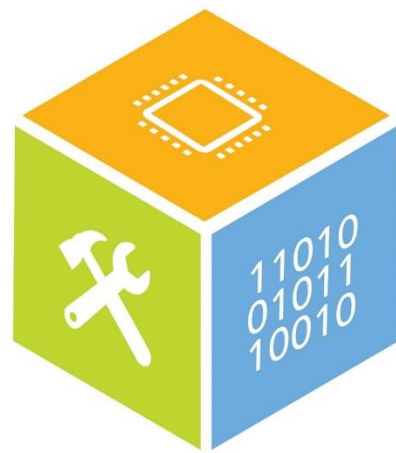


OASIS LITE runtime library

Programing Guide



SECURE CONNECTIONS
FOR A SMARTER WORLD

Table of Contents

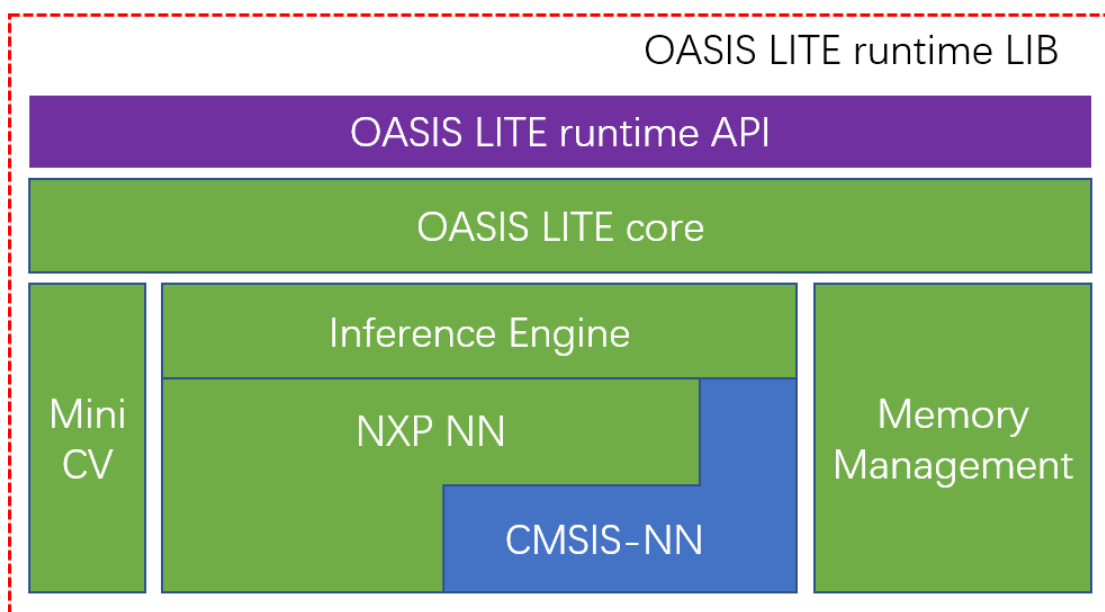
1	Introduction.....	4
2	High Level Architecture.....	4
3	Release and delivery.....	4
4	OASIS LITE runtime API.....	5
4.1	Version Information.....	5
4.2	Function enable flags.....	5
4.3	Run mode flags.....	5
4.4	Emotion IDs.....	6
4.5	Emotion recognition mode.....	6
4.6	FAR values.....	7
4.7	Image formats.....	7
4.8	Event IDs.....	7
4.9	Face Box structure.....	8
4.10	Image frame structure.....	8
4.11	Event callback parameter structure.....	8
4.12	Event callback function.....	9
4.13	Get registered faces function.....	9
4.14	Face add/update function.....	9
4.15	Log output function.....	10
4.16	Lib initialize parameters structure.....	10
5	APIs.....	11
5.2	OASISLT_init.....	11
5.3	OASISLT_run.....	11
5.4	OASISLT_uninit.....	12
5.5	OASISLT_getFaceltemSize.....	12
6	API Calling sequence.....	12
6.2	Initializing.....	13
6.3	Scheduling.....	13
6.4	Uninitializing.....	14
7	Sample code.....	14
7.2	Initialization.....	14
7.3	run.....	15
7.4	event callback function.....	15

version	Author	Description
0.1	Dongsheng (dongsheng.zhang@nxp.com)	initial version
3.0	Raymond Lei (Raymond.lei@nxp.com)	Update all chapters according new interface of OASIS LITE runtime library V3

1 Introduction

OASIS LITE runtime Library is a tiny, high efficient and customized and optimized AI library, which include a mini CV, quantized inference engine, memory management unit. It is a idea runtime library for face detection, face recognition, emotion recognition application development based on NXP NN and CMSIS NN implementation.

