



celepixel

# 芯仑科技 SDK 使用说明

芯仑科技（上海）有限公司



## 目录

1	CeleX-5 SDK 使用步骤.....	3
1.1	安装 MIPI 转 USB3.0 驱动.....	3
1.1.1	Windows.....	3
1.1.2	Linux.....	4
1.2	运行 CeleX-5 Demo GUI .....	5
1.2.1	Windows.....	5
1.2.2	Linux.....	5
1.3	编译 CeleX-5 API 的 Source Code .....	6
1.3.1	Windows.....	6
1.3.2	Linux.....	6
1.4	编译 CeleX-5 Demo GUI 的 Source Code.....	7
1.5	编译 CeleX-5 Driver 的 Source Code .....	8
1.5.1	Windows.....	8
1.5.2	Linux.....	8
1.6	生成 FPN 文件 .....	9
2	CeleX-5 Demo Kit GUI 的功能.....	10
2.1	Sensor 的工作模式介绍.....	10
2.2	Sensor 的模式切换 (Fixed Mode) .....	11
2.3	录制 Sensor 数据功能.....	13
2.4	播放录制的 bin 文件功能.....	14
2.5	Configurations.....	15
2.5.1	基本控制参数 (Basic Controls) .....	15
2.5.2	设备信息 (Device Info) .....	16
2.5.3	其他设置参数 (Others) .....	17
2.6	采集 FPN 功能 .....	18
2.7	图像翻转功能.....	20
2.8	bin 文件转视频功能.....	21
2.9	bin 文件转 CSV 文件功能.....	22

# 1 CeleX-5 SDK 使用步骤

## 1.1 安装 MIPI 转 USB3.0 驱动

### 1.1.1 Windows

把 CeleX-5 Sensor 通过 USB 线连接到 PC 上，双击 **zadig-2.4.exe** 弹出如下图 1-1 所示的界面，选择 Options → List All Devices (图 1-2)，然后选择设备 FX3 (图 1-3)，点击 Install Driver 或 Reinstall Driver (图 1-4) 安装驱动，安装成功后会弹出图 1-5 所示的界面。

