GigE Series Products

SDK Instructions

Directory

Introduction 1

Data structure and enumerated types 2

tagGigECameraInfo Structure 2

VTGE_CAMERA_TYPE Enum 3

VTGE_FLIP_MODE Enum 5

VTGE_ZOOM_MODE Enum 6

VTGE_CAP_MODE Enum 7

VTGE_SPEED_MODE Enum 8

VTGE_IP_CFG Enum 9

VTGE_IMG_TYPE Enum 10

VTGE_COLOR_MODE Enum 11

Functions Overview 12

Function description 17

VTGE_Initialize 17

To establish a network service connection 17

VTGE_UnInitialize 17

Disconnect the network service connection. 17

VTGE_GetCameraCount 18

VTGE_GetCameraType 18

VTGE_GetCameraInfo 19

VTGE_InitCamera 20

VTGE_UnInitCamera 20

VTGE_LiveStart 21

VTGE_LiveStop 21

VTGE_ResetCameraPort 22

VTGE_GetCameraPort 23

VTGE_SetExposure 24

VTGE_GetExposure 25

VTGE_SetGainR 26

VTGE_SetGainG 27
VTGE_SetGainB 28
VTGE_SetGain_A 29
VTGE_SetResolution 30
VTGE_GetNICFrameRate 31
VTGE_SetVideoWindowPos 31
VTGE_SetFlipMode 33
VTGE_SetZoomMode 34
VTGE_SetCapMode 35
VTGE_SetSpeedMode 36
VTGE_SetOffsetH 37
VTGE_SetOffsetV 38
VTGE_SetFrameCallBack 39
VTGE_GetFrame 40
VTGE_GetFrameToImage 41
VTGE_SetCameraIP 42
VTGE_IPConfig 43
VTGE_SetContrast 43
VTGE_SetTriggerState 44
VTGE_FrameToImage 45
VTGE_SetFlashState 46
VTGE_SetDoAE 47
VTGE_SetDoAWB 48
VTGE_SetColorMode 49
VTGE_SetDhcpEnable 49
VTGE_CaptureStart 50
VTGE_CaptureStop 50
VTGE_SelectAviCompressor 51
VTGE_SetAviCompressor 52
VTGE_SetTriggerMode 52
VTGE_GetTriggerMode 53

VTGE_SetPackageDelayTime 53
VTGE_GetPackageDelayTime 53
VTGE_SetWatchdogState 54
VTGE_GetWatchdogState 55
VTGE_SetIO 56
VTGE_GetIO 57
VTGE_SetTriggerDelayTime 58
VTGE_SoftTrigger 58
VTGE_HDR_Enable 59
VTGE_MCResolutionShow 60
VTGE_IsMCResolutionShow 60
VTGE_SetMCResolution 61
VTGE_GetMCResolution 62
VTGE_ValidBitsOffSet 63
VTGE_AEParam 64
VTGE_SetNewHwnd 64
VTGE_ SetPackageSize 65
VTGE_ GetPackageSize 65
The use of a callback function 66
AWB/AE callback function 66
Video Preview 69
AE 71
AWB 72
Appendix 73

Introduction

Support all of micro vision GigE series products.

Data structure and enumerated types

tagGigECameraInfo Structure

```
typedef struct _tagGigECameraInfo
{
    USHORT version_major;
    USHORT version_minor;
    ULONG    device_mode;
    USHORT reserved0;
    UCHAR    MAC_ADDR[MAC_ADDR_LEN];
    ULONG    IP_CONFIG;
    ULONG    IP_CONFIG_CURRENT;
    ULONG    IP_ADDR_RESERVED[3];
    ULONG    IP_ADDR;
    ULONG    MASK_RESERVED[3];
    ULONG    IP_MASK;
    ULONG    GATEWAY_RESERVED[3];
    ULONG    GATEWAY;
    UCHAR    MANUFACTURER_NAME[32];
    UCHAR    MODEL_NAME[32];
    UCHAR    DEVICT_VERSION[32];
    UCHAR    MANUFACTURER_INFO[48];
    UCHAR    SERIAL_NUMBER[16];
    UCHAR    USER_DEFINE[16];
}GigECameraInfo,*PGigECameraInfo;
```

VTGE_CAMERA_TYPE Enum

```
enum VTGE_CAMERA_TYPE{
    VTGE_CAMERA_UNKNOW          = 0x00000000,
    VTGE_CAMERA_CMOS_036M       = 0x10003601,
    VTGE_CAMERA_CMOS_300C       = 0x10030000,
    VTGE_CAMERA_CCD_200C        = 0x00020000,
    VTGE_CAMERA_CCD_140C        = 0x00014000,
    VTGE_CAMERA_CCD_140M        = 0x00014001,
    VTGE_CAMERA_CMOS_036C       = 0x10003600,
    VTGE_CAMERA_CMOS_500C       = 0x10050000,
    VTGE_CAMERA_CMOS_500M       = 0x10050001,
    VTGE_CAMERA_CMOS_130M       = 0x10013001,
    VTGE_CAMERA_CCD_140HM       = 0x000140B1,
    VTGE_CAMERA_V_CCD_100HM     = 0x000100B1,
    VTGE_CAMERA_CCD_140HC       = 0x000140B0,
    VTGE_CAMERA_CCD_400HC       = 0x000400B0

};
```

Get the GigE camera type from the VTGE_GetCameraType function.

Data item

VTGE_CAMERA_UNKNOW

Unknown type of camera

VTGE_CAMERA_CMOS_036M

The type of camera is VT-EXGCM360

VTGE_CAMERA_CMOS_300C

The type of camera is VT-EXGC3000

VTGE_CAMERA_CCD_200C

The type of camera is VT-EXGC2010D

VTGE_CAMERA_CCD_140C

The type of camera is VT-EXGC1400D

VTGE_CAMERA_CCD_140M

The type of camera is VT-EXGM1400D

VTGE_CAMERA_CMOS_036C

The type of camera is VT-EXGC360

VTGE_CAMERA_CMOS_500C

The type of camera is VT-EXGC5000

VTGE_CAMERA_CMOS_500M

The type of camera is VT-EXGCM5000

VTGE_CAMERA_CMOS_130M

The type of camera is VT-EXGM1000

VTGE_CAMERA_CCD_140HM

The type of camera is VT-EXGM2900D

VTGE_CAMERA_V_CCD_100HM

The type of camera is VT-EXGM1000D

VTGE_CAMERA_CCD_140HC

The type of camera is VT-EXGC2900D

VTGE_CAMERA_CCD_400HC

The type of camera is VT-EXGC4000D

VTGE_FLIP_MODE Enum

```
enum VTGE_FLIP_MODE
{
    VTGE_FLIP_NORMAL        = 0,
    VTGE_FLIP_HORIZONTAL    = 1,
    VTGE_FLIP_VERTICAL      = 2,
    VTGE_FLIP_ROTATE        = 3,
    VTGE_FLIP_LEFT90        = 4,
    VTGE_FLIP_RIGHT90       = 5
};
```

This enum is used in the VTGE_SetFlipMode function, used to setting the image flip mode

Data item

VTGE_FLIP_NORMAL

Normal mode, do not flip the image.

