

eYs3D Windows SDK 1.5.0.3

Generated by Doxygen 1.9.1

1 Introduction	3
2 Data Structure Index	5
2.1 Data Structures	5
3 File Index	7
3.1 File List	7
4 Data Structure Documentation	9
4.1 DEVINFORMATIONEX	9
4.1.1 Detailed Description	9
4.1.2 Field Documentation	9
4.1.2.1 nChipID	9
4.1.2.2 nDevType	9
4.1.2.3 strDevName	10
4.1.2.4 wPID	10
4.1.2.5 wUsbNode	10
4.1.2.6 wVID	10
4.2 eSPCtrl_RectLogData	10
4.2.1 Detailed Description	11
4.3 ParaLUT	11
4.3.1 Detailed Description	13
4.4 tagDEVINFORMATION	13
4.4.1 Detailed Description	14
4.4.2 Field Documentation	14
4.4.2.1 nChipID	14
4.4.2.2 nDevType	14
4.4.2.3 strDevName	14
4.4.2.4 wPID	14
4.4.2.5 wUsbNode	15
4.4.2.6 wVID	15
5 File Documentation	17
5.1 eSPDI_Common.h File Reference	17
5.1 eSPDI_Common.n File Releience	21
	21
5.1.2 Typedef Documentation	
5.1.2.1 DEVINFORMATION	21
5.1.2.2 eSPCtrl_RectLogData	21
5.1.2.3 PARALUT	21
5.1.3 Enumeration Type Documentation	21
5.1.3.1 APC_DEVICE_TYPE	21
5.1.3.2 SENSOR_TYPE_NAME	22
5.1.3.3 USERDATA_SECTION_INDEX	22
5.1.4 Function Documentation	23

5.1.4.1 APC_CloseDevice()
5.1.4.2 APC_DisableAE()
5.1.4.3 APC_DisableAWB()
5.1.4.4 APC_EnableAE()
5.1.4.5 APC_EnableAWB()
5.1.4.6 APC_EnableGPUAcceleration()
5.1.4.7 APC_FindDevice()
5.1.4.8 APC_GetCTPropVal()
5.1.4.9 APC_GetCurrentlRValue()
5.1.4.10 APC_GetDepthFilterVersion()
5.1.4.11 APC_GetDeviceNumber()
5.1.4.12 APC_GetDeviceResolutionList()
5.1.4.13 APC_GetFlexibleGyroData()
5.1.4.14 APC_GetFlexibleGyroLength()
5.1.4.15 APC_GetFWRegister()
5.1.4.16 APC_GetFwVersion()
5.1.4.17 APC_GetGPIOValue()
5.1.4.18 APC_GetHWRegister()
5.1.4.19 APC_GetIRMaxValue()
5.1.4.20 APC_GetIRMinValue()
5.1.4.21 APC_GetLogData()
5.1.4.22 APC_GetPidVid()
5.1.4.23 APC_GetPUPropVal()
5.1.4.24 APC_GetSensorRegister()
5.1.4.25 APC_GetSlaveHWRegister()
5.1.4.26 APC_GetSlaveLogData()
5.1.4.27 APC_GetSlaveSensorRegister()
5.1.4.28 APC_Init()
5.1.4.29 APC_Init2()
5.1.4.30 APC_Is360Device()
5.1.4.31 APC_OpenDevice()
5.1.4.32 APC_ReadFlashData()
5.1.4.33 APC_RefreshDevice()
5.1.4.34 APC_RegisterDeviceEvents()
5.1.4.35 APC_Release()
5.1.4.36 APC_SetCTPropVal()
5.1.4.37 APC_SetCurrentIRValue()
5.1.4.38 APC_SetFWRegister()
5.1.4.39 APC_SetGPIOCtrl()
5.1.4.40 APC_SetGPIOValue()
5.1.4.41 APC_SetHuffmanTableData()
5.1.4.42 APC_SetHWRegister()

5.1.4.43 APC_SetIRMaxValue()	43
5.1.4.44 APC_SetLogData()	43
5.1.4.45 APC_SetPidVid()	44
5.1.4.46 APC_SetPUPropVal()	44
5.1.4.47 APC_SetQuantizationTableData()	45
5.1.4.48 APC_SetSensorTypeName()	45
5.1.4.49 APC_SetSlaveHWRegister()	46
5.1.4.50 APC_SetSlaveLogData()	46
5.1.4.51 APC_SetUserData()	47
5.2 eSPDI_DM.h File Reference	47
5.2.1 Detailed Description	48
5.2.2 Function Documentation	48
5.2.2.1 APC_GetDepthDataType()	48
5.2.2.2 APC_GetRectifyMatLogData()	49
5.2.2.3 APC_GetRectifyTable()	49
5.2.2.4 APC_GetSlaveRectifyTable()	50
5.2.2.5 APC_GetSlaveYOffset()	50
5.2.2.6 APC_GetYOffset()	51
5.2.2.7 APC_GetZDTable()	52
5.2.2.8 APC_SetDepthDataType()	52
5.2.2.9 APC_SetHWPostProcess()	53
5.2.2.10 APC_SetSlaveYOffset()	53
5.2.2.11 APC_SetYOffset()	54
5.3 eSPDI_ErrCode.h File Reference	54
5.3.1 Detailed Description	54
Index	55

# **Chapter 1**

# Introduction

This document describes the usage of Application Programming Interfaces of eYs3D Windows SDK

## What's inside the SDK

Table 1.1 File List

Folder	Subfolder	Filename	Description
bin	Win32	All files	Sample executables on Win32 platform
DIII	x64	All files	Sample executables on Windows 64-bits platform
		eSPDI_Common.h	Basic API declaration header
	include	eSPDI_DM.h	Depth Map specific API declaration header
		eSPDI_ErrCode.h	Error code definitions
eSPDI	Win32	eSPDI_DM.dll	eSPDI dynamical linked library for Win32 platform
	VVIIIOZ	eSPDI_DM.lib	eSPDI static linked library for Win32 platform
	x64	eSPDI_DM.dll	eSPDI dynamical linked library for Windows 64-bits
	X04	eSPDI_DM.lib	eSPDI static linked library for Windows 64-bits
doc	html	index.html	This documentation
DMPreview			A sample VC++ project demonstrating how to open multiple devices in an application

4 Introduction

# **Chapter 2**

# **Data Structure Index**

## 2.1 Data Structures

Here are the data structures with brief descriptions:

DEVINFORMATIONEX	9
eSPCtrl_RectLogData	
ESPCtrl_RectLogData	10
ParaLUT	
ParaLUT	11
tagDEVINFORMATION	
DEVINFORMATION	13

6 Data Structure Index

# **Chapter 3**

# File Index

## 3.1 File List

Here is a list of all documented files with brief descriptions:

eSPDI_Common.h	
EYs3D SDK API export functions, data structure and variable definition	17
eSPDI_DM.h	
EYs3D SDK API export functions, data structure and variable definition for depth map module .	47
eSPDI_ErrCode.h	
Definition of eYs3D SDK error code Copyright: This file copyright (C) 2017 by	54

8 File Index

## **Chapter 4**

## **Data Structure Documentation**

## 4.1 DEVINFORMATIONEX

## **Data Fields**

- unsigned short wPID
- unsigned short wVID
- char strDevName [512]
- unsigned short nChipID
- unsigned short nDevType
- unsigned short wUsbNode

## 4.1.1 Detailed Description

extended device information class

## 4.1.2 Field Documentation

## 4.1.2.1 nChipID

unsigned short nChipID

chip ID, 0x18 for AXES1, 0x1C for KIWI, 0x15 for PUMA

## 4.1.2.2 nDevType

unsigned short nDevType

chip enum value, see APC\_DEVICE\_TYPE

## 4.1.2.3 strDevName

char strDevName[512]

device name

#### 4.1.2.4 wPID

unsigned short wPID

product ID

Table 4.1 PID List

Chip Name	Chip ID	PID
		0x0568
		0x0668
AXES1	0x18	0x0113
		0x0115
		0x0116
KIWI	0x1C	0x0118
PUMA	0x15	0x0112
		0x0120

#### 4.1.2.5 wUsbNode

unsigned short wUsbNode

**USB** Node

## 4.1.2.6 wVID

unsigned short wVID

vender ID, 0x1E4E for ApcDI device

The documentation for this class was generated from the following file:

• eSPDI\_Common.h

## 4.2 eSPCtrl\_RectLogData

eSPCtrl\_RectLogData

4.3 ParaLUT

## 4.2.1 Detailed Description

```
eSPCtrl_RectLogData
```

Rectified log data structure

The documentation for this struct was generated from the following file:

• eSPDI\_Common.h

## 4.3 ParaLUT

ParaLUT.

