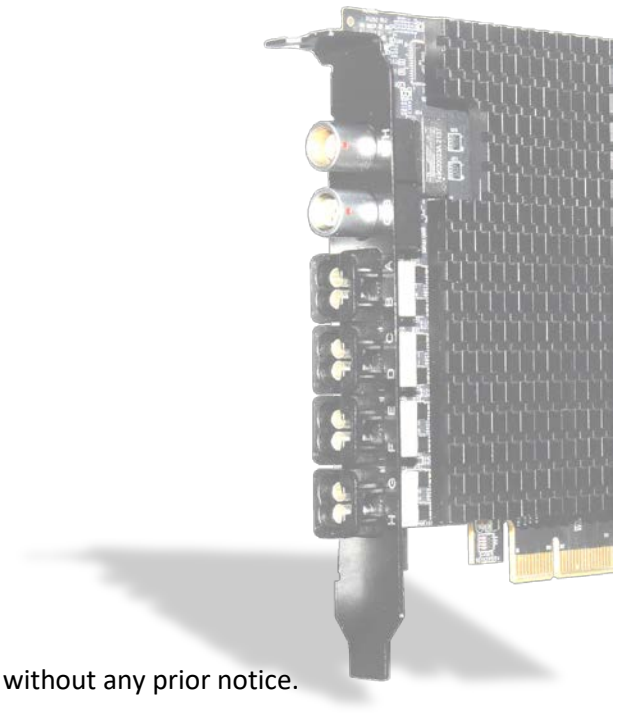
User Manual

GMSL Video capture card（Gen2）

CoaxCapture II

CCG3-4H

CCG3-8H



Version 1.2

Disclaimer

SENSING reserves the right to edit or modify this document without any prior notice.

Revision History

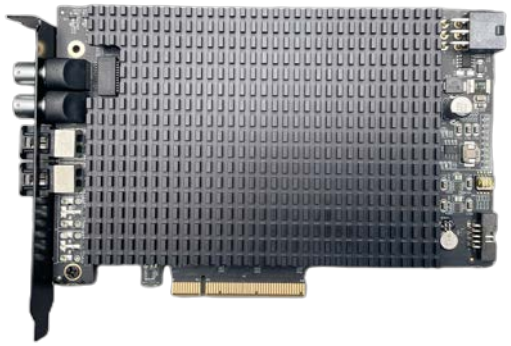
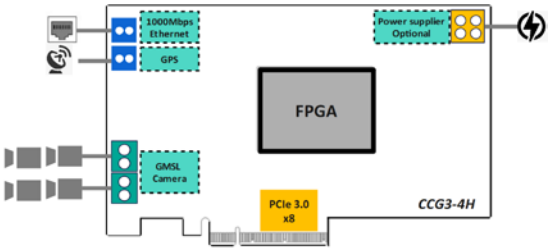
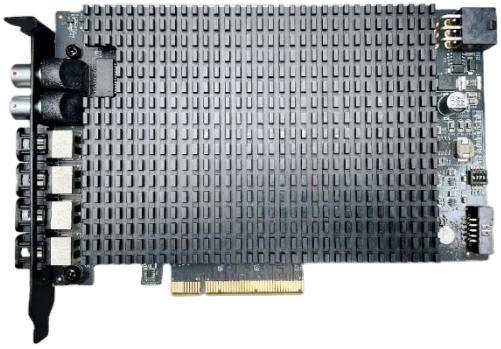
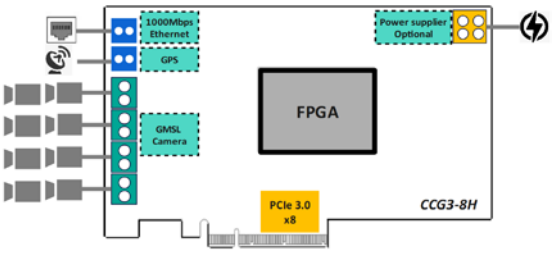
Rev	Date	Description	Author
1.0	2022.6	First edition release	R&D
1.1	2023.12	Update release	R&D
1.2	2024.6	Update time synchronization configuration, update Appendix 1	R&D

Catalogue

Chapter I Product Model

CoaxCapture II TE GMSL Video Capture card (hereinafter referred to as Capture Card) is an image capture product that can connect GMSL cameras. It has the advantages of high frame rate, high speed, low delay, etc., and it is widely used in unmanned vehicles, automatic driving, autonomous machines, data capture and other scenes.

The following are two different models of CCG3 series available for purchase.

Model	Image	Working diagram
CCG3-4H		
CCG3-8H		

*CCG3-4H for 4 GMSL cameras input

*CCG3-8H for 8 GMSL cameras input

Specification

Video Capture Card Parameter	
Mainframe Interface	PCIe Gen3 x8
Bandwidth	Continuous bandwidth up to 4800 MB/s
Onboard Memory	DDR4
Network Interface	Gigabit Ethernet: Support PTP time synchronization, accuracy<1ms
I/O Interface	TTL Trigger signal，GPS signal
Weight	300 g
Power consumption	When the power supply is 12 V, the current is about 1 A
Operation Temperature	-20°C ~ 70°C
Storage Temperature	-40°C ~ 80°C
Relative Humidity	5% ~ 90% non condensing (operation), 0% ~ 95% (storage)
MTBF	5 years
Camera Interface Specification Parameter	
Serializer	Support GMSL/GMSL2/GMSL2F interface, the corresponding chip models are as follows: MAX96705/MAX9295A/MAX96717F/MAX96717...
Connector Model	Amphenol double channel Min-fakra MFK2252AW-001-TL30G-50
Coaxial Cable	Amphenol 250mm MKA2FZ-302LL-FKMZ1-21-01
POC Power Supply	Output of each channel is 9V, power consumption is 4W
Quantity of Connect Cameras	CCG3-4H: 4 channel cameras connect CCG3-8H: 8 channel cameras connect
Camera Pixels	1-8 megapixel
Bit width	8 bit (YUV422), 8/12/16-bit (RAW)

