CeleX-5 SDK ZYNQ Platform Getting Started Guide

CelePixel Technology Co. Ltd.

Content

1 CeleX-5 SDK ZYNQ Platform

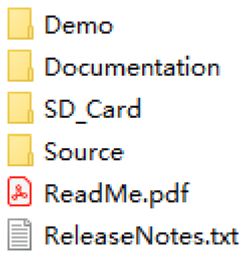
CeleX-5 SDK ZYNQ Platform is composed of the hardware package and software package.

- Hardware package
  - CeleX5 ZYNQ board with lens

  - USB cable
  - Power adaptor
  - Power cable

- Software package

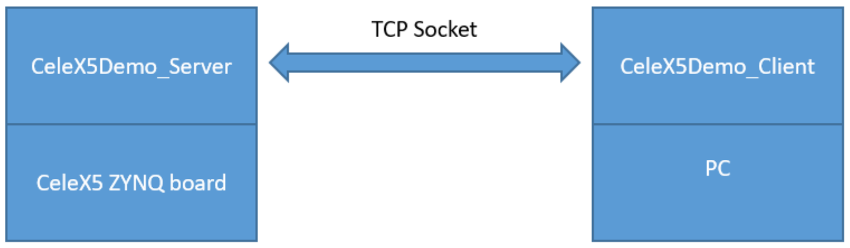
  The software package contains the following directories:



Demo
Documentation
SD_Card
Source
ReadMe.pdf
ReleaseNotes.txt

  - 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.
  - 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.



Demo software server client relationship

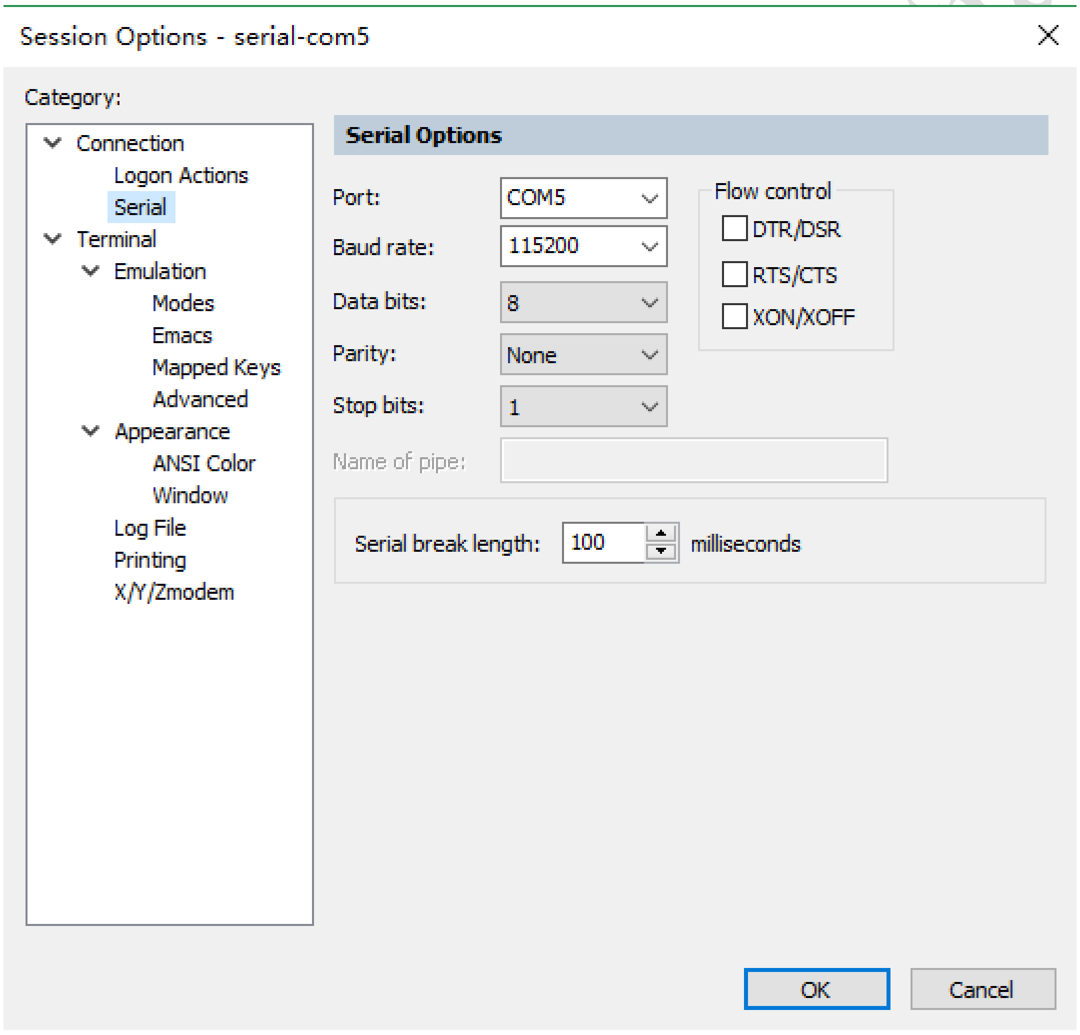
2 How to use CeleX-5 SDK ZYNQ Platform

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



Port: COM5

Baud rate: 115200

Data bits: 8

Parity: None

Stop bits: 1

Name of pipe:

