1 Introduction

The file directory of *Samples* is as follows:



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

1.1 GetEventDataVector

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

1.2 GetFrameBuffer

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

1.3 GetFrameBufferByCallback

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

1.4 GetFrameMat

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

1.5 GetOpticalFlowBuffer

It shows how to get the optical flow data.

1.6 ReadBinFile

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

1.7 Samples

There are three folders in the Samples folder:



1.7.1 include

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

1.7.2 lib

The API libraries is included in the *lib* directory (including dynamic link library of 64-bit Windows and Linux). To run the program developed by CeleXTM Sensor library, you must copy the dynamic link library file to the directory where the executable file is located, or the program will not be executed. In these samples, dynamic link library is automatically copied to the executable directory.

1.7.3 config

By default, the FPN file used in the sample is placed in this *config* directory.

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:

CMakeLists.txt
 getEventDataVector.cpp
 GetEventDataVector.vcxproj

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\lib\Linux\x64** directory (including dynamic link library: **libCeleX.so**).

く 〉 企 Home De	esktop Samples	GetEventDataVector	۹						
⊘ Recent									
✿ Home									
🛅 Desktop	CMakeFiles	CMakeCache.txl	t cmake_install. cmake						
Documents		i marina Marina Marina Marina Marina							
Downloads		c++							
J Music	CMakeLists.t	xt getEventDataVec r.cpp	to GetEventDataVect or.vcxproj						
Pictures									
Videos	A final sector and the sector and th								
💮 Trash	Makefile								
8 🖶 🗉 hana@ubuntu:	🛞 🖨 💷 hana@ubuntu: ~/Desktop/Samples/GetEventDataVector								
<pre>hana@ubuntu:~/Desktop/Samples/GetEventDataVector\$ cmake The C compiler identification is GNU 5.4.0 - The CXX compiler identification is GNU 5.4.0 - Check for working C compiler: /usr/bin/cc - Check for working C compiler ABI info - Detecting C compiler ABI info - done - Detecting C compile features - Detecting C compile features - done - Check for working CXX compiler: /usr/bin/c++ - Check for working CXX compiler: /usr/bin/c++ - Detecting C X compiler ABI info</pre>									
Detecting CXX compiler ABI info - done									
Detecting CXX compile features Detecting CXX compile features - done									
Found OpenCV: /usr/local (found version "3.3.0")									
Configuring done									

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

SO build								
< > ✿Home Desi	ctop Samples bu	ild		۹				
⊘ Recent☆ Home								
🕞 Desktop 😣 😑 🗊 hana@ubuntu: ~/	getEventDataVect Desktop/Samples/	o GetEventDataVeo	tor.					
 The CXX compiler identification is GNU 5.4.0 Check for working C compiler: /usr/bin/cc Check for working C compiler: /usr/bin/cc works Detecting C compiler ABI info Detecting C compile features Detecting C compile features - done 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 features Detecting CXX compile features Configuring done 								
Generating done Build files have been written to: /home/hana/Desktop/Samples/GetEventDataVect or								
hana@ubuntu:~/Desktop/ Scanning dependencies [50%] Building CXX ob [100%] Linking CXX exe [100%] Built target sa	Samples/GetEven of target sampl ject CMakeFiles cutable/buil mples	tDataVector\$ m es /samples.dir/g d/getEventData 	ake etEventDataVecto Vector	ог.срр.о				

The executable file is generated to the build directory. All sample executables are compiled and

placed in the *build* directory. Then you can run the generated file and wait until the client program connected to the ZYNQ server program.

SO build			
< > 🏠 Home De	sktop Samples	build	Q
 Recent Home Desktop hana@ubuntu: - 	getEventDatav r ~/Desktop/Sampl	Vecto	
<pre>nanagupuntu:~/Desktop XBase::getApplicatior create socket success connect failed, try t connect failed, try t</pre>	p/Samples/Dull DirPath: read fully! to connect aga to connect aga	d\$./getEventDatavector link count = 51 in, please wait in, please wait	