2022

EHD ICM-Series Manual

EHD imaging GmbH

3.5.2022

1.1 ICM series monochrome camera specifications

Model Number	Image Sensor	Pixel Size(μm)	G Sensitivity/Dark Signal	FPS/Resolution	Binning	Exposure Time
ICM433M-TR IM700500A	0.5M/IMX433LLJ(M, GS) 1/1.7” (7.31x5.58)	9.0x9.0	8100mv with 1/30s 0.30mv with 1/30s	166.5fps@812 × 620	1x1	6us~15s
ICM273M-TR IM701500A	1.5M/IMX273LLR(M, GS) 1/2.9“(4.97x3.73)	3.45×3.45	1830mv with 1/30s 0.19mv with 1/30s	226.5fps@1440 × 1080 506fps@720 × 540	1x1 2x2	15us~15s
ICM252M-TR IM703100A	3.1M/IMX252LLR(M, GS) 1/1.8“(7.07x5.30)	3.45×3.45	1830mv with 1/30s 0.15mv with 1/30s	110.6fps@2048 × 1536 233.8fps@1024 × 768	1x1 1x1	15us~15s
ICM265M-TR IM703100B	3.1M/IMX265LLR(M, GS) 1/1.8“(7.07x5.30)	3.45×3.45	1830mv with 1/30s 0.15mv with 1/30s	55.4fps@2048 × 1536 115.1fps@1024 × 768	1x1 1x1	15us~15s
ICM250M-TR IM705000A	5.0M/IMX250LLR(M, GS) 2/3“(8.45x7.07)	3.45×3.45	1830mv with 1/30s 0.15mv with 1/30s	70.9fps@2448 × 2048 175.2fps@1224 × 1024	1x1 1x1	15us~15s
ICM264M-TR IM705000B	5.0M/IMX264LLR(M, GS) 2/3“(8.45x7.07)	3.45×3.45	1830mv with 1/30s 0.15mv with 1/30s	35.6fps@2448 × 2048 87.6fps@1224 × 768	1x1 1x1	15us~15s
ICM178M-TR IM706300A	6.3M/IMX178LLJ(M, RS) 1/1.8“ (7.37x4.92)	2.4x2.4	760mv with 1/30s 0.15mv with 1/30s	58.7fps@3072 × 2048 59.5fps@1536 × 1024	1x1 2x2	17us~15s

M: Monochromatic; C: Color; RS: Rolling Shutter; GS: Global Shutter.

2 ICM series technical specifications

2.1 ICM433M-TR

Model Parameter	ICM433-M-TR
	0.5M pixel 1 / 1.7 "CMOS USB3.0 industrial camera
Camera	
Sensor model	Sony IMX433LLJ
Pixel size	9.0 µm × 9.0 µm
Sensor size	1/1.7”
Frame rate	166.5fps@812 × 620
Dynamic range	71dB
Signal-to-Noise ratio	40dB
Sensitivity	8100mV
Dark current	0.3mV
Gain range	1x-50x
Exposure time	6µs-15sec
Shutter	Global Shutter
Binning	software2×2, 3×3, 4×4
Data interface	USB3.0(USB3.1 GEN1)
Digital I/O	One optical-coupling isolated input, one optical-coupling isolated output, one non-isolated input, one non-isolated output
Data Format	Mono8 / Mono12
General Specifications	
Power supply	Power with USB3.0
Power consumption	<3.5W
Temperature	Working temperature -10~50℃; Storage temperature -30~70℃
Humidity	20% - 80% No condensation
Size	33mm×33mm×33mm
Weight	70g
Lens mount	C-mount
Software	EHDView/SDK
Operating system	Win32/WinRT/Linux/macOS/Android
Certification	CE, FCC

Table 2-1 ICM433M-TR camera specifications

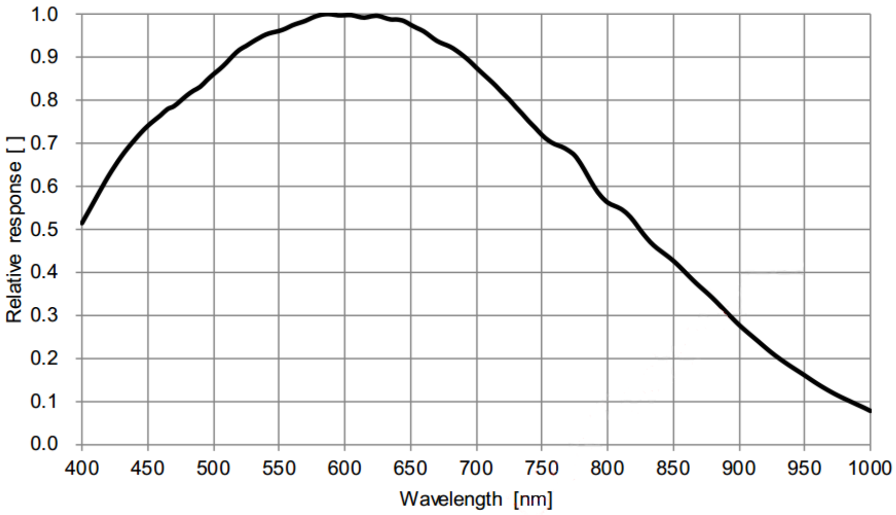


Figure 2-1 ICM433M-TR spectral response curve

2.2 ICM273M-TR

Model Parameter		**ICM273M-TR**
		1.5M pixels 1/2.9" CMOS USB3.0 industrial camera
Camera		
Sensor model		Sony IMX273LLR
Pixel size		3.45 µm × 3.45 µm
Sensor size		1/2.9"
Frame rate		226.5fps@1440 × 1080, 506fps@720 × 540
Dynamic range		71dB
Signal-to-Noise ratio		40dB
Sensitivity		1830mV
Dark current		0.19mV
Gain range		1x-50x
Exposure time		15µs-15sec
Shutter		Global shutter
Binning		hardware2×2; software2×2, 3×3, 4×4
Data interface		USB3.0 (USB3.1 GEN1)
Digital I/O	Hardware version number V1.0	One optical-coupling isolated input, one optical-coupling isolated output, one non-isolated input, one non-isolated output
	Hardware version number V2.0 and above	One opto-coupling isolated input, one opto-coupling isolated output, one non-isolated input/output
Data Format		Mono8 / Mono12
General specification		
Power supply		Power with USB3.0
Power consumption		<3.5W
Temperature		Working temperature -10~50℃, storage temperature -30~70℃
Humidity		20%-80%, no condensation
Size		33mm×33mm×33mm
Weight		70g
Lens mount		C-mount
Software		EHDView/ SDK
Operating system		Win32/WinRT/Linux/macOS/Android
Certification		CE, FCC

Table 2-2 ICM273M-TR camera specifications

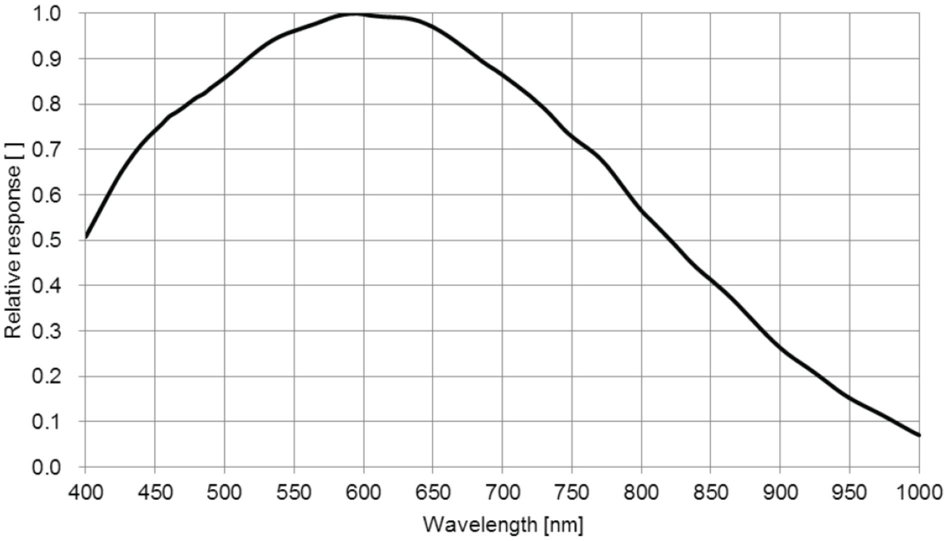


Figure 2-2 ICM273M-TR spectral response curve

2.3 ICM252M-TR

Model Parameter		ICM252M-TR
		3.1M pixels 1/1.8" CMOS USB3.0 industrial camera
Camera		
Sensor model		Sony IMX252LLR
Pixel size		3.45 µm × 3.45 µm
Sensor size		1/1.8"
Frame rate		110.6fps@2048 × 1536, 233.8fps@1024 × 768
Dynamic range		75dB
Signal-to-Noise ratio		40dB
Sensitivity		1830mV
Dark current		0.15mV
Gain range		1x-50x
Exposure time		15µs-15sec
Shutter		Global shutter
Binning		Software 2×2, 3×3, 4×4
Data interface		USB3.0(USB3.1 GEN1)
Digital IO	Hardware version number V1.0	One optical-coupling isolated input, one optical-coupling isolated output, one non-isolated input, one non-isolated output
	Hardware version number V2.0 and above	One opto-coupling isolated input, one opto-coupling isolated output, one non-isolated input / output
Data format		Mono8 / Mono12
General specification		
Power supply		Power with USB3.0
Power consumption		<3.5W
Temperature		Working temperature -10~50℃, storage temperature-30~70℃
Humidity		20%-80%, no condensation
Size		33mm×33mm×33mm
Weight		70g
Lens mount		C-mount
Software		EHDView/ SDK
Operating system		Win32/WinRT/Linux/macOS/Android
Certification		CE, FCC

Table 2-3 ICM273M-TR camera specifications

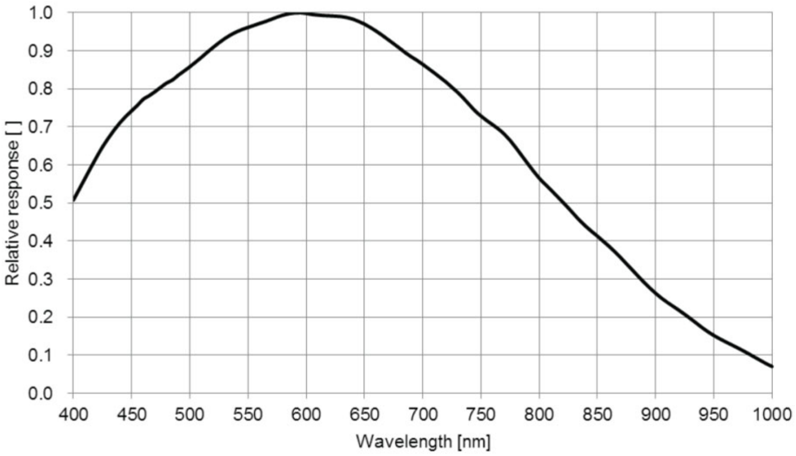


Figure 2-3 ICM273M-TR spectral response curve

2.4 ICM265M-TR

Model / Parameter		ICM265M-TR
		3.1M pixels 1/1.8" CMOS USB3.0 industrial camera
Camera		
Sensor model		Sony IMX265LLR
Pixel size		3.45 µm × 3.45 µm
Sensor size		1/1.8"
Frame rate		55.4fps@2048 × 1536,115.1fps@1024 × 768
Dynamic range		75dB
Signal-to-Noise ratio		40dB
Sensitivity		1830mV
Dark current		0.15mV
Gain range		1x-50x
Exposure time		15µs-15sec
Shutter		Global shutter
Binning		Software 2×2, 3×3, 4×4
Data interface		USB3.0(USB3.1 GEN1)
Digital IO	Hardware version number V1.0	One optical-coupling isolated input, one optical-coupling isolated output, one non-isolated input, one non-isolated output
	Hardware version number V2.0 and above	One opto-coupling isolated input, one opto-coupling isolated output, one non-isolated input / output
Data format		Mono8 / Mono12
General specification		
Power supply		Power with USB3.0
Power consumption		<3.5W
Temperature		Working temperature -10~50℃, storage temperature-30~70℃
Humidity		20%-80%, no condensation
Size		33mm×33mm×33mm
Weight		70g
Lens mount		C-mount
Software		EHDView/ SDK
Operating system		Win32/WinRT/Linux/macOS/Android
Certification		CE, FCC

Table 2-4 ICM265M-TR specifications

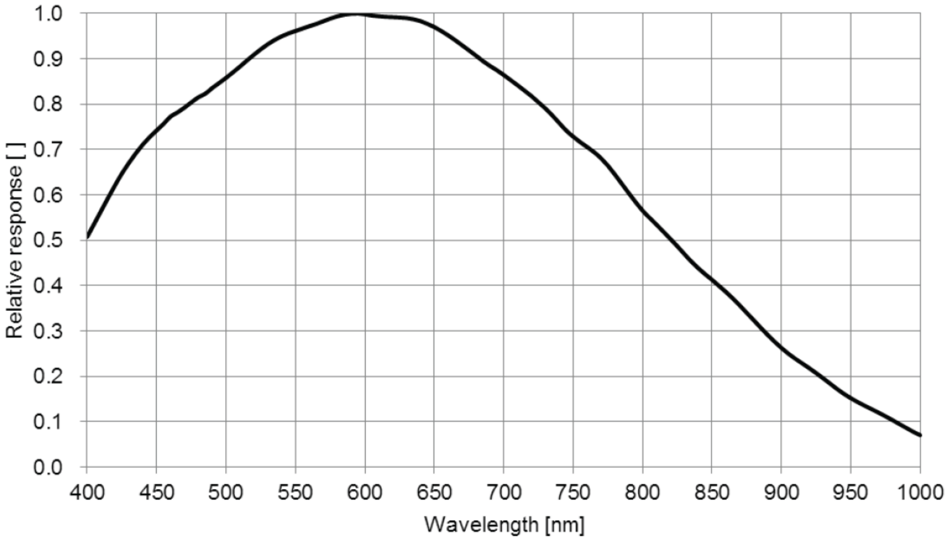


Figure 2-4 ICM265M-TR spectral response curve

2.5 ICM250M-TR

Model Parameter		ICM250M-TR
		5M pixels 2/3” CMOS USB3.0 industrial camera
Camera		
Sensor model		Sony IMX250LLR
Pixel size		3.45 µm × 3.45 µm
Sensor size		2/3”
Frame rate		70.9fps@2448 × 2048, 175.2fps@1224 × 1024
Dynamic range		75dB
Signal-to-Noise ratio		40dB
Sensitivity		1830mV
Dark current		0.15mV
Gain range		1x-50x
Exposure time		15µs-15sec
Shutter		global shutter
Binning		software 2×2, 3×3, 4×4
Data interface		USB3.0(USB3.1 GEN1)
Digital IO	Hardware version number V1.0	One optical-coupling isolated input, one optical-coupling isolated output, one non-isolated input, one non-isolated output
	Hardware version number V2.0 and above	One opto-coupling isolated input, one opto-coupling isolated output, one non-isolated input / output
Data format		Mono8 / Mono12
General specification		
Power supply		Power with USB3.0
Power consumption		<3.5W
Temperature		Working temperature -10~50℃, storage temperature-30~70℃
Humidity		20%-80%, no condensation
Size		33mm×33mm×33mm
Weight		70g
Lens mount		C-mount
Software		EHDView/ SDK
Operating system		Win32/WinRT/Linux/macOS/Android
Certification		CE, FCC