Flow control: ☐ DTR/DSR ☐ RTS/CTS ☐ XON/XOFF

Serial break length: 100 milliseconds

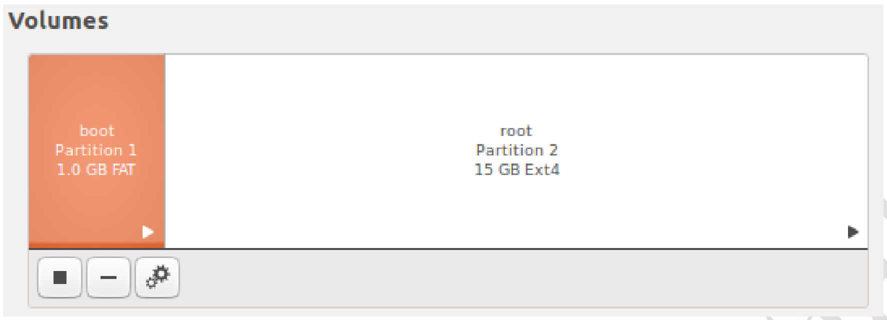
2.2 IP address configuration

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:



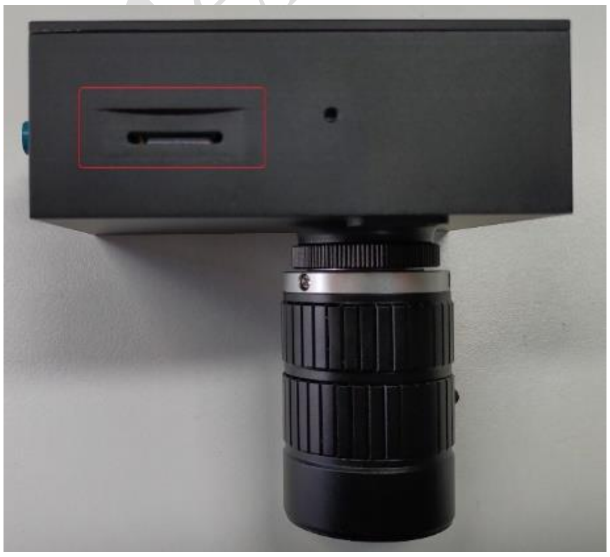
- Partition 'boot'
  - FAT file system, size of 1 GB
  - Partition 'boot' contains files required to start the system.
- Partition 'root'
  - Ext4 file system, remaining space of SD card.
  - Partition 'root' contains CeleX5 Demo server program, its libraries, configuration files, etc.

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'.