**备注：**数据线必须连接 PC 上的 USB3.0 端口。

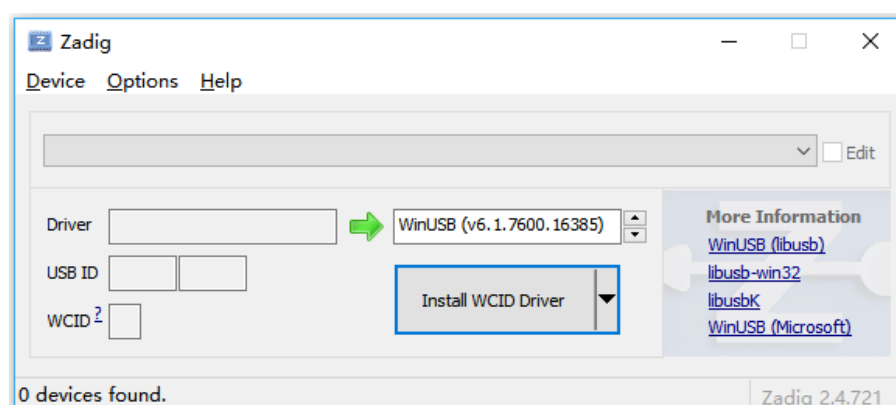


图 1-1

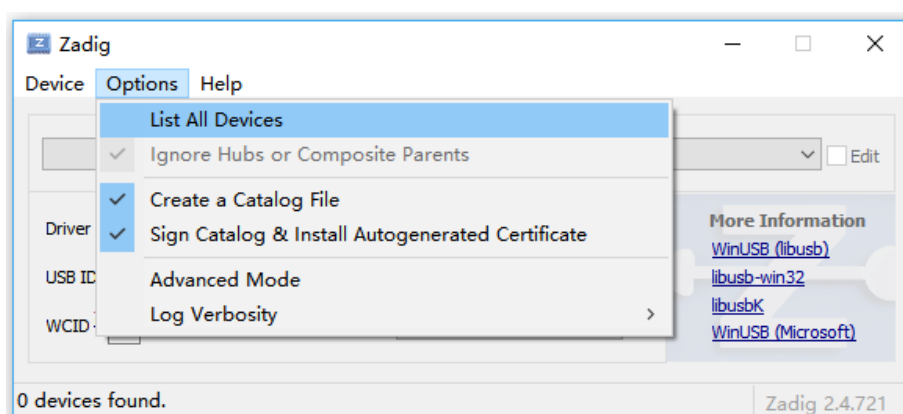


图 1-2

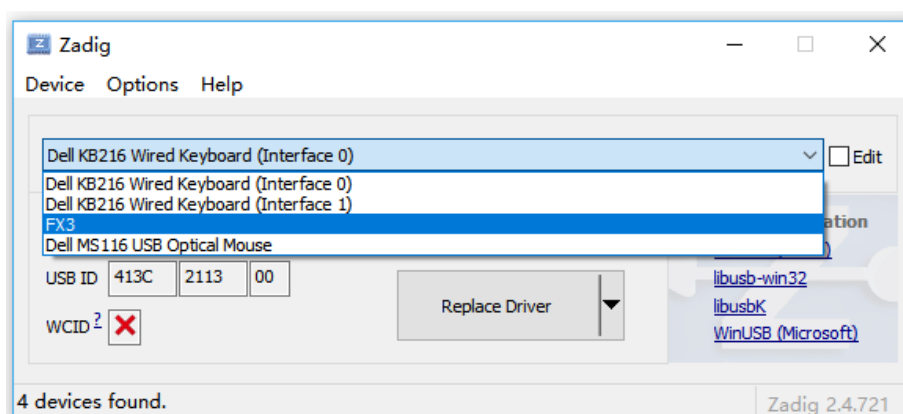


图 1-3

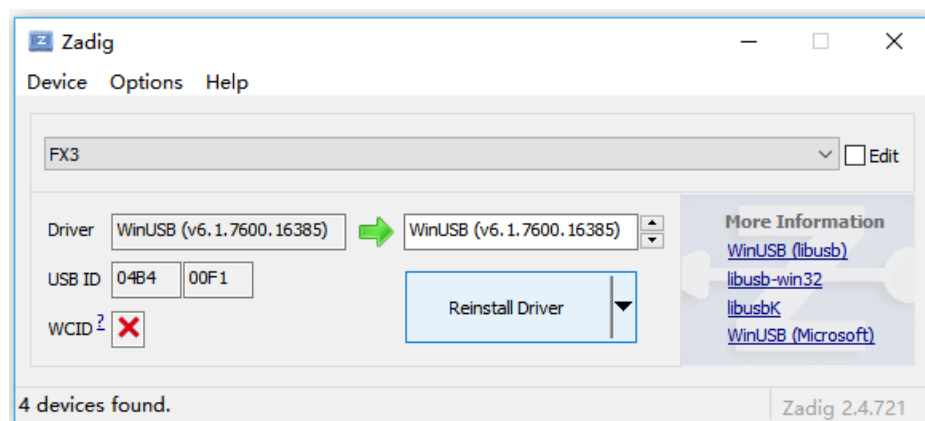


图 1-4

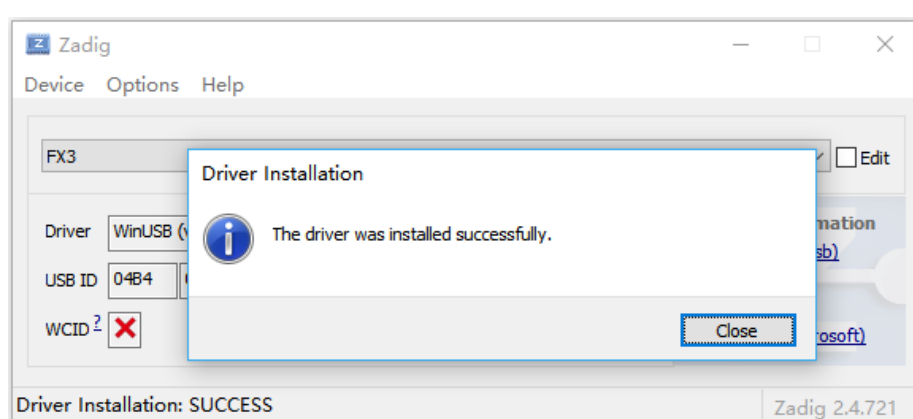


图 1-5

### 1.1.2 Linux

Linux 下安装 CeleX-5 Sensor 所需的驱动,将发布包 *Drivers/Linux/*目录下的压缩包解压,如图 1-6 所示,运行命令“*sudo sh install.sh*”即可进行安装,如图 1-7。

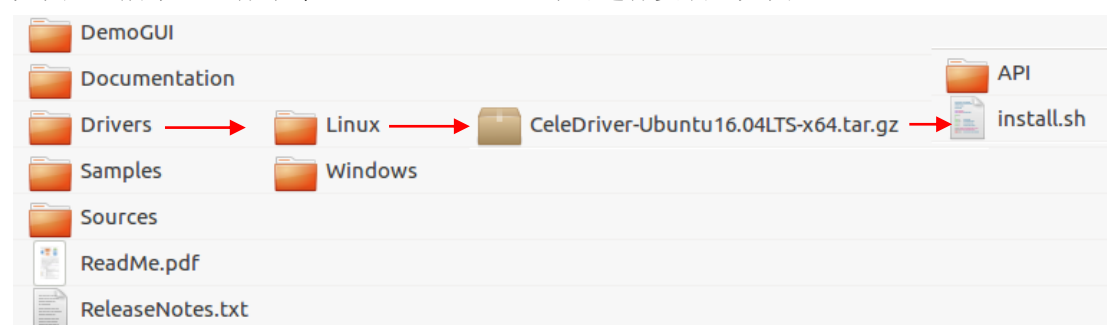


图 1-6

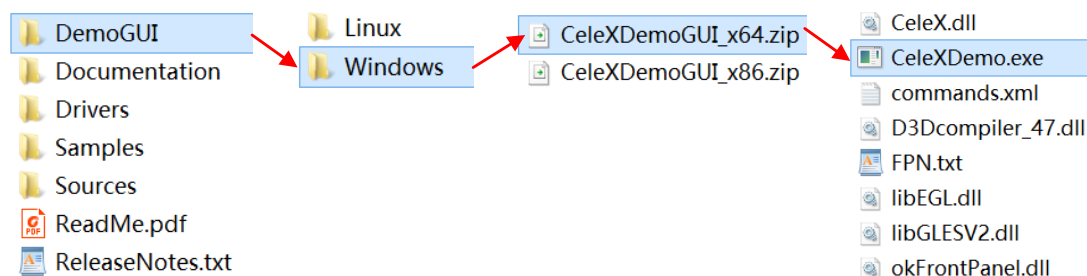
```
sudo sh ./install.sh
```

图 1-7

