SenseID 智能人脸识别模组

SEV2 通信接口文档

版本：V1.5-210910

声明

下列文件包含深圳市商汤科技有限公司（以下简称为商汤）的私有信息，在没有获得商汤正式许可的情况下，第三方不得使用或随意泄露，任何在没有授权、特殊条件、限制或告知的情况下对此信息的复制和擅自修改都是侵权行为。一经发现，经追究其法律责任

在任何时间，无需告知任何方的情况下，商汤有权对本公司产品和服务进行更改、添加、删除、改进以及其它任何变更。

产品文档修订记录

版本	说明	修订作者
V1.0-210405	基于 SEV1S，创建 SEV2 通信接口文档	王再阔
V1.1-210415	修正 START_OTA 指令描述	王国君
V1.2-210419	修正文中细节	张德建
V1.3-210706	修正 ENROLL_ITG 指令参数 enroll_type 添加 MID_FACERESET 指令 修正文中细节	张德建
V1.4-210813	改版	张德建
V1.5-210910	修正错误	张德建

目录

1 通信协议说明 1
1.1 串口配置说明 1
1.2 通信消息格式 1
1.3 通信消息详细说明 2
1.3.1 消息头文件定义 2
1.3.2 消息列表 2
1.4 模组消息的发送和接收 4
1.4.1 消息接收(H>>M) 4
1.4.2 消息发送(M>>H) 4
1.5 上下电流程 8
1.6 主控接收消息流程 9
1.7 一般消息处理流程 10
1.8 录入流程 10
1.9 解锁流程 11
1.10 加密通信流程 11
1.11 OTA 升级流程 12
1.12 补充说明 13
1.12.1 设置加密规则 MID_SET_RELEASE_ENC_KEY 13
1.12.2 进入加密模式 MID_INIT_ENCRYPTION 14
1.12.3 待机指令 MID_RESET 16
1.12.4 获取模组状态 MID_GETSTATUS 16
1.12.5 识别 MID_VERIFY 17
1.12.6 录入指令 20

1.12.7 清除人脸录入状态 MID_FACERESET 27
1.12.8 删除单个用户 MID_DELUSER 27
1.12.9 删除所有用户 MID_DELALL 28
1.12.10 获取指定 ID 的用户信息 MID_GETUSERINFO 29
1.12.11 获取已注册用户的 ID MID_GET_ALL_USERID 30
1.12.12 演示模式 MID_DEMOMODE 31
1.12.13 获取模组固件版本 MID_GET_VERSION 32
1.12.14 获取算法库版本 MID_GETLIBRARY_VERSION 32
1.12.15 设置安全等级 MID_SET_THRESHOLD_LEVEL 33
1.12.16 图像抓拍 34
1.12.17 获取断电保存的日志 38
1.12.18 获取售后 debug 日志 40

1 通信协议说明

人脸模组和主控之间采用串口方式通信，人脸模组始终处于从机地位，主控需通过发送不同的指令让模组完成相应的功能。主控发送给模组的指令、模组对主控的应答需按照规定的格式来进行。本章将阐述串口的配置及通信的协议。

1.1 串口配置说明

表 1-1

配置项	说明
波特率	支持大多波特率，默认 115200
硬件/软件流控制	不使用
数据位	8
停止位	1
奇偶校验位	不使用
信号电平	3.3V

1.2 通信消息格式

主控与模组通信的基本消息格式如表 1-2 所示。

表 1-2

SyncWord	MsgID	Size	Data	ParityCheck
2 bytes	1 byte	2 bytes	N bytes	1 byte

表 1-3 是对各个字段的详细说明。

表 1-3

字段	长度	说明
SyncWord	2bytes	固定的消息开头同步字 0xEF 0xAA
MsgID	1byte	消息 ID (例如 RESET)
Size	2bytes	Data size，单位 byte

Data	N bytes	消息对应的 data，如 command 消息对应的参数。 65535>=N>=0，N=0 表示此消息无参数。
Parity Check	1 byte	协议的奇偶校验码，计算方式为整条协议除去 Sync Word 部分后，其余字节按位做 XOR 运算。

1.3 通信消息详细说明

1.3.1 消息头文件定义

消息头文件定义见附件 message_v2.h 文件。

1.3.2 消息列表

表 1-4 中列出了模组(M)和主控(H)的所有通信消息。**每条消息处理都对应有超时时间定义，Timeout 指的是主控等待模组处理的最长时间，即模组反馈处理完成的时间。**

当指令处理超时，主控可采用如下几种处理方式:

1. 发送 MID_GETSTATUS 指令

此消息能快速获取模组的工作状态，若死机则无返回，主控可直接断电。

2. 发送 RESET 指令

停止所有当前的处理，模组进入 standby 的状态，等待主控新的指令。

3. 认为模组死机

主控可采用断电重启的处理方式。

表 1-4

MsgID	Code	DIR	Timout(s)	说明
MID_REPLY	0x00	M»H	N/A	Reply 是模组对主控发送出的命令的应答，对于主控的每条命令，模组最终都会进行 reply
MID_NOTE	0x01	M»H	N/A	Note 是模组主动发给主控的信息，根据 note id 判断消息类型和适配的 data 结构

MID_IMAGE	0x02	M»H	N/A	模组给主控传送图片
MID_RESET	0x10	H»M	2	停止所有当前在处理的消息，模组进入待机状态
MID_GETSTATUS	0x11	H»M	2	立即返回模组当前状态
MID_VERIFY	0x12	H»M	自定义	鉴权解锁
MID_ENROLL	0x13	H»M	自定义	交互式录入
MID_SNAPIMAGE	0x16	H»M	10	抓拍图片并存储到本地
MID_GETSAVEDIMAGE	0x17	H»M	2	获取待上传图片大小
MID_UPLOADIMAGE	0x18	H»M	2	将本地存储的图片上传到主控
MID_ENROLL_SINGLE	0x1D	H»M	自定义	单帧录入
MID_DELUSER	0x20	H»M	5	删除一个注册用户
MID_DELALL	0x21	H»M	5	删除所有注册用户
MID_GETUSERINFO	0x22	H»M	2	获得某个注册用户的信息
MID_FACERESET	0x23	H»M	2	重置算法状态，如果正在进行交互式录入，会清空已录入的方向
MID_GET_ALL_USERID	0x24	H»M	2	获取所有已注册用户的数量和 ID
MID_ENROLL_ITG	0x26	H»M	自定义	集成支持并扩展所有录入方式
MID_GET_VERSION	0x30	H»M	2	获得软件版本信息
MID_START_OTA	0x40	H»M	2	进入 OTA 升级模式
MID_STOP_OTA	0x41	H»M	2	退出 OTA 模式模组重启
MID_GET_OTA_STATUS	0x42	H»M	2	获取 OTA 状态以及传输升级包的起始包序号
MID_OTA_HEADER	0x43	H»M	2	发送升级包的大小，总包数，分包的大小，升级包的 md5 值
MID_OTA_PACKET	0x44	H»M	2	发送升级包：包序号、包大小、包数据
MID_INIT_ENCRYPTION	0x50	H»M	2	设置加密随机数
MID_CONFIG_BAUDRATE	0x51	H»M	2	OTA 模式下设定通信口波特率
MID_SET_RELEASE_ENC_KEY	0x52	H»M	2	设定量产加密秘钥序列，掉电会保存

MID_SET_DEBUG_ENC_KEY	0x53	H»M	2	设定调试加密秘钥序列，掉电丢失
MID_GET_LOGFILE	0x60	H»M	2	获取 log 文件的大小
MID_UPLOAD_LOGFILE	0x61	H»M	2	将保存的 log 上传至主控
MID_SET_THRESHOLD_LEVEL	0xD4	H»M	2	设置算法安全等级
MID_POWERDOWN	0xED	H»M	2	模组断电前，保存简单的 log
MID_DEBUG_MODE	0xF0	H»M	2	使能 debug 模式， 会存储所有的图片和较多的 log。
MID_GET_DEBUG_INFO	0xF1	H»M	2	获取 debug 模式下存储的数据包大小
MID_UPLOAD_DEBUG_INFO	0xF2	H»M	2	上传 debug 模式下存储的数据包
MID_GETLIBRARY_VERSION	0xF3	M»H	2	获取当前解锁库版本信息
MID_DEMOMODE	0xFE	H»M	2	进入演示模式