VTGE_FLIP_HORIZONTAL

Image flip horizontal

VTGE_FLIP_VERTICAL

Image flip vertical

VTGE_FLIP_ROTATE

Flip 270 °

VTGE_FLIP_LEFT90

Image filp to the left 90 °

VTGE_FLIP_RIGHT90

Image filp to the right 90 °

VTGE_ZOOM_MODE Enum

```
enum VTGE_ZOOM_MODE
{
    VTGE_ZOOM_1_1       = 0,
    VTGE_ZOOM_1_2       = 1,
    VTGE_ZOOM_1_4       = 2
};
```

This enum is used in the VTGE_SetZoomMode function, used to set the video scaling mode

Data item

VTGE_ZOOM_1_1

No scaling, the effect of cutting

VTGE_ZOOM_1_2

The narrow double ranks. Such as *1536 image reduction, will become the*768 image

VTGE_ZOOM_1_4

The ranks of the reduced two times. Such as *1536 image reduction, will become the *384 image

VTGE_CAP_MODE Enum

```
enum VTGE_CAP_MODE
{
    VTGE_CAP_TRIG       = 0,
    VTGE_CAP_SEQU       = 1,
    VTGE_CAP_ANTI       = 2
};
```

This enum is used in the VTGE_SetCapMode function, used to set the video acquisition mode

Data item

VTGE_CAP_TRIG

The trigger, trigger mode.（CCD and 36M CMOS open the video，a trigger signal, the camera will transmit a picture）

VTGE_CAP_SEQU

Sequence, flow pattern. Open the video, the camera does not stop to upload data.

VTGE_CAP_ANTI

（CCD and 36M CMOS）Resistance flash mode.

VTGE_SPEED_MODE Enum

```
enum VTGE_SPEED_MODE
{
    VTGE_SPEED_1_1      = 0,
    VTGE_SPEED_1_2      = 1,
    VTGE_SPEED_1_4      = 2,
    VTGE_SPEED_1_8      = 3
};
```

This enum is used in the VTGE_SetSpeedMode function, used to set the camera speed mode.

Data item

VTGE_SPEED_1_1

100% High speed mode

VTGE_SPEED_1_2

50% Medium speed mode

VTGE_SPEED_1_4

25% Low speed mode

VTGE_SPEED_1_8

12.5% Not used

VTGE_IP_CFG Enum

```
enum VTGE_IP_CFG
{
    VTGE_IP_RESET       = 0,
    VTGE_IP_SAVE        = 1
};
```

This enum is used in the VTGE_IPConfig function, used to set the IP

Data item

VTGE_IP_RESET

Reset the IP address

VTGE_IP_SAVE

Save the setted IP address

VTGE_IMG_TYPE Enum

```
enum VTGE_IMG_TYPE
{
    VTGE_IMG_BMP    = 0,
    VTGE_IMG_JPG    = 1,
    VTGE_IMG_TIF    = 2,
    VTGE_IMG_PNG    = 3
};
```

This enum is used in the VTGE_FrameToImage function，used to set the image format.

Data item

VTGE_IMG_BMP

BMP type

VTGE_IMG_JPG

JPGE type

VTGE_IMG_TIF

TIF type

VTGE_IMG_PNG

PNG type

VTGE_COLOR_MODE Enum

```
enum VTGE_COLOR_MODE {
    VTGE_COLOR_RGB24,
    VTGE_COLOR_BW24,
    VTGE_COLOR_GRAY,
    VTGE_COLOR_RAWDATA,
    VTGE_COLOR_MONO12,
    VTGE_COLOR_XWJ
};
```

This enum is used in the VTGE_SetColorMode function，used to set the video color mode

Data item

COLOR_RGB24

The 8 bit valid data is used to show the fast interpolation algorithm, Display image is RGB24 type.

COLOR_BW24

Display black and white images,the type of the image is RGB24,the three components are equal, all of that are the gray value of the current pixel.

COLOR_GRAY,

Display 8 bit gray image

COLOR_RAWDATA

The 8 bit data specified of the sensor acquired data to display.

COLOR_MONO12

The 16 bit original data which effective place for the 12 bit (Need a camera support)

COLOR_XWJ

Microscope mode (only supports color camera)

Functions Overview

VTGE_Initialize	To establish a network service connection
VTGE_UnInitialize	Disconnect the network service connection
VTGE_GetCameraCount	Gets the number of cameras
VTGE_GetCameraType	Gets the type of camera
VTGE_GetCameraInfo	Get the camera parameters
VTGE_InitCamera	Initialize camera
VTGE_UnInitCamera	Reverse initialize camera
VTGE_LiveStart	Open the video
VTGE_LiveStop	Close the video
VTGE_ResetCameraPort	Set the port number
VTGE_GetCameraPort	Get the port number
VTGE_SetExposure	To set the exposure time
VTGE_GetExposure	Get the exposure time
VTGE_SetGainR	Set the red gain
VTGE_SetGainG	Set the green gain
VTGE_SetGainB	Set the blue gain
VTGE_SetGain_A	Set the global gain
VTGE_SetResolution	Set the image resolution
VTGE_GetCameraFrameRate	To obtain the current camera display frame rate
VTGE_GetNICFrameRate	
VTGE_SetVideoWindowPos	Set the video display window size
VTGE_SetFlipMode	Setting flip mode of the video display
VTGE_SetZoomMode	Set the video scaling mode
VTGE_SetCapMode	Set the video acquisition mode
VTGE_SetSpeedMode	Set the camera speed mode
VTGE_SetOffsetH	Set horizontal offset of the camera sensor acquisition
VTGE_SetOffsetV	Set vertical offset of the camera sensor acquisition
VTGE_SetFrameCallBack	Setting the frame callback function
VTGE_GetFrame	Obtaining a frame of data into the specified memory
VTGE_GetFrameToImage	A frame of image data is saved as the format specified.