Table 2-5 ICM250M-TR camera specification

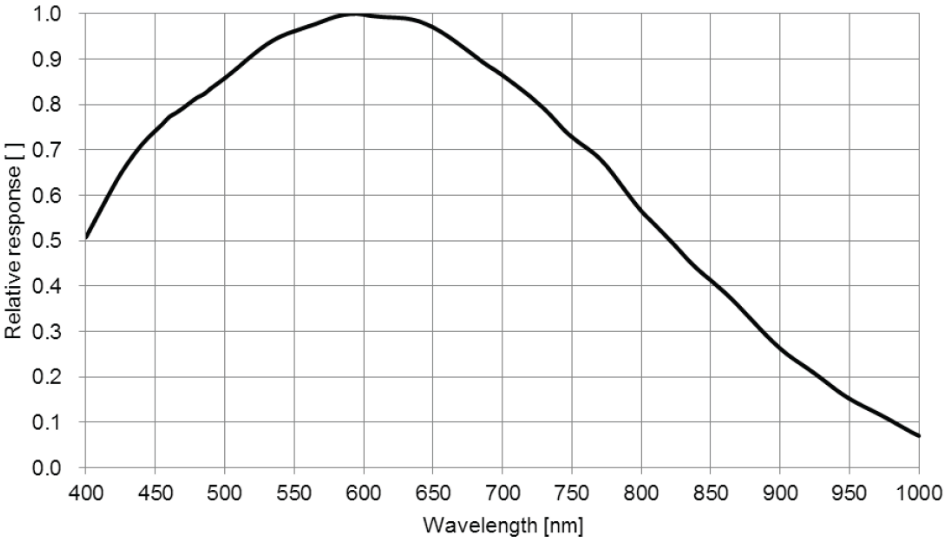


Figure 2-5 ICM250M-TR spectral response curve

2.6 ICM264M-TR

Model Parameter		ICM264M-TR
		5M pixels 2/3” CMOS USB3.0 industrial camera
Camera		
Sensor model		Sony IMX264LLR
Pixel size		3.45 µm × 3.45 µm
Sensor size		2/3”
Frame rate		35.6fps@2448 × 2048,87.6fps@1224 × 1024
Dynamic range		75dB
Signal-to-Noise ratio		40dB
Sensitivity		1830mV
Dark current		0.15mV
Gain range		1x-50x
Exposure time		15µs-15sec
Shutter		Global shutter
Binning		Software 2×2, 3×3, 4×4
Data interface		USB3.0(USB3.1 GEN1)
Digital IO	Hardware version number V1.0	One optical-coupling isolated input, one optical-coupling isolated output, one non-isolated input, one non-isolated output
	Hardware version number V2.0 and above	One opto-coupling isolated input, one opto-coupling isolated output, one non-isolated input / output
Data format		Mono8 / Mono12
General specification		
Power supply		Power with USB3.0
Power consumption		<3.5W
Temperature		Working temperature -10~50℃, storage temperature 30~70℃
Humidity		20%-80%, no condensation
Size		33mm×33mm×33mm
Weight		70g
Lens mount		C-mount
Software		EHDView/ SDK
Operating system		Win32/WinRT/Linux/macOS/Android
Certification		CE, FCC

Table 2-6 ICM264M-TR camera specifications

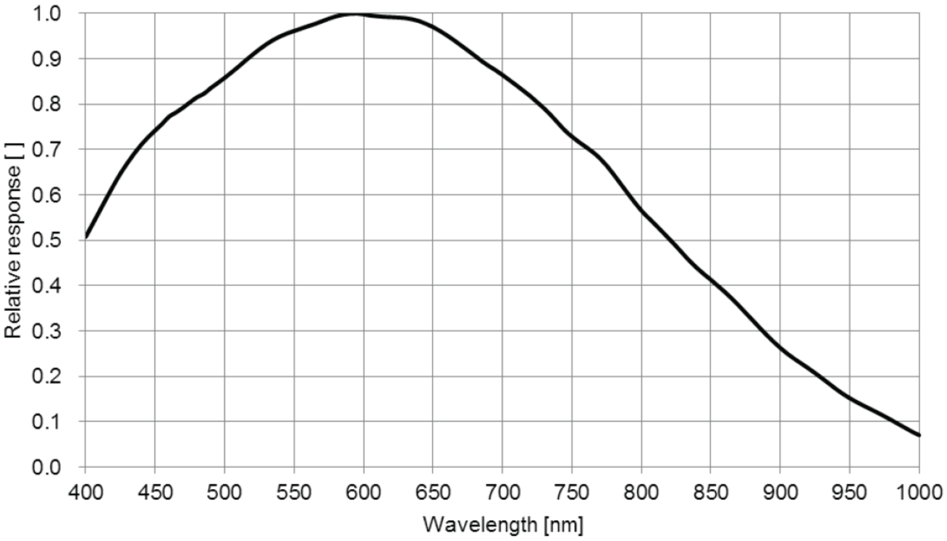


Figure 2-6 ICM264M-TR spectral response curve

2.7 ICM178M-TR

Model Parameter		ICM178M-TR
		6.3M pixels 1/1.8” CMOS USB3.0 industrial camera
Camera		
Sensor model		Sony IMX178LLJ
Pixel size		2.4 µm × 2.4 µm
Sensor size		1/1.8”
Frame rate		58.7fps@3072 × 2048,59.5fps@1536 × 1024
Dynamic range		71dB
Signal-to-Noise ratio		40dB
Sensitivity		760mV
Dark current		0.15mV
Gain range		1x-50x
Exposure time		17µs-15sec
Shutter		Rolling shutter
Binning		Hardware 2x2;Software 2×2, 3×3, 4×4
Data interface		USB3.0(USB3.1 GEN1)
Digital IO	Hardware version number V1.0	One optical-coupling isolated input, one optical-coupling isolated output, one non-isolated input, one non-isolated output
	Hardware version number V2.0 and above	One opto-coupling isolated input, one opto-coupling isolated output, one non-isolated input / output
Data format		Mono8 / Mono12
General specification		
Power supply		Power with USB3.0
Power consumption		<3.5W
Temperature		Working temperature -10~50℃, storage temperature 30~70℃
Humidity		20%-80%, no condensation
Size		33mm×33mm×33mm
Weight		70g
Lens mount		C-mount
Software		EHDView/ SDK
Operating system		Win32/WinRT/Linux/macOS/Android
Certification		CE, FCC

Table 2-7 ICM178M-TR camera specifications

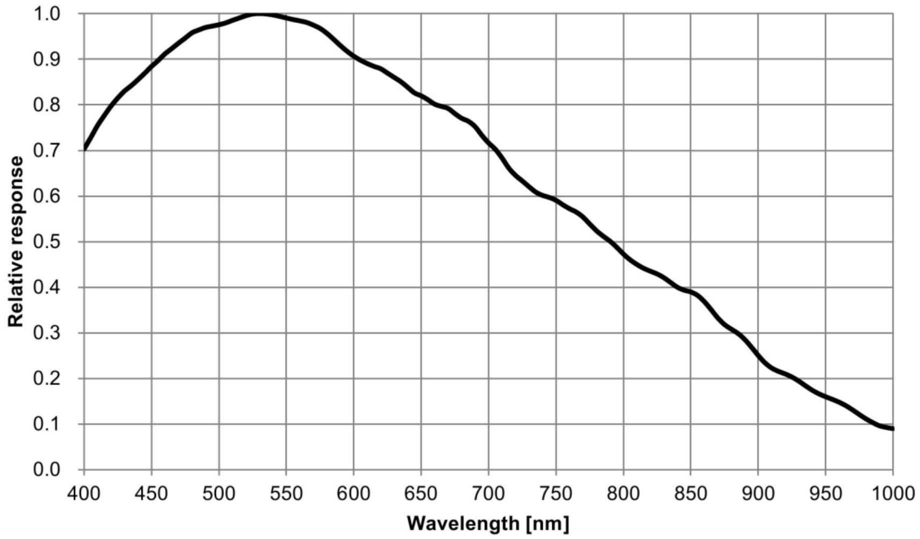


Figure 2-7 ICM178M-TR spectral response curve

3 Camera Dimension and Interface

3.1 ICM series camera dimensions and outputs

3.1.1 ICM series camera mechanical housing dimensions

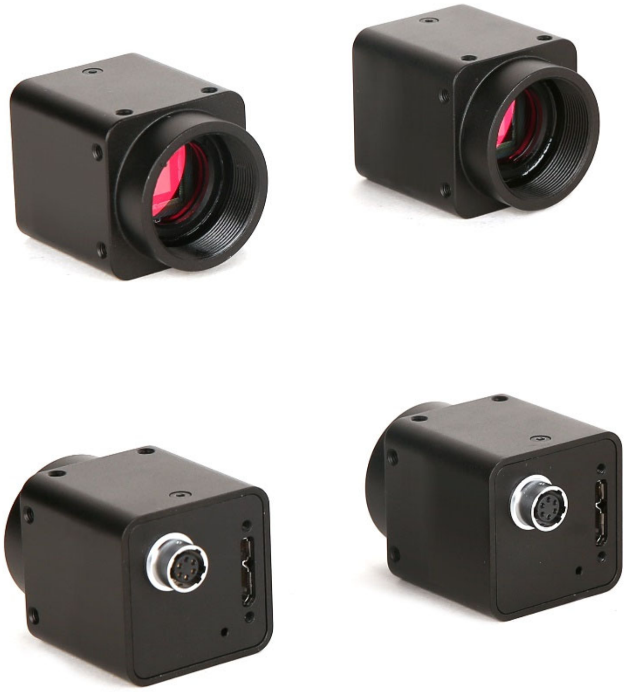


Figure 3-1 ICM series camera

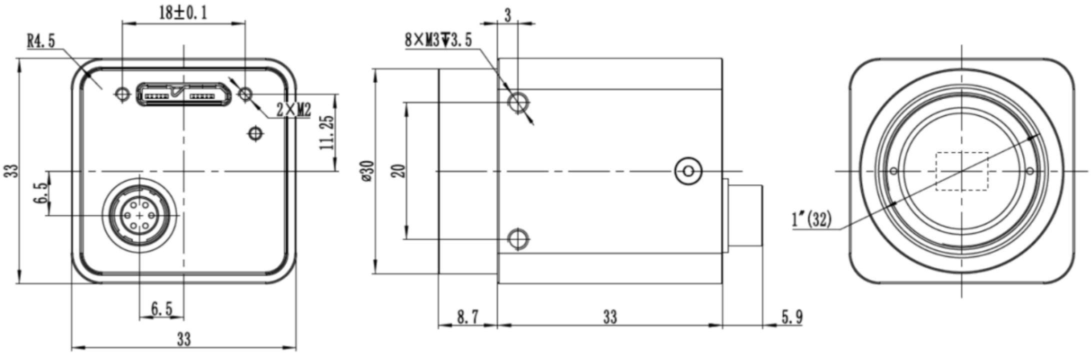


Figure 3-2 Dimensions of camera housing (mm)

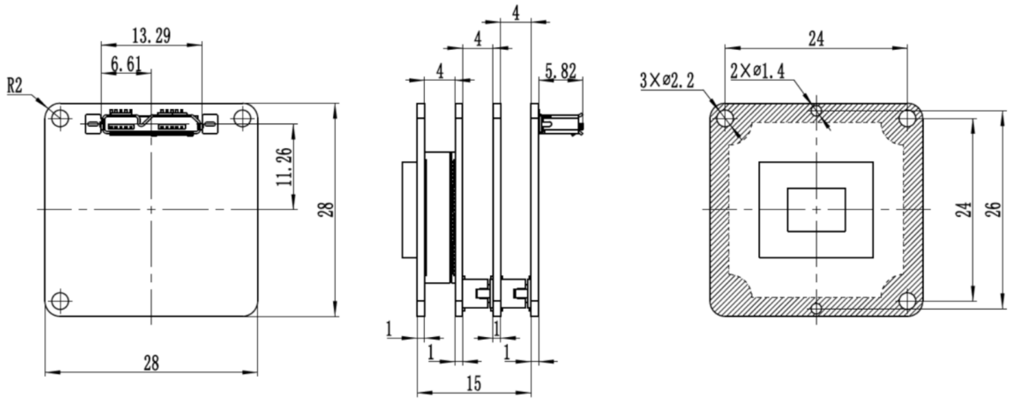


Figure 3-3 Dimensions of circuit board(mm)

3.1.2 ICM series camera interface

The back of the industrial camera is shown in Figure 3-4. It has standard USB3.0 output, 6 Pin I/O port (aviation head) and on/off indicator. It has two M2 screw holes on both sides of USB 3.0 port to fix the cable. The holes reduce cable loosening caused by field vibration.

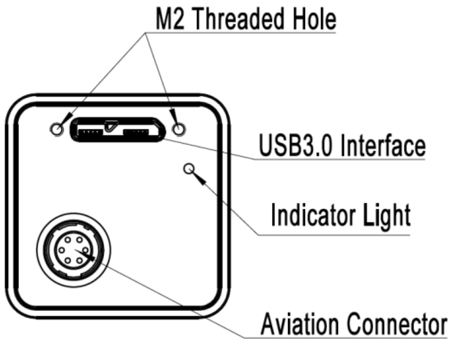


Figure 3-4 Schematic diagram of camera back panel

3.1.3 ICM series camera power supply and I/O connector

The pin signal definition for the camera 6 Pin I/O connector of hardware version V1.0 is shown in Table 3-1.

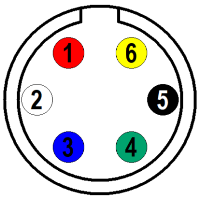
	Color	Pin	Signal	Signal description
	red	1	DIR_IN	Direct-coupled input signal (line2)
	white	2	OPTO_GND	Opto-isolated signal ground
	blue	3	OPTO_OUT	Opto-isolated output signal(line1)
	green	4	OPTO_IN	Opto-isolated input signal(line0)
	black	5	GND	Direct-coupled signal ground
	yellow	6	DIR_OUT	Direct-coupled output signal(line3)

Table 3-1 Pin signal definition

The pin signal definition for the camera 6 Pin I/O connector with hardware version number V2.0 and above is shown in Table 3-2.

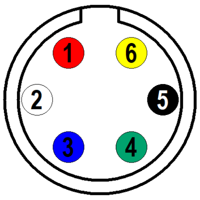
	Color	Pin	Signal	Signal description
	red	1	DIR_GPIO	Direct-coupled General Purpose I/O (Software configurable input / output) (line2)
	white	2	OPTO_GND	Opto-isolated signal ground
	blue	3	OPTO_OUT	Opto-isolated output signal(line1)
	green	4	OPTO_IN	Opto-isolated input signal(line0)
	black	5	GND	Direct-coupled signal ground
	yellow	6	5V	5 VDC power input

Table 3-2 V2.0 and above pin signal definitions

3.1.4 ICM series camera packing Information

For normal use of industrial cameras, please prepare the required accessories as shown in Table 3-3 before installation.

Order number	Accessories name	Quantity	Instruction
1	Camera	1	Camera referred in this manual
2	I/O cable	1	6 Pin cable or extended cable
3	USB3.0 cable	1	Suitable length of Micro USB3.0 cable
4	Lens (optional)	1	C-mount lens

Table 3-3 Packing information and recommended accessories

4. Description of Software Development

4.1 SDK introduction

4.1.1 SDK support platform

- Win32:
  x86: XP SP3 and above versions; CPU needs to support the SSE2 instruction set at least
  x64: Win7 and above versions
  arm: Win10 and above versions
  arm64: Win10 and above versions
- WinRT: x86, x64, arm, arm64□Windows 10 or above versions
- macOS: x86 and x64 bundle□macOS 10.10 or above versions
- Linux: core 2.6.27 and above versions
  x86: CPU needs to support at least SSE3 instruction sets; GLIBC 2.8 and above
- x64: GLIBC 2.14 and above versions
- armel: GLIBC 2.17 and above versions. Compiled by toolchain arm-linux-gnueabi (version 4.9.2)
  armhf: GLIBC 2.17 and above versions; Compiled by toolchain arm-linux-gnueabihf (version 4.9.2)
  arm64: GLIBC 2.17 and above versions; Compiled by toolchain aarch64-linux-gnu (version 4.9.2)
- Android: arm, arm64, x86, x64; Compiled by android-ndk-r18b.
-

4.1.2 SDK content brief introduction

ICM series cameras support a variety of API, including: Native C/C++, NET / C#VB.NET, DirectShow, Twain, LabView and so on. Compared with the other API, as the low-level API, native C/C++ API is characterized by using pure C/C++ development, independent of other runtime libraries, having simple interface and flexible control. This SDK package contains all the resources and information you need to use, as follows:

- inc

nncam.h, C/C++ head file

- win: For Microsoft Windows
  - dotnet

    nncam.cs, for C#. The nncam.cs use P/Invoke to call into nncam.dll. Please copy nncam.cs to your C# project to use it.
    nncam.vb, for VB.NET. The nncam.vb use P/Invoke to call into nncam.dll. Please copy nncam.vb to your VB.NET project to use it.

  - x86

    nncam.lib, lib file for x86
    nncam.dll, dll file for x86
    democpp.exe, x86 C++ demo exe file

- x64

  nncam.lib, lib file for x64
  nncam.dll, dll file for x64
  democpp.exe, x64 C++ demo exe file

- arm

  nncam.lib, lib file for arm
  nncam.dll, dll file for arm

- arm64

  nncam.lib, lib file for arm64
  nncam.dll, dll file for arm64

- winrt

DLL files for WinRT / UWP (Universal Windows Platform) / Windows Store App.
These dll files are compatible with Windows Runtime, and can be consumed from a Universal Windows Platform app.
If use C# to develop the UWP, nncam.cs wrapper class can be used to P/Invoke into nncam.dll.
Please pay attation to:

1. uwp must use winusb, cannot use the proprietary driver. If the proprietary driver has already been installed, please uninstall it in the device manager, after this, Windows will use winusb automatically.
2. DeviceCapability of uwp, see: How to add USB device capabilities to the app manifest. (It seems that microsoft limits the number of Device items to 100)

demouwp.zip is a simple demo, before build and run this demo, please change the value of vidpid in file Package.appxmanifest

- drivers (**The cameras produced after Jan. 1, 2017 support WinUSB, it's no longer need to install driver in Windows8 and above versions**)

1. x86 folder contains the kernel mode drivers for x86, including nncam.cat, nncam.inf and nncam.sys.
2. x64 folder contains the kennel mode driver for x64, including nncam.cat, nncam.inf and nncam.sys.