1.4 模组消息的发送和接收

1.4.1 消息接收(H>>M)

模组接收到的完整协议格式如表 1-5 所示，主控应按照该格式向模组发送命令。

表 1-5

名称	SyncWord	MsgID	Size		Data	ParityCheck
字节数	2 bytes	1 byte	2 bytes		N bytes	1 byte
内容	0xEFAA	command	高八位	低八位	data	checksum

1.4.2 消息发送(M>>H)

模组主要发送三种类型的消息，分别是 REPLY，NOTE，IMAGE。

(1) REPLY 消息的发送

模组向主控发送的 REPLY 消息的完整协议如表 1-6 所示。

表 1-6

名称	SyncWord	MsgID	Size		Data			ParityCheck
字节数	2 bytes	1 byte	2 bytes		N bytes			1 byte
内容	0xEFAA	MID_REPLY (0x00)	高八位	低八位	s_msg_reply_data			checksum
					mid (1byte)	result (1byte)	data[0] (n-byte)	

mid 表示模组当前正在处理的任务，例如当 mid 为 MID_ENROLL_SINGLE 时，表示该消息是模组处理完单帧录入任务后回复的消息。

result 表示该命令的最终执行结果，详细如表 1-7 所示。

表 1-7

Result	code	说明
MR_SUCCESS	0	指令执行成功
MR_REJECTED	1	模组拒绝该命令
MR_ABORTED	2	录入/解锁算法已终止
MR_FAILED4_CAMERA	4	相机打开失败
MR_FAILED4_UNKNOWNREASON	5	未知错误
MR_FAILED4_INVALIDPARAM	6	无效的参数
MR_FAILED4_NOMEMORY	7	内存不足
MR_FAILED4_UNKNOWNUSER	8	未录入的用户
MR_FAILED4_MAXUSER	9	录入超过最大数量
MR_FAILED4_FACEENROLLED	10	人脸已录入
MR_FAILED4_LIVENESSCHECK	12	活体检测失败
MR_FAILED4_TIMEOUT	13	录入或解锁超时
MR_FAILED4_AUTHORIZATION	14	加密芯片授权失败
MR_FAILED4_READ_FILE	19	读文件失败
MR_FAILED4_WRITE_FILE	20	写文件失败
MR_FAILED4_NO_ENCRYPT	21	未采用加密通讯

(2) NOTE 消息的发送

NOTE 消息主要作用是主动向主控返回一些信息，目前 NOTE 消息主要在二种情况下发送：

a) 上电握手信号 NID_READY：模组上电初始化完成后，会通过串口向主控发送 NID_READY(EF AA 01 0001 00 00)。主机在接收到握手信号后，可以和模组进行指令交互。

b) License 验证失败 NID_AUTHORIZATION；

c) 录入/识别过程中模组向主控发送人脸状态信息 NID_FACE_STATE；

d) OTA 升级完成，携带升级成功结果 NID_OTA_DONE。

模组向主控发送的 NOTE 消息的完整协议如表 1-8 所示。

表 1-8

名称	SyncWord	MsgID	Size		Data		ParityCheck
字节数	2 bytes	1 byte	2 bytes		N bytes		1 byte
内容	0xEFAA	MID_NOTE (0x01)	高八位	低八位	s_msg_note_data		checksum
					nid (1byte)	data[0] (n-bytes)	

nid 表示几类 NOTE 信息，详细如下：

表 1-9

nid (*表示携带 data)	code	说明
NID_READY	0	模组已准备好
NID_FACE_STATE*	1	算法执行成功，并返回人脸信息 s_note_data_face
NID_UNKNOWNERROR	2	未知错误
NID_OTA_DONE*	3	OTA 升级完毕
NID_AUTHORIZATION	8	License 验证失败

其中 NID_FACE_STATE 是录入和比对过程中模组返回的一些人脸状态信息，它所携带的数据 data 主要存储了人脸信息，详细如下：

表 1-10

结构	s_note_data_face							
内容	state	left	top	Right	bottom	yaw	pitch	roll
类型	int16_t	int16_t	int16_t	int16_t	int16_t	int16_t	int16_t	int16_t
字节	2bytes	2bytes	2bytes	2bytes	2bytes	2bytes	2bytes	2bytes

state:表示人脸当前的状态，主要包括以下几种情况：

表 1-11

人脸状态(state)	code	说明

FACE_STATE_NORMAL	0	检测到人脸
FACE_STATE_NOFACE	1	未检测到人脸
FACE_STATE_TOOUP	2	人脸太靠近图片上边沿，未能录入
FACE_STATE_TOODOWN	3	人脸太靠近图片下边沿，未能录入
FACE_STATE_TOOLEFT	4	人脸太靠近图片左边沿，未能录入
FACE_STATE_TOORIGHT	5	人脸太靠近图片右边沿，未能录入
FACE_STATE_FAR	6	人脸距离太远，未能录入
FACE_STATE_CLOSE	7	人脸距离太近，未能录入
FACE_STATE_EYEBROW_OCCLUSION	8	眉毛遮挡
FACE_STATE_EYE_OCCLUSION	9	眼睛遮挡
FACE_STATE_FACE_OCCLUSION	10	脸部遮挡
FACE_STATE_DIRECTION_ERROR	11	录入人脸方向错误

s_note_data_face 中其它成员的具体含义如下所示：

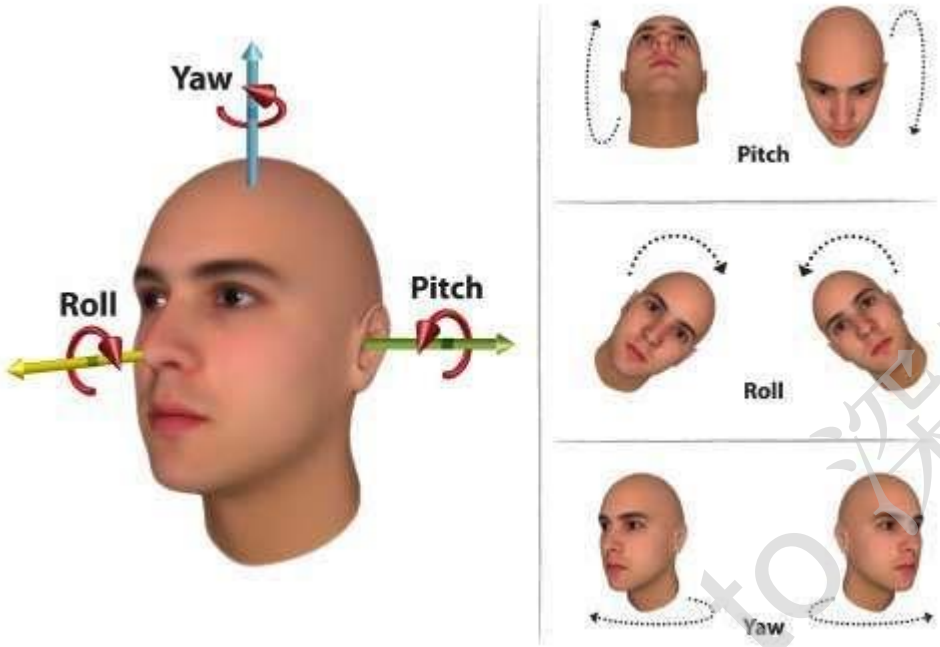
left：人脸框距离图片最左侧的距离（负数表示人脸框已超出图片最左侧）

top：人脸框距离图片最上方的距离（负数表示人脸框已超出图片上方）

right：人脸框距离图片最右侧的距离（负数表示人脸框已超图片最右侧）

bottom：人脸框距离图片最下方的距离（负数表示人脸框已超出图片最下方）

yaw，pitch，roll 表示的旋转方向分别如下图所示。其中 yaw 为负表示左转头，yaw 为正表示右转头；pitch 为负表示向上抬头，pitch 为正表示向下低头；roll 为负表示向右歪头，roll 为正表示向左歪头。



