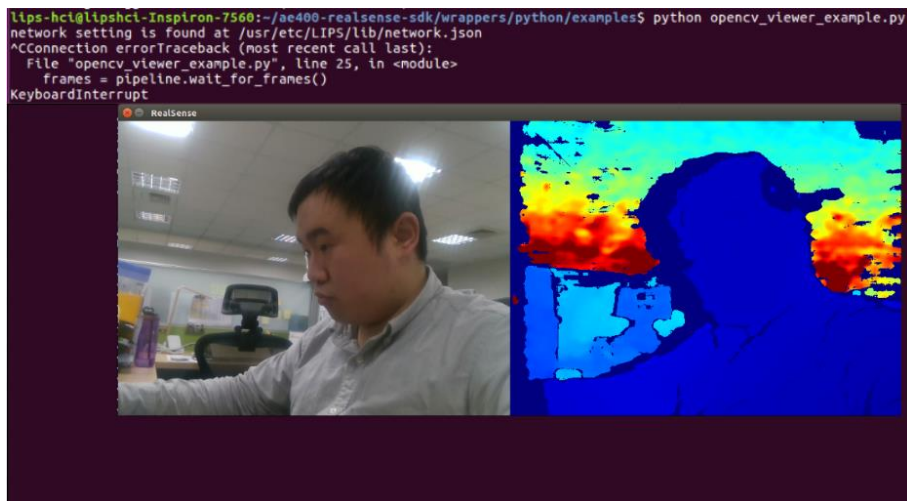


Figure 7-12

7.1.4 Editing Sample Codes

You can edit the sample codes of the test application through the **Text Editor** of your choice.

