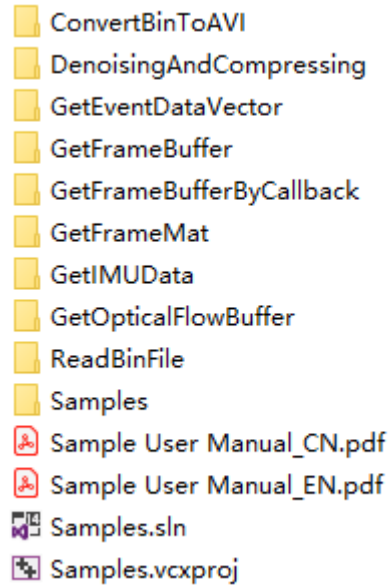


# 1 Introduction

The file directory of *Samples* is as follows:



There are ten sub directories in the *Samples*. The first nine directories are the main function instance code for the CeleX™ Sensor. The last directory mainly includes some necessary configure file, include files and library. More related content is introduced as follows:

## 1.1 ConvertBinToAVI

It shows how to convert a recorded bin file into an AVI format video.

## 1.2 DenoisingAndCompressing

It shows how to get images after denoising or compressing in Event mode.

## 1.3 GetEventDataVector

It shows how to get the (X, Y, A, T) information and use it to create an image frame.

## 1.4 GetFrameBuffer

It shows how to set the working mode of the CeleX™ Sensor and obtain the data that the CeleX™ Sensor works in different modes.

## 1.5 GetFrameBufferByCallback

It shows how to set the working mode of the CeleX™ Sensor and register to monitor the data that the CeleX™ Sensor works in different modes.

## 1.6 GetFrameMat

It shows how to set the working mode of the CeleX™ Sensor and obtain the data in cv::Mat form that the CeleX™ Sensor works in different modes.

## 1.7 GetIMUData

It shows how to get IMU data including parameters of accumulator and gyros.

## 1.8 GetOpticalFlowBuffer

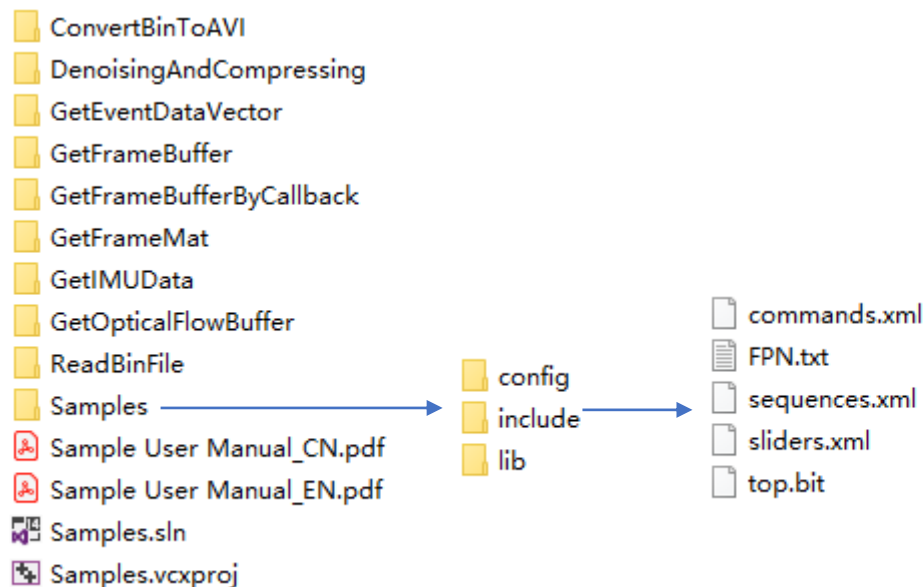
It shows how to get the optical flow data in Event mode.

## 1.9 ReadBinFile

It shows how to read the recorded bin file and display it.

## 1.10 Samples

There are three folders in the *Samples* folder:



### 1.10.1 include

The header files of the API are placed in the *include* directory.

### 1.10.2 lib

The API libraries is included in the *lib* directory (including 64-bit release and debug libraries and 32-bit release and debug libraries).

### 1.10.3 config

There are a FPN file and four configuration files in *config* directory that CeleX™ Sensor will use when starting up. To run the program developed by CeleX™ Sensor library, you must copy the following four configuration files to the directory where the executable file is located, or the program will not be executed. In these samples, these files are automatically copied to the build directory, and users do not need to manually copy these files.

#### Notes:

Users generally do not need to modify these configuration parameters. They can be adjusted by calling the APIs. For details, please refer to the API User Manual.

The OpenCV library is not included in the samples, so to use these, it need to install OpenCV first.

## 2 Compile Samples

You can compile and run these sample codes in Windows or Linux. There are some differences when compiling under Windows and Linux. The detailed description is as follows:

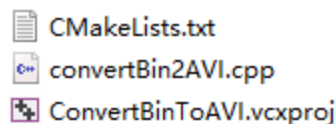
### 2.1 Windows

The samples are developed by the Visual Studio 2015 under Windows 10. Before compiling the sample code, you need to install OpenCV first. The OpenCV version used in samples is 3.3.0. The current OpenCV include directory and lib directory are under the local D:\Program Files\opencv. You need to specify the include and lib path to your OpenCV installation directory in the property list.

