Guide of SIMCAM SDK APIs

version 1.0.0

Contents

SimCam AI Camera

Release History

version 1.0.0	2019/03/16	First version content

1 SPI

1.1 APIs

1.1.1 void *openSpi*(void)：

[description]

Initialize SPI parameters and turn on SPI communication function.

[param]

Null.

[remark]

The SPI parameters are configured already and this function can be directly called by users.

1.1.2 void *closeSpi*(void)：

[description]

Turn off SPI communication function.

[param]

Null.

[remark]

Usually not necessary to use.

1.1.3 void *readCmd*(void (**Cmd_Handler*)())：

[description]

Read data from SPI transferred from Movidius.

SimCam AI Camera

[param]

Cmd_Handler : A function pointer to a function that parses data read from SPI.

[remark]

This function reads SPI data, verifies header and length of the data, passes correct data into a global variable *gCmd*, and parses it through *cmdHandler* function.

1.1.4 void *sendApp*(const char* *app_path*):

[description]

Write service process program from HI3516 to SPI and transfer it to Movidius.

[param]

app_path : This file is a service process program running on Movidius.

[remark]

The service process program is provided in SDK, it supports the running of detection model in this SDK. We will provide more service process project to support the operation of other models.

1.1.5 void *sendBlob*(const char* *model_path*):

[description]

Write model file from HI3516 to SPI and transfer it to Movidius.

[param]

SimCam AI Camera

model_path : This file is named as *graph* which contains the structure and weight information of a neural network.

[remark]

The *graph* can be generated by *mvNCCompile* in this SDK, see document *Toolchain_Install_Use.pdf*.

1.1.6 void *sendCfg*(uint8_t* *buf*, uint32_t *sz*):

[description]

Write configure file from HI3516 to SPI and transfer it to Movidius.

[param]

buf : Data to be written to SPI.

sz : Size of data written to SPI.

[remark]

This function is used to send parameters needed by Movidius to process the network model.

1.2 Data types

```
Typedef struct
{
    char cmd;              //store header of data come from SPI
    int bytes;             //store size of data come from SPI
    float data[1024];      //store data come from SPI
}cmd_t :
```

[description]

SimCam AI Camera

This structure is the data type of the global variable *gCmd* .The data read from SPI is verified and stored in the global variable *gCmd*.

[remark]

All SPI data will be stored in *gCmd*, but *cmdHandler* is invoked only if data validation is successful. So *data[1024]* parsed in *cmdHandler* is reliable, which represents the output of the network.

2 SERIALPORT

2.1 APIs

2.1.1 int *openSerial*(char* device):

[description]

Enable serial port function.

[remark]

Available device paths can be viewed in the */ dev* folder of the embedded board.

[return]

File describe handle.

2.1.2 void *closeSerial*(int fd):

[description]

Close serial port function.

[remark]

2.1.3 int *initSerial*(int fd, int speed, int flow, int bits, int stop, int parity):

[param]

SimCam AI Camera

fd: file descirbe handle

speed: select the Serial speed.115200,19200,9600...

flow: if use flow control

bits: data bit select

stop: stopbits select

parity: partiy select

[return]

-1: initialize error

1: initialize succeed

2.1.4 void *sendSerial*(int fd, char* buf, int len):

[param]

*fd:*file descirbe handle

*buf:*send data storage address

*len:*send data length

2.1.5 void *recvSerial*(int fd, char* buf, int len):

[param]

*fd:*file descirbe handle

*buf:*recv data storage address

len: recv data length

2.1.6 int *sendDone*(void):

[return]

SimCam AI Camera

-1: Not finish sending

1: Sending is done

2.2 Data types

Null.

3 SIMCAM_LIB

3.1 APIs

3.1.1 void *initSystem*(void):

[description]

Initialize HI3516 system.

[remark]

This function includes resetting GPIO and SPI pins, initializing shared memory, etc. Developers only have to call it once in the main function.

3.1.2 void *reset2450*(void):

[description]

Initialize and reset Movidius system.

3.1.3 void *readConfig*(CNN_Config_t* config)

[description]

Get the configuration file for model preprocessing.

[remark]

SimCam AI Camera

This function reads *config.txt* from SD-Card, which contains the parameters needed by Movidius to process the network model. The default *config.txt* applies to the detection model provided by this SDK.

3.2 Data types

```
typedef struct
{
    float x1;          // coordinates of the rect in video, in a range of 0~1
    float x2;
    float y1;
    float y2;
}Rectf_t :
```

[description]

This structure is used to store the rect's coordinates of the recognized objects.

[remark]

Null.

```
typedef struct
{
    uint8_t Y;          // color of the rect in video, in a range of 0~255
    uint8_t U;
    uint8_t V;
}YUVColor_t :
```

[description]

SimCam AI Camera

This structure is used to store the rect's color of the recognized objects.

[remark]

Null.

```
typedef struct
{
    Rectf_t rect;
    YUVColor_t color;
    int if_show;                // 0: hide rect in video;1:  show rect in video
}RectEx_t :
```

[description]

This structure describes coordinates and color of a target recognition rect in video. *if_show* determines whether show the rect in video.

[remark]

Null.

```
typedef struct
{
    int if_record;              // 1: start to record;0:  stop record
    RectEx_t rect_data;
    char path_audio[64];        //the name of audio file , should be format of .pcm
}stMemory :
```

[description]

This structure is used to transfer data in shared memory. *if_record* determines whether record a video.

SimCam AI Camera

[remark]

A global variable *viraddr* of this type has been defined, users can reassign *viraddr* to realize video recording and other operations.

```
typedef struct
{
    int model_have;
    int input_width;
    int input_height;
    int shave_num;
    int input_color;
    float mean0;
    float mean1;
    float mean2;
    float std;
    int label;
    float conf_thresh;
}CNN_Param_t ;
```

[description]

The contents of *config.txt* will be stored in this structure and sent to Movidius.

SimCam AI Camera

4 SEND DETECTION RESULT.

The SIMCAM camera can send the detection results to the user's server for each frame. Users can define if they want to send detection results to their server by changing *if_send* parameter inside *config.txt* (1-for sending, 0-not send), and also they can set server IP and port in that file.

The client:

SIMCAM camera

http method name:

/simcam/event/

http request method:

GET

Example:

"*GET http://%s:%d/simcam/event?modeltype=%d&pointA_X=%f&pointA_Y=%f&pointB_X=%f&pointB_Y=%f&timestamp=%lld\r\n\r\nCache-Control: no-cache*"

Description of parameters:

Parameter	Type	Description
modeltype	*int*	The class id of the detected object （customize）
pointA_X	*float, in range（0-1）*	The X coordinates of the upper left corner of the target recognition box
pointA_Y	*float, in range（0-1）*	The Y coordinates of the upper left corner of the target recognition box
pointB_X	*float, in range（0-1）*	The X coordinates of the lower right corner of the target recognition box
pointB_Y	*float, in range（0-1）*	The Y coordinates of the lower right corner of the target recognition box
timestamp	*int64*	Timestamp for each frame (in milliseconds）

SimCam AI Camera