VTGE_SetCameraIP	Set the camera IP address, subnet mask, gateway
VTGE_IPConfig	Set the IP configuration
VTGE_SetContrast	Set the video contrast
VTGE_SetTriggerState	Set the external trigger state
VTGE_FrameToImage	The incoming data are saved as image
VTGE_SetFlashState	Set the flash status
VTGE_SetDoAE	Set up automatic exposure
VTGE_SetDoAWB	Set up automatic white balance
VTGE_SetColorMode	Set the video color mode
VTGE_SetDhcpEnable	Set automatically obtain IP address
VTGE_CaptureStart	Open the video
VTGE_CaptureStop	Close the video
VTGE_SelectAviCompressor	Select the video compressor
VTGE_SetAviCompressor	Set the video compressor
VTGE_SetTriggerMode	Set trigger mode
VTGE_GetTriggerMode	Get trigger mode
VTGE_SetPackageDelayTime	Set the delay time when the camera to send data packets
VTGE_GetPackageDelayTime	Get the delay time when the camera to send data packets
VTGE_SetWatchdogState	Set camera watchdog state.
VTGE_GetWatchdogState	Get camera watchdog state.
VTGE_SetIO	Set the camera state of the output IO
VTGE_GetIO	Get the camera state of the output IO
VTGE_SetTriggerDelayTime	Set the delay acquisition time of the camera after the trigger.
VTGE_SoftTrigger	Soft trigger gathering a frame of image
VTGE_HDR_Enable	To open the high dynamic range function of the camera sensor.
VTGE_MCResolutionShow	
VTGE_IsMCResolutionShow	Gets whether open cutting resolution display

VTGE_SetMCResolution	Set the parameters of the cutting resolution display.
VTGE_GetMCResolution	Get the relevant parameters about of cutting resolution display.
VTGE_ValidBitsOffSet	Set 8 bit effective data are according to the offset of the 12 bit effective data
VTGE_AEParam	Set automatic exposure parameter
VTGE_SetPackageSize	Set the size of the data packet
VTGE_GetPackageSize	Get the size of the data packet
VTGE_SetNewHwnd	Set the window handle

.Function return values description

ResSuccess	0x0000	Return success
ResNullHandleErr	0x0001	Invalid handle
ResNullPointerErr	0x0002	Pointer is null
ResFileOpenErr	0x0003	File / open failed
ResNoDeviceErr	0x0004	No device available
ResOutOfMemoryErr	0x0005	Memory is lack of allocation
ResNoPreviewRunningErr	0x0006	Do not open the preview
ResCreatCameraErr	0x0007	Camera initialization anomaly
ResNoUsablePortErr	0x0008	Port record anomaly
ResHasUsedPortErr	0x0009	The port number is occupied
ResNoMapPort2CameraErr	0x0010	The port number and the camera mapping table anomaly
ResNotSupportedErr	0x0011	The camera does not support the operation
ResMutexErr	0x0012	The current operation with other ongoing operations mutex
ResUnknownCameraErr	0x0013	Unknown type of camera
ResParamValueErr	0x0014	The parameter value is wrong
ResIndex2CameraErr	0x0015	Get the specified index camera is anomaly
ResReadPipeErr	0x0016	Read the pipeline anomaly
ResPipeValueErr	0x0017	Pipeline data anomaly
ResCreatNICManageErr	0x0018	Network manager to create anomaly
ResNoNICManageErr	0x0019	Network manager is not created
ResCreatMemoryErr	0x0020	The memory initialization is anomaly
ResNoPlayingErr	0x0021	Not open the preview
ResHasPlayingErr	0x0022	Open the preview
ResNoNICErr	0x0023	The card is not initialized
ResEnumerateCameraErr	0x0024	Enum the camera anomaly
ResProtocolDriverErr	0x0025	Protocol driver anomaly
ResNICStartWorkErr	0x0026	The card opening anomaly

ResNICStopWorkErr	0x0026	The card get off anomaly
ResNICHasWorkErr	0x0027	The card is already opened
ResNICNoWorkErr	0x0028	The card is already closed
ResNICCreatErr	0x0029	Card creation failed
ResNoControlRightErr	0x0030	No control camera right
ResReadRegisterErr	0x0031	Reading the register anomaly
ResWriteRegisterErr	0x0032	Writting the register anomaly
ResDevWorkingErr	0x0033	The equipment is being used
ResNICEnumeratedErr	0x0034	The card isn't full enum

Function description

VTGE_Initialize

To establish a network service connection

Syntax

VT_API VTGE_Initialize();

Description

The user must call for the function before getting the camera parameters which are connected tothe system.

VTGE_UnInitialize

Disconnect the network service connection.

Syntax

VT_API VTGE_UnInitialize();

D**escription**

The user call the function at the end of the program

VTGE_GetCameraCount

This function is used to detect the number of cameras which are connected to the system, and returns the number of cameras.

Syntax

```
VT_API    VTGE_GetCameraCount(
        OUT int          *Count
);
```

Parameters

Count

[out] Returns the number of cameras which are connected to the system.

VTGE_GetCameraType

Gets the type of camera

Syntax

```
VT_API    VTGE_GetCameraType(
        IN int           nIndex,
        OUT DWORD        *dType
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera

dType

[out] The type of the current camera.

D**escription**

Type number is defined in the VTGE_CAMERA_TYPE. The user can decide the camera classification directly through the camera returns hardware models.

VTGE_GetCameraInfo

Read the camera hardware parameters, such as camera model, IP address, MAC address and so on.

Syntax

```
VT_API    VTGE_GetCameraInfo(
    IN int              nIndex,
    OUT LPVOID          pCameraInfo
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

pCameraInfo

[out] Returns the current camera hardware parameters, data format are like to theGigECameraInfo structure.

VTGE_InitCamera

Initialize camera

Syntax

```
VT_API    VTGE_InitCamera(
    IN int              nIndex,
    IN HWND             hWnd
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

hWnd

[in] The specified window handle is used to display the video

VTGE_UnInitCamera

Reverse initialize camera. Release of pre allocation of space camera.

Syntax

```
VT_API    VTGE_UnInitCamera(
    IN int          nIndex
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

Description

To be called before the VTGE_UnInitialize function.

VTGE_LiveStart

Open the video

Syntax

VT_API VTGE_LiveStart(

IN int nIndex

);

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

VTGE_LiveStop

Close the video

Syntax

VT_API VTGE_LiveStop(

IN int nIndex

);

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

VTGE_ResetCameraPort

Set the port number

Syntax

```
VT_API    VTGE_ResetCameraPort(
    IN int          nIndex,
    IN USHORT       nPort
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

nPort

[in] The port number, range of 32768~65535

Description

The range of the port number provided by the WINDOWS system is 0~ 65535, have been part of a fixed port occupied, the user can not be assigned. The software in the initialization has been assigned the default port number for the user, the user can not set in most cases, but does not guarantee that there is no conflict. When a conflict occurs, require the user to call this function to change the port number, users are advised in the distribution of the port number, use the port number greater than 32768.

VTGE_GetCameraPort

Get the port number

Syntax

VT_API VTGE_GetCameraPort(

IN int nIndex,

IN USHORT *Port

);

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

nPort

[in] The port number pre allocation

VTGE_SetExposure

To set the exposure time

Syntax

VTGE_SetExposure(

IN int nIndex,

IN DWORD dwEx

);

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

dwEx

[in] The camera exposure time

Description

The exposure time of different types of cameras are not the same. The exposure time and photosensitive chip is the valid time of a line as a unit. Is mainly affected by the sensor clock and a row of data, a time from several microseconds to hundreds of microseconds range. In practical application, the system will automatically trip time integer.

VTGE_GetExposure

Get the exposure time

Syntax

```
VT_API    VTGE_GetExposure(
    IN int              nIndex,
    OUT DWORD    *dwEx
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

dwEx

[out] The camera exposure time

VTGE_SetGainR

Set the red gain

Syntax

VT_API VTGE_SetGainR(

IN int nIndex,

IN DWORD dwR

);

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

dwR

[in] Red gain value，The effective range is not the same in different camera.