(3) IMAGE 消息发送

模组向主控发送的 IMAGE 消息的完整协议如表 1-11 所示。模组传输图片时，一帧数据最大支持 4000 字节数据的传输。

表 1-12

名称	SyncWord	MsgID	Size		Data	ParityCheck
字节数	2 bytes	1 byte	2 bytes		N bytes	1 byte
内容	0xEFAA	MID_IMAGE (0x02)	高八位	低八位	image data	checksum

1.5 上下电流程

主控主动控制模组的上电和下电，如图 1-1 所示。模组上电即进入启动流程， 启动进入 standby 状态会通知主控 MSG::NOTE::READY，主控开始发送命令消息，模组处理并返回结果。无消息发送时，主控调用 MSG::POWERDOWN，模组做处理返回，主控可断电。

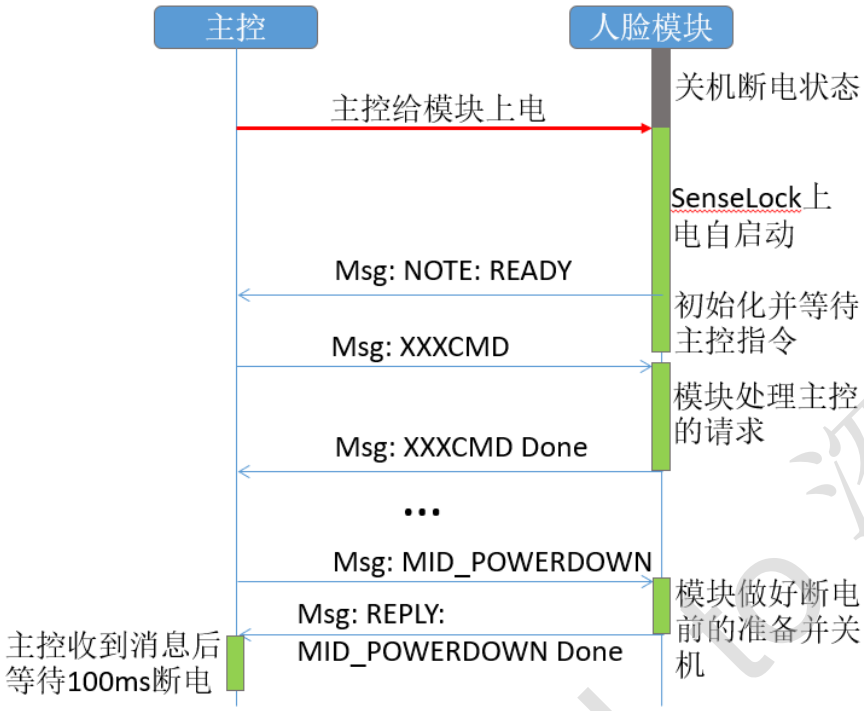


图 1-1

1.6 主控接收消息流程

主控接收模组发送的消息可以遵循图1-2所示流程。

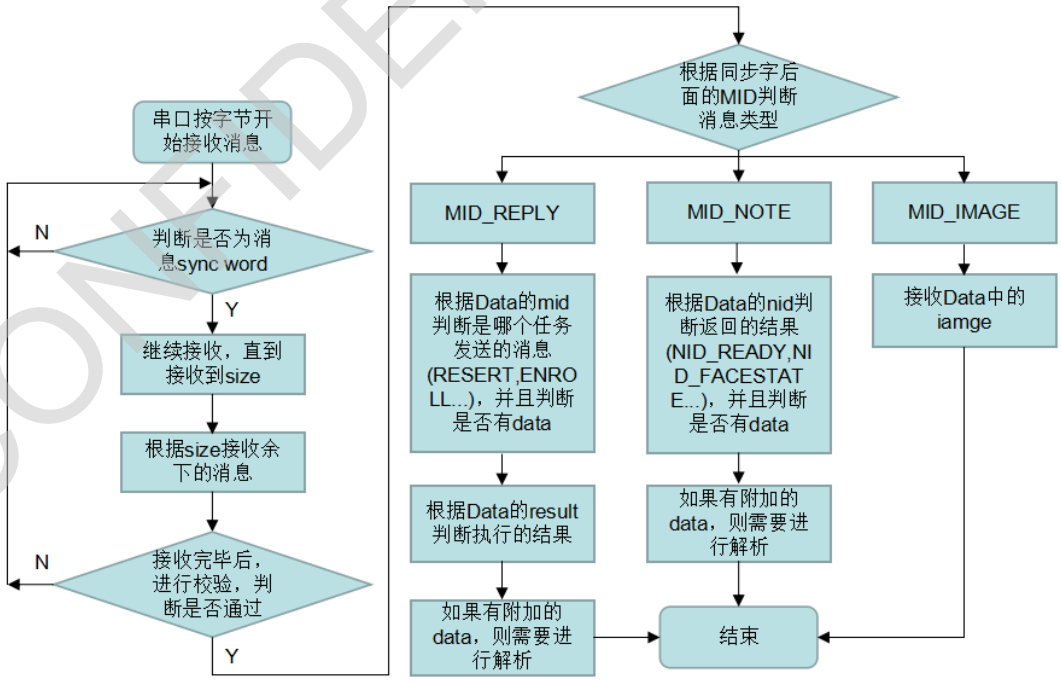


图 1-2

1.7 一般消息处理流程

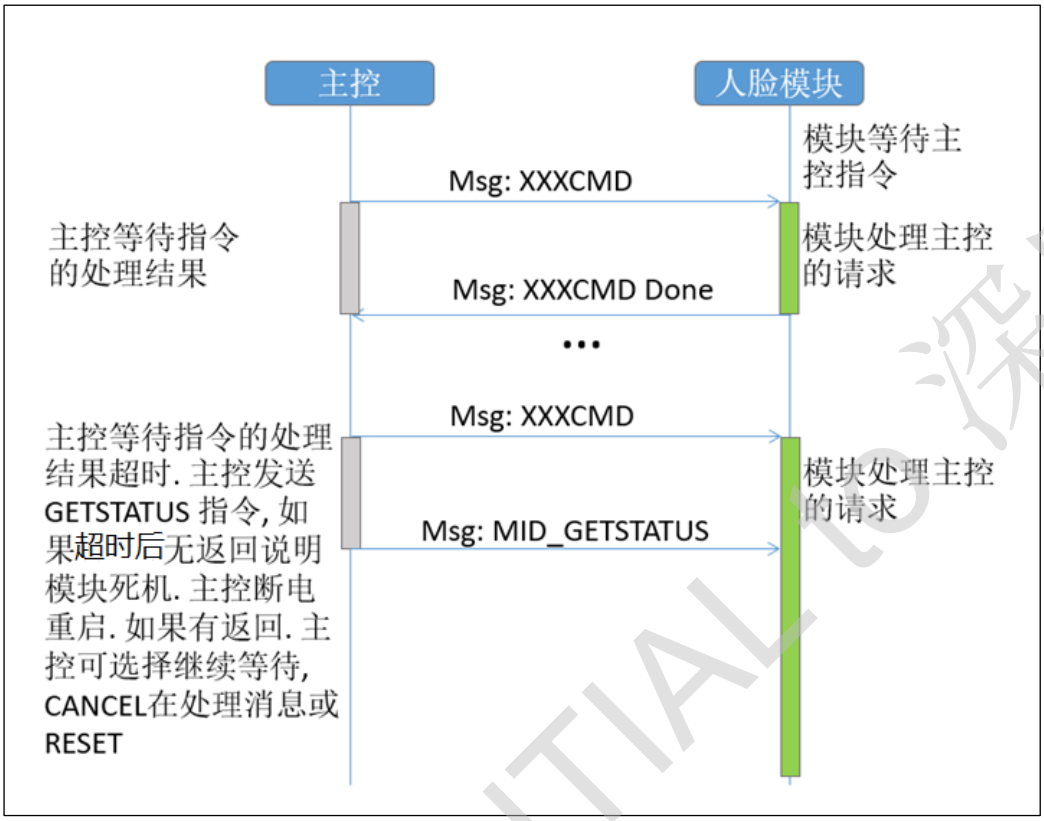


图 1-3

1.8 录入流程

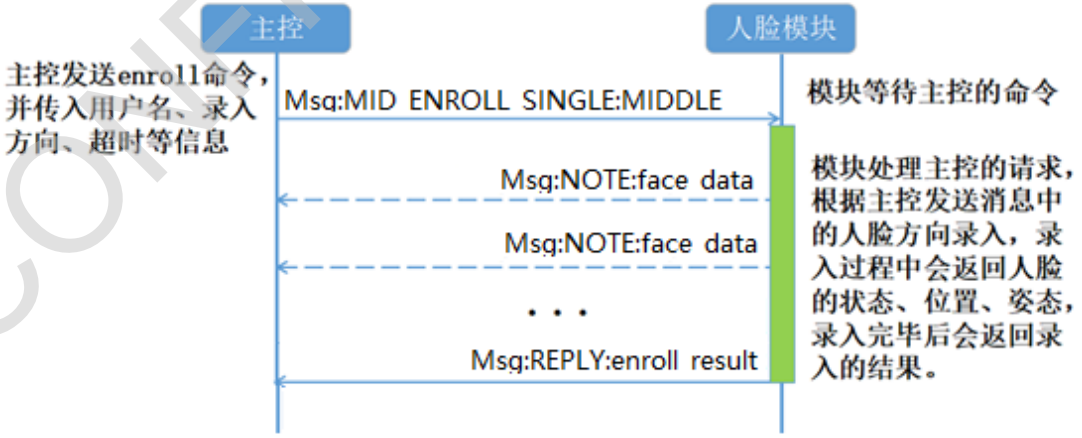


图 1-4

1.9 解锁流程

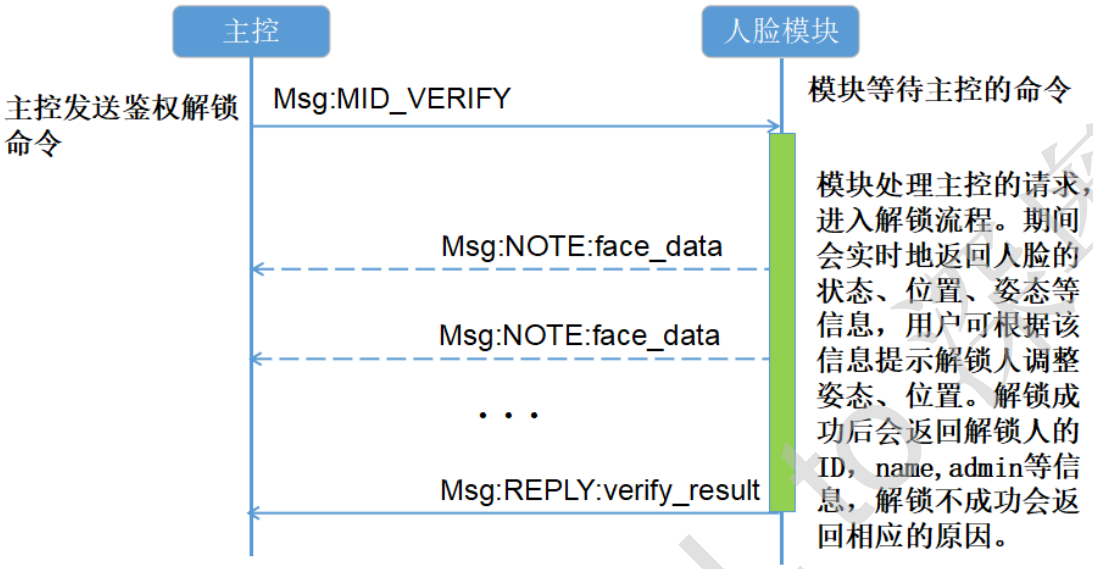


图 1-5

1.10 加密通信流程

