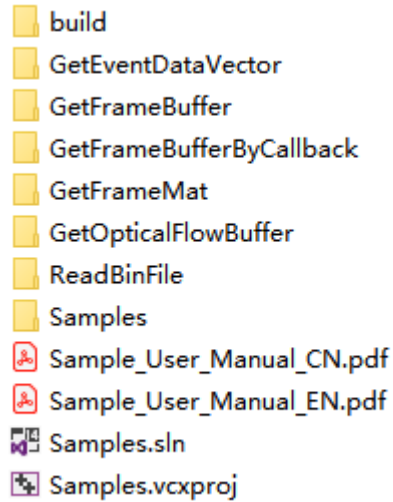


1 介绍

示例代码文件夹 *Samples* 的结构如下图所示：



Samples 文件夹下包括 8 个子文件夹、工程文件和使用手册。其中 6 个文件夹是 Celex™ Sensor 的主要功能的示例代码。最后一个文件夹里放的是运行代码所必须的配置文件以及包含目录和库文件。更多相关示例代码的介绍会在下面进行详细介绍：

1.1 GetEventDataVector

本示例给出了如何获取一帧数据的 (X, Y, A, T) 信息，并使用它建立图像帧。

1.2 GetFrameBuffer

本示例给出了如何设置 Celex™ Sensor 的工作模式，并**主动获取** Celex™ Sensor 工作在不同模式的数据。

1.3 GetFrameBufferByCallback

本示例给出了如何设置 Celex™ Sensor 的工作模式，并注册监听 Celex™ Sensor 工作在不同模式的数据。（**被动接收**）

1.4 GetFrameMat

本示例给出了如何设置 Celex™ Sensor 的工作模式，并**主动获取** Celex™ Sensor 工作在不同模式的数据（**cv::Mat** 类型）。

1.5 GetOpticalFlowBuffer

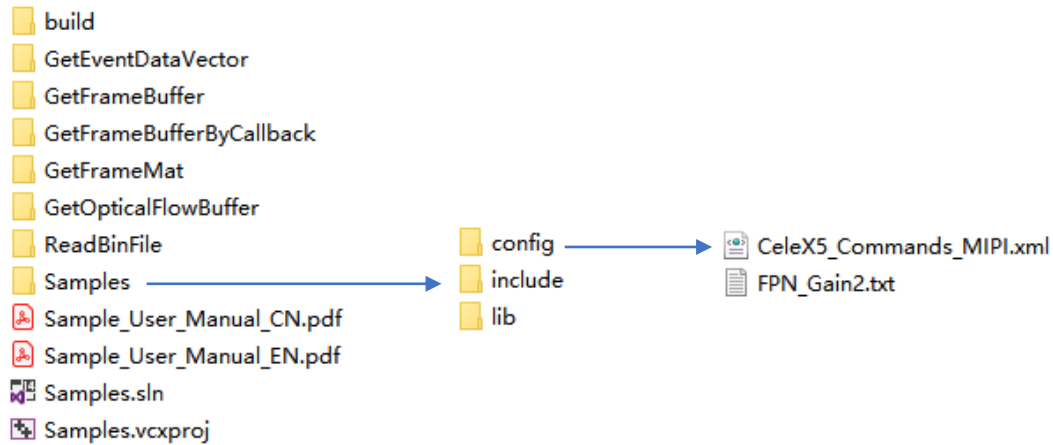
本示例给出了如何获取光流数据。

1.6 ReadBinFile

本示例给出了如何读取录制的 bin 文件，并实时显示。

1.7 Samples

子文件夹 *Samples* 中有如下 3 个文件夹：



1.7.1 include

include 中放的 API 的头文件。

1.7.2 lib

lib 中放的是 API 库文件（包括 win64 位以及 Linux 下的动态链接库）。

1.7.3 config

config 中放的是 CeleX™ Sensor 启动时会用到的配置文件以及一个 FPN 文件，要运行使用 CeleX™ Sensor 库开发的程序，必须要把该配置文件拷贝到与可执行文件所在的目录，否则程序会启动不了。该示例中会自动拷贝这些文件到编译目录下，使用者不需要手动拷贝这些文件。

