



FacePass_L9K Module Protocol Design

This document is the access document for the Face Recognition Technology L9K module protocol, targeting end-users. It now provides a detailed and comprehensive description of the protocol usage instructions and some usage suggestions. For actual integration, please refer to the complete usage of the supporting Demo project.

FacePass_L9K Module Protocol Design

- Protocol Header

- Message Structure

- Protocol Content

 - Module Initialization

 - Source Identifier

 - URI Content

 - Response Content

 - De-initialization

 - Source Identifier

 - URI Content

 - Response Content

 - Start Recognition

 - Source Identifier

 - URI Content

 - Response Content

 - Pause Recognition

 - Source Identifier

 - URI Content

 - Response Content

 - Set Liveness Configuration

 - Source Identifier

 - URI Content

 - Request

 - Response Content

 - Set Recognition Configuration

 - Source Identifier

 - URI Content

 - Request

 - Response Content

 - Create Palm Group

 - Source Identifier

 - URI Content

 - Response Content

 - Create Face Group

 - Source Identifier

- URI Content
- Response Content
- Single User Enrollment
 - Source Identifier
 - URI Content
 - Request
 - Response Content
- Batch Feature Enrollment
 - Source Identifier
 - URI Content
 - Request
 - Response Content
- Delete User
 - Source Identifier
 - URI Content
 - Request
 - Response Content
- Single Frame Enrollment
 - Source Identifier
 - URI Content
 - Request
 - Response Content
- JPG Image Enrollment
 - Source Identifier
 - URI Content
 - Request
 - Response Content
- Dual-channel Fusion Feature Enrollment (Temporarily for Palm Use)
 - Source Identifier
 - URI Content
 - Request
 - Response Content
- Set UVC Resolution
 - Source Identifier
 - URI Content
 - Request
 - Response Content
- Set Coding Type
 - Source Identifier
 - URI Content
 - Request
 - Response Content
- Query Software Version Number
 - Source Identifier
 - URI Content
 - Response Content
- Query User Count
 - Source Identifier
 - URI Content
 - Request
 - Response Content
- Status Report
 - Source Identifier

URI Content
Request
Energy-saving Mode
Source Identifier
URI Content
Request
Response Content
Set System Time
Source Identifier
URI Content
Request
Response Content
Database Reset
Source Identifier
URI Content
Request
Response Content
Enable QR Code Detection
Source Identifier
URI Content
Request
Response Content
OTA Upgrade Start
Source Identifier
URI Content
OTA Upgrade End
Source Identifier
URI Content
Response Content
OTA Upgrade Content
Source Identifier
URI Content
Request

Protocol Header

```
typedef struct
{
    unsigned int sync_flag;           // Synchronization code, fixed as EB CF 79
    unsigned int msg_crc32;           // CRC32 check code for the entire message (excluding the
first 8 bytes)
    unsigned short head_crc16;        // Check code for this header information, CRC16
(excluding the first 12 bytes)
    unsigned char head_len;           // Header length
    unsigned char uri_len;            // Length of the interface URI information
    unsigned int msg_len;             // Total message length
    unsigned short serial;            // Packet serial number
    unsigned char type;               // Message type: 0: request, 1: response, 2: recog result

    unsigned char need_resp;          // whether a response is needed
} FPMSG_S;
```

Message Structure

Sync flag (4byte)	CRC32 (4byte)	head crc16 (2byte)	Head len (1byte)	uri len (1byte)	msg len (4byte)	serial (2byte)	msg type (2byte)	need resp (2byte)	uri	msg
0x79CFEB	msg part crc32	Reserved, set to 0	Header length	Source identifier length	Message length	Number	Message type 0: request packet 1: response packet 2: algorithm result push	Whether a response is needed	Source identifier	Specific message content JSON format

Special Notes:

- The header structure of request commands, response packets, and algorithm result push data is consistent.
- The module does not store features. Users need to call the enrollment interface or send face images for temporary local storage, and the device sends them to the module through the enrollment protocol during startup.
- Both feature and image content transmission use base64 encoding.
- The interaction between the module and the upper-level device is in JSON data format.