2 High Level Architecture



3 Release and delivery

The OASIS LITE runtime library is released in static C library format with C header files. We have different library for different platform, supported architectures are:

- ARM cortex M core for bare system or RTOS.
- ARM cortex A core on Android platform.
- X86 64bit on Linux platform.

Delivery package include:

- Several static C library named as:
 - “liboasis_lite.a” : used for any ARM cortex M bare system or RTOS, for internal debug only.
 - “liboasis_lite_ae.a”: can only be used on NXP RT serial MCU.

- “liboasis_lite_c.a”: X86 64 bit lib based on linux platform, for internal debug only.
 - “liboasis_lite_a.a”: Cortex A android platform lib.
- b) oasislite_runtime.h
This is the only header file caller need include in their source code.

4 OASIS LITE runtime API

4.1 Version Information

Name	Type	Description
VERSION_MAJOR	int	The major version number
VERSION_MINOR	int	The minor version number

4.2 Function enable flags

These flags are used in library initializing procedure, caller can combine these flags to enable different function modules.

Name	Description
OASIS_ENABLE_DET	1 bit for each function module. 0: this function module is disabled. 1: this function module is enabled.
OASIS_ENABLE_REC	
OASIS_ENABLE_EMO	
OASIS_ENABLE_LIVENESS	

4.3 Run mode flags

This flag is used in OASISLT_run function, it indicates what jobs is going to execute on the current input frame data.

Name	Type	Value	Description
OASIS_DET_ONLY	Uint8_t	0	Do face detection only
OASIS_DET_REC	Uint8_t	1	Do face detection and face recognition
OASIS_DET_EMO	Uint8_t	2	Do face detection and emotion recognition.
OASIS_DET_REC_EMO	Uint8_t	3	Do face detection, face recognition and emotion recognition.
OASIS_REG_MODE	Uint8_t(the	0/1	This bit indicate whether do face

	5 th bit)		recognition based on face recognition result. 0: no face recognition is required 1: do face recognition if this is a new face and a qualified face.
--	----------------------	--	---

4.4 Emotion IDs

Structure name: OASISLTEmoID_t

Name	Type	Value	Description
OASIS_EMO_ID_ANGER	Uint8_t	0	
OASIS_EMO_ID_DISGUST	Uint8_t	1	
OASIS_EMO_ID_FEAR	Uint8_t	2	
OASIS_EMO_ID_HAPPY	Uint8_t	3	
OASIS_EMO_ID_SAD	Uint8_t	4	
OASIS_EMO_ID_SURPRISED	Uint8_t	5	
OASIS_EMO_ID_NEUTRAL	Uint8_t	6	
OASIS_EMO_ID_NUM	Uint8_t	7	Emotion ID total number

4.5 Emotion recognition mode

Structure name: OASISLTEmoMode_t

This flag is used in OASISLT_init function, it indicates which emotion recognition mode is going to be activated in OASIS LITE runtime lib. This flag is only valid

Name	Type	Value	Description
OASIS_EMOTION_MODE_2	Uint8_t	2	2 emotions recognition model is activated. OASIS_EMO_ID_HAPPY OASIS_EMO_ID_NEUTRAL
OASIS_EMOTION_MODE_4	Uint8_t	4	4 emotions recognition model is activated. OASIS_EMO_ID_ANGER OASIS_EMO_ID_HAPPY OASIS_EMO_ID_SURPRISED OASIS_EMO_ID_NEUTRAL
OASIS_EMOTION_MODE_7	Uint8_t	7	7 emotions recognition model is activated. All 7 emotion ID are returned.
OASIS_EMOTION_MODE_INVALID	Uint8_t	0xFF	

4.6 FAR values

This value indicate what false accept rate can be allowed in face recognition procedure.
A lower FAR means a higher TRR(true rejection rate).