#### **Data Fields**

```
• long long file_ID_header
```

[00]-[000] File ID header: 2230

long long file\_ID\_version

[01]-[008] File ID version: 4

double FOV

[02]-[016] Field of view with degree

long long semi\_FOV\_pixels

[03]-[024] Pixels for semi-FOV

long long img\_src\_cols

[04]-[032] Width for source image (single image)

• long long img\_src\_rows

[05]-[040] Height for source image

double img\_L\_src\_col\_center

[06]-[048] Center of width for L side source image

double img\_L\_src\_row\_center

[07]-[056] Center of height for L side source image

double img\_R\_src\_col\_center

[08]-[064] Center of width for R side source image

double img\_R\_src\_row\_center

[09]-[072] Center of height for R side source image

double img\_L\_rotation

[10]-[080] Rotation for L side image

• double img\_R\_rotation

[11]-[088] Rotation for R side image

• double spline\_control\_v1

[12]-[096] Spline control value for row = DIV x 0 pixel, DIV = rows/6

double spline\_control\_v2

[13]-[104] Spline control value for row = DIV x 1 pixel, DIV = rows/6

double spline\_control\_v3

[14]-[112] Spline control value for  $row = DIV \times 2$  pixel, DIV = rows/6

double spline\_control\_v4

[15]-[120] Spline control value for row = DIV x 3 pixel, DIV = rows/6

```
• double spline_control_v5
     [16]-[128] Spline control value for row = DIV x 4 pixel, DIV = rows/6

    double spline control v6

     [17]-[136] Spline control value for row = DIV x 5 pixel, DIV = rows/6

    double spline_control_v7

     [18]-[144] Spline control value for row = DIV x 6 pixel, DIV = rows/6
• long long img_dst_cols
     [19]-[152] Width for output image (single image), according to "Original" parameters

    long long img_dst_rows

     [20]-[160] Height for output image, according to "Original" parameters
· long long img L dst shift
     [21]-[168] Output L side image shift in row
• long long img_R_dst_shift
     [22]-[176] Output R side image shift in row
· long long img overlay LR
     [23]-[184] Overlay between L/R in pixels, far field, (YUV must be even)
· long long img_overlay_RL
     [24]-[192] Overlay between R/L in pixels, far field, (YUV must be even)
· long long img stream cols
     [25]-[200] Output image stream of cols
· long long img stream rows
     [26]-[208] Output image stream of rows

    long long video_stream_cols

     [27]-[216] Output video stream of cols
· long long video_stream_rows
     [28]-[224] Output video stream of rows

    long long usb_type

     [29]-[232] 2 for usb2, 3 for usb3

    long long img_type

     [30]-[240] 1 for yuv422, 2 for BGR, 3 for RGB

    long long lut_type

     [31]-[248] Output LUT tye eys::LutModes
• long long blending_type
     [32]-[256] 0 for choosed by function, 1 for alpha-blending, 2 for Laplacian pyramid blending
· double overlay ratio
      [33]-[264] far field overlay value is equal to img_overlay_LR(RL) = overlay_value + overlay_ratio

    long long serial number date0

     [34]-[272] 8 bytes, yyyy-mm-dd

    long long serial_number_date1

     [35]-[280] 8 bytes, hh-mm-ss-xxx, xxx for machine number
· double unit_sphere_radius
     [36]-[288] Original: Unit spherical radius for dewarping get x and y
· double min col
     [37]-[296] Original: Parameters of min position of image width

    double max_col

     [38]-[304] Original: Parameters of max position of image width
· double min row
```

[39]-[312] Original: Parameters of min position of image height

· double max row

[40]-[320] Original: Parameters of max position of image height

long long AGD\_LR

[41]-[328] Err: Average gray-level value discrepancy at LR boundary

· long long AGD\_RL

[42]-[336] Err: Average gray-level value discrepancy at RL boundary

long long out\_img\_resolution

[43]-[344] Set output resolution eys::ImgResolutionModes

· long long out lut cols

[44]-[352] Output side-by-side lut width, according to the set of out\_img\_resolution

long long out\_lut\_rows

[45]-[360] Output lut height, according to the set of out img resolution

long long out\_lut\_cols\_eff

[46]-[368] Output effective pixels in out\_lut\_cols, 0 is for all

• long long out\_lut\_rows\_eff

[47]-[376] Output effecitve pixels in out\_lut\_rows, 0 is for all

• long long out\_img\_cols

[48]-[384] Output side-by-side image width after dewarping and stitching, according to the set of out\_img\_resolution

long long out\_img\_rows

[49]-[392] Output image height, according to the set of out\_img\_resolution

· long long out overlay LR

[50]-[340] Output L/R overlay value, according to the set of out\_img\_resolution

· long long out overlay RL

[51]-[408] Output R/L overlay value, according to the set of out\_img\_resolution

long long reserve [44]

[52]-[416] Reserve 44 parameter to use

## 4.3.1 Detailed Description

#### ParaLUT.

Spherical look-up table conversion parameters

The documentation for this struct was generated from the following file:

· eSPDI\_Common.h

## 4.4 tagDEVINFORMATION

DEVINFORMATION.

#### **Data Fields**

- · unsigned short wPID
- · unsigned short wVID
- char \* strDevName
- unsigned short nChipID
- unsigned short nDevType
- unsigned short wUsbNode

## 4.4.1 Detailed Description

DEVINFORMATION.

device information

## 4.4.2 Field Documentation

## 4.4.2.1 nChipID

unsigned short nChipID

chip ID, 0x18 for AXES1, 0x1C for KIWI, 0x15 for PUMA

#### 4.4.2.2 nDevType

unsigned short nDevType

chip enum value,

See also

APC\_DEVICE\_TYPE

#### 4.4.2.3 strDevName

char\* strDevName

pointer to device name stored inside the SDK

#### 4.4.2.4 wPID

unsigned short wPID

product ID

Table 4.2 PID List

Chip Name	Chip ID	PID
AXES1		0x0568
		0x0668
	0x18	0x0113
		0x0115
		0x0116
KIWI	0x1C	0x0118
DLIMA	0v15	0x0112

0x15

0x0120

**PUMA** 

Generated by Doxygen

#### 4.4.2.5 wUsbNode

unsigned short wUsbNode

**USB** Node

## 4.4.2.6 wVID

unsigned short wVID

vender ID, 0x1E4E for ApcDI device

The documentation for this struct was generated from the following file:

• eSPDI\_Common.h

## **Chapter 5**

## **File Documentation**

## 5.1 eSPDI Common.h File Reference

eYs3D SDK API export functions, data structure and variable definition

#### **Data Structures**

• struct eSPCtrl\_RectLogData

eSPCtrl\_RectLogData

struct ParaLUT

ParaLUT.

struct tagDEVINFORMATION

DEVINFORMATION.

• class DEVINFORMATIONEX

## **Typedefs**

• typedef struct eSPCtrl\_RectLogData eSPCtrl\_RectLogData

eSPCtrl\_RectLogData

typedef struct ParaLUT PARALUT

ParaLUT.

• typedef struct tagDEVINFORMATION DEVINFORMATION

DEVINFORMATION.

#### **Enumerations**

```
enum APC_DEVICE_TYPE { OTHERS = 0 , AXES1 , PUMA , PLUM }
enum USERDATA_SECTION_INDEX {
    USERDATA_SECTION_0 = 0 , USERDATA_SECTION_1 , USERDATA_SECTION_2 , USERDATA_SECTION_3 ,
    USERDATA_SECTION_4 , USERDATA_SECTION_5 , USERDATA_SECTION_6 , USERDATA_SECTION_7 ,
    USERDATA_SECTION_8 , USERDATA_SECTION_9 , USERDATA_SECTION_10 , USERDATA_SECTION_NUM }
enum SENSOR_TYPE_NAME {
    APC_SENSOR_TYPE_H22 = 0 , APC_SENSOR_TYPE_OV7740 , APC_SENSOR_TYPE_AR0134 ,
    APC_SENSOR_TYPE_AR0135 ,
    APC_SENSOR_TYPE_AR0144 , APC_SENSOR_TYPE_OV9714 , APC_SENSOR_TYPE_OV9282 ,
    APC_SENSOR_TYPE_AR0330 ,
    APC_SENSOR_TYPE_AR1335 , APC_SENSOR_TYPE_H65 , APC_SENSOR_TYPE_AR0522 }
```

#### **Functions**

• int APC\_API APC\_Init (void \*\*ppHandleApcDI, bool blsLogEnabled)

entry point of eYs3D camera SDK. This API allocates resource and find all the eSPI camera devices connected to the system.

int APC\_API APC\_Init2 (void \*\*ppHandleApcDI, bool blsLogEnabled, bool bAutoRestart)

entry point of eYs3D camera SDK. This API allocates resource and find all the eSPI camera devices connected to the system.

