GigE Camera controls OCX Instructions

OCX controls instructions

1. It doesn't bound to a control specifies camera when the contorls initialize.
2. The camera's serial number by systematic enumeration, serial number starting with 0.
3. The return value of function to temporarily retain no special instructions.

Control method instructions

LONG GetDeviceCount(void);
Description：
Get the number of the available cameras from the system.

void RemoveDevice(void);
Description：
Remove control for camera , release the camera.

void GetCurDeviceIndex(LONG* Index);
Description：
Get the control camera's serial number.
Parameters:
* Index：[OUT] The camera's serial number.

void SelectDevice(LONG deviceIdx);
Description：
To specified the camera which bind to the controls.
Parameters:
deviceIdx：[IN] The camera's serial number.

void GetDeviceName(LONG DeviceIdx, VARIANT* deviceName);
Description：
Get the camera's name.
Parameters:
DeviceIdx：[IN] The camera's serial number.
* deviceName：[OUT] The camera's name.

LONG GetDeviceType(LONG DeviceType);
Description：
Get the camera's medel which bind to the controls.
Parameters:
DeviceType，the camera's medel，Description see Appendix VTGE_CAMERA_TYPE.

void VideoLiveStart(void);
Description：

Open the camera preview.

void VideoLiveStop(void);

Description：

Close the camera preview.

void SetFullScreen(VARIANT_BOOL bFullScreen);

Description：

Image preview setting controls whether the size control filling.Parameters:

bFullScreen：FALSE，Display mode scaling；TRUE，The whole control fill mode.

LONG IsVideoLive(void);

Description：

Whether to open the preview，The return value：0，Close；1，Open.

FLOAT GetFrameRate(void);

Description：

Get the current display frame rate.

LONG GetResolutions(LONG* Count, LONG* pWidth, LONG* pHeight);

Description：

The return value：The current display frame rate.

Gets the resolution information.

Parameters:

Count: In the acquisition of resolution information, this parameter is used to specify the index number of the resolution. When the need to obtain the resolution number, the parameters reported several resolution current camera support.

pWidth: If the Count specifies a valid index, this parameter is used to receive the index specified resolution width. This parameter can be NULL, this function using Count and return value returns the number of resolution.

pHeight: If Count specifies a valid index, this parameter is used to receive the index specified resolution height. This parameter can be NULL, this function using Count and return value returns the number of resolution.

The return value：

Returns the number of resolution.

LONG GetCurShowResolution(LONG* pWidth, LONG* pHeight);

Description：

Get the current image data in the actual resolution.

Parameters:

* pWidth：Resolution width.

* pHeight：Resolution height.

LONG SetImageType(LONG dwType);

Description：

Set the save image file types.

Parameters:

dwType: Specify the image file type，Description see Appendix VTGE_IMG_TYPE.

LONG VideoSnapWithPath(VARIANT filePath);

Description：

According to the file path of the incoming to save the image to your hard disk, save the picture type specified by SetImageType，

（Ultimately saved the file name will automatically add .bmp or .jpg）

Parameters:

filePath：The path of the file，don't need to add the suffix。

void RButtonMenuEnable(VARIANT_BOOL bEnable);

Description：

Set whether to use the right click menu.

Parameters:

bEnable：FALSE，cancel the right-click menu；TRUE，Using the right-click menu.

void VideoAE(void);

Description：

Open automatic exposure process.

LONG VideoAWB(VARIANT_BOOL bReCountAWB, VARIANT_BOOL bAWB);

Description：

Set the white balance effect.（White balance will automatically save the last white balance effect）Only in the calculation of the white balance, load the white balance effect will be effective.

Parameters:

bReCountAWB：Recalculate the white balance.

FALSE，don't recalculate the white balance；TRUE，recalculate the white balance。

bAWB：Whether the loading is white balance effect.FALSE，don't act on white balance effect；TRUE，act on white balance effect.

void SetResolution(LONG index);

Description：

The resolution of the camera is set to the index of the specified resolution.

Parameters:

index: To specify the resolution index number. Serial number starting with 0.

LONG ResetPort(LONG nPort);

Description：

Reset camera port number

Parameters:

nPort：The port number specified.

LONG GetPort(LONG* nPort);

Description：

Get the current port number of the camera.

Parameters:

nPort：the port number of the camera.

LONG SetGainAll(LONG nGainA);

Description：

Set the camera overall gain.

Parameters:

nGainA：The overall gain value.

LONG GetGainAll(LONG* nGainA);

Description：

Get the camera overall gain.

Parameters:

nGainA：The overall gain value.

LONG SetGainR(LONG nGainR);

Description：

Set camera red gain.

Parameters:

nGainR：Red gain value.

LONG GetGainR(LONG* nGainR);

Description：

Get the camera red gain.

Parameters:

nGainR：Red gain value.

LONG SetGainG(LONG nGainG);

Description：

Set camera red gain.

Parameters:

nGainG：Green gain value.

LONG GetGainG(LONG* nGainG);

Description：

Get the camera green gain.

Parameters:

nGainR：Green gain value.

LONG SetGainB(LONG nGainB);

Description：

Set camera blue gain.

Parameters:

nGainB：Blue gain value.

LONG GetGainB(LONG* nGainB);

Description：

Get the camera blue gain.

Parameters:

nGainB：Blue gain value.

LONG SetContrast(LONG nContrast);

Description：

Set camera contrast.

Parameters:

nContrast：Contrast value.

LONG SetOffX(LONG nOffX);

Description：

The x offset of the image in the cutting mode.

Parameters:

nOffX：x offset.

LONG SetOffY(LONG nOffY);

Description：

The y offset of the image in the cutting mode.

Parameters:

nOffY：x offset.

LONG SetExpMode(LONG nExpMode);

Description：

To set the exposure mode.

Parameters:

nExpMode：Exposure mode，Description see Appendix:VTGE_EXP_MODE.

LONG SetCapMode(LONG nCapMode);

Description：

Set acquisition mode.

Parameters:

nExpMode：Acquisition mode，Description see Appendix:VTGE_CAP_MODE.

LONG SetFlipMode(LONG nFilpMode);

Description：

Set the flip mode.

Parameters:

nFilpMode：Flip mode，Description see Appendix:VTGE_FLIP_MODE

LONG SetZoomMode(LONG nZoomMode);

Description：

Set the zoom mode.

Parameters:

nZoomMode：Zoom mode，Description see Appendix:VTGE_ZOOM_MODE

LONG SetSpeedMode(LONG nSeepMode);

Description：

Set the speed mode.

Parameters:

nSeepMode：Speed mode，Description see Appendix:VTGE_SPEED_MODE

LONG SetColorMode(LONG nColorMode);

Description：

Set the color mode.

Parameters:

nColorMode：Color mode，Description see Appendix:VTGE_COLOR_MODE

LONG SetFlashState(VARIANT_BOOL bEnable);

Description：

Set the flash status (active high).

Parameters:

bEnable：Whether or not open flash，TRUE，Open flash；FALSE，Flash off.

LONG SetTriggerMode(LONG nTriggerMode);

Description：

Set trigger mode.

Parameters:

nTriggerMode，Trigger mode，Description see Appendix:VTGE_TRIGGER_MODE

LONG SetPackageDelayTime(LONG nTime);

Description：

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

Parameters:

nTime，The delay time，0-65535。

LONG GetPackageDelayTime(LONG* nTime);
Description：
Get the delay time when the camera to send data packets , unit: 20ns.
Parameters:
nTime，The delay time.

LONG SetWatchdogState(VARIANT_BOOL bEnable);
Description：
Set camera watchdog state.
Parameters:
bEnable，Whether to enable the watchdog.TRUE，Open；FALSE，Close.

LONG GetWatchdogState(VARIANT_BOOL* bEnable);
Description：
Get camera watchdog state.
Parameters:
bEnable，Whether to enable the watchdog.

LONG IOSet(LONG nIndex, LONG nValue);
Description：
To control the output state of the IO（The camera is provided with 3 IO output port）.
Parameters:
nIndex，Index range of IO port：0-2.
nValue，The IO state，Value：0 或 1.

LONG IOGet(LONG nIndex, LONG* nValue);
Description：
Get the camera state of input IO（The camera is provided with 2 IO input port）
Parameters:
nIndex，Index range of IO port：0-1
nValue，The IO state，Value：0 或 1

LONG SelectAviCompressor(void);
Description：
Select the video compressor（Click the pop-up window to cancel button to cancel the video compressor）

LONG VideoCaptureStart(VARIANT aviFilePath);
Description：
Open the video recording function.
Parameters:
aviFilePath，AVI Video storage path（The full path to the file）

LONG VideoCaptureStop(void);
Description：
Close the video recording function.

LONG IsVideoCapture(void);
Description：
Whether to open the video.
The return value:0，Close；1，Open.

LONG SetTriggerDelayTime(LONG nTime);
Description：
Set the delay acquisition time of the camera after the trigger.
Parameters:
nTime，The delay time, unit: 20ns

LONG ReadFrameToMemery(LONG pDestPtr, LONG* width, LONG* height, LONG* nBit);
Description：
Calling this method after the preview will read a frame data to the specified memory.
Parameters:
pDestPtr，To specify the first address of the saved image's data memory, need to apply for space;
Width，the width of the image data；
Height，the height of the image data；
nBit，the digits of image data （8,16 or 24），The data and set up consistent with the color mode；

LONG SoftTrigger();
Description：
Soft trigger gathering a frame of image, the method execution will trigger frame callback event.

LONG SetHDR(VARIANT_BOOL bEnable);
Description：
To open or close the high dynamic range function of the camera sesor.
Parameters:
bEnable, whether to open the high dynamic range.TRUE,Open;FALSE,Close.

LONG MCResolutionShow(VARIANT_BOOL bEnable);
Description：
The Preview settings are used custom tailoring resolution display.In the opening cut resolution display, preview, take pictures, the image width data recording and callback function in the all modified for cutting the size of the resolution.

Parameters:

bEnable, Identify whether to open the custom tailoring resolution.TRUE,Open;FALSE,Close.

LONG IsMCResolutionShow(VARIANT_BOOL *bEnable);

Description：

Gets whether open cut resolution display.

Parameters:

bEnable, Identify whether to open the custom tailoring resolution.TRUE,Open;FALSE,Close.

LONG SetMCResolution(LONG nOffX, LONG nOffY, LONG nWidth, LONG nHeight);

Description: Set the parameters of the cutting resolution display.

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),

Parameters:

nOffX，Set clipping resolution relative to the X offset the lower left corner of the current camera captured image.

nOffY，Set clipping resolution relative to the Y offset the lower left corner of the current camera captured image.

nWidth，Set clipping resolution width.

nHeight, Set clipping resolution height.

LONG GetMCResolution(LONG* nOffX, LONG* nOffY, LONG* nWidth, LONG* nHight);

Description: Get the relevant parameters about of cutting resolution display.

Parameters：

nOffX，Get clipping resolution relative to the X offset the lower left corner of the current camera captured image.

nOffY，Get clipping resolution relative to the Y offset the lower left corner of the current camera captured image.

nWidth，Get clipping resolution width.

nHeight，Get clipping resolution height.

LONG GetMCResOffSetMax(LONG* nOffXMax, LONG* nOffYMax);

Description: Maximum range gets cut resolution can be adjusted.

Parameters：

nOffXMax，Clipping resolution relative to the X offset the lower left corner of the current camera captured image.

nOffYMax，Clipping resolution relative to the Y offset the lower left corner of the current camera captured image.

LONG SetValidBitsOff(LONG nValid);

Description: 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.

Parameters：

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

LONG SetAEParam(VARIANT_BOOL bLiving, LONG nAELevel);

Description: To set the level of exposure and real time exposure.

Parameters：

bLiving, Whether to open the real time exposure.

nAELevel, the level of exposure. 0-high; 1- medium; 2-low.

LONG SetTrgFileFolder(VARIANT &strTrgFolder, VARIANT_BOOL bSave);

Description：Set the directory of the trigger memory images and whether to store image.

Parameters：

strTrgFolder，the directory of the store images.

bSave，whether to store image，true, store image，fales,Don't save the diagram。

LONG FrameToImage(VARIANT &strFileName, LONG pImageData, LONG width,
LONG height, LONG nColor, LONG nImgType);

Description：A frame of data storage for the image. The method can be used in frame callback or triggere callback.