模组和主控间通讯必须采用加密通讯，加密通信流程如图 1-6 所示。

① 主控给模组上电；
② 模组发送 NOTE READY(明文)给主控；
③ 主控给模组第一次上电开机需发送 MID_SET_RELEASE_ENC_KEY(明文)指令设置 16 Bytes 序列，后续上电不需再次发送。
④ 主控发送 MID_INIT_ENCRYPTION (明文)携带 4 Bytes 随机序列、加密模式给模组;
⑤ 模组和主控双方根据这随机序列，采用自定义的私有协议生成 16 Bytes 的密码。双方采用相同算法生成相同的密码。之后双方本次会话都采用此密码加解密;
⑥ 模组返回状态并采用随机密码 AES/SMPL 加密发送产品序列号给主控；
⑦ 主控解密，鉴别设备 ID 确认身份后开始处理需求指令，录入或解锁等；
⑧ 主控采用随机生成的密码，AES/SMPL 加密的方式对指令和数据加密并发送给模组；

⑨ 模组用第 4 步解密出来的密码解密，判断指令合法性并处理该指令；

⑩ 模组用第 4 步解密出来的密码对指令处理结果和数据加密，并发送给主控。

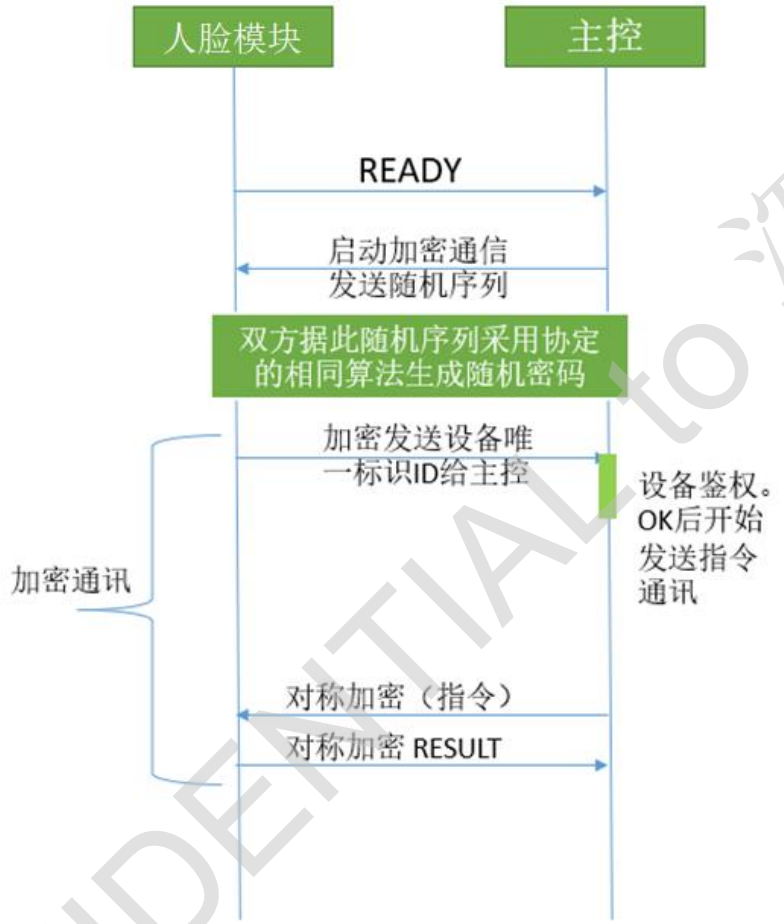


图 1-6

1.11 OTA 升级流程

① 主控发送 MID_START_OTA(密文)通知模组进入 OTA 模式；

② 模组反馈 OK(明文，后续都是明文模式)，模组进入 OTA 模式；

③ 主控发送 MID_CONFIG_BAUDRATE 配置更高的波特率，模组反馈 OK 后，主机同步设置相同波特率。延时 100ms 后才能开始串口数据传输。（该配置过程可选，波特率默认是 115200）