Protocol Content

Module Initialization

Source Identifier

FACE_PASS_CMD_INIT_MODULE

URI Content

/api/module/init

Response Content

```
{ "response": 0 }
```

- 0 indicates success
- Non-0 indicates failure

De-initialization

Source Identifier

FACE_PASS_CMD_RELEASE_MODULE

URI Content

/api/module/release

Response Content

```
{ "response": 0 }
```

- 0 indicates success
- Non-0 indicates failure

Start Recognition

Source Identifier

FACE_PASS_CMD_START_RECOG

URI Content

/api/module/start/recog

Response Content

```
{ "response": 0 }
```

- 0 indicates success
- Non-0 indicates failure

Pause Recognition

Source Identifier

FACE_PASS_CMD_PAUSE

URI Content

/api/module/pause

Response Content

```
{ "response": 0 }
```

- 0 indicates success
- Non-0 indicates failure

Set Liveness Configuration

Source Identifier

FACE_PASS_CMD_SET_LIVE_CONFIG

URI Content

/api/recog/set/config/live

Request

```
{  
  "en_live": 0,  
  "live_th": 80,  
  "retry_cnts": 3  
}
```

- live_th: Liveness threshold
- en_live: Liveness switch
- retry_cnts: Number of retries for non-liveness determination

Response Content

```
{ "response": 0 }
```

- 0 indicates success
- Non-0 indicates failure

Set Recognition Configuration

Source Identifier

FACE_PASS_CMD_SET_RECOG_CONFIG

URI Content

/api/recog/set/config/recog

Request

```
{
  "face_th": 80,
  "palm_th": 43,
  "retry_cnts": 3,
  "push_interval": 2
}
```

- face_th: Face recognition threshold
- palm_th: Palm recognition threshold
- retry_cnts: Number of retries for suspected persons
- push_interval: Image push interval

Response Content

```
{ "response": 0 }
```

- 0 indicates success
- Non-0 indicates failure

Create Palm Group

Source Identifier

FACE_PASS_CMD_BOOK_PALM_GROUP

URI Content

/api/book/create/group/palm

Response Content

```
{
  "response": 0,
  "group_id": 1
}
```

- 0 indicates success
- Non-0 indicates failure

- group_id: Palm group ID

Create Face Group

Source Identifier

FACE_PASS_CMD_BOOK_FACE_GROUP

URI Content

/api/book/create/group/face

Response Content

```
{
  "response": 0,
  "group_id": 0
}
```

- 0 indicates success
- Non-0 indicates failure
- group_id: Palm group ID

Single User Enrollment

Source Identifier

FACE_PASS_CMD_BOOK_ADD

URI Content

/api/book/add/user

Request

```
{
  "group_id": 0,
  "face_id": 1,
  "ft_len": 376,
  "ft": "TUZYMQABCQAAAAAAAAAAPmdrgYAAAAAIiS6/seuxlT5kmtmUnr2Jet9/fRjivnD=
B26DiamjJTcpCbxo0eZQejwCpI3UBgb40QKqt3hLX8f7e80XSyFI0p+yh2o07n6mUe1J/VaDwRt=
Ko6yibkoXGIK2fJ8mzzvdd8HS1UWFhKCK/HDYNG5zswrWEKEBD06t/E19A1j1J0B3CDKAL/YsZr=
Jo68tfAfLgMSSmCH+x8tB14PsRcq1e8G4JydxBVqEDG44Lssako66GhZudZKTUwfH081t/v4k5t=
cZkwtk7oYwf8FtqQ56tP0VD/YKyWRMOY3GdvniDbMqzQPS7GQkwwHdM89B70nzT2GaCu78oF4n=
q+Uz3xBSA=3D=3D"
}
```