## 1.2 运行 CeleX-5 Demo GUI

### 1.2.1 Windows

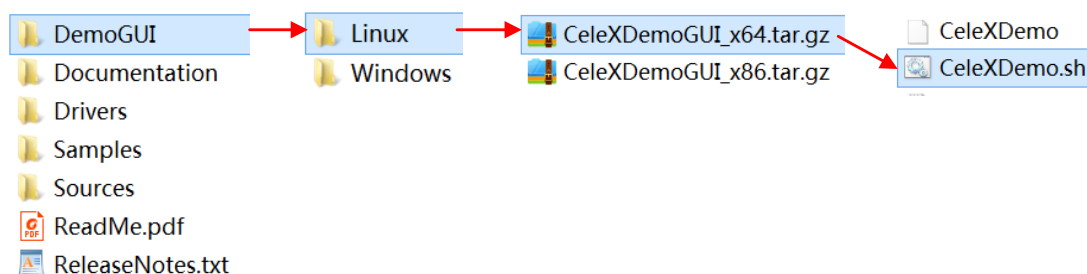
安装驱动程序后，用户可以从发布文件夹打开 Demo GUI，双击“CeleXDemo.exe”即可正常打开 CeleX Demo GUI。打开后的界面如图 2-2 所示（第 2 章）。



**备注：**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 *Drivers/Windows* 文件夹下安装“vc\_redist.x64.exe”并再次尝试，则 CeleX Demo 应该可以正常工作。

### 1.2.2 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，输入命令“*sudo sh CeleXDemo.sh*”，即可打开 CeleX-5 Demo GUI，打开后的界面如图 2-2 所示（第 2 章）。



```
$ sudo sh CeleXDemo.sh
```

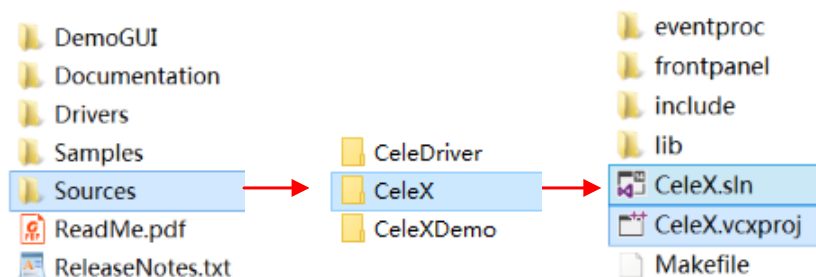
**备注：**这里要用 root 权限打开 Demo，因为我们需要对 usb driver 进行读写操作，没有 root 权限可能会造成打开 usb 设备失败的问题。

