ASICamera2 Software Development Kit

Revision:1, 2
2016.9.8

Table of Contents

1 Introduction

This SDK is used to operate ASI serial cameras, can be used by C, C++, C# and other develop tools, is suit for Windows, Linux, OSX operating system of x86 and x64.

Header file: ASICamera2.h

Under Windows the import library and dynamic library: ASICamera2.lib、ASICamera2.dll

Under Linux the dynamic library and static library: ASICamera2.so、ASICamera2.a

Under OSX the dynamic library and static library: ASICamera2.dylib、ASICamera2.a

Installation method:

Under Windows, extract the downloaded zip file to any directory, and add DLL's path to system environment variables, sometimes logout and re-login is required

2 Definition of enum-type and struct

```
2.1 typedef enum ASI_BAYER_PATTERN
{
    ASI_BAYER_RG=0,
    ASI_BAYER_BG,
    ASI_BAYER_GR,
    ASI_BAYER_GB
}ASI_BAYER_PATTERN;
```

Bayer filter type

```
2.2 typedef enum ASI_IMG_TYPE
{
    ASI_IMG_RAW8 = 0,// 1 byte every pixel
    ASI_IMG_RGB24,// Each pixel consists of RGB, 3 bytes totally (color cameras only)
    ASI_IMG_RAW16,// 2 byte every pixel
    ASI_IMG_Y8,// mono mode，1 byte every pixel (color cameras only)
    ASI_IMG_END = -1

}ASI_IMG_TYPE;
```

Image type

```
2.3 typedef enum ASI_GUIDE_DIRECTION
{
    ASI_GUIDE_NORTH=0,
    ASI_GUIDE_SOUTH,
    ASI_GUIDE_EAST,
    ASI_GUIDE_WEST
}ASI_GUIDE_DIRECTION;
```

Moving direction when guiding

```
2.4 typedef enum ASI_FLIP_STATUS
  {
    ASI_FLIP_NONE = 0,// none flip
    ASI_FLIP_HORIZ,// horizontal flip
    ASI_FLIP_VERT,// vertical flip
    ASI_FLIP_BOTH,// horizontal + vertical flip

}ASI_FLIP_STATUS;
```

Image flip

2.5 typedef enum ASI_ERROR_CODE

```
{
	ASI_SUCCESS,//operate successfully
	ASI_ERROR_INVALID_INDEX,//invalid camera index
	ASI_ERROR_INVALID_ID,//invalid camera ID
	ASI_ERROR_INVALID_CONTROL_TYPE,//invalid control type
	ASI_ERROR_CAMERA_CLOSED,//camera isn't opened
	ASI_ERROR_CAMERA_REMOVED,//can't find camera, maybe is removed
	ASI_ERROR_INVALID_PATH,//can't find the file
	ASI_ERROR_INVALID_FILEFORMAT,//invalid file format
	ASI_ERROR_INVALID_SIZE,//invalid image size
	ASI_ERROR_INVALID_IMGTYPE,//invalid image type
	ASI_ERROR_OUTOF_BOUNDARY,//the start coordinate is out of boundary
	ASI_ERROR_TIMEOUT,//time out
	ASI_ERROR_INVALID_SENQUENCE,//invalid operate sequence, for example set format when
video capture
	ASI_ERROR_BUFFER_TOO_SMALL,//image buffer isn't big enough
	ASI_ERROR_VIDEO_MODE_ACTIVE,//video capture is working，can't snap
	ASI_ERROR_EXPOSURE_IN_PROGRESS,//snap is working，can't capture video
	ASI_ERROR_GENERAL_ERROR,//other error
	ASI_ERROR_END
}ASI_ERROR_CODE;
```

Returned error code

2.5 typedef enum ASI_BOOL

```
{
	ASI_FALSE =0,
	ASI_TRUE
}ASI_BOOL;
```

True or false

2.7 typedef struct _ASI_CAMERA_INFO

```
{
	char Name[64]; //camera name, can be displayed on UI
	int CameraID; //camera ID, use it to operate special camera
	long MaxHeight; //maximum image height
	long MaxWidth;	// maximum image width
	ASI_BOOL IsColorCam; //is color camera?
	ASI_BAYER_PATTERN BayerPattern;//Bayer filter type
	int SupportedBins[16]; //array consisted of supported bin value, end with 0
	ASI_IMG_TYPE SupportedVideoFormat[8];// array consisted of supported image type, end with
ASI_IMG_END
	double PixelSize; //pixel size(um)
	ASI_BOOL MechanicalShutter;// is mechanical shutter supported
	ASI_BOOL ST4Port;//is there ST4 port
	ASI_BOOL IsCoolerCam;//whether camera have cooler
	ASI_BOOL IsUSB3Host;//is working under USB3?
	ASI_BOOL IsUSB3Camera;//is USB3 camera?
	float ElecPerADU;//system gain
	int OffsetLGain;
	int OffsetHGain;
	char Unused[16];
} ASI_CAMERA_INFO;
```

Camera information

2.8 typedef enum ASI_CONTROL_TYPE

```
{
    ASI_GAIN = 0,//gain
    ASI_EXPOSURE,//exposure time(us)
    ASI_GAMMA,//gamma
    ASI_WB_R,//red component of white balance
    ASI_WB_B,// blue component of white balance
    ASI_BRIGHTNESS,//offset
    ASI_BANDWIDTHOVERLOAD,//USB band width
    ASI_OVERCLOCK,//over clock
    ASI_TEMPERATURE,// sensor temperature，10 times the actual temperature
    ASI_FLIP,//image flip
    ASI_AUTO_MAX_GAIN,//maximum gain when auto adjust
    ASI_AUTO_MAX_EXP,//maximum exposure time when auto adjust，unit is second
    ASI_AUTO_MAX_BRIGHTNESS,//target brightness when auto adjust
    ASI_HARDWARE_BIN,//hardware bin
    ASI_HIGH_SPEED_MODE,//high speed mode
    ASI_COOLER_POWER_PERC,//cooler power percent(only cool camera)
    ASI_TARGET_TEMP,//sensor's target temperature(only cool camera)，don't multiply by 10
    ASI_COOLER_ON//open cooler(only cool camera)
    ASI_MONO_BIN,//lead to less grid at software bin mode for color camera
    ASI_FAN_ON
}ASI_CONTROL_TYPE;
```

Camera control type

2.9 typedef struct _ASI_CONTROL_CAPS

```
{
    char Name[64]; /control type name，like "Gain" "Exposure"...
    char Description[128]; //description
    long MaxValue;//maximum value
    long MinValue;//minimum value
    long DefaultValue;//default value
    ASI_BOOL IsAutoSupported; //is auto adjust supported?
    ASI_BOOL IsWritable; //can be writed, for example sensor temperature can't be modified
    ASI_CONTROL_TYPE ControlType;//control type ID
    char Unused[32];
} ASI_CONTROL_CAPS;
```

Capacity of control type

note：maximum and minimum value of ASI_TEMPERATURE is multiplied by 10

2.10 typedef enum ASI_EXPOSURE_STATUS

```
  {
    ASI_EXP_IDLE = 0,//idle, ready to start exposure
    ASI_EXP_WORKING,//exposuring
    ASI_EXP_SUCCESS,// exposure successfully, image can be read out
    ASI_EXP_FAILED,// exposure fail, require restart exposure
}ASI_EXPOSURE_STATUS;
```

Used under snap mode to describe exposure status

2.11 typedef struct _ASI_ID

```
{
    unsigned char id[8];
}ASI_ID;
```

ID to be write into camera flash, 8 bytes totally

3 Function declaration

3.1 ASIGetNumOfConnectedCameras

Syntax：int ASIGetNumOfConnectedCameras();

Usage：get the count of connected cameras

3.2 ASIGetCameraProperty

Syntax：ASI_ERROR_CODE ASIGetCameraProperty(ASI_CAMERA_INFO *pASICameraInfo, int iCameraIndex);

Usage：get camera's information of special index(0 is the first one)

Description：

ASI_CAMERA_INFO *pASICameraInfo：pointer to camera info struct

int iCameraIndex：camera index

example code：

```
int iNumofConnectCameras = ASIGetNumOfConnectedCameras();
ASI_CAMERA_INFO **ppASICameraInfo = (ASI_CAMERA_INFO
*)malloc(sizeof(ASI_CAMERA_INFO *)*iNumofConnectCameras);
for(int i = 0; i < iNumofConnectCameras; i++)
    ASIGetCameraProperty(pASICameraInfo[i], i);
```

Notes：

Camera name can be get before camera is opened

3.3 ASIOpenCamera

Syntax：ASI_ERROR_CODE ASIOpenCamera(int iCameraID);

Usage：open camera of special ID

3.4 ASICloseCamera

Syntax：ASI_ERROR_CODE ASICloseCamera(int iCameraID);

Usage：close camera then resource will be released

3.5 ASIGetNumOfControls

Syntax：ASI_ERROR_CODE ASIGetNumOfControls(int iCameraID, int * piNumberOfControls);

Usage：get the count of control type

3.6 ASIGetControlCaps

Syntax：ASI_ERROR_CODE ASIGetControlCaps(int iCameraID, int iControlIndex, ASI_CONTROL_CAPS * pControlCaps);

Usage：get control type's capacity of special index

Description：

int iCameraID：camera ID

int iControlIndex：control index

ASI_CONTROL_CAPS * pControlCaps：pointer to control capacity

Notes: iControlIndex is control index, is different from ControlType

3.7 ASIGetControlValue

Syntax：ASI_ERROR_CODE ASIGetControlValue (int iCameraID, ASI_CONTROL_TYPE ControlType, long *plValue, ASI_BOOL *pbAuto);
Usage：get control's value
Description：
    int iCameraID：camera ID
    ASI_CONTROL_TYPE ControlType：control type
    long *plValue：pointer to value
    ASI_BOOL *pbAuto：whether the control is auto adjusted

3.8 ASISetControlValue
Syntax：ASI_ERROR_CODE ASISetControlValue(int iCameraID, ASI_CONTROL_TYPE ControlType, long lValue, ASI_BOOL bAuto);
Usage：set control's value
Description：
    int iCameraID：camera ID
    ASI_CONTROL_TYPE ControlType：control type
    long lValue：control value
    ASI_BOOL bAuto：whether the control is auto adjusted
Notes：when set to auto adjust(bAuto=ASI_TRUE)，lValue should be current value

3.9 ASISetROIFormat
Syntax：ASI_ERROR_CODE ASISetROIFormat(int iCameraID, int iWidth, int iHeight, int iBin, ASI_IMG_TYPE Img_type);
Usage：set ROI size and image type
Description：
    int iCameraID：camera ID
    int iWidth：image width
    int iHeight：image height
    int iBin：bin value
    ASI_IMG_TYPE Img_type：image type
Return：success or error code

Notes：make sure iWidth%8=0，iHeight%2=0. For USB2.0 camera ASI120，make sure iWidth* iHeight%1024=0，otherwise setting will be failed.

3.10 ASIGetROIFormat
Syntax：ASI_ERROR_CODE ASIGetROIFormat(int iCameraID, int *piWidth, int *piHeight, int *piBin, ASI_IMG_TYPE *pImg_type);
Usage：get ROI arguments
Description：
    int iCameraID：camera ID
    int *piWidth：image width
    int *piHeight：image height
    int *piBin：bin value
    ASI_IMG_TYPE *pImg_type：image type

3.11 ASISetStartPos

Syntax：ASI_ERROR_CODE ASISetStartPos(int iCameraID, int iStartX, int iStartY);
Usage：set start position of ROI
Description：
int iCameraID：camera ID
int iStartX：start position of x-axis
int iStartY：start position of y-axis
Notes：the position is relative to the image after binning. call this function to change ROI area to the origin after ASISetROIFormat, because ASISetROIFormat will change ROI to the center.

3.12 ASIGetStartPos

Syntax：ASI_ERROR_CODE ASIGetStartPos(int iCameraID, int *piStartX, int *piStartX);
Usage：get start position of ROI
Description：
int iCameraID：camera ID
int *piStartX：start position of x-axis
int *piStartX：start position of y-axis
Notes：the position is relative to the image after binning.

3.13 ASIGetDroppedFrames

Syntax：ASI_ERROR_CODE ASIGetDroppedFrames(int iCameraID,int *piDropFrames);
Usage：get dropped frames' count during video capture

3.14 ASIEnableDarkSubtract

Syntax：ASI_ERROR_CODE ASIEnableDarkSubtract(int iCameraID, char *pcBMPPath);
Usage：enable dark subtract function
Description：
int iCameraID：camera ID
char * pcBMPPath：path of dark field image(.bmp)
Return: success or error code
Notes：dark field image is get by camera's direct show driver，located in capture application's menu "video capture filter"->"ROI and others" table

3.15 ASIDisableDarkSubtract

Syntax：ASI_ERROR_CODE ASIDisableDarkSubtract(int iCameraID);
Usage：disable dark subtract function

3.16 ASIStartVideoCapture

Syntax：ASI_ERROR_CODE ASIStartVideoCapture(int iCameraID);
Usage：start video capture

3.17 ASIStopVideoCapture

Syntax：ASI_ERROR_CODE ASIStopVideoCapture(int iCameraID);
Usage：stop video capture

3.18 ASIGetVideoData

Syntax：ASI_ERROR_CODE ASIGetVideoData(int iCameraID, unsigned char* pBuffer, long lBuffSize, int iWaitms);
Usage：after ASIStartVideoCapture ()，call this function to get image continuously
Description：

unsigned char* pBuffer：pointer to image buffer
long lBuffSize：size of buffer
int iWaitms： wait time, unit is ms, -1 means wait forever

Notes：
If read out speed isn't fast enough, the frame will be discard
bufSize Byte length：for RAW8 and Y8，bufSize >= image_width*image_height，for RAW16，bufSize >= image_width*image_height *2，for RGB8，bufSiz >= image_width*image_height *3
suggested iWaitms value: exposure_time*2

3.19 ASIPulseGuideOn
Syntax：ASI_ERROR_CODE ASIPulseGuideOn(int iCameraID, ASI_GUIDE_DIRECTION direction);
Usage：send ST4 guiding pulse，start guiding，only the camera with ST4 port support
Notes：ASIPulseGuideOff must be called to stop guiding

3.20 ASIPulseGuideOff
Syntax：ASI_ERROR_CODE ASIPulseGuideOff(int iCameraID, ASI_GUIDE_DIRECTION direction);
Usage：send ST4 guiding pulse，stop guiding，only the camera with ST4 port support

3.21 ASIStartExposure
Syntax：ASI_ERROR_CODE ASIStartExposure(int iCameraID);
Usage：start snap

3.22 ASIStopExposure
Syntax：ASI_ERROR_CODE ASIStopExposure(int iCameraID);
Usage：stop snap
Notes：if exposure status is success after stop exposure，image can still be read out

3.23 ASIGetExpStatus
Syntax：ASI_ERROR_CODE ASIGetExpStatus(int iCameraID, ASI_EXPOSURE_STATUS *pExpStatus);
Usage：get snap status
Notes：after snap is started，the status should be checked continuously

3.24 ASIGetDataAfterExp
Syntax：ASI_ERROR_CODE ASIGetDataAfterExp(int iCameraID, unsigned char* pBuffer, long lBuffSize);
Usage：get image after snap successfully
Description：
int iCameraID：camera ID
unsigned char* pBuffer：pointer to image buffer
long lBuffSize：size of buffer
Notes：lBuffSize refer to ASIGetVideoData ()

3.25 ASIGetID
Syntax：ASI_ERROR_CODE ASIGetID(int iCameraID, ASI_ID* pID);
Usage：get camera id stored in flash, only available for USB3.0 camera

3.26 ASISetID

Syntax：ASI_ERROR_CODE ASISetID(int iCameraID, ASI_ID ID);
Usage：write camera id to flash, only available for USB3.0 camera

4 Suggested call sequence

4.1 Initialization
Get count of connected cameras--> ASIGetNumOfConnectedCameras
Get cameras' name--> ASIGetCameraProperty
Open camera -->ASIOpenCamera（Notes： this SDK can operate multiple cameras，distinguish by CameraID）
Get count of control type--> ASIGetNumOfControls
Get capacity of every control type-->ASIGetControlCaps
Set image size and format-->ASISetROIFormat
Set start position when ROI-->ASISetStartPos

4.2 Get and set control value

```
    ASIGetControlValue
    ASISetControlValue
    allowed during capture
```

4.3 Capture image
There are two mode：video and snap mode. Images are captured continuously under video mode, and only single image is captured under snap mode.

- video mode

Start video capture-->ASIStartVideoCapture
Stop video capture-->ASIStopVideoCapture

It is suggested that get and save data in single thread:

```
while(1)
{
    ASIGetVideoData
    ...
}
```

- snap mode

```
ASIStartExposure

while(1)
{
    ASIGetExpStatus(,&status)
    ...
}
```

Cancel exposure：ASIStopExposure

```
if(status ==ASI_EXP_SUCCESS)//get image if snap successfully
    ASIGetDataAfterExp
```

4.4 Close camera
ASICloseCamera//release resource