1. Go to **LIPSEdge™ AE400 SDK > wrappers > python > examples**.

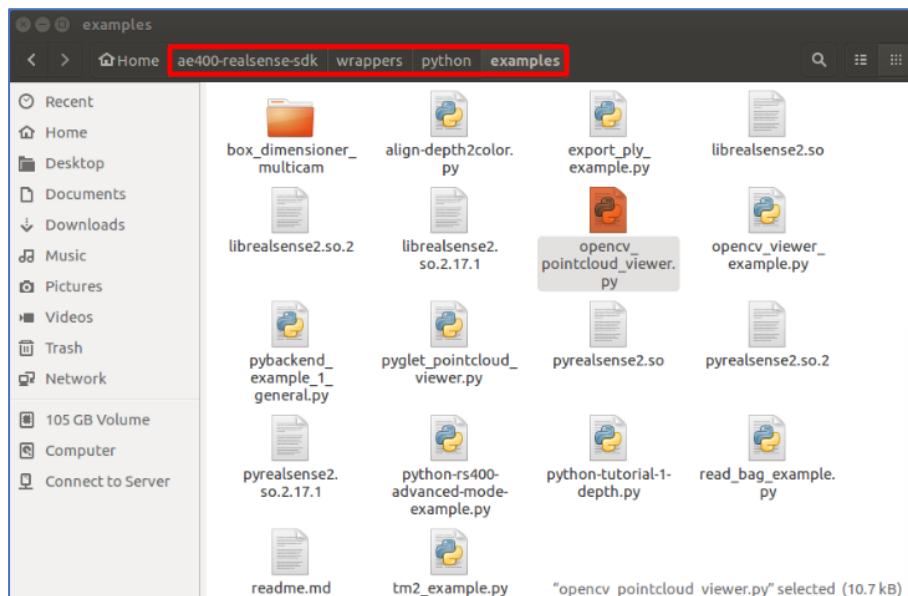


Figure 7-13

2. Double-click the test application.

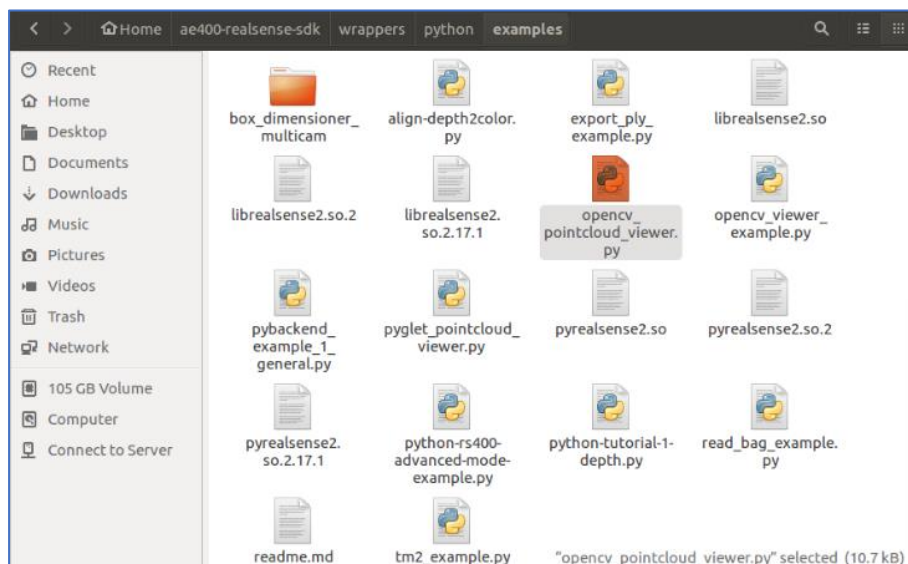
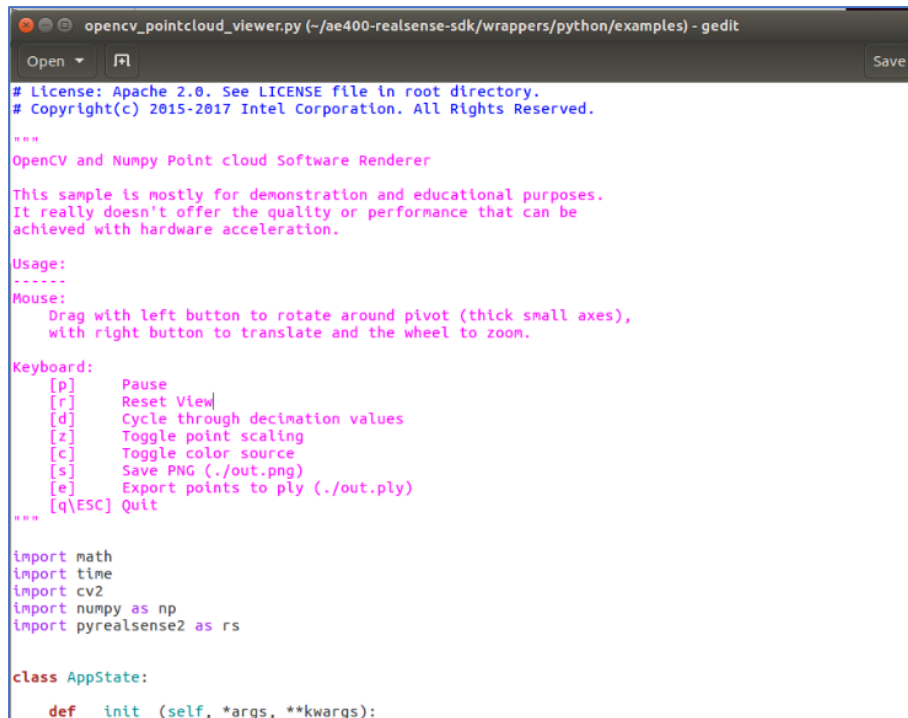


Figure 7-14

3. A **Text Editor** loaded with the sample codes pops up. Click **Save** to keep the changes as you finished.



```
opencv_pointcloud_viewer.py (-/ae400-realsense-sdk/wrappers/python/examples) - gedit
Open [ ] Save

# License: Apache 2.0. See LICENSE file in root directory.
# Copyright(c) 2015-2017 Intel Corporation. All Rights Reserved.

"""
OpenCV and Numpy Point cloud Software Renderer

This sample is mostly for demonstration and educational purposes.
It really doesn't offer the quality or performance that can be
achieved with hardware acceleration.

Usage:
-----
Mouse:
    Drag with left button to rotate around pivot (thick small axes),
    with right button to translate and the wheel to zoom.

Keyboard:
    [p]   Pause
    [r]   Reset View
    [d]   Cycle through decimation values
    [z]   Toggle point scaling
    [c]   Toggle color source
    [s]   Save PNG (./out.png)
    [e]   Export points to ply (./out.ply)
    [q\ESC] Quit
"""

import math
import time
import cv2
import numpy as np
import pyrealsense2 as rs

class AppState:
    def __init__(self, *args, **kwargs):
```

Figure 7-15

Note: The sample codes are based on **Python Wrapper for Intel RealSense SDK 2.0**. For detail information, refer to <https://github.com/IntelRealSense/librealsense/tree/master/wrappers/python/examples>.



