celepixel

CeleX-5 SDK ZYNQ Platform Getting Started Guide

CelePixel Technology Co. Ltd.

Content

1 CeleX-5 SDK ZYNQ Platform

CeleX-5 SDK ZYNQ Platform is composed of the hardware package and software package.

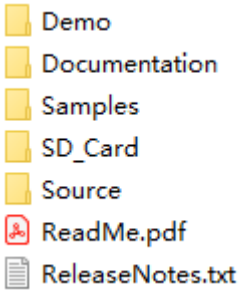
1.1 Hardware package

- CeleX5 ZYNQ board with lens

- USB cable
- Power adaptor
- Power cable

1.2 Software package

The software package contains the following directories:



- **Demo**

  CeleX5Demo_Client: Demo GUI(Windows platform)

  CeleX5Demo_Server: Demo server(ZYNQ platform)

- **Documentation**

  CeleX5_SDK_Reference: The introduction of CeleX-5 Sensor and the references of all the classes and functions in the SDK.

  CeleX5_ZYNQ_SDK_Getting_Started_Guide: The guide for how to use the CeleX5 ZYNQ Demo Kit and compile the source code.

- **Samples**

  Several examples developed based on SDK and a sample user manual file.

- **SD_Card**

  boot: System files required to boot the CeleX5 ZYNQ board.

root: Demo executable as well as its libraries and configuration files.

- **Sources**

  CeleX: CeleX-5 SDK source code.

  CeleX5Demo_Client: Demo Client source code (developed with Qt).

  CeleX5Demo_Server: Demo Server source code.

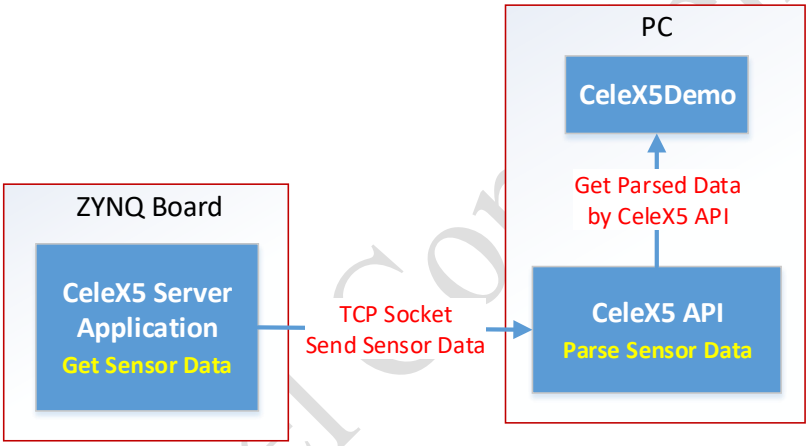
- **ReadMe**

  Brief Introduction of the CeleX-5 Sensor SDK.

- **ReleaseNotes**

  New features, fixed bugs and SDK development environment.

This figure shows the basic working principle of the CeleX-5+ZYNQ Chipset. Applications could configure the ZYNQ board and eventually configure the sensor. Similarly, Applications are able to obtain the sensor data from ZYNQ board over the network.