Name	Type	Value	Description
OASIS_FAR_1_10000	UInt8_t	0	one 10000th FAR
OASIS_FAR_1_100000	UInt8_t	1	one 100000th FAR
OASIS_FAR_1_1000000	UInt8_t	2	one 1000000th FAR

4.7 Image formats

This value indicate color format, bits for each color and their storage sequence in memory.
For all images, we assume pixel format is HWC.

Name	Type	Value	Description
OASIS_IMG_FORMAT_RGB888	UInt8_t	0	3 bytes for each pixel, 8 bits for each color(Red, Green, and Blue) and storage sequence is red, green and blue
OASIS_IMG_FORMAT_BGR888	UInt8_t	1	Same as above but the storage sequence is blue, green and red.
OASIS_IMG_FORMAT_GREY	UInt8_t	2	1 bytes for each pixel and this byte is grey value.

4.8 Event IDs

In procedure of OASISLT_run, when a job(face detection/recognition, emotion recognition and face registration) is started/completed, caller will be notified by a event callback function with a given event ID, this event list is as below.

Name	Type	Value	Description
OASISLT_EVT_DET_START	UInt8_t	0	Face detection start.
OASISLT_EVT_DET_COMPLETE	UInt8_t	1	Face detection complete.
OASISLT_EVT_REC_START	UInt8_t	2	Face recognition start.
OASISLT_EVT_REC_COMPLETE	UInt8_t	3	Face recognition complete.
OASISLT_EVT_EMO_REC_START	UInt8_t	4	Emotion recognition start.
OASISLT_EVT_EMO_REC_COMPLETE	UInt8_t	5	Emotion recognition complete.
OASISLT_EVT_REG_START	UInt8_t	6	Face registration start.
OASISLT_EVT_REG_COMPLETE	UInt8_t	7	Face registration complete.

4.9 Face Box structure

Structure name: FBox

Name	Type	Description
rect	Int[4]	The left, top, right, bottom of the face bounding box
fld	Float[10]	The x, y axes position of the 5 face landmark points

4.10 Image frame structure

Structure name: ImageFrame_t

Name	Type	Description
height	short	Image height in pixel
width	short	Image width in pixel
Reserved0	int	Reserved for internal use only, must be set to 0
data	Unsigned char*	Pointer of image data. Image data format is HWC format.

4.11 Event callback parameter structure

Structure name: OASISLTCbPara_t

Name	Type	Description
pfaceBox	FBox*	Face box detected. NULL indicates that no face is detected.
emoID	OASISLTEmoID_t	The result of emotion recognition. See emotion IDs for detail. Only valid for emotion recognition complete event callback.
faceID	UInt16_t	For Face recognition complete event callback, this ID indicates recognized face ID of pfaceBox. For Registration complete event callback, this ID is the new generated face ID for pfaceBox. For other event, faceID is invalid.

4.12 Event callback function

Prototype:

```
typedef void (*OASISLTEvtCb)(ImageFrame_t* frame,
                             OASISLTEvt_t evt,
                             OASISLTCbPara_t* para,
                             void* user_data);
```

Description :

In the process of OASISLT_run, this callback function triggers when each event in 4.8 happens.

Parameter Name	Input/output	Description
frame	Input	Current image frame on which all jobs is handling.
evt	Input	Event ID which triggers this callback.
Para	Input	Other parameters which can be used in this callback, see 4.11 for detail.
User_data	Input	User data pointer input by OASISLT_run.

4.13 Get registered faces function

Prototype:

```
typedef int (*GetRegisteredFaces)(int start_idx,void* faces,unsigned int* face_num);
```

Description :

This function should be implemented by the caller. By this interface, OASIS LITE runtime library can get face information records from a given index(start from 0 to the number of face info records minus 1).

Return 0 if success, other value indicate a failure.

Parameter Name	Input/output	Description
Start_idx	input	Start index of face information records.
faces	output	Buffer used to save output face information records. Size of this buffer is: face_num*OASISLT_getFaceltemSize()
Face_num	Input/output	For input, it indicates how many face information records is required. For output, it indicates how many face information records are returned in fact.

4.14 Face add/update function

Prototype:

```
typedef int (*FaceOperation)(uint16_t face_id,void* face_data);
```

Description :

This function should be implemented by the caller. By this interface, OASIS LITE runtime library can add/update face information record by face ID.

Return 0 if success, other value indicate a failure.

Parameter Name	Input/output	Description
Face_id	input	Which face information record to be operated.
Face_data	input	Pointer to a face information record. It's size can be get by OASISLT_getFaceltemSize()

4.15 Log output function

Prototype:

```
typedef int (*LogOutputFun)(const char* format);
```

Description :

This function should be implemented by the caller. By this interface, OASIS LITE runtime library can output log information.

Parameter Name	Input/output	Description
format	input	Output ASCII string

4.16 Lib initialize parameters structure

Description :

Initializing parameter is the only parameter OASISLT_init need.

Parameter Name	Type	Description
height	int	Maximum height of input images OASIS library can handle.
width	int	Maximum width of input images OASIS library can handle.
Img_format	Input/output	For input, it indicates how many face information records is required. For output, it indicates how many face information records are returned in fact.
Min_face	Input	Minimum face size can be detected. This size should less or equal to minimum face size for each OASISLT_run operation. Range 40-150.
Mem_pool	Input	Memory pool pointer can be used by OASIS library. If this parameter is NULL, OASISLT_init would return memory pool size required.

Size	Input	Memory pool size.
auth	output	Current HW platform is authenticated or not. 1: authenticated. 0: not authenticated.
cbs	Input	Callback function set, which include: Event callback function Get registered faces function Add/update face function Log output function
Enable_flags	Input	What function modules are enabled in OASIS Library? It can be a combine of flags in 4.2
Emo_mode	Input	What emotion recognition mode is activated in OASIS library? Refer to 4.5 This parameter is only valid when emotion recognition module is activated in Enable_flags(OASIS_ENABLE_EMO).
far	Input	False accept rate caller required.

5 APIs

5.2 OASISLT_init

Prototype	int OASISLT_init(OASISLTInitPara_t* para);
Description	Initialize OASIS LITE runtime library. If mem_pool is NULL, return memory pool size.
Input Param	Para, refer to 4.16 for detail information.
Return Value	0: Success >0 : memory pool size. <0 : Error.

5.3 OASISLT_run

Prototype	int OASISLT_run(ImageFrame_t *frame,uint8_t flag,int minFace,void* userData);
Description	Do Jobs(face detection/recognition, emotion recognition and face registration, up to flag parameter) on given image frame.
Input Param	Frame: image frame on which do jobs Flag: what jobs is going to take? It can be one of

	OASIS_DET_ONLY OASIS_DET_REC, OASIS_DET_EMO, OASIS_DET_REC_EMO Ored with/without OASIS_REG_MODE minFace: minimum face size can be detected on the current image frame, it should not less than min_face in library initializing parameters. userData: user data transfer to callback functions.
Return value	0: success Others: failure.

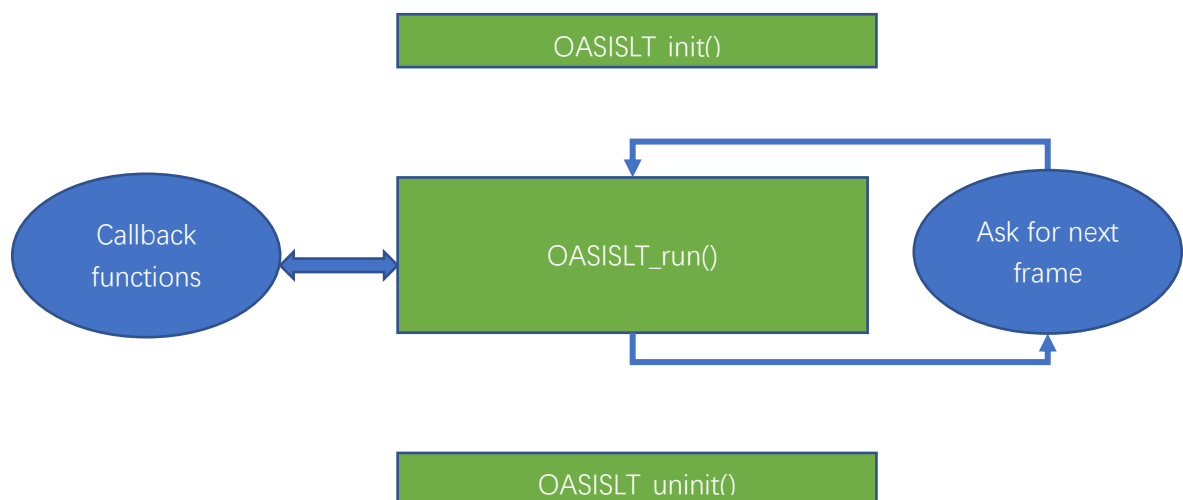
5.4 OASISLT_uninit

Function	int OASISLT_uninit();
Description	Uninitialized OASIS lite runtime library.
Input Param	None
Return Value	0: success Other value: failure

5.5 OASISLT_getFaceltemSize

Function	uint32_t OASISLT_getFaceltemSize();
Description	When adding/updating face information record or get face information records , caller need know face information item size by this function.
Input Param	None
Return Value	Face information record size in bytes.

6 API Calling sequence



6.2 Initializing

OASISLT_init is the first API should be called. In Initializing procedure, caller need provide: Input image frame maximum height/width in pixels, generally these number is up to the resolution of camera you are using. Following image size in calling of OASISLT_run can be less or equal to these values but must not exceed these values.

Minimum face size can be detected in pixel, following minimum face size in calling of OASISLT_run can be larger or equal this value.

We recommend OASISLT_init be called twice, in the first call of OASISLT_init, all other parameters should be filled except setting mem_pool as NULL, the return value will be memory size needed. Caller can allocate memory according the return value and call OASISLT_init again with a return of 0.

6.3 Scheduling

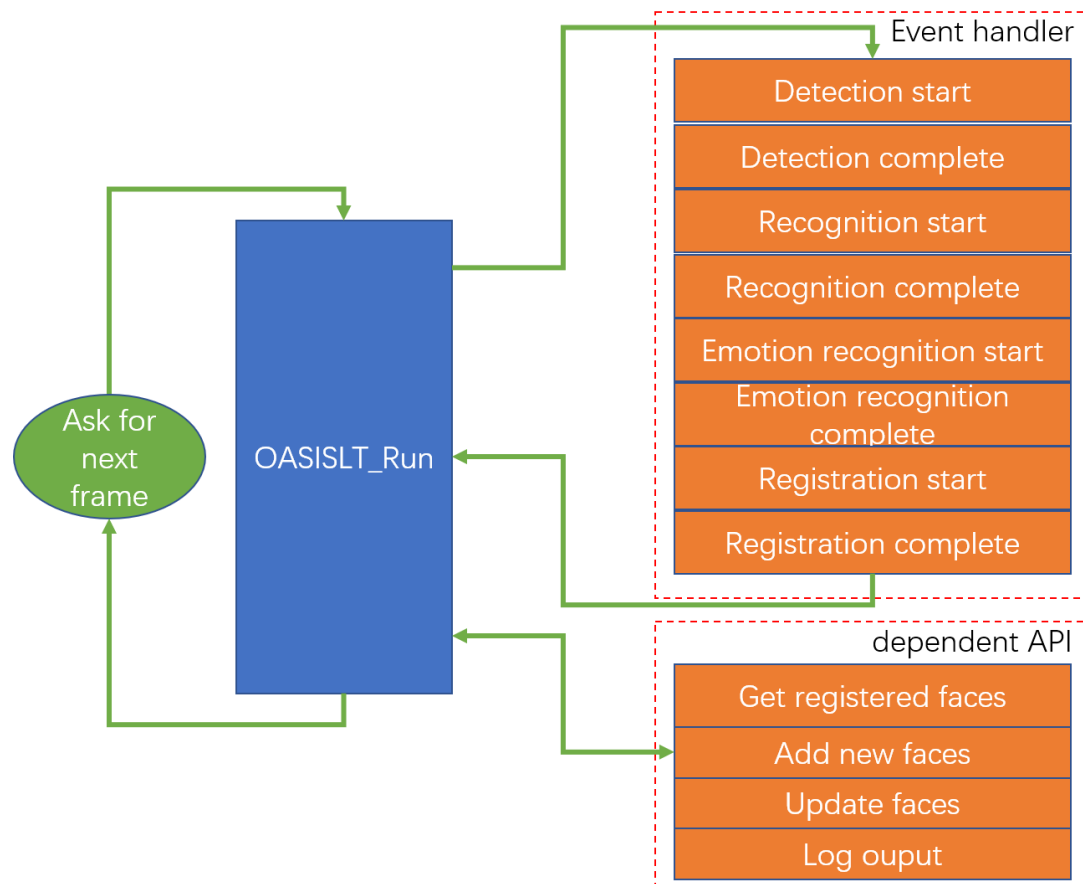
In procedure of OASISLT_run, it always communicate with the caller by callback functions provided in OASISLT_init parameters.