## 1.3 编译 CeleX-5 API 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。

### 1.3.1 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：

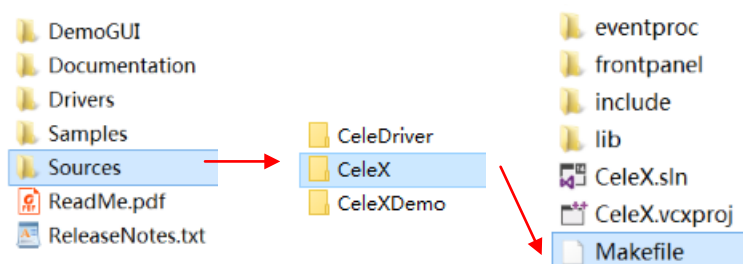


#### 备注：

- (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。

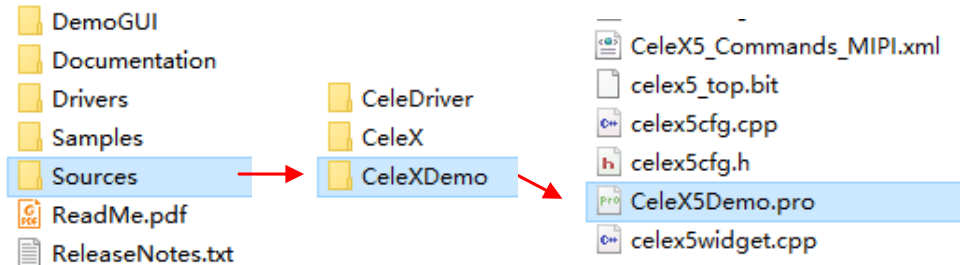
### 1.3.2 Linux

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。



## 1.4 编译 CeleX-5 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：

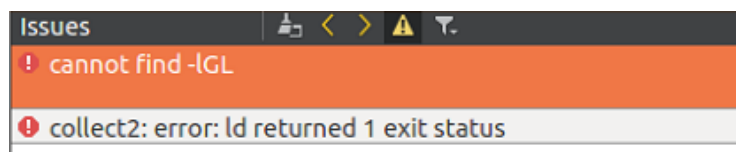
```
win32 {
    INCLUDEPATH += D:/opencv/build/include \
                  D:/Program Files/opencv/build/include/opencv \
                  D:/Program Files/opencv/build/include/opencv2
}
else {
    INCLUDEPATH += /usr/local/include \
                  /usr/local/include/opencv \
                  /usr/local/include/opencv2

    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) OpenCV 版本不兼容的问题（本 SDK 中使用的 OpenCV 版本为 3.3.0）

如果在编译的过程中，遇到该问题，则需要先编译一下 API 库（详见 1.3 章节），然后把编译好的 libCeleX.so 文件，替换掉 ../Sources/CeleXDemo/lib/Linux/x64/ 目录下的 libCeleX.so，再重新编译即可解决该问题。

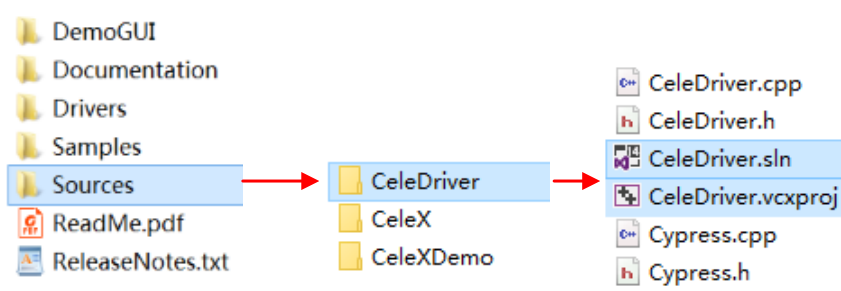
Sources > CeleXDemo > lib > Linux > x64	
名称	修改日期
 libCeleDriver.so	2019/3/15 18:16
 libCeleX.so	2019/3/15 18:16

## 1.5 编译 CeleX-5 Driver 的 Source Code

CeleX-5 Driver 用于获取 USB 端的数据，SDK 再通过 CeleX-5 Driver 来获取数据进行后续处理。

### 1.5.1 Windows

在 Windows 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 Driver 的 Source Code 目录：

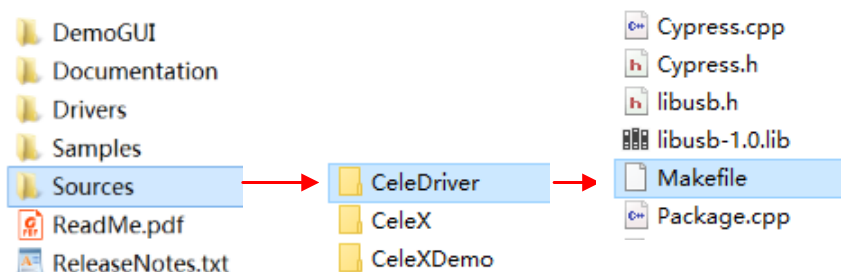


#### 备注：

- (1) 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（CeleDriver.dll 和 CeleDriver.lib）会被自动导入到该目录下。

### 1.5.2 Linux

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