Basic working principle of the CeleX-5+ZYNQ Chipset

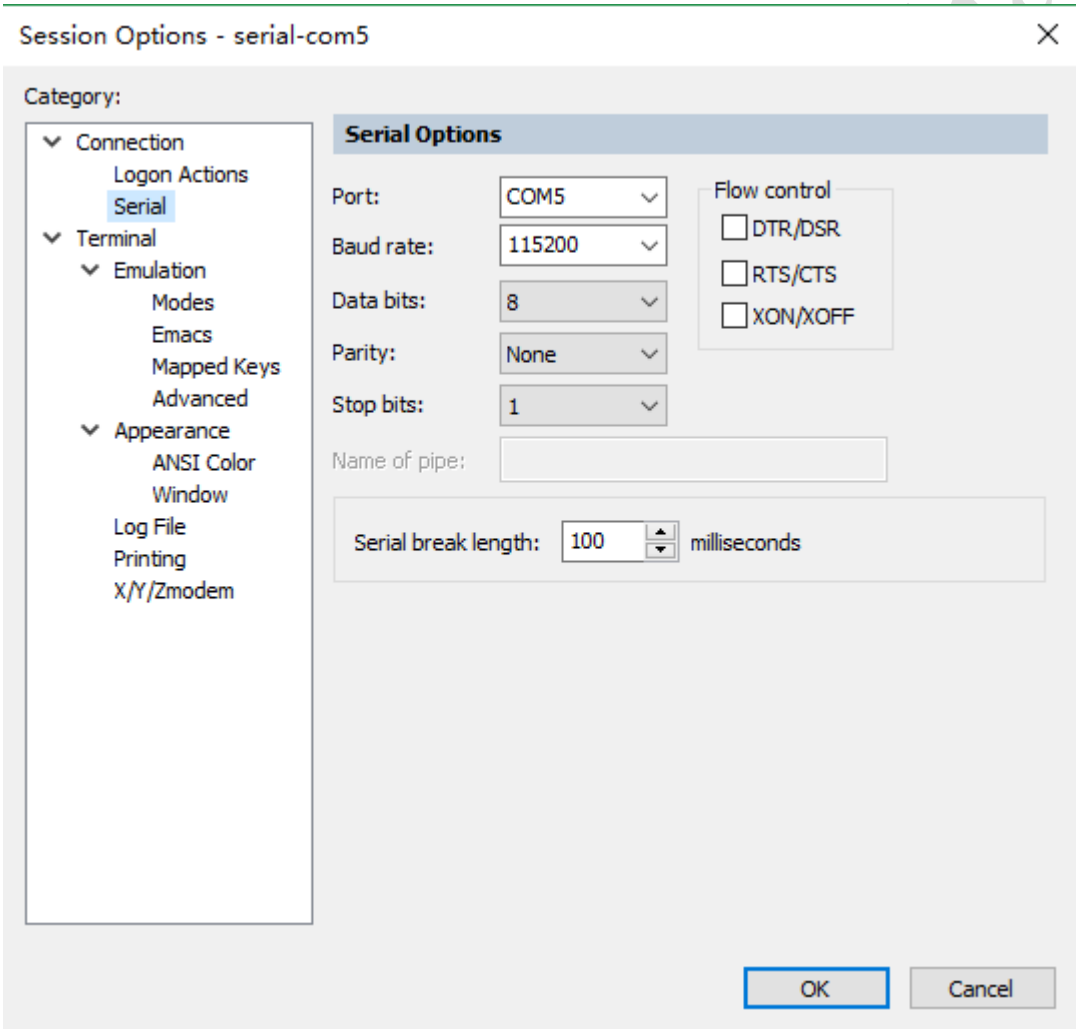
2 How to use CeleX-5 SDK ZYNQ Platform

2.1 Serial terminal

In order to view the status of the server program running on CeleX5 ZYNQ board, a serial terminal (SecureCRT, Putty or any other terminal) is required to display the output from the CeleX5 ZYNQ board uart port (emulated serial port).

The baud rate of the serial port is 115200 and the serial port should be configured to the one that connects to the CeleX5 ZYNQ board.

The following image shows a typical configuration



2.2 IP address configuration

CeleX5 ZYNQ board communicates with PC via Ethernet port, by default, CeleX5 ZYNQ board has a fixed IP of 192.168.1.11, and the IP of the PC that runs CeleX5 Demo Client program should be configured in the same subnet. For example, the PC could use an IP address of 192.168.1.100 with subnet mask of 255.255.255.0.

2.3 Prepare SD card partitions*

***This step is for a new SD card only.**

A. Create two partitions on SD card:

- Partition ‘boot’
  - FAT file system, size of 1 GB
  - Partition ‘boot’ contains files required to start the system.
- Partition ‘root’
  - Ext4 file system, remaining space of SD card.
  - Partition ‘root’ contains CeleX5 Demo server program, its libraries, configuration files, etc.

B. Copy all files in “SD_Card/boot” folder from software package to partition ‘boot’ of the SD card and copy all files in “SD_Card/root” folder from software package to partition ‘root’ of the SD card.

Note: Don’t forget to assign execute permission to executable “CeleX5Demo” and script file “start.sh” in partition ‘root’.

2.4 Insert SD card

Insert SD card into the SD card slot of CeleX5 ZYNQ board.