④ 主控发送 MID_OTA_HEADER，把固件相关信息发送给模组；

⑤ 模组反馈 OK；

⑥ 主控发送 MID_GET_OTA_STATUS；

⑦ 模组反馈 OTA 状态和起始包序号；

⑧ 主控循环发送升级包 MID_OTA_PACKET，收到 OK 后，继续下一包。

⑨ 模组收完固件后，开始升级。

⑩ 模组升级结束后，给主控反馈携带升级结果的 NOTE 信息(NID_OTA_DONE)，10 秒后模组自动重启。**主控发送完升级包后等待模组回复 NID_OTA_DONE 信息的超时时间须大于 120 秒。**

1.12 补充说明

1.12.1 设置加密规则 MID_SET_RELEASE_ENC_KEY

- 功能说明：主控和模组之间须采用加密通讯，该指令和 MID_INIT_ENCRYPTION 两个指令配合共同生成加密密钥。该指令传入的数据模组会保存，掉电不丢失。
- 输入参数：s_msg_enc_key_number_data
- 返回参数：无
- 指令代码：52H
- 指令包格式：

SyncWord	MsgID	Size		Data	ParityCheck
		H	L	s_msg_enc_key_number_data	
EFAA	52H	0010H		xx….xx	xxH

设置 Release Key 指令携带的数据 s_msg_enc_key_number_data 定义如下

```
typedef struct {
    uint8_t enc_key_number[16];     // 00 ~ 1F
} s_msg_enc_key_number_data;
```

- 指令包示例：

  EFAA 52 0010 **000102030405060708090A0B0C0D0E0F** 42

- 应答包格式：

SyncWord	MsgID	Size	Data	ParityCheck

		H	L	mid	result	data	
EFAA	00H	0002H		52H	xxH	无	xxH

注：result=00H(MR_SUCCESS) 表示设置 KEY 成功；

result=01H(MR_REJECTED) 表示 KEY 曾经已设置过；

result=06H(MR_FAILED4_INVALIDPARAM) 表示传入的参数错误。

1.12.2 进入加密模式 MID_INIT_ENCRYPTION

- 功能说明：该指令和 1.12.1 MID_SET_RELEASE_KEY 两个指令配合共同生成加密密钥，并采用设定的加密模式进行加密通讯。模组收到该指令后模组进入加密模式，将模组的 SN 号进行加密后回复给主控。主控收到数据采用相同的密钥、相同解密方式解密后得到明文数据。
- 输入参数：s_msg_init_encryption_data_v2
- 返回参数：无
- 指令代码：50H
- 指令包格式：

Sync Word	Msg ID	Size		Data					Parity Check
		H	L	s_msg_init_encryption_data_v2					
				seed[4]	mode	crttime[4]	power_mode	reserved	
EFAA	50H	0009H		xxxx xxxx	xx	xxxx xxxx	/	/	xxH

- 参数说明：
  1) seed：随机数，用于生成加密密钥使用；
  2) mode：加密模式。AES 加密：01H，简单加密模式：02H
  3) crttime：用于同步模组和主控的系统时间，该数组存储的内容代表当前时区的秒数(1970 年至今)
  4) power_mode：预留参数
  5) reserved[2]：预留参数

指令中携带参数 s_msg_init_encryption_data_v2 结构体定义如下：

```
typedef struct {
    uint8_t seed[4];
    uint8_t mode;
    uint8_t crttime[4];
    uint8_t power_mode;
    uint8_t reserved[2];
} s_msg_init_encryption_data_v2;
```

- 指令包示例

  EFAA 50 0009 **3C56B385 01** 5E689A73 DB

  指令释义：

  随机数 seed 为 3C56B385；采用 AES 加密模式(01)；时间戳 crtime 为 5E689A73。seed 的 md5 为 ee71535357ad9bb47cb6e1ba638c26a6，根据 MID_SET_RELEASE_ENC_KEY 设置的密钥规则，最终生成的加解密密钥为 ee71535357ad9bb4。

- 应答包格式：

SyncWord	Size		Data	Parity Check
	H	L	密文	
EFAA	00xxH		xx….xxH	xxH

- 应答包示例：

  EF AA 0020 **B9 87 85 D6 D6 D1 A6 72 B6 A4 21 B7 AD 7E 0F 98 83 BA 62 5B 52 83 50 2C C7 4A F9 AE 2C 2F 84 03** 8D

  0020 为应答包中加密数据的长度 size，粗体部分为模组回复的加密数据，主控对粗体部分数据进行解密得到下表中明文数据：00 0016 50 00 43 4F 42 41 31 33 38 39 30 31 32 34 30 37 38 39 33 32 31 即该模块的 SN: COBA138901240789321。

  即：模组加密的消息体为：明文指令中的 Size 和 Data 部分，主控加密操作也是类似。

Sync Word	Msg ID	Size		Data			Parity Check
		H	L	mid	result	data	
						s_msg_reply_init_encryption_data	
						device_id[20]	

EFAA	00H	**0016H**	**50H**	**00H**	**434F4241313338393031323 430373839 333231**	xxH

1.12.3 待机指令 MID_RESET

- 功能说明：主控发送这条命令后，模组将终止正在执行的命令（例如录入、解锁等），返回待机状态。
- 输入参数：无
- 返回参数：无
- 指令代码：10H
- 指令包格式：

SyncWord	MsgID	Size		Data	ParityCheck
		H	L		
EFAA	10H	0000H		无	10H

- 指令包示例：

  EFAA 10 0000 10

- 应答包格式：

SyncWord	MsgID	Size		Data			ParityCheck
		H	L	mid	result	data	
EFAA	00H	0002H		10H	xxH	/	xxH

- 应答包示例：

  EF AA 00 00 02 10 00 12

  注：result=00H 表示执行成功；其他为失败。

1.12.4 获取模组状态 MID_GETSTATUS

- 功能说明：主控发送该命令后，模组返回自身当前的状态。
- 输入参数：无
- 返回参数：s_msg_reply_getstatus_data
- 指令代码：11H
- 指令包格式：

SyncWord	MsgID	Size	Data	ParityCheck

		H	L		
EFAA	11H	0000H		无	11H

- 指令包示例：

  EFAA 11 0000 11

- 应答包格式：

Sync Word	Msg ID	Size		Data			Parity Check
		H	L	mid	result	data s_msg_reply_getstatus_data status	
EFAA	00H	0003H		11H	xxH	xxH	xxH

应答包携带数据 s_msg_reply_getstatus_data 结构体定义如下：

```
typedef struct {
    uint8_t status;
} s_msg_reply_getstatus_data;
```

- 应答包示例：

  EF AA 00 0003 11 00 **00** 12

  注：status=00H, MS_STANDBY 模组处于空闲状态，等待主控命令;

  status=01H, MS_BUSY 模组处于工作状态;

  status=02H, MS_ERROR 模组出错，不能正常工作;

  status=03H, MS_INVALID 模组未进行初始化。

1.12.5 识别 MID_VERIFY

- 功能说明：主控发送该指令给模组，模组开始识别人脸进行比对
- 输入参数： s_msg_verify_data
- 返回参数： s_msg_note_data 、s_msg_reply_verify_data
- 指令代码：12H
- 指令包格式：

SyncWord	MsgID	Size		Data		ParityCheck
		H	L	s_msg_verify_data		
				pd_rightaway	timeout	
EFAA	12H	0002H		00	xxH	xxH

