

USB3.0 SDK Development Manual

Version	Reviser	Overview
V1.0	Fan Sh	Edit fuction interface and data type definition
V1.1	Fan Sh	Add interface “GetDeviceList” and “OpenStreamByDevID”

Catalog

1. Fuction Interface	3
1.1 Initialize()	3
1.2 Exit()	3
1.3 GetDeviceList()	3
1.4 OpenStream()	4
1.5 OpenStreamByDevID()	4
1.6 CloseStream()	5
1.7 SetPalette()	5
2. Data Type Definition	6
2.1 enum guide_usb_code_e	6
2.2 struct guide_usb_device_info_t	6
2.3 enum guide_usb_video_mode_e	7
2.4 struct guide_usb_frame_data_t	7
2.5 OnFrameDataReceivedCB	8
2.6 enum guide_usb_device_status_e	8
2.7 OnDeviceConnectStatusCB	8
2.8 struct device_info device_info_list	9

1. Fuction Interface

1.1 Initialize()

Prototype

int Initialize()

Description

Initialize module, invoke once when the software boot up.

Return

1 Success <0 Failure

1.2 Exit()

Prototype

int Exit()

Description

Uninitialize module, invoke once when the software exit.

Return

1 Success <0 Failure

1.3 GetDeviceList()

Prototype

int GetDeviceList ()

Description

Get device list.

parameter

devInfos: Include device number, device ID, device name.

Return

1 Success <0 Failure

1.4 OpenStream()

Prototype

```
int OpenStream(guide_usb_device_info_t *deviceInfo,  
               OnFrameDataReceivedCB frameRecvCB,  
               OnDeviceConnectStatusCB connectStatusCB)
```

Description

Open the video stream, the first device is default.

parameter

deviceInfo: Include width, height and video mode.

frameRecvCB: Video stream callback.

connectStatusCB: Connection status callback.

Return

1 Success <0 Failure

1.5 OpenStreamByDevID()

Prototype

```
int OpenStream(int devID,  
               guide_usb_device_info_t *deviceInfo,  
               OnFrameDataReceivedCB frameRecvCB,  
               OnDeviceConnectStatusCB connectStatusCB)
```

Description

Open the video stream by device id.

parameter

devID: device ID, from GetDeviceList.

deviceInfo: Include width, height and video mode.

frameRecvCB: Video stream callback.

connectStatusCB: Connection status callback.

Return

1 Success <0 Failure

1.6 CloseStream()

Prototype

int CloseStream()

Description

Close the video stream.

Return

1 Success <0 Failure

1.7 SetPalette()

Prototype

int SetPalette(int index)

Description

Set the color.

parameter

index: from 0 to 9, 10 color in total, 0 is default.

Return

1 Success <0 Failure

2. Data Type Definition

2.1 enum guide_usb_code_e

Definition

typedef enum

```
{  
    ERROR_NO = 1,                //1:no error  
    ERROR_DEVICE_NOT_FOUND = -1, // -1:can't find device  
    ERROR_POINT_NULL = -2,       //-2:null point  
    ERROR_POINTS_TOO_LARGE = -3, //-3:point too large  
    ERROR_POINTS_TOO_SMALL = -4, //-4: point too small  
    ERROR_MALLOC_FAILED = -5,    //-5:alloc memory failure  
    ERROR_RESOLUTION = -6,       //-6: set resolution error  
    ERROR_UNKNOW = -999          //-999:unknow error  
} guide_usb_code_e;
```

Description

Function return value.

2.2 struct guide_usb_device_info_t

Definition

typedef struct

```
{  
    int width;                //video width  
    int height;               //video height  
    guide_usb_video_mode_e video_mode; //video mode  
} guide_usb_device_info_t;
```

Description

Video info, be used for open device.

2.3 enum guide_usb_video_mode_e

Definition

typedef enum

```
{
    X16 = 0,                //X16
    X16_PARAM = 1,          //X16+paraline
    Y16 = 2,                //Y16
    Y16_PARAM = 3,          //Y16+paraline
    YUV = 4,                //YUV
    YUV_PARAM = 5,          //YUV+paraline
    Y16_YUV = 6,            //Y16+YUV
    Y16_PARAM_YUV = 7       //Y16+paraline+YUV
} guide_usb_video_mode_e;
```

Description

Video mode, according to the device configuration, input related type.

2.4 struct guide_usb_frame_data_t

Definition

typedef struct

```
{
    int frame_width;        //frame width
    int frame_height;       //frame height
    unsigned char *frame_rgb_data; //rgb data stream
    int frame_rgb_data_length; //rgb data length
    short *frame_src_data;  //X16/Y16 data stream
    int frame_src_data_length; //X16/Y16 data length
    short *frame_yuv_data;  //yuv data stream
    int frame_yuv_data_length; //yuv data length
    short *paraLine;        //paraline
    int paraLine_length;    //paraline length
} guide_usb_frame_data_t;
```

Description

Video frame data.

2.5 OnFrameDataReceivedCB

Definition

```
typedef void(__stdcall *OnFrameDataReceivedCB)(  
    const guide_usb_frame_data_t frame_data);
```

Description

Video stream callback.

2.6 enum guide_usb_device_status_e

Definition

```
typedef enum  
{  
    DEVICE_CONNECT_OK = 1,                //connection successful  
    DEVICE_DISCONNECT_OK = -1,            //connection unsuccessful  
} guide_usb_device_status_e;
```

Description

Device connection status.

2.7 OnDeviceConnectStatusCB

Definition

```
typedef void(__stdcall *OnDeviceConnectStatusCB)(  
    const guide_usb_device_status_e device_status);
```

Description

Connection status callback.

2.8 struct device_info device_info_list

Definition

typedef struct

```
{  
    int devID;                //device ID  
    char devName[128];        //device name  
} device_info;
```

typedef struct

```
{  
    int devCount;              //device number  
    device_info devs[32];      //device info  
} device_info_list;
```

Description

Device list info.