2.5 Connect CeleX5 ZYNQ board to PC

A. Connect CeleX5 ZYNQ board to PC using a USB cable for displaying CeleX5 Demo Server status. It is only for viewing server status and does not transfer any data captured from sensor.

B. Connect CeleX5 ZYNQ board to PC using a 1000M Ethernet cable for transferring data to and from PC.

2.6 Power up CeleX5 ZYNQ board

Connect power cable to the CeleX5 ZYNQ board to power up the system.

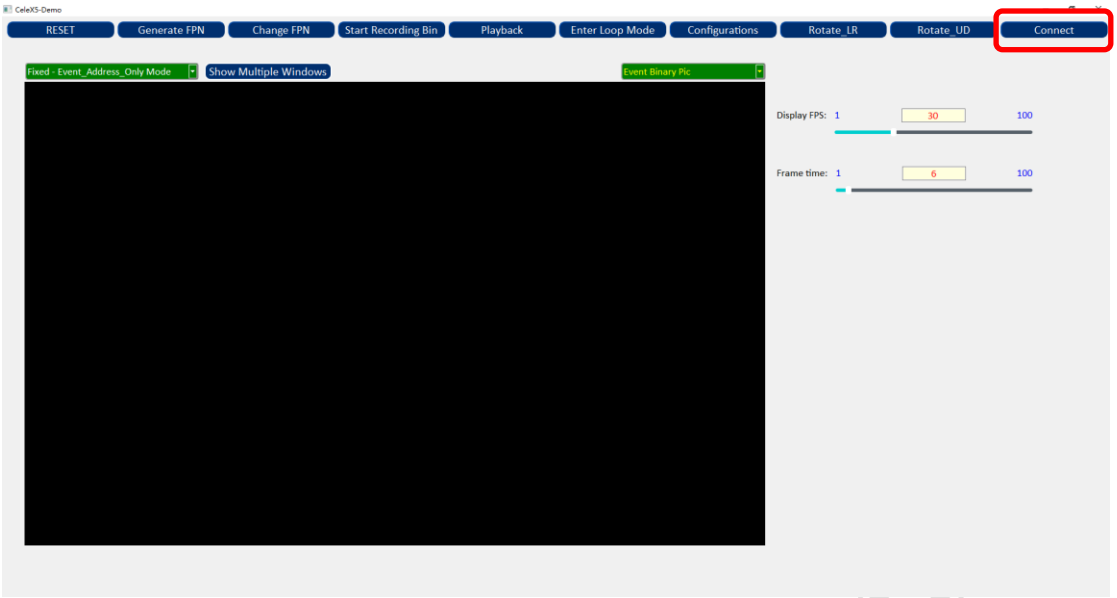
2.7 Open serial terminal program

Open the serial terminal program.

```
-----startup.sh-----
--------------- CeleX5Demo ZYNQ---------------
axidma_open 115
sensor_init 414
-------------- m_uiDMABufferSize = 655360
------reset------
create socket and wait connect
CeleX5::wireIn: adaxidma_fasync=================================================
dress = 93, value = 0
CeleX5::wireIn: address = 90, value = 1
wbefor axidma_read_transfer_cyclic() come time sec=9, usec=798178
ait connect...
CeleX5::wireIn: address = 53, value = 0
CeleX5:xilinx_dma_prep_dma_cyclic() return good!
:wireIn: address = 90, value = 0
CeleX5::wireIn: address = 93, xilinx_dma_tx_submit() return cookie good
value = 1
fcntl before
1===============
2===============
3==dmaengine submit
=============
4===============
fcntl after
axidma_pre_transfer_cyclic() good
axidma_read_transfer_cyclic() after axidma prep
dma_async_issue_pending
xilinx_dma char->has_sg==========0
xilinx_dma char->has_sg&&!char->xdev->mcdma==========1
xilinx_dma char->has_sg==============2
before xilinx_dma_start=======
xilinx_dma_start() before poll
xilinx_dma_start() after poll
after xilinx_dma_start==========
xilinx_dma char->has_sg==========3
xilinx_dma char->has_sg==========30
xilinx_dma char->has_sg==========301
XILINX_DMA_REG_DMASR status = 0x10008
XILINX_DMA_REG_DMACR status = 0x17013
XILINX_DMA_REG_DMASR status = 0x10008
XILINX_DMA_REG_DMACR status = 0x17013
dma_status xilinx_dma_tx_status ret == DMA_COMPLETE || !txstate
after axidma_read_transfer_cyclic() return time sec=9, usec=900916
get pic successfully !!!!!!!!!!!!!!!!
random: fast init done
```

Once the message “get pic successfully!!!!!!!!!!!!!!!” is printed, the server is waiting for connection and ready to send raw data to connected client.