• int APC\_API APC\_RegisterDeviceEvents (void \*pHandleApcDI, APC\_DeviceEventFn cbFunc, void \*pData)

Register the USB device plug or unplug events. Any USB capture device attachment or detachment events will call the callback function cbFunc.

void APC API APC Release (void \*\*ppHandleApcDI)

release all resource that APC\_Init had allocated

int APC API APC FindDevice (void \*pHandleApcDI)

find out all eYs3D USB devices by PID, VID and ChipID, also remember device types

int APC API APC RefreshDevice (void \*pHandleApcDI)

refresh all eYs3D UVC devices

int APC API APC GetDeviceNumber (void \*pHandleApcDI)

get eYs3D USB device numbers

• int APC\_API APC\_GetSlaveSensorRegister (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, int nld, unsigned short address, unsigned short \*pValue, int flag, int nSensorMode)

get value from sensor register

• int APC\_API APC\_GetSensorRegister (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, int nld, unsigned short address, unsigned short \*pValue, int flag, int nSensorMode)

get value from sensor register

int APC\_API APC\_GetFWRegister (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, unsigned short address, unsigned short \*pValue, int flag)

get firmware register value

• int APC\_API APC\_SetFWRegister (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, unsigned short address, unsigned short nValue, int flag)

set firmware register value

 int APC\_API APC\_GetSlaveHWRegister (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, unsigned short address, unsigned short \*pValue, int flag)

get hardware register value

int APC\_API APC\_GetHWRegister (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, unsigned short address, unsigned short \*pValue, int flag)

get hardware register value

 int APC\_API APC\_SetSlaveHWRegister (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, unsigned short address, unsigned short nValue, int flag)

set hardware register

• int APC\_API APC\_SetHWRegister (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, unsigned short address, unsigned short nValue, int flag)

set hardware register

int APC\_API APC\_GetFwVersion (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, char \*pszFwVersion, int nBufferSize, int \*pActualLength)

get the firmware version of device, the version is a string

• int APC\_API APC\_GetPidVid (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, unsigned short \*pPidBuf, unsigned short \*pVidBuf)

get PID(product ID) and VID(vendor ID) of device

int APC\_API APC\_SetPidVid (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, unsigned short \*pPidBuf, unsigned short \*pVidBuf)

set PID and VID to device

• int APC\_API APC\_GetSlaveLogData (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, BYTE \*buffer, int BufferLength, int \*pActualLength, int index)

get log data from flash

• int APC\_API APC\_GetLogData (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, BYTE \*buffer, int BufferLength, int \*pActualLength, int index)

get log data from flash

• int APC\_API APC\_SetSlaveLogData (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, BYTE \*buffer, int BufferLength, int \*pActualLength, int index)

set log data to flash

• int APC\_API APC\_SetLogData (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, BYTE \*buffer, int BufferLength, int \*pActualLength, int index)

set log data to flash

• int APC\_API APC\_SetUserData (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, BYTE \*buffer, int BufferLength, USERDATA SECTION INDEX usi)

set user data to flash

• int APC\_API APC\_ReadFlashData (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, FLASH\_DATA\_TYPE fdt, BYTE \*pBuffer, unsigned long int nLengthOfBuffer, unsigned long int \*pActualBufferLen)

read firmware code(.bin) form flash The firmware code is the combination of boot loader, firmware body and plug-in data. This input buffer length has to match with the flash data type

 int APC\_API APC\_OpenDevice (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, int colorStreamIndex, int depthStreamIndex, int depthStreamSwitch, int iFps, APC\_ImgCallbackFn callbackFn, void \*pCallbackParam, int pid=-1)

open camera device with image callback support

• int APC\_API APC\_GetColorImage (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, BYTE \*pBuf, unsigned long int \*pImageSize, int \*pSerial=NULL)

get color image

int APC\_API APC\_CloseDevice (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo)
 close device and stop video render

int APC\_API APC\_GetDeviceResolutionList (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, int nMax
 Count0, APC\_STREAM\_INFO \*pStreamInfo0, int nMaxCount1, APC\_STREAM\_INFO \*pStreamInfo1)
 get the device resolution list

bool APC\_API APC\_Is360Device (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo)
 check module is spherical device or not

• int APC\_API APC\_GetSerialNumberFromLog (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, char \*p↔ SerialNum, int nBufferSize, int \*pActualLength)

get the module serial number

- int APC\_API APC\_SetCurrentIRValue (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, WORD wType) set current infrared radiation(IR) value
- int APC\_API APC\_GetCurrentIRValue (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, WORD \*pwType) get current infrared radiation(IR) value
- int APC\_API APC\_GetIRMinValue (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, WORD \*pwType) get minimum IR value the module support
- int APC\_API APC\_SetIRMaxValue (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, WORD wType)
   set maximum IR value the module support
- int APC\_API APC\_GetIRMaxValue (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, WORD \*pwType)
   get maximum IR value the module support
- int APC\_API APC\_SetIRMode (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, WORD wType)
   set IR mode, left, right or both
- int APC\_API APC\_GetIRMode (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, WORD \*pwType)
   set IR mode, left, right or both
- int APC\_API APC\_EnableSensorIF (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, bool blsEnable)
   turn on/off sensor IF function

int APC\_API APC\_SetSensorTypeName (void \*pHandleApcDI, SENSOR\_TYPE\_NAME stn)
 select which sensor to operate

int APC\_API APC\_EnableAE (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo)
 enable auto exposure function of ISP

• int APC\_API APC\_DisableAE (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo)

disable auto exposure function of ISP

• int APC\_API APC\_EnableAWB (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo)

enable auto white balance function of ISP

• int APC\_API APC\_DisableAWB (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo)

disable auto white balance of ISP

• int APC\_API APC\_GetGPIOValue (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, int nGPIOIndex, BYTE \*pValue)

get general purpose IO value

int APC\_API APC\_SetGPIOValue (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, int nGPIOIndex, BYTE nValue)

set GPIO value

int APC\_API APC\_SetGPIOCtrl (void \*pHandleApcDl, PDEVSELINFO pDevSelInfo, int nGPIOIndex, BYTE nValue)

set GPIO control address

• int APC\_API APC\_GetPUPropVal (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, int nld, int \*pValue)

get processing unit property value <a href="https://msdn.microsoft.com/en-us/library/windows/hardware/ff568122">https://msdn.microsoft.com/en-us/library/windows/hardware/ff568122</a> 85) .aspx The PROPSETID\_VIDCAP\_VIDEOPROCAMP property set controls devices that can adjust image color attributes of analog or digital signals.

• int APC\_API APC\_SetPUPropVal (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, int nId, int nValue)

get processing unit property value https://msdn.microsoft.com/en-us/library/windows/hardware/ff568122 (
85).aspx https://msdn.microsoft.com/en-us/library/windows/hardware/ff566089(v=vs.↔
85).aspx

• int APC\_API APC\_GetCTPropVal (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, int nld, int \*pValue)

set control terminal property value <a href="https://msdn.microsoft.com/en-us/library/windows/hardware/ff567802">https://msdn.microsoft.com/en-us/library/windows/hardware/ff567802</a> 85) .aspx The PROPSETID\_VIDCAP\_CAMERACONTROL property set controls camera device settings. The controls it provides are a subset of the ITU T.RDC standard.

• int APC\_API APC\_SetCTPropVal (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, int nId, int nValue)

get control terminal property value https://msdn.microsoft.com/en-us/library/windows/hardware/ff56780285).aspx https://msdn.microsoft.com/en-us/library/windows/hardware/ff566089(v=vs.  $\leftrightarrow$  85).aspx

 int APC\_API APC\_GetAutoExposureMode (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, unsigned short \*mode)

misc function : get auto exposure mode

 int APC\_API APC\_SetAutoExposureMode (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, unsigned short mode)

misc function : set auto exposure mode

• int APC\_API APC\_GetFlexibleGyroData (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, int length, BYTE \*pGyroData)

get IMU(Gyro) data

 int APC\_API APC\_GetFlexibleGyroLength (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, unsigned short \*GyroLen)

get the IMU(Gyro) data length

 int APC\_API APC\_SetHuffmanTableData (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, const char \*filename, bool bLogFile)

set huffman table data for jpeg encode

• int APC\_API APC\_SetQuantizationTableData (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, const char \*filename)

set quantication table data for jpeg encode

- int APC\_API APC\_SetPlumAR0330 (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, bool bEnable) Set Plum Sensor AR0330.
- int APC\_API APC\_EnableGPUAcceleration (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, bool enable) enable depth filter with GPU acceleration or not
- APC\_API char \* APC\_GetDepthFilterVersion (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo)
   get depth filter version

#### 5.1.1 Detailed Description

eYs3D SDK API export functions, data structure and variable definition

Copyright

This file copyright (C) 2017 by eYs3D company

An unpublished work. All rights reserved. This file is proprietary information, and may not be disclosed or copied without the prior permission of eYs3D company.

## 5.1.2 Typedef Documentation

#### 5.1.2.1 DEVINFORMATION

typedef struct tagDEVINFORMATION DEVINFORMATION

DEVINFORMATION.

device information

#### 5.1.2.2 eSPCtrl\_RectLogData

typedef struct eSPCtrl\_RectLogData eSPCtrl\_RectLogData

eSPCtrl RectLogData

Rectified log data structure

#### **5.1.2.3 PARALUT**

typedef struct ParaLUT PARALUT

ParaLUT.