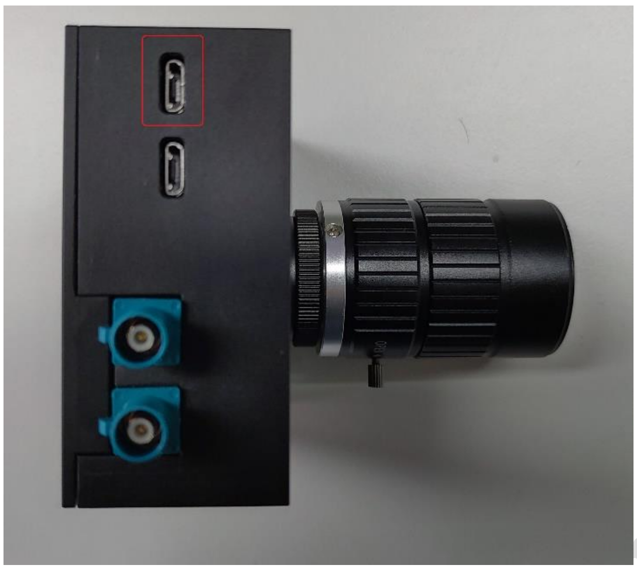
2.4 Insert SD card

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

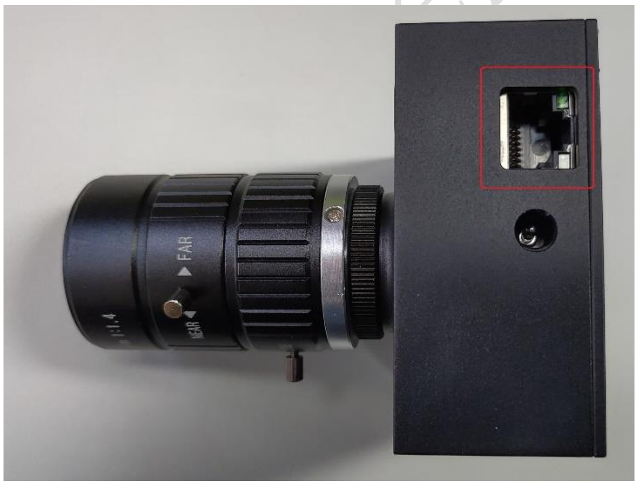


2.5 Connect CeleX5 ZYNQ board to PC

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.