It is recommended to use DPInst.exe to automatically install the driver. If you use NSIS to make the installation file, you can use a statement similar to the following:

```
    ExecWait '"$INSTDIR\drivers\x64\DPInst.exe" /SA /SW /PATH
"$INSTDIR\drivers\x64"'
```

✧ ***samples:***

1.democpp, take C++ for example. This example shows an enumeration device, an open device, a preview video, a snap image, a set resolution, a trigger and a wide variety of picture formats (bmp, jpg, png etc.) save the image to the file, wmv format video, trigger mode, I/O control, etc. This example uses the Pull Mode mechanism. In order to keep the code clean, the WTL library used by the example can be downloaded from this link http: //sourceforge.net/projects/wtl/

2.demopush, take C++ for example, using the Push Mode mechanism, StartPushModeV3

3.demomfc, A simple C++ example. it uses MFC as the GUI library, supports opening devices, previews video, captures images, sets resolution and saves images to files in a variety of image formats (.bmp, .jpg, .png, etc.). This example uses the Pull Mode mechanism.

4.demowinformcs1, take C# winform for example. It supports to open the device, preview video, capture images, save pictures to files and set white balance. This example uses the PullMode mechanism, called StartPullModeWithWndMsg.

5.demowinformcs2, take C# winform for example. It supports to open the device, preview video, capture images, save pictures to files, set white balance. This example uses the Pull Mode mechanism called StartPullModeWithCallback

6.demowinformcs3, take C# winform for example. It supports to open the device, preview video, capture images, save pictures to files, set white balance. This example uses the Push Mode mechanism called StartPushMode

7.demowinformvb, take VB.NET winform for example. It supports to open the device, preview video, capture images, save pictures to files and set white balance. This example uses the Pull Mode mechanism.

- linux: For Linux

  - udev: 99-nncam.rules, udev rule file. Please see: http://reactivated.net/writing_udev_rules.html
  - c#: nncam.cs, for .Net Core C#. The nncam.cs use P/Invoke to call into libnncam.so. Please copy nncam.cs to your C# project to use it
  - x86: libnncam.so, so file for x86
  - x64: libnncam.so, so file for x64
  - armel: libnncam.so, so file for armel, use toolchain arm-linux-gnueabi
  - armhf: libnncam.so, so file for armhf, use toolchain arm-linux-gnueabihf
  - arm64: libnncam.so, so file for arm64, use toolchain aarch64-linux-gnu

- android: Android platform. libtoupcam.so. for the four architectures of arm, arm64, x86, x64
- mac: macOS platform file
- python: toupcam.py and example code.
- java: toupcam.java and example code(Console and Swing)
- doc: SDK uses documentation, simplified Chinese, English.
- sample:
  - ✧ ***demosimplest, the simplest example is about 60 lines of code.***
  - ✧ ***demoraw, RAW data and static capture, about 120 lines of code.***
- extras:
  - ✧ ***directshow: DirectShow SDK and demo programs.***
  - ✧ ***twain: TWAIN SDK***
  - ✧ ***labview: Labview SDK and demo programs.***
  - ✧ ***MATLAB: MATLAB demo programs.***

4.2 Client democpp description

As shown, “1” is the control menu area and "2" is the video display area.

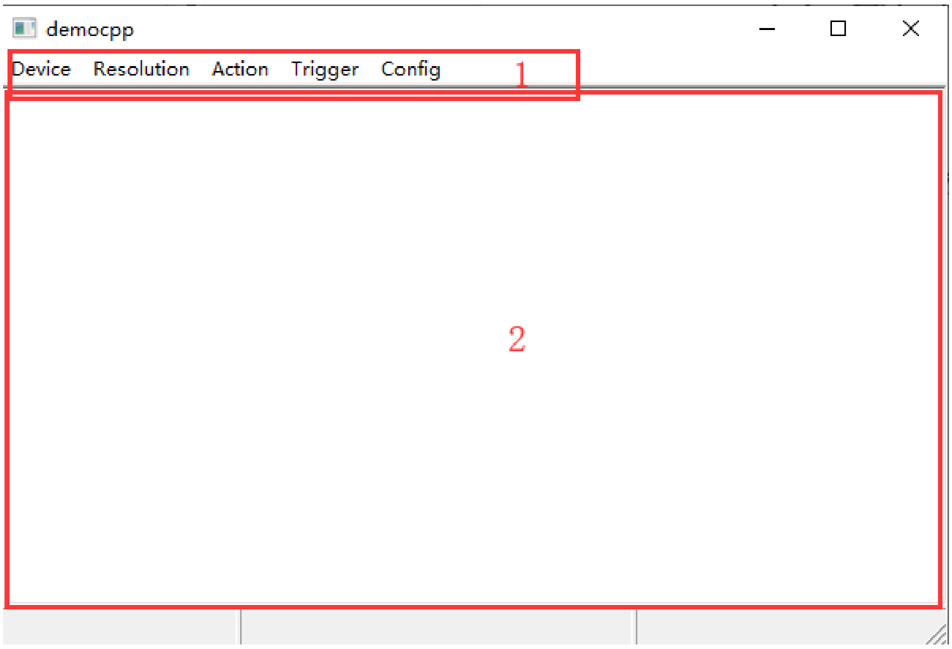


Figure 0-1 democpp interface

The main features of the control menu are:

Device: camera names installed are listed under this menu. Clicking the camera name to open the camera;

Resolution: switch the resolution and capture the image;

Action: Pause, ROI setting, test image, read firmware version number, hardware version number, production date, etc.

Trigger: Defining the trigger mode, the I/O port setting and the like;

Config: Set exposure, gain, white balance, frame rate, etc.

4.3 EHDView UI description

EHDView software fully controls all camera features and streams high-speed videos by USB port using Ultra Fine^{TM} color engine. Ultra Fine^{TM} color engine contains excellent procedure of processing RAW data and thus realizes the conversion of sensor detected data to image. Furthermore, EHDView also provides many advanced video and image processing features, such as image gray level correction, 2D measurement, stitching, depth of field extension, video watermarking, color synthesis, image segmentation and counting and so on. EHDView’s multilingual environment can support any language and currently includes, but is not limited to, English, simplified Chinese, traditional Chinese, German, Japanese, Russian, French, Italian, Polish, Turkish, etc.

Exposure and gain	Automatic exposure, manual exposure, gain up to 5x;
White balance	Automatic white balance; can be adjusted by manually setting the color temperature and color;
Color adjustment	Color, saturation, brightness, contrast, gamma value adjustment feature;
Frame rate control	According to the performance of different computers and USB, the compatibility of the camera can be realized by

	adjusting the frame rate;
Light source frequency control	Natural light / DC, AC50HZ, AC60HZ selection button eliminates video flicker;
Acoustic image	The direction of the adjustable sample can be adjusted by "horizontal" or "vertical" and is consistent with the direction of the visual system;
Sampling and neighborhood averaging and other features	Neighborhood average can improve the signal-to-noise ratio of video stream and sampling extraction mode can ensure the sharpness of video stream. Support video stream histogram expansion, image negative and positive film switching, gray calibration, clarity factor calculation to facilitate video focusing;
Parameter saving	Load, save, overwrite, load, export custom camera panel control;
Video feature	video broadcasting, timing capture, video recording, video watermarking, watermark movement alignment, watermark rotation alignment, video grid overlay, video measurement, video scaling, grayscale scaling calibration, video high dynamic (HDR), video depth of field expansion, video image stitching, video scale, the date and the like are superimposed;
Image processing and enhancement	Image contrast control and adjustment, image de-noising, various image filter algorithms, image mathematical morphology algorithm, image rotation, image scaling and image printing;
2D measurement	Convenient and practical video and image size calibration, a variety of video and image two-dimensional geometric measurements such as length, area, perimeter and angle, etc., the measurement results can be controlled according to image characteristics or preferences;
Image mosaicking	The image stitching can automatically splice the sequence image into a mosaic image. a video window, an image window and a browsing window splicing operation are supported;
EDF (Expansion of depth field)	Depth of field expansion can get ultra-clear images beyond the conventional depth of field by focusing on different layers of images. EHDView supports EDF depth expansion in three windows: video window, image window and browsing window. For different images, EHDView also provides three different depth of field extension algorithms, such as maximum contrast, weighted average and FFDSSD. In addition, the translation, rotation and automatic depth of field expansion between different focus images are considered to ensure the accuracy and rapidity of EDF;
Professional segmentation and counting	The segmentation and counting of EHDView provides six image segmentation methods for users to call according to different image characteristics. The six segmentation methods are watershed, dark OTSU, bright OTSU, RGB histogram, HSV histogram, color segmentation and so on. Users can select any of the six segmentation methods for segmentation, but after selecting any segmentation method, other segmentation methods will be disabled. After the segmentation is completed, there may be the adhesion of the counting object, which can be manually segmented by manual segmentation and the counting results can be counted and analyzed by selecting the counting result menu after confirming the expected results;
Image superposition denoising	EHDView image superposition denoising feature introduces advanced image matching technology, users only need to record a small video of their own image to be superimposed, then they can superimpose and output high fidelity images under the condition of displacement, rotation and magnification of frame burst images, which is simple and easy to use;
Color synthesis	Color synthesis can use black and white fluorescent light source images to create and configure color composite images. Fluorescent probes and colors can be selected directly from predefined data. The dye database of special probe can also be built by the user himself;

Table 0-1 Main features of EHDView

5 Camera Installation and Operation

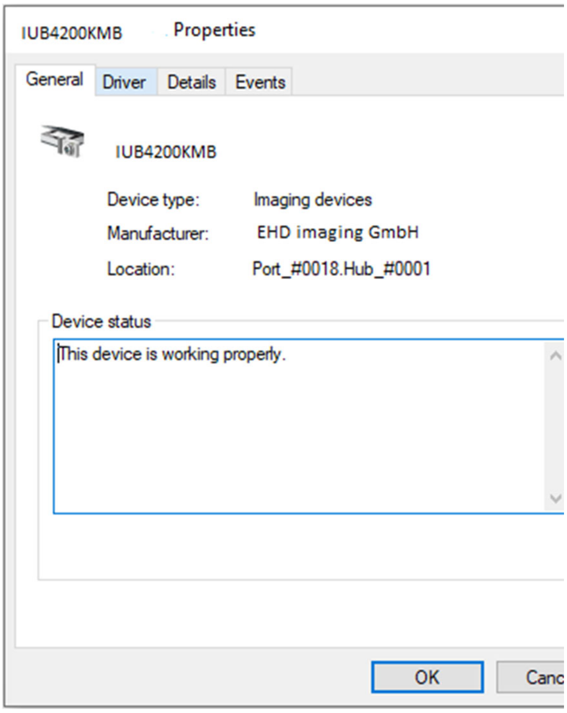
5.1 Installation steps

1. Fix the camera onto the installation position and attach appropriate C-mount lens to the camera.
2. Verify that the camera is properly connected to the industrial computer or PC using the Micro USB3.0 cable that comes with the camera. Tighten the cable by fastening screw at the camera side.

5.2 Driver check

The Operation Systems under Windows 7(Including WIN7) require a normal installation of the drive before the camera is used and if the drive installation fails, the camera will not be found by the client software.

After the installation is complete, in the **Device Manager**, you can see the new device type, such as SCM2020-UV-TR and then right-click the mouse button to see if the device drive is properly installed or not, as shown in 0-1.



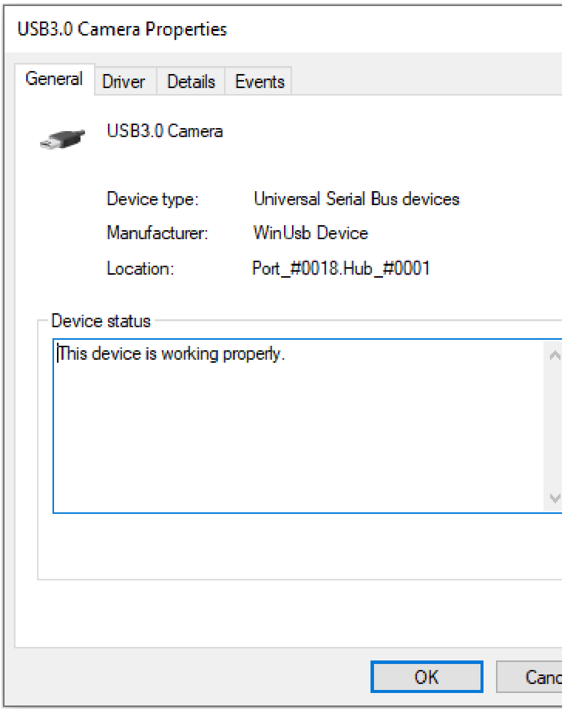
IUB4200KMB

Device type: Imaging devices
Manufacturer: EHD imaging GmbH
Location: Port_#0018.Hub_#0001

Device status

This device is working properly.

Figure 0-1 EHD ICM driver attribute



USB3.0 Camera

Device type: Universal Serial Bus devices
Manufacturer: WinUsb Device
Location: Port_#0018.Hub_#0001

Device status

This device is working properly.

Figure 0-2 Win USB driver attribute

The operating systems above Windows 8(Including Windows 8) will install the driver automatically after the camera is connected and the driver name is USB3.0 Camera, as shown in 0-2.

5.3 Setup and operation

As shown in 0-3, open democpp.exe and click "Device" in the top control menu, where all connected cameras are displayed and click on the corresponding camera name to run the camera.

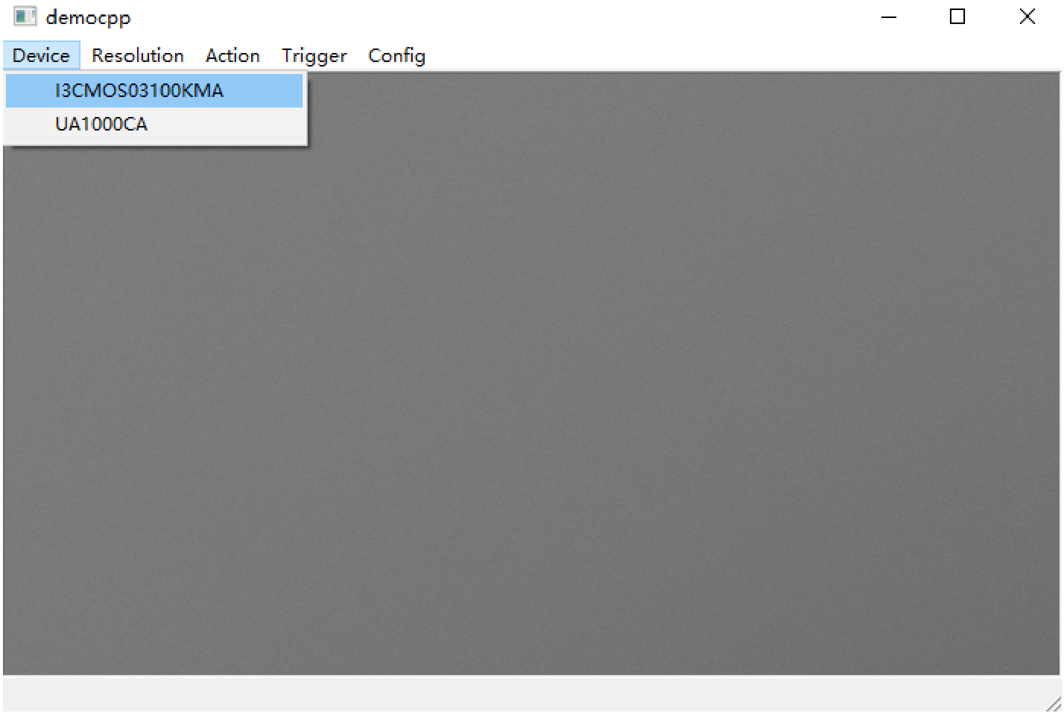


Figure 0-3 democpp UI

1 Main Features of democpp

5.4 Description of main features

As shown in1-1, in the **democpp**, click the "Resolution-> Preview" in the top control menu to select the resolution of the camera; "Resolution->Snap" captures image at current resolution; "Resolution->Snap Multiple" captures multiple images at the specified resolution.