备注：

用户一般不需要直接修改这些配置参数，可以通过调用 API 控制 Sensor，具体请参考 CeleX_API_Reference_CN。

Sample 中不包含三方库 OpenCV，所以要使用本示例，需要先安装 OpenCV 并设置它的路径。本示例 OpenCV 的包含目录路径以及库目录位于本机 D:\Program Files\opencv 下。

2 示例代码的编译

您可以在 Windows 或 Linux 中编译和运行这些示例代码。在 Windows 和 Linux 下编译时存在一些差异。详细说明如下：

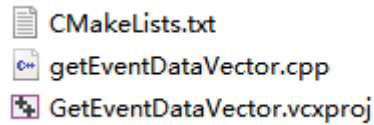
2.1 Windows

这些示例由 Windows 10 下的 Visual Studio 2015 开发。在编译示例代码之前，您需要先安装 OpenCV。样本中使用的 OpenCV 版本是 3.3.0。当前的 OpenCV 包含目录和 lib 目录位于本地 D:\Program Files\opencv 下。您需要在属性列表中指定 OpenCV 安装目录的 include 和 lib 路径。

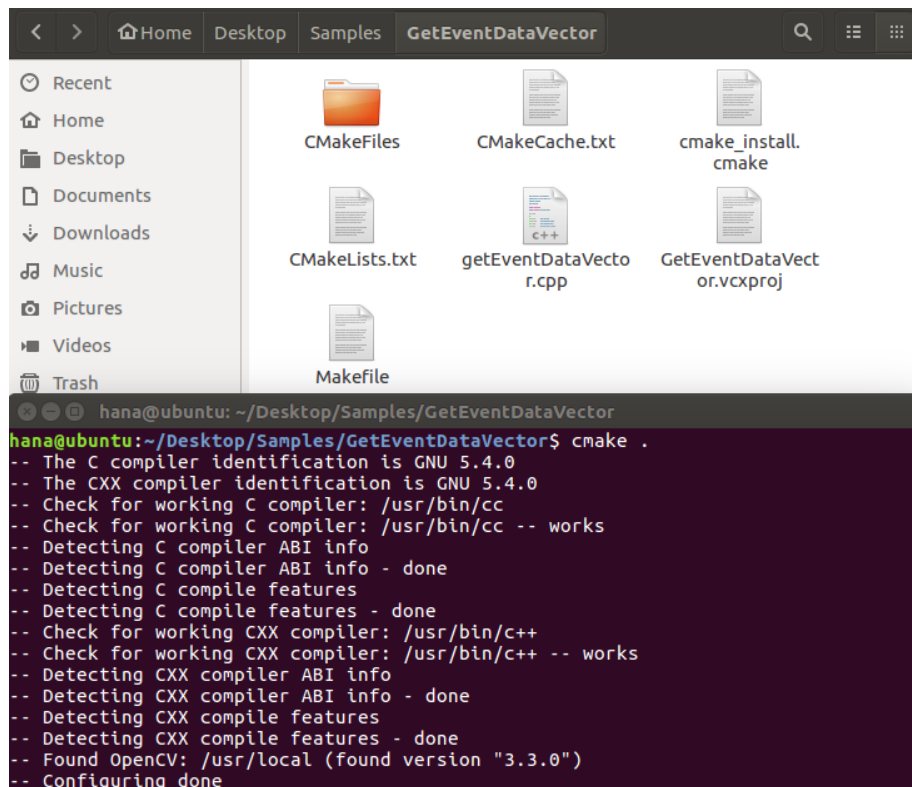
使用 Visual Studio 2015，您可以通过打开.sln 或.vcxproj 文件来打开项目。通过将其设置为启动项来编译和运行任何一个示例。