识别指令携带数据 s_msg_verify_data 定义如下：

```
// verify
typedef struct {
    uint8_t pd_rightaway; // power down right away after verifying
    uint8_t timeout; // timeout, unit second, default 10s
} s_msg_verify_data;
```

- 识别指令参数说明：

  1）pd_rightaway 表示是否在解锁后立刻关机(预留参数)，默认设 00；

  2）timeout 为解锁超时时间(单位 s)，默认 10s，超时时间用户可自行设定，最大不超过 255s。**主控等待模组应答包超时时间应大于该参数设置值。**

- 应答包格式：

解锁过程中，模组应答 NOTE 和 REPLY 两种消息。一次识别过程中模组可能会应答多条 NOTE(MSGID=01H)来表示当前图像帧的人脸状态、位置和姿态，但最终只有一个 REPLY(MSGID=00H)来表示该次识别的最终结果。

- 应答 NOTE 消息格式：

Sync Word	Msg ID	Size		Data								Parity Check
				s_msg_note_data								
		H	L	nid	data							
					s_note_data_face							
					state	left	top	right	bottom	yaw	pitch	roll
EFAA	01H	xxx xH		xxH	xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx							xxH

注：nid=01H 表示 NID_FACE_STATE，表示模组返回的数据为当前帧的人脸状态；

NOTE 消息中携带的 s_note_data_face 结构体定义如下：

```
typedef struct {
    int16_t state; // corresponding to FACE_STATE_*
    // position
    int16_t left;    // in pixel
    int16_t top;
```

```
    int16_t right;
    int16_t bottom;
    // pose
    int16_t yaw;     // up and down in vertical orientation
    int16_t pitch; // right or left turned in horizontal orientation
    int16_t roll;    // slope
} s_note_data_face;
```

state 表示当前人脸状态 state，两个字节，**低字节在前，高字节在后**；state 的定义见表 1-11；

left、top、right、bottom 分别表示人脸的在图像中的位置，两个字节，**低字节在前，高字节在后**；

yaw、pitch、roll 表示人脸的姿态，分别为偏航角、俯仰角、翻滚角，两个字节，**低字节在前，高字节在后**；

- 应答 NOTE 消息示例：

  EFAA 01 0011 01 **0700** 00 00 00 00 00 00 00 00 00 00 00 00 00 16

  模组返回当前图像帧检测到的人脸状态为：**0700** 表示人脸距离过近。

- 识别指令 REPLY 消息格式：

Sync Word	Msg ID	Size		Data						Parity Check
		H	L	mid	result	data				
						s_msg_reply_verify_data				
						user_id	user_ name	admin	unlock Stauts	
EFAA	00H	0026H		10H	xxH	xxxx xxxx	xx.. xx	xxH	xxH	xxH

- 识别指令 REPLY 应答包参数说明：

  1) user_id：识别成功的用户 ID；
  2) user_name：识别成功的用户名字；
  3) admin：识别成功的用户是否为管理员用户；
  4) unlockStatus：识别成功时的人眼状态(保留参数)；

  识别成功 s_msg_reply_verify_data 结构体定义如下：

```
typedef struct {
```

```
    uint8_t user_id_heb;
    uint8_t user_id_leb;
    uint8_t user_name[USER_NAME_SIZE]; // 32Bytes
    uint8_t admin;
    uint8_t unlockStatus;
} s_msg_reply_verify_data;
```

- 识别返回结果 result 确认码的返回值见表 1-7
- 识别指令包示例：

  EFAA 00 0026 12 00 **0001**
  00000000000000000000000000000000000000000000000000000000000000000000000000 00 C8 FD，识别成功(00)，识别的用户 ID 为 **0001**，用户名为 0，该用户为普通用户 00。

  EF AA 00 0002 12 **0D** 1D

  指令释义：识别结果 **0D**，超时时间到，识别失败。

1.12.6 录入指令

- 功能说明：录入指令集中有三个录入指令。
  1) 单帧录入 MID_ENROLL_SINGLE，该指令只录入一张正脸即可，同一张人脸可以录入多次，生成不同的人脸 ID;
  2) 交互式录入 MID_ENROLL，该指令需录入上下左右中五个方向的人脸才完成一次录入生成一个人脸 ID；同一张人脸可录入多次；
  3) 综合录入 MID_ENROLL_ITG，该指令综合了单帧录入和交互式录入两种录入方式，通过传参形式控制录入方式以及同一张人脸是否可以重复录入。

1.12.6.1 单帧录入 MID_ENROLL_SINGLE

- 功能说明：单帧录入，仅录入正向人脸即可
- 输入参数：s_msg_enroll_data
- 返回参数：s_msg_reply_enroll_data
- 指令代码：1DH

- 指令包格式：

SyncWord	MsgID	Size		Data				Parity Check
		H	L	s_msg_enroll_data				
				admin	name	direction	timeout	
EFAA	1DH	0023H		00H	xxxx…	xxH	xxH	xxH

单帧录入指令携带参数 s_msg_enroll_data 定义如下：

```
typedef struct {
    uint8_t admin; // the user will be set to admin
    uint8_t user_name[32];
     s_face_dir face_direction; // depleted, user ignore this field
     uint8_t timeout;
} s_msg_enroll_data;
```

- 参数说明：

1) admin：设置改录入的人为管理员；
2) user_name：录入用户的用户名；
3) face_direction：无效参数，可设置为 00；
4) timeout：录入过程的超时时间(单位 s)，默认为 10s，用户可以随意设置，最大不超过 255s。**主控等待模组录入应答的超时时间应大于此参数设置值。**

- 单帧录入指令包示例：

EFAA 1D 0023 00 000000000000000000000000000000000000000000000000000000000000000000000000000000 00 **0A** 34

指令释义：设置录入的用户为普通用户 admin(00)，用户名为 0，人脸方向为 0，超时时间为 10(**0A**)秒。

- 应答包格式：

录入过程中同样也会返回 NOTE 和 REPLY 两种应答消息。NOTE 同 MID_VERIFY 指令中 NOTE 相关说明。

Sync Word	Msg ID	Size		Data				Parity Check
		H	L	mid	Result	data		
						s_msg_reply_enroll_data		
						user_id	face_direction	

EFAA	00H	0005H	1DH	xxH	xxxxxH	00H	xxH

- 录入结果 result 确认码的返回值见表 1-7

单帧录入成功模组返回数据 s_msg_reply_enroll_data 定义如下：

```
typedef struct {
    uint8_t user_id_heb;
    uint8_t user_id_leb;
    uint8_t face_direction; // depleted, user ignore this field
} s_msg_reply_enroll_data;
```

注：其中用户 ID 每次录入都会增加 1，即使删除某个 ID 后，再次录入 ID 会继续增加，直至 65530，ID 变为 0。

- 单帧录入应答包指令示例：

  EFAA 00 0005 1D 00 **0003** 00 1B，录成功 00，录入的用户 ID 为 **0003**

  EFAA 00 0005 1D **08** 10，识别失败，失败原因为该用户未录入 MR_FAILED4_UNKNOWNUSER (08H)

1.12.6.2 交互式录入 MID_ENROLL

- 功能说明：交互式录入，完成上中下左右 5 个方向的录入，最终生成一个人脸 ID
- 输入参数：s_msg_enroll_data
- 返回参数：s_msg_reply_enroll_data
- 指令代码：13H
- 指令包格式：

表 1-12-7-1 交互式录入指令包格式

SyncWord	MsgID	Size		Data				Parity Check
		H	L	s_msg_enroll_data				
				admin 1 Byte	name 32 Bytes	direction 1 Byte	timeout 1 Byte	
EFAA	13H	0023H		00H	xxxx…	xxH	xxH	xxH

交互式录入指令携带参数 s_msg_enroll_data 定义如下：

```
typedef struct {
    uint8_t admin; // the user will be set to admin
```

```
    uint8_t user_name[32];
     s_face_dir face_direction; // depleted, user ignore this field
     uint8_t timeout;
} s_msg_enroll_data;
```