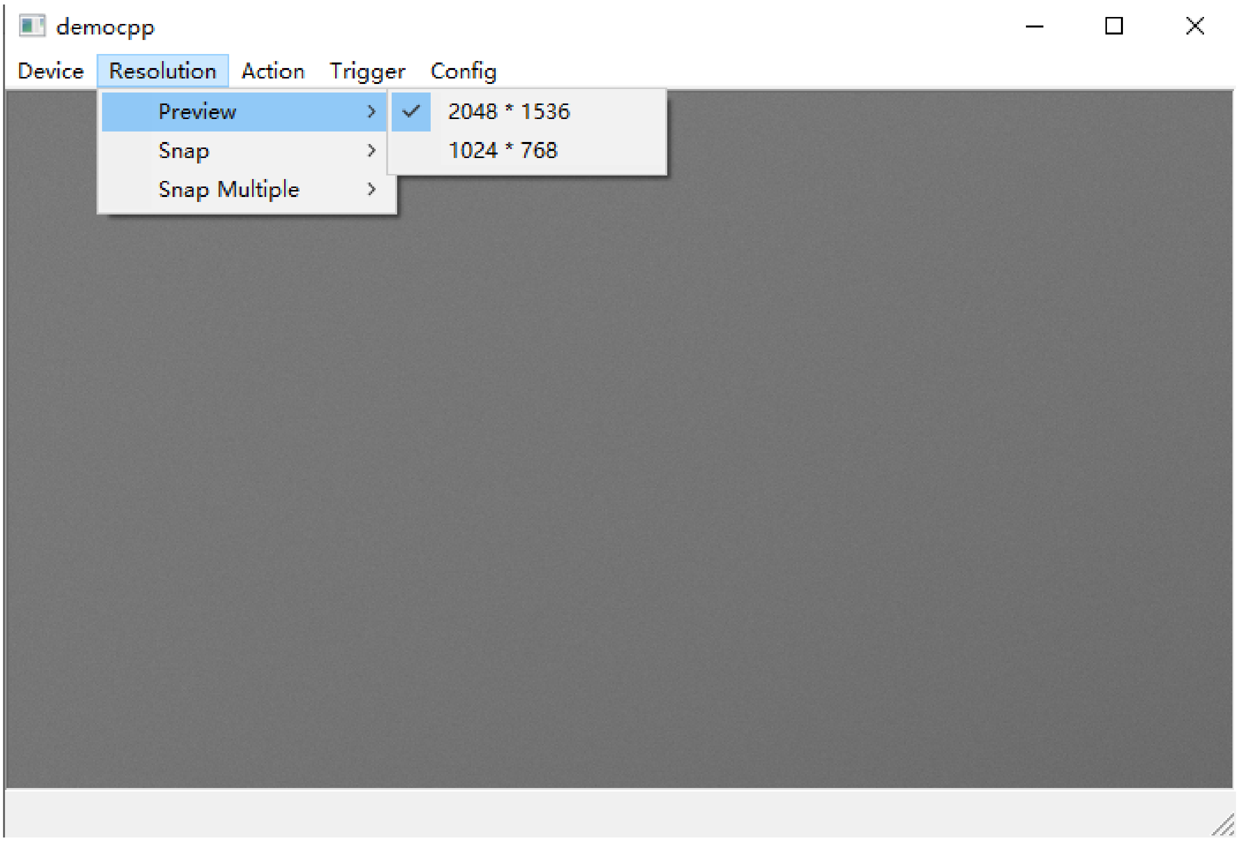


Figure 1-1 Acquisition and resolution

The following is the API code for the image capture operation:

```
//still image snap
nnam_Snap(HnnCam h, unsigned nResolutionIndex);
//multiple still image snap
nncam_SnapN (HnnCam h, unsigned nResolutionIndex, unsigned nNumber);
//The following is the API code that sets the resolution.:
nncam_put_Size(HnnCam h, int nWidth, int nHeight);
nncam_put_eSize(HnnCam h, unsigned nResolutionIndex);
```

5.6 Image format and frame rate

The camera supports a variety of image file formats and the setup of image region of interest. The smaller image ROI will have higher frame rate.

5.6.1 Camera data format

The list of pixel formats supported by the ICM series cameras are shown in. (“Y” for supported; “-” for not supported)

The list of pixel formats supported by the IUX series cameras are shown in 1-1 (“Y” for supported; “-” for not supported)

Format	RAW8	RAW10	RAW12	RAW14	RGB8	RGB24	RGB32	RGB48
ICM433M-TR	Y	-	Y	-	Y	Y	Y	Y
ICM273M-TR	Y	-	Y	-	Y	Y	Y	Y
ICM252M-TR	Y	-	Y	-	Y	Y	Y	Y
ICM265M-TR	Y	-	Y	-	Y	Y	Y	Y
ICM250M-TR	Y	-	Y	-	Y	Y	Y	Y
ICM264M-TR	Y	-	Y	-	Y	Y	Y	Y
ICM178M-TR	Y	---	Y	---	Y	Y	Y	Y
ICM433C-TR	Y	-	Y	-	Y	Y	Y	Y
ICM273C-TR	Y	Y	-	-	Y	Y	Y	Y
ICM252C-TR	Y	-	Y	-	Y	Y	Y	Y
ICM265C-TR	Y	-	Y	-	Y	Y	Y	Y
ICM250C-TR	Y	-	Y	-	Y	Y	Y	Y
ICM264C-TR	Y	-	Y	-	Y	Y	Y	Y
ICM178C-TR	Y	---	Y	---	Y	Y	Y	Y

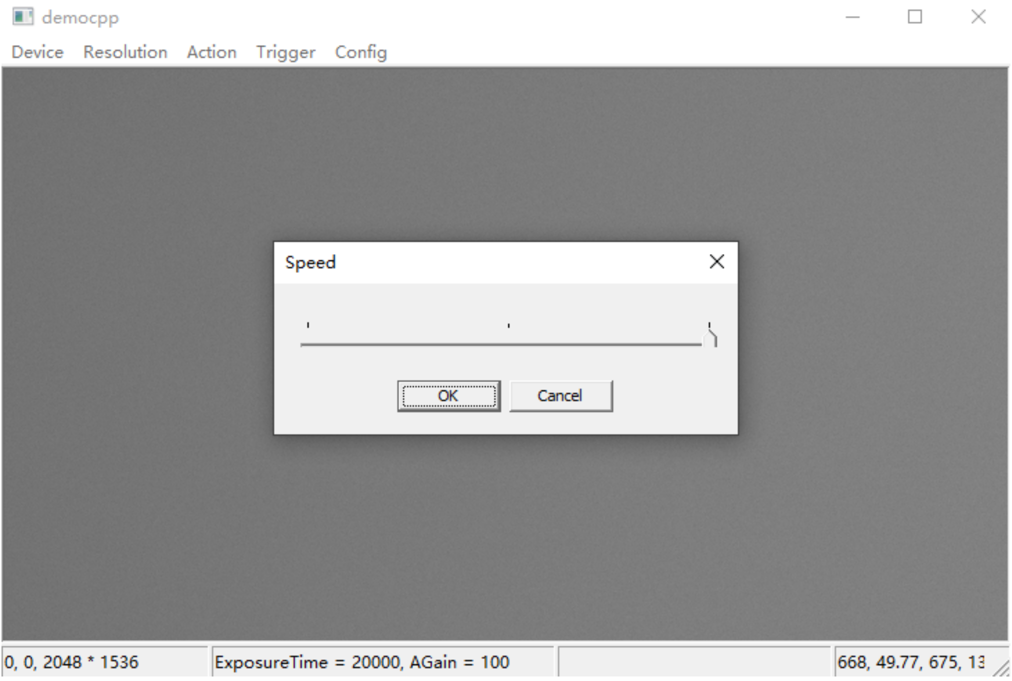
Table 1-1 ICM series camera image data format

5.6.2 Frame rate

The maximum frame rate that the camera can achieve is determined by the following three factors:

- Frame readout time, the smaller the image height, the shorter the time required to read out, the higher the frame rate.
- Exposure time, the shorter the exposure time, the higher the frame rate.
- Bandwidth, the larger the bandwidth, the higher the frame rate that supports transmission.

As shown in 1-2, in **democpp**, click "Config->Speed" in the top control menu and drag the slider bar in "Speed" dialog to set the frame rate.



0, 0, 2048 * 1536

ExposureTime = 20000, AGain = 100

668, 49.77, 675, 13

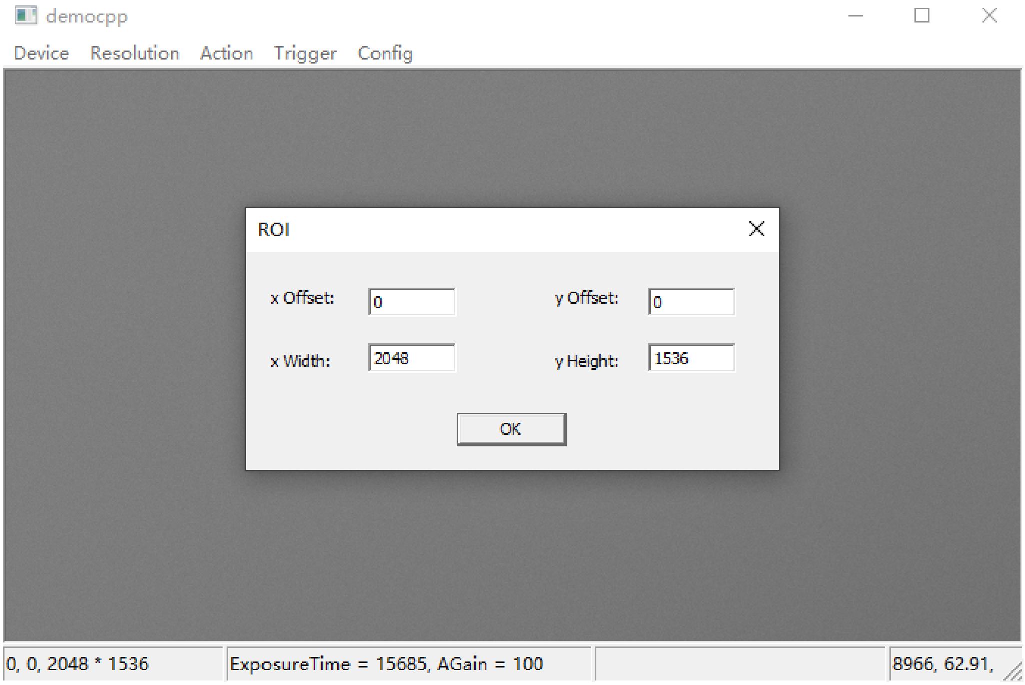
Figure 1-2 Frame rate setup

The following is the API code for the setup of the frame rate:

nncam_put_Speed(HnnCam h, unsigned short nSpeed);

5.6.3 Area of interest setup

When the user is only interested in some details of the image, the camera can output the image ROI according to the requirement. Setup the image ROI can reduce the transmission data bandwidth and improve the camera frame rate to a certain extent.



x Offset: 0

y Offset: 0

x Width: 2048

y Height: 1536

0, 0, 2048 * 1536

ExposureTime = 15685, AGain = 100

8966, 62.91,

Figure 1-3 Area of interest setup

As shown in 1-3, in **democpp**, click "Action->ROI" in the upper control menu and in "ROI" dialog, fill x Offset, y Offset, x Width, y Width to adjust the ROI, where the values in x Offset and y Offset represent the starting point of the ROI up left corner.

The following is the API code for the setup of the image ROI:

nncam_put_Roi(HnnCam h, unsigned xOffset, unsigned yOffset, unsigned xWidth, unsigned yHeight);

6.0 Global Shutter and Rolling Shutter

6.1 Global Shutter

For cameras that support global shutter, exposure starts in each line simultaneously. After the exposure, data is read out line by line, as shown in 1-4.

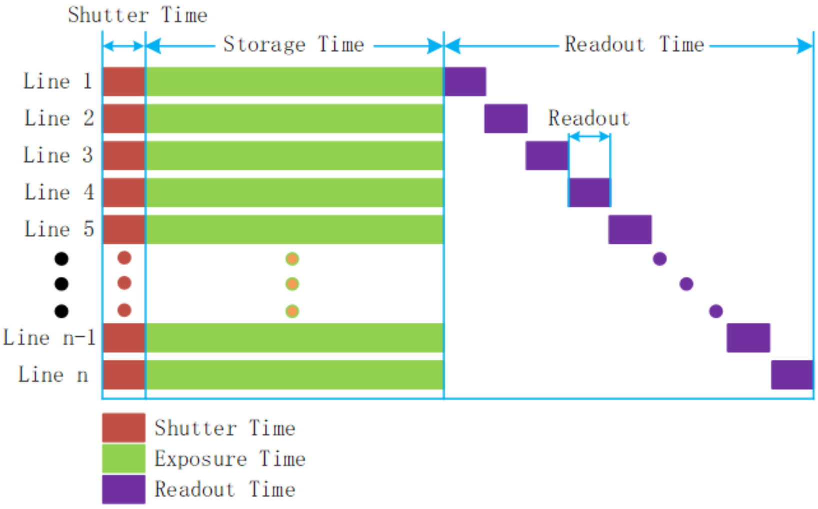


Figure 1-4 global shutter exposure principle

6.2 Rolling Shutter

For cameras that support rolling shutter, after the first line exposure, the next line begins to exposure, repeat in this way. Sensor receive exposure and data read the time length to be consistent, but the time of begin to receive exposure is inconsistent, as shown in 1-5.

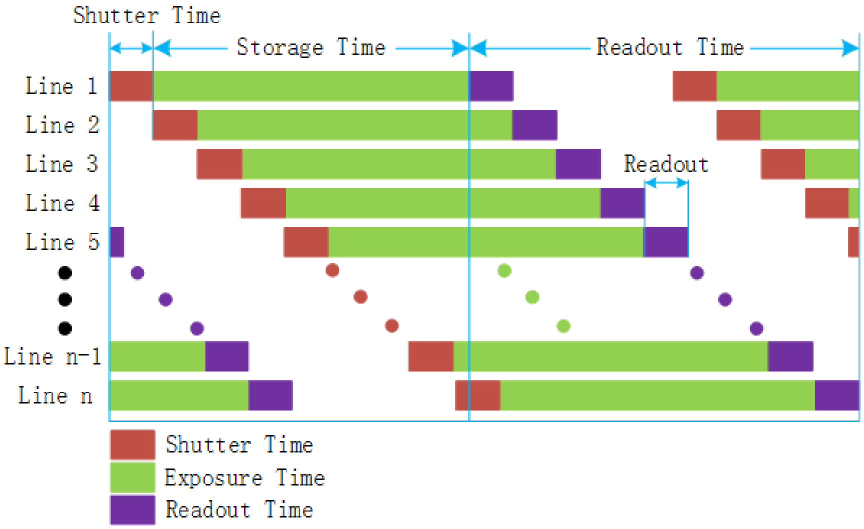


Figure 1-5 Rolling

shutter exposure principle

7.0 Image acquisition and transmission

There are two image acquisition modes, free run mode and trigger mode. Among them, the free run mode is continuous acquisition mode and the trigger mode captures one or more frames of images according to the trigger signal. The trigger sources include software trigger and external trigger.

As shown in 1-6, in democpp, the free run mode and trigger mode are switched by clicking on "Trigger" menu and choosing “Enter Trigger Mode” command. The "✓" checkmark indicates that the current mode is in trigger mode, otherwise free run mode.

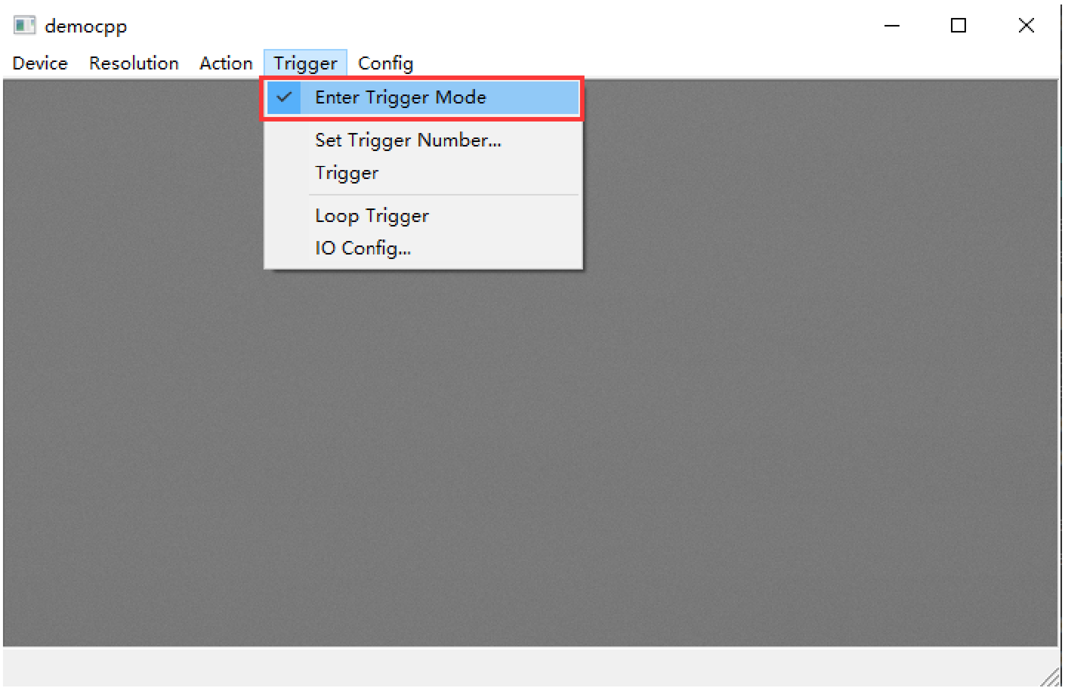


Figure 1-6 Image acquisition mode switching

The following are the API codes for the switching of the free run mode to trigger mode:

```
// 0 = video mode, 1 = software or simulated trigger mode, 2 = external trigger mode
nncam_put_Option(m_hCam, NNCAM_OPTION_TRIGGER, val);
```

7.1.1 Free run mode

Under free run mode, the user can control the camera to continuously output images. Starting the **democpp** software, connecting the camera and clicking run, the camera will run under free run mode by default. The camera continuously outputs the image according to the current setting.

7.1.2 Trigger mode

After the camera enters the trigger mode, it enters the waiting trigger state automatically. After receiving a trigger signal, the camera begins to expose and after the exposure is finished, the image data will be flashed out. Under trigger mode, image acquisition methods have single frame trigger, multi frame trigger, counter trigger and PWM trigger mode.

7.1.3 Trigger signal source selection

Under the trigger mode, trigger signal source is either from software trigger, or from external trigger. The external trigger signal is from either the pin isolated by opto-coupler or the non-isolated pin.

The following is the API code for the setup of the trigger source:

```
// Trigger Source:   0-> line0 , 1-> line2 , 2-> line3 , 3-> Counter , 4-> PWM , 5-> Software
nncam_IoControl(m_hCam, 0, NNCAM_IOCONTROLTYPE_SET_TRIGGERSOURCE, val, NULL);
```

- Software trigger

The camera supports software trigger mode. When a software trigger is executed, the client software will send the command through USB3.0 to activate the camera to acquire and transmit images.

As shown in 1-7, in **democpp**, first click "Enter Trigger Mode" to enter trigger mode. Click "Set trigger Number" to define the number of triggers and finally click "Trigger" and the software will receive the number of triggers. If you click "Loop Trigger", you will enter a continuous trigger mode and clicking it again will exit the current trigger mode.

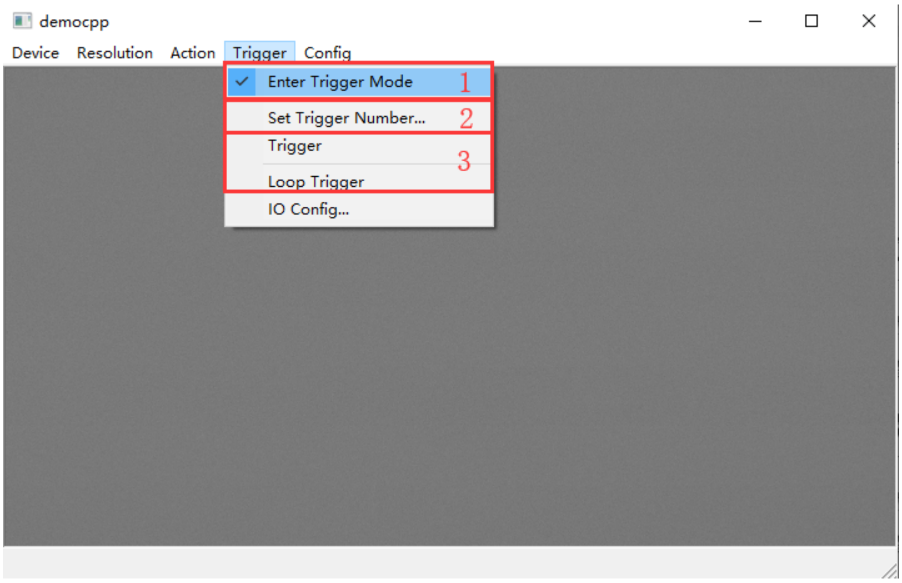


Figure 1-7 Software trigger setup

- External trigger

(For I3 series only)The camera has an opto-isolated input line “Opt_in” on the hardware and a direct GPI input line (the hardware version number V2.0 and above is configurable GPIO port). The user can select either line as the trigger source. Counter mode and PWM mode are also provided.

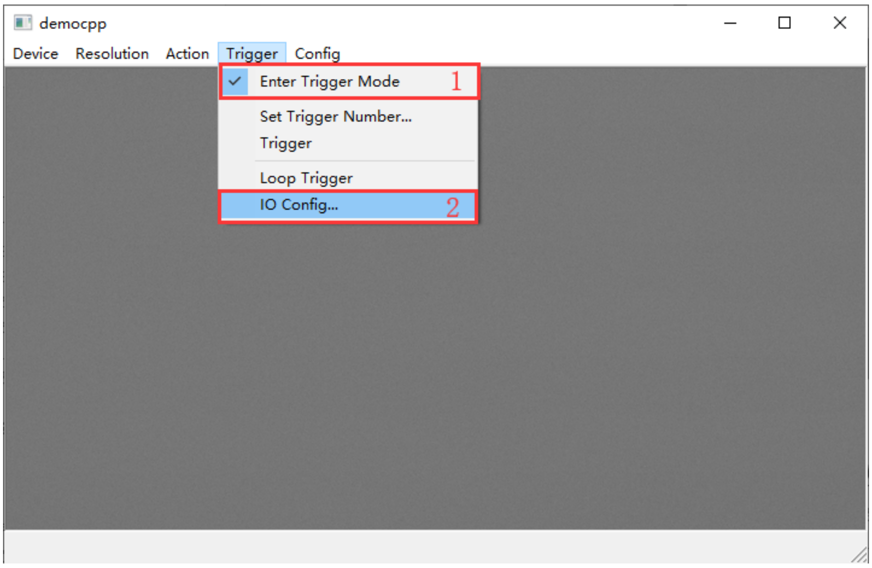
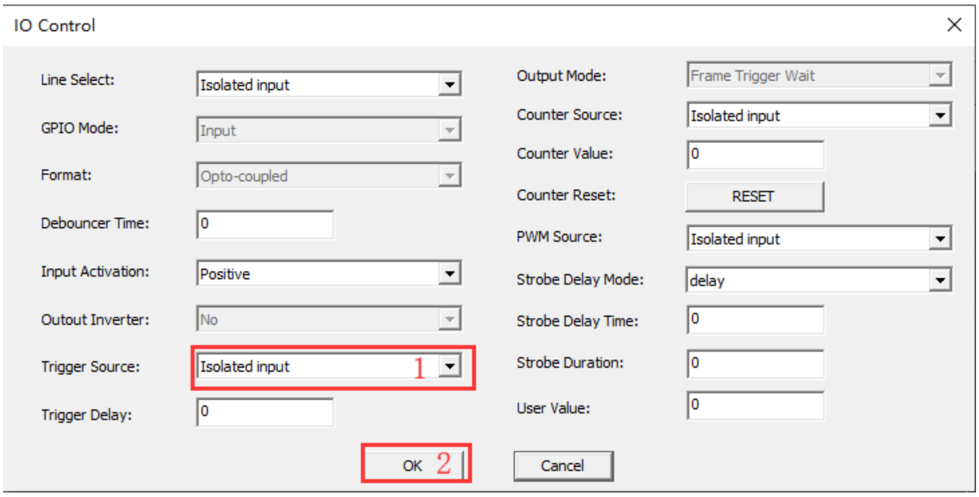


Figure 1-8 Set the external trigger source

1-8 shows the configuration of external trigger. First select "Enter Trigger Mode" to enter trigger mode. Then click "IO Config" and the I/O Control dialog will pop up as shown in 1-9. The trigger source is selected in the "Trigger Source" and then click "OK". At this time a high pulse will trigger the camera on the corresponding line.



IO Control

Line Select: Isolated input

GPIO Mode: Input

Format: Opto-coupled

Debouncer Time: 0

Input Activation: Positive

Outout Inverter: No

Trigger Source: Isolated input

Trigger Delay: 0

Output Mode: Frame Trigger Wait

Counter Source: Isolated input

Counter Value: 0

Counter Reset: RESET

PWM Source: Isolated input

Strobe Delay Mode: delay

Strobe Delay Time: 0

Strobe Duration: 0

User Value: 0

Figure 1-9 I/O control dialog

7.1.4 Frame burst mode

The camera provides a frame burst mode, that is, receiving one trigger signal and producing multiple burst images. The trigger frame value can range from 1 to 1023. “Burst Count = 1” means a one-frame image output, as shown in 1-10, “Burst Count = 3” means a three-frame image output.

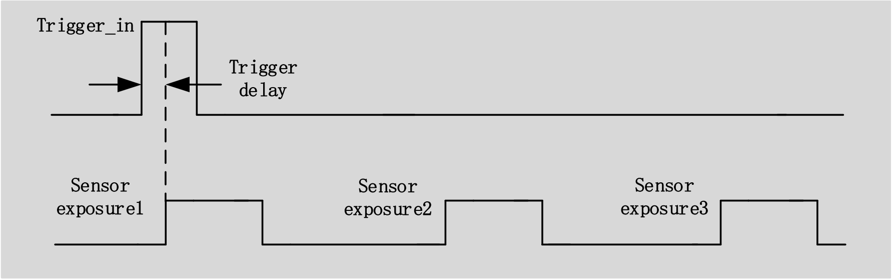


Figure 1-10 Frame burst trigger timing

Here is the API code for the setup of the trigger frame value

nncam_IoControl(m_hCam, 0, NNCAM_IOCONTROLTYPE_SET_BURSTCOUNTER, val, NULL);

7.1.5 Counter trigger mode

Under this mode, trigger signal number is divided by user-defined counter value. For example, when you set the counter to 3, the camera needs to receive three trigger signals before it can begin exposure, as shown in 1-11.

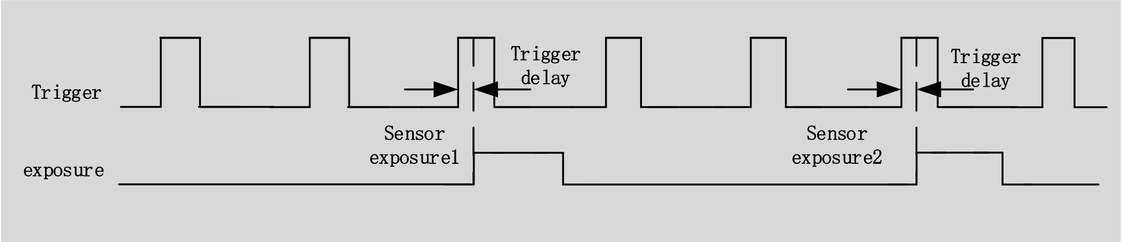
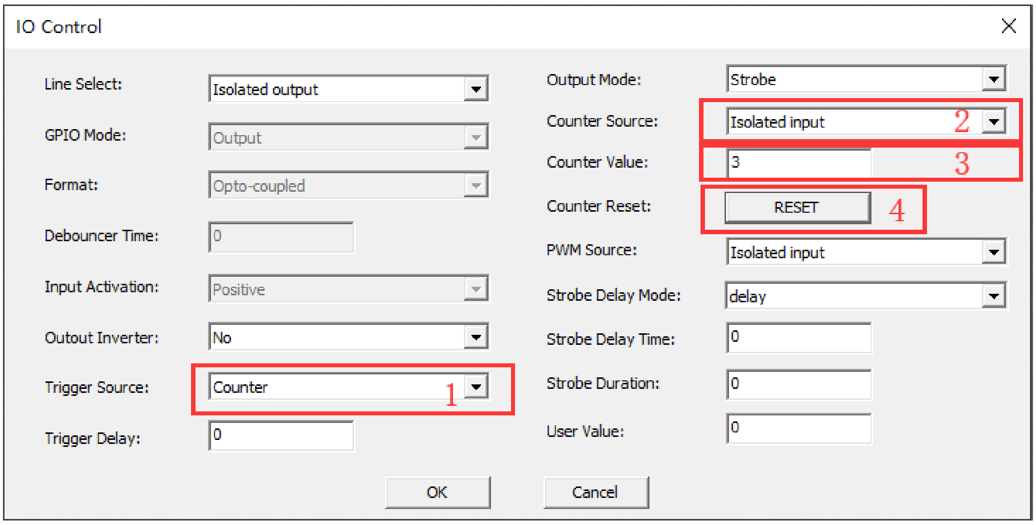


Figure 1-11 Counter trigger mode

The specific operation in **democpp** are shown in 1-12. First, under "Trigger Source", select the trigger source as Counter, then click "Counter Source" to select the external trigger source that needs to be divided and configure the frequency division coefficient in "Counter Value" in the range of 1-1023. "Counter Reset" can clear the current frequency division counter to zero.


IO Control

Line Select: Isolated output

GPIO Mode: Output

Format: Opto-coupled

Debouncer Time: 0

Input Activation: Positive

Outout Inverter: No

Trigger Source: Counter

Trigger Delay: 0

Output Mode: Strobe

Counter Source: Isolated input

Counter Value: 3

Counter Reset: RESET

PWM Source: Isolated input

Strobe Delay Mode: delay

Strobe Delay Time: 0

Strobe Duration: 0

User Value: 0

Figure 1-12 Counter trigger mode setup

The following is the API code for the setup of the counter trigger mode:

nncam_IoControl(m_hCam, 0, NNCAM_IOCONTROLTYPE_SET_TRIGGERSOURCE, 3, NULL);

//Counter Source: 0-> line0 , 1-> line2 , 2-> line3

nncam_IoControl(m_hCam, 0, NNCAM_IOCONTROLTYPE_SET_COUNTERSOURCE, val, NULL);

nncam_IoControl(m_hCam, 0, NNCAM_IOCONTROLTYPE_SET_COUNTERVALUE, val, NULL);

7.1.6 PWM trigger mode

The camera provides Pulse Width Modulation (PWM) trigger mode, which controls exposure time by pulse width. The main difference between this mode and the standard single frame trigger mode is the exposure method. The exposure time per frame is determined by the trigger pulse width, as shown in 1-13.

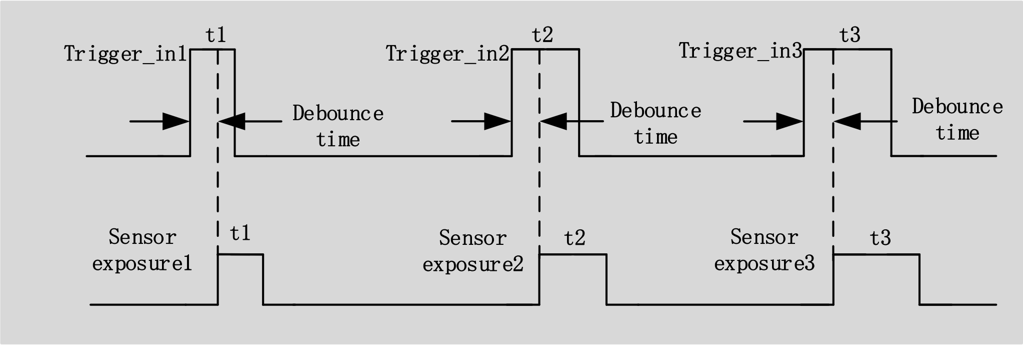


Figure 1-13 PWM mode timing

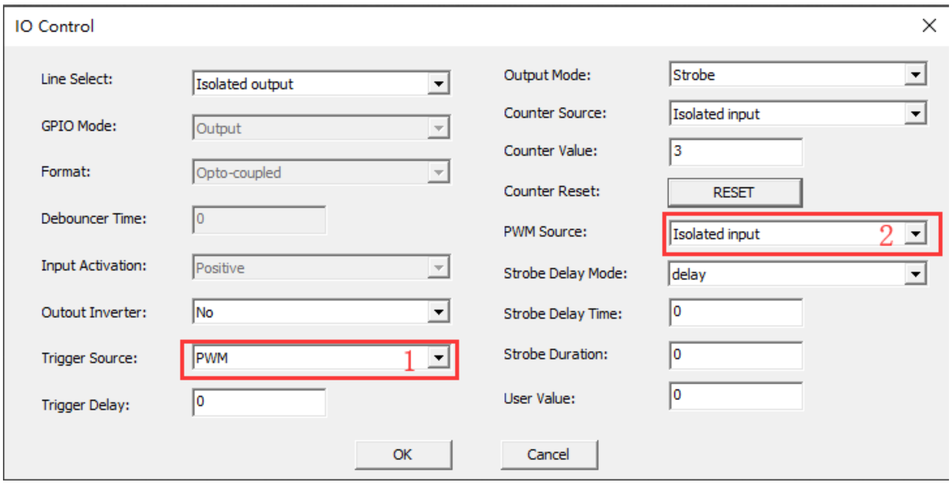
The following cameras with rolling shutter sensors do not support PWM trigger mode：

As shown in 1-14 in **democpp**, select the trigger source as PWM, under "Trigger Source" and click "PWM Source" to select the external trigger source for input.

