USB 2.0 camera controls OCX directions for use

Control method

void VideoLiveStart(void);
Description:
Open dynamic preview.

void VideoLiveStop(void);
Description:
Closed dynamic preview.

void VideoAE(void);
Description:
Open the AE process.

void VideoAWB(void);
Description:
Open the AWB process.

void VideoCaptureStart(void);
Description:
Start video. Generate *. avi files to time named calls the control in applications directory.

void VideoCaptureStop(void);
Description:
Stop video.

void VideoShowPropertyPage(void);
Description:
Open the parameters adjustment property page.

void VideoSnap(void);
Description:
Acquisition of a frame image and save for pictures. Documents in time named in the application of calls the control program directory.

void SetBW(void);
Description:

The first calls will be converted to black and white mode, camera called again will convert color mode camera.

void SetZoom(void);
Description:

The first call will camera settings for scaling mode, called again will camera settings for cutting mode.

void SetImageType(LONG dwType);
Description:

Set the image file types preserved collection.
Parameters:

dwType: Designated picture file types. 0 for *. bmp file, 1 for *. JPG

LONG IsVideoLive(void);
Description:

Get the camera is on the state of the dynamic preview. If open the dynamic preview, return 1, otherwise return zero.

LONG IsVideoCapture(void);
Description:

For if the state of the video camera open. If open the video, return 1, otherwise return zero.

LONG IsBW(void);
Description:

For if the camera is set to black and white and display mode. If the camera for black and white and the display mode, return 1, otherwise return zero.

LONG IsZoom(void);
Description:

For if the camera is set to zoom mode. If the camera to zoom mode, return 1., or return zero.

void SetCrossLine(void);
Description:

Open graticule marker shows, if called again, shut down that graticule marker.

LONG IsVideoCrossLine(void);

Description:

Get the camera is on display, if open the graticule marker graticule marker display, return 1, otherwise return zero.

void SetResolution(LONG index);

Description:

The resolution of the camera is set to the index number of designated resolution.

Parameters:

index: Designated resolution the index number.

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

Description:

Get resolution information.

Parameters:

Count: The acquisition of resolution, the parameters of information used to specify the resolution of the need to acquire the index number.

When need to acquire resolution, the parameters of the number of the current camera resolution of support from the number.

pWidth: If the **Count** parameters specified a effective index, the parameters for receiving the index number. The width of the resolution of the specified parameters can be the **NULL**, this time the function through the parameters and return value to return to **Count** the number of resolution.

pHeight: If the **Count** parameters specified a effective index, the parameters for receiving the index number. The height of the resolution of the specified parameters can be the **NULL**, this time the function through the parameters and return value to return to **Count** the number of resolution.

FLOAT GetFrameRate(void);

Description:

Get the camera at current time frame rate

LONG GetDeviceCount(void);

Description:

To get the system of the number of available in the camera.

void GetDeviceName(LONG DeviceIdx, VARIANT* deviceName);

Description:

For the name of the specified index number camera.

Parameters:

DeviceIdx: The index number of designated camera.

deviceName: Return to index number specified camera name.

void SelectDevice(LONG deviceIdx);

Description:

Choose the index number of designated for the current camera camera.

Parameters

deviceIdx: The index number of designated camera.

void SetFullScreen(LONG bFullScreen);

Description:

The Settings control whether all the control size image preview filling.

This method calls in SelectDevice every time, and in VideoLiveStart after before to take effect.

Parameters

bFullScreen:0, Scaling the display mode;1, The control filling mode.

LONG VideoSnapWithPath(VARIANT &filePath);

Description:

According to the specified storage path take pictures, call this function call SetImageType function need before specified to save the image types, do the file name and image types are consistent.

Parameters

filePath, Images of the need to save the path to the file

返回值

0, saved successfully; The return value of the other that did not call is successful

void RefScreen(LONG bFullScreen);

Description:

Open preview, display control size change the method used to call, refresh preview interface

Parameters

bFullScreen:0, Scaling the display mode;1, The control filling mode

void SetDeviceParma (LONG nType, LONG nValue);

Description:

The method of setting the camera parameters

Parameters:

nType: Set type， See appendix enumeration type ENUM_PARMA。

nValue，Set the value of the parameter

LONG SetTriggerMode(VARIANT_BOOL bTrigger);
Description：
Set the camera's state trigger (open trigger camera in a trigger after acquisition state)
Parameters：
bTrigger：Whether open state trigger. TRUE, open；FALSE, close。

LONG SoftTrigger(void);
Description:
Soft trigger acquisition image, need VT_SetTriggerMode open trigger collection mode application.
CMOS camera will need to hardware IO output and trigger connected.
CCD camera does not need hardware connection.

LONG ReadFrameToMemery(LONG pDestPtr, LONG* width, LONG* height, LONG* nBit);
Description：
Call this method in open preview will read a frame data to specified memory.
Parameters：
pDestPtr，Designated save image data memory of the first address, need to apply on their own space；
Width，The image data from the wide；
Height，The image data from the height；
nBit，The image data from the bit (8 or 24), the data and the color of the corresponding set mode；

LONG ReadFrameToMemeryTransform(LONG pDestPtr, LONG* width, LONG* height,
LONG* nBit, VARIANT_BOOL bUpDown);
Description：
Call this method in open preview will read a frame data to the specified memory, can specify whether you to lie down to flip
Parameters：
pDestPtr，Designated save image data memory of the first address, need to apply on their own space；
Width，The image data from the wide；
Height，The image data from the height；
nBit，The image data from the bit (8 or 24), the data and the color of the corresponding set mode；
bUpDown,TRUE, turned up and down; FALSE, don't flip

LONG SetTriggerDelayTime(LONG nTime);
Description：

The method of setting the camera trigger delay time. At present in the CCD camera in the new firmware support. In the CMOS camera VT-F300CF, VT-F130MF, VT-F500CF in support.

Parameters：

nTime, trigger delay time, unit: ms

void SetShowVideo(LONG bShow);

Description：

This method simply stop or open video display, data collection action not stop (acquisition thread continue interpolation, etc.) operations

Parameters：

bShow, Open or closed video preview

void VideoPause(LONG bPause);

Description：

The method to stop or open video preview, (acquisition thread hung, does not take up the CPU)

Parameters：

bPause, Open or closed video preview

void SetHDR(LONG bHDR);

Description：

This method is open or closed high dynamic range effect.（At present support color camera and VT36M）

Parameters：

bHDR, Open or closed HDR。

void VideoSnapZoom(void);

Description：

Will a frame image save for the wide high half the size of file.

Documents in time named in the application of calls the control program directory. (currently only support CMOS series)

LONG VideoSnapZoomWithPath(VARIANT &filePath);

Description：

According to the specified storage path , will a frame image save for the wide high half the size of file.

Documents in time named in the application of calls the control program directory. (currently only support CMOS series)

Parameters:

filePath, images of the need to save the path to the file

Return value:

0, Saved successfully; The return value of the other that did not call is successful

void EEProm_Set(BYTE* pData);

Description：

Set store the data. To VT-F300CF, VT-F130MF, VT-F500CF have 40 bytes is available, to VT-F130C, VT-F300C, VT-F500C, VT36M have 240 bytes is available.

Parameters：

pData，Set data content

void EEProm_Get(BYTE* pData);

Description：

Get store the data. To VT-F300CF, VT-F130MF, VT-F500CF have 40 bytes is available, to VT-F130C, VT-F300C, VT-F500C, VT36M have 240 bytes is available.

Parameters：

pData，Get data content

void EEProm_SetByte(LONG nIndex, LONG nCount, BYTE* pData);

Description：

Set a fixed position of fixed size data. To VT-F300CF, VT-F130MF, VT-F500CF have 40 bytes is available, to VT-F130C, VT-F300C, VT-F500C, VT36M have 240 bytes is available.

Parameters：

nIndex，Start bytes, value range is 0-39/239

nCount，Data size, value range is 1-40/240, nIndex + nCount < = 40/240

pData，data

void EEProm_GetByte(LONG nIndex, LONG nCount, BYTE* pData);

Description：

Get a fixed position of fixed size data. To VT-F300CF, VT-F130MF, VT-F500CF have 40 bytes is available, to VT-F130C, VT-F300C, VT-F500C, VT36M have 240 bytes is available.

Parameters：

nIndex，Start bytes, value range is 0-39/239

nCount，Data size, value range is 1-40/240, nIndex + nCount < = 40/240

pData，data

void SetCustomIO(BYTE data);

Description：

Set the user defined IO. At present, only for VT-F300CF, VT-F130MF, VT-F500CF effective

Parameters：

data，0 low level, 1 high level

void SetTriggerSource(BYTE data);

Description：

Set the camera that can trigger source. At present, only for VT-F300CF, VT-F130MF, VT-F500CF effective

Parameters：

data，Control, 0 close，1　open

void SetTriggerSourceTime(LONG Time);

Description：

Set the camera trigger source time interval. At present, only for VT-F300CF, VT-F130MF, VT-F500CF effective.

Parameters：

Time，time, unit: ms

void SetHBlank(LONG nHBlank);

Description：

Set the camera line blank.

VT-F300CF range (142-1023), default 142

VT-F130MF range (9-1023) , default 9

VT-F500CF range (0-1023) , default 0

VT-F130C range (25-2000), default 25

VT-F300C range (19-2000), default 142

VT-F500C range (0-2000), default 0

VT36M range (43-1023), default 94

Parameters：

nHBlank，line blank value

void SetVBlank(LONG nVBlank);

Description：

Set the camera frame blank

VT-F300CF、VT-F130MF、VT-F500CF range (25-1023), default 25

VT-F130C range (25-2000), default 25

VT-F300C range (3-2000), default25

VT-F500C range (25-2000), default 25

VT36M range (4-1023), default 45

Parameters：

nVBlank，frame blank value

void SetAEParam(LONG bAE);

Description：

Set up automatic exposure parameters, real-time automatic exposure

Parameters：

bLiving，Whether open real-time automatic exposure

AELevel，Exposure levels

pfAECallback，Exposure callback function

lpContext，The user context

void SetExpouseDelayTime(LONG nTime);

Description：

Set the exposure time delay. Currently only support VT-F130C, VT-F300C, VT36M, VT-F500C.

Parameters：

nTime，the exposure time delay，range 0-2047

void GetNewCalibrationRawFrame(ULONG lHeight_In, ULONG lWidth_In,
LONG iColourType_In, LONG iRawDateType_In, BYTE cTotalGain_In,
LONG iContrastValue_In, BYTE* pRawFrame_In, BYTE cNewTotalGain_In,
LONG iNewContrastValue_In, BYTE* pRawFrame_Out);

Description：

According to primitive calibration image generation new calibration image, function is set up according to the new gain and contrast, by the original image generation new calibration image, the new calibration image is still as the primary data format (bit). Currently only support CMOS series.

Parameters：

lHeight_In，Data acquisition height

lWidth_In，Data acquisition width

iColourType_In，Reserves

iRawDateType_In，Reserves

cTotalGain_In，Reserves

iContrastValue_In，Reserves

pRawFrame_In，Need to convert image data pointer

cNewTotalGain_In，After converting the brightness of the image data

iNewContrastValue_In，After converting image data contrast

pRawFrame_Out，After converting the image of the pointer

void RawFrameToRgb(ULONG lHeight_In, ULONG lWidth_In, ULONG iColourType_In,
ULONG iRawDateType_In, BYTE* pRawFrame_In, BYTE* pRgbData_Out);

Description：

The original data to create BMP image data. This method is currently only support VT-F500CF, old VT-F500CF original image sequence (BGR), for new VT-F500CF for (GRB).

Parameters：

lHeight_In，Data acquisition height

lWidth_In， Data acquisition width

iColourType_In，Reserves

iRawDateType_In，Reserves

pRawFrame_In，A pointer to the original data

pRgbData_Out，The output of the image data area RGB pointer

void SetLight(BYTE cTotalGain_In);

Description：

Set the brightness values of output image (currently only support CMOS series).

Parameters：

cTotalGain_In，the brightness values of output image

void GetLight(BYTE* cTotalGain_Out);

Description：

Get the brightness values of output image (currently only support CMOS series)。

Parameters：

cTotalGain_Out，Return to the brightness values of current output image

void MCResolutionShow(LONG nEnable);

Description：

Set up if application custom cut resolution show preview. In the display resolution after cutting open the window, preview images to display the long image data original image data is specified in the location of the image (currently only support CMOS series).

Parameters：

nEnable，TRUE, open; FALSE, close.

void IsMCResolutionShow(LONG* nEnable);

Description：

Get whether open cuting resolution display (currently only support CMOS series)。

Parameters：

nEnable，TRUE, open; FALSE, close.

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

Description：

Setting cutting resolution display related parameters. In the application of the set the current

function setting resolution after resolution or change the color mode, cutting the migration will resume for resolution, size will set to the current setting and the resolution of the same. In the set please ensure that data set the legitimacy of the nOffX + nWidth < = : (the current camera Settings of the resolution of the wide), nOffY + nHeight < = (the current camera Settings of the resolution of the high), nWidth and nHeight try to set to integer times. Coordinates with the origin in the callback function of image data obtained in the coordinates of origin is consistent(currently only support CMOS series).

Parameters：

nOffX，Setting cutting resolution relative to the current camera to the bottom left corner of the image collection of X offset

nOffY，Setting cutting resolution relative to the current camera to the bottom left corner of the image collection of Y offset

nWidth，Set the width of cutting resolution

nHeight，Set the height of cutting resolution

void GetMCResolution(LONG* nOffX, LONG* nOffY, LONG* nWidth, LONG* nHeight);

Description：

Getting cutting resolution display related parameters

Parameters：

nOffX，Getting cutting resolution relative to the current camera to the bottom left corner of the image collection of X offset

nOffY，Getting cutting resolution relative to the current camera to the bottom left corner of the image collection of Y offset

nWidth，Get the width of cutting resolution

nHeight，Get the height of cutting resolution

void SetAWBParam(VARIANT_BOOL bAWB, VARIANT_BOOL bReCount);

Description：

Open to be automatic white balance. currently only support CCD series.

Parameters：

bAWB，Whether loading white balance, TRUE for open, FALSE to shut down

bReCount，Whether a calculation white balance

void SetExpouseUnit(LONG nUnit);

Description：

Exposure unit set，default for line number

Parameters：

nUnit，Exposure unit, 0 for line number, 1 for exposure time , unit : us

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

Description：

Incoming data in memory is saved as an image, nColor parameters need to format the image data

to set specific 8-bit grayscale or 24-bit RGB images.
Parameters：
strFileName，To save the image name
pImageData，Save the image memory data pointer
width，Image width
height，Image height
nColor，Color bits (8 or 24)
ImgType，Image type, 0 to BMP, 1 to JPG

LONG SetTrgFileFolder(VARIANT &strTrgFolder, VARIANT_BOOL bSave);
Description：
Sets the trigger mode is stored map, and store the picture folder.
Parameters：
strTrgFolder，CCD Series storage folder for pictures, CMOS series accurate to the file name.
bSave，Whether to save the picture

void SetCustomResolution(LONG nWidth, LONG nHeight);
Description：
Set Custom Resolution (only support CMOS series).
Parameters：
nWidth,Width.
nHeight,Height.

void SetTriggerCheckSignalTime(LONG nTime);
Description：
Set trigger testing time, currently only support 36M
Parameters：
nTime, testing time, units: ms.

void SetCameraLED(LONG nLED, LONG nLight);
Description：
Setting LED lights. This function needs hardware support.
Parameters：
nLED，serial number, 0-3
nLight，the state of LED lights. When nLED equal to 0, 1 or 2, nLight is equal to 0 lights off, equal to 127 lights on. When nLED equal to 3, nLight is equal to 1 lights off, equal to 0 lights on.

Control Event Description

void OnDeviceFinishedAE (LONG* nExposure);
Description:
After exposure in the camera automatically sends an event
Parameters

nExposure，Auto exposure incident came after the current value of the exposure

void OnDeviceFinishedAWB (LONG* nExposure, LONG* nR, LONG* nG, LONG* nB);
Description:
Auto white balance the camera to send after the event
Parameters
nExposure，Auto white balance after the incident came to the current exposure
nR，Auto white balance after the incident came to the current R value
nG，Auto white balance after the incident came to the current G value
nB，Auto white balance after the incident came to the current B value

void OnDeviceFinishedSnap(VARIANT* filePath);
Description:
After the camera to save the image to send an event
Parameters
filePath，Returns the path to save the picture

void OnDeviceTrgFrameEvent(LONG* DataPtr, LONG* Width, LONG* Height, LONG* nBit);
Description：
Data frame triggers a callback event, please return as soon as possible.
Parameters：
DataPtr：Image data of the first address
Width：Wide resolution image data
Height：High resolution image data
nBit：Image data bits (8 or 24), the data set corresponds to the color mode

void OnDeviceFrameEvent(LONG* DataPtr, LONG* Width, LONG* Height, LONG* nBit);
Description：
Frame callback event, please return as soon as possible.
Parameters：
DataPtr：Image data of the first address
Width：Wide resolution image data
Height：High resolution image data
nBit：Image data bits (8 or 24), the data set corresponds to the color mode

void OnDeviceEvents(LONG* nEvent);
Description:
Equipment-related incident, there are currently two events, remove the device and insertion equipment

Parameters:
nEvent, 0 - device removal; 1 - device into.

Sample code (VC)

```
    // Camera initialization
    CVTCamera    m_cam;                                  // Camera control
    // … …
    // The number of available devices
    long nCount = m_cam.GetDeviceCount();
    if(nCount<=0)
    {
        // Equipment not available
        return TRUE;
    }
    VARIANT varName;
    int i;
    for(i=0; i<nCount; i++)                          // Obtain the names of all available equipment.
    {
        m_cam.GetDeviceName(i, &varName);
        CString strName(varName.bstrVal);
                    // Device name can be used to update the user interface
    }
    m_cam.SelectDevice(0);                                  //  Select the 0 devices.
m_cam. SetFullScreen(1);                              // Fill mode set to full control

    //  Get the current number of camera resolution
    LONG nResoCount = -1;
    nResoCount = m_cam.GetResolutions(&nResoCount, NULL, NULL);

    LONG Width, Height;
    for(int j=0; j<nResoCount; j++)
    {
        LONG nCount = j;
        m_cam.GetResolutions(&nCount, &Width, &Height);
                        // Resolution information obtained by the index number
        CString str;
        str.Format(_T("%d * %d"), Width, Height);
                        // String can be used to update the user interface
    }
    m_cam.SetResolution(0);                                 // Select the first resolution of 0
    m_cam.VideoLiveStart();                                 // Open the Dynamic Preview
.
```

Appendix

```
typedef enum
{
     idExposure         = 0,      //Exposure (1-5000)
     idContrast         = 1,      //Contrast (1-32)
     idGamma            = 2,      //Gamma (1-32)
     idGain_R           = 3,      //Red gain (CMOS:1-6000; CCD:1-63)
     idGain_G           = 4,      //Green gain (CMOS:1-6000; CCD:1-63)
     idGain_B           = 5,      //Blue gain (CMOS:1-6000; CCD:1-63)
     idFlipMode         = 6,      //Flip Mode (0-4) Enum: FLIP_MODE
     idColorMode        = 7,      //Color Mode (0-5) Enum: COLOR_MODE
     idCrossLine        = 8,      // Crosshairs on or off (0,off; 1,on)
     idCrossLine_X      = 9,      // Vertical cross-line x position (0- width)
     idCrossLine_Y      = 10,     // Y position of the cross-line horizontal (0-height)
     idCrossLine_Center = 11,// Crosshairs center (0, forced center; 1,depends on the value XY)

     //////////////
     //CMOS Camera
     idColorAdjust      = 101,        // Color correction mode (0-2) Enum: COLOR_ADJUST_TYPE
     idGain_All         = 102,        // Overall gain (1-63)
     idZoom             = 103,        // Zoom mode (0, Cutting mode; 1, Zoom mode)
     idOffsetX          = 104,        // Cutting mode, the image offset in x direction

     idOffsetY          = 105,        // Cutting mode, the image offset in y direction

     //////////////
     //CCD Camera
     idAD               = 201,        // AD gain (1-255)
     idValidBits        = 202         // Set data valid bit (0-1) Enum: CCD_VALID_BIT
}ENUM_PARMA;

enum COLOR_ADJUST_TYPE
{
     COLOR_ADJUST_NO,
     COLOR_ADJUST_COLD, // Cool priority
     COLOR_ADJUST_WARM  // Warm priority

};

// CCD-bit data is valid data, the mode determines the need to use the bit data, use of valid
data bits
enum CCD_VALID_BIT
{
     CCD_VALID_HIGH,         // High effective
     CCD_VALID_LOW           // Low effective
};

enum COLOR_MODE
{
     COLOR_RGB24,
     COLOR_RGB24_CLEAR,
     COLOR_BW24,
     COLOR_GRAY,
     COLOR_RAWDATA,
```

```
    COLOR_XWJ     // Microscope mode, currently only supports color camera
};
enum FLIP_MODE
{
    FLIP_NATURAL,           // Normal display
    FLIP_LEFTRIGHT,         // Flip Horizontal

    FLIP_UPDOWN,            // Flip Vertical
    FLIP_ROTATE180          // Rotate 180 degrees
};
```