2.2 Linux

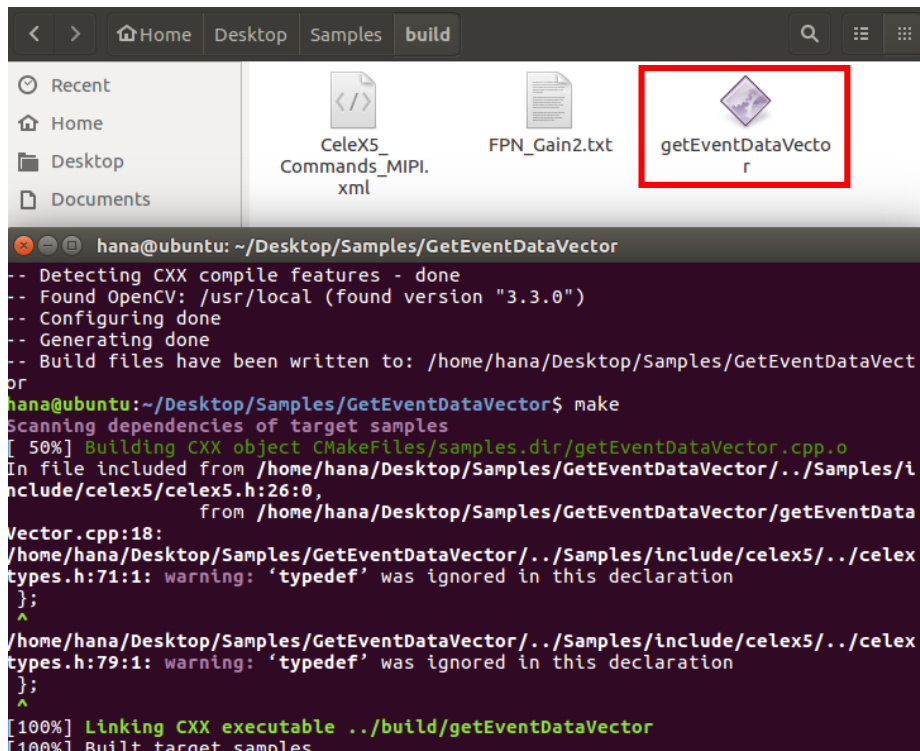
您还可以在 Linux 中编译和运行这些示例代码。每个示例文件中都有以下三个文件：



其中，CMakeList.txt 用于 Linux，.cpp 文件是源代码，.vcxproj 文件用于 Windows。在 Linux 中，您可以使用 CMakeLists 文件来编译示例。**注：**编译之前，先检查一下 Samples\Samples\lib\Linux 目录下是否有 Linux 的库文件（包括程序运行所需动态链接库：libCeleDriver.so, libCeleX.so, libokFrontPanel.so）。



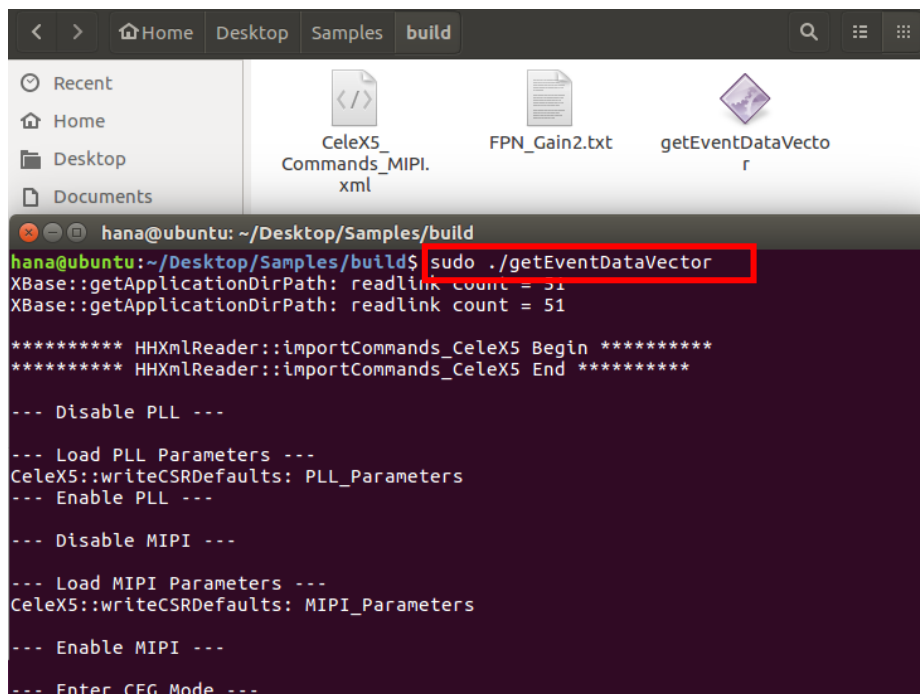
然后，如果 include 和 lib 目录中没有链接错误，则可以使用 make 进行编译。