2.8 Run CeleX5 Demo program

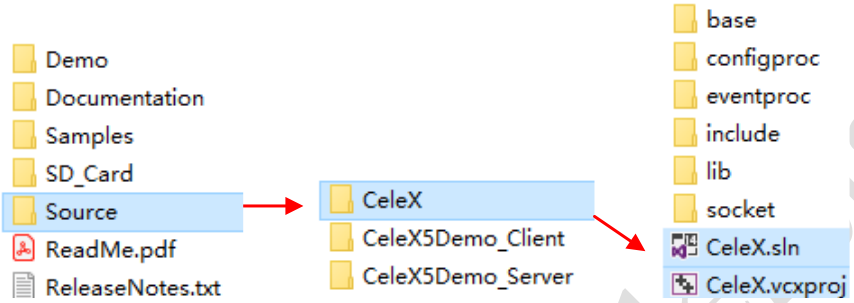


Open the CeleX5 Demo program and click the "***Connect***" button on the right upper corner to connect to the server program on CeleX5 ZYNQ board.

3 Compile CeleX5 SDK Source Code

The OpenCV library (Version 3.3.0) is involved in the CeleX-5 API to develop some interfaces, so you need to install OpenCV and configure its development environment before compiling the source code of CeleX-5 library.

On the Windows platform, a VS2015 project file is provided to compile the source code, and the build library (CeleX.dll and CeleX.lib) will be imported into the directory build/Windows.



Note:

(1) You need to modify the settings of the OpenCV Include and Lib paths in the project properties. Otherwise, the compilation will fail because the OpenCV header files and libraries are not found.

(2) After the compilation is completed, a build/Windows directory will be automatically created in the directory where the project is located, and the compiled library files (CeleX.dll and CeleX.lib) will be automatically imported into this directory.

4 Compile Source Code of CeleX5 Demo GUI (Client)

Development Environment： Qt5.6.3 + OpenCV3.3.0

The CeleX5 Demo GUI is developed using Qt, you could open the project file "**CeleX5Demo.pro**" in the directory 'Source/CeleX5Demo_Client/'and compile it using Qt Creator or any other Qt IDE.

Notes: It needs to modify the INCLUDEPATH and LIBS of OpenCV in the file *CeleXDemo.pro*.

```
win32 {
    INCLUDEPATH += $$quote(D:/Program Files/opencv/build/include) \
                   $$quote(D:/Program Files/opencv/build/include/opencv) \
                   $$quote(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 \
            /usr/lib/x86_64-linux-gnu/libusb-1.0.so
}
```

5 Compile Source Code of CeleX5 Demo (Server)

CeleX5 Demo Server is developed using Code::Blocks IDE, the codeblocks project file "CeleX5Demo.cbp" can be found in the directory "Source/CeleX5Demo_Server/CeleX5Server".

The CeleX5Sever program is the onboard program of the ZYNQ platform and the compilation environment of the program is **ARM-Linux**.

The CeleX5Sever executable file has been placed in the root directory by default. If you want to develop on the server side, you can refer to the source code.

6 The Functions of CeleX-5 Demo GUI

If the demo server is not connected, the interface screen is shown as Fig.6-1, when there is a demo server connected, the interface screen is shown as Fig.6-2.