The following is the API code for the setup of the counter trigger mode:

nncam_IoControl(m_hCam, 0, NNCAM_IOCONTROLTYPE_SET_TRIGGERSOURCE, 4, NULL);

nncam_IoControl(m_hCam, 0, NNCAM_IOCONTROLTYPE_SET_PWMSOURCE, val, NULL);



IO Control

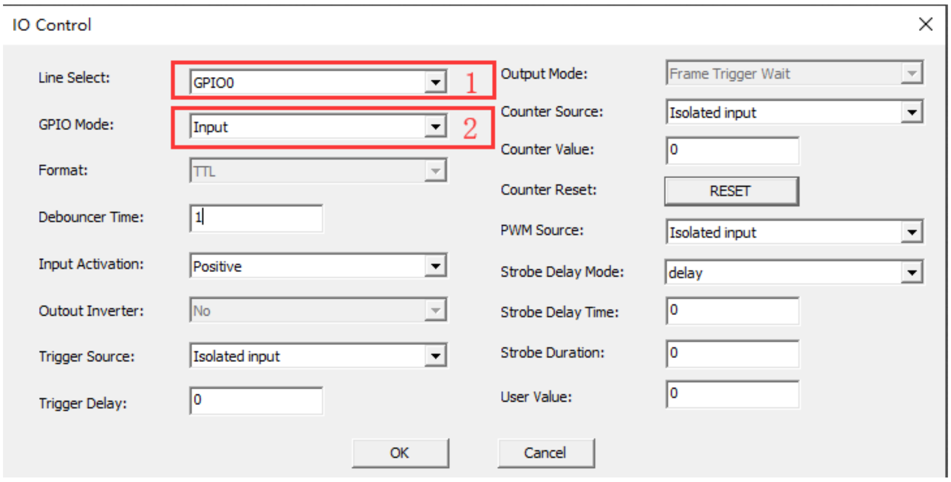
Line Select: Isolated output
GPIO Mode: Output
Format: Opto-coupled
Debouncer Time: 0
Input Activation: Positive
Outout Inverter: No
Trigger Source: PWM
Trigger Delay: 0

Output Mode: Strobe
Counter Source: Isolated input
Counter Value: 3
Counter Reset: RESET
PWM Source: Isolated input
Strobe Delay Mode: delay
Strobe Delay Time: 0
Strobe Duration: 0
User Value: 0

Figure 1-14 PWM mode parameter control

7.2 General Purpose I/O configuration

(This is for I3 series only)The camera with the hardware version number V2.0 and above has a configurable GPIO port, as shown in 1-15, in the demo, enter the "IO Config" dialog, select the "Line Select" to be GPIO0 and then click the "GPIO Mode" to configure the input or output.



IO Control

Line Select: GPIO0
GPIO Mode: Input
Format: TTL
Debouncer Time: 1
Input Activation: Positive
Outout Inverter: No
Trigger Source: Isolated input
Trigger Delay: 0

Output Mode: Frame Trigger Wait
Counter Source: Isolated input
Counter Value: 0
Counter Reset: RESET
PWM Source: Isolated input
Strobe Delay Mode: delay
Strobe Delay Time: 0
Strobe Duration: 0
User Value: 0

Figure 1-15 GPIO configuration

The following is the API code to configure the GPIO input and output direction:

nncam_IoControl(m_hCam, index, NNCAM_IOCONTROLTYPE_SET_GPIODIR, val, NULL);

7.3 Input signal

7.3.1 Signal debouncer

Because the external trigger input signal of the camera may have burr, if it goes directly into the internal logic of the camera, it will cause false trigger. The input trigger signal should be debounced. In addition, the effective pulse width of the trigger signal inputted by the user should be greater than the debouncer time, otherwise the trigger signal will be ignored.

The timing is shown in 1-16. If the effective pulse width of Trigger_in1 is less than the debouncer time, the trigger signal will be ignored.

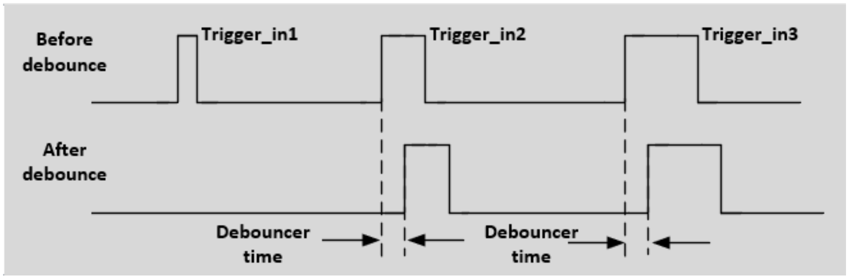
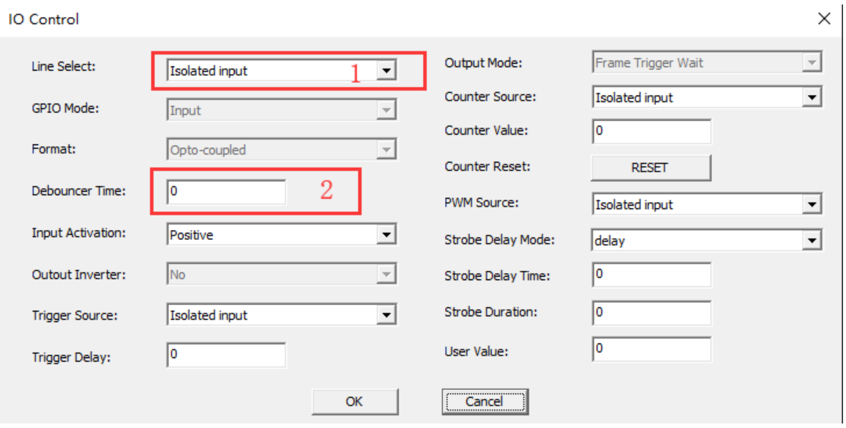


Figure 1-16 Signal debounce timing

As shown in 1-17, in **democpp**, enter the "IO Config" dialog, click "Line Select" to select the input line and then set the debouncer time at "Debounce Time" in the range of 0≤20000 in microseconds.


IO Control

Line Select: Isolated input

GPIO Mode: Input

Format: Opto-coupled

Debouncer Time: 0

Input Activation: Positive

Outout Inverter: No

Trigger Source: Isolated input

Trigger Delay: 0

Output Mode: Frame Trigger Wait

Counter Source: Isolated input

Counter Value: 0

Counter Reset: RESET

PWM Source: Isolated input

Strobe Delay Mode: delay

Strobe Delay Time: 0

Strobe Duration: 0

User Value: 0

Figure 1-17 Signal Debounce setup

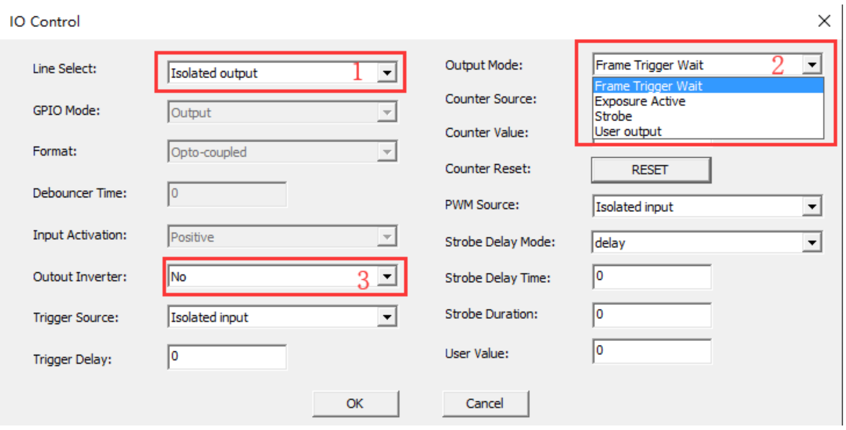
The following is the API code for the setup of the debouncer time:

nncam_IoControl(m_hCam, index, NNCAM_IOCONTROLTYPE_SET_DEBOUNCERTIME, val, NULL);

7.4 Output signal

The camera provides 4 output signal modes: Frame Trigger Wait, Exposure Active, Strobe and User Output.

As shown in 1-18, in the "IO Config" dialog, first select the “Isolated output” in the "Line Select" combobox, then select the output signal mode in the "Output Mode" combobox, click "Output Inverter" to reverse the output signal.



IO Control

Line Select: Isolated output

GPIO Mode: Output

Format: Opto-coupled

Debouncer Time: 0

Input Activation: Positive

Outout Inverter: No

Trigger Source: Isolated input

Trigger Delay: 0

Output Mode: Frame Trigger Wait

Counter Source:

Counter Value:

PWM Source: Isolated input

Strobe Delay Mode: delay

Strobe Delay Time: 0

Strobe Duration: 0

User Value: 0

Figure 1-18 Output signal mode setup

The following is the API code for the setup of the output signal mode:

nncam_IoControl(m_hCam, index, NNCAM_IOCONTROLTYPE_SET_OUTPUTMODE, val, NULL);

// Output Mode: 0-> Frame Trigger Wait , 1-> Exposure Active , 2-> Strobe , 3-> User output

// index: 0-> line0 , 1-> line1, 2-> line2 , 3-> line3

7.4.1 Frame Trigger Wait

The “Frame Trigger Wait” signal is pulled low at the start of the exposure and is pulled high when the last frame of data is read out. The trigger signal inputted by the user should be in the valid period of the signal. If the user inputs a trigger signal when the signal is low, the trigger signal input at this time will be ignored. The following example is the case when Burst Count = 2, as shown in 1-19.

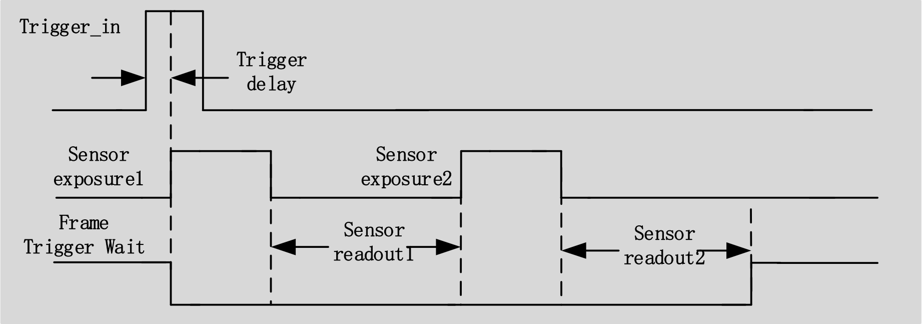


Figure 1-19 Frame Trigger Wait signal timing

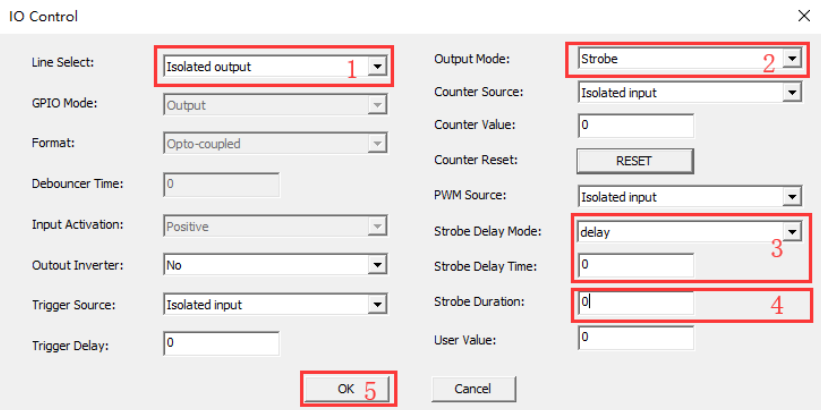
7.4.2 Exposure Active

When “Exposure Active” signal is high, it indicates that the sensor is in the exposure process. This signal can be used as a flash trigger and is also useful when you are operating a system where either the camera or the object being imaged is movable. For example, assume that the camera is mounted on an arm mechanism and that the mechanism can move the camera to view different portions of a product assembly.

7.4.3 Strobe

Strobe can be used to control flash and other external devices. User can set the effective level duration, delay time and pre-delay time.

As shown in 1-20, in the "IO Config" dialog of **democpp**, select the Output Mode as “Strobe”, click "Strobe Delay Mode" to select the “delay” or “pre-delay” and set the time in "Strobe Delay Time". "Strobe Duration" can set the effective level duration of Strobe.



IO Control

Line Select: Isolated output

GPIO Mode: Output

Format: Opto-coupled

Debouncer Time: 0

Input Activation: Positive

Outout Inverter: No

Trigger Source: Isolated input

Trigger Delay: 0

Output Mode: Strobe

Counter Source: Isolated input

Counter Value: 0

Counter Reset: RESET

PWM Source: Isolated input

Strobe Delay Mode: delay

Strobe Delay Time: 0

Strobe Duration: 0

User Value: 0

OK　Cancel

Figure 1-20 Strobe settings

The following is the API code for the setup of the output Strobe signal:

nncam_IoControl(m_hCam, index, NNCAM_IOCONTROLTYPE_SET_OUTPUTMODE, 2, NULL);

nncam_IoControl(m_hCam, 0, NNCAM_IOCONTROLTYPE_SET_STROBEDURATION, val, NULL);

nncam_IoControl(m_hCam, 0, NNCAM_IOCONTROLTYPE_SET_STROBEDELAYMODE, val, NULL);

nncam_IoControl(m_hCam, 0, NNCAM_IOCONTROLTYPE_SET_STROBEDELAYTIME, val, NULL);

- Strobe effective electrical level duration

As shown in 1-21, the Strobe signal is activated at a high level. When the shutter starts to expose, the Strobe signal’s high duration is determined by the "Strobe Duration" value: when the "Strobe Duration" value is 0, the high level duration of the Strobe signal is equal to the exposure time; if the "Strobe Duration" value is not 0, The Strobe signal’s high continuity time is equal to the "Strobe Duration" value.

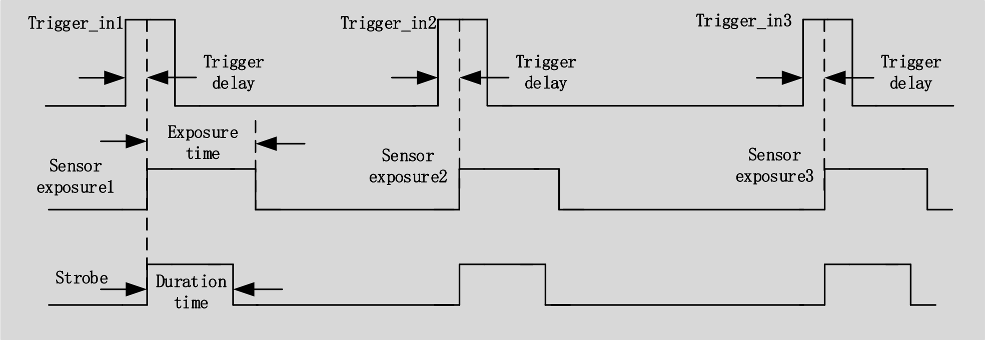


Figure 1-21 Strobe effective level duration

- Strobe output delay

The camera provides the feature of output delay to strobe signal to meet the special usage of users. When the exposure begins, the Strobe signal output does not immediately take effect, delayed according to the value set by "Strobe Delay Time". As shown in 1-22.

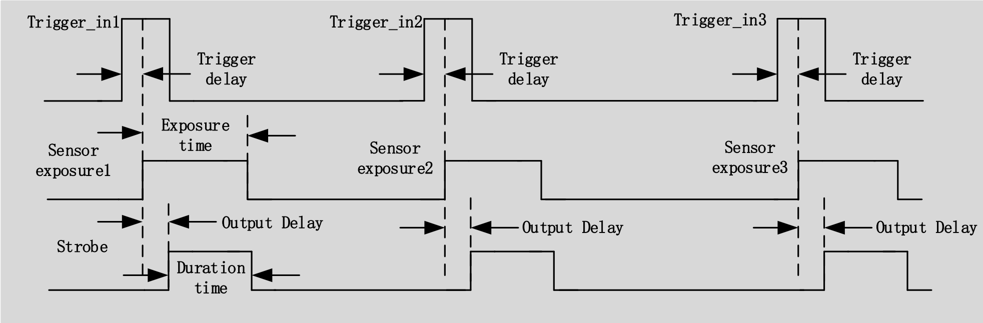


Figure 1-22 Strobe output delay

- Strobe Pre-output

The camera also provides a pre-output feature of the strobe signal, that is, the strobe signal takes effect earlier than the exposure begins.

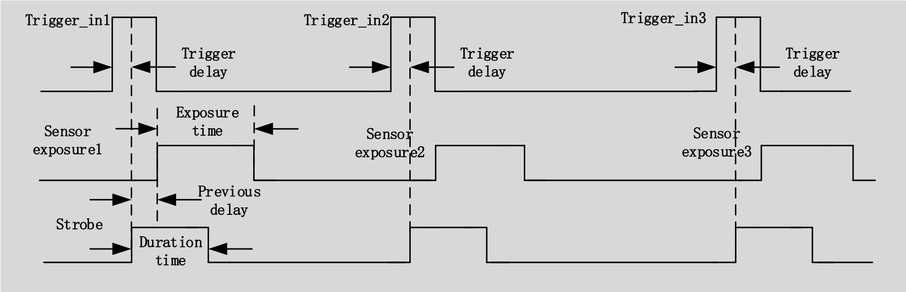
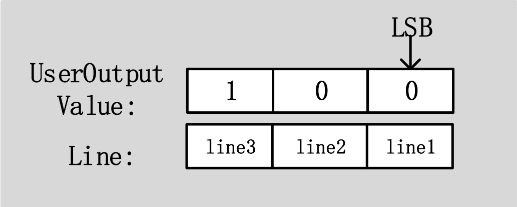


Figure 1-23 Strobe Pre-output

This feature can be applied to flash lamps with slow response. The pre-output time is set by "Strobe Delay Time". The timing is shown in 1-23.

7.4.4 User Output

When choosing the "User Output" output mode, the user can enter a value after the "User Value" control to set the corresponding line output 0 or 1. The value here is only the low three bits of binary, for example when line1, line3 is set to the "User Output" output mode and the "User Value" is set to 4 (undefinedb100), then line3 outputs 1, line1 output 0. as shown in 1-24.



UserOutput Value:	1	0	0
Line:	line3	line2	line1

Figure 1-24 User-defined output schematic diagram

The following is the API code to set User Output:

nncam_IoControl(m_hCam, index, NNCAM_IOCONTROLTYPE_SET_OUTPUTMODE, 3, NULL);

nncam_IoControl(m_hCam, 0, NNCAM_IOCONTROLTYPE_SET_USERVALUE, val, NULL);

7.5 Camera control parameter configuration

7.5.1 Exposure time

The exposure time range is specified in the camera technical specifications section(Sec.). Exposure time control supports manual control and automatic exposure control. When the camera is in trigger mode, the automatic control exposure feature will be disabled.

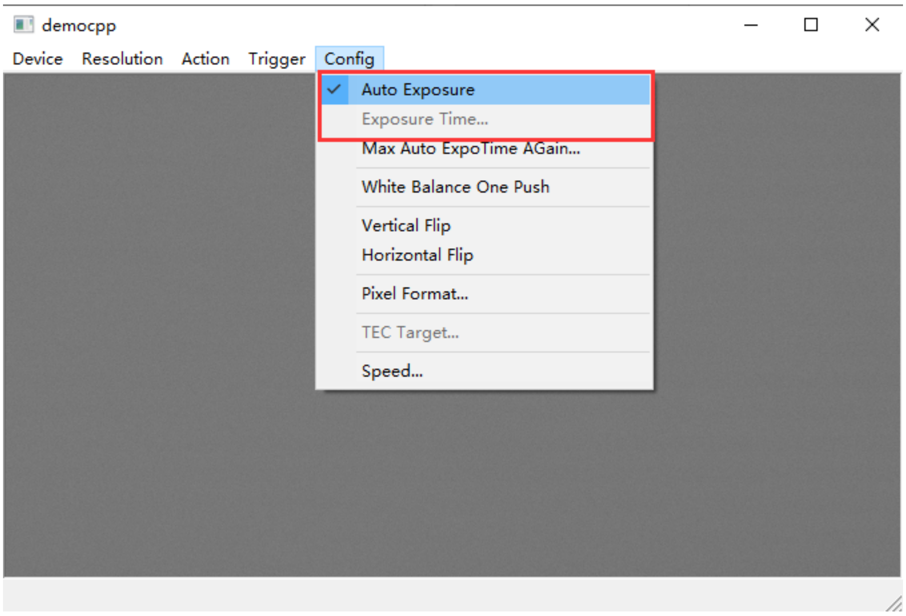


Figure 1-25 Exposure time setup

As shown in 1-25, in **democpp**, click "Config" in the top control menu. Click "Auto Exposure" and the "✓" checkmark indicates that automatic control exposure mode is on. Click again to exit the current mode. Click "Exposure Time" menu and a dialog called Exposure Time will pop up and drag the slider in the new dialog to manually set the exposure time control.

Here is the API code for the setup the exposure time:

nncam_put_AutoExpoEnable(HToupCam h, int bAutoExposure);

nncam_put_AutoExpoTarget(HToupCam h, unsigned short Target);

nncam_put_ExpoTime(HToupCam h, unsigned Time); /* in microseconds */

7.5.2 Gain control

The gain value range is specified in the camera technical specifications section in Sec.1-26. When the gain increases, the image noise increases.

The following is the API code for the setup of the gain control:

nncam_put_ExpoAGain(HToupCam h, unsigned short AGain); /* percent */

7.5.3 White balance

White balance means that the camera performs color adjustment under different light source. The user can make the white area always white at different color temperatures by adjusting the “R”, “B” component’s gain on the image. Ideally, the ratio of R, G and B components in the white region is 1:1:1.

The white balance setting is shown in 1-26. Click "Config" on the control menu at the top of democpp and click "White Balance One Push" to automatically balance white once.

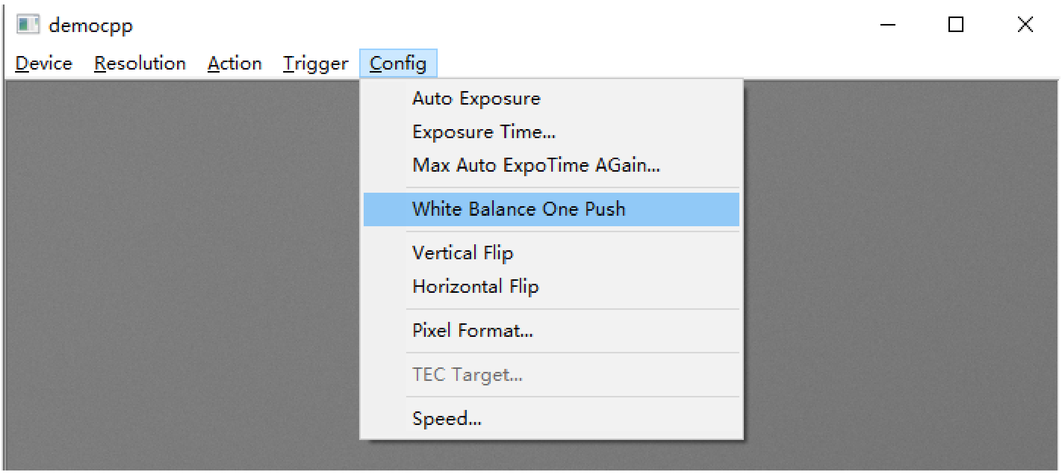


Figure 1-26 White balance setup

The following is the API code for the setup of the white balance one push:

//auto white balance "one push". This feature must be called AFTER nncam_StartXXXX

nncam_AwbOnePush(HToupCam h,PITOUPCAM_TEMPTINT_CALLBACK fnTTProc, void* pTTCtx);

7.5.4 Color adjustment

User can adjust hue, saturation, brightness, contrast and gamma value.

The following is the API code for the color adjustment:

nncam_put_Hue(HToupCam h, int Hue);

nncam_put_Saturation(HToupCam h, int Saturation);

nncam_put_Brightness(HToupCam h, int Brightness);

nncam_put_Contrast(HToupCam h, int Contrast);

nncam_put_Gamma(HToupCam h, int Gamma);   /* percent */

7.5.6 Image flip

As shown in 1-27, in **democpp**, click "Config" in the control menu. Click "Vertical Flip" to flip the image vertically and "Horizontal Flip" to flip horizontally.

The following is the API code to flip the image:

nncam_put_VFlip(HToupCam h, int bVFlip); /* vertical flip */

nncam_put_HFlip(HToupCam h, int bHFlip); /* horizontal flip */

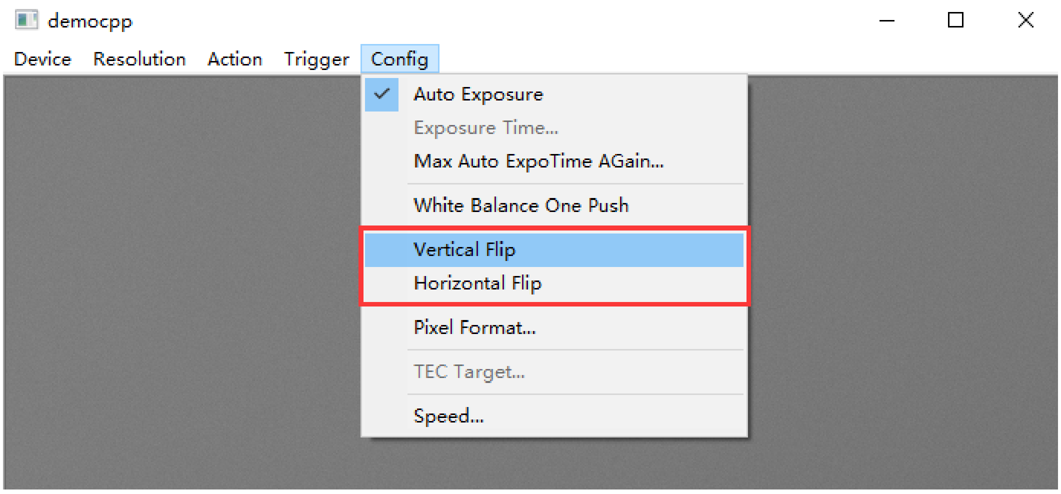


Figure 1-27 Image flip setup

7.5.7 Test pattern

In **democpp**, as shown in 1-28, clicking "Action" in the control menu, then clicking “Test Pattern”. selecting "Normal" to display unaltered sensor captured image, "Test Pattern 1" a gray scale gradient oblique stripe showing the movement, "Test Pattern 2" a gray scale gradient vertical stripe showing the movement and "Test Pattern 3" a grayscale gradient horizontal stripe that shows the movement. The color camera can output a corresponding test pattern.

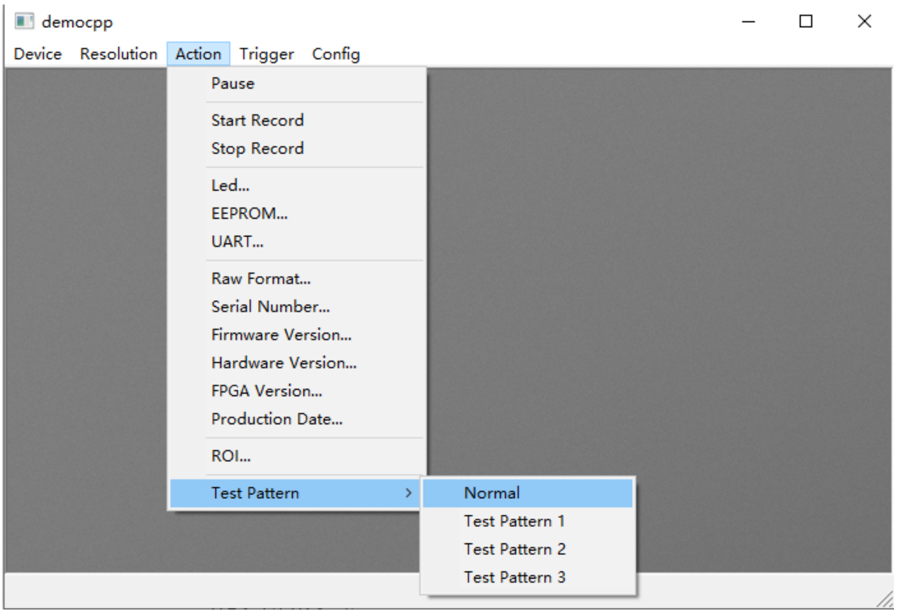
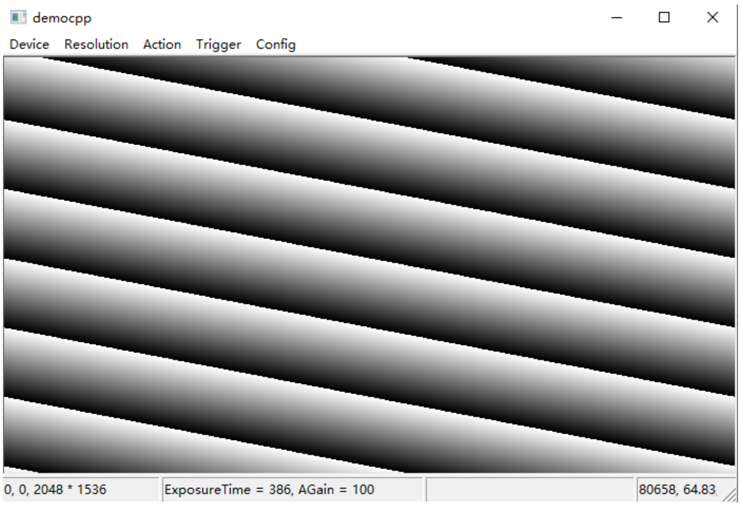
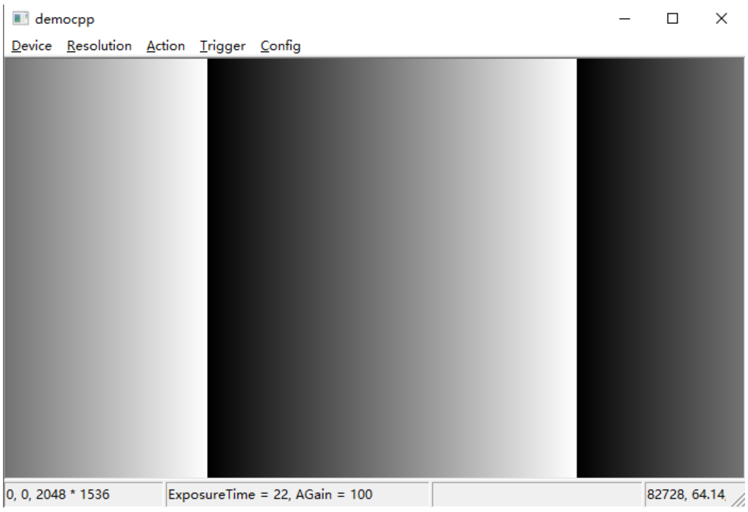


Figure 1-28 Set the test pattern



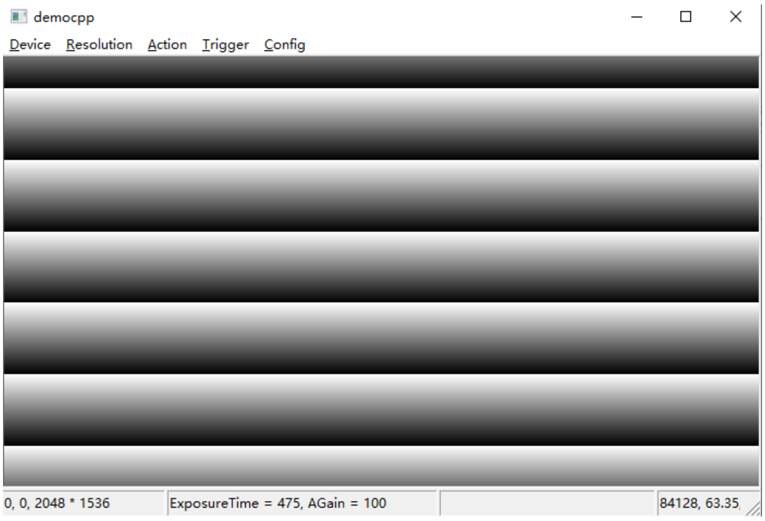
0, 0, 2048 * 1536 | ExposureTime = 386, AGain = 100 | 80658, 64.83

Figure 1-29 Grayscale gradient oblique stripes



0, 0, 2048 * 1536 | ExposureTime = 22, AGain = 100 | 82728, 64.14

Figure 1-30 Grayscale gradient vertical stripes



0, 0, 2048 * 1536 | ExposureTime = 475, AGain = 100 | 84128, 63.35

Figure 1-31 Grayscale gradient horizontal stripes

The following is the API code for the setup of the test pattern:

// TestPattern: 0-> TestPattern Off , 3-> Moving Diagonal Gray Gradient , 5-> Moving Vertical Gray Gradient , 7-> Moving Horizontal Gray Gradient , 9-> Moving Diagonal Chromatic Gradientnncam_put_Option(m_hCam, NNCAM_OPTION_TESTPATTERN, val);

8.0 ICM series camera's I/ O electrical properties

8.1.1 ICM series camera's opto-isolated input circuit (line0)

In the camera I/O control, opto-isolated input circuit is shown in Figure 0-1.