Spherical look-up table conversion parameters

## 5.1.3 Enumeration Type Documentation

#### 5.1.3.1 APC\_DEVICE\_TYPE

enum APC\_DEVICE\_TYPE

chip enum value

## Enumerator

OTHERS	Other
AXES1	AXIS1
PUMA	PUMA

## 5.1.3.2 SENSOR\_TYPE\_NAME

enum SENSOR\_TYPE\_NAME

## Enumerator

APC_SENSOR_TYPE_H22	H22
APC_SENSOR_TYPE_OV7740	OV7740
APC_SENSOR_TYPE_AR0134	AR0134
APC_SENSOR_TYPE_AR0135	AR0135
APC_SENSOR_TYPE_AR0144	AR0144
APC_SENSOR_TYPE_OV9714	OV9714
APC_SENSOR_TYPE_OV9282	OV9282
APC_SENSOR_TYPE_AR0330	AR0330
APC_SENSOR_TYPE_AR1335	AR1335

## 5.1.3.3 USERDATA\_SECTION\_INDEX

enum USERDATA\_SECTION\_INDEX

## Enumerator

USERDATA_SECTION_0	Section 0
USERDATA_SECTION_1	Section 1
USERDATA_SECTION_2	Section 2
USERDATA_SECTION_3	Section 3
USERDATA_SECTION_4	Section 4
USERDATA_SECTION_5	Section 5
USERDATA_SECTION_6	Section 6
USERDATA_SECTION_7	Section 7
USERDATA_SECTION_8	Section 8
USERDATA_SECTION_9	Section 9
USERDATA_SECTION_10	Section 10
USERDATA SECTION NUM	Total Section Number

## 5.1.4 Function Documentation

## 5.1.4.1 APC\_CloseDevice()

close device and stop video render

#### **Parameters**

pHandleApcDI	the pointer to the initilized ApcDI SDK instance
pDevSelInfo	pointer of device select index

#### Returns

success:APC\_OK, others:see eSPDI\_ErrCode.h

## 5.1.4.2 APC\_DisableAE()

disable auto exposure function of ISP

## **Parameters**

pHandleApcDI	CApcDI handler
pDevSelInfo	pointer of device select index

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

## 5.1.4.3 APC\_DisableAWB()

disable auto white balance of ISP

#### **Parameters**

pHandleApcDI	CApcDI handler
pDevSelInfo	pointer of device select index

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

## 5.1.4.4 APC\_EnableAE()

enable auto exposure function of ISP

#### **Parameters**

pHandleApcDI	CApcDI handler
pDevSelInfo	pointer of device select index

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

## 5.1.4.5 APC\_EnableAWB()

enable auto white balance function of ISP

#### **Parameters**

pHandleApcDI	CApcDI handler
pDevSelInfo	pointer of device select index

### Returns

success: APC\_OK, others: see eSPDI\_ErrCode.h

## 5.1.4.6 APC\_EnableGPUAcceleration()

enable depth filter with GPU acceleration or not

#### **Parameters**

pHandleApcDl	the pointer to the initilized ApcDI SDK instance
pDevSelInfo	pointer of device select index
enable	true:enable, fales:diable

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

## 5.1.4.7 APC\_FindDevice()

find out all eYs3D USB devices by PID, VID and ChipID, also remember device types

#### **Parameters**

pHandleApcDI	the pointer to the initilized ApcDI SDK instance
--------------	--

#### Returns

```
success: APC OK, others: see eSPDI ErrCode.h
```

## 5.1.4.8 APC\_GetCTPropVal()

set control terminal property value <a href="https://msdn.microsoft.com/en-us/library/windows/hardware/ff5678">https://msdn.microsoft.com/en-us/library/windows/hardware/ff5678</a> 85) .aspx The PROPSETID\_VIDCAP\_CAMERACONTROL property set controls camera device settings. The controls it provides are a subset of the ITU T.RDC standard.

The KSPROPERTY\_VIDCAP\_CAMERACONTROL enumeration in Ksmedia.h specifies the properties of this set.

Support for this property set is optional and should be implemented only by minidrivers of devices that offer camera control settings. For more information, see the ITU website.

Prior to USB video class, this enumeration contained the following properties: KSPROPERTY\_CAMERACONTROL \_EXPOSURE KSPROPERTY\_CAMERACONTROL\_FOCUS KSPROPERTY\_CAMERACONTROL\_IRIS KSPROPERTY \_CAMERACONTROL\_ZOOM KSPROPERTY\_CAMERACONTROL\_PAN KSPROPERTY\_CAMERACONTROL \_ROLL KSPROPERTY\_CAMERACONTROL\_TILT

 $https://msdn.microsoft.com/en-us/library/windows/hardware/ff566089 (v=vs. {\it }e-vs. {\it }e-vs.$ 

#### **Parameters**

*pHandleApcDI	CApcDI handler
pDevSelInfo	pointer of device select index
nld	specifies the member of the property set
pValue	pointer of store CT property value

#### Returns

success: APC\_OK, others: see eSPDI\_ErrCode.h

## 5.1.4.9 APC\_GetCurrentIRValue()

get current infrared radiation(IR) value

#### **Parameters**

pHandleApcDl	CApcDI handler
pDevSelInfo	pointer of device select index
рwТуре	value of current IR

#### Returns

success: APC\_OK, others: see eSPDI\_ErrCode.h

## 5.1.4.10 APC\_GetDepthFilterVersion()

get depth filter version

#### **Parameters**

pHandleApcDI	the pointer to the initilized ApcDI SDK instance
pDevSelInfo	pointer of device select index

#### Returns

success: get version string, others: get N/A string

## 5.1.4.11 APC\_GetDeviceNumber()

get eYs3D USB device numbers

#### **Parameters**

#### Returns

number of eYs3D device

## 5.1.4.12 APC\_GetDeviceResolutionList()

```
int APC_GetDeviceResolutionList (
    void * pHandleApcDI,
    PDEVSELINFO pDevSelInfo,
    int nMaxCount0,
    APC_STREAM_INFO * pStreamInfo0,
    int nMaxCount1,
    APC_STREAM_INFO * pStreamInfo1 )
```

get the device resolution list

#### **Parameters**

pHandleApcDI	the pointer to the initilized ApcDI SDK instance
pDevSelInfo	pointer of device select index
nMaxCount0	max count of endpoint1 resolutions
pStreamInfo0	resolution infos of endpoint1
nMaxCount1	max count of endpoint2 resolutions
pStreamInfo1	resolutions infos of endpoint2

#### Returns

```
success: nCount0*256+nCount1, others: see eSPDI_ErrCode.h
```

## 5.1.4.13 APC\_GetFlexibleGyroData()

## get IMU(Gyro) data

#### **Parameters**

pHandleApcDI	CApcDI handler
pDevSelInfo	pointer of device select index
length	length of IMU data to read, should be get from APC_GetFlexibleGyroLength
pGyroData	data buffer to store IMU data

## 5.1.4.14 APC\_GetFlexibleGyroLength()

## get the IMU(Gyro) data length

#### **Parameters**

pHandleApcDI	CApcDI handler
pDevSelInfo	pointer of device select index
GyroLen	pointer to store IMU data length

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

## 5.1.4.15 APC\_GetFWRegister()

#### get firmware register value

#### **Parameters**

pHandleApcDI	CApcDI handler
pDevSelInfo	pointer of device select index
address	register address
pValue	pointer of value got from register address
flag	address and value data length(2 or 1 byte) ie FG_Address_2Byte   FG_Value_2Byte is 2 byte address and 2 byte value #define FG_Address_1Byte 0x01 #define FG_Address_2Byte 0x02 #define FG_Value_1Byte 0x10 #define FG_Value_2Byte 0x20

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

## 5.1.4.16 APC\_GetFwVersion()

```
int APC_GetFwVersion (
          void * pHandleApcDI,
          PDEVSELINFO pDevSelInfo,
          char * pszFwVersion,
          int nBufferSize,
          int * pActualLength )
```

get the firmware version of device, the version is a string

### **Parameters**

pHandleApcDI	CApcDI handler
pDevSelInfo	pointer of device select index
pszFwVersion	firmware version string
nBufferSize	input buffer length to receive FW version
pActualLength	the actual length of FW version in byte

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

## 5.1.4.17 APC\_GetGPIOValue()

## get general purpose IO value

#### **Parameters**

pHandleApcDl	CApcDI handler
pDevSelInfo	pointer of device select index
nGPIOIndex	GPIO index, 1 or 2 is valid
pValue	pointer of GPIO value

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

## 5.1.4.18 APC\_GetHWRegister()

#### get hardware register value

#### **Parameters**

pHandleApcDI	CApcDI handler
pDevSelInfo	pointer of device select index
address	register address
pValue	pointer of value got from register address
flag	address and value data length(2 or 1 byte) ie FG_Address_2Byte   FG_Value_2Byte is 2 byte address and 2 byte value #define FG_Address_1Byte 0x01 #define FG_Address_2Byte 0x02 #define FG_Value_1Byte 0x10 #define FG_Value_2Byte 0x20