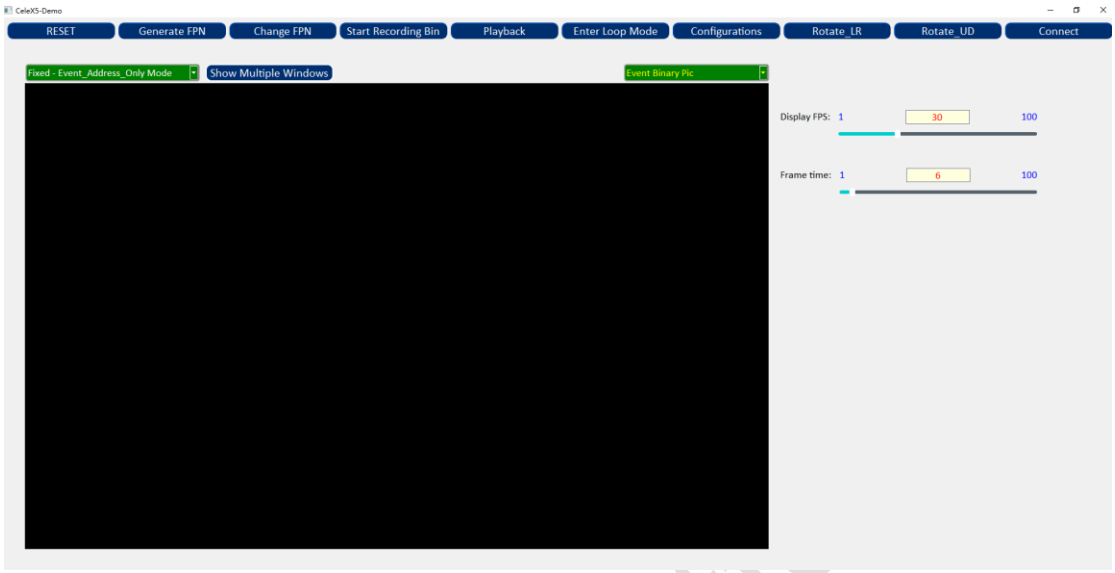


Fig. 6-1

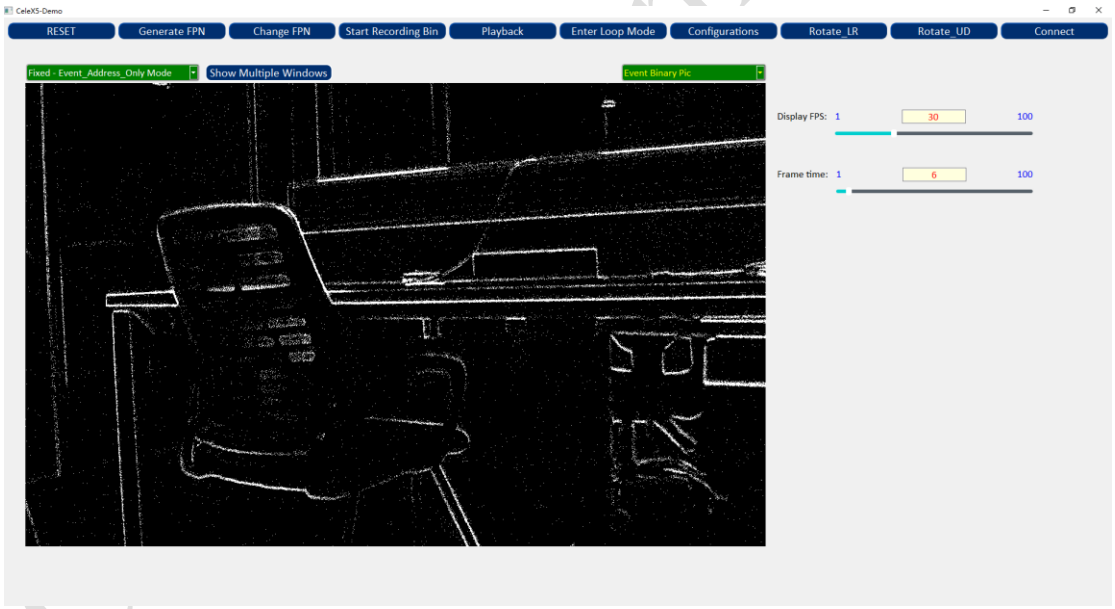


Fig. 6-2

6.1 Change Sensor Mode

In the Fixed Mode, click the button "***Enter Loop Mode***" shown in Figure 6-3-1 to enter the Loop Mode. The images of *Loop Mode* are displayed as shown in Figure 6-4. Loop A is the first loop, its mode is *Full-frame Picture mode*, Loop B is the second loop, its mode is *Event mode*, and Loop C is the third loop, its mode is *Full-frame Optical-flow mode*.

