# USB3.0 SC-SDK
# User Guide

Documentation Version V2.0.0

Release Date:2025-03-01

# Revision records

The revision record accumulates instructions for each document update. The latest version of a document contains updated content from all previous document versions.

| Revision sheet | | | |
|---|---|---|---|
| **Revision No.** | **Date** | **Revision** | **Sig.** |
| V2.0.0 | 20250301 | Integrate existing SDK functions and release a new version | 11215wuc |

# Contents

# 1.Image & Communication (API)

## 1.1 API Interface

### 1.1.1 guide_usb_initialize

int guide_usb_initialize (const char* deviceName);

**Function Description**

Initialize USB device

**Input Parameter**

deviceName:device name,for example: /dev/video0

**Return Value**

>=0 success, <0 fail

### 1.1.2 guide_usb_openStream

int guide_usb_openStream(guide_usb_device_info_t* deviceInfo,OnFrameDataReceivedCB frameRecvCB,OnDeviceConnectStatusCB connectStatusCB);

**Function Description**

Open video stream

**Input Parameter**

deviceInfo:Device information, including image width and height, video mode, and device version (valid versions 1, 2, 3)

frameRecvCB:Video stream callback function

connectStatusCB:Connection status callback function

**Return Value**

>=0 success,<0 fail

### 1.1.3 guide_usb_closeStream

int guide_usb_closeStream();

**Function Description**

Close video stream

**Input Parameter**

None

**Return Value**

>=0 success,<0 fail

### 1.1.4 guide_usb_exit

int guide_usb_exit();

**Function Description**

Exit,clear data,free

**Input Parameter**

无

**Return Value**

>0 success,<0 fail

### 1.1.5 guide_usb_setLogLevel

int guide_usb_setLogLevel(int level);

**Function Description**

Set log switch and level

**Input Parameter**

level:log level,See also guide_usb_log_level_e.

**Return Value**

>0 success,<0 fail

## 1.2 Data Types

### 1.2.1 enum guide_usb_video_mode_e

**Type Definition**

typedef enum

```
{
    YUV = 0,                        //YUV
    YUV_PARAM = 1,                  //YUV +parameters
    Y16 = 2,                        //Y16
    Y16_PARAM = 3,                  //Y16+ parameters
    Y16_YUV = 4,                    //Y16+ YUV
    Y16_YUV_PARAM = 5,              //Y16+ YUV+parameters
    X16 = 6,                        //Not Supported
    X16_PARAM = 7,                  //Not Supported
    TMP = 8,
    TMP_PARAM = 9,
    TMP_YUV = 10,
    TMP_PARAM_YUV = 11
} guide_usb_video_mode_e;
```

**Functional Description**

Cameras video mode: The video mode parameters in the SDK code must be consistent with the movement video mode.

Only some cameras support modes 6-11, and video modes with TMP data are only supported by ASIC2.0 cameras.

**The application and difference of various modes:**

**YUV:**

Generally used for viewing movement, image data in YUV422 UYVY format.

**YUV_PARAM:**

Can be used for viewing movement and temperature measurement movement. If the temperature measuring movement does not require a full picture temperature matrix, then this mode is most suitable, YUV image data is used for display, and parameter row data can obtain the minimum, maximum, average and cursor point temperature of the region.

**Y16:**

This mode outputs pure 16-bit data for analysis.

**Y16_PARAM :**

This mode outputs pure 16-bit data and parameter row data, this mode can be used for temperature measurement movement.

**Y16_YUV:**

output pure 16-bit data and YUV data, can only be used for viewing the movement, but the bandwidth requirement for viewing the movement is twice that of YUV mode, this mode is not recommended.

**Y16_YUV_PARAM:**

can be used for both viewing movement and temperature measurement movement, in the case of bandwidth permitting, if the full picture temperature matrix is required, this mode is most suitable, if the full picture temperature matrix is not required, YUV_PARAM is more suitable.

**TMP:**

This mode outputs 16-bit temperature matrix data, and each 16-bit data needs /10 when parsing.

**TMP_PARAM:**

This mode outputs 16-bit temperature matrix data and parameter row data, and each 16-bit data needs /10 when analyzing the temperature matrix.

**TMP _YUV:**

Output 16-bit temperature matrix data and YUV data, each 16-bit data needs /10 when parsing the temperature matrix, YUV image data is used for display, this mode is not recommended.

**TMP_PARAM_YUV:**

Used for ASIC2.0 temperature measuring movement, output 16-bit temperature matrix data and YUV data, each 16-bit data needs /10 when analyzing the temperature matrix, YUV image data is used for display.

## 1.2.2 enum guide_usb_device_status_e

**Type Definition**

```
typedef enum
{
        DEVICE_CONNECT_OK = 1,                    //open
        DEVICE_DISCONNECT_OK = -1,                //close
 } guide_usb_device_status_e;
```

**Functional Description**

Camera video streaming status.

## 1.2.3 struct guide_usb_device_info_t

**Type Definition**

```
typedef struct
{
    int width;                                    //image width
    int height;                                   //image height
    guide_usb_video_mode_e video_mode;            //voide mode
    int device_version;            // The device version has only 1, 2, and 3 valid values
} guide_usb_device_info_t;
```

**Functional Description**

Camera video information, information required when opening the device, configured according to the device.

## 1.2.4 struct guide_usb_frame_data_t

**Type Definition**

```
typedef struct
{
        int frame_width;                          //image width
        int frame_height;                         //image height
        short* frame_src_data;                    //y16 data(ASIC2.0 :temperature data)
        int frame_src_data_length;    //y16 data length(ASIC2.0 :temperature data length)
        short* frame_yuv_data;                    //yuv422 data
        int frame_yuv_data_length;                //yuv422 data length
        short* paramLine;                         //parameters
        int paramLine_length;                     //parameters length
} guide_usb_frame_data_t;
```

**Functional Description**

Image related data.

**Y16 data:** frame_src_data, ASIC2.0 infrared camera temperature data TMP is obtained from frame_src_data.

**YUV data:** frame_yuv_data

### 1.2.5 enum guide_usb_log_level_e

**Type Definition**

```
typedef enum
{
    CLOSE = 0,                          //close log
    LOG_FATALEER = 1,
    LOG_ERROR = 3,
    LOG_WARN = 7,
    LOG_INFO = 15,
    LOG_TEST = 31,
} guide_usb_log_level_e;
```

**Functional Description**

Log level Settings.

### 1.2.6 OnDeviceConnectStatusCB

**Type Definition**

```
typedef int ( *OnDeviceConnectStatusCB)(guide_usb_device_status_e deviceStatus);
```

**Functional Description**

Video stream connection status callback function.

### 1.2.7 OnFrameDataReceivedCB

**Type Definition**

```
typedef int ( *OnFrameDataReceivedCB)(guide_usb_frame_data_t *pVideoData);
```

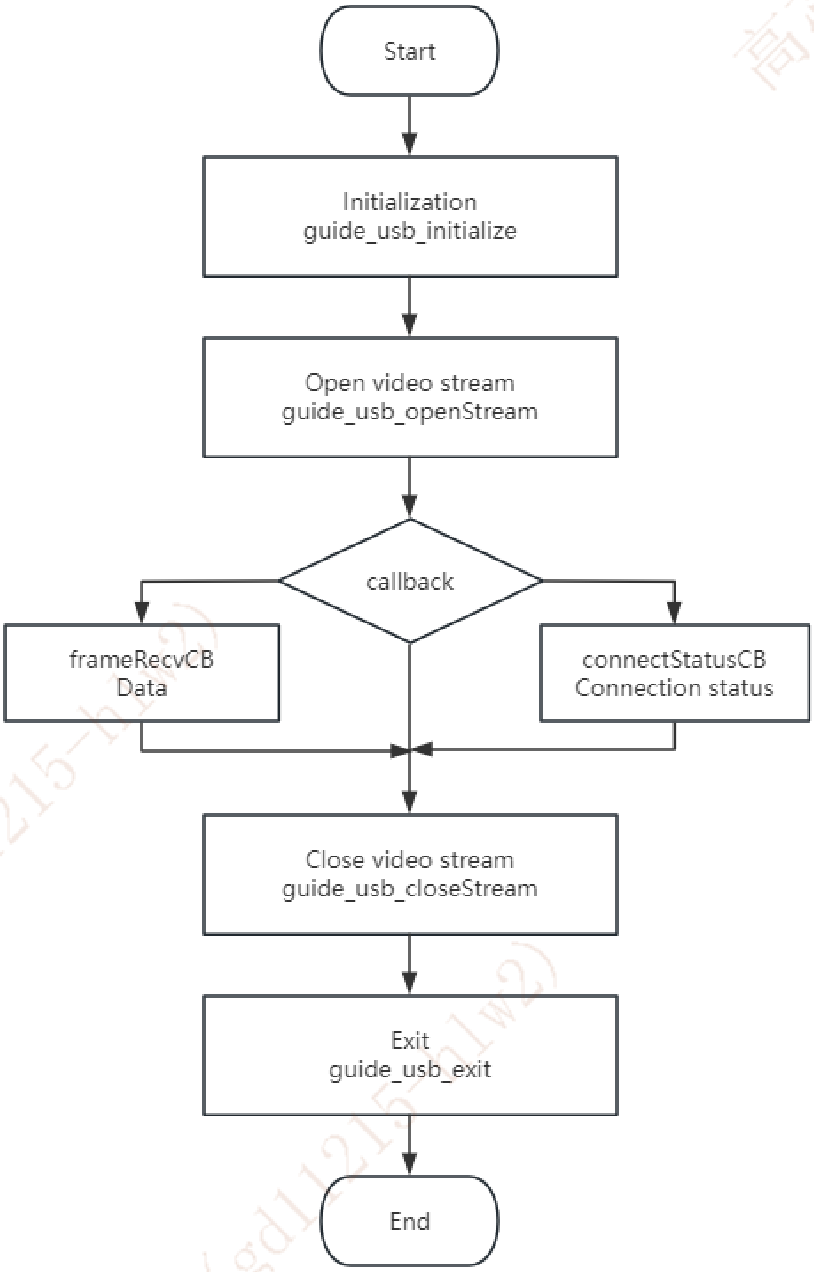
**Functional Description**

Video stream callback function.

# 2.GUIDE Lib Development

## 2.1 Development Process

### 2.1.1 Flow chart



# 2 parameters Protocol

PARAM data is a necessary parameter of the temperature measurement interface. In addition to being used for temperature measurement, this data also comes with some information.

| | Parameter line | | | | |
|---|---|---|---|---|---|
| **Index** | Serial number | Delivering content | | Bit width | Description |
| 0 | head1 | 0x55AA | | | Frame header |
| 1 | head2 | 0x0038 | | | Frame header |
| 2 | 1 | Correlation of temperature measurement | [15:0] | 16 | humidity tempsysctl.usrelum |
| 3 | 2 | | [15:0] | 16 | distance (True distance*10) |
| 4 | 3 | | [15:0] | 16 | emissivity |
| 5 | 4 | | [15:0] | 16 | reflectivity |
| 6~27 | 5~26 | reservation | [15:0] | 16 | reservation |
| 28 | 27 | Shutter status flag | [15:0] | 16 | 0: Not shooting the shutter, 1: shooting the shutter |
| 29~43 | 28~42 | reservation | [15:0] | 16 | reservation |
| 44 | 43 | Hot spot X coordinates | [15:0] | 16 | Enlarge the measured temperature by 10 times |
| 45 | 44 | Hot spot Y coordinates | [15:0] | 16 | |
| 46 | 45 | Hot spot temperature | [15:0] | 16 | |
| 47 | 46 | Coldest point X coordinate | [15:0] | 16 | |
| 48 | 47 | Coldest point Y coordinate | [15:0] | 16 | |
| 49 | 48 | Coldest point temperature | [15:0] | 16 | |
| 50 | 49 | The x-coordinate of the cursor point | [15:0] | 16 | |
| 51 | 50 | The Y-coordinate of the cursor point | [15:0] | 16 | |
| 52 | 51 | Temperature of the cursor point | [15:0] | 16 | |
| 53 | 52 | Regional mean temperature | [15:0] | 16 | |
| 54~58 | 53~57 | reservation | [15:0] | 16 | reservation |
| 59 | 58 | 0X6666 | [15:0] | 16 | Frame end (different movement frame end position is different) |