## 1.6 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-5 Sensor 生成一个 FPN 文件, 具体的操作步骤请参考 [2.6](#) 章节。

## 2 CeleX-5 Demo Kit GUI 的功能

### 2.1 Sensor 的工作模式介绍

本 SDK 提供了 Sensor 的 5 种工作模式，用户可以通过 Demo GUI 在这 5 种模式中任意切换，下图给出了每个模式的名称以及对应的图像。



Event Address Only Mode



Event Optical-flow Mode



Event Intensity Mode



Full-frame Picture Mode

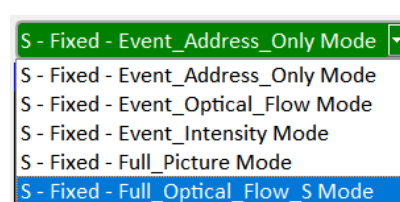
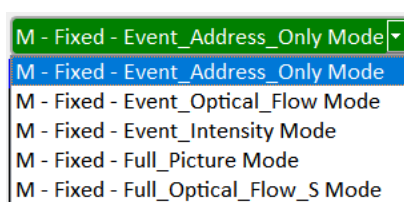
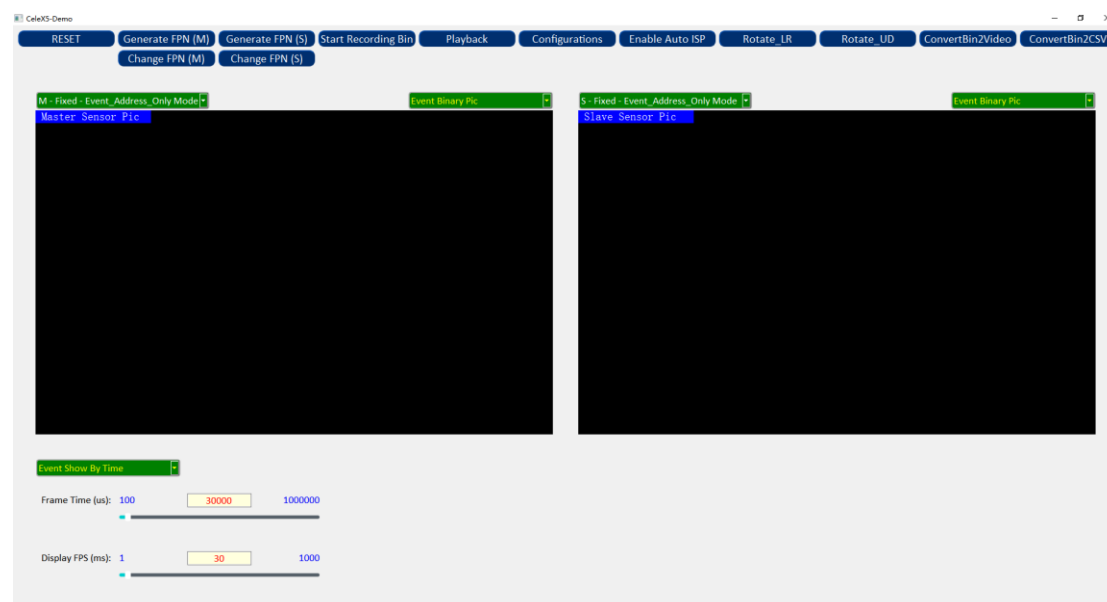


Single Full-frame Optical-flow Mode

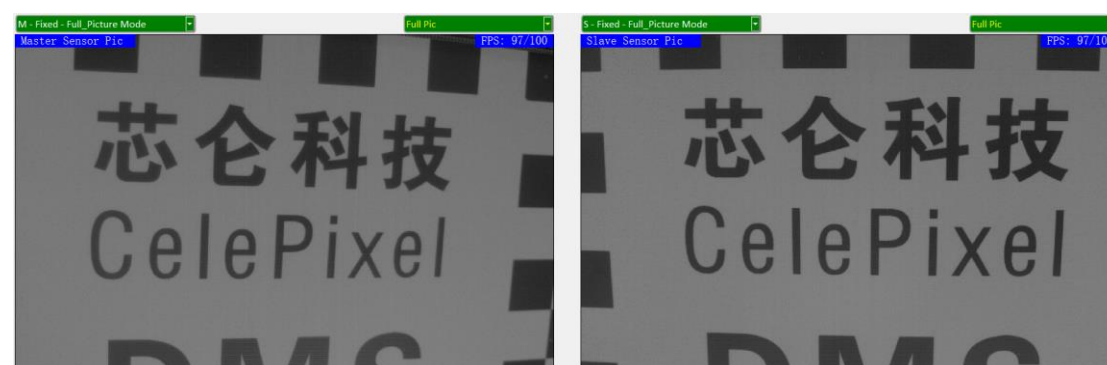
## 2.2 Sensor 的模式切换（Fixed Mode）

由于本文档是针对 2 台 Sensor 的，所以打开 CeleXDemo.exe 之后的 GUI 画面与之前发布的版本会有所不同。

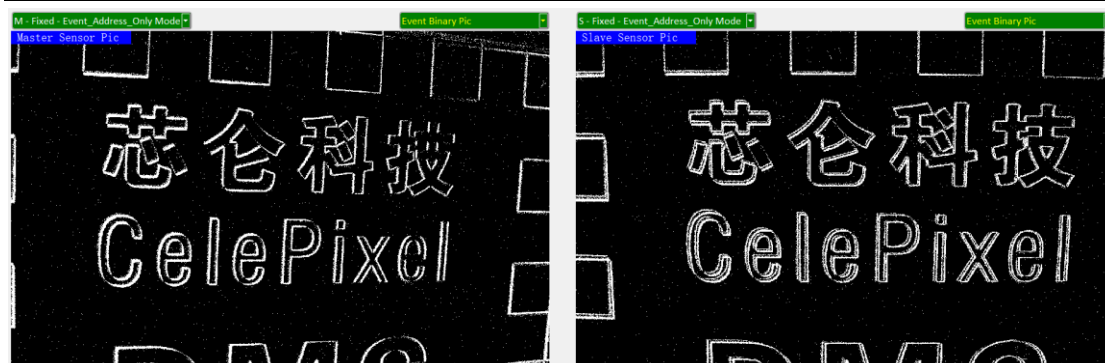
如下图所示，左边显示是 Master Sensor 的图像，右边显示的是 Slave Sensor 的图像，至于如何设置 Master Sensor 和 Slave Sensor，将会在 [2.5.2](#) 章节里介绍。



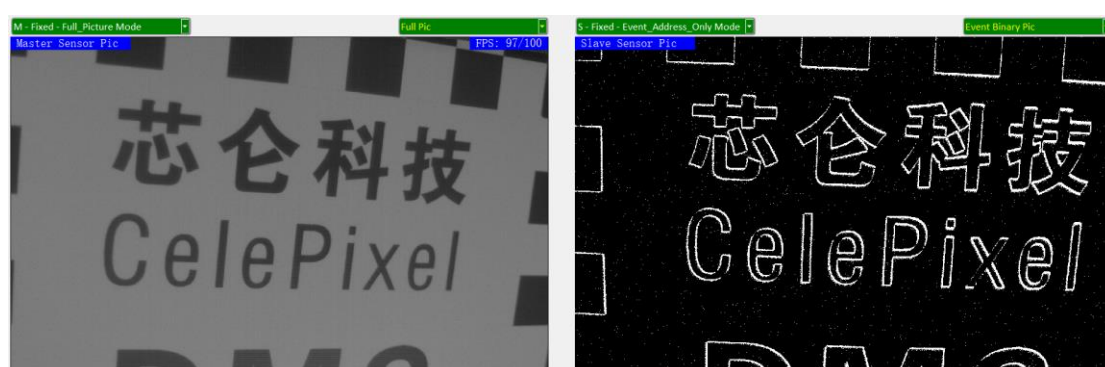
用户可以通过点击上图所示的选择框来切换模式，左边对应的是 Master Sensor，右边对应的是 Slave Sensor。下面将给出几组 2 个 Sensor 工作在不同模式下的画面展示：



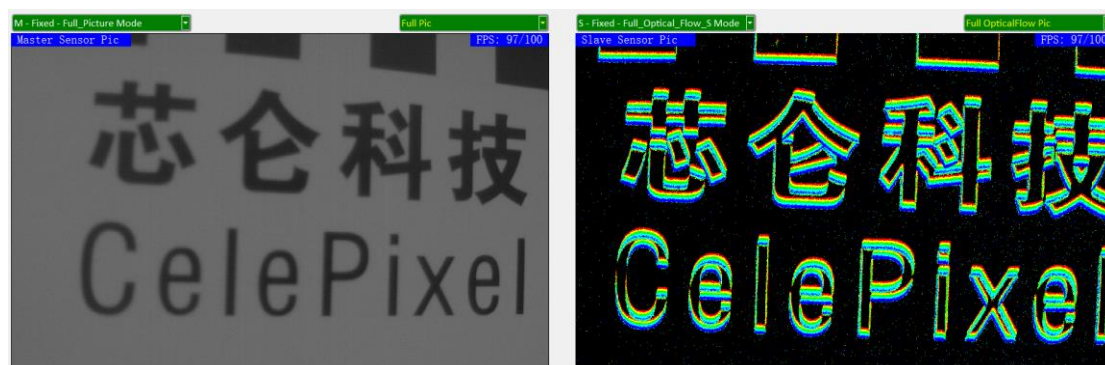
两个 Sensor 同时工作在 *Full-frame Picture* 模式



两个 Sensor 同时工作在 *Event Address Only* 模式



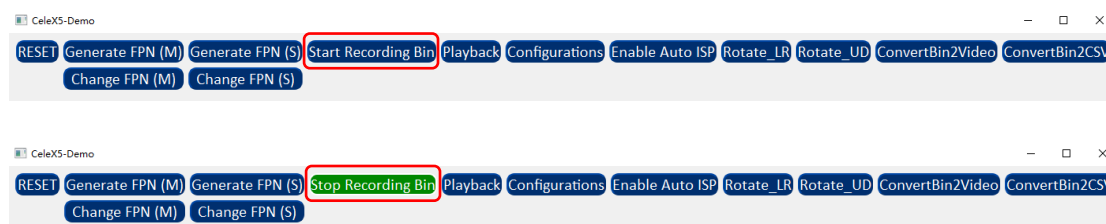
两个 Sensor 分别工作在 *Full-frame Picture* 和 *Event Address Only* 模式



两个 Sensor 分别工作在 *Full-frame Picture* 和 *Single Full-frame Optical-flow* 模式

## 2.3 录制 Sensor 数据功能

点击 “**Start Recording Bin**” 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 “**Stop Recording Bin**”，那点击“**Stop Recording Bin**”按钮即停止录制 bin 数据。



如果连接了 2 台 Sensor，那么就会**同时录制 2 台 Sensor 的数据**，录制的 bin 文件就存在 CeleXDemo.exe 的同目录下。其中 Master Sensor 录制的 bin 文件名称为：

MipiData\_YYYYMMDD\_HHMMSSSSS\_SensorMode\_ClockRate\_M.bin

Slave Sensor 录制的 bin 文件名称为：

MipiData\_YYYYMMDD\_HHMMSSSSS\_SensorMode\_ClockRate\_S.bin