With Visual Studio 2015, you can open the project by opening the .sln or .vcxproj file. You can compile and run any one sample by setting this as startup item.

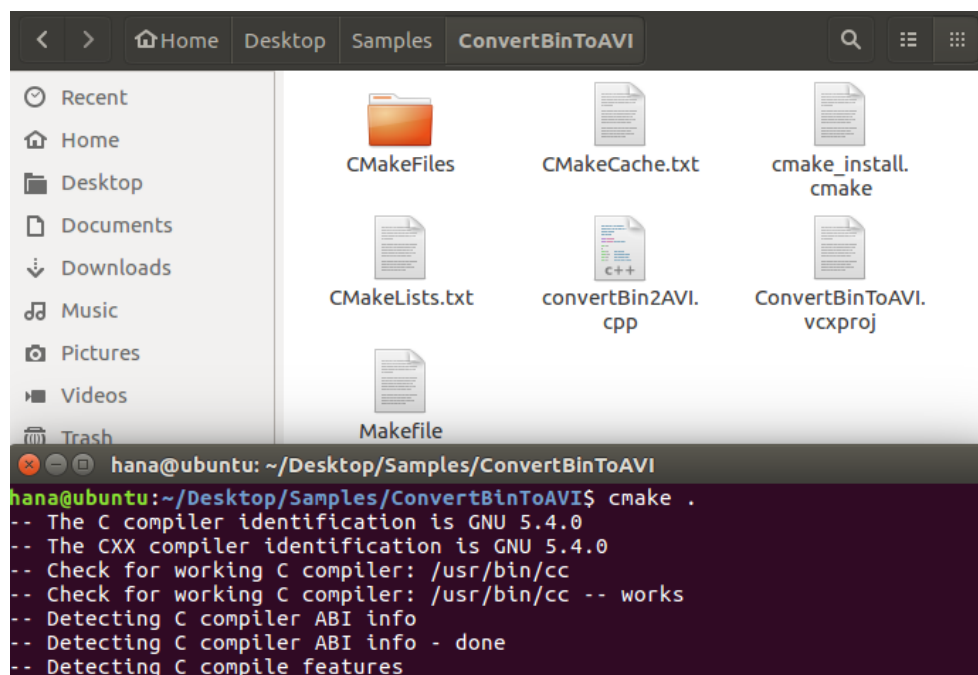
### 2.2 Linux

You can also compile and run these sample code in Linux. Each sample has the following three files:

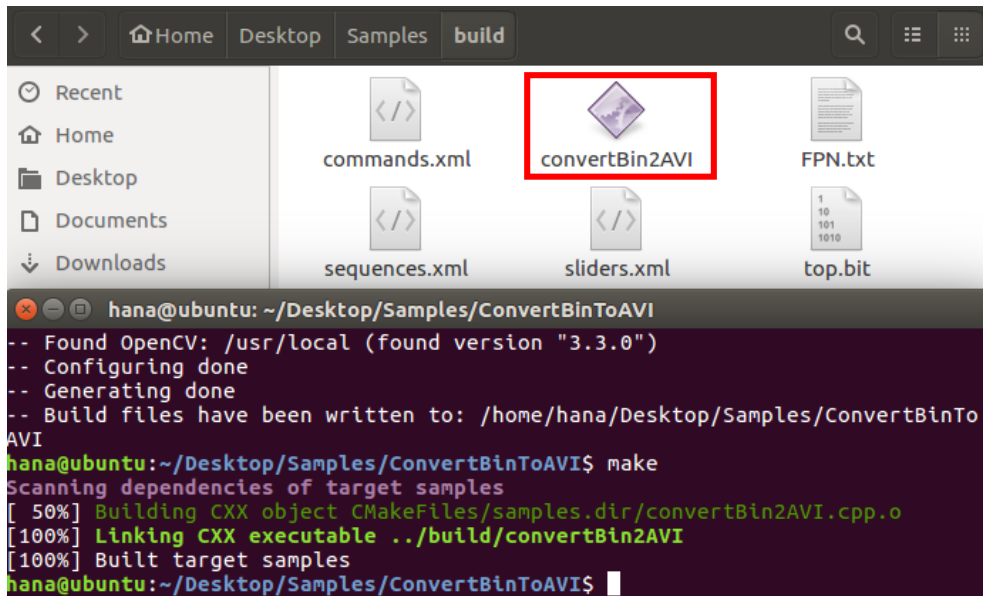


The CMakeList.txt is used in Linux, .cpp file is the source code and the .vcxproj file is used in Windows.

In Linux, you can use the CMakeList file to compile the samples. Note: Before compiling, check if there is a Linux library file in the Samples\Samples\lib\Linux directory (where x64 is a 64-bit library and x86 is a 32-bit library).



Then if there is no link error in the include and lib directories, you can use make to compile.



The executable file is generated to the *build* directory (If not able to find the directory, please build a new *build* folder). There are four configuration files in *build* directory that CeleX™ Sensor will use when starting up. Then you can run the generated file. If the operation fails, check if the necessary files are included in the current running file directory.