Gain range as follows:

GigE-CMOS30M: The black and white camera, the function is not valid.

GigE-CMOS300C: 1 ~ 63

GigE-CCD200C: 1 ~ 63

Description

This function is not valid for the black and white camera. The gain value is too large, the image noise becomes large. The gain value is too small, the image color reduction degree of variation get worse.

VTGE_SetGainG

Set the green gain

Syntax

VT_API VTGE_SetGainG(

IN int nIndex,

IN DWORD dwG

);

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

dwG

[in] Green gain value，The effective range is not the same in different camera.

Gain range as follows:

GigE-CMOS30M: The black and white camera, the function is not valid.

GigE-CMOS300C: 1 ~ 63

GigE-CCD200C: 1 ~ 63

Description

This function is not valid for the black and white camera. The gain value is too large, the image noise becomes large. The gain value is too small, the image color reduction degree of variation get worse.

VTGE_SetGainB

Set the blue gain

Syntax

```
VT_API    VTGE_SetGainB(
    IN int          nIndex,
    IN DWORD        dwB
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

dwG

[in] Blue gain value，The effective range is not the same in different camera.

Gain range as follows:

GigE-CMOS30M: The black and white camera, the function is not valid.

GigE-CMOS300C: 1 ~ 63

GigE-CCD200C: 1 ~ 63

Description

This function is not valid for the black and white camera. The gain value is too large, the image noise becomes large. The gain value is too small, the image color reduction degree of variation get worse.

VTGE_SetGain_A

Set the global gain.

Syntax

```
VT_API    VTGE_SetGain_A(
    IN int          nIndex,
    IN DWORD        dwGain
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

dwGain

[in] Gain value，The effective range is not the same in different camera.

Gain range as follows:

GigE-CMOS30M:	1 ~ 63
GigE-CMOS300C:	Color camera, the function is not valid
GigE-CCD200C:	1~255 AD gain.

Description

The gain value have obvious influence on the image brightness. Larger gain value, image brightness increase, while the noise is bigger. In a dark scene, can pass to obtain appropriate brightness adjusting the parameters. In the capture of moving objects, can increase the parameter values, and appropriately reduce the exposure time, can effectively reduce the tailing phenomenon.

VTGE_SetResolution

Set the image resolution

Syntax

```
VT_API    VTGE_SetResolution(
    IN int          nIndex,
    IN DWORD        dwWidth,
    IN DWORD        dwHeight
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

dwWidth

[in] The width of the collected image

dwHeight

[in] The height of the collected image

VTGE_GetCameraFrameRate

To obtain the current camera display frame rate

Syntax

```
VT_API    VTGE_GetCameraFrameRate(
    IN int          nIndex,
    OUT float       *fr
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

fr

[out] Display the current frame rate

VTGE_GetNICFrameRate

Syntax

```
VT_API    VTGE_GetNICFrameRate(
    IN int        nIndex,
    OUT float     *fr
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

fr

[out] The card frame rate

VTGE_SetVideoWindowPos

Set the video display window size.

Syntax

```
VT_API    VTGE_SetVideoWindowPos(
    IN int        nIndex,
    IN int        StartX,
    IN int        startY,
    IN int        nWidth,
    IN int        nHeight
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

StartX

[in] The level starting position of the video in the display window.

StartY

[in] The vertical starting position of the video in the display window.

nWidth

[in] The width of the video display.

nHeight

[in] The height of the video display.

VTGE_SetFlipMode

Setting flip mode of the video display

Syntax

```
VT_API    VTGE_SetFlipMode(
    IN int                  nIndex,
    IN VTGE_FLIP_MODE       FlipMode
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

FlipMode

[in] Video flip mode

Description

Supported flip types as definition at VTGE_FLIP_MODE.

VTGE_SetZoomMode

Set the video zoom mode.

Syntax

```
VT_API    VTGE_SetZoomMode(
    IN int                  nIndex,
    IN VTGE_ZOOM_MODE       ZoomMode
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

ZoomMode

[in] Video scaling mode

Description

Supported flip types as definition at VTGE_ZOOM_MODE.

VTGE_SetCapMode

Set the video acquisition mode

Syntax

```
VT_API    VTGE_SetCapMode(
    IN int                  nIndex,
    IN VTGE_CAP_MODE   CapMode
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

CapMode

[in] Camera video acquisition mode

Description

This function is only valid for CCD series cameras and VTGECMOS36M camera.Parameters are defined at the VTGE_CAP_MODE function.

VTGE_SetSpeedMode

Set the camera speed mode

Syntax

```
VT_API    VTGE_SetSpeedMode(
    IN                          int nIndex,
    IN VTGE_SPEED_MODE     SpeedMode
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

SpeedMode

[in] Speed grade

Description

Control the collection speed of a single camera video.

VTGE_SetOffsetH

Set horizontal offset of the camera sensor acquisition.

Syntax

VT_API

VTGE_SetOffsetH(

IN int nIndex,

IN DWORD dwH

);

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

dwH

[in] Horizontal offset.

Description

CCD series camera does not support this function. Valid range of parameters should be: 0 to(the maximum resolution width of camera- the width of the current camera resolution)

VTGE_SetOffsetV

Set vertical offset of the camera sensor acquisition.

Syntax

```
VT_API    VTGE_SetOffsetV(
    IN int          nIndex,
    IN DWORD        dwV
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

dwV

[in] Vertical offset

Description

CCD series camera does not support this function. Valid range of parameters should be: 0 to(the maximum resolution height of camera- the height of the current camera resolution)

VTGE_SetFrameCallBack

Setting the frame call back function.

Syntax

```
VT_API    VTGE_SetFrameCallBack(
    IN int                          nIndex,
    IN VTGE_FRAMECALLBACK   FrameCB,
    IN LPVOID                       lpContext
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

FrameCB

[in] Callback function pointer

lpContext

[in] The device context pointer

Description

Through the function setting a callback for each frame of data processing.

VTGE_GetFrame

Obtaining a frame of data into the specified memory.

Syntax

```
VT_API    VTGE_GetFrame(
    IN int              nIndex,
    OUT PBYTE      pData
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

pData

[out] The first pointer of memory address

Description

The user must allocate enough memory in here, or it is likely to overflow the memory.

VTGE_GetFrameToImage

A frame of image data is saved as the format specified.

Syntax

```
VT_API    VTGE_GetFrameToImage(
    IN int                  nIndex,
    IN LPCTSTR              lpszFileName,
    IN VTGE_IMG_TYPE        ImgType,
    IN PBYTE                pData,
    IN DWORD                dwParam
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

lpszFileName

[in] Set the path and file name

ImgType

[in] Save image type

pData

[in] Retention parameter, set to NULL

dwParam

[in] Retention parameter, set to 0

VTGE_SetCameraIP

Set the camera IP address, subnet mask, gateway

Syntax

```
VT_API    VTGE_SetCameraIP(
    IN int nIndex,
    IN DWORD dwIpAddr,
    IN DWORD dwMask,
    IN DWORD dwGateway
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

dwIpAddr

[in] IP address

dwMask

[in] subnet mask

dwGateway

[in] gateway

Description

This function is used to set up a temporary IP address, but is not saved in the camera. Set in the current operating effectively, but after the power set of IP addresses and other information will be lost.

VTGE_IPConfig

Set the IP configuration

Syntax

```
VT_API    VTGE_IPConfig(
    IN int                  nIndex,
    IN VTGE_IP_CFG    IpCfg
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

IpCfg

[in] Select the IP configuration mode.

VTGE_SetContrast

Set the video contrast

Syntax

```
VT_API    VTGE_SetContrast(
    IN int              nIndex,
    IN DWORD       dwContrast
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

dwContrast

[in] Contrast value, the effective range of 1~63, the default value of 32.

VTGE_SetTriggerState

Adjust the video contrast

Syntax

```
VT_API    VTGE_SetTriggerState(
    IN int                          nIndex,
    IN BOOL                         bCheck,
    IN VTGE_FRAMECALLBACK           TrgCallBack,
    IN LPVOID                           lpContext
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

bCheck

[in] External trigger flag, TRUE effectively

TrgCallBack

[in] The callback function pointer

lpContext

[in] The device context

VTGE_FrameToImage

The incoming data are saved as image

Syntax

```
VT_API    VTGE_FrameToImage(
    IN LPCTSTR              strFileName,
    IN LPVOID                   pImageData,
    IN ULONG                    _ulWidth,
    IN ULONG                    _ulHeight,
    IN ULONG                    nColor,
    IN VTGE_IMG_TYPE        ImgType);
```

Parameters

strFileName

[in] The name of the saved picture.

pImageData

[in] Save the storage data pointer of the image

_ulWidth

[in] The width of the image

_ulHeight

[in] The height of the image

nColor

[in] Color bits（8 or 24）

ImgType

[in] The type of the image.

Description

The nColor parameter is needed to set the 8 bit grayscale or 24 bit RGB images according to the specific format of image data. Supported image types are defined at thet VTGE_IMG_TYPE function.

VTGE_SetFlashState

Set the flash status

Syntax

```
VT_API    VTGE_SetFlashState(
    IN int          nIndex,
    IN BOOL         bFlashState,
    IN BOOL         bLeverState
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

bFlashState

[in] Flash status. TRUE effectively

bLeverState

[in] Set the flash in effective level (TRUE active high, FALSE active low)

Description

Currently supports VTGECMOS36M camera, and CCD200C camera.

VTGE_SetDoAE

Set up automatic exposure

Syntax

```
VT_API    VTGE_SetDoAE(
    IN int                      nIndex,
    IN BYTE                     btTarget,
    IN BOOL                     bAE,
    IN VTGE_AUTOCALLBACK        pfAECallback,
    IN LPVOID                       lpContext
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

btTarget

[in] Whether to display the logo.

bAE

[in] Automatic exposure effective mark, TRUE:open automatic exposure.

pfAECallback

[in] Callback functions, used to return the current exposure value.

pContext

[in] The device context.

VTGE_SetDoAWB

Set up automatic white balance

Syntax

```
VT_API    VTGE_SetDoAWB(
    IN int nIndex,
    IN BYTE btTarget,
    IN BOOL bAWB,
    IN VTGE_AUTOCALLBACK pfAWBCallback,
    IN LPVOID lpContext
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

btTarget

[in] Recalculate the white balance data.

bAWB

[in] Whether to open the loading of white balance

pfAECallback

[in] The callback function (currently reserved for future use, set to NULL)

pContext

[in] The device context (currently reserved for future use, set to NULL)

Description

The black and white camera does not support the operation.

VTGE_SetColorMode

Set the video color mode.

Syntax

```
VT_API    VTGE_SetColorMode(
    IN int                  nIndex,
    IN VTGE_COLOR_MODE    ColorMode
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

ColorMode

[in] Select the video display mode

Description

The parameters are definited at VTGE_COLOR_MODE function.

VTGE_SetDhcpEnable

To obtain IP address automatically.

Syntax

```
VT_API    VTGE_SetDhcpEnable(
    IN int          nIndex,
    IN BOOL         bEnableDhcp
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

bEnableDhcp

[in] Automatically obtain IP address When the state is TRUE.

VTGE_CaptureStart

Open the video.

Syntax

```
VT_API    VTGE_CaptureStart(
    IN int              nIndex,
    IN LPCTSTR          strFileName
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

strFileName

[in] The path of the video files.

VTGE_CaptureStop

Close the video

Syntax

```
VT_API    VTGE_CaptureStop(
    IN int          nIndex
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

VTGE_SelectAviCompressor

Select the video compressor

Syntax

```
VT_API    VTGE_SelectAviCompressor(
    IN int              nIndex,
    OUT DWORD           *fccHandler
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

fccHandler

[out] The output selection of compressor logo.

VTGE_SetAviCompressor

Set the video compressor

Syntax

VT_API VTGE_SetAviCompressor(

IN int nIndex,

IN DWORD fccHandler

);

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

fccHandler

[in] Need to set the compressor identification

Description

fccHandler If the parameters is 0, without the use of compressor.

VTGE_SetTriggerMode

Set trigger mode

Syntax

VT_API

VTGE_SetTriggerMode(

IN int nIndex,

IN VTGE_TRIGGER_MODE TriggerMode

);

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

TriggerMode

[in] The trigger mode which needed to set.

VTGE_GetTriggerMode

Get trigger mode

Syntax

```
VT_API    VTGE_GetTriggerMode(
    IN int                          nIndex,
    OUT VTGE_TRIGGER_MODE           *TriggerMode
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

TriggerMode

[out] The trigger mode of the current camera.

VTGE_SetPackageDelayTime

Set the delay time when the camera to send data packets，unit: 20ns.

Syntax

```
VT_API    VTGE_SetPackageDelayTime(
    IN int          nIndex,
    IN int          nTime
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

nTime

[in] Set the camera data packet delay sending time, range: 0-65535.

Description

Can greatly reduce the probability of packet loss by setting the camera camera data packet transmission delay.

VTGE_GetPackageDelayTime

Get the delay time when the camera to send data packets, unit: 20ns.