- group_id: Target group ID, obtained from the module return in FACE_PASS_CMD_BOOK_PALM_GROUP
- face_id: Unique user identifier
- ft_len: Feature length
- ft: Actual feature

Response Content

```
{ "response": 0 }
```

- 0 indicates success
- Non-0 indicates failure

Batch Feature Enrollment

Source Identifier

FACE_PASS_CMD_BOOK_BATCH_ADD

URI Content

/api/book/add/batch

Request

```
{
  "user_cnt": 1,
  "ft_datas": [
    {
      "group_id": 0,
      "face_id": 1,
      "ft_len": 376,
      "ft": "TUZYMQABCQAAAAA2GaCu7.....8oF4nq+Uz3xBSA=3D=3D"
    }
  ]
}
```

- user_cnt: Number of batch modeling features
- ft_datas: Batch modeling feature content
- group_id: Target group ID, obtained from the module return in FACE_PASS_CMD_BOOK_PALM_GROUP
- face_id: Unique user identifier
- ft_len: Feature length
- ft: Actual feature

Response Content

```
{ "response": 0 }
```

- 0 indicates success
- Non-0 indicates failure

Delete User

Source Identifier

/api/book/del/user

URI Content

/api/recog/set/config/recog

Request

```
{  
  "face_id": 1,  
  "group_id": 1,  
  "del_flag": 0  
}
```

- del_flag: 0, delete the face_id user feature in the specified group group_id
- del_flag: 1, delete all features in the specified group group_id
- del_flag: 2, delete all user features in all groups

Response Content

```
{ "response": 0 }
```

- 0 indicates success
- Non-0 indicates failure

Single Frame Enrollment

Source Identifier

FACE_PASS_CMD_ENROLL_TYPE_FRM

URI Content

/api/enroll/frm

Request

```
{  
  "face_id": 2,  
  "obj_type": 1,  
  "time": 15000  
}
```

- face_id: Unique user identifier
- obj_type: Target type, 0 for face, 1 for palm
- The enrollment result is returned through single-frame algorithm processing result

- time: Enrollment timeout time

Response Content

```
{ "response": 0 }
```

- 0 indicates success
- Non-0 indicates failure

JPG Image Enrollment

Source Identifier

FACE_PASS_CMD_ENROLL_TYPE_JPEG

URI Content

/api/enroll/pic/jpeg

Request

```
{
  "face_id": 5000,
  "pic_len": 541700,
  "pic": "/9j/4AAQSkZJRgAQklNBCUAAAAAABDUHYZzZjwCyBomACZjs+EJ+/+ICKElDQ19Q=
Uk9GSUXFA AEBAACGGFWCGWEAAAAbw50clJHQiBYWVogB+YAAQABAAAAAAAYWNZCEFQUEWAAA=
AAQVBQTAAAAAAAAAAAAAAAAAAAAA AAPbWAAEAAAAA0y1hCHBs7P2jjjiFR8NttL....."
}
```

- face_id: Unique user identifier
- pic_len: Length of image content after encoding
- pic: Image content, base64 encoded

Response Content

```
{
  "response": 0,
  "face_id": 5000,
  "err_info": 0,
  "otype": 0,
  "ft": "TUZYMQABCQAAAAAAAAAAAAA.....PmdrgYAAAAAIiS6\seux1T5kmtmUnr2J+Uz3=
xBSA=3D=3D",
  "ft_len": 376
}
```

- face_id: Unique user identifier
- err_info: Enrollment error information, refer to the specific error information definition in the initial document, <FacePass_L9K_SDK_API>
- otype: Target type

- ft: Feature
- ft_len: Feature length

To determine whether the enrollment is successful, refer to `err_info` and `response` values. `err_info` returns 0 for success, and other values for failure.