- 参数说明：

5) admin：设置改录入的人为管理员；

6) user_name：录入用户的用户名；

7) face_direction：当前录入的人脸方向；

face_direction=10H(FACE_DIRECTION_UP)，表示当前录入朝上的人脸方向;

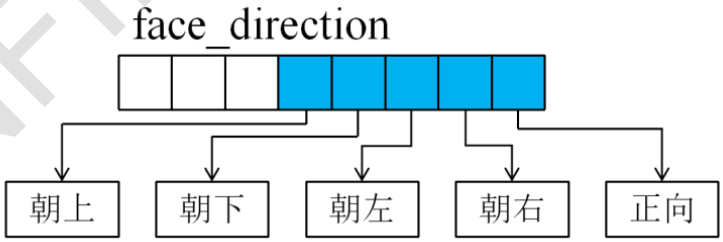
face_direction=08H(FACE_DIRECTION_ DOWN)，表示录入朝下的人脸方向;

face_direction=04H(FACE_DIRECTION_ LEFT)，表示录入朝左的人脸方向;

face_direction=02H(FACE_DIRECTION_ RIGHT)，表示录入朝右人脸方向;

face_direction=01H(FACE_DIRECTION_ MIDDLE)，表示录入居中人脸方向;

face_direction的低5位从高到低分别表示人脸方向的朝上、朝下、朝左、朝右和正向，如下图所示，1表示该朝向已录入，0表示该朝向未录入。



8) timeout：录入过程的超时时间(单位 s)，默认为 10s，用户可以随意设置，最大不超过 255s。主控等待模组录入应答的超时时间应大于等于此参数设置值。

- 交互式录入应答包格式：

Sync Word	Msg ID	Size		Data			Parity Check
		H	L	mid	result	data	

						s_msg_reply_enroll_data		
						user_id	face_direction	
EFAA	00H	0005H		13H	xxH	xxxxxH	xxH	xxH

- 交互式录入指令包&应答包示例：

  应答包同单帧 MID_ENROLL_SINGLE，NOTE 和 REPLY 两种。

  按照中右左下上的顺序录入一个普通用户，参考指令如下：

  ① **录入中间方向指令：**

  EFAA 13 0023 00

  000000000000000000000000000000000000000000000000000000000000000000000000 **01** 05 34

  录入中间方向 REPLY 应答包：

  EFAA 00 0005 13 00 FFFF **01** 17

  中间方向录入成功，face_direction=**0000 0001(BIN)**，最后 1 位置 1 表示已录入中间一个方向。

  ② **录入右向指令：**

  EFAA 13 0023

  000000000000000000000000000000000000000000000000000000000000000000000000 **02** 05 37

  录入右向 REPLY 应答包：

  EFAA 00 0005 13 00 FFFF **03** 15

  右向录入成功，face_direction=**0000 0011(BIN)**，后两位置 1 表示已经录入中间和右两个方向。

  ③ **录入左向指令：**

  EFAA 13 0023

  000000000000000000000000000000000000000000000000000000000000000000000000 **04** 05 31

  录入左向 REPLY 应答包：

  EFAA 00 0005 13 00 FFFF **07** 11

  右向录入成功，face_direction=**0000 0111(BIN)**，后三位置 1 表示已经录入中间、右、左三个方向。

  ④ **录入下向指令：**

EFAA 13 0023

00000000000000000000000000000000000000000000000000000000

00000000 **08** 05 3D

录入下向 REPLY 应答包：

EFAA 00 0005 13 00 FFFF **0F** 19

右向录入成功，face_direction=**0000 1111(BIN)**，后四位置 1 表示已经录入中间、右、左、下四个方向。

⑤ **录入上向指令：**

EFAA 13 0023

00000000000000000000000000000000000000000000000000000000

00000000 **10** 05 25

录入向上 REPLY 应答包：

EFAA 00 0005 13 00 **0003 1F** 0A

右向录入成功，face_direction=**0001 1111(BIN)**，后五位置 1 表示已经录入中间、右、左、下、上五个方向，并生成有效的 ID **0003**。

1.12.6.3 综合录入指令 MID_ENROLL_ITG

- 功能说明：该指令是集成了单帧录入指令 MID_ENROLL_SINGLE、交互录入指令 MID_ENROLL 等多种录入方式，录入方式可以在参数结构中指定，并可控制同一张人脸是否能够重复录入；
- 输入参数：s_msg_enroll_itg
- 返回参数：s_msg_reply_enroll_data
- 指令代码：26H
- 指令包格式：

表 1-13-6-1 单帧录入指令包格式

Sync Word	Msg ID	Size		Data							Parity Check
		H	L	s_msg_enroll_itg							
				admin	name	face_ direction	enroll_ type	enable_ duplicate	timeout	reserved[3]	
EFAA	26H	0025H		xxH	x..xH	xxH	xxH	xxH	xxH	xx.. xxH	xxH

ITG 综合录入指令携带参数 s_msg_enroll_itg 结构体定义如下:

```
typedef struct {
    uint8_t admin; // the user will be set to admin, 1:admin user, 0:normal user
    uint8_t user_name[USER_NAME_SIZE]; // 32Bytes
    uint8_t face_direction;
    uint8_t enroll_type;
     uint8_t enable_duplicate; // enable user enroll duplicatly, 1:enable, 0:disable
    uint8_t timeout;                    // timeout unit second default 10s
    uint8_t reserved[3];              // reserved field
} s_msg_enroll_itg;
```

- 参数说明
  1) admin：控制该录入用户是否为管理员用户；
  2) user_name：录入用户的用户名；
  3) enroll_type：采用的录入方式，0 表示交互式上中下左右 5 个方向的录入方式、1 表示单帧录入方式；
  4) face_direction：采用单帧录入时，参数无效，可以使用 0；采用交互式录入方式可参考交互式录入 MID_ENROLL 指令中的说明；
  5) enable_duplicate：0 表示同一个人不能重复录入，1 表示同一个人脸能够多次录入；username 都可以重复。
  6) timeout：录入过程的超时时间(单位 s)，默认为 10s，用户可以随意设置，最大不超过 255s。主控等待模组录入应答的超时时间应大于等于此参数设置值。
- 指令包示例：

  EFAA 26 0025 00 0000000000000000000000000000000000000000000000000000000000000000000000 00 **01 00** 05 07，采用单帧录入(**01**)方式录入一个名为 0 的普通用户，不能重复录入(**00**)。
- 应答包格式同 MID_ENROLL_SINGLE/MID_ENROLL
- 应答包示例：

  EF AA 00 00 05 26 00 **0004** 00 27

  EF AA 00 00 02 26 **0A** 2E

设置成不能重复录入时，成功录入一张人脸，ID 为 **0004**，再次录入该人脸模组反馈 MR_FAILED4_FACEENROLLED(**0A**)，错误码见表 1-7。

1.12.7 清除人脸录入状态 MID_FACERESET

- 功能说明：重置 SDK 算法状态，如果正在进行交互式录入 MID_ENROLL，发送该指令会重置录入的状态，已录入的方向全部被清空。
- 输入参数：无
- 返回参数：无
- 指令代码：23H
- 指令包格式：

SyncWord	MsgID	Size		Data	ParityCheck
		H	L		
EFAA	23H	0000H		无	xxH

- 应答包格式：

SyncWord	MsgID	Size		Data			ParityCheck
		H	L	mid	result	data	
EFAA	00H	0002H		23H	xxH	无	xxH

注：result=00H(MR_SUCCESS) 删除成功；

result=08H(MR_FAILED4_UNKNOWNREASON) 未知错误；

1.12.8 删除单个用户 MID_DELUSER

- 功能说明：通过传入用户 ID，删除指定 ID 的单个用户
- 输入参数：s_msg_deluser_data
- 返回参数：无
- 指令代码：20H
- 指令包格式：

SyncWord	MsgID	Size	Data	

		H	L	s_msg_deluser_data		ParityCheck
				user_id_heb	user_id_leb	
EFAA	20H	0002H		xxH	xxH	xxH