Syntax

```
VT_API    VTGE_GetPackageDelayTime(
    IN int          nIndex,
    OUT int         *nTime
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

nTime

[out] Set the camera data packet delay sending time, range: 0-65535.

VTGE_SetWatchdogState

Set camera watchdog state.

Syntax

```
VT_API    VTGE_SetWatchdogState(
    IN int              nIndex,
    IN BOOL             bUse
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

bUse

[in] Whether to open the watchdog. TRUE, open; FALSE ,close.

VTGE_GetWatchdogState

Get camera watchdog state.

Syntax

```
VT_API    VTGE_GetWatchdogState(
    IN int          nIndex,
    OUT BOOL        *bUse
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

bUse

[out] Get camera watchdog state. TRUE, open; FALSE ,close.

VTGE_SetIO

Set the camera state of the output IO

Syntax

VT_API VTGE_SetIO(

IN int nIndex,

IN int nIOIndex,

IN int nValue

);

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

nIOIndex

[in] Index range of IO port：0-2

nValue

[in] The IO state, value: 0 or 1

Parameters

The camera is provided with three output IO ports,

VTGE_GetIO

Get the camera state of the output IO.

Syntax

```
VT_API    VTGE_GetIO(
    IN int        nIndex,
    IN int        nIOIndex,
    OUT int       *nValue
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

nIOIndex

[in] The index range of IO port: 0-1

nValue

[out] The IO state, value: 0 or 1

Description

The camera is provided with two input IO ports.

VTGE_SetTriggerDelayTime

Set the delay acquisition time of the camera after the trigger.

Syntax

VT_API VTGE_SetTriggerDelayTime(

IN int nIndex,

IN int nTime

);

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

nTime

[in] The delay time, range: 0-65535 unit: 20ns

Description

The method for exposure camera trigger delay time flash will open until the chip end exposure; frame exposure camera flash in delay time not to open the flash exposure with the chip synchronization.

VTGE_SoftTrigger

Soft trigger gathering a frame of image

Syntax

VT_API VTGE_SoftTrigger(

IN int nIndex

);

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

Description

The method will call the trigger frame callback function after execution.

VTGE_HDR_Enable

To open the high dynamic range function of the camera sensor.

Syntax

```
VT_API    VTGE_HDR_Enable(
    IN int          nIndex,
    IN LONG     nEnable
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

Description

This method is effective for 36W CMOS camera and the color camera.

VTGE_MCResolutionShow

Syntax

```
VT_API    VTGE_MCResolutionShow(
    IN int          nIndex,
    IN BOOL     nEnable
);
```

Parameters

nIndex

[in] Camera index, used to distinguish the different camera.

nEnable

[in] Indicates whether to open. TRUE, open;FALSE, close .

Description

In the opening cut resolution display, preview, take pictures, the image size of the vedio and the callback function is modified as the size of cutting resolution.

VTGE_IsMCResolutionShow

Set the parameters of the cutting resolution display.

Syntax

```
VT_API    VTGE_IsMCResolutionShow(
    IN int              nIndex,
    OUT BOOL            *nEnable
);
```

Parameters

nIndex

[in] Camera index, used to distinguish between different camera clipping resolution display.

nEnable

[out] The return value indicates whether to open.TRUE，open；FALSE，close.

VTGE_SetMCResolution

Set the relevant parameters about of cutting resolution display.

Syntax

```
VT_API    VTGE_SetMCResolution(
    IN int        nIndex,
    IN int        nOffX,
    IN int        nOffY,
    IN int        nWidth,
    IN int        nHeight
);
```

Parameters

nIndex

[in] Camera index, used to distinguish between different camera clipping resolution display.

nOffX

[in] Set clipping resolution relative to the X offset the lower left corner of the current camera captured image.

nOffY

[in] Set clipping resolution relative to the Y offset the lower left corner of the current camera captured image.

nWidth

[in] Set clipping resolution width.

nHeight

[in] Set clipping resolution height.

Description

Set the current resolution in the application set the resolution function, offset cutting resolution will recover 0, size will be set to the current setting of the resolution.In the set time please ensure the legitimacy of the data set:

nOffX+nWidth <= (The current resolution width of the camera),

nOffY+nHeight <= (The current resolution height of the camera),

VTGE_GetMCResolution

Get the relevant parameters about of cutting resolution display.

Syntax

```
VT_API    VTGE_GetMCResolution(
    IN int        nIndex,
    OUT int       *nOffX,
    OUT int       *nOffY,
    OUT int       *nWidth,
    OUT int       *nHeight
);
```

Parameters

nIndex

[in] Camera index, used to distinguish between different camera clipping resolution display.

nOffX

[out] Get clipping resolution relative to the X offset the lower left corner of the current camera captured image.

nOffY

[out] Get clipping resolution relative to the Y offset the lower left corner of the current camera captured image.

nWidth

[out] Get clipping resolution width.

nHeight

[out] Get clipping resolution height.

VTGE_ValidBitsOffSet

Set the data in Mono12 format when the camera to upload 16 effective data, which the 8 bit effective data is relate to the 12 bit valid data.

Syntax

```
VT_API    VTGE_SetValidBitsOff(
    IN int        nIndex,
    IN int        nValid
);
```

Parameters

nIndex

[in] Camera index, used to distinguish between different camera clipping resolution display.

nValid

[in] The effective upload data is relate to the offset of a valid data values,range:0-4.

Description

Need to be able to support the Mono12 format can upload 16 bit data of the camera can be used.

VTGE_AEParam

To set the level of exposure and real time exposure.

Syntax

```
VT_API    VTGE_AEParam(
    IN                  int nIndex,
    IN BOOL             bLiving,
    IN VTGE_AE_LEVEL    AELevel
);
```

Parameters

nIndex

[in] Camera index, used to distinguish between different camera clipping resolution display.

bLiving

[in] Real time exposure opening marker

AELevel

[in] Set automatic exposure level.

VTGE_SetNewHwnd

Set the window handle.

Syntax

```
VT_API    VTGE_SetNewHwnd (
    IN int              nIndex,
    IN HWND             hwnd
);
```

Parameters

nIndex

[in] Camera index, used to distinguish between different camera clipping resolution display.

hwnd

[in] The window handle

VTGE_ SetPackageSize

Set the size of the data packet，The default is 1500.

Syntax

VT_API VTGE_ SetPackageSize (

IN int nIndex,

IN USHORT nPackage

);

Parameters

nIndex

[in] Camera index, used to distinguish between different camera clipping resolution display

nPackage

[in] The size of the packet, range of 1500-8000, need is a multiple of 4.

Description

The method requires network support giant frame transmission.

VTGE_ GetPackageSize

Get the size of the data packet,the default is 1500

Syntax

VT_API VTGE_ GetPackageSize (

IN int nIndex,

IN USHORT *nPackage

);

Parameters

nIndex

[in] Camera index, used to distinguish between different camera clipping resolution display

nPackage

[in] The size of the packet, range of 1500-8000, need is a multiple of 4.

Description

The method requires network support giant frame transmission.

The use of a callback function

AWB/AE callback function

By the callback function is called at the end of the AWB and AE, used to return the results of automatic process

The function definition