In the Loop Mode, click the button "***Enter Fixed Mode***" shown in Figure 6-3-2 to switch to the

Fixed mode (the default mode is *Event mode*).

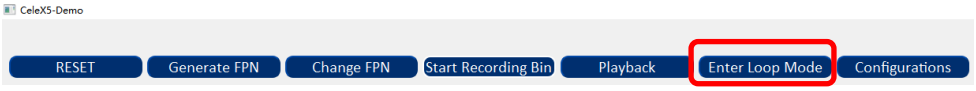


Fig. 6-3-1

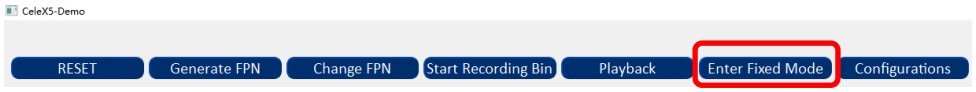


Fig. 6-3-2

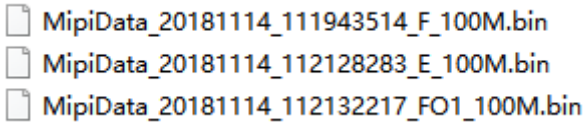
Fig. 6-4 Sensor works in Loop Mode

6.2 Record Raw Data of Sensor (Bin Files)

Click the "***Start Recording Bin***" button in Figure 6-5-1 to start recording bin data, then the text on the button will change to "***Stop Recording Bin***" as shown in Figure 6-5-2.

Click the "***Stop Recording Bin***" button to stop recording bin data. The recorded bin file will store in the same directory as CeleXDemo.exe and is named in the form of

MipiData_YYYYMMDD_HHMMSSSSS_SensorMode_ClockRate.bin, as follow:



F: Full-frame Picture mode,

E: Event mode

FO1: Single Full-frame Optical-flow mode

100MHz: The operating frequency of the Sensor is 100MHz

Fig. 6-5-1

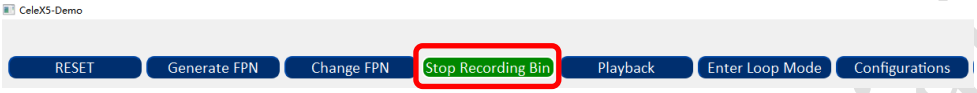


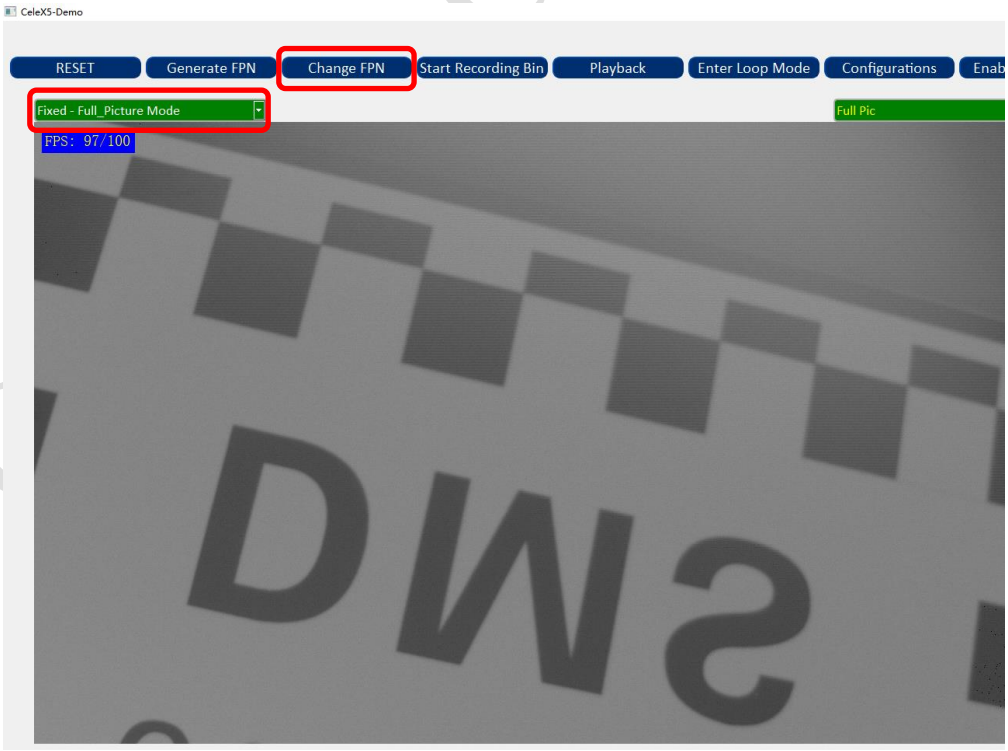
Fig. 6-5-2

6.3 Generate FPN file

FPN (Fixed Pattern Noise) is the term given to a particular noise pattern on digital imaging sensors often noticeable during longer exposure shots where particular pixels are susceptible to giving brighter intensities above the general background noise. To get rid of FPN, we need to create a FPN file for CeleX-5 Sensor. Each sensor requires its own FPN.

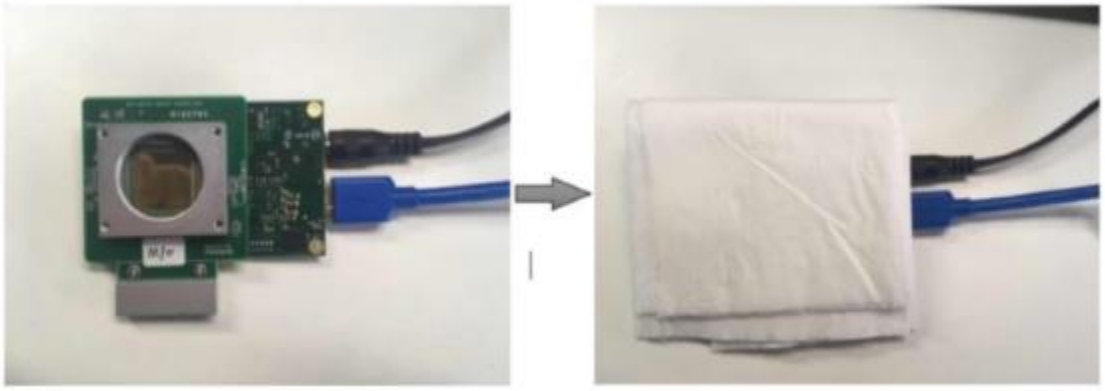
Each sensor requires its own FPN, and steps for generating FPN are illustrated as below:

1) Switch the Sensor operating mode into “Full-frame Picture Mode”.



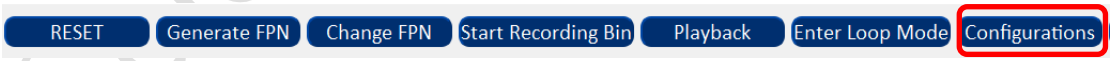
2) Since the FPN should be conducted under the condition of uniform illumination, we could use the way of removing optical lens and covering a piece of white paper (thin tissue or A4

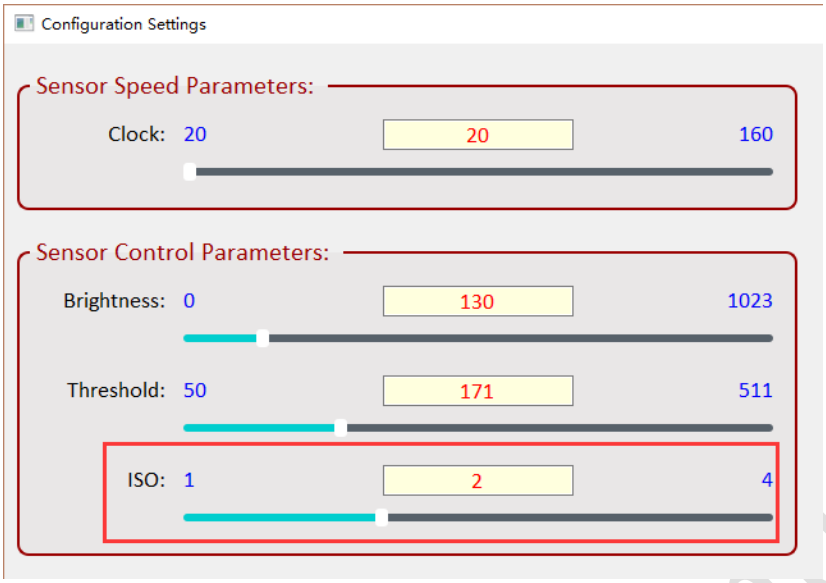
paper) over the exposed Sensor. Make sure that paper completely covers the sensor and sheet is stationary. **NOTE: the effect will be better if you operate in natural light rather than the LED lamp.**