Dual-channel Fusion Feature Enrollment (Temporarily for Palm Use)

Source Identifier

FPMSG_CMD_ENROLL_TYPE_PIC_RGB_IR

URI Content

/api/enroll/pic/rgb_ir

Request

```
{
  "face_id": 5000,
  "pic_len": 541700,
  "pic": "/9j/4AAQSkZJRgAQk1NBCUAAAAAL....",
  "ir_pic_len": 11111,
  "ir_pic": "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA..."
}
```

- face_id: Unique user identifier
- pic_len: Length of image content after encoding
- pic: Image content, base64 encoded
- ir_pic_len: Length of infrared image content after encoding
- ir_pic: Infrared image content, base64 encoded

Response Content

```
{
  "response": 0,
  "face_id": 5000,
  "err_info": 0,
  "otype": 0,
  "ft": "TUZYM=.....3D=3D",
  "ft_len": 376
}
```

- face_id: Unique user identifier
- err_info: Enrollment error information, refer to the specific error information definition in the initial document, <FacePass_L9K_SDK_API>
- otype: Target type
- ft: Feature

- ft_len: Feature length

To determine whether the enrollment is successful, refer to `err_info` and `response` values. `err_info` returns 0 for success, and other values for failure.

Set UVC Resolution

Source Identifier

FACE_PASS_CMD_SET_VIDEO_UVC_RES

URI Content

/api/video/uvc/set/resolution

Request

```
{"resolution":1}
```

```
typedef enum
{
    FACE_PASS_UVC_RES_1080P = 0,
    FACE_PASS_UVC_RES_720P ,
    FACE_PASS_UVC_RES_480P,
    FACE_PASS_UVC_RES_320P
}FACE_PASS_UVC_RES_E
```

Response Content

```
{ "response": 0,}
```

- 0 indicates success
- Non-0 indicates failure

Set Coding Type

Source Identifier

FACE_PASS_CMD_SET_VIDEO_UVC_CODE_TYPE

URI Content

/api/video/uvc/set/code_typecode_type

Request

```
{"code_type":1}
```

code_type:

FACE_PASS_UVC_PAYLOAD_TYPE_H264 = 0,
FACE_PASS_UVC_PAYLOAD_TYPE_JPEG,
FACE_PASS_UVC_PAYLOAD_TYPE_MJPEG,
FACE_PASS_UVC_PAYLOAD_TYPE_H265,
FACE_PASS_UVC_PAYLOAD_TYPE_BUTT

Response Content

```
{ "response": 0, }
```

- 0 indicates success
- Non-0 indicates failure

Query Software Version Number

Source Identifier

FACE_PASS_CMD_GET_FT_VERSION

URI Content

/api/query/soft/version

Response Content

```
{  
  "response": 0,  
  "soft_ver": "0240914",  
  "sdk_ver": "0240914",  
  "sn": "0000000000000000",  
  "feature_id": 1111  
}
```

- soft_ver: Firmware version
- sdk_ver: Algorithm SDK version
- sn: Device serial number
- feature_id: Feature version. The feature version will be updated when the model is identified. Previously enrolled user features need to be re-extracted

Query User Count

Source Identifier

FACE_PASS_CMD_GET_USER_CNT

URI Content

/api/query/user/cnt

Request

```
{"obj_type":1}
```

obj_type: 0, number of faces; 1, number of palms; 2, total number of all users (faces + palms)

Response Content

```
{
  "response": 1,
  "user_cnt": 1,
  "palm_cnt": 1,
  "face_cnt": 1
}
```

- user_cnt: Total number of users
- palm_cnt: Number of palms
- face_cnt: Number of faces

Status Report

The data actively reported by the module to the upper layer, including single-frame algorithm processing results/upgrade status

Source Identifier

FACE_PASS_CMD_SEND_RECOG_RESULT

URI Content

/api/push/recog_result

Request

```
{
  "err_info": 0,
  "fid": 0,
  "ppl_code": 56,
  "obj_type": 0,
  "track_id": 0,
  "ft": "",
  "ft_len": 0,
  "img": "",
  "img_size": 0,
  "ir_img": "",
  "ir_img_size": 0,
  "attr": {
    "yaw": 0,
    "roll": 0,