```
typedef  VOID      (WINAPI *VTGE_AUTOCALLBACK )(
          DWORD      dw1,
          DWORD      dw2,
          LPVOID lpContext
          );
```

Parameters

dw1

Returns the result of automatic process

dw2

Returns the result of automatic process

lpContext

In the open automatic process, user context settings, used for passing parameters.

Description

In the automatic white balance process, dw1 is the gain after the AWB , dw2 is the AE value after the AWB , the following results:

```
// If the value of dw1 is 0x00AABBCC
     ULONG Result        = dw1;                          // 0x00AABBCC
     BYTE btRed          = ( BYTE )(Result);             // 0xCC  Red Gain
     BYTE btGreen        = ( BYTE )( Result>>8 );        // 0xBB Green Gain
BYTE btBlue          = ( BYTE )( Result>>16 );       // 0xAA Blue Gain
ULONG Exposure     = dw2;
```

In the automatic exposure process, dw1 is the exposure value after the AE.

Frame the callback function

The frame of the callback function VTGE_SetFrameCallBack set, used to frame data operation.

The function definition

```
typedef    VOID    (WINAPI *VTGE_FRAMECALLBACK)(
        LPVOID lpParam1,
        LPVOID lpParam2,
        LPVOID lpContext
);
```

Parameters

lpParam1

The memory address is stored frame data.

lpParam2

Retain

lpContext

Setting the frame callback function, user context settings, used for passing parameters.

GigE SDK use Examples

Initialize camera

```
int          nIndex = 0;                    // The specified camera serial number, the first camera was 0
if( ResSuccess == VTGE_Initialize())       // Initialize the DLL Dynamic Link Library

    {
        If(ResSuccess != VTGE_InitCamera(   // Initialize camera
        nIndex,                             // Camera serial number
                this->m_hWnd ) )            // Camera display window handle
        {
        VTGE_UnInitCamera( nIndex );    //If you don't succeed, then anti initialization
            VTGE_UnInitialize();
            ::MessageBox( NULL, _T("InitCamera Error"), _T("Demo"), 0 );
            return;
        }
    }
    else
    {
        VTGE_UnInitialize();      // The uninitialization of DLL Dynamic Link Library
                ::MessageBox( NULL, _T("Initial Error"), _T("Demo"), 0 );
        return;
    }
```

Set the camera parameters

```
int   m_nExp=480;                  // The specified exposure time
int   m_nGain[25];                 // The specified global gain
memset(&m_nGain[0], 0, sizeof(m_nGain));
m_nGain[0] = 32;
int   m_GainR=8;                   // Specify the red gain
int   m_GainG=8;                   // Specify the green gain
int   m_GainB=8;                   // Specify the blue gain
int   m_nCollectionWidth=1024;     // Camera acquisition resolution
int   m_nCollectionHeight=768;
int   m_nContrast=32;              // Specify the contrast
VTGE_SetResolution( nIndex,m_nCollectionWidth, m_nCollectionHeight);
VTGE_SetGain_A(nIndex, m_nGain[0]);
VTGE_SetGainR(nIndex,m_GainR);
VTGE_SetGainG(nIndex, m_GainG);
VTGE_SetGainB(nIndex, m_GainB);
VTGE_SetColorMode(nIndex, VTGE_COLOR_MODE(0));
VTGE_SetExposure(nIndex ,m_nExp);
VTGE_SetFlipMode(nIndex, VTGE_FLIP_MODE(0));
```

Video Preview

```
VTGE_LiveStart(nIndex);// Open the preview
VTGE_LiveStop(nIndex);// Clo the preview
```

Acquisition of A Frame Image

```
CString strPath;
strPath = _T("Sample");
TCHAR szPath[255];
memset(szPath, 0, 255*sizeof(TCHAR));
// A frame of data are saved in the BMP file
_stprintf(szPath, _T("%s.jpg"), strPath);
VTGE_GetFrameToImage(
        nIndex,
        szPath , // Set the path and file name of the saved files
        VTGE_IMG_TYPE(1), // The file type
        0, 0);

// A frame of data are saved in the JPG file
_stprintf(szPath, _T("%s.jpg"), strPath);
VTGE_GetFrameToImage(
        nIndex,
        szPath , // Set the path and file name of the saved files
        VTGE_IMG_TYPE(1), // The file type
        0, 0);
```

AE

// We need to define an automatic exposure callback function, called after the automatic exposure process.

```
VOID CALLBACK AECallBack(
    DWORD dw1,          // the results of the automatic process
    DWORD dw2,          // the results of the automatic process
    LPVOID lpContext)  // Set context in the open AE
{
    // dw1 is the automatic exposure value in the AE process
    // Update the user interface with dw1 results
};

//Open the AE
VTGE_SetDoAE(
    nIndex,             // Camera serial number
    1,                  // Display identifiers
    bAE,                // AE effective mark
    AECallBack,         // AE callback function
lpContext );            //Context
```

AWB