```

MipiData_20190410_170219546_F_100M_M.bin
MipiData_20190410_170219546_F_100M_S.bin
MipiData_20190410_184853568_E_100M_M.bin
MipiData_20190410_184853568_E_100M_S.bin
    
```

其中，E 表示录制的是 *Event Address Only* 模式下的数据，F 是 *Full-frame Picture* 模式下的数据，下面给出了其他几个模式下的缩写：

- (1) *Event Optical-flow* 模式：用 EO 表示
- (2) *Event Intensity* 模式：用 EI 表示
- (3) *Single Full-frame Optical-flow* 模式：用 FO1 表示

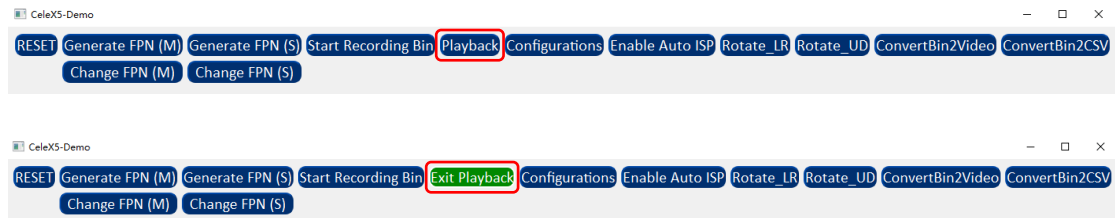
M 表示 Master Sensor 或是第 1 个被识别出来的 Sensor，S 表示 Slave Sensor 或是第 2 个被识别出来的 Sensor。

100MHz 表示 Sensor 的工作频率为 100MHz。

## 2.4 播放录制的 bin 文件功能

Playback 这个功能既支持播放单个 bin 文件，也支持同时播放两个 bin 文件，具体操作如下：

点击“**Playback**”按钮，选择一个 bin 文件或两个 bin 文件，开始播放后，按钮上的文字会“**Exit Playback**”，那点击“**Exit Playback**”按钮即会停止播放 bin 文件并退出 playback 的界面，回到实时显示的界面。



下图是同时播放两个 bin 文件时的界面，其中，左边的图像显示的是第一个 bin 文件的数据，右边的图像显示的是第二个 bin 文件的数据。



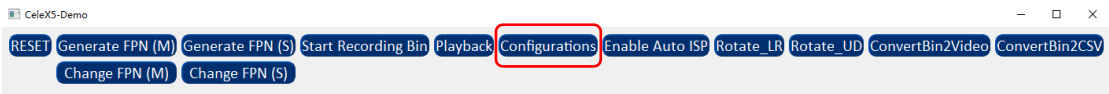
Playback 时还支持“Replay”，“Play/Pause”，保存图片（分为从当前播放位置开始保存图片 and 从头开始保存图片）等功能，操作按钮在上图所示的黄色框区域。

用户可以选择该 bin 文件的各种 Pic 模式进行显示(上图中红色框所示)，也可以修改帧时长（Frame Time）或是刷新频率（Display FPS）等参数(上图中蓝色框所示)。

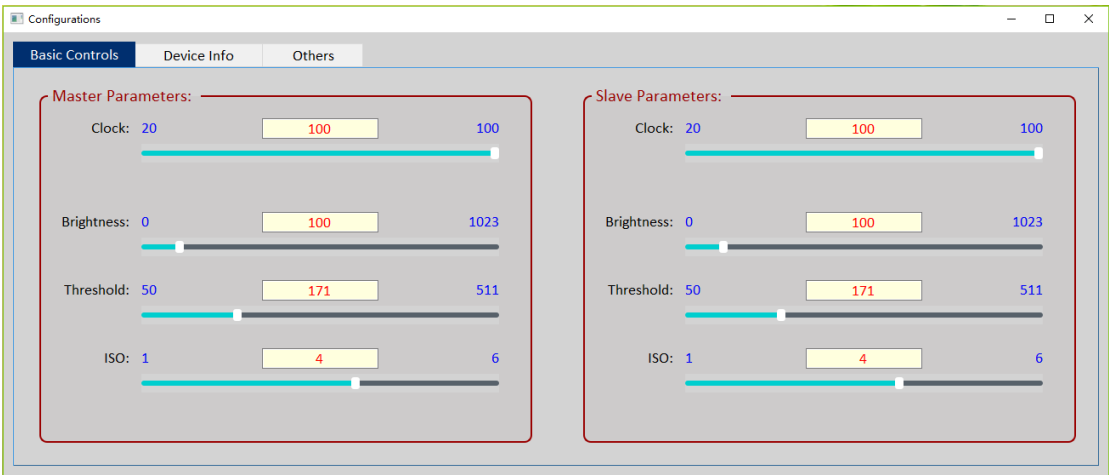


2.5 Configurations

点击 “*Configurations*” 按钮，打开配置界面，可以对 Sensor 和硬件功或软件功能进行一些配置。



2.5.1 基本控制参数（Basic Controls）



这一组参数都是控制 Sensor 的配置参数（硬件参数），只有在实时显示时起作用，Playback 时不需要调节这一组参数。

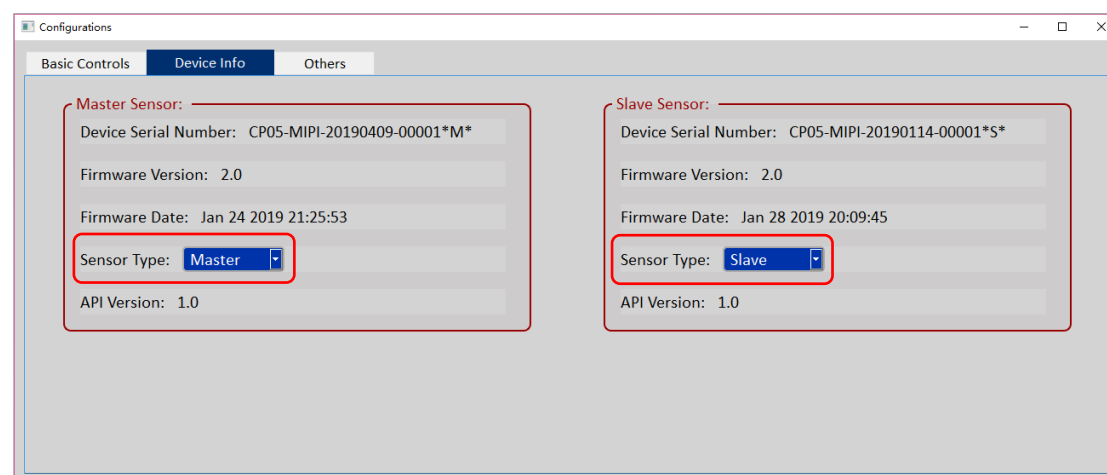
当只有一个 Sensor 连接时，那么只需要调节左边的 Master Parameters 即可。当有两个 Sensor 连接时，那么左边的 Master Parameters 对应的是 Master Sensor，右边的 Slave Parameters 对应的是 Slave Sensor。至于如何区分 Master Sensor 和 Slave Sensor，将会在接下来的设备信息章节里介绍。

基本控制中每个参数的功能以及意义

Clock	调节 Sensor 的工作频率，默认值为 100MHz，该值越大，Sensor 的检测速度越快。 对所有模式都有效
Brightness	调节图像的亮度，该值越大，图像会越暗。 对 Full-Frame Picture 和 Event Intensity 模式有效
Threshold	调节触发 Event 的阈值，同样的条件下，该值越大，触发的 Event 的数量越小。 对 Event Address Only，Event Optical-flow，Event Intensity 以及 Single Full-frame Optical-flow 模式都有效