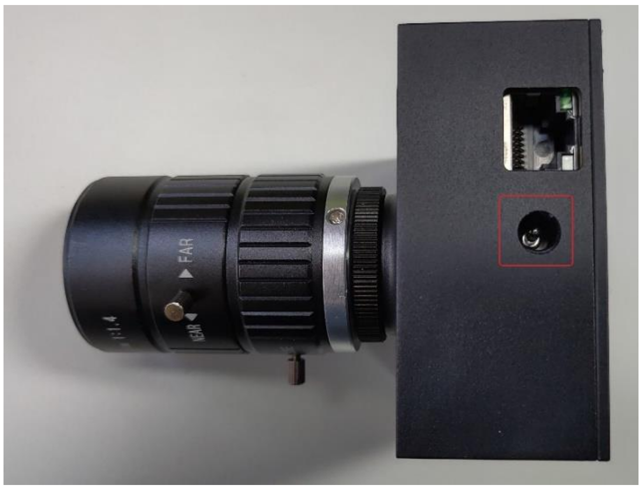


B. Connect CeleX5 ZYNQ board to PC using a 1000M Ethernet cable for transferring data to and from PC.



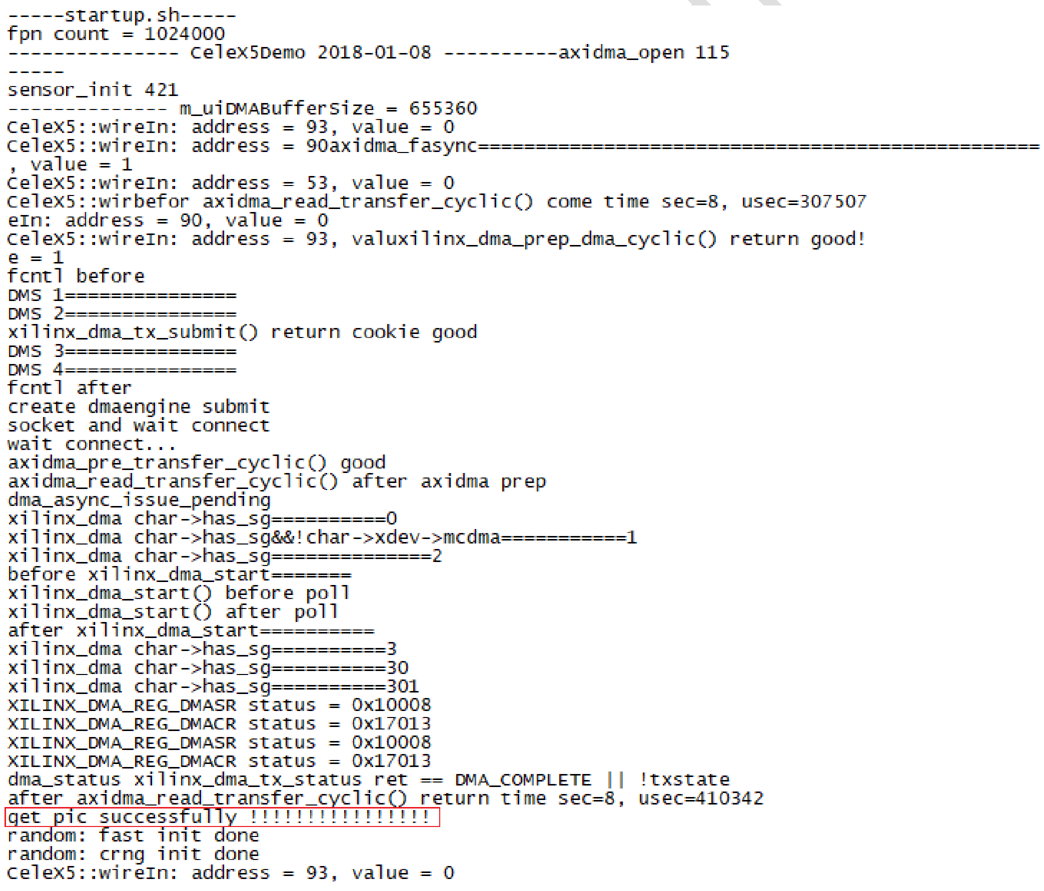
2.6 Power up CeleX5 ZYNQ board

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



2.7 Open serial terminal program

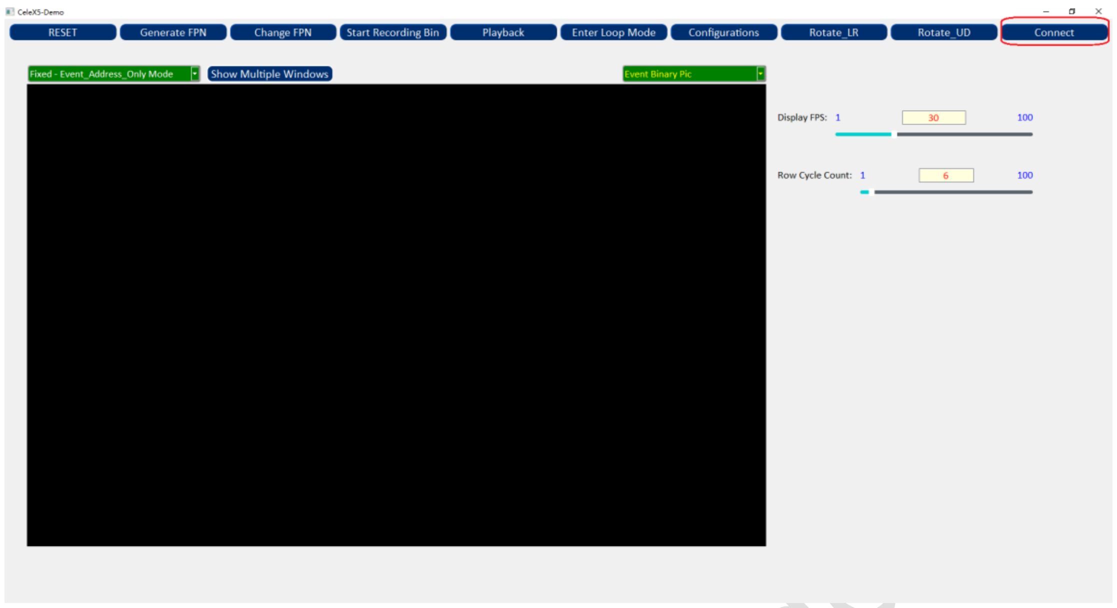
Open the serial terminal program.



```
-----startup.sh-----
fpn count = 1024000
--------------- CeleX5Demo 2018-01-08 ----------axidma_open 115
-----
sensor_init 421
-------------- m_uiDMABufferSize = 655360
CeleX5::wireIn: address = 93, value = 0
CeleX5::wireIn: address = 90axidma_fasync==================================================
, value = 1
CeleX5::wireIn: address = 53, value = 0
CeleX5::wirbefor axidma_read_transfer_cyclic() come time sec=8, usec=307507
eIn: address = 90, value = 0
CeleX5::wireIn: address = 93, valuxilinx_dma_prep_dma_cyclic() return good!
e = 1
fcntl before
DMS 1===============
DMS 2===============
xilinx_dma_tx_submit() return cookie good
DMS 3===============
DMS 4===============
fcntl after
create dmaengine submit
socket and wait connect
wait connect...
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=8, usec=410342
get pic successfully !!!!!!!!!!!!!!!!
random: fast init done
random: crng init done
CeleX5::wireIn: address = 93, value = 0
```

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

2.8 Run CeleX5 Demo program



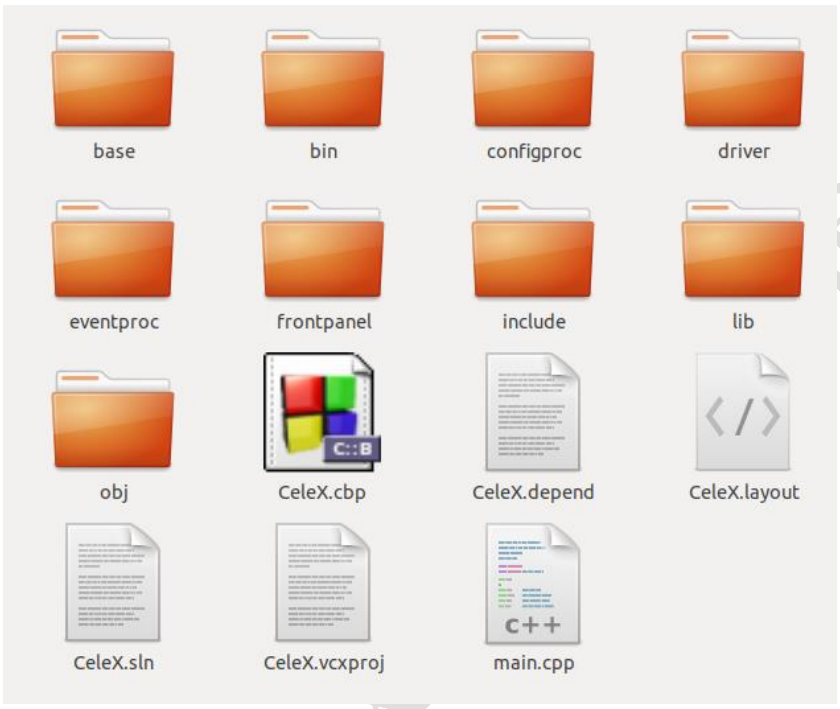
Open the CeleX5 Demo program and click the 'Connect' button on the right upper corner to connect to the server program on CeleX5 ZYNQ board.

3 Compile CeleX-5 SDK Source Code

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

3.1 ZYNQ Platform

On ZYNQ Platform, the code block project ‘CeleX.cbp’ is provided in the folder ‘Source/CeleX’, OpenCV header file and library path should be configured properly in order to compile the CeleX5 SDK source code.



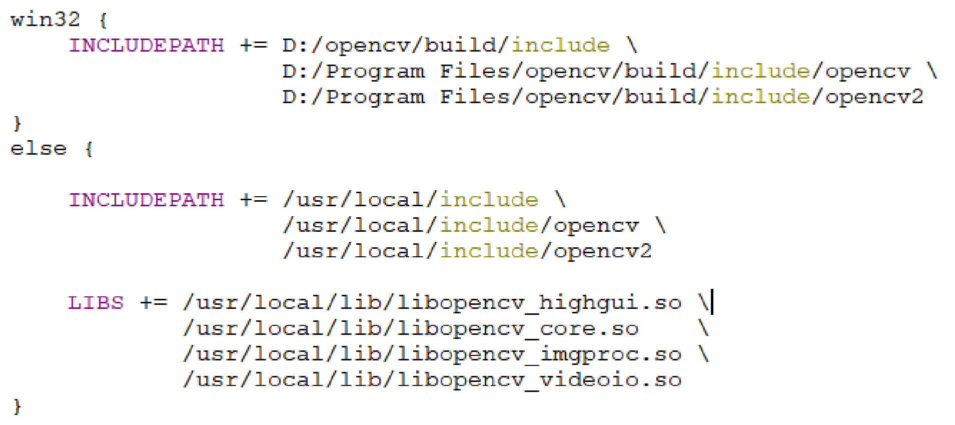
The generated library ‘libCeleX.so’ will be placed in the ‘bin/Release’ folder.

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 += 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
}
```

5 Compile Source Code of CeleX5 Demo (Server)

CeleX5 Demo Server is developed using Code Block IDE, the code block project file ‘CeleX5Demo.cbp’ can be found in the directory ‘Source/CeleX5Demo_Server/CeleX5Demo’.

OpenCV header files and libraries should be configured properly.

- CeleX5 SDK library header files location:
  - ‘Source/CeleX5Demo_Server/CeleX5Demo/include’
- CeleX5 SDK library file ‘libCeleX.so’ location:
  - ‘Source/CeleX5Demo_Server/CeleX5Demo/lib’

6 The Functions of CeleX-5 Demo GUI

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

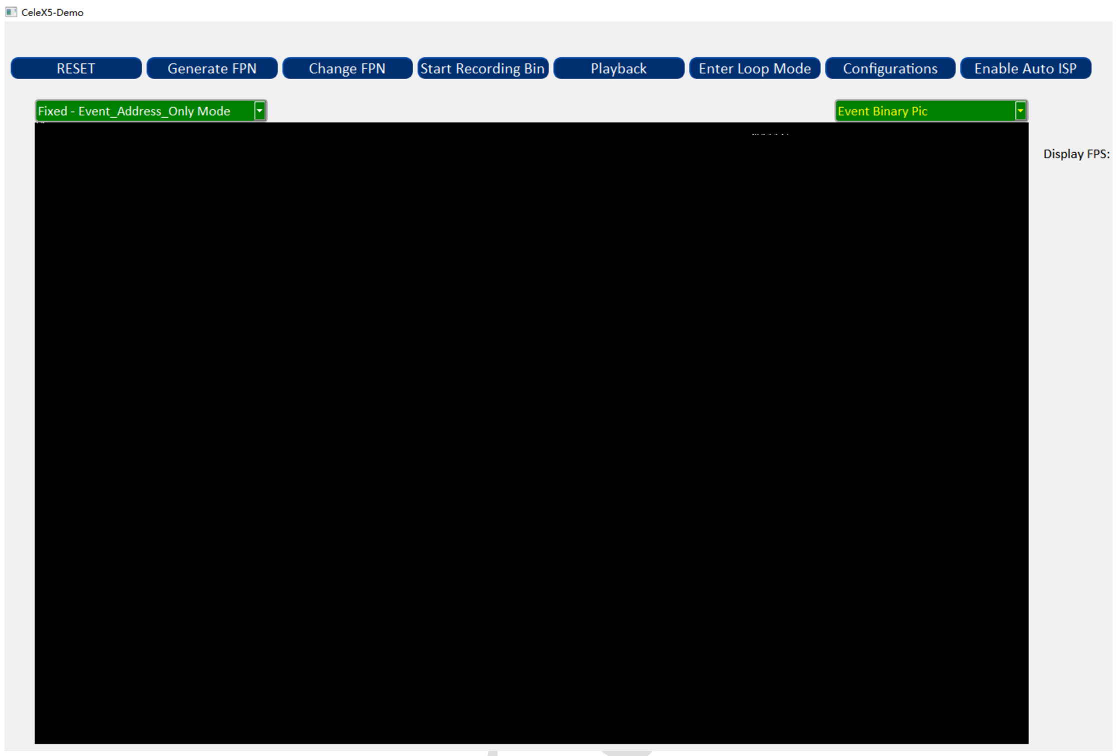


Fig. 2-1

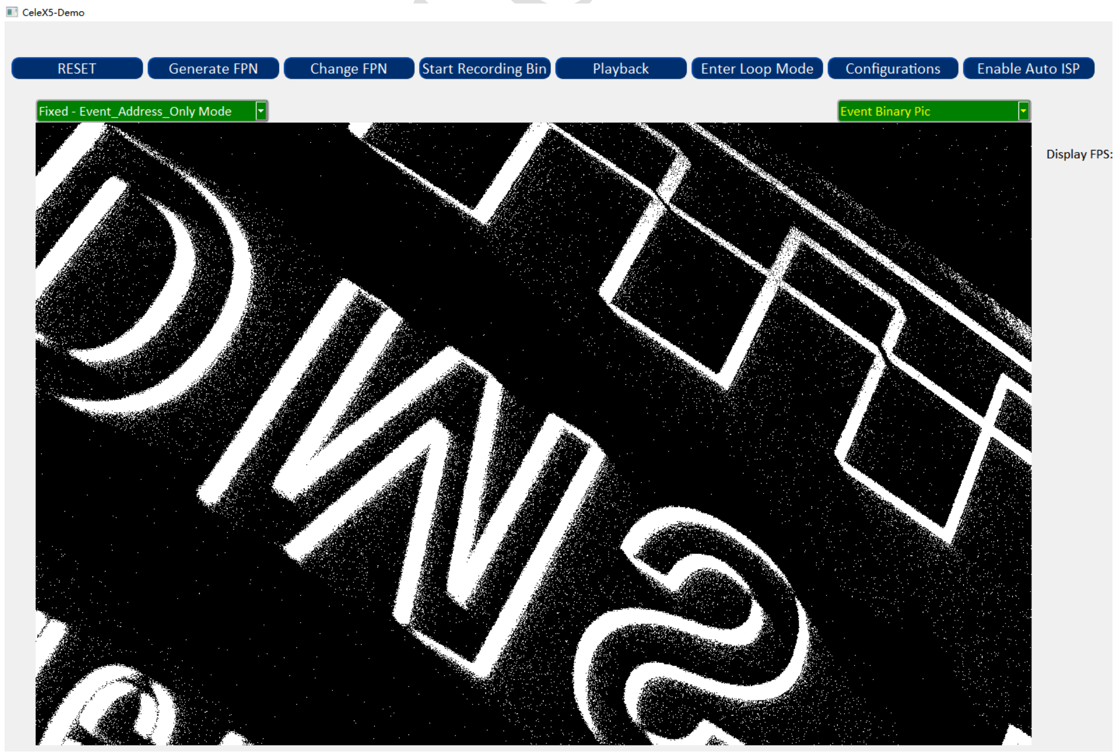


Fig. 2-2

6.1 Change Sensor Mode

In the Fixed Mode, click the button "***Enter Loop Mode***" shown in Figure 2-3-1 to enter the Loop Mode. The images of *Loop Mode* are displayed as shown in Figure 2-4. 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 2-3-2 to switch to the Fixed mode (the default mode is *Event mode*).

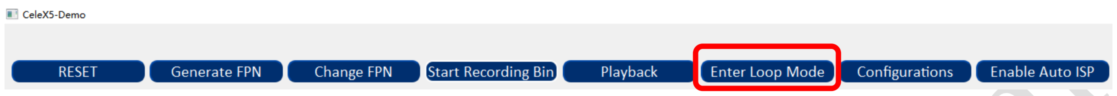


Fig. 2-3-1

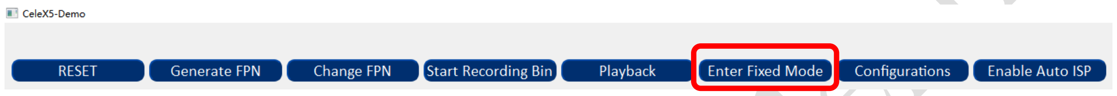


Fig. 2-3-2

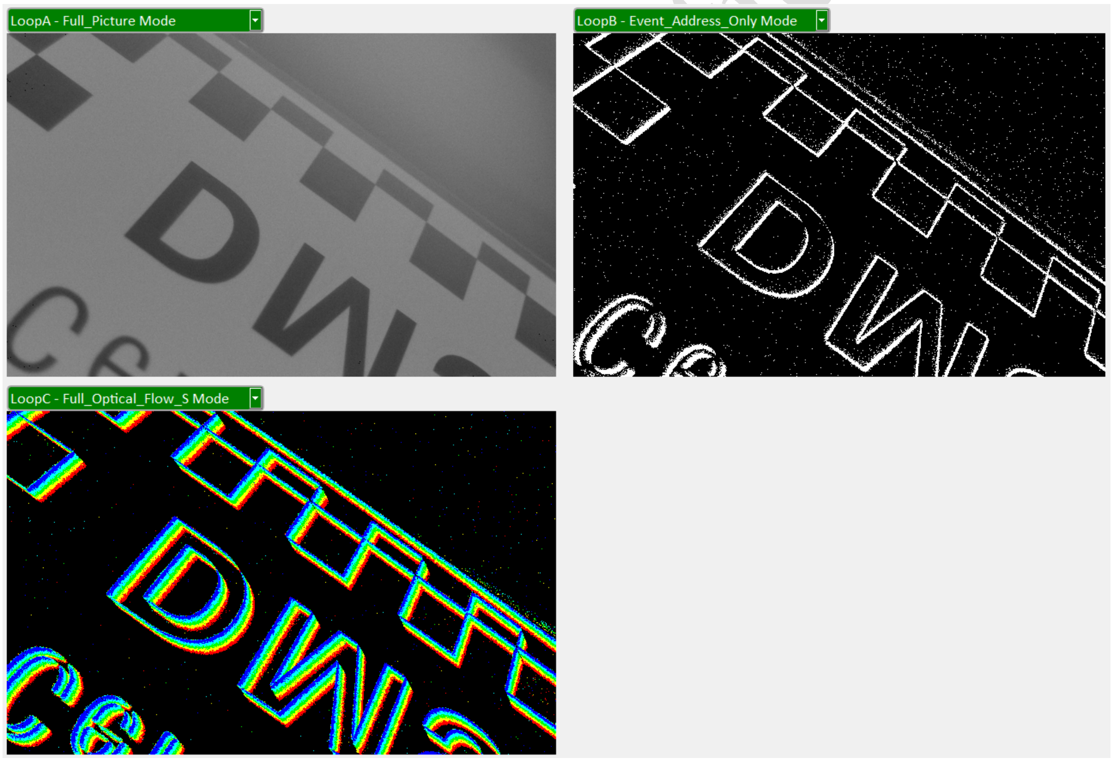

LoopA - Full_Picture Mode
LoopB - Event_Address_Only Mode
LoopC - Full_Optical_Flow_S Mode

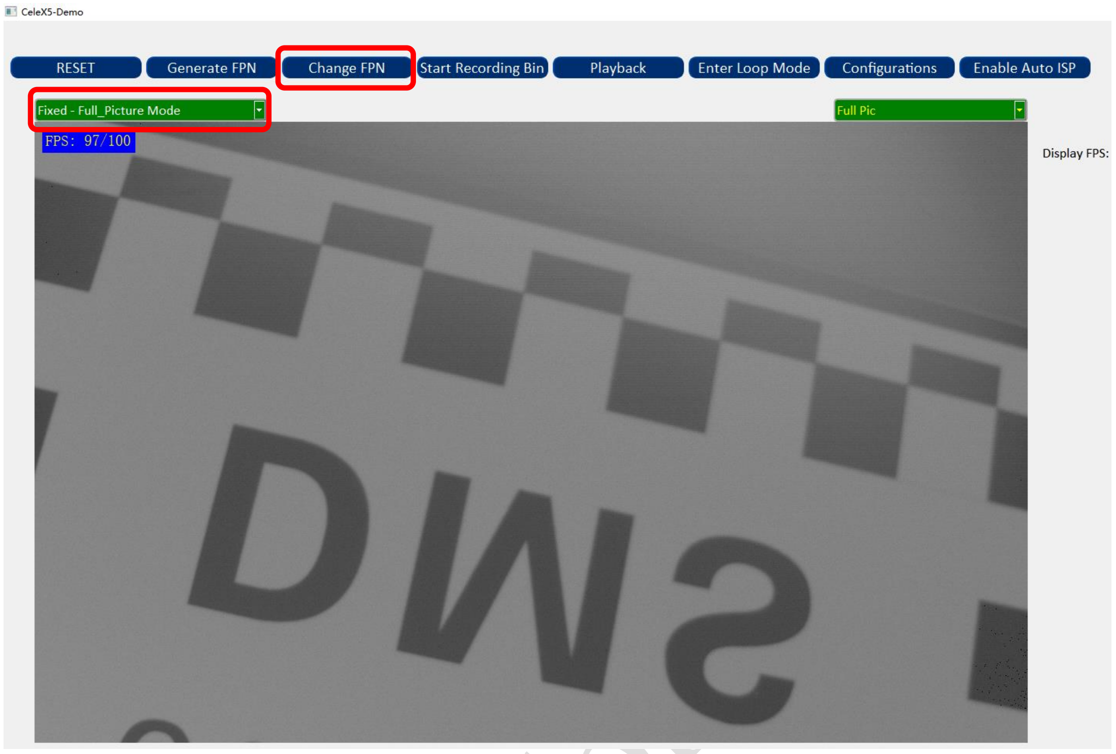
Fig. 2-4 Sensor works in Event Mode

6.2 Generate FPN file

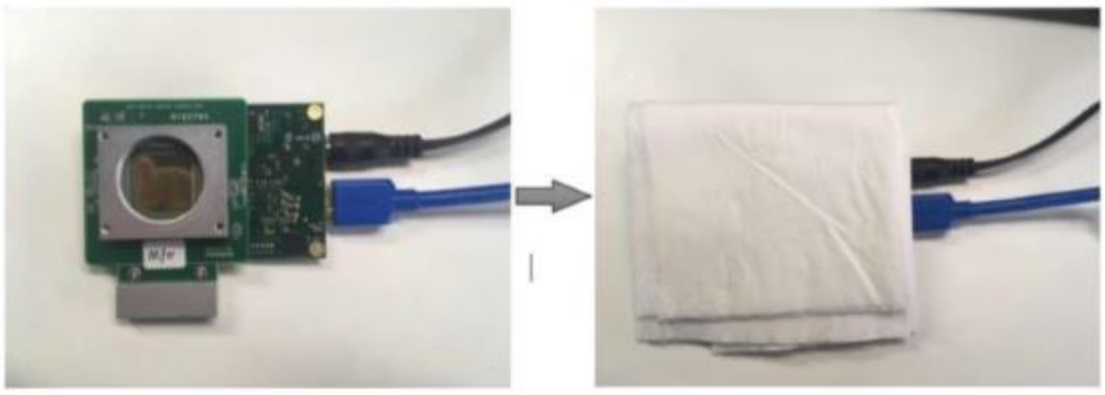