Data bandwidth	Every channel support 6 Gbps
Camera manufacturer	SENSING (Recommended brand), If you need to access other manufacturers, you can contact them

Chapter II Typical Application

Scene 1: Automatic Driving Vehicle

The video capture card can be applied to the unmanned vehicle. It needs multiple cameras, millimeter wave radar, Lidar, GPS, integrated navigation and other sensors, and these sensors need to be connected to a powerful computing platform, such as an industrial personal computer. Then a product that can simultaneously connect multiple cameras and synchronize with other sensors is needed. The video capture card is to provide high-speed, multi camera, low delay image access solution for unmanned vehicle. The following is the overall frame diagram of a company's automatic driving fleet.

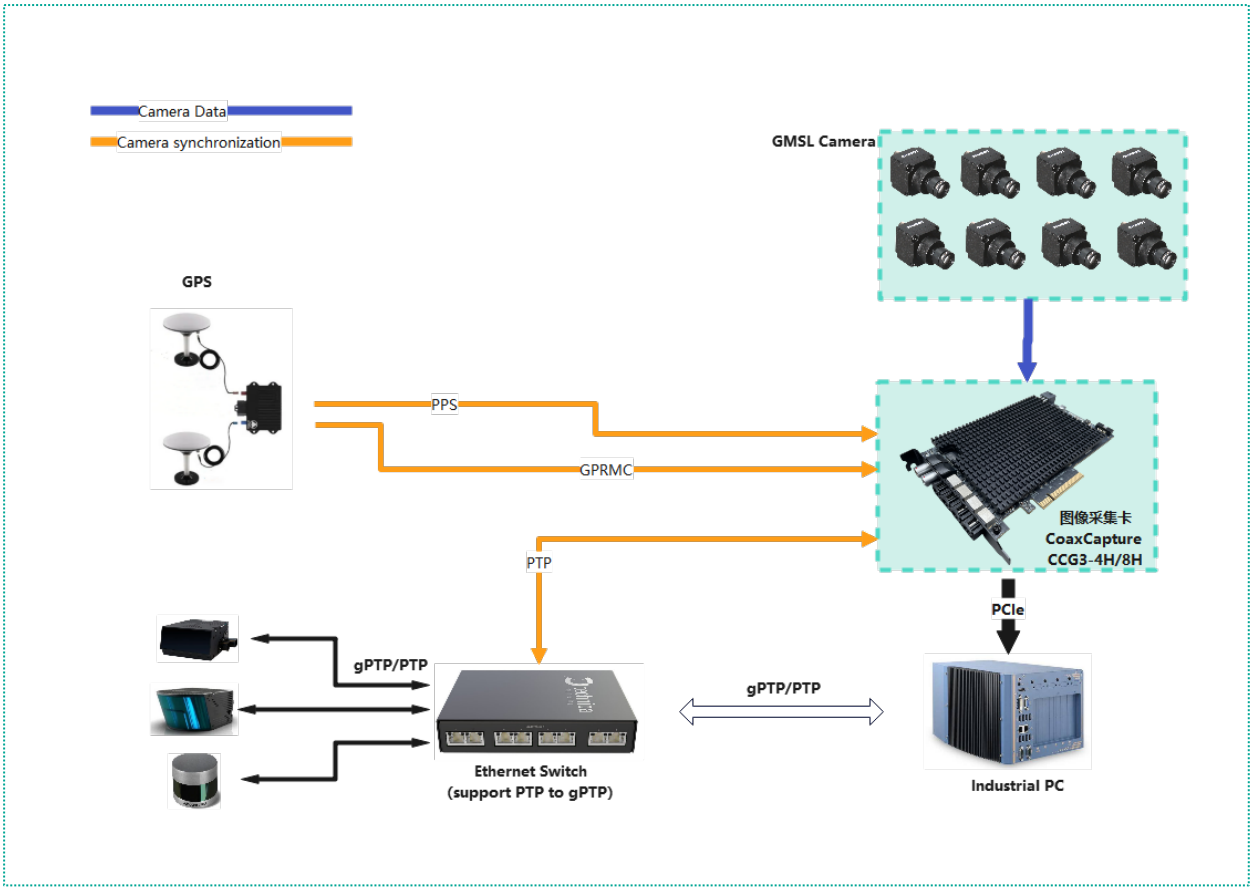


Diagram 1: autonomous vehicle frame diagram

Scene 2: Data Capture Vehicle

The data capture vehicle usually needs to collect the vehicle's multi-sensor data (camera, millimeter wave radar, Lidar, GPS, integrated navigation). At the same time, these sensor data also need to be able to be sent back to the ACU on the customer's vehicle for calculation, and the ACU calculation results are also saved. After that, the collected data is used for re-injection to the ACU for comparison of results, so as to complete the development of the ACU algorithm. The video capture card can capture the image data with low delay, and other sensors can be well aligned synchronously. In order to input camera video data to ACU and capture card at the same time, we need SENSING's SG8-BP0102-GMSL2 GMSL Bypass product, which can simultaneously send GMSL cameras to ACU and capture card without lossy.