ISO	调节图像对比度和动态范围，该值越大，对比度越低动态范围越高。 如果该值被修改了，需要重新采集 FPN（如何采集 FPN，请参考 2.6 章节），本 SDK 会自动切换并使用新的 ISO 值对应的 FPN 文件。 对 Full-Frame Picture 和 Event Intensity 模式有效



## 2.5.2 设备信息（Device Info）



这一组参数中除了 Sensor Type 之外，其他的信息都是只读的，下表给出了每个信息的具体意义。

Device Serial Number	设备序列号，每个 Sensor 都有一个唯一的序列号
Firmware Version	固件的版本号
Firmware Date	固件的 Release 日期
Sensor Type	用来区分 Master Sensor 和 Slave Sensor，默认所有的 Sensor 都是 Master，用户可以通过上图所示的选择框来修改 Sensor 的 Master 和 Slave 属性
API Version	API 的版本号

当只有一个 Sensor 连接时，设备信息会显示 在左边的 Master Sensor 中，右边的 Slave Sensor 则是空的。

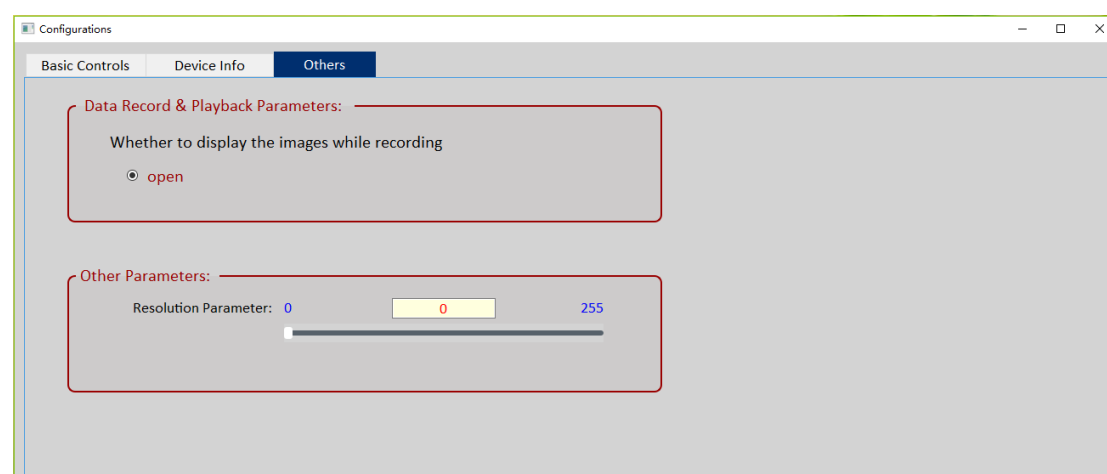
当有两个 Sensor 连接时，如果用户没有修改过 Sensor Type，那么左边显示的是第 1 个被识别到 Sensor 设备信息，右边显示的是第 2 个被识别到的 Sensor 设备信息，并且两边的 Sensor Type 都为 Master。

**备注：**设备被识别的先后顺序，跟连接 Sensor 的时间先后顺序无关，跟连接的 USB 的端口位置有关。

用户需要根据 2 个 Sensor 的实际连线情况，主动把 Master Sensor 的 Sensor Type 设置为“Master”，把 Slave Sensor 的 Sensor Type 设置为“Slave”，然后重启程序，这样即可保证，每次启动时，Master Sensor 的图像都会显示在左边，Slave Sensor 的图像都会显示在右边。



### 2.5.3 其他设置参数（Others）

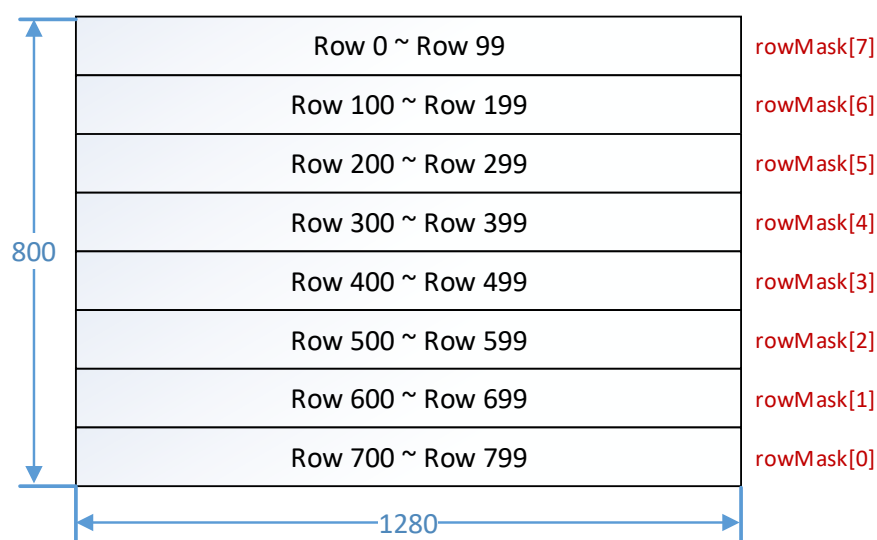


在进行 bin 文件录制操作时，可以选择关闭 Sensor 的图像显示功能以保证录制的数据的完整性（由于解析数据和显示图像非常耗时，可能会导致录制的数据丢失）。

此外，该界面还开放了一个修改分辨率的配置。用户可以通过修改“*Resolution Parameter*”来关闭掉一些行的显示。

如下图所示，Sensor 默认输出的图像分辨率为 1280\*800，它可以按行被分割成 8 个区块，每一区块的大小都是 1280\*100，且每个区块都有对应的控制位（从上到下分别为 rowMask[7]到 rowMask[0]）。

例如，要关闭最上面的 100 行，那么只需将 rowMask[7]设置为 1，其他的控制位都设置为 0，即 *Resolution Parameter* 的值设置为 128（b'10000000）即可。

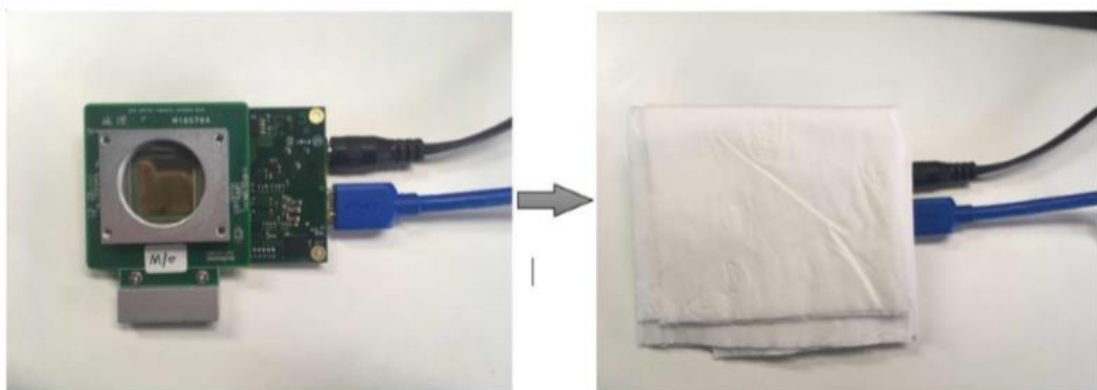


## 2.6 采集 FPN 功能

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-5 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-frame Picture* 模式。如果是 2 个 Sensor, 则把它们的工作模式都切换至 *Full-frame Picture* 模式。
- (2) 由于 FPN 生成操作必须在光照均匀的环境下进行, 所以我们可以取下一张白纸 (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。



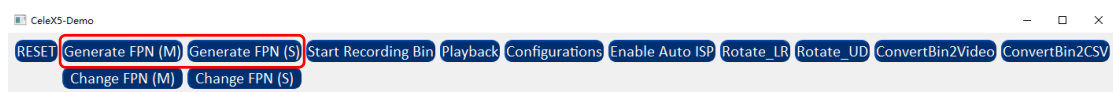