删除单个用户指令携带参数 s_msg_deluser_data 定义如下：

```
typedef struct {
    uint8_t user_id_heb;
    uint8_t user_id_leb;
} s_msg_deluser_data;
```

- 删除单个用户指令包示例：

  EFAA 20 0002 0006 24

  指令释义：删除 ID 为 0006 的人脸用户

- 应答包格式：

SyncWord	MsgID	Size		Data			ParityCheck
		H	L	mid	result	data	
EFAA	00H	0002H		20H	xxH	无	xxH

注：主控等待应答超时时间建议设置为 5 秒。

result=00H(MR_SUCCESS) 删除成功；

result=05H(MR_FAILED4_ UNKNOWNREASON) 未知错误；

result=08H(MR_FAILED4_UNKNOWNUSER) 删除的 ID 不存在；

1.12.9 删除所有用户 MID_DELALL

- 功能说明：删除所有已注册的用户
- 输入参数：无
- 返回参数：无
- 指令代码：21H
- 指令包格式：

SyncWord	MsgID	Size		Data	ParityCheck
		H	L		
EFAA	21H	0000H		无	xxH

- 删除所有用户指令示例：

  EFAA 21 0000 21

- 应答包格式：

SyncWord	MsgID	Size		Data			ParityCheck
		H	L	mid	result	data	
EFAA	00H	0002H		21H	xxH	无	xxH

- 删除所有用户应答示例：

  EFAA 00 0002 21 **00** 23，成功删除所有人脸用户

  EF AA 00 00 02 21 **05** 26，未录入过人脸的全新模组，模组内未生成人脸模板文件，导致删除失败。或出现其他的未知错误导致删除失败。

  注：result=00H(MR_SUCCESS) 删除成功；

  result=05H(MR_FAILED4_UNKNOWNREASON) 未知错误；

1.12.10 获取指定 ID 的用户信息 MID_GETUSERINFO

- 功能说明：主控通过 user id 指定要返回的注册用户，模组接收到该命令后会返回指定用户的信息
- 输入参数：s_msg_getuserinfo_data
- 返回参数：无
- 指令代码：22H
- 指令包格式：

SyncWord	MsgID	Size		Data		ParityCheck
		H	L	s_msg_getuserinfo_data		
				user_id_heb	user_id_leb	
EFAA	22H	0002H		xxH	xxH	xxH

获取指定 ID 的用户指令携带参数 s_msg_getuserinfo_data 定义如下：

```
typedef struct {
    uint8_t user_id_heb; // high eight bits of user_id to get info
    uint8_t user_id_leb; // low eight bits of user_id to get info
} s_msg_getuserinfo_data;
```

- 应答包格式：

SyncWord	Msg ID	Size		Data			Parity Check
		H	L	mid	result	data	

						s_msg_getuserinfo_data			
						user_id	user_name	admin	
EFAA	00H	xxxxH		22H	xxH	xxH	xx..xxH	xxH	xxH

成功获取指定 ID 的用户信息成功后，指令模组回复数据

s_msg_getuserinfo_data 定义如下：

```
typedef struct {
    uint8_t user_id_heb;
    uint8_t user_id_leb;
    uint8_t user_name[USER_NAME_SIZE]; // 32 Bytes
    uint8_t admin;
} s_msg_reply_getuserinfo_data;
```

注：result=00H(MR_SUCCESS) 获取用户信息成功；

result=05H(MR_FAILED4_UNKNOWNREASON) 未知错误；

result=08H(MR_FAILED4_UNKNOWNUSER) 该 ID 不存在；

1.12.11 获取已注册用户的 ID MID_GET_ALL_USERID

- 功能说明：获取所有已注册用户的数量和所有已注册用户的 ID。
- 输入参数：无
- 返回参数：s_msg_reply_all_userid_data
- 指令代码：24H
- 指令包格式：

SyncWord	MsgID	Size		Data	ParityCheck
		H	L		
EFAA	24H	0000H		无	24H

- 指令包示例：

  EFAA 24 0000 24

- 应答包格式：

SyncWord	MsgID	Size		Data			Parity Check
		H	L	mid	result	data	
						s_msg_reply_all_userid_data	
						user_counts	users_id[100]

EFAA	00H	002BH	24H	xxH	xxH	xx….xx	xxH

- 应答包示例：

  EF AA 00 0067 24 00 **03 0001 0002 0003** 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 40

指令释义：成功获取到模组已注册人脸个数 **3** 个，这 3 个人脸 ID 分别为 **0001 0002 0003**。

模组最大可以注册 50 个 ID，请注意主控端分配保存应答包数据的 buffer 大小。

1.12.12 演示模式 MID_DEMOMODE

- 功能说明：演示模式仅用于演示使用，安全性较低，任何人都可以解锁。掉电不保存，重启后模组恢复到正常模式。
- 输入参数：mode
- 返回参数：无
- 指令代码：FEH
- 指令包格式：

SyncWord	MsgID	Size		Data	ParityCheck
		H	L	mode	
EFAA	FEH	0001H		01H	FEH

- 指令报示例：

  进入演示模式：EFAA FE 0001 01 FE

  退出演示模式：EFAA FE 0001 00 FF

- 应答包格式：

SyncWord	MsgID	Size		Data			ParityCheck
		H	L	mid	result	data	
EFAA	00H	0002H		FEH	xxH	无	xxH

- 应答包示例：

  EFAA 00 0002 FE 00 FC

1.12.13 获取模组固件版本 MID_GET_VERSION

- 功能说明：获取模组固件版本
- 输入参数：无
- 返回参数：s_msg_reply_version_data
- 指令代码：30H
- 指令包格式：

SyncWord	MsgID	Size		Data	ParityCheck
		H	L		
EFAA	30H	0000H		无	xxH

- 指令包示例：

  EFAA 30 0000 30

- 应答包格式：

Sync Word	Msg ID	Size		Data			Parity Check
		H	L	mid	result	data	
						s_msg_reply_version_data	
						version_info[32]	
EFAA	00H	0002H		30H	xxH	xx….xx	xxH

应答包数据包携带数据 s_msg_reply_version_data 定义如下：

```
typedef struct {
    uint8_t version_info[VERSION_INFO_BUFFER_SIZE];    // 32 Bytes
} s_msg_reply_version_data;
```

- 应答包格式：

  EF AA 00 00 22 30 00 **53 45 56 32 5F 41 30 5F 44 42 47 2D 56 31 2E 30 2E 30 30 38 0A 00 00 00 00 00 00 00 00 00 00 00** 18

  转换成字符串，模组软件版本号为：SEV2_A0_DBG-V1.0.008

1.12.14 获取算法库版本 MID_GETLIBRARY_VERSION

- 功能说明：获取当前模组里人脸算法库的版本号信息。
- 输入参数：无
- 返回参数：s_msg_reply_library_version_data

- 指令代码：F3H
- 指令包格式：

SyncWord	MsgID	Size		Data	ParityCheck
		H	L		
EFAA	F3H	0000H		无	xxH

- 应答包格式：

SyncWord	MsgID	Size		Data		Parity Check
		H	L	s_msg_reply_library_version_data		
				library_version_info[20]		
EFAA	F3H	xxxxH		xx….xxH		xxH

应答包中携带数据 s_msg_reply_library_version_data 结构体如下：

```
typedef struct {
    uint8_t library_version_info[20];
} s_msg_reply_library_version_data;
```

1.12.15 设置安全等级 MID_SET_THRESHOLD_LEVEL

- 功能说明：修改算法安全等级，在不同厂商的与用户的理解中，对于安全的要求标准不一致，所以商汤开放安全等级接口，用于设置liveness和verify的安全等级，可设置四个等级
  低、较低、正常、较高、高。
- 输入参数：s_msg_algo_threshold_level
- 返回参数：无
- 指令代码：D4H
- 指令包格式：

SyncWord	MsgID	Size		Data		ParityCