ASICamera2 Software Development Kit

Revision:2, 3
2017. 5. 2

Table of Contents

Change History

Change date	revision	comment
2017.5.2	2.3	Correct description of ASIGetCameraProperty
2017.4.12	2.2	Edit content
2017.2.24	2.1	Add ASI_CONTROL_TYPE: ASI_AUTO_MAX_EXP_MS
2016.12.9	2.0	Add ASI_CONTROL_TYPE: ASI_ANTI_DEW_HEATER Add ASIGetProductIDs
2016.9.19	1.3	Add ASI_CONTROL_TYPE: ASI_PATTERN_ADJUS, etc Add ASIInitCamera

1 Introduction

This Software Development Kit (SDK) describes a set of functions that can be used to operate the ASI line of serial cameras, via C, C++, C# and other development tools, and is suitable to be run under Windows, Linux, and OSX operating systems for either x86 or x64. The “2” brings the added functionality over previous versions of handling multiple ASI cameras within the same application.

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 the DLL's path to the system environment variables, sometimes logout and re-login is required. You may also place the DLL in the folder containing the application’s executable.

2 Definition of enum-type and struct

Several internal constants have been defined for the SDK.

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,// Each pixel is an 8-bit (1 byte) gray level
    ASI_IMG_RGB24,// Each pixel consists of RGB, 3 bytes totally (color cameras only)
    ASI_IMG_RAW16,// 2 bytes for every pixel with 65536 gray levels
    ASI_IMG_Y8,// monochrome 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,// no flip
    ASI_FLIP_HORIZ,// horizontal image flip
    ASI_FLIP_VERT,// vertical image flip
    ASI_FLIP_BOTH,// horizontal + vertical image flip

}ASI_FLIP_STATUS;
```
Image flip

```
2.5 typedef enum ASI_ERROR_CODE
{
    ASI_SUCCESS = 0,// operation was successful
    ASI_ERROR_INVALID_INDEX, //no camera connected or index value out of boundary
    ASI_ERROR_INVALID_ID, //invalid ID
    ASI_ERROR_INVALID_CONTROL_TYPE, //invalid control type
    ASI_ERROR_CAMERA_CLOSED, //camera didn't open
    ASI_ERROR_CAMERA_REMOVED, //failed to find the camera, maybe the camera has been
removed
    ASI_ERROR_INVALID_PATH, //cannot find the path of the file
    ASI_ERROR_INVALID_FILEFORMAT,
    ASI_ERROR_INVALID_SIZE, //wrong video format size
    ASI_ERROR_INVALID_IMGTYPE, //unsupported image format
    ASI_ERROR_OUTOF_BOUNDARY, //the startpos is outside the image boundary
    ASI_ERROR_TIMEOUT, //timeout
    ASI_ERROR_INVALID_SEQUENCE,//stop capture first
    ASI_ERROR_BUFFER_TOO_SMALL, //buffer size is not big enough
    ASI_ERROR_VIDEO_MODE_ACTIVE,
    ASI_ERROR_EXPOSURE_IN_PROGRESS,
    ASI_ERROR_GENERAL_ERROR,//general error, eg: value is out of valid range
    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 a specific camera
    long MaxHeight; //maximum image height
    long MaxWidth;   // maximum image width
    ASI_BOOL IsColorCam; //is this a color camera?
    ASI_BAYER_PATTERN BayerPattern;//Bayer filter type
    int SupportedBins[16]; //array consisting of supported bin values, list ends with 0
    ASI_IMG_TYPE SupportedVideoFormat[8];// array consisted of supported image types,list ends with ASI_IMG_END
    double PixelSize; //pixel pitch size(um)
    ASI_BOOL MechanicalShutter;// is a mechanical shutter supported
    ASI_BOOL ST4Port;//is there a ST4 port
    ASI_BOOL IsCoolerCam;//does the camera have a cooler
    ASI_BOOL IsUSB3Host;//camera operating under USB3?
    ASI_BOOL IsUSB3Camera;//is this a 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 (microsecond)
    ASI_GAMMA,//gamma with range 1 to 100 (nominally 50)
    ASI_WB_R,//red component of white balance
    ASI_WB_B,// blue component of white balance
    ASI_BRIGHTNESS,//pixel value offset (a bias, not a scale factor)
    ASI_BANDWIDTHOVERLOAD,//The total data transfer rate percentage
    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 seconds
    ASI_AUTO_MAX_BRIGHTNESS,//target brightness when auto adjust
    ASI_HARDWARE_BIN,//hardware binning of pixels
    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 a smaller grid at software bin mode for color camera
    ASI_FAN_ON,//only cooled camera has fan
    ASI_PATTERN_ADJUST.//currently only supported by 1600 mono camera
    ASI_ANTI_DEW_HEATER,
    ASI_AUTO_MAX_EXP_MS//maximum exposure time when auto adjust，unit is micro second

}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]; //control parameter 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 adjusted, for example sensor temperature can't be modified
    ASI_CONTROL_TYPE ControlType;//control type ID
    char Unused[32];
} ASI_CONTROL_CAPS;
    Capacity or value ranges 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,//exposure in progress
    ASI_EXP_SUCCESS,// exposure completed successfully, image can be read out
    ASI_EXP_FAILED,// exposure failure, need to restart exposure
}ASI_EXPOSURE_STATUS;
    Use under snap shot mode to obtain exposure status

2.11 typedef struct _ASI_ID
{
    unsigned char id[8];
}ASI_ID;
    ID to be writen into camera flash, 8 bytes totally

3 Function declaration

3.1 ASIGetNumOfConnectedCameras
Syntax：int ASIGetNumOfConnectedCameras()
Usage：get the count of connected ASI cameras

3.2 ASIGetCameraProperty
Syntax：ASI_ERROR_CODE ASIGetCameraProperty(ASI_CAMERA_INFO *pASICameraInfo, int iCameraIndex)
Usage：get the camera's information for a specific camera index (0 is the first camera)
Description：
    ASI_CAMERA_INFO *pASICameraInfo：pointer to the camera’s info structure
    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++)
{

```
ppASICameraInfo[i] = (ASI_CAMERA_INFO *)malloc(sizeof(ASI_CAMERA_INFO ));
ASIGetCameraProperty(ppASICameraInfo[i], i);
}
```

Notes：

Camera name can be obtained before the camera is opened with ASIOpenCamera

3.3 ASIOpenCamera

Syntax：ASI_ERROR_CODE ASIOpenCamera(int iCameraID)

Usage：open camera of a specific camera ID. This will not affect any other camera which is capturing. This should be the first call to start up a camera.

3.4 ASIInitCamera

Syntax：ASI_ERROR_CODE ASIInitCamera (int iCameraID)

Usage：initialize the specified camera ID, this API only affect the camera you are going to initialize and won't affect other cameras. This should be the second call to start up a camera.

3.5 ASICloseCamera

Syntax：ASI_ERROR_CODE ASICloseCamera(int iCameraID)

Usage：close a specific camera ID so that its resources will be released. This should be the last call to shut down a camera.

3.6 ASIGetNumOfControls

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

Usage：get the number of control types for the specific camera ID

3.7 ASIGetControlCaps

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

Usage：get control type's capacity or range of values for a specific control 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

3.8 ASIGetControlValue

Syntax：ASI_ERROR_CODE ASIGetControlValue (int iCameraID, ASI_CONTROL_TYPE ControlType, long *plValue, ASI_BOOL *pbAuto)

Usage：get a specific control type's value as currently set for a specific camera ID

Description：

int iCameraID：camera ID

ASI_CONTROL_TYPE ControlType：control type

long *plValue：pointer to the current value

ASI_BOOL *pbAuto：return whether the control is auto adjusted

3.9 ASISetControlValue

Syntax：ASI_ERROR_CODE ASISetControlValue(int iCameraID, ASI_CONTROL_TYPE ControlType, long lValue, ASI_BOOL bAuto)

Usage：set a specific control type's value for a specific camera ID

Description：

int iCameraID：camera ID

ASI_CONTROL_TYPE ControlType：control type

long lValue：control value to be set

ASI_BOOL bAuto：set whether the control is to be auto adjusted

Notes：when setting to auto adjust(bAuto=ASI_TRUE)，the lValue should be the current value

3.10 ASISetROIFormat

Syntax：ASI_ERROR_CODE ASISetROIFormat(int iCameraID, int iWidth, int iHeight, int iBin, ASI_IMG_TYPE Img_type)

Usage：set region of interest (ROI) size, binning, and image type

Description：

int iCameraID：camera ID

int iWidth：image width

int iHeight：image height

int iBin：NxN binning value

ASI_IMG_TYPE Img_type：image type

Return：success or error code

Notes：In general make sure iWidth%8=0，iHeight%2=0. For the USB2.0 camera ASI120，make sure iWidth* iHeight%1024=0，otherwise the call will result is an error code.

3.11 ASIGetROIFormat

Syntax：ASI_ERROR_CODE ASIGetROIFormat(int iCameraID, int *piWidth, int *piHeight, int *piBin, ASI_IMG_TYPE *pImg_type)

Usage：get the region of interest (ROI) values for size, binning, and image type

Description：

int iCameraID：camera ID

int *piWidth：image width

int *piHeight：image height

int *piBin：bin value

ASI_IMG_TYPE *pImg_type：image type

3.12 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.

3.13 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.

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

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

3.16 ASIDisableDarkSubtract
Syntax：ASI_ERROR_CODE ASIDisableDarkSubtract(int iCameraID)
Usage：disable dark subtraction function

3.17 ASIStartVideoCapture
Syntax：ASI_ERROR_CODE ASIStartVideoCapture(int iCameraID)
Usage：start the continuous video capture

3.18 ASIStopVideoCapture
Syntax：ASI_ERROR_CODE ASIStopVideoCapture(int iCameraID)
Usage：stop the continuous video capture

3.19 ASIGetVideoData
Syntax：ASI_ERROR_CODE ASIGetVideoData(int iCameraID, unsigned char* pBuffer, long lBuffSize, int iWaitms)
Usage：after ASIStartVideoCapture ()，call this function repeatedly to get images on a continuous basis. The function resets the capture to the next frame so you cannot get the same frame twice if the function is called two times in very short succession. The iWaitms is a timeout argument
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, new frame is discarded, it is best to create a circular buffer for holding the imagery to operate on the frames asynchronously.
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

3.20 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

3.21 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

3.22 ASIStartExposure
Syntax：ASI_ERROR_CODE ASIStartExposure(int iCameraID)
Usage：start a single snap shot. Note that there is a setup time for each snap shot, thus you cannot get two snapshots in succession with a shorter time span that these values.

3.23 ASIStopExposure
Syntax：ASI_ERROR_CODE　ASIStopExposure(int iCameraID)
Usage：stop a single snap shot, this API can be used for very long exposure and you don't want to wait so long such like exposure 5 minutes and you want to cancel after 1 min, then you can call this API
Notes：if exposure status is success after stop exposure，image can still be read out

3.24 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

3.25 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 ()

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

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

3.28 ASIGetProductIDs
Syntax：int ASIGetProductIDs(int* pPIDs)
Usage: get the product ID of each supported camera, at first set pPIDs as 0 and get length and then malloc a buffer to contain the PIDs

Description:
int* pPIDs: pointer to array of PIDs

Return: length of the array.

4 Suggested call sequence

4.1 Initialization
Get count of connected cameras--> ASIGetNumOfConnectedCameras
Get cameras' ID and other information like name, resolution, etc. Refreshing devices won't change this ID--> ASIGetCameraProperty
Open camera -->ASIOpenCamera（Notes： this SDK can operate multiple cameras which aredistinguished uniquely by CameraID)
Initialize-->ASIInitCamera
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 modes for capturing frames：video mode and snap shot mode. Images are captured continuously under video mode, and only a single image is captured under snap shot mode.

- video mode

Start video capture-->ASIStartVideoCapture
Operate on video frames as they are captured. Have the thread below signal that a new frame is available.
Stop video capture-->ASIStopVideoCapture

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

```
while(1)
{
    if(ASIGetVideoData == ASI_SUCCESS)(internally uses a waitFor so does not spin CPU cycles until a frame is digitized and available)
    {
      ...
    }
}
```

- 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 for each camera