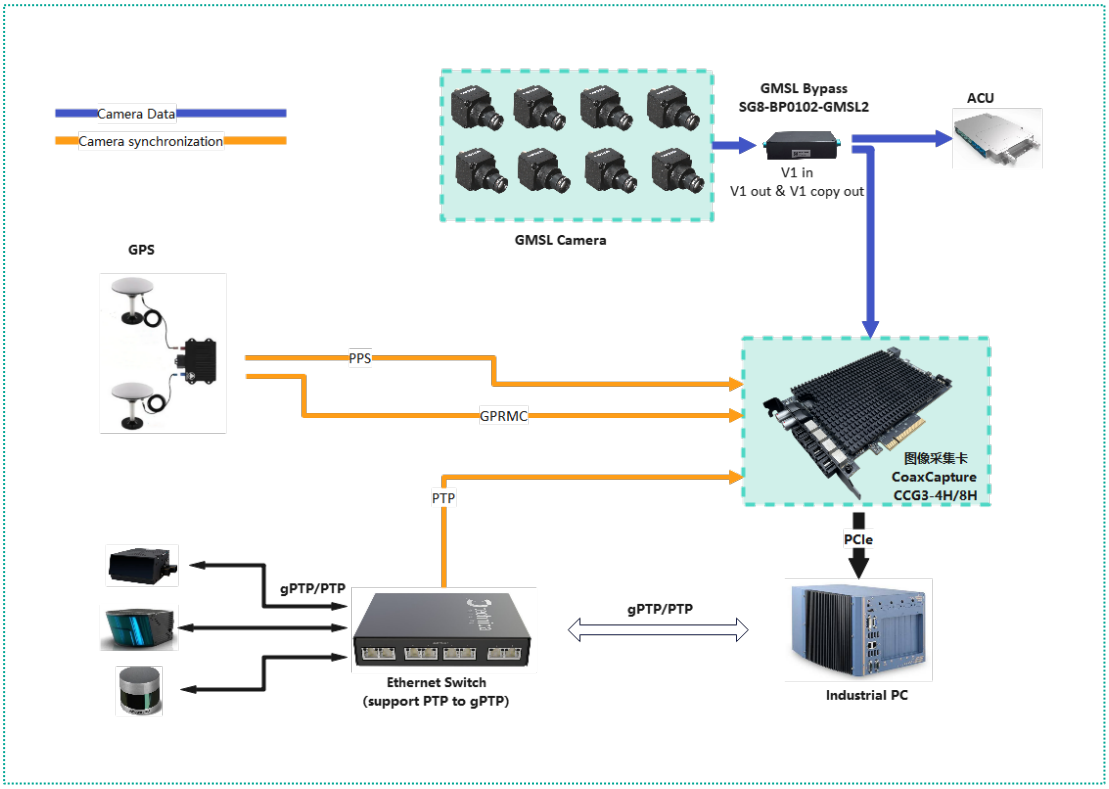


Diagram 2: data capture vehicle frame diagram

※ SG8-BP0102-GMSL2 / GMSL Bypass Product specification

产品型号/中英文名称 Model	输入路数 Input Port	输出路数 Output Port	支持分辨率和最大帧率 Resolution and max fps
SG8-BP0102-GMSL2	1路	2路	Up to 3840*2160@30fps

Chapter III Video Capture Card Installation

1 PCIe Slot Selection

As shown in Diagram 1, there are four mainstream PCIe on the mainboard: x1, x4, x8, and x16. The video capture card needs to select the appropriate slot according to Diagram 1.

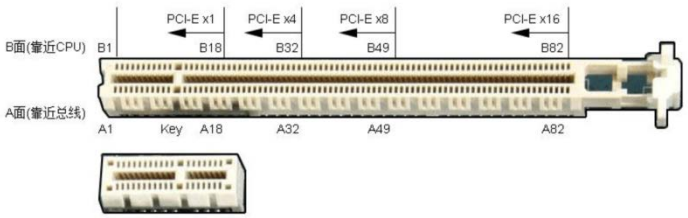


Diagram 3: PCIE slot

Table 1： Recommendation table for slot selection of capture card

Capture card model	**Slot Available**
CCG3-4H	x8、x16（**Must have an electrical signal of 4Lane or above**）
CCG3-8H	x8、x16（**Must have an electrical signal of 4Lane or above**）
Note:When selecting the slot, please note whether the slot pin is complete	

2 Baffle Fix

After the capture card is installed in an appropriate slot, the baffle should be fixed on the chassis as shown in Diagram 4 to ensure that the capture card can work stably under various operating conditions.

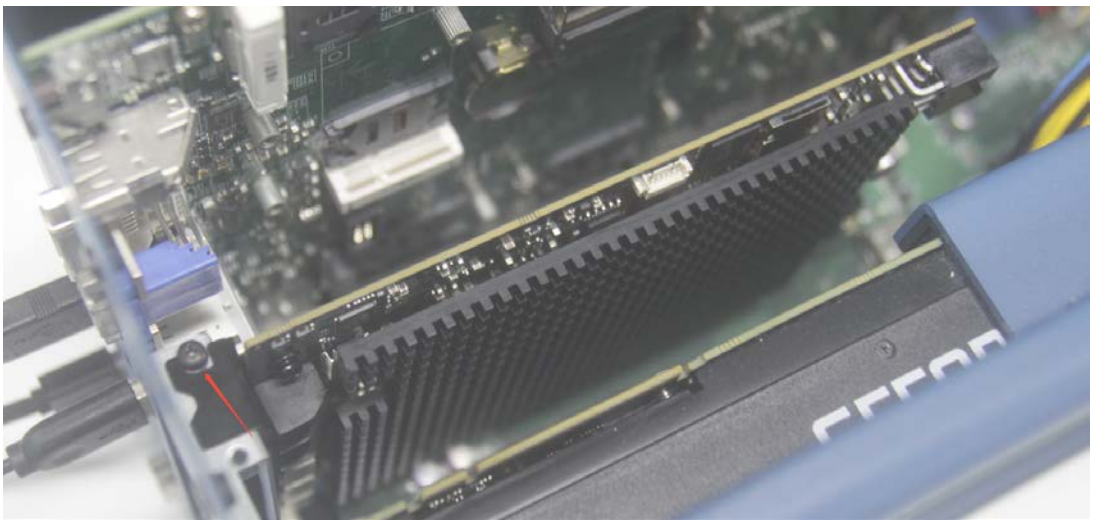
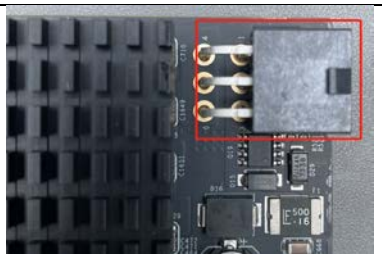


Diagram 4： Capture card baffle fixed

3 Video Capture Card Power Supply

The capture card supports PCIe slot power supply and external power supply. At present, PCIe slot power supply is mainly used, **and no external power supply is required**. If the power supply capacity of the mainboard is insufficient, external power supply can be used.

The external power supply way is as follows:

External input voltage **12±0.5V**	

连接器位置	型号	品牌/备注
板端连接器	50362462	molex
线端连接器	39012065 (白色) 39039062 (黑色) 39012060 (透明)	molex
电源+	Pin1、Pin2、Pin3	三个引脚作为电源+
GND	Pin4、Pin5、Pin6	三个引脚作为 GND

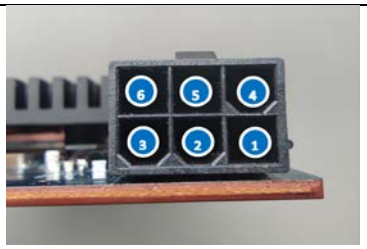


Diagram 4: external power supply interface of capture card

4 System Connection Diagram

The capture card supports 4 channels or 8 channels camera input and 1 channels GPS signal input. Refer to Diagram 6 and Diagram 7 for specific connection and use of the system.

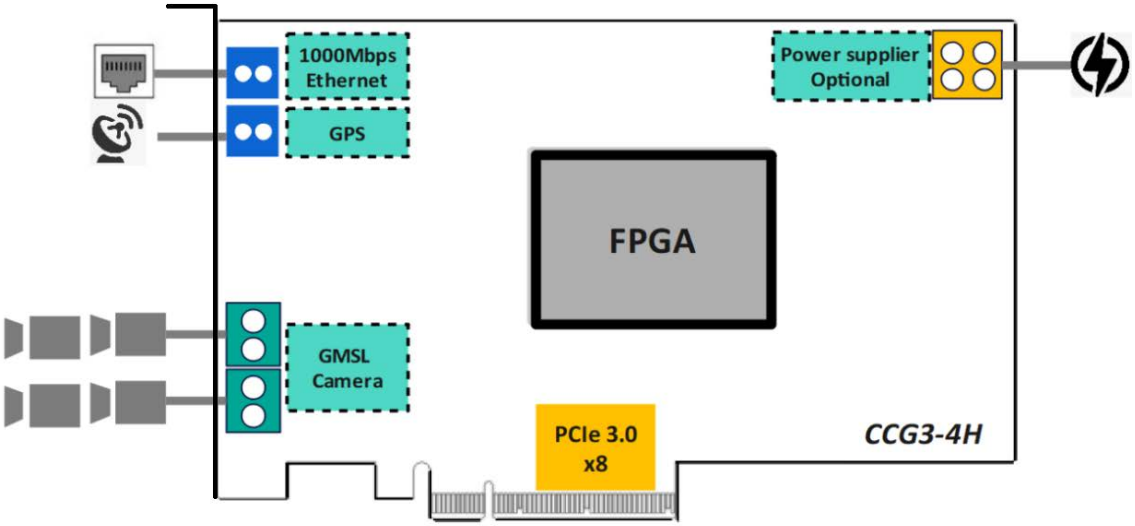


Diagram 6: CCG3-4H capture card connection diagram

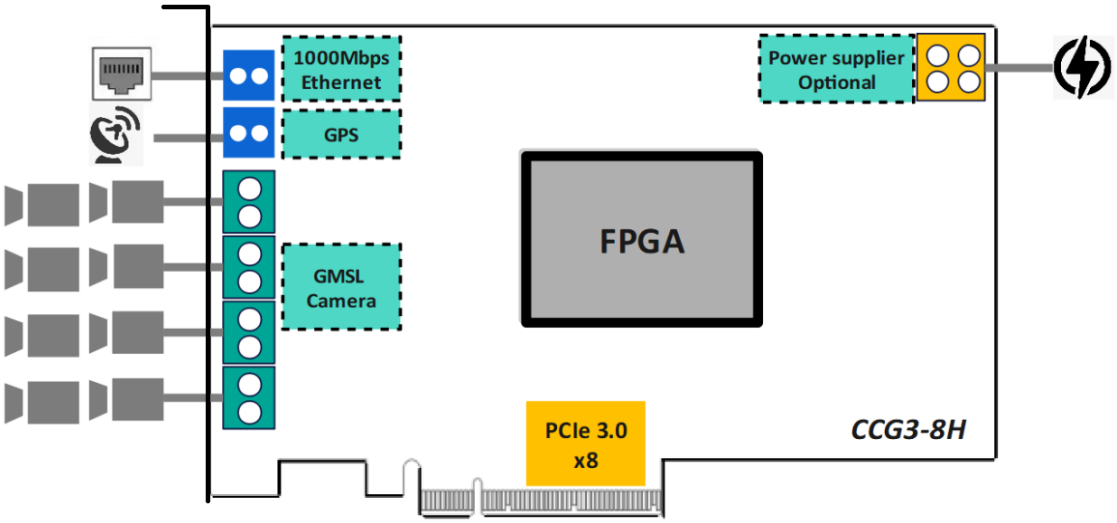


Diagram 7: CCG3-8H capture card connection diagram

5 Video Capture Card Interface Instruction

The interface of the capture card is as follows:

Diagram 8: CCG3-8H capture card interface

5.1 Camera Interface Instruction

The A, B, C, D, E, F, G and H on the baffle correspond to the 0, 1, 2, 3, 4, 5, 6 and 7 of the 8 channels camera. The "A" channel refers to the channel close to the aviation socket, and the other channels are next to each other in turn. Note: A (0) and B (1) must be connected to the same camera, C (2) and D (3) must be connected to the same camera, E (4) and F (5) must be connected to the same camera, and G (6) and H (7) must be connected to the same camera.

The card adopts a one-divided-two fakra connector, and the model is Amphenol MFK2252AW-001-TL30G-50. When connecting the camera, the customer needs to select a fakra one-divided-two switching line. The model is: Amphenol MKA2FZ-302LL-FKMZ1-21-01. As shown in the diagram below:

Diagram 9: Fakra one-divided-two switching line

5.2 Network Interface Instruction

ETH on the baffle is an aviation socket connected to the gigabit network, which needs to be connected with the aviation plug end of the cable in the following diagram. Note that the red point of the plug corresponds to the red point of the socket. The other end of the cable below is a standard crystal head, which is connected to an industrial personal computer or a switch.

Diagram 10: Aviation plug turn to network cable

5.3 GPS Interface Instruction

The GPS on the baffle is an aviation socket for connecting GPS signals. It needs to be connected with the aviation plug end of the cable in the following diagram. Note that the red point of the plug corresponds to the red point of the socket. The other end of the cable is defined in the following diagram:

Red	White	Green	Blue	Black	Bold Black
PPS Signal	UART_RX	GND	None	UART_TX	Enclosure

Note: If some cables are not used during connecting wire, they need to be cut off or insulated, and the copper core cannot be exposed.

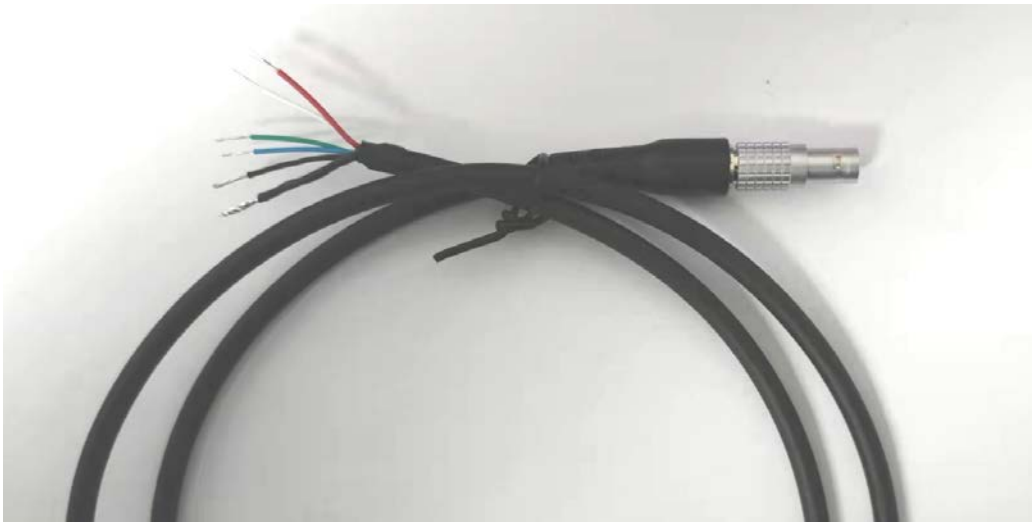


Diagram 11: GPS connection wire

Chapter IV Software Instruction

1 SDK Instruction

1.1 Driver Introduction

To facilitate customers to use the capture card speedily and efficiently, Shenzhen SENSING Technology Co., Ltd. will provide soft driver package based on ubuntu18.04 (recommended) and ubuntu20.04, the details are as follows

The video capture card SDK drive package contains the following catalogue and files:

files/catalogue name	Instruction
bash	execute Script
include	header file
xdma_v4l2	video capture card driver
tools	applications for configuring the serializer/deserializer, etc
Makefile(file)	build the overall Makefile file of the driver and application program, and it also can enter each catalogue separately to proceed separate compilation;
Readme(file)	Instructions for use of SDK drive package

1.2 Instruction

Step1：Load the driver, enter the bash catalogue, and execute the following script: (It will be success when start and load for the first time)

sudo ./load_modules.sh

When the loading is successful, the terminal shows as following:

```
Loading Pcie driver...
Pcie driver installed correctly.
Video devices were recognized.
DONE
```

After the driver is loaded, /dev catalogue will produce video0, video1…….video device files etc. And xdma0_bypass, xdma0_control, xdma0_user of files of control interface equipment, the figure 0 in the control interface device file represents the first GMSL video capture card, and the figure 1 represents the second card. If there are multiple cards, and so on.

Step 2: the GMSL video capture card need to be Initialized, and configure the camera (that is, modify the configuration file according to the camera information), then execute the following command: (It needs to be executed again when the camera is powered off or the configuration parameters are

modified)

sudo ./pcie_init_card0.sh

After successful execution, the terminal shows the following information, details as below.

```
Reset Process!
Serdes Params Init Processed!
Serdes CH 0 Done!
Serdes CH 1 Done!
Serdes CH 2 Done!
Serdes CH 3 Done!
Serdes CH 4 Done!
Serdes CH 5 Done!
Serdes CH 6 Done!
Serdes CH 7 Done!
Info: All process in pcie_init.sh passed.
```

Note: If the program cannot run successfully, and enter SDK pcie release catalogue, you can rebuild and generate drivers and applications through the make clean&make command.

Step3: verification of GMSL video capture card drive function

After initialization, GMSL video capture card can verify whether the system works normally through guvcview open source program. If the system is not installed, you can install it according to following instruction: sudo apt-get install guvcview。

The image test method is as below:

```
guvcview-d  /dev/video0
guvcview-d  /dev/video1
guvcview-d  /dev/video2
guvcview-d  /dev/video3
guvcview-d  /dev/video4
guvcview-d  /dev/video5
guvcview-d  /dev/video6
guvcview-d  /dev/video7
```

Test the cameras of each channel through the above command, and the video stream can be viewed in real time successfully, then the system works normally.

Note: For 8 channel cameras (0, 1, 2, 3, 4, 5, 6, 7), "0" means the channel close to the aviation plug, and the other channels are next to each other in turn.

2 Video Capture Card and Camera Configuration

The video capture card can connect GMSL and GMSL2 cameras, and supports different resolutions (1MP-8MP) input. At present, the supported serializers include GMS1 and GMSL2, such as MAX96705, MAX9295A, MAX96717F, and MAX96717.

2.1 Video Output Format Configuration

The video capture card supports the YUV format of the camera set by the customer. It can modify the pcie_ init_ card0.sh file to realize camera YUV format configuration, details as below:

```
camera_input_format_conversion[<n1>]=<n2>
```

- camera_input_format_conversion: camera input format conversion
- <n1>: camera number, value range 0-7
- <n2>: camera input format conversion, value is “0” or “1”

"0" represents that the card doesn't convert the camera format, and "1" represents that the card convert the camera format.

```
video_output_yuv_format    <n>    <video_format>
```

- video_output_yuv_format: video output format configuration of upper computer
- <n> : camera number, value range 0-7
- <video_format>: video output format conversion, value is “YUYV” or “UYVY”

For example:

Video 0's camera input format is YUYV, upper computer’s video output format configuration is YUYV, the configuration is as below

```
camera_input_format_conversion[0]=0
video_output_yuv_format 0 "YUYV"
```

Video 0's camera input format is YUYV, upper computer’s video output format configuration is UYVY, the configuration is as below

```
camera_input_format_conversion[0]=1
video_output_yuv_format 0 "UYVY"
```

Video 0's camera input format is UYVY, upper computer’s video output format configuration is YUYV, the configuration is as below

```
camera_input_format_conversion[0]=1
video_output_yuv_format 0 "YUYV"
```

Video 0's camera input format is UYVY, upper computer’s video output format configuration is UYVY, the configuration is as below

```
camera_input_format_conversion[0]=0
video_output_yuv_format 0 "UYVY"
```

2.2 Multi Channel Camera Trigger Configuration

2.2.1 Trigger Mode

The video capture card supports multiple trigger modes. Configure the "Trigger Mode" through the following command:

```
card_trigger_signal_mode    <"n">
```

- card_trigger_signal_mode: trigger mode configuration
- <"n">: trigger mode, value range 0-3 , “0”:no trigger; “1”:reserved; “2”:inner trigger; “3”:external trigger

For example: trigger mode is internal trigger, and the configuration is as below

```
card_trigger_signal_mode “2”
```

Note: If trigger mode is "external trigger", you need to configure "3.2.2 TTL trigger signal input frequency" and "3.2.3 external trigger output frequency of video capture card". If the trigger mode is "inner trigger", you need to configure "3.2.4 Internal trigger output frequency of video capture card".

2.2.2 TTL Trigger Signal Input Frequency

Configure the "**TTL trigger signal input frequency**" through the following command (provided that card_trigger_signal_mode is "3" and "external trigger" mode):

```
card_external_signal_input_fps        <"n"> Hz
```

- card_external_signal_input_fps: TTL trigger signal input frequency configuration
- <"n">: trigger input frequency, value range 1-30

For example: TTL trigger signal input frequency is 1Hz, configuration is as below

```
card_external_signal_input_fps “1”Hz
```

2.2.3 External Trigger Output Frequency of Video Capture Card

Configure the “**External trigger output frequency of video capture card**” through the following command (provided that card_trigger_signal_mode is "3" and "external trigger" mode):

```
camera_external_output_fps          <"n"> Hz
```

- camera_external_output_fps: external trigger output frequency of video capture card configuration
- <"n">: output frequency, value range 1-30

For example: external trigger output frequency of video capture card is 30Hz， configuration is as below

```
camera_external_output_fps “30”Hz
```

2.2.4 Internal Trigger Output Frequency of Video Capture Card

Configure the “**internal trigger output frequency of video capture card**” through the following command (provided that card_trigger_signal_mode is "2" and "inner trigger" mode):

```
camera_inner_output_fps     <"n"> Hz
```

- camera_inner_output_fps: internal trigger output frequency of video capture card configuration
- <"n">: output frequency, value range 1-30

For example: internal trigger output frequency of video capture card is 30Hz， configuration is as below

```
camera_inner_output_fps“30”Hz
```

2.2.5 Trigger Delay Setting

Configure the "**Trigger Delay**" through the following command

```
camera_triger_delay[<n1>]=<n2>
```

- camera_triger_delay: trigger delay setting command
- <n1>: camera channel, value range 0-7
- <n2>: trigger delay, unit is us

For example, the trigger delay of channel 0 camera is 5ms, and that of channel 1 camera is 10ms. The configuration is as below:

```
camera_triger_delay[0]=5000
camera_triger_delay[1]=10000
```

2.2.6 Examples of Common Synchronous Frame Rate Configuration

Examples of common frame rates for internal trigger

Internal trigger output frequency	Configuration
30 Hz	# Firstly, configuration of “trigger mode” is “inner trigger” card_trigger_signal_mode “2” #Then, configure "Internal trigger output frequency of video capture card" to 30hz camera_inner_output_fps“30”Hz
20 Hz	#Firstly, configuration of “trigger mode” is “inner trigger” card_trigger_signal_mode “2” #Then, configure "Internal trigger output frequency of video capture card" to 20hz camera_inner_output_fps“20”Hz
10 Hz	#Firstly, configuration of “trigger mode” is “inner trigger” card_trigger_signal_mode “2” #Then, configure "Internal trigger output frequency of video capture card" to 10hz camera_inner_output_fps“10”Hz

Example of TTL external trigger

TTL trigger signal input frequency	External trigger output frequency	Configuration
1Hz	30Hz	#Firstly, configuration of “trigger mode” is “external trigger” card_trigger_signal_mode “3” #Then, configure "TTL trigger signal input frequency" to 1Hz card_external_signal_input_fps “1”Hz #After that, configure "external trigger output frequency of video capture card" to 30hz camera_external_output_fps“30”Hz
1Hz	20Hz	#Firstly, configuration of “trigger mode” is “external trigger” card_trigger_signal_mode “3” #Then, configure "TTL trigger signal input frequency" to 1Hz card_external_signal_input_fps “1”Hz #After that, configure "external trigger output frequency of

		video capture card" to 20hz camera_external_output_fps“20”Hz
1Hz	10Hz	#Firstly, configuration of “trigger mode” is “external trigger” card_trigger_signal_mode “3” #Then, configure "TTL trigger signal input frequency" to 1Hz card_external_signal_input_fps “1”Hz #After that, configure "external trigger output frequency of video capture card" to 10hz camera_external_output_fps“10”Hz
10Hz	30Hz	#Firstly, configuration of “trigger mode” is “external trigger” card_trigger_signal_mode “3” #Then, configure "TTL trigger signal input frequency" to 10Hz card_external_signal_input_fps “10”Hz #After that, configure "external trigger output frequency of video capture card" to 30hz camera_external_output_fps“30”Hz
10Hz	20Hz	#Firstly, configuration of “trigger mode” is “external trigger” card_trigger_signal_mode “3” #Then, configure "TTL trigger signal input frequency" to 10Hz card_external_signal_input_fps “10”Hz #After that, configure "external trigger output frequency of video capture card" to 20hz camera_external_output_fps“20”Hz
10Hz	10Hz	#Firstly, configuration of “trigger mode” is “external trigger” card_trigger_signal_mode “3” #Then, configure "TTL trigger signal input frequency" to 10Hz card_external_signal_input_fps “10”Hz #After that, configure "external trigger output frequency of video capture card" to 10hz camera_external_output_fps“10”Hz

2.3 Multi Channel Camera Resolution Configuration

The video capture card supports different camera resolution configurations. Enter bash folder and open pcie_ init_ Card0.sh file, configure the camera resolution through following command.

Note: A (0) and B (1) must be connected to the same camera, C (2) and D (3) must be connected to the same camera, E (4) and F (5) must be connected to the same camera, and G (6) and H (7) must be connected to the same camera.

```
camera_resolution <n> <width> <height>
```

- camera_resolution: camera resolution configuration
- <n>: camera channel, value range 0-7
- <width>: image width
- <height>: image height

For example, channel 0 camera resolution is 3840 * 2160, channel 1 camera resolution is 3840 * 2160, channel 2 camera resolution is 1920 * 1080, and channel 3 camera resolution is 1920 * 1080. The configuration is as below:

```
camera_resolution 0 3840 2160
camera_resolution 1 3840 2160
camera_resolution 2 1920 1080
camera_resolution 3 1920 1080
```

2.4 Serdes Configuration

The video capture card supports configuring camera serdes to match with cameras of different models, details is as below:

```
camera_serdes_type[n]=type
```

- camera_serdes_type： camera serdes configuration
- n: camera channel, value range 0-7
- Type: the kind of camera serdes, value range0-2， 0 represents MAX9296&MAX96705， 1 represents MAX9296&MAX9295， 2 represents MAX9296&MAX96717F.

For example, appoint video 0 serdes type is MAX9296&MAX96705, which can be configured as:

```
camera_serdes_type[0]=0
```

2.5 Time Synchronization Configuration

The video capture card supports PTP time synchronization, and when powered on, it defaults to PTP slave mode. It is necessary to connect PTP master devices such as time synchronization box or industrial computer for time synchronization.

Chapter V Precautions and Maintenance

1 Precautions

*Before installing the card, it should be ensured that the IPC is completely powered off.
* Wear electrostatic gloves with both hands to install the board, so as to avoid skin direct contact the components of card.

2 Maintenance

*Storage: please store the product in a ventilated and dry place. The recommended storage temperature is - 40 ℃ to +70 ℃, and the humidity is lower than 85%.
*Transportation: shockproof packaging materials should be used to package products to avoid damage during transportation.
*Product maintenance: please don't try to repair the product. If it need to be repaired, you can contact our company in time.

Appendix 1:

The camera capture card delay test is as below:

Latency	**Description**	**2MP@30fps (SG2-IMX390C-5200-GMSL2)**	**8MP @30fps (SG8-OX08B-5300-GMSL2)**	**2MP@30fps (SG2-OX03C-5200-GMSL2)**
Link	Trigger-->MIPI OUT	55.9827ms	36.4ms	41.4768ms
Camera Transport	MIPI OUT-->Frame in card DDR	31.9818ms	32.2ms	32.2ms
IRQ Transport	Card IRQ OUT-->PC response	0.083ms	0.052ms	0.036ms
DMA Transport	Frame in card DDR-->Frame in PC DDR(DMA)	0.9ms	3.6ms	0.9ms
Total		88.9475ms	72.252ms	74.6128ms

Appendix 2:

The timestamp that the PC obtains the current image is the moment when the PC receives the interrupt signal from video capture card. If timestamps at other times are required, they can be calculated according to Appendix 1.

There are two methods to obtain the timestamp of the video capture card:

a) Obtain timestamp through v4l2 framework

Under Linux circumstances, the sensing camera hardware has been mapped as the device file "/dev/video0", and it uses timestamp value (type timestamp) of v4l2_buffer. The customer obtains v4l2_ Buffer by themselves according to the v4l2 framework.

3.5.2. struct v4l2_buffer

struct v4l2_buffer

__u32	index	Number of the buffer, set by the application except when calling VIDIOC_DQBUF, then it is set by the driver. This field can range from zero to the number of buffers allocated with the ioctl VIDIOC_REQBUFS ioctl (struct v4l2_requestbuffers count), plus any buffers allocated with ioctl VIDIOC_CREATE_BUFS minus one.
__u32	type	Type of the buffer, same as struct v4l2_format type or struct v4l2_requestbuffers type, set by the application. See v4l2_buf_type
__u32	bytesused	The number of bytes occupied by the data in the buffer. It depends on the negotiated data format and may change with each buffer for compressed variable size data like JPEG images. Drivers must set this field when type refers to a capture stream, applications when it refers to an output stream. If the application sets this to 0 for an output stream, then bytesused will be set to the size of the buffer (see the length field of this struct) by the driver. For multiplanar formats this field is ignored and the planes pointer is used instead.
__u32	flags	Flags set by the application or driver, see Buffer Flags.
__u32	field	Indicates the field order of the image in the buffer, see v4l2_field. This field is not used when the buffer contains VBI data. Drivers must set it when type refers to a capture stream, applications when it refers to an output stream.
struct timeval	timestamp	For capture streams this is time when the first data byte was captured, as returned by the clock_gettime() function for the relevant clock id; see V4L2_BUF_FLAG_TIMESTAMP_* in Buffer Flags. For output streams the driver stores the time at which the last data byte was actually sent out in the timestamp field. This permits applications to monitor the drift between the video and system clock. For output streams that use V4L2_BUF_FLAG_TIMESTAMP_COPY the application has to fill in the timestamp which will be copied by the driver to the capture stream.
struct v4l2_timecode	timecode	When the V4L2_BUF_FLAG_TIMECODE flag is set in flags, this structure contains a frame timecode. In V4L2_FIELD_ALTERNATE mode the top and bottom field contain the same timecode. Timecodes are intended to help video editing and are typically recorded on video tapes, but also embedded in compressed formats like MPEG. This field is independent of the timestamp and sequence fields.
__u32	sequence	Set by the driver, counting the frames (not fields!) in sequence. This field is set for both input and output devices.

In V4L2_FIELD_ALTERNATE mode the top and bottom field have the same sequence number. The count starts at zero and includes dropped or repeated frames. A dropped frame was received by an input device but could not be stored due to lack of free buffer space. A repeated frame was displayed again by an output device because the application did not pass new data in time.

Note

This may count the frames received e.g. over USB, without taking into account the frames dropped by the remote hardware due to limited compression throughput or bus bandwidth. These devices identify by not enumerating any video standards, see Video Standards.

eg:

*v4l2_buffer *argp;*

struct timeval timestamp;

*int ioctl(int fd, VIDIOC_DQBUF, struct v4l2_buffer *argp);*

timestamp = argp->timestamp;

b) View the timestamp of the output image directly (for reference only)

v4l2-ctl --stream-mmap --stream-count=3 -d /dev/video0 --verbose

Contact Us

SZ Sensing TECH CO.,LTD

HQ Address: 8/F,block B, building 1, wisdom home, No.76 Baohe Avenue, Baolong street, Longgang District, Shenzhen

Phone: +86-755-28990915

Email: Sales@sensing-world.com

Company Website: www.sensing-world.com

Click the following website to contact the manufacturer

https://www.sensing-world.com/lxwm

WECHAT

SKYPE

WhatsApp