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

## 5.1.4.19 APC\_GetIRMaxValue()

get maximum IR value the module support

#### **Parameters**

pHandleApcDI	the pointer to the initilized ApcDI SDK instance
pDevSelInfo	pointer of device select index
рwТуре	pointer strors maximum IR value

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

## 5.1.4.20 APC\_GetIRMinValue()

get minimum IR value the module support

#### **Parameters**

pHandleApcDl	the pointer to the initilized ApcDI SDK instance
pDevSelInfo	pointer of device select index
рwТуре	pointer strors minimum IR value

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

## 5.1.4.21 APC\_GetLogData()

```
int APC_GetLogData (
    void * pHandleApcDI,
    PDEVSELINFO pDevSelInfo,
    BYTE * buffer,
    int BufferLength,
    int * pActualLength,
    int index )
```

## get log data from flash

#### **Parameters**

pHandleApcDI	the pointer to the initilized ApcDI SDK instance
pDevSelInfo	pointer of device select index
buffer	buffer to store log data
BufferLength	input buffer length
pActualLength	actual length has written to buffer
index	index to identify log data for corresponding depth

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

## 5.1.4.22 APC\_GetPidVid()

## get PID(product ID) and VID(vendor ID) of device

## **Parameters**

pHandleApcDI	CApcDI handler
pDevSelInfo	pointer of device select index
pPidBuf	4 byte buffer to store PID value
pVidBuf	4 byte buffer to store VID value

#### Returns

success: APC\_OK, others: see eSPDI\_ErrCode.h

#### 5.1.4.23 APC\_GetPUPropVal()

get processing unit property value <a href="https://msdn.microsoft.com/en-us/library/windows/hardware/ff568185">https://msdn.microsoft.com/en-us/library/windows/hardware/ff568185</a>) .aspx The PROPSETID\_VIDCAP\_VIDEOPROCAMP property set controls devices that can adjust image color attributes of analog or digital signals.

The KSPROPERTY\_VIDCAP\_VIDEOPROCAMP enumeration in ksmedia.h specifies the properties of this set.

Support for this property set is optional and should be implemented only by devices that allow adjustment of brightness, contrast, hue, and other image quality settings.

Prior to USB video class, this enumeration contained the following property items: KSPROPERTY\_← VIDEOPROCAMP\_BACKLIGHT\_COMPENSATION KSPROPERTY\_VIDEOPROCAMP\_BRIGHTNESS KSPROPERTY← \_VIDEOPROCAMP\_COLORENABLE KSPROPERTY\_VIDEOPROCAMP\_CONTRAST KSPROPERTY\_← VIDEOPROCAMP\_GAMMA KSPROPERTY\_VIDEOPROCAMP\_HUE KSPROPERTY\_VIDEOPROCAMP\_← SATURATION KSPROPERTY\_VIDEOPROCAMP\_SHARPNESS KSPROPERTY\_VIDEOPROCAMP\_WHITEBALANCE KSPROPERTY\_VIDEOPROCAMP\_GAIN

https://msdn.microsoft.com/en-us/library/windows/hardware/ff566089(v=vs.  $\leftarrow$  85).aspx The KSPROPERTY\_VIDEOPROCAMP\_S structure describes filter-based property settings in the PROPSETID VIDCAP VIDEOPROCAMP property set.

#### **Parameters**

pHandleApcDl	CApcDI handler
pDevSelInfo	pointer of device select index
nld	specifies the member of the property set
pValue	pointer of store PU property value

#### Returns

success: APC\_OK, others: see eSPDI\_ErrCode.h

#### 5.1.4.24 APC GetSensorRegister()

get value from sensor register

#### **Parameters**

pHandleApcDI	CApcDI handler
pDevSelInfo	pointer of device select index
nld	sensor slave address. see SENSOR_TYPE_NAME enum definition
address	register address
pValue	pointer of value got from register address
flag	address and value data length(2 or 1 byte) ie FG_Address_2Byte   FG_Value_2Byte is 2 byte address and 2 byte value #define FG_Address_1Byte 0x01 #define FG_Address_2Byte 0x02 #define FG_Value_1Byte 0x10 #define FG_Value_2Byte 0x20
nSensorMode	sensor mode(sensor A, B or Both) A is 0, B is 1, Both is 2

### Returns

success: APC\_OK, others: see eSPDI\_ErrCode.h

### 5.1.4.25 APC\_GetSlaveHWRegister()

### get hardware register value

#### **Parameters**

pHandleApcDI	CApcDI handler
pDevSelInfo	pointer of device select index
address	register address
pValue	pointer of value got from register address
flag	address and value data length(2 or 1 byte) ie FG_Address_2Byte   FG_Value_2Byte is 2 byte address and 2 byte value #define FG_Address_1Byte 0x01 #define FG_Address_2Byte 0x02 #define FG_Value_1Byte 0x10 #define FG_Value_2Byte 0x20

### Returns

success: APC\_OK, others: see eSPDI\_ErrCode.h

### 5.1.4.26 APC\_GetSlaveLogData()

```
PDEVSELINFO pDevSelInfo,
BYTE * buffer,
int BufferLength,
int * pActualLength,
int index )
```

### get log data from flash

#### **Parameters**

pHandleApcDI	the pointer to the initilized ApcDI SDK instance
pDevSelInfo	pointer of device select index
buffer	buffer to store log data
BufferLength	input buffer length
pActualLength	actual length has written to buffer
index	index to identify log data for corresponding depth

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

### 5.1.4.27 APC\_GetSlaveSensorRegister()

### get value from sensor register

pHandleApcDI	CApcDI handler
pDevSelInfo	pointer of device select index
nld	sensor slave address. see SENSOR_TYPE_NAME enum definition
address	register address
pValue	pointer of value got from register address
flag	address and value data length(2 or 1 byte) ie FG_Address_2Byte   FG_Value_2Byte is 2 byte address and 2 byte value #define FG_Address_1Byte 0x01 #define FG_Address_2Byte 0x02 #define FG_Value_1Byte 0x10 #define FG_Value_2Byte 0x20
nSensorMode	sensor mode(sensor A, B or Both) A is 0, B is 1, Both is 2

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

### 5.1.4.28 APC\_Init()

entry point of eYs3D camera SDK. This API allocates resource and find all the eSPI camera devices connected to the system.

#### **Parameters**

ppHandleApcDI	a pointer of pointer to receive ApcDI SDK instance
blsLogEnabled	set to true to generate log file, named log.txt in current folder

#### Returns

success: none negative integer to indicate numbers of devices found in the system.

#### 5.1.4.29 APC\_Init2()

entry point of eYs3D camera SDK. This API allocates resource and find all the eSPI camera devices connected to the system.

#### **Parameters**

ppHandleApcDI	a pointer of pointer to receive ApcDI SDK instance
blsLogEnabled	set to true to generate log file, named log.txt in current folder
bEnableAutoRestart	set true to auto-restart the device if the device was detached and attached again.

### Returns

success: none negative integer to indicate numbers of devices found in the system.

### Note

Calls APC\_Init or APC\_Init2 to initilize the ApcDI SDK. APC\_Init2 adds the auto-restart function to the initilization options. If you call APC\_Init, the bEnableAutoRestart is set as disabled.

### 5.1.4.30 APC\_Is360Device()

check module is spherical device or not

#### **Parameters**

pHandleApcDI	the pointer to the initilized ApcDI SDK instance
pDevSelInfo	pointer of device select index

#### Returns

true: module support 360, false: not support

### 5.1.4.31 APC\_OpenDevice()

open camera device with image callback support

pHandleApcDI	the pointer to the initilized ApcDI SDK insta	anco	
' '	·		
pDevSelInfo	pointer of device select index		
colorStreamIndex	index of the desired color stream		
depthStreamIndex	index of the desired sdepth tream		
depthStreamSwitch	depth switch for S0, S1 or S2		
iFps	pointer to the desired frame rate, returns the actual frame rate.		
callbackFn	set image callback function		
pCallbackParam	the data to associate with the callback function		
pid	Specify device pid.		
	Table 5.34 Image Control Mode		
	Mode	Description	
	0x01	color and depth frame output	
		synchrously, for depth map module only	
	0x02	enable post-process, for Depth Map	
		module only	
	0x04	stitch images if this bit is set, for fisheye	
	0x04	stitch images if this bit is set, for fisheye spherical module only	
Generated by Doxygen	0x04 0x08		

#### Returns

```
success:APC_OK, others:see eSPDI_ErrCode.h
```

### 5.1.4.32 APC\_ReadFlashData()

read firmware code(.bin) form flash The firmware code is the combination of boot loader, firmware body and plug-in data. This input buffer length has to match with the flash data type

#### **Parameters**

pHandleApcDI	the pointer to the initilized ApcDI SDK instance
pDevSelInfo	pointer of device select index
fdt	segment type of flash be read
pBuffer	buffer to store firmware code
nLengthOfBuffer	input buffer length
pActualBufferLen	actual length has written to pBuffer

#### Returns

success: APC\_OK, others: see eSPDI\_ErrCode.h

### 5.1.4.33 APC\_RefreshDevice()

```
int APC_RefreshDevice (
     void * pHandleApcDI )
```

refresh all eYs3D UVC devices

#### **Parameters**

pHandleApcDI   the pointer to the initilized ApcDI SDK instance
---

#### Returns

success: APC\_OK, others: see eSPDI\_ErrCode.h

#### 5.1.4.34 APC\_RegisterDeviceEvents()

```
int APC_RegisterDeviceEvents (
     void * pHandleApcDI,
     APC_DeviceEventFn cbFunc,
     void * pData )
```

Register the USB device plug or unplug events. Any USB capture device attachment or detachment events will call the callback function cbFunc.

#### **Parameters**

pHandleApcDI	a pointer to ApcDI SDK instance
cbFunc	a callback function of type #APC_DeviceEventFn that will receive USB cappure device events when the device is attached or detached.
pData	user defined data which will send to the callback function

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

#### 5.1.4.35 APC\_Release()

release all resource that APC\_Init had allocated

#### **Parameters**

ppHandleApcDI	pointer of the pointer to the initilized ApcDI SDK instance.
---------------	--

#### Returns

none

#### Note

the pointer to ppHandleApcDI will be set to NULL when this call returns successfully.

### 5.1.4.36 APC\_SetCTPropVal()

```
int nId,
int nValue )
```

get control terminal property value https://msdn.microsoft.com/en-us/library/windows/hardware/ff5678
85).aspx https://msdn.microsoft.com/en-us/library/windows/hardware/ff566089(v=vs.↔
85).aspx

#### **Parameters**

pHandleApcDI	CApcDI handler
pDevSelInfo	pointer of device select index
nld	specifies the member of the property set
nValue	CT property value to set

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

### 5.1.4.37 APC\_SetCurrentlRValue()

set current infrared radiation(IR) value

#### **Parameters**

pHandleApcDI	CApcDI handler
pDevSelInfo	pointer of device select index
wType	value to set

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

#### 5.1.4.38 APC\_SetFWRegister()

set firmware register value

#### **Parameters**

pHandleApcDI	CApcDI handler
pDevSelInfo	pointer of device select index
address	register address
nValue	register value to set
flag	address and value data length(2 or 1 byte) ie FG_Address_1Byte   FG_Value_1Byte is 1 byte address and 1 byte value #define FG_Address_1Byte 0x01 #define FG_Address_2Byte 0x02 #define FG_Value_1Byte 0x10 #define FG_Value_2Byte 0x20

#### Returns

success: APC\_OK, others: see eSPDI\_ErrCode.h

### 5.1.4.39 APC\_SetGPIOCtrl()

### set GPIO control address

### **Parameters**

nGPIOIndex	index of GPIO (1 $\sim$ 4)
nValue	register value to set

#### Returns

success: APC\_OK, others: see eSPDI\_ErrCode.h

### 5.1.4.40 APC\_SetGPIOValue()

### set GPIO value

#### **Parameters**

pHandleApcDI	CApcDI handler
pDevSelInfo	pointer of device select index
nGPIOIndex	GPIO index, 1 or 2 is valid
nValue	GPIO value to set

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

### 5.1.4.41 APC\_SetHuffmanTableData()

set huffman table data for jpeg encode

#### **Parameters**

pHandleApcDl	CApcDI handler
pDevSelInfo	pointer of device select index
filename	huffman table file, see jh_vga_422.dat sample file
bLogFile	if true then puma_htable.dat file is generated

### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

### 5.1.4.42 APC\_SetHWRegister()

### set hardware register

#### **Parameters**

pHandleApcDI	CApcDI handler
pDevSelInfo	pointer of device select index
address	register address
nValue	register value to set
flag	address and value data length(2 or 1 byte) ie FG_Address_1Byte   FG_Value_1Byte is 1 byte address and 1 byte value #define FG_Address_1Byte 0x01 #define FG_Address_2Byte 0x02 #define FG_Value_1Byte 0x10 #define FG_Value_2Byte 0x20

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

### 5.1.4.43 APC\_SetIRMaxValue()

set maximum IR value the module support

### **Parameters**

pHandleApcDI	the pointer to the initilized ApcDI SDK instance
pDevSelInfo	pointer of device select index
wType	pointer strors maximum IR value

### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

### 5.1.4.44 APC\_SetLogData()

set log data to flash

#### **Parameters**

pHandleApcDI	the pointer to the initilized ApcDI SDK instance
pDevSelInfo	pointer of device select index
buffer	buffer to store log data
BufferLength	input buffer length
pActualLength	actual length has written to buffer
index	index to identify log data for corresponding depth

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

### 5.1.4.45 APC\_SetPidVid()

#### set PID and VID to device

#### **Parameters**

pHandleApcDI	CApcDI handler
pDevSelInfo	pointer of device select index
pPidBuf	4 byte PID value buffer to set
pVidBuf	4 byte VID value buffer to set

### Returns

```
success: EtronDI_OK, others: see eSPDI_ErrCode.h
```

### 5.1.4.46 APC\_SetPUPropVal()

get processing unit property value https://msdn.microsoft.com/en-us/library/windows/hardware/ff5681
85).aspx https://msdn.microsoft.com/en-us/library/windows/hardware/ff566089(v=vs.↔
85).aspx

#### **Parameters**

pHandleApcDI	CApcDI handler
pDevSelInfo	pointer of device select index
nld	specifies the member of the property set
nValue	value to set

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

## 5.1.4.47 APC\_SetQuantizationTableData()

set quantication table data for jpeg encode

#### **Parameters**

pHandleApcDI	CApcDI handler
pDevSelInfo	pointer of device select index
filename	quantization table file, see FS_DEF_010.txt sample file

### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

### 5.1.4.48 APC\_SetSensorTypeName()

select which sensor to operate

pHandleApcDI	CApcDI handler	
stn	sensor type	

#### Returns

APC\_OK

### 5.1.4.49 APC\_SetSlaveHWRegister()

#### set hardware register

#### **Parameters**

pHandleApcDI	CApcDI handler
pDevSelInfo	pointer of device select index
address	register address
nValue	register value to set
flag	address and value data length(2 or 1 byte) ie FG_Address_1Byte   FG_Value_1Byte is 1 byte address and 1 byte value #define FG_Address_1Byte 0x01 #define FG_Address_2Byte 0x02 #define FG_Value_1Byte 0x10 #define FG_Value_2Byte 0x20

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

### 5.1.4.50 APC\_SetSlaveLogData()

```
int APC_SetSlaveLogData (
          void * pHandleApcDI,
          PDEVSELINFO pDevSelInfo,
          BYTE * buffer,
          int BufferLength,
          int * pActualLength,
          int index )
```

### set log data to flash

pHandleApcDI	the pointer to the initilized ApcDI SDK instance
pDevSelInfo	pointer of device select index
buffer	buffer to store log data
BufferLength	input buffer length
pActualLength	actual length has written to buffer
index	index to identify log data for corresponding depth

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

#### 5.1.4.51 APC\_SetUserData()

set user data to flash

#### **Parameters**

pHandleApcDI	the pointer to the initilized ApcDI SDK instance
pDevSelInfo	pointer of device select index
buffer	buffer to store user data
BufferLength	input buffer length
usi	which user index data to select

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

# 5.2 eSPDI\_DM.h File Reference

eYs3D SDK API export functions, data structure and variable definition for depth map module

#### **Functions**

• int APC\_API APC\_GetSlaveYOffset (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, BYTE \*buffer, int BufferLength, int \*pActualLength, int index)

```
get Y offset data
```

• int APC\_API APC\_GetYOffset (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, BYTE \*buffer, int Buffer ← Length, int \*pActualLength, int index)

get Y offset data

• int APC\_API APC\_GetSlaveRectifyTable (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, BYTE \*buffer, int BufferLength, int \*pActualLength, int index)

get rectify values from flash

• int APC\_API APC\_GetRectifyTable (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, BYTE \*buffer, int BufferLength, int \*pActualLength, int index)

get rectify values from flash

• int APC\_API APC\_GetZDTable (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, BYTE \*buffer, int BufferLength, int \*pActualLength, PZDTABLEINFO pZDTableInfo)

get disparity and Z values from flash

• int APC\_API APC\_SetSlaveYOffset (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, BYTE \*buffer, int BufferLength, int \*pActualLength, int index)

set Y offset data

int APC\_API APC\_SetYOffset (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, BYTE \*buffer, int Buffer←
Length, int \*pActualLength, int index)

set Y offset data

• int APC\_API APC\_SetSlaveRectifyTable (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, BYTE \*buffer, int BufferLength, int \*pActualLength, int index)

set rectify data to flash, see APC\_SetRectifyTable except set

• int APC\_API APC\_SetRectifyTable (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, BYTE \*buffer, int BufferLength, int \*pActualLength, int index)

set rectify data to flash, see APC\_SetRectifyTable except set

• int APC\_API APC\_SetZDTable (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, BYTE \*buffer, int BufferLength, int \*pActualLength, PZDTABLEINFO pZDTableInfo)

set disparity and Z values to flash, see APC\_GetZDTable except get

• int APC\_API APC\_GetRectifyMatLogData (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, eSPCtrl\_RectLogData \*pData, int index)

get rectify log data from flash for Puma IC

- int APC\_API APC\_SetDepthDataType (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, WORD wType) set depth data type, 11 bit for disparity data, 14 bit for Z data notice: only PUMA type IC can support this setting
- int APC\_API APC\_GetDepthDataType (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, WORD \*pwType) get current depth data type setting
- int APC\_API APC\_SetHWPostProcess (void \*pHandleApcDI, PDEVSELINFO pDevSelInfo, bool enable) enable or disable internal chip post processing function

#### 5.2.1 Detailed Description

eYs3D SDK API export functions, data structure and variable definition for depth map module

Copyright

This file copyright (C) 2017 by eYs3D company

An unpublished work. All rights reserved. This file is proprietary information, and may not be disclosed or copied without the prior permission of eYs3D company.

### 5.2.2 Function Documentation

### 5.2.2.1 APC\_GetDepthDataType()

get current depth data type setting

#### **Parameters**

pHandleApcDI	the pointer to the initilized ApcDI SDK instance	
pDevSelInfo	pointer of device select index	
рwТуре	pointer of current depth data type in device	

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

### 5.2.2.2 APC\_GetRectifyMatLogData()

get rectify log data from flash for Puma IC

#### **Parameters**

pHandleApcDI	the pointer to the initilized ApcDI SDK instance
pDevSelInfo	pointer of device select index
pData	rectify log data, its buffer size is 4096 bytes see eSPCtrl_RectLogData for detailed members
index	index to identify rectify log data for corresponding depth

### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

### 5.2.2.3 APC\_GetRectifyTable()

```
APC_GetRectifyTable (
    void * pHandleApcDI,
    PDEVSELINFO pDevSelInfo,
    BYTE * buffer,
    int BufferLength,
    int * pActualLength,
    int index )
```

get rectify values from flash

#### **Parameters**

pHandleApcDI	the pointer to the initilized ApcDI SDK instance
pDevSelInfo	pointer of device select index
buffer	buffer to store rectify table data
BufferLength	input buffer length
pActualLength	actual length has written to buffer
index	index to identify rectify table for corresponding depth

#### Returns

```
success:APC_OK, others: see eSPDI_ErrCode.h
```

### 5.2.2.4 APC\_GetSlaveRectifyTable()

### get rectify values from flash

### **Parameters**

pHandleApcDl	the pointer to the initilized ApcDI SDK instance
pDevSelInfo	pointer of device select index
buffer	buffer to store rectify table data
BufferLength	input buffer length
pActualLength	actual length has written to buffer
index	index to identify rectify table for corresponding depth

#### Returns

```
success:APC_OK, others: see eSPDI_ErrCode.h
```

### 5.2.2.5 APC\_GetSlaveYOffset()

```
int BufferLength,
int * pActualLength,
int index )
```

#### get Y offset data

#### **Parameters**

pHandleApcDI	the pointer to the initilized ApcDI SDK instance
pDevSelInfo	pointer of device select index
buffer	buffer to store
BufferLength	length of buffer
pActualLength	actual byte of reading
index	index of Y offset file ID

#### Returns

```
success:APC_OK, others:see eSPDI_ErrCode.h
```

### 5.2.2.6 APC\_GetYOffset()

### get Y offset data

#### **Parameters**

pHandleApcDI	the pointer to the initilized ApcDI SDK instance
pDevSelInfo	pointer of device select index
buffer	buffer to store
BufferLength	length of buffer
pActualLength	actual byte of reading
index	index of Y offset file ID

#### Returns

```
success:APC_OK, others:see eSPDI_ErrCode.h
```

### 5.2.2.7 APC\_GetZDTable()

```
int APC_GetZDTable (
    void * pHandleApcDI,
    PDEVSELINFO pDevSelInfo,
    BYTE * buffer,
    int BufferLength,
    int * pActualLength,
    PZDTABLEINFO pZDTableInfo )
```

get disparity and Z values from flash

- 1. if depth data type is APC\_DEPTH\_DATA\_14\_BITS then just get Z value from depth buffer
- 2. if depth data type is APC\_ZD\_TABLE\_FILE\_SIZE\_11\_BITS then using depth buffer value as a index to get Z value inside ZD table
- 3. see GetZValue() of example.c to get Z value from different depth data type

#### **Parameters**

pHandleApcDI	the pointer to the initilized ApcDI SDK instance
pDevSelInfo	pointer of device select index
buffer	bufer to store ZD table
BufferLength	input buffer length
pActualLength	actual length has written to buffer
pZDTableInfo	index to identify ZD table and data type for corrresponding depth

### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

#### 5.2.2.8 APC\_SetDepthDataType()

set depth data type, 11 bit for disparity data, 14 bit for Z data notice: only PUMA type IC can support this setting

pHandleApcDI	the pointer to the initilized ApcDI SDK instance
pDevSelInfo	pointer of device select index
wТуре	depth data type you want to set, see APC_DEPTH_DATA_xxx in APC_O.h \output success: APC_OK, others: see eSPDI_ErrCode.h

### 5.2.2.9 APC\_SetHWPostProcess()

enable or disable internal chip post processing function

#### **Parameters**

pHandleApcDI	the pointer to the initilized ApcDI SDK instance
pDevSelInfo	pointer of device select index
enable	set true to enable post-process, or set false to disable post-process

#### Returns

```
success: APC_OK, others: see eSPDI_ErrCode.h
```

### 5.2.2.10 APC\_SetSlaveYOffset()

```
int APC_SetSlaveYOffset (
          void * pHandleApcDI,
          PDEVSELINFO pDevSelInfo,
          BYTE * buffer,
          int BufferLength,
          int * pActualLength,
          int index )
```

### set Y offset data

#### **Parameters**

pHandleApcDI	the pointer to the initilized ApcDI SDK instance	
pDevSelInfo	pointer of device select index	
buffer	buffer to store	
BufferLength	length of buffer	
pActualLength	actual byte of reading	
index	index of Y offset file ID	

#### Returns

```
success:APC_OK, others:see eSPDI_ErrCode.h
```

### 5.2.2.11 APC\_SetYOffset()

```
int APC_SetYOffset (
          void * pHandleApcDI,
          PDEVSELINFO pDevSelInfo,
          BYTE * buffer,
          int BufferLength,
          int * pActualLength,
          int index )
```

#### set Y offset data

#### **Parameters**

pHandleApcDI	the pointer to the initilized ApcDI SDK instance	
pDevSelInfo	pointer of device select index	
buffer	buffer to store	
BufferLength	length of buffer	
pActualLength	actual byte of reading	
index	index of Y offset file ID	

#### Returns

success:APC\_OK, others:see eSPDI\_ErrCode.h

# 5.3 eSPDI\_ErrCode.h File Reference

definition of eYs3D SDK error code Copyright: This file copyright (C) 2017 by

### 5.3.1 Detailed Description

definition of eYs3D SDK error code Copyright: This file copyright (C) 2017 by

eYs3D company

An unpublished work. All rights reserved.

This file is proprietary information, and may not be disclosed or copied without the prior permission of eYs3D company.

# Index

APC_CloseDevice	eSPDI_Common.h, 32
eSPDI_Common.h, 23	APC_GetRectifyMatLogData
APC_DEVICE_TYPE	eSPDI DM.h, 49
eSPDI_Common.h, 21	APC_GetRectifyTable
APC DisableAE	eSPDI_DM.h, 49
eSPDI_Common.h, 23	APC_GetSensorRegister
APC DisableAWB	eSPDI_Common.h, 33
eSPDI_Common.h, 23	APC_GetSlaveHWRegister
APC EnableAE	eSPDI_Common.h, 34
eSPDI_Common.h, 24	APC GetSlaveLogData
APC EnableAWB	eSPDI Common.h, 34
eSPDI_Common.h, 24	APC GetSlaveRectifyTable
APC_EnableGPUAcceleration	eSPDI_DM.h, 50
eSPDI_Common.h, 24	APC_GetSlaveSensorRegister
APC FindDevice	eSPDI_Common.h, 35
eSPDI_Common.h, 25	APC_GetSlaveYOffset
APC GetCTPropVal	eSPDI_DM.h, 50
eSPDI Common.h, 25	APC_GetYOffset
APC GetCurrentIRValue	eSPDI DM.h, 51
eSPDI Common.h, 26	APC_GetZDTable
APC_GetDepthDataType	eSPDI DM.h, 51
eSPDI DM.h, 48	APC Init
APC_GetDepthFilterVersion	eSPDI_Common.h, 36
eSPDI_Common.h, 26	APC_Init2
APC GetDeviceNumber	eSPDI_Common.h, 36
eSPDI_Common.h, 27	APC Is360Device
APC GetDeviceResolutionList	eSPDI_Common.h, 36
eSPDI_Common.h, 27	APC_OpenDevice
APC_GetFlexibleGyroData	eSPDI_Common.h, 37
eSPDI Common.h, 28	APC ReadFlashData
APC_GetFlexibleGyroLength	eSPDI_Common.h, 38
eSPDI_Common.h, 28	APC_RefreshDevice
APC_GetFWRegister	eSPDI_Common.h, 38
eSPDI_Common.h, 29	APC RegisterDeviceEvents
APC GetFwVersion	eSPDI Common.h, 38
eSPDI_Common.h, 29	APC Release
APC GetGPIOValue	<del>-</del>
eSPDI_Common.h, 30	eSPDI_Common.h, 39
	APC_SENSOR_TYPE_AR0134
APC_GetHWRegister	eSPDI_Common.h, 22
eSPDI_Common.h, 30	APC_SENSOR_TYPE_AR0135
APC_GetIRMaxValue	eSPDI_Common.h, 22
eSPDI_Common.h, 31	APC_SENSOR_TYPE_AR0144
APC_GetIRMinValue	eSPDI_Common.h, 22
eSPDI_Common.h, 31	APC_SENSOR_TYPE_AR0330
APC_GetLogData	eSPDI_Common.h, 22
eSPDI_Common.h, 31	APC_SENSOR_TYPE_AR1335
APC_GetPidVid	eSPDI_Common.h, 22
eSPDI_Common.h, 32	APC_SENSOR_TYPE_H22
APC_GetPUPropVal	eSPDI_Common.h, 22