Detection start/complete event are always triggered.

Emotion/face recognition start/complete event: only be triggered when a face is detected and emotion/face recognition modules are enabled.

Registration start/complete event: only be triggered when:

- A face is detected.
- The detected face is not in face records database.
- The detected face is a qualified face.
- OASIS_REG_MODE is activated.



In the event callback, caller can hook different handling functions for different event.
If caller want to some detail info in debug mode, set "print" function in callbacks of `OASISLT_init`.

6.4 Uninitializing

Just call `OASISLT_uninit()`

7 Sample code

7.2 Initialization

```
OASISLTInitPara_t init_para = {  
    .height = 240,  
    .width = 320,
```

```

        .img_format = OASIS_IMG_FORMAT_BGR888,
        .min_face = 100,
        .mem_pool = NULL,
        .size = 0,
        .cbs = {
            EvtHandler,
            GetRegisteredFacesHandler,
            AddNewFaceHandler,//add face
            UpdateFaceHandler,
            OasisV3_Printf,
        },
        .enable_flags = OASIS_ENABLE_DET|OASIS_ENABLE_REC,
        .emo_mode = OASIS_EMOTION_MODE_2,
        .far = OASIS_FAR_1_1000000,
    };

    int ret = 0;
    ret = OASISLT_init(&init_para);
    if (ret > 0)
    {
        init_para.mem_pool = (char*)malloc(ret);
        init_para.size = ret;
        ret = OASISLT_init(&init_para);
        assert(ret == 0);
    }

```

7.3 run

```

while (1)
{
    {Wait for a new frame arriving}
    int ret = OASISLT_run(&frame,OASIS_DET_REC_EMO|OASIS_REG_MODE,100,NULL);
    assert(ret == 0);
}

```

7.4 event callback function

```

switch(evt)
{

    case OASISLT_EVT_DET_START:
    {

```

```
        //do something you want, for example: detection time statistics
    }
    break;

case OASISLT_EVT_DET_COMPLETE:
{
    if (para->pfaceBox == NULL)
    {
        //no face is detected

    }else
    {
        //face detected, face rectangle information is in para->pfaceBox->rect

    }
}
    break;

case OASISLT_EVT_REC_START:
{
    //do something you want, for example: face recognition time statistics

}
    break;
case OASISLT_EVT_REC_COMPLETE:
{
    unsigned id = para->faceID;
    if (id != INVALID_FACE_ID)
    {
        LOGD("[OASISV3]:face id:%d\r\n",id);
        //face is recognized
    }else
    {
        //face is not recognized

    }
}
    break;
case OASISLT_EVT_EMO_REC_START:
{
    //do something you want, for example: emotion recognition time statistics

}
    break;
case OASISLT_EVT_EMO_REC_COMPLETE:
```



```
{
    //emotion recognition complete, what's the emotion→ para->emolD
}
    break;
case OASISLT_EVT_REG_START:
{
    //can hook some function
}
    break;
case OASISLT_EVT_REG_COMPLETE:
{
    unsigned id = para->faceID;
    if (id != INVALID_FACE_ID)
    {
        LOGD("[OASISV3]:new face id:%d\r\n",id);
        //a new face has been added into the database, it's ID is para->faceID
    }else
    {
        //face quality is not so good, registration is canceled.
    }
}
    break;
default:
    assert(0);
}
```