FPN (Fixed Pattern Noise) is the term given to a particular noise pattern on digital imaging sensors often noticeable during longer exposure shots where particular pixels are susceptible to giving brighter intensities above the general background noise. To get rid of FPN, we need to create a FPN file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

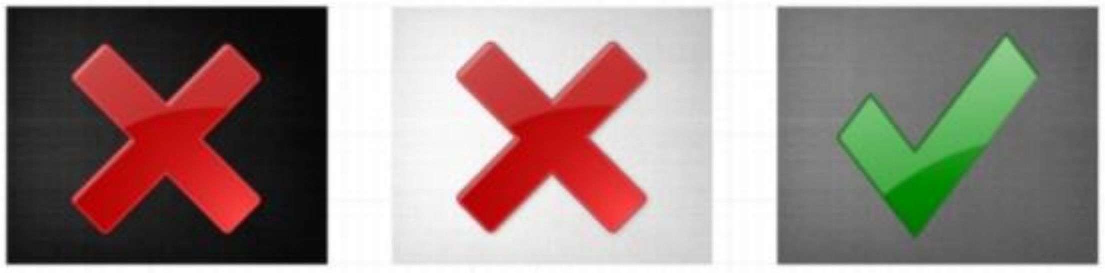
1) Switch the Sensor operating mode into " Full-frame Picture Mode".



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.txt file in assigned direction after FPN file was successfully generated.

5) It will automatically use the new generated FPN file after restarting the GUI terminal. You should be able to see the differences of image quality then.