The screenshot shows a file manager window with the path `~/Desktop/Samples/build`. It contains files `CeleX5_Commands_MIPI.xml`, `FPN_Gain2.txt`, and `getEventDataVector` (highlighted with a red box). Below it, a terminal window shows the execution of `make` in the `~/Desktop/Samples/GetEventDataVector` directory. The terminal output indicates that the build files have been written to the build directory and that the target `samples` has been successfully built.

```
hana@ubuntu: ~/Desktop/Samples/GetEventDataVector
-- Detecting CXX compile features - done
-- Found OpenCV: /usr/local (found version "3.3.0")
-- Configuring done
-- Generating done
-- Build files have been written to: /home/hana/Desktop/Samples/GetEventDataVector
hana@ubuntu:~/Desktop/Samples/GetEventDataVector$ make
Scanning dependencies of target samples
[ 50%] Building CXX object CMakeFiles/samples.dir/getEventDataVector.cpp.o
In file included from /home/hana/Desktop/Samples/GetEventDataVector/./Samples/include/celeX5/celeX5.h:26:0,
                 from /home/hana/Desktop/Samples/GetEventDataVector/getEventDataVector.cpp:18:
/home/hana/Desktop/Samples/GetEventDataVector/./Samples/include/celeX5/./celeXtypes.h:71:1: warning: 'typedef' was ignored in this declaration
};
^
/home/hana/Desktop/Samples/GetEventDataVector/./Samples/include/celeX5/./celeXtypes.h:79:1: warning: 'typedef' was ignored in this declaration
};
^
[100%] Linking CXX executable ../build/getEventDataVector
[100%] Built target samples
```

可执行文件将默认生成到 **build** 目录下。 **build** 目录中有配置文件，Celex™ Sensor 在启动时将使用这个文件。最后，您可以运行生成的文件。如果操作失败，请检查当前运行的文件目录中是否包含必要的文件。（注：由于五代 Sensor 需要对 USB 设备进行读写操作，需要使用 root 权限）



The screenshot shows the same file manager window as before, but now the `getEventDataVector` file is highlighted with a red box. Below it, a terminal window shows the execution of `sudo ./getEventDataVector` in the `~/Desktop/Samples/build` directory. The terminal output shows the program's initialization, including disabling PLL and MIPI, loading parameters, and entering CFG Mode.

```
hana@ubuntu: ~/Desktop/Samples/build
hana@ubuntu:~/Desktop/Samples/build$ sudo ./getEventDataVector
XBase::getApplicationDirPath: readlink count = 51
XBase::getApplicationDirPath: readlink count = 51

***** HHXmlReader::importCommands_CeleX5 Begin *****
***** HHXmlReader::importCommands_CeleX5 End *****

--- Disable PLL ---
--- Load PLL Parameters ---
CeleX5::writeCSRDefaults: PLL_Parameters
--- Enable PLL ---

--- Disable MIPI ---
--- Load MIPI Parameters ---
CeleX5::writeCSRDefaults: MIPI_Parameters
--- Enable MIPI ---
--- Enter CFG Mode ---
```