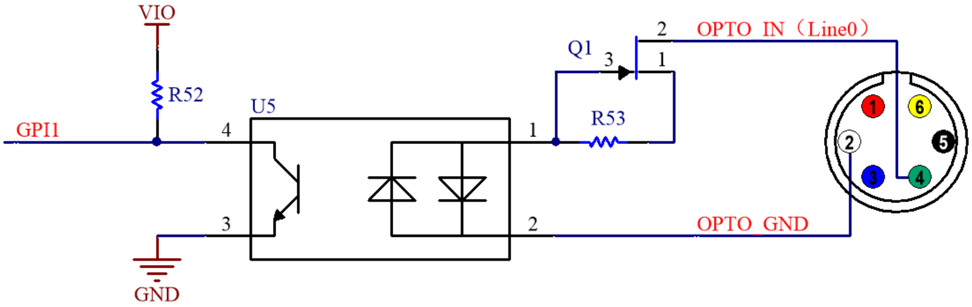


Figure 0-1 Opto-isolated input circuit

Logic 0 input level: 0~1.4VDC (OPTO_IN pin)

Logic 1 input level: 2.2~24VDC (OPTO_IN pin)

Maximum input current: 30mA

The input level is between 1.4V and 2.2V, the circuit action state is uncertain, please avoid the input voltage working in this range.

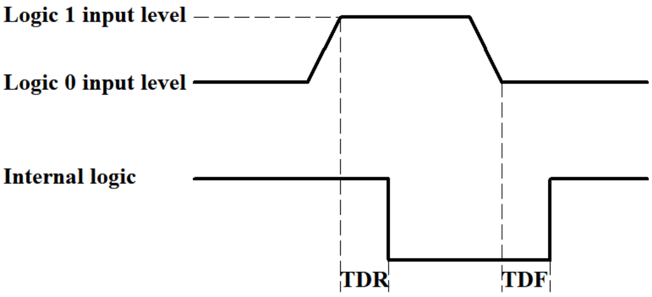


Figure 0-2 Input logic level

Input rise delay (TDR): 5us

Input drop delay (TDF): 25us

8.1.2 ICM series camera's opto-isolated output circuit (line1)

In camera I/O control, opto-isolated output circuit is shown in Figure 0-3.

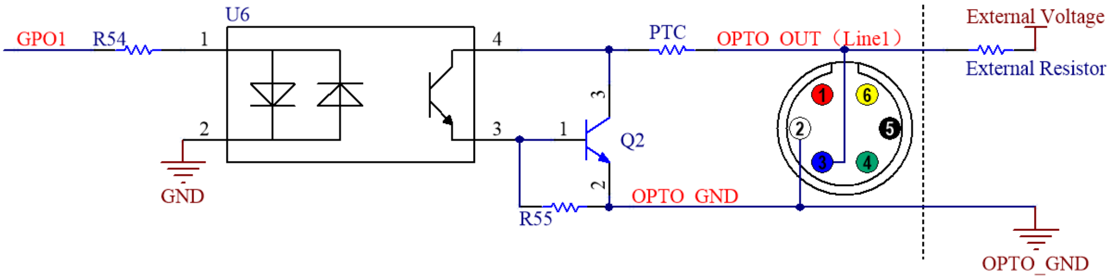


Figure 0-3 Opto-isolated output circuit

Opto-isolated output maximum current: 30mA

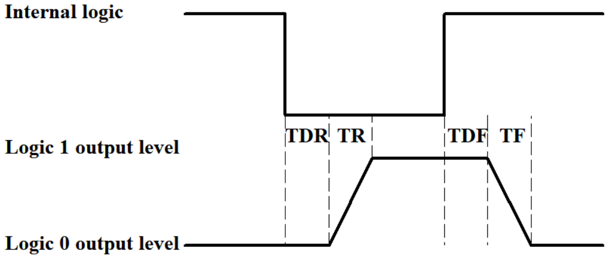


Figure 0-4 Output logic level

The electrical characteristics of the opto-isolated output signal (external voltage 5V, external resistor 1K) are shown in Table 0-1.

Parameter name	Parameter symbol	Parameter values
Output logic low level	VL	760mV
Output logic high	VH	5V
output rise time	TR	8.6us
Output downtime	TF	2.2us
Output rising delay	TDR	17.5us
Output drop delay	TDF	4.2us

Table 0-1 Opto-isolated output signal's electrical characteristics

The corresponding current and output logic low level parameters are shown in Table 0-2 when different voltage and resistors are used in external circuit.

External voltage	Non-essential resistance	VL	Output current
3.3V	1KΩ	668mV	2.82mA
5V	1KΩ	760mV	4.31mA
12V	2.4KΩ	798mV	4.68mA
24V	4.7KΩ	833mV	4.97mA

Table 0-2 Opto-isolated output logic's low level parameters

8.1.3 ICM series camera's input and output I/O circuit (line2/line3)

In camera I/O control with hardware version number V1.0, non-isolated input, output I/O circuit is shown in Figure 0-5.

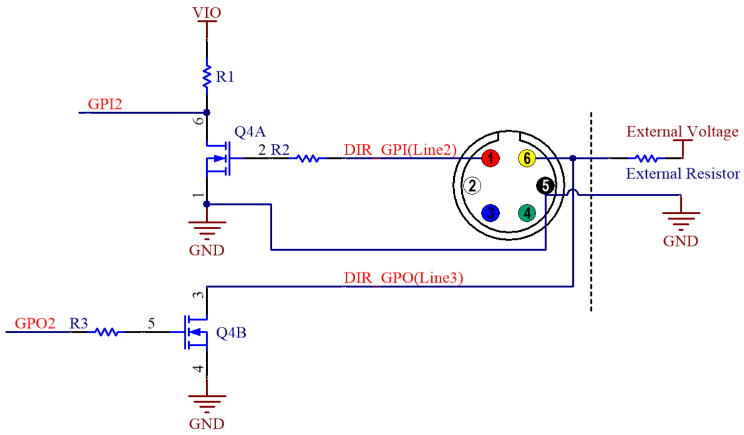


Figure 0-5 Non-isolated input, output I/ O circuit

1, GPI2 input level parameter:

Logic 0 input level: 0-0.9 VDC (DIR _ GPI pin)

Logic 1 input level: 1~20VDC (DIR_GPI pin)

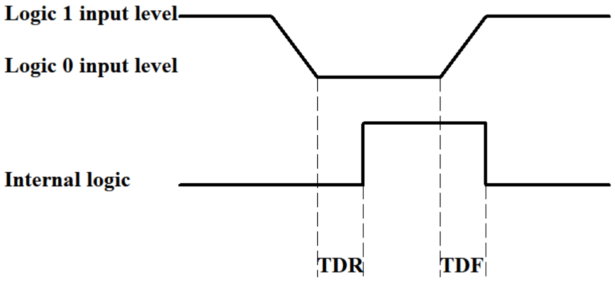


Figure 0-6 Input logic level

To prevent damage to the GPI pin, connect the GND pin before entering voltage to the DIR_GPI pin.

2, GPO2 output level parameter:

The maximum current allowed through this pin is 25 mA.

When the ambient temperature is 25 degrees Celsius, the relationships between the external voltage, resistance and output low level are shown in Table 0-3.

External voltage	Non-essential resistance	VL(GPO2)
3.3V	1KΩ	0V
5V	1KΩ	0V
12V	2.4KΩ	0V
24V	4.7KΩ	0V

Table 0-3 Non-isolated output logic's low level parameters

The external pull-up voltage 5V pull-up resistance 1K Ω, GPO2 output logic level, electrical characteristics are shown in Figure 0-7 and Table 0-4 respectively.

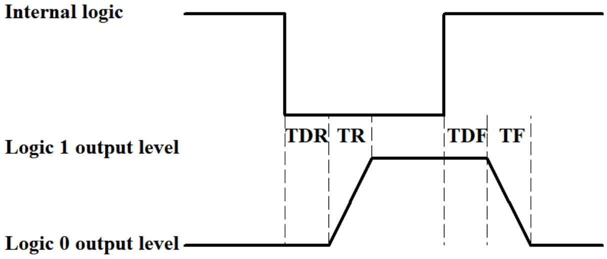


Figure 0-7 Output logic level

Parameter name	Parameter symbol	Parameter values
Output rise time	TR	0.01us
Output downtime	TF	0.01us
Output rising delay	TDR	0.02us
Output drop delay	TDF	0.04us

Table 0-4 Non-isolated output's electrical characteristics

8.1.4 ICM series camera's configurable input and output I/O circuits (line2)

Camera with hardware version V2.0 and above, its input and output I/O circuits are shown in Figure 0-8.

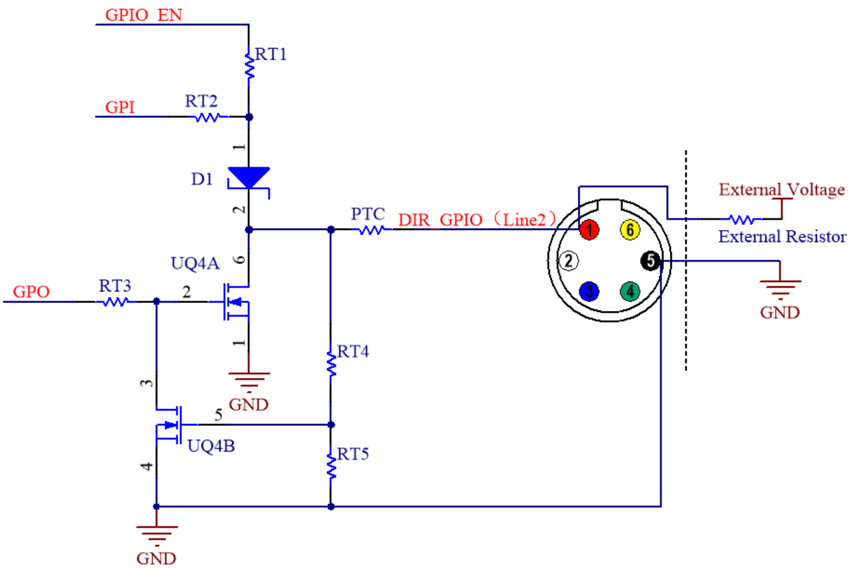


Figure 0-8 Non-isolated configurable input / output I/O circuits

- Line2 is set as input pin

Logic 0 input level: 0~0.6VDC (DIR_GPIO pin)

Logic 1 input level: 2~24VDC (DIR_GPIO pin)

Maximum input current: 25mA

When the input level is between 0.6 V and 2 V, the circuit action is uncertain. Please avoid the input voltage working in this range.

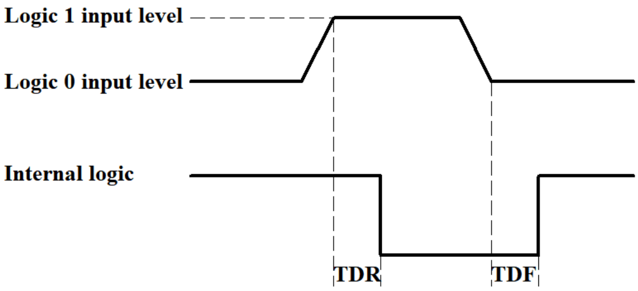


Figure 0-9 Input logic level

To prevent the GPIO pin from being damaged, first connect the ground pin GND and then input the voltage to the Line2 pin

Input rise delay (TDR): 0.02us

Input drop delay (TDF): 0.02us

- Line2 is set as output pin

The maximum current allowed through this pin is 25 mA.

When the ambient temperature is 25 degrees Celsius, the relationships between external voltage, resistance and output low level are shown in Table 0-5

External voltage	Non-essential resistance	VL(GPIO)
3.3V	1KΩ	0.11V
5V	1KΩ	0.167V
12V	2.4KΩ	0.184V
24V	4.7KΩ	0.385V

Table 0-5 Non-isolated output logic low level parameters

The external pull-up voltage is 5V, the pull-up resistance is 1K Ω, the GPIO is configured as the output logic level and the electrical characteristics are shown in Figure 0-10.

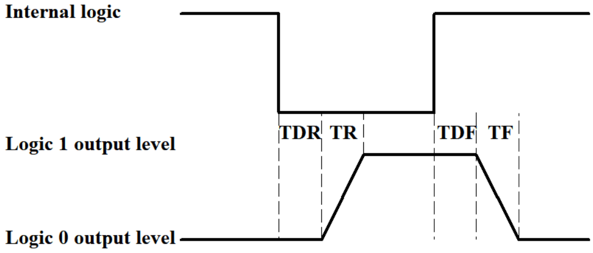


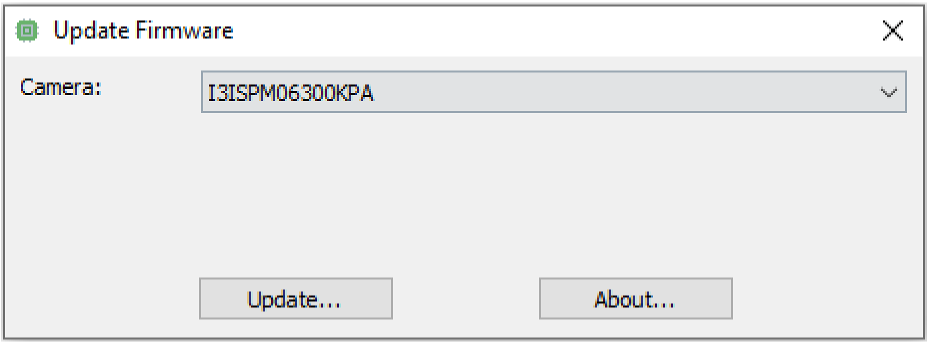
Figure 0-10 Output logic level

Parameter name	Parameter symbol	Parameter values
Output rise time	TR	0.08us
Output downtime	TF	0.02us
Output rising delay	TDR	0.1us
Output drop delay	TDF	0.04us

Table 0-6 Non-isolated output's electrical characteristics

9.0 Firmware upgrade

The I3 series industrial camera can use the updatefw.exe tool available in the SDK for firmware field upgrades. For the latest version of the firmware xxxxx.ufw file, please contact technical support.



Camera: I3ISPM06300KPA

Figure 0-11 Firmware upgrade tool

10.0 Frequently asked questions

1. How long does the camera start exposure after triggering?

After receiving the trigger signal, the camera needs to determine the following three parameters: "Trigger Delay", "Debouncer Time" and "Strobe Delay Time". As shown in Figure 0-1, the trigger signal is first subjected to a debounce delay and then a trigger delay and if the Strobe is set to a pre-output, a pre-output delay is required.

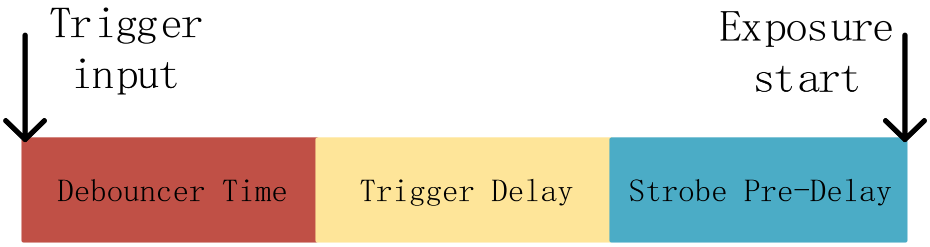


Figure 0-1 Exposure start time

11.0 Support

You can also get support in the following ways:

- Web site support: https: //www.ehd.de for relevant documentation and on-line technical support.
- E-mail support: info@ehd.de
- Other support: please contact the sales representative.

EHD imaging GmbH
Zum Rennplatz 15
D-49401 Damme (Germany)
Tel: +49-5491-2090
email: info@ehdimaging.de
www.ehd.de