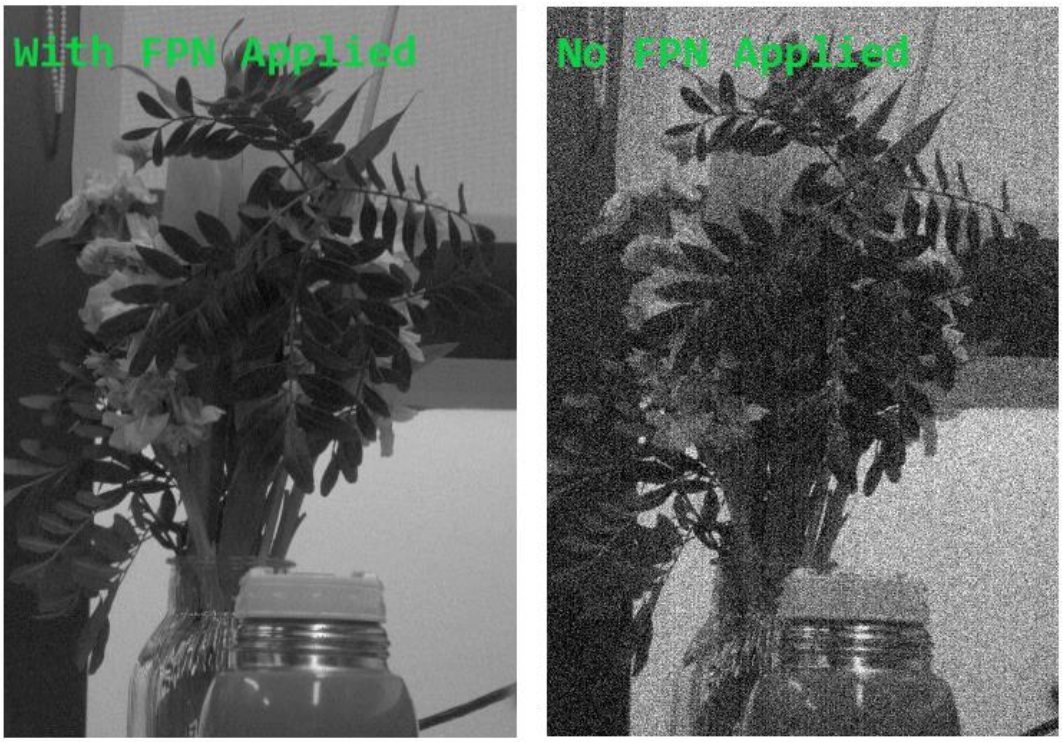
3) Before generating FPNs, please check the image screen and make sure it is normal, which is neither too dark nor too bright. Then, you could adjust the amount of paper over the Sensor or switch the "Brightness" slider in the GUI to change the luminance. **NOTE: the 3rd figure is the right luminance among the three figures below.**

4) Click the "***Generate FPN***" button in GUI. Then, you could see the FPN_2.txt file in assigned direction after FPN file was successfully generated. **Note: Different ISO levels correspond to different FPN files. ISO has a total of four levels. The default is the second level, which corresponds to the FPN_2.txt file. The user can obtain a brighter or darker image by adjusting the ISO level in the Configuration settings.**





5) After generating the FPN file under the corresponding ISO settings, we can choose to switch to the corresponding FPN by clicking the "***Change FPN***" button. **Note: If the FPN file is generated according to the steps, but the image resolution is not improved after switching, check whether the current ISO corresponds to the FPN; check whether the selected FPN path contains the Chinese path.**



6.4 Flip image

Click the "***Rotate_LR***" or "***Rotate_UD***" button to flip the image left and right or up and down.