Parameters：

strFileName，The path and the type of the storaged image .

pImageData，the first address of the image data.

width，The width of image data resolution.

height，The height of image data resolution.

nColor，Image data bits（8,16 or 24）.

nImgType，Specify the storage type of the image file，

Description see appendix:VTGE_IMG_TYPE。

Control event description

void OnDeviceFinishedAE (LONG* nExposure);

Description：To send events after the automatic exposure completed.

Parameters：

nExposure，To send the current exposure value after the automatic exposure completed.

void OnDeviceFinishedSnap(VARIANT* filePath);

Description：

To send events after the camera save images completed Parameters：

filePath，Back to the path of the saved image.

void OnDeviceFrameEvent(LONG* DataPtr, LONG* Width, LONG* Height, LONG* nBit);

Description：

Frame callback events, each frame of data calls this event, you can operate on the display of

image data.

Parameters：

DataPtr：The first address of the image data.

Width：The width of image data resolution.

Height：The height of image data resolution.

nBit：Image data bits（8,16 or 24）.

void OnDeviceTrgFrameEvent(LONG* DataPtr, LONG* Width, LONG* Height, LONG* nBit);

Description：

The trigger data frame callback event. Please return as soon as possible.

Parameters：

DataPtr：The first address of the image data.

Width：The width of image data resolution.

Height：The height of image data resolution.

nBit：Image data bits（8,16 or 24）.

Appendix

```
// The camera model
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
}


// Flip mode
enum VTGE_FLIP_MODE
{
    VTGE_FLIP_NORMAL        = 0,    // Normal
    VTGE_FLIP_HORIZONTAL    = 1,    // Flip Horizontal
    VTGE_FLIP_VERTICAL      = 2,    // Flip vertical
    VTGE_FLIP_ROTATE        = 3,    // Flip 270 °
    VTGE_FLIP_LEFT90        = 4,
    VTGE_FLIP_RIGHT90       = 5
};


// Zoom mode
enum VTGE_ZOOM_MODE
{
    VTGE_ZOOM_1_1 = 0, // No scaling, the effect of cutting
    VTGE_ZOOM_1_2 = 1, // The narrow double ranks. Such as *1536 image reduction, will
                       //become the *768 image
    VTGE_ZOOM_1_4 = 2  // The ranks of the reduced two times. Such as *1536 image
                       //reduction, will become the *384 image
};
```

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


// Exposure mode
enum VTGE_EXP_MODE
{
    VTGE_EXP_SYNC = 0,      // Synchronous
    VTGE_EXP_ASYN = 1       // Asynchronous
};


// Acquisition mode
enum VTGE_CAP_MODE
{
VTGE_CAP_TRIG = 0, // The trigger, trigger mode.（CCD and 36M CMOS open the video，A
                   //trigger signal, the camera will transmit a picture）
VTGE_CAP_SEQU = 1,     // Sequence, flow pattern. Open the video, the camera does not stop to
                   //upload data.
    VTGE_CAP_ANTI = 2 //（CCD and 36M CMOS）Resistance flash mode.
};


// Collection rate
enum VTGE_SPEED_MODE
{
    VTGE_SPEED_1_1 = 0,     //      100%   High speed mode
    VTGE_SPEED_1_2 = 1,     //       50%   Medium speed mode
    VTGE_SPEED_1_4 = 2,     //       25%   Low speed mode
    VTGE_SPEED_1_8 = 3 //     12.5%    Not used
};


// Color mode
enum VTGE_COLOR_MODE {
    VTGE_COLOR_RGB24,           //RGB24 bit true color image
    VTGE_COLOR_BW24,            //24 bit Black and white images   R ＝G ＝B
    VTGE_COLOR_GRAY,            //8 bit Gray scale image
```

```
    VTGE_COLOR_RAWDATA,             //8 bit Original data
     VTGE_COLOR_MONO12
     VTGE_COLOR_XWJ              // Microscope mode（Color correction）
};

// Trigger mode
enum VTGE_TRIGGER_MODE
{
     VTGE_TRIGGER_DISABLE = 0,     // Disable trigger.
     VTGE_TRIGGER_RISE    = 1,     // Rising edge trigger
     VTGE_TRIGGER_DOWN    = 2      // Falling edge trigger
};
```