```

```

        "pitch": 0,
        "blur": 0,
        "live_st": 0,
        "liveness": 0
    },
    "det_info": {
        "det_st": 0,
        "fail_index": 0,
        "rect_x": 0,
        "rect_y": 0,
        "rect_w": 0,
        "rect_h": 0
    },
    "reg_info": {
        "face_id": 0,
        "id_existed": 0,
        "time_out": 0
    },
    "iden_info": {
        "iden_ges": 0,
        "iden_score": 0,
        "stranger": 0,
        "top1_id": 0
    },
    "ota_info": {
        "process": 100,
        "err_info": 1
    },
    "qrcode_info": {
        "str_len": 100,
        "qrcode": "hello world"
    }
}

```

- process: Upgrade progress
- err_info: Error information during the upgrade process, refer to the specific error information definition in the initial document, <FacePass_L9K_SDK_API>
- fid: Current frame ID
- track_id: Target tracking ID
- ppl_code: Algorithm return value
- obj_type: Pushed image target type
- ft: Feature data after base64 encoding
- ft_len: Feature length
- img: Captured image, database image data after encoding
- img_size: Image size after encoding
- ir_img: IR channel database image data
- ir_img_size: IR channel database image data size

Energy-saving Mode

Source Identifier

FACE_PASS_CMD_SET_ECO_MODE

URI Content

/api/recog/set/push/eco_mode

Request

```
{  
  "eco_st": 1  
}
```

Energy-saving mode: eco_st: 0, disabled; 1, enabled (under normal light, if no target is detected within 2 minutes, the infrared light will be automatically turned off to reduce the algorithm frame rate)

Response Content

```
{ "response": 0 }
```

- 0 indicates success
- Non-0 indicates failure

Set System Time

Source Identifier

FACE_PASS_CMD_SET_SYSTEM_TIME

URI Content

/api/set/systime

Request

```
{  
  "year": 1900,  
  "month": 1,  
  "day": 1,  
  "hour": 1,  
  "min": 1,  
  "sec": 1  
}
```

- year: Year
- month: Month
- day: Day

- hour: Hour
- min: Minute
- sec: Second

Response Content

```
{ "response": 0, }
```

- 0 indicates success
- Non-0 indicates failure

Database Reset

Source Identifier

FACE_PASS_CMD_GET_FT_VERSION

URI Content

/api/set/face_repeat

Request

```
{  
  "repeat_st": 1  
}
```

- repeat_st 1: Enable database reset
- repeat_st 0: Disable database reset

Response Content

```
{ "response": 0, }
```

- 0 indicates success
- Non-0 indicates failure

Enable QR Code Detection

Source Identifier

FACE_PASS_CMD_SET_QRCODE_DET

URI Content

/api/set/qrcode

Request

```
{  
  "qrcode_st": 1  
}
```

- qrcode_st 1: Enable QR code detection
- qrcode_st 0: Disable QR code detection. If no QR code is detected within a certain period, the QR code detection process will be exited

Response Content

```
{ "response": 0, }
```

QR code data is returned through the data report interface, with qrcode field

OTA Upgrade Start

Source Identifier

FACE_PASS_CMD_OTA_START

URI Content

/api/ota/start

OTA Upgrade End

Source Identifier

FACE_PASS_CMD_OTA_STOP

URI Content

/api/ota/stop

Response Content

Upgrade status information is reported in `FACE_PASS_CMD_SEND_RECOG_RESULT`

OTA Upgrade Content

Source Identifier

FACE_PASS_CMD_OTA_DATA

URI Content

/api/ota/data

Request

```
{  
  "packet_size": 1,  
  "ota_data": ""  
}
```

- packet_size: Number of packets after upgrade packet segmentation (it is recommended to split by 128K according to the specification)
- ota_data: Content of a single packet data