56 INDEX

APC_SENSOR_TYPE_OV7740	eSPCtrl_RectLogData, 10
eSPDI_Common.h, 22	eSPDI_Common.h, 21
APC_SENSOR_TYPE_OV9282	eSPDI_Common.h, 17
eSPDI_Common.h, 22	APC_CloseDevice, 23
APC_SENSOR_TYPE_OV9714	APC_DEVICE_TYPE, 21
eSPDI_Common.h, 22	APC_DisableAE, 23
APC_SetCTPropVal	APC DisableAWB, 23
eSPDI_Common.h, 39	APC EnableAE, 24
APC SetCurrentIRValue	APC EnableAWB, 24
eSPDI_Common.h, 40	APC_EnableGPUAcceleration, 24
APC_SetDepthDataType	APC FindDevice, 25
eSPDI_DM.h, 52	APC_GetCTPropVal, 25
APC_SetFWRegister	APC_GetCurrentIRValue, 26
eSPDI_Common.h, 40	APC_GetDepthFilterVersion, 26
APC SetGPIOCtrl	APC_GetDeviceNumber, 27
eSPDI_Common.h, 41	APC_GetDeviceResolutionList, 27
	APC GetFlexibleGyroData, 28
APC_SetGPIOValue	<del>_</del>
eSPDI_Common.h, 41	APC_GetFlexibleGyroLength, 28
APC_SetHuffmanTableData	APC_GetFWRegister, 29
eSPDI_Common.h, 42	APC_GetFwVersion, 29
APC_SetHWPostProcess	APC_GetGPIOValue, 30
eSPDI_DM.h, 52	APC_GetHWRegister, 30
APC_SetHWRegister	APC_GetIRMaxValue, 31
eSPDI_Common.h, 42	APC_GetIRMinValue, 31
APC_SetIRMaxValue	APC_GetLogData, 31
eSPDI_Common.h, 43	APC_GetPidVid, 32
APC_SetLogData	APC_GetPUPropVal, 32
eSPDI_Common.h, 43	APC_GetSensorRegister, 33
APC_SetPidVid	APC_GetSlaveHWRegister, 34
eSPDI_Common.h, 44	APC_GetSlaveLogData, 34
APC_SetPUPropVal	APC_GetSlaveSensorRegister, 35
eSPDI_Common.h, 44	APC_Init, 36
APC_SetQuantizationTableData	APC_Init2, 36
eSPDI_Common.h, 45	APC_Is360Device, 36
APC_SetSensorTypeName	APC_OpenDevice, 37
eSPDI_Common.h, 45	APC_ReadFlashData, 38
APC_SetSlaveHWRegister	APC_RefreshDevice, 38
eSPDI_Common.h, 46	APC_RegisterDeviceEvents, 38
APC_SetSlaveLogData	APC Release, 39
eSPDI Common.h, 46	APC SENSOR TYPE AR0134, 22
APC_SetSlaveYOffset	APC_SENSOR_TYPE_AR0135, 22
eSPDI DM.h, 53	APC_SENSOR_TYPE_AR0144, 22
APC_SetUserData	APC_SENSOR_TYPE_AR0330, 22
eSPDI Common.h, 47	APC_SENSOR_TYPE_AR1335, 22
APC_SetYOffset	APC_SENSOR_TYPE_H22, 22
eSPDI DM.h, 53	APC SENSOR TYPE OV7740, 22
AXES1	APC_SENSOR_TYPE_OV9282, 22
eSPDI Common.h, 22	APC_SENSOR_TYPE_OV9714, 22
33. 2. <u>_</u> 33, <u></u>	APC SetCTPropVal, 39
DEVINFORMATION	APC_SetCurrentIRValue, 40
eSPDI Common.h, 21	APC SetFWRegister, 40
DEVINFORMATIONEX, 9	APC SetGPIOCtrl, 41
nChipID, 9	APC SetGPIOValue, 41
nDevType, 9	APC_SetHuffmanTableData, 42
strDevName, 9	APC SetHWRegister, 42
wPID, 10	APC SetIRMaxValue, 43
wUsbNode, 10	<del>-</del>
wVID, 10	APC_SetDidVid_44
	APC_SetPidVid, 44

INDEX 57

APC_SetPUPropVal, 44 APC_SetQuantizationTableData, 45	DEVINFORMATIONEX, 9 tagDEVINFORMATION, 14
APC_SetSensorTypeName, 45 APC_SetSlaveHWRegister, 46	tagDEVINFORMATION, 13
APC_SetSlaveLogData, 46	nChipID, 14
APC_SetUserData, 47	nDevType, 14
AXES1, 22	strDevName, 14
DEVINFORMATION, 21	wPID, 14
eSPCtrl_RectLogData, 21	wUsbNode, 15
OTHERS, 22	wVID, 15
PARALUT, 21	
PUMA, 22	USERDATA_SECTION_0
SENSOR_TYPE_NAME, 22	eSPDI_Common.h, 22
USERDATA_SECTION_0, 22	USERDATA_SECTION_1
USERDATA_SECTION_1, 22	eSPDI_Common.h, 22
USERDATA_SECTION_10, 22	USERDATA_SECTION_10
USERDATA_SECTION_2, 22	eSPDI_Common.h, 22
USERDATA_SECTION_3, 22	USERDATA_SECTION_2
USERDATA_SECTION_4, 22	eSPDI_Common.h, 22
USERDATA_SECTION_5, 22	USERDATA_SECTION_3
USERDATA_SECTION_6, 22	eSPDI_Common.h, 22
USERDATA_SECTION_7, 22	USERDATA_SECTION_4
USERDATA_SECTION_8, 22	eSPDI_Common.h, 22
USERDATA_SECTION_9, 22	USERDATA_SECTION_5 eSPDI_Common.h, 22
USERDATA_SECTION_INDEX, 22	USERDATA_SECTION_6
USERDATA_SECTION_NUM, 22	eSPDI_Common.h, 22
eSPDI_DM.h, 47	USERDATA SECTION 7
APC_GetDepthDataType, 48	eSPDI Common.h, 22
APC_GetRectifyMatLogData, 49	USERDATA SECTION 8
APC_GetRectifyTable, 49	eSPDI Common.h, 22
APC_GetSlaveRectifyTable, 50	USERDATA SECTION 9
APC_GetSlaveYOffset, 50	eSPDI_Common.h, 22
APC_Get7DTeble_51	USERDATA SECTION INDEX
APC_SetDonth Data Type 53	eSPDI Common.h, 22
APC_SetDepthDataType, 52 APC SetHWPostProcess, 52	USERDATA_SECTION_NUM
APC SetSlaveYOffset, 53	eSPDI_Common.h, 22
APC_Set/Offset, 53  APC_Set/Offset, 53	_ ,
eSPDI ErrCode.h, 54	wPID
eor bi_Eiroude.ii, 54	DEVINFORMATIONEX, 10
nChipID	tagDEVINFORMATION, 14
DEVINFORMATIONEX, 9	wUsbNode
tagDEVINFORMATION, 14	DEVINFORMATIONEX, 10
nDevType	tagDEVINFORMATION, 15
DEVINFORMATIONEX, 9	wVID
tagDEVINFORMATION, 14	DEVINFORMATIONEX, 10
	tagDEVINFORMATION, 15
OTHERS eSPDI_Common.h, 22	
DADALUT	
PARALUT	
eSPDI_Common.h, 21	
ParaLUT, 11	
PUMA	
eSPDI_Common.h, 22	
SENSOR_TYPE_NAME	
eSPDI_Common.h, 22	
strDevName	