```
//Open the AWB
VTGE_SetDoAWB(
        nIndex,         // Camera serial number
        0,              // Recalculate the white balance data
        bAWB,           // Whether to open the loading of white balance
        NULL,           // Retain
        NULL );         // Retain
```

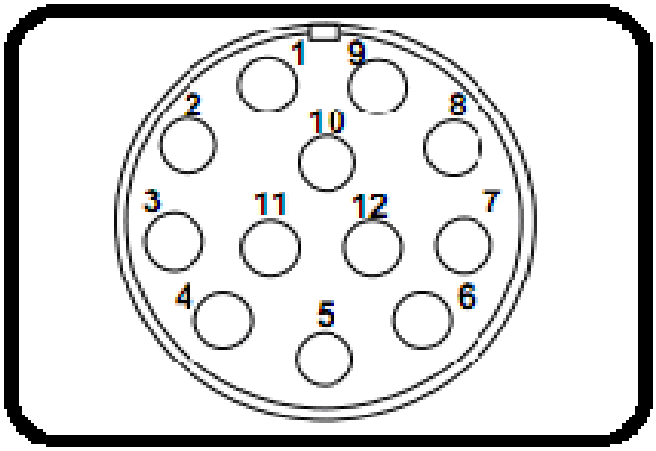
Appendix

Camera interface description

Interface overview

VTGE series camera provides two interfaces, a GigE interface RJ45 socket standard, another is the terminal socket, mainly to provide camera IO and power. In the RJ45 socket with the above two indicating lights, yellow light flashing when the camera data received, the green light flash that camera in sending data.

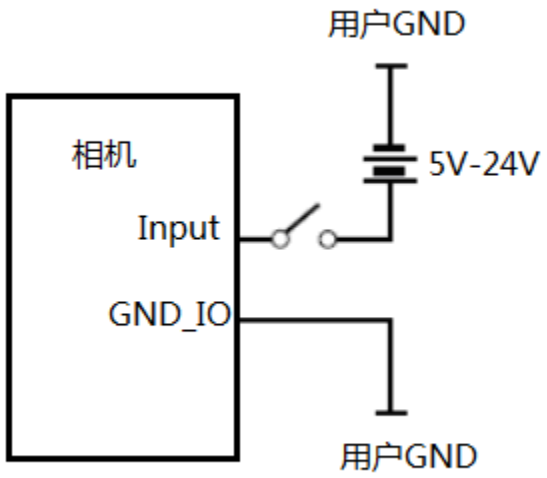
Terminal pin definition



PIN	Color	Signal	Description
1	Orange	Input4	AC synchronous signal input
2	Yellow	Input3	External trigger signal input
3	Green	Output3	The camera output port
4	Navy Blue	Output2	Flash output control
5	Light blue	GND_IO	The user, as the input signal reference ground

6	Violet	VCC_IO	The user power supply, used to control the output signal level, 5V-24V
7	Grey	Input2	The user input port
8	Pink	Input1	The user input port
9	White	Output1	The camera output port
10	Red	VCC	Power input 9V-12V/1A
11	Brown	Output4	The camera output port
12	Black	GND	Power ground input

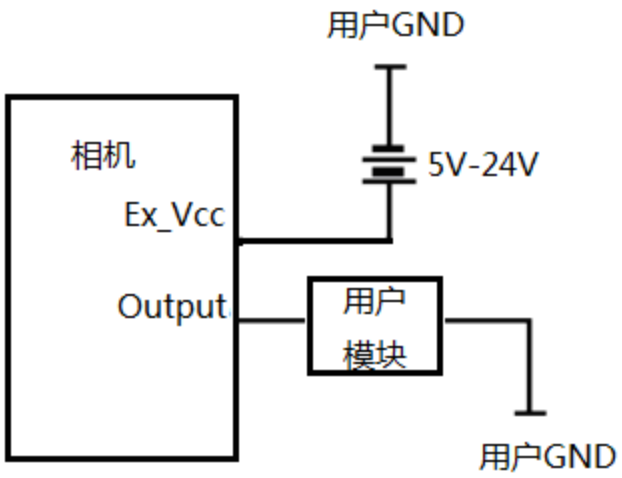
The input circuit control description



VTGE series camera provides four input control, are equipped with internal optical coupling isolation, designated one of two AC anti flicker and the external trigger special, the rest of the two reserved for user to develop two times, in to the camera provides the input control signal, the control circuit as shown above, provides the input signal to the specified input tube feet, voltage range of the input signal is 5~24V. In the input signal to the control signal of the reference and camera users to GND_IO together.

Note: voltage range here refers to the input voltage range at the port, if the signal long-distance transmission, users should consider the pressure drop loss on the line. The long-distance transmission, the input voltage of users should be more than 5V, instead of just the 5V input.

The output circuit control description



VTGE series camera provides four output control, are equipped with internal optical coupling isolation, specify a path which is for flash pin, three road to the rest of the reserved for user to develop two times, in to the camera provides an output control signal, the control circuit as shown above, providing power signals output signal to the EX_VCC pin the range of 5V-24V, EX_VCC, the output pin will output voltage range of 0-EX_VCC control signal, the driving current of each output can provide 50mA.

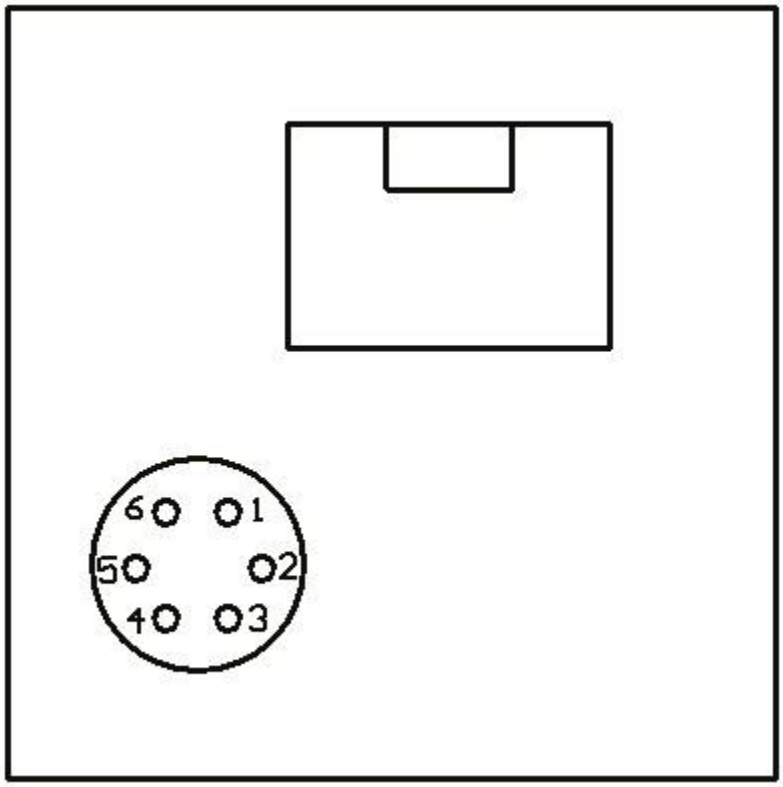
Note: the voltage range of EX_VCC refers to the input voltage range at the port of. If the signal long-distance transmission, to consider the loss of transmission line. In addition to output drive current maximum support only 50mA, over time will burn the camera port, not directly drive high power devices. If the flash output can only be used as a strobe control signal, rather than the direct signal driving high-power flash or light board.

Cable requirements

Ethernet interface to recommend the use of super class five or six lines, wiring distance not exceeding 100m.Fit cable IO terminal and socket, power supply and IO control. Please refer to the terminal pin definitions wiring sequence.

MINI camera interface specifications

Pin color description



Serial number	Line color	Description
1	Yellow (brown)	the negative pole of optocoupler output
2	Black	The power input ground
3	Red	The anode of 12V input power
4	Blue	The input anode of the Optocoupler
5	Green	The output negative pole of the Optocoupler
6	White	The output anode of the Optocoupler

Port description

Voltage range optocoupler input signal is 5~24V, the current in the 5-15mA.

Note:

Voltage range here refers to the input voltage range at the port, if the signal long-distance transmission, users should consider the pressure drop loss on the line. The long-distance transmission, the input voltage of users should be more than 5V, instead of just the 5V input. Optocoupler input with a reverse diode protection, access please note the positive and negative optocoupler input pole.

Optocoupler output signal terminal of the driving current maximum 50mA, maximum saturation voltage is less than 0.8V. Optocoupler output also connected with reverse diode protection, please note that the polarity connection.

Input and output the maximum pressure to 30V, more than likely to damage the camera.