- (3) 执行 FPN 生成操作之前, 请检查图像屏幕, 确保其显示正常, 不要太暗或太亮。只需在裸露的 Sensor 上放置更多或更少的纸张, 或者在 GUI 窗口上打开或关闭“亮度”滑块, 即可更改照明。

备注: 下图中的第 3 幅图就是正常亮度的图。

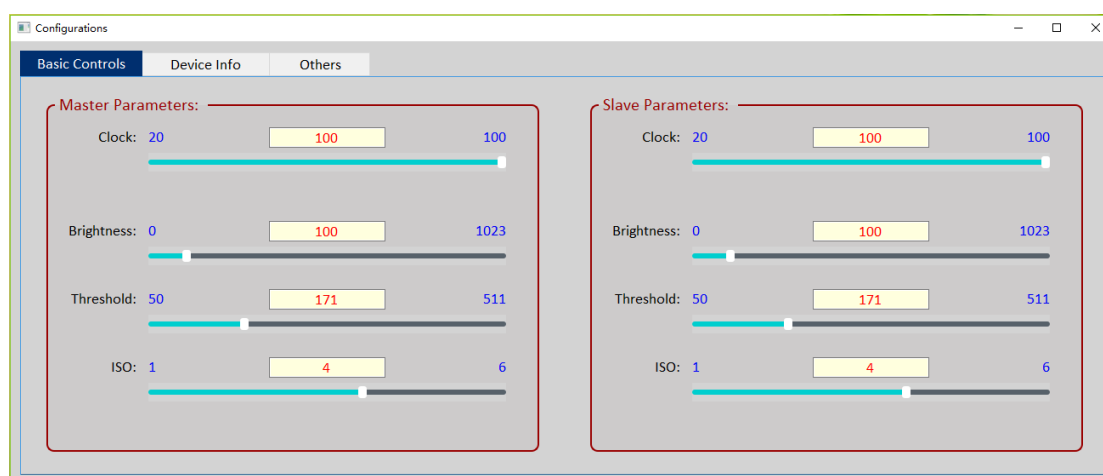


- (4) 点击 GUI 窗口中的“**Generate FPN**”按钮, 当你在指定目录下看到 FPN\_3.txt 文件时表明 FPN 文件成功生成了。

备注: 如果是 2 个 Sensor, 则需要分别生成 FPN, Master Sensor 点击 “**Generate FPN (M)**”按钮, 生成的 FPN 文件名称为 FPN\_3\_M.txt; Slave Sensor 点击 “**Generate FPN (S)**”按钮, 生成的 FPN 文件名称为 FPN\_3\_S.txt。



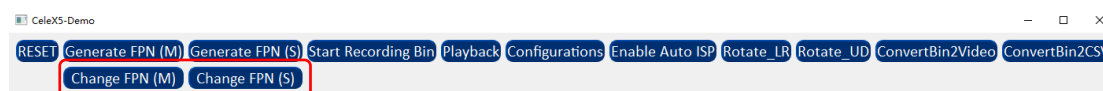
备注：不同的 ISO 档位对应了不同的 FPN 文件，ISO 一共有六档，默认是第三档，对应了 FPN\_3.txt 文件。用户可以通过调整 Configuration 设置中的 ISO 档位来获取更亮或更暗的图像。



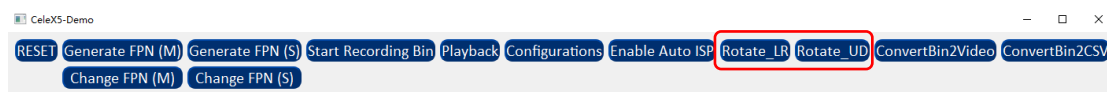
- (5) 生成相应 ISO 设置下的 FPN 文件后，我们通过点击 “**Change FPN**” 按钮可以选择切换到相应的 FPN。

备注：

- 如果按照步骤生成了 FPN 文件，但是切换后图像清晰度没有提高，检查确认当前 ISO 与 FPN 是否对应；检查选择的 FPN 路径是否包含中文路径。
- 如果是 2 个 Sensor，则 Master Sensor 则点击 “**Change FPN (M)**”按钮选择切换到相应的 FPN；Slave Sensor 则点击 “**Change FPN (S)**”按钮选择切换到相应的 FPN。



## 2.7 图像翻转功能

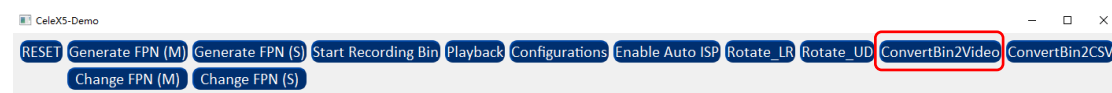


通过点击上图所示的“*Rotate\_LR*”和“*Rotate\_UD*”按钮，可以对图像进行左右或者上下翻转。

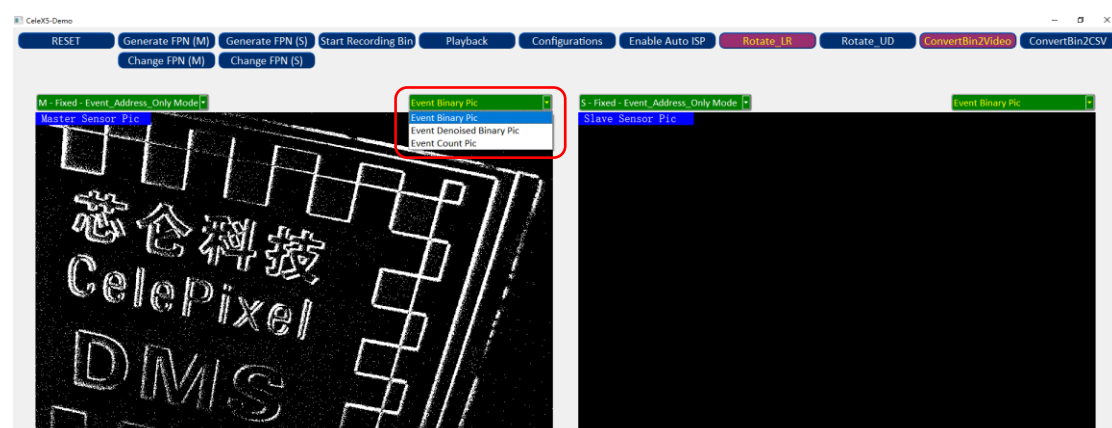
备注：如果是 2 个 Sensor，则会同时翻转 2 个 Sensor 的图像，目前 Demo GUI 还不支持只翻转某一个 Sensor 的图像。

## 2.8 bin 文件转视频功能

点击“**ConvertBin2Video**”按钮，可以将录制的 bin 文件转换出与该文件同名的视频文件，且一次只支持转一个 bin 文件。

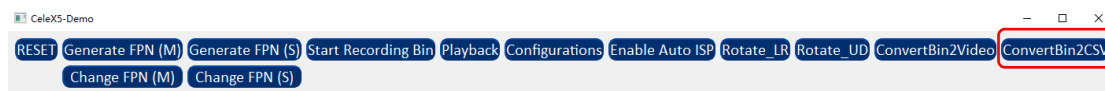


Windows 下生成.mkv 格式的视频文件，Linux 下生成.mp4 格式的视频文件。选择不同的图片格式，可以转换出该 bin 相应的图片格式视频。例如：选择去噪的图片格式，可以将 Event-Address\_Only 模式的 bin 文件转换成去噪后的图像视频。



## 2.9 bin 文件转 CSV 文件功能

点击 “**ConvertBin2CSV**” 按钮，可以将录制好的 bin 文件转换成 CSV 文件，且一次只支持转一个 bin 文件。



bin 文件转 CSV 文件的功能只支持 event 模式下录制的数据，且不同的 event 模式转出来的数据也有所不同，具体如下：

- (1) 对于 Event Address Only 模式，该文件将 bin 数据存储为像素的 Row, Colum, Timestamp 信息。
- (2) 对于 Event Intensity 模式，该文件将 bin 数据存储为像素的 Row, Colum, Intensity, Polarity, Timestamp 信息。
- (3) 对于 Event Optical-flow 模式，该文件将 bin 数据存储为像素的 Row, Colum, Optical-flow Info, Timestamp 信息。