Appendix - System Tools Overview

Besides the Intel® RealSense™ viewer, LIPS IMU reader and LIPS AE400 Toolkit, LIPS Corp. also provides a series of handy system tools use for program development purposes. 3 types of tools are provided:

Intel® RealSense™ System Tools, Debugging Tools and LIPS System Tools. You can access these tools at **LIPSedge™ AE400 SDK > build_Release > Debug.**

Name	Type	Date modified	Size
libtm_util.exe	Application	2020/2/15 1:52 PM	23,733 KB
libtm_util.ilink	Incremental Linke...	2020/2/15 1:52 PM	14,458 KB
libtm_util.pdb	Program Debug D...	2020/2/15 1:52 PM	21,964 KB
lips_imu_reader.exe	Application	2020/2/12 3:57 PM	1,810 KB
lips_imu_reader.ilink	Incremental Linke...	2020/2/12 3:57 PM	4,426 KB
lips_imu_reader.pdb	Program Debug D...	2020/2/12 3:57 PM	8,012 KB
network.json	JSON File	2020/2/15 1:52 PM	1 KB
opencv_world340.dll	Application exten...	2017/12/23 5:22 AM	63,905 KB
opencv_world340d.dll	Application exten...	2017/12/23 5:28 AM	101,735 KB
pybackend2.ilink	Incremental Linke...	2020/2/13 10:46 AM	25,829 KB
pybackend2.pdb	Program Debug D...	2020/2/13 10:46 AM	45,420 KB
pybackend2.pyd	PYD File	2020/2/13 10:46 AM	9,250 KB
pyrealsense2.ilink	Incremental Linke...	2020/2/13 10:35 AM	19,437 KB
pyrealsense2.pdb	Program Debug D...	2020/2/13 10:35 AM	37,124 KB
pyrealsense2.pyd	PYD File	2020/2/13 10:35 AM	8,747 KB
realsense2.dll	Application exten...	2020/2/15 1:52 PM	33,115 KB
realsense2.exp	Exports Library File	2020/2/12 3:56 PM	38 KB
realsense2.ilink	Incremental Linke...	2020/2/15 1:52 PM	104,486 KB
realsense2.lib	Object File Library	2020/2/12 3:56 PM	63 KB
realsense2.pdb	Program Debug D...	2020/2/15 1:52 PM	127,444 KB
realsense-viewer.exe	Application	2020/2/12 3:57 PM	7,914 KB
realsense-viewer.ilink	Incremental Linke...	2020/2/12 3:57 PM	18,285 KB

No.	File Name	Function
Intel® RealSense™ System Tools		
1.	realsense-viewer	Accesses RGB and depth images.
2.	rs-depth-quality	Accesses depth matrices of a surface to assess the quality of the depth data.
3.	rs-convert	Converts the ROS-bag file to other formats.

No.	File Name	Function
Debugging Tools		
4.	rs-enumerate-devices	Provides information about the connected device.
5.	rs-fw-logger	Collects internal camera logs.
6.	rs-data-collect	Produces a report of frame statistics in a Microsoft Excel File (.csv).
7.	rs-terminal	Enables programmers to send commands to the camera firmware.
8.	rs-rosbag-inspector	Displays .bag files available.
LIPS System Tools		
9.	LIPS IMU Reader	Displays raw IMU data live.

Note: Aside from these core system tools, LIPS Corp. also provides other native Intel® RealSense™ system tools. Currently, LIPS Corp. does **NOT** provide additional technical supports for any possible errors over the native Intel® RealSense™ system tools that are not listed in this section. Use the native Intel® RealSense™ system tools with caution.



LIPS CORPORATION

2F, No. 100, Ruiguang Road, Neihu District,
Taipei City 114, Taiwan

Tel.: + 886-2-8791-6998 ext. 206

MP: + 886-927-982-829

Fax: +886-2-8791-8996

Official Website: <https://www.lips-hci.com/>

E-Mail: info@lips-hci.com