



celepixel

# 芯仑科技 SDK 使用说明

芯仑科技（上海）有限公司

## 目录

1	Celex Demo Kit 硬件 .....	3
2	Celex Demo Kit SKD 使用步骤 .....	4
2.1	安装 Opal Kelly 驱动 .....	4
2.1.1	Windows .....	4
2.1.2	Linux .....	4
2.2	运行 CeleX Demo GUI.....	5
2.2.1	Windows .....	5
2.2.2	Linux .....	5
2.3	编译 CeleX SDK 的 Source Code .....	5
2.3.1	Windows .....	6
2.3.2	Linux .....	6
2.4	编译 CeleX Demo GUI 的 Source Code .....	6
2.5	生成 FPN 文件 .....	8
3	Celex Demo Kit GUI 的功能 .....	9
3.1	切换 Sensor 工作模式.....	9
3.2	录制 Sensor 数据功能.....	11
3.3	播放录制的 Bin 文件功能.....	12
3.4	生成 FPN 功能 .....	12
3.5	各种图像形式的展示.....	14
3.6	界面上所有的按钮功能.....	17
3.7	界面上所有的滑动条功能.....	17

## 1 Celex Demo Kit 硬件

请按以下顺序安装 CeleX 芯片套件：

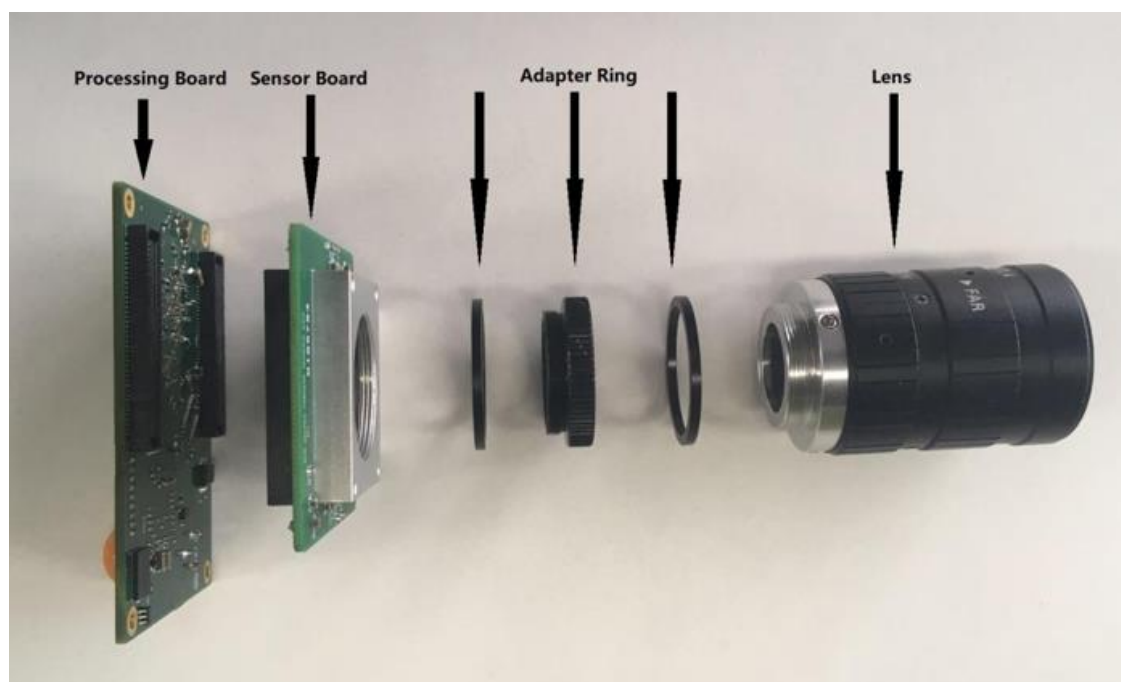


图 1-1 Celex 芯片套件

**备注：**请记得正确地放置适配器环，以保持良好的焦点。（如果没有适配环，请在安装时忽略它们）

### （1） OpalKelly XEM6310 处理板的电源：

XEM6310 要求这个电源是干净的，过滤的，范围在 4.5v 到 5.5v 之间。此电源必须通过两个设备的两个扩展连接器上的+VDC 引脚或直流电源连接器提供。XEM6310 上的直流电源连接器是 CUI, Inc 的部件号 PJ-102AH。它是标准的“canon-style”2.1mm / 5.5mm 插孔，外环连接到 DGND，中心引脚连接到+VDC。

它也可以是 PC 上的 USB3.0 端口。

**备注：**请勿将处理板连接到产生高于 5V 输出电压的任何电源，这会损坏当前的电路板组件。

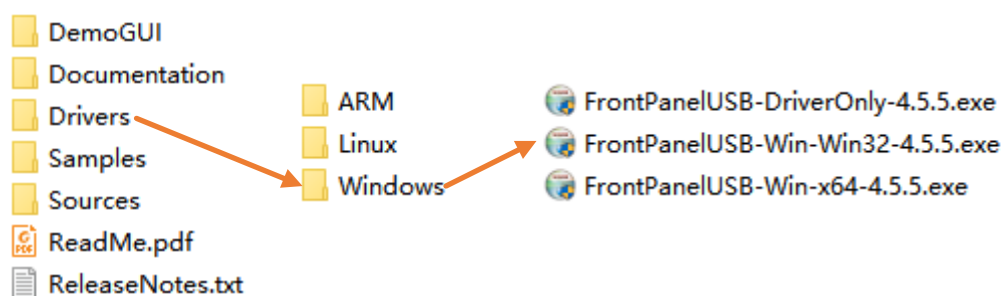
### （2） 数据线必须连接 PC 上的 USB3.0 端口，因为 USB2.0 会降低传输数据的速度。

## 2 Celex Demo Kit SKD 使用步骤

### 2.1 安装 Opal Kelly 驱动

#### 2.1.1 Windows

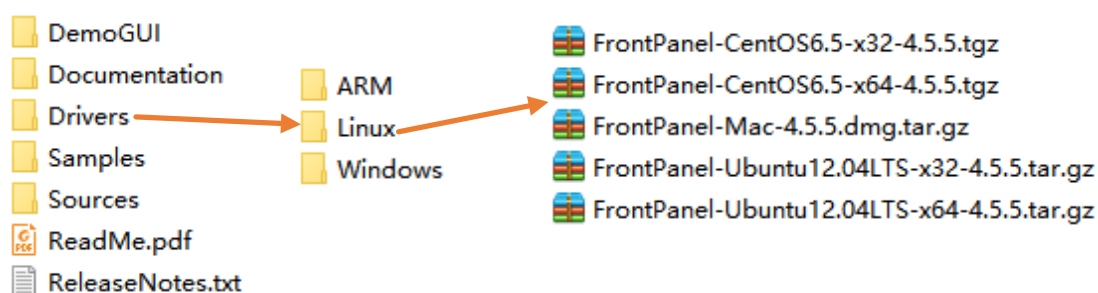
请从以下文件夹安装 Opal Kelly 驱动程序：



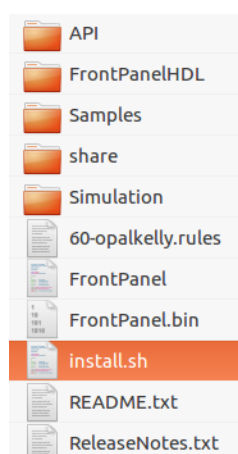
用户可以根据自己的系统（32-bit 或 64-bit）选择相应的驱动来安装。

#### 2.1.2 Linux

请从以下文件夹安装 Opal Kelly 驱动程序：



用户可以根据自己的系统选择相应的驱动来安装，例如，Ubuntu 16.04LTS-x64 的系统，则选择 FrontPanel-Ubuntu12.04LTS-x64-4.5.5.tar.gz，解压后会看到下面的文件：



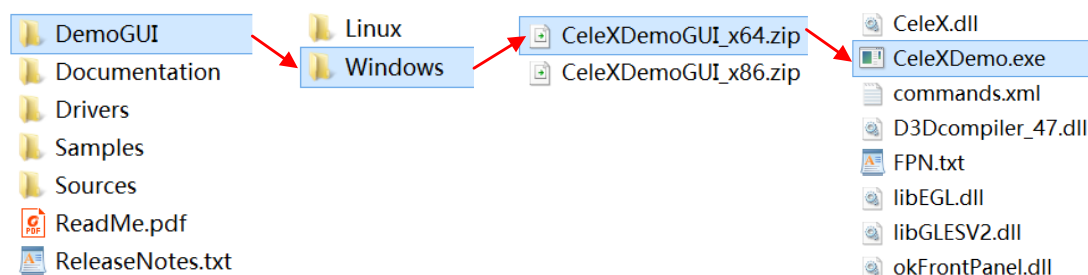
打开终端，进入到 install.sh 所在的目录，执行以下命令安装驱动：

```
sudo sh ./install.sh
```

## 2.2 运行 CeleX Demo GUI

### 2.2.1 Windows

安装驱动程序后，用户可以从以下文件夹打开 Demo GUI：

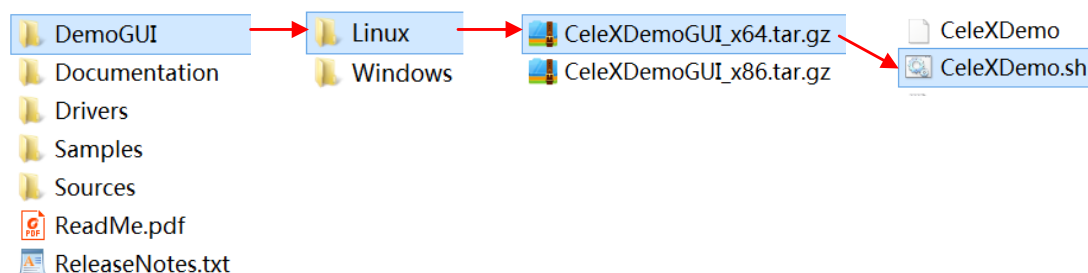


双击“CeleXDemo.exe”即可正常打开 CeleX Demo GUI。打开后的界面如图 3-2 所示（第 3 章）。

**Notes:** 如果 CeleXDemo.exe 无法打开，且 Windows 消息框显示缺失某些 dll 文件，这可能是由于缺少 Visual C ++ 支持包所造成的。可以在 *Driver/Windows* 文件夹下安装“vc\_redist.x86.exe”并再次尝试，则 CeleX Demo 应该可以正常工作。

### 2.2.2 Linux

安装驱动程序后，用户可以从以下文件夹打开 Demo GUI：



打开终端，进入 CeleXDemo.sh 所在的目录，输入以下命令即可打开 CeleX Demo GUI，打开后的界面如图 3-2 所示（第 3 章）。

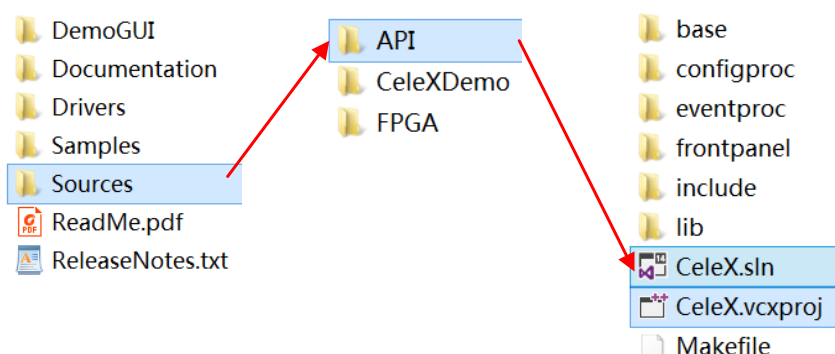
```
$ sh CeleXDemo.sh
```

## 2.3 编译 CeleX SDK 的 Source Code

本 SDK 中会使用 OpenCV 库（版本为 3.3.0），所以在编译源码之前请先安装 OPenCV 库并配置好其编译环境。

### 2.3.1 Windows

在 Window 平台上，我们提供了 VS2015 的工程直接编译该源码，可以按照以下图示进入 SDK 的 Source Code 目录：

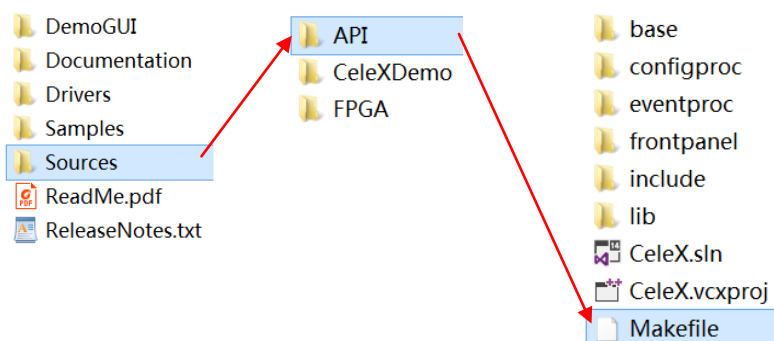


#### 备注：

- (1) 需要修改工程属性中关于 OpenCV 的 Include 和 Lib 的路径的设置，否则会因为找不到 OpenCV 的头文件和库而编译失败。
- (2) 编译完成后，会在工程所在的目录下自动创建一个 build/Windows 目录，编译生成的库文件（CeleX.dll 和 CeleX.lib）会被自动导入到该目录下。

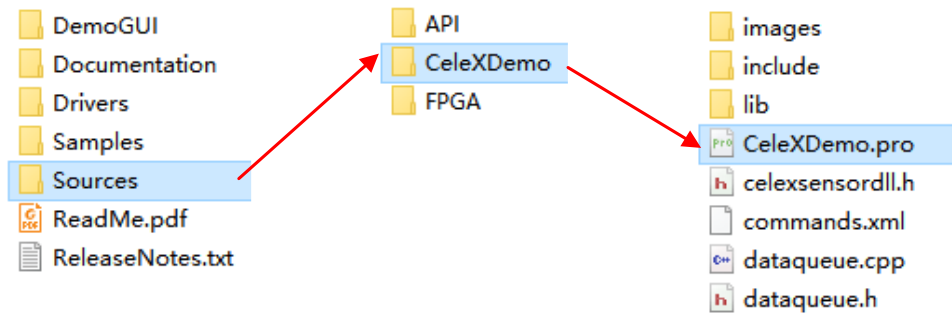
### 2.3.2 Linux

在 Linux 平台上，我们提供一个 Makefile 编译该代码，库文件（libCeleX.so）将生成在当前目录下。



## 2.4 编译 CeleX Demo GUI 的 Source Code

由于本 Demo 是用 Qt 开发的，所以在编译该代码之前要先安装 Qt（本 Demo 使用的 Qt 版本为：qt-opensource-windows-x86-msvc2015\_64-5.6.3.exe）。由于 Qt 也是跨平台的，所以 Windows 和 Linux 平台上，都可以用 Qt Creator 打开 *CeleXDemo.pro* 即可编译。



需要注意的是，由于本 Demo 中也用到了 OpenCV 的一些接口，所以需要修改一下 *CeleXDemo.pro* 文件中关于 OpenCV 的路径设置，如下所示：

```

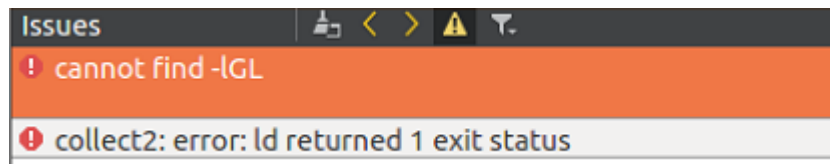
win32 {
    INCLUDEPATH += D:/opencv/build/include \
                  D:/Program Files/opencv/build/include/opencv \
                  D:/Program Files/opencv/build/include/opencv2
}
else {
    INCLUDEPATH += /usr/local/include \
                  /usr/local/include/opencv \
                  /usr/local/include/opencv2

    LIBS += /usr/local/lib/libopencv_highgui.so \
            /usr/local/lib/libopencv_core.so \
            /usr/local/lib/libopencv_imgproc.so \
            /usr/local/lib/libopencv_videoio.so
}
  
```

### 备注：Linux 下可能会遇到的编译错误

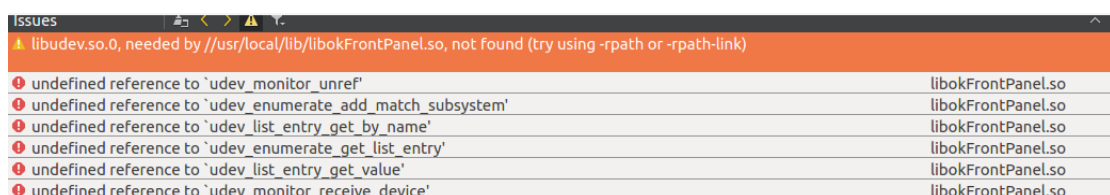
#### (1) OpenGL 错误

如果在编译的过程中，遇到以下错误，则需要安装 OpenGL 库（Qt 依赖 OpenGL 库），否则跳过该步骤。在终端上输入命令：`sudo apt-get install libgl1-mesa-dev`



#### (2) udev 错误

如果在编译的过程中，遇到以下错误，这是因为 Opal Kelly 的驱动 FrontPanel SDK-v4.5.5 只提供了 Ubuntu12.04LTS 的版本，所以当我们在高版本的 Ubuntu 上使用时会遇到“libudev”版本不兼容的问题。



用户可以从以下链接中下载兼容的 libudev，也可以从我们的发布包 (*Drivers/Linux/libudev.zip*) 中直接获取。下载链接：[https://ubuntu.pkgs.org/12.04/ubuntu-main-i386/libudev0\\_175-](https://ubuntu.pkgs.org/12.04/ubuntu-main-i386/libudev0_175-)

[0ubuntu9\\_i386.deb.html](#)

解压 libudev.zip 后会看到以下内容：

```
libudev0_175-0ubuntu9_i386.deb
```

```
libudev0_175-0ubuntu9_amd64.deb
```

安装命令：

```
sudo dpkg -i libudev0_175-0ubuntu9_i386.deb
```

```
sudo dpkg -i libudev0_175-0ubuntu9_amd64.deb
```

## 2.5 生成 FPN 文件

固定模式噪声 (FPN, Fixed Pattern Noise) 是数字图像传感器上的特定噪声模式的术语，在较长的曝光镜头中经常可见，其中特定像素易于在一般背景噪声之上提供较亮的强度。如果不从图像中减去 FPN，则图像可能显示出高水平的背景噪声，因此变得粗糙。为了解决该问题，我们需要为 CeleX Sensor 生成一个 FPN 文件，具体的操作步骤请参考 [3.4](#) 章节。



### 3 Celex Demo Kit GUI 的功能

打开 CelexDemo.exe，当没有连接 Sensor 设备时，界面如图 3-1 所示；当有 Sensor 设备连接时，界面如图 3-2 所示。

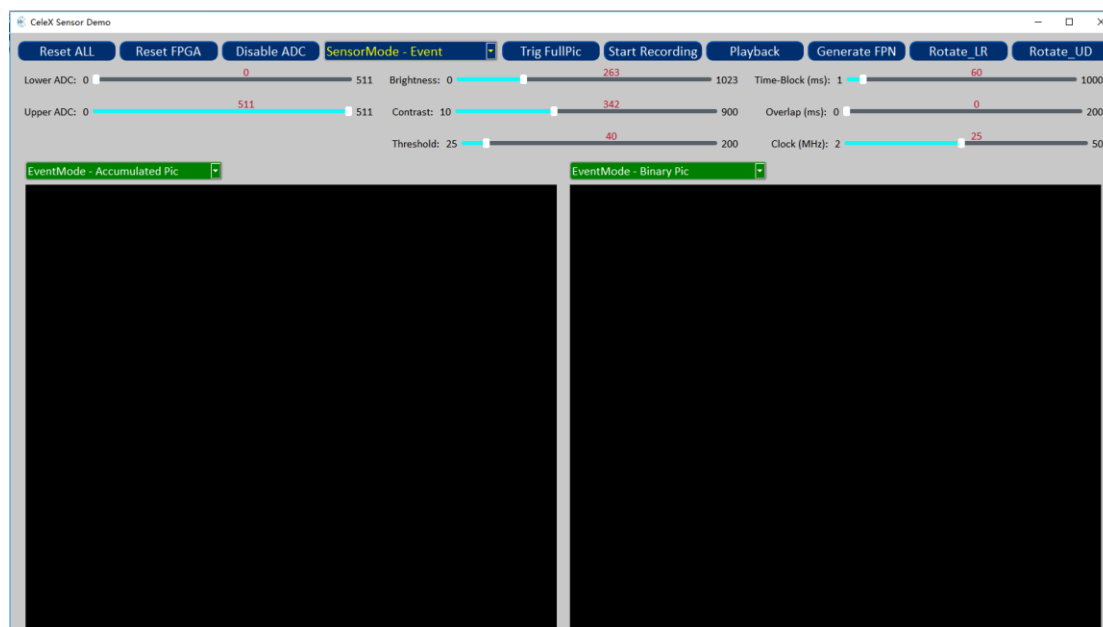


图 3-1

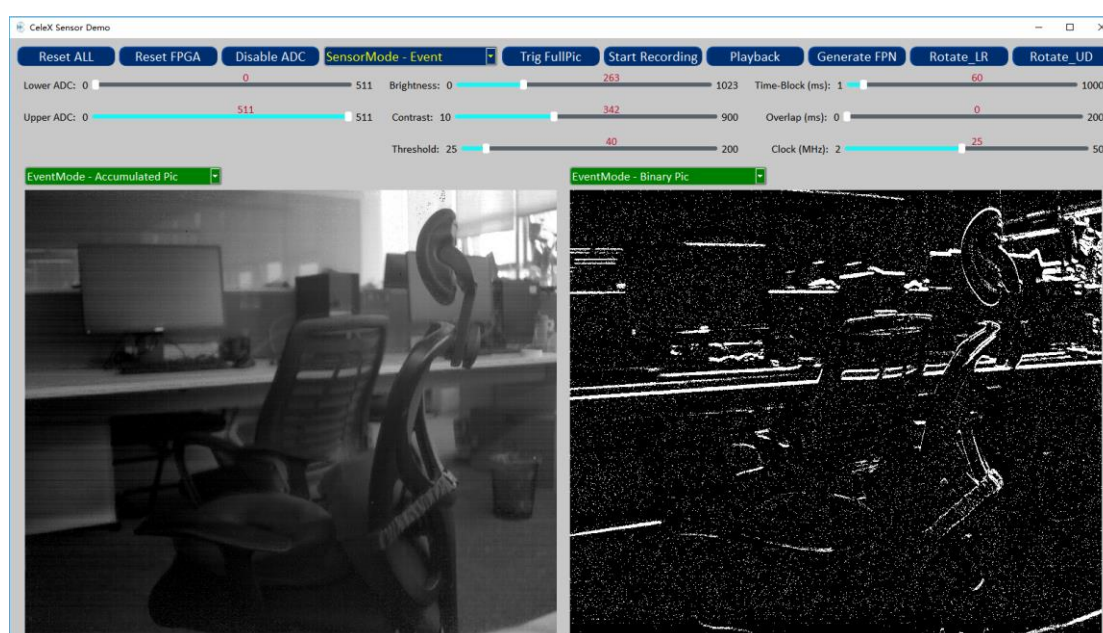


图 3-2

#### 3.1 切换 Sensor 工作模式

点击红色框标记的组合框按钮可以切换 Sensor 的工作模式，图 3-3 所示的是 Sensor 工作在 Event 模式下的画面展示，其中左边的图像显示是 Event 模式的累加的 Full Picture 图像，右边的图像显示的是动态图像。



图 3-3 Sensor 工作在 Event 模式

图 3-4 给出了 Sensor 工作在 Full-Picture 模式下的画面展示，其中左边的图像即是 Full Picture 图像，由于该模式下只有 Full Picture 图像，所以右边的图像为黑色，表示没有图像输出。

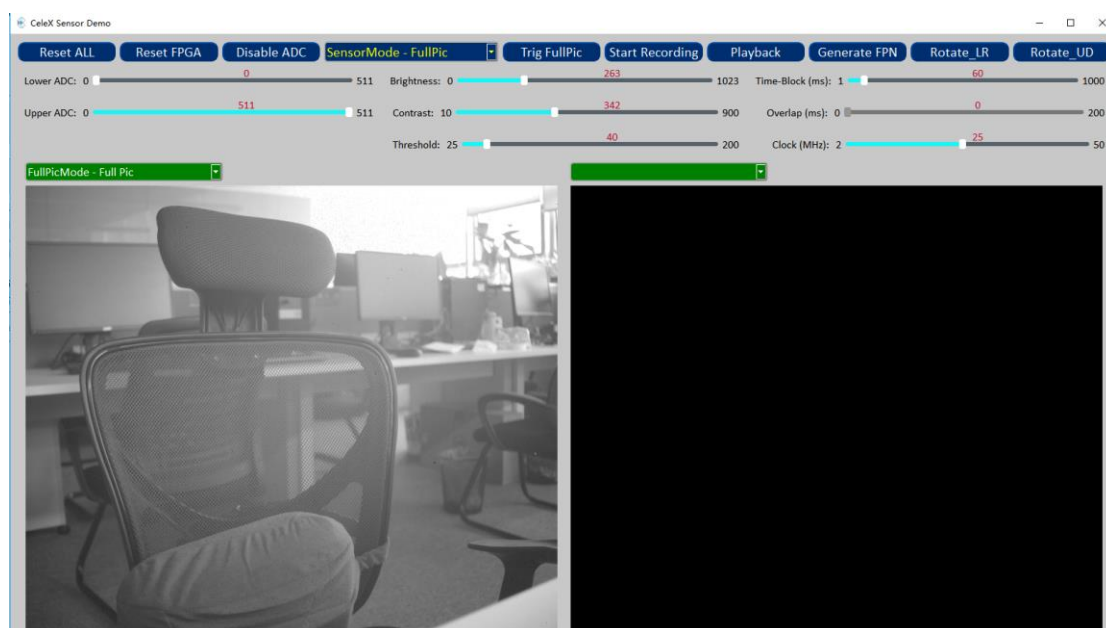


图 3-4 Sensor 工作在 Full-Picture 模式

图 3-5 给出了 Sensor 工作在 FullPic\_Event 模式下的画面展示，其中左边的图像即是 Full Picture 图像，右边的图像是动态图像。

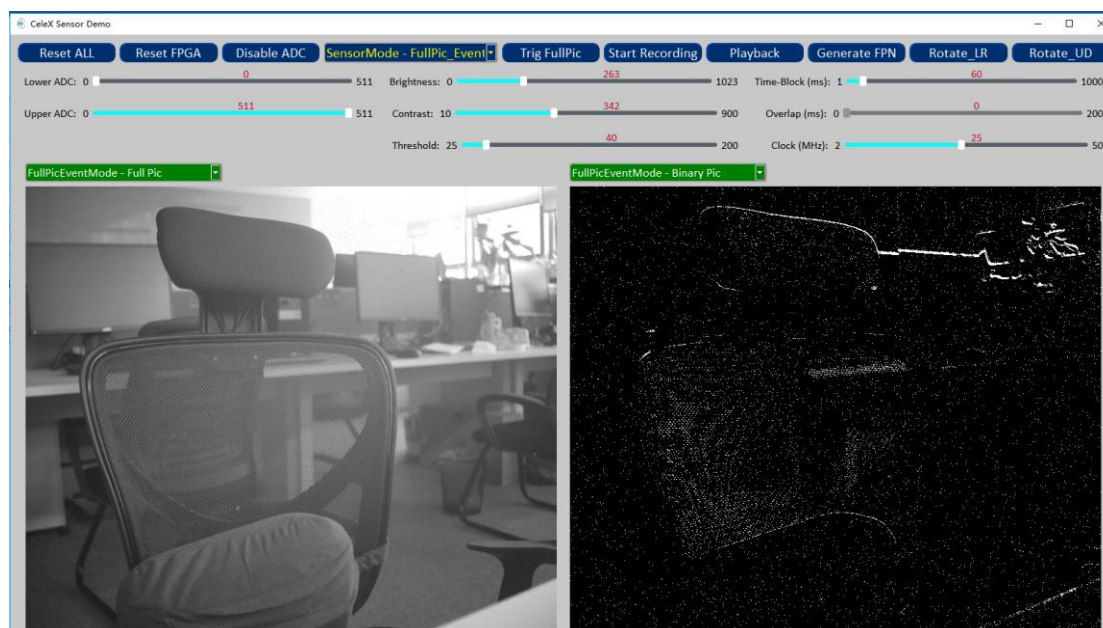


图 3-5 Sensor 工作在 FullPic\_Event 模式

### 3.2 录制 Sensor 数据功能

点击图 3-6-1 中“Start Recording”按钮即可开始录制 bin 数据，开始录制数据后，按钮上的文字会变成图 3-6-2 所示的“Stop Recording”，那点击“Stop Recording”按钮即停止录制 bin 数据。录制的 bin 文件就存在 CeleXDemo.exe 的同目录下，以 Recording\_YYYYMMDD\_HHMMSSSSS\_SensorMode\_ClockRate.bin 的形式命名，如下所示：

```

Recording_20180527_154146463_E_25MHz.bin
Recording_20180527_154150534_F_25MHz.bin
Recording_20180527_154153694_FE_25MHz.bin
    
```

其中，E 表示录制的是 Event 模式下的数据，F 是 FullPic 模式下的数据，FE 是 FullPic\_Event 模式下的数据。25MHz 表示录制数据是，Sensor 的工作频率为 25MHz。

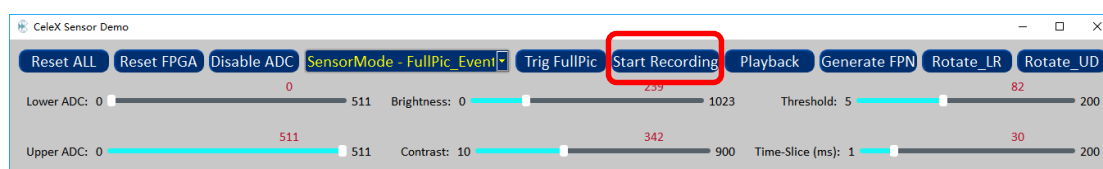


图 3-6-1

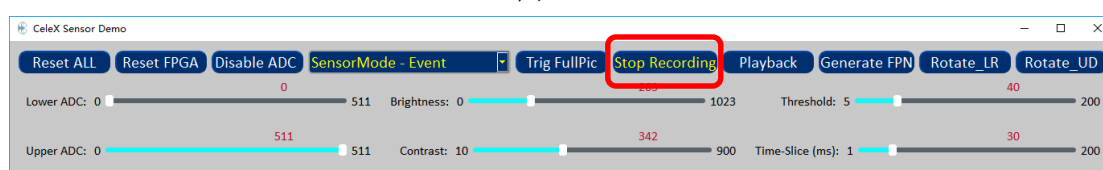


图 3-6-2



### 3.3 播放录制的 Bin 文件功能

点击“**Playback**”按钮，选择一个 bin 文件播放，Playback 时的界面展示如图 3-7 所示。其中左右两边图像显示的内容，跟你录制 bin 数据时的 Sensor 模式有关。关于各种模式下有哪些图像输出，请参考 3.1 章节的描述。

在播放 bin 文件时，可以调节**滑动条 Time-Slice** 改变建帧的时间，也可以调节**滑动条 Display** 改变播放每一帧的时间间隔。

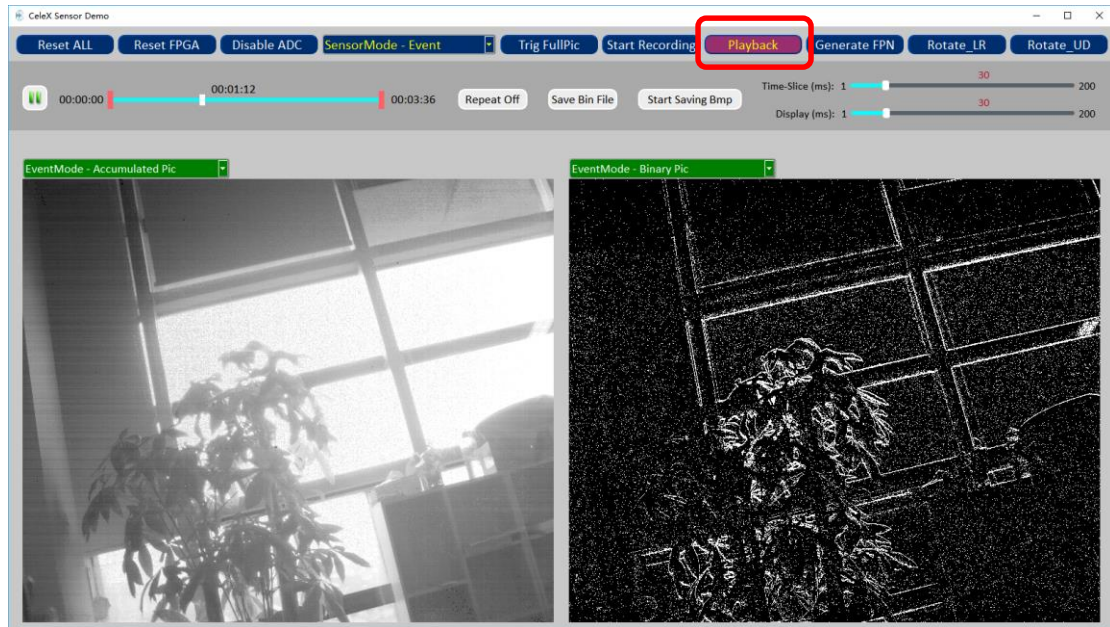
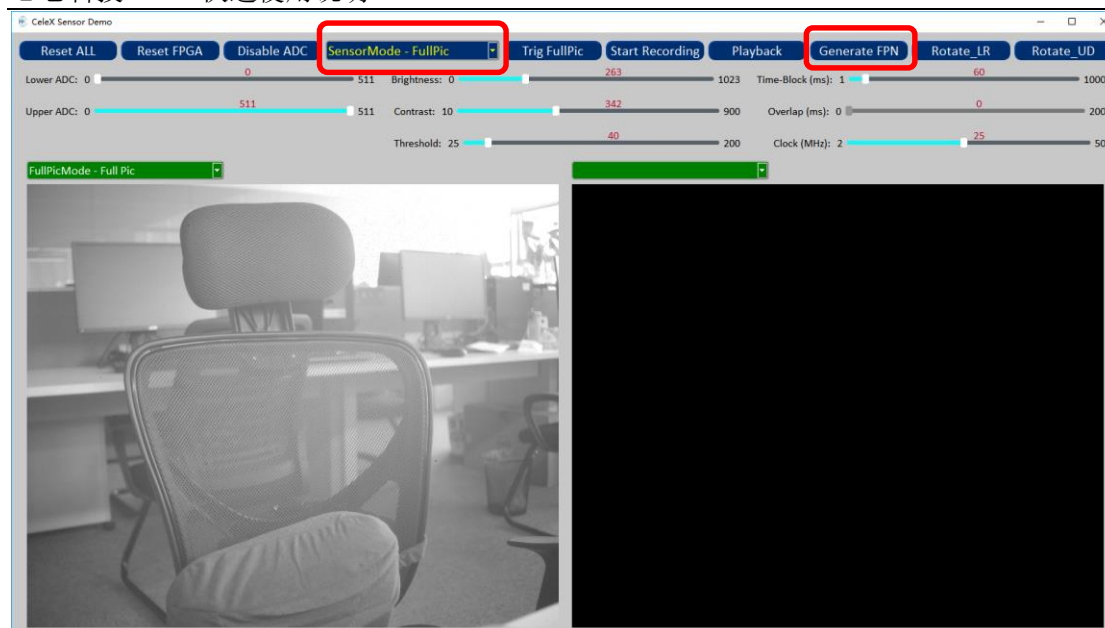


图 3-7

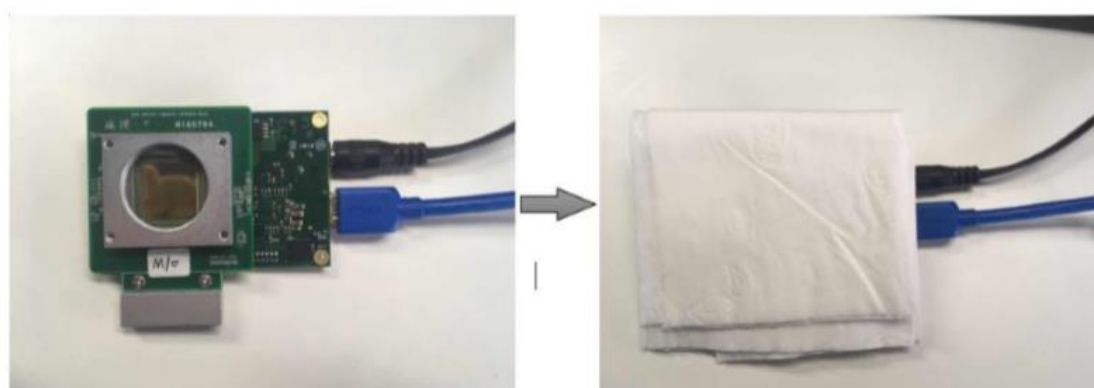
### 3.4 生成 FPN 功能

固定模式噪声 (FPN, Fixed Pattern Noise) 是数字图像传感器上的特定噪声模式的术语，在较长的曝光镜头中经常可见，其中特定像素易于在一般背景噪声之上提供较亮的强度。如果不从图像中减去 FPN，则图像可能显示出高水平的背景噪声，因此变得粗糙。为了解决该问题，我们需要为 CeleX Sensor 生成一个 FPN 文件，具体的操作步骤如下：

- (1) 将 Sensor 的工作模式切换至 Full-Picture 模式。你将只会看到左边的屏幕显示图片，右边的屏幕将被关闭。



- (2) 由于 FPN 生成操作必须在光照均匀的环境下进行，所以可以通过取下光学镜头并用一张白纸（薄纸或 A4 打印纸）覆盖裸露的 Sensor 来实现这种情况。确保纸张均匀地完全覆盖传感器，并且纸张保持静止。**备注：如果你是在阳光下而不是 LED 灯下操作，效果会更好。**



- (3) 执行 FPN 生成操作之前，请检查图像屏幕，确保其显示正常，不要太暗或太亮。只需在裸露的 Sensor 上放置更多或更少的纸张，或者在 GUI 窗口上打开或关闭“亮度”滑块，即可更改照明。**备注：下图中的第 3 幅图就是正常亮度的图。**

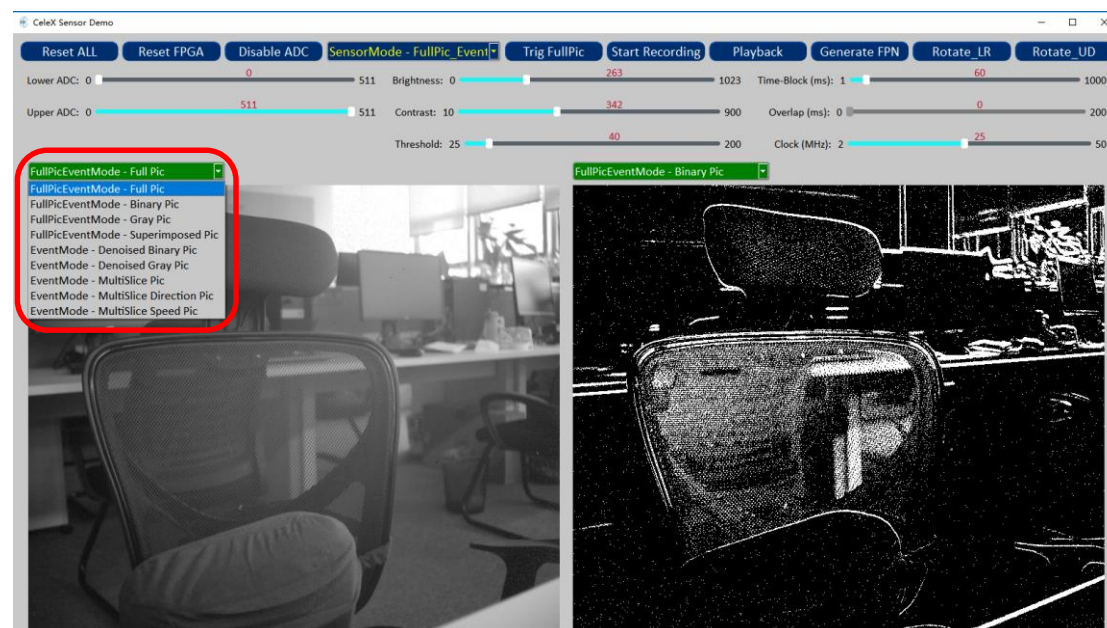


- (4) 点击 GUI 窗口中的“Generate FPN”按钮，当你在指定目录下看到 FPN.txt 文件时表明 FPN 文件成功生成了。
- (5) 重启应用程序后将使用新的 FPN 文件，您应该能够看到图像质量的差异。

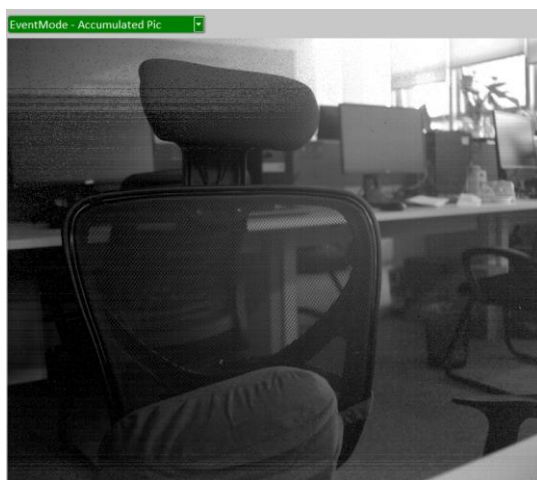


### 3.5 各种图像形式的展示

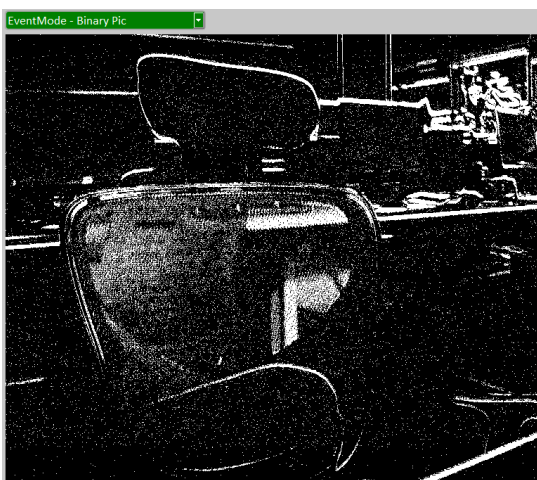
我们以 Event 模式为例，来说明各种图像形式的展示。可以通过点击下图中的组合框按钮来设置要显示的图像类型，具体如下：



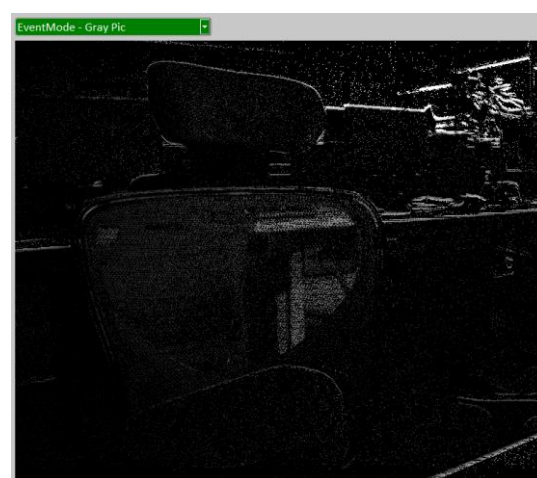




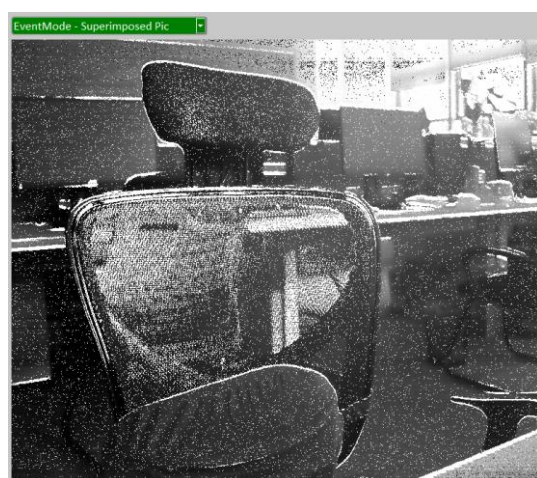
Accumulated Pic



Binary Pic



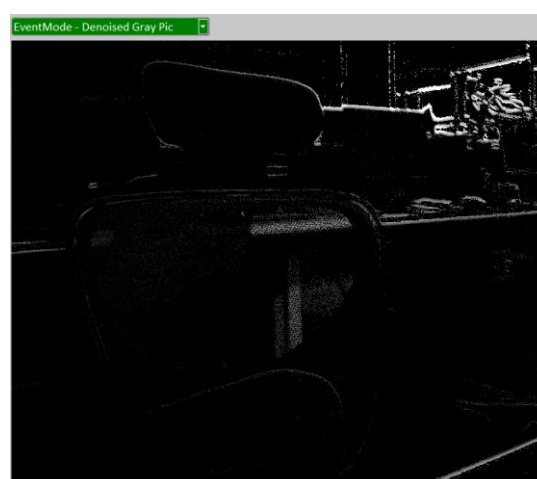
Gray Pic



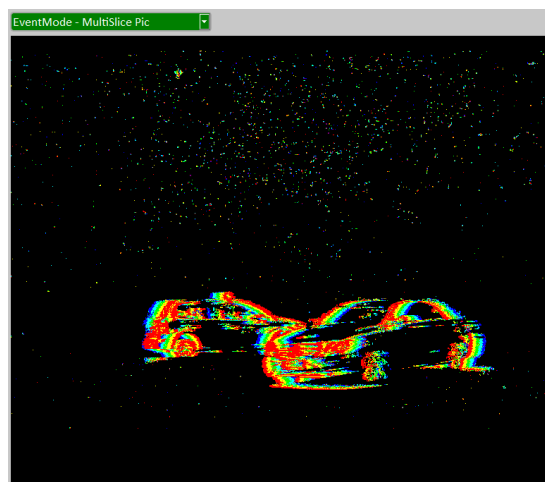
Superimposed Pic



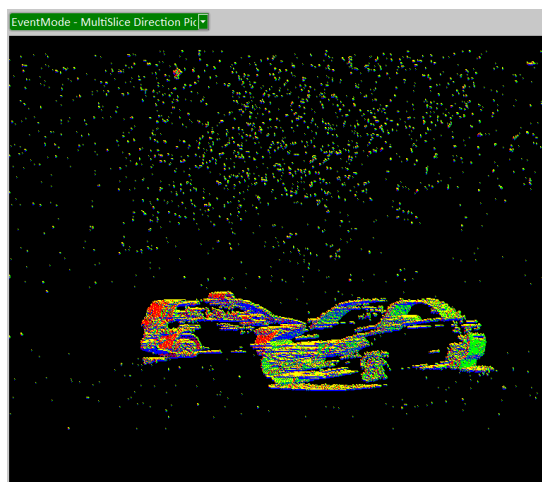
Denoised Binary Pic



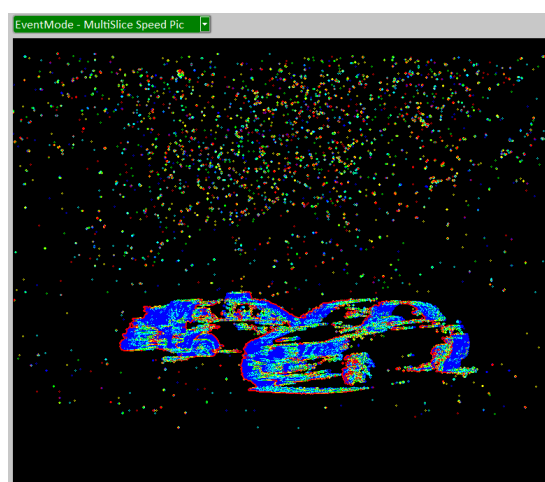
Denoised Gray Pic



Optical-Flow Pic



Optical-Flow Direction Pic



Optical-Flow Speed Pic

**备注：**关于 Optical-Flow 的功能介绍请参考我们 API 手册中的相关章节。

Optical-Flow Pic 中用了 5 中颜色（红，黄，绿，蓝绿，蓝）表示光流信息，从红色到蓝色表示时间上的信息是由新到旧，其中红色表示的是最新的信息（终点），蓝色表示的是最旧的信息（起点）。

Optical-Flow Direction Pic 中用了 4 种颜色（红，黄，绿，蓝）表示物体运动的方向，红色表示向右运动，绿色表示向左运动，黄色表示向上运动，蓝色表示向下运动。

Optical-Flow Speed Pic 中用了 5 种颜色（红，黄，绿，蓝绿，蓝）表示物体运动的速度，从红色到蓝色表示速度由慢变快，其中红色表示的速度最慢，蓝色表示的速度最快。



### 3.6 界面上所有的按钮功能

No.	名称	功能
1	Reset ALL	Reset Sensor 和 FPGA
2	Reset FPGA	Reset FPGA
3	Disable ADC	启用和禁用 Event 模式的下的灰度值数据功能
4	组合框 SensorMode - Event	切换 Sensor 的工作模式
5	Trig FullPic	触发生成一帧 Full Picture 图像
6	Start Recording	录制 Sensor 数据
7	Playback	播放录制的 Sensor 数据
8	Generate FPN	生成 FPN 文件
9	Rotate_LR	左右旋转显示的图像（同时控制左右两幅图像）
10	Rotate_UP	上下旋转显示的图像（同时控制左右两幅图像）

### 3.7 界面上所有的滑动条功能

No.	名称	功能
1	滑动条 Lower ADC	调节可变亮度范围的下限
2	滑动条 Upper ADC	调节可变亮度范围的上限
3	滑动条 Brightness	调节 Sensor 输出的图像亮度
4	滑动条 Contrast	调节 Sensor 输出的图像对比度
5	滑动条 Threshold	调节触发 Event 的阈值（当像素点的光强变化大于此阈值时，该值越小，触发的 Event 数据越多，反之，则触发的 Event 数据越少）
6	滑动条 Frame-Time	当 Sensor 工作在 Event 模式下，调节该滑动条可以修改建立动态图像帧的时长（值越大，建帧的时间就越长；反之，建帧的时间就越短）；当 Sensor 工作在 FullPic_Event 模式，调节该滑动条可以修改每一个 Frame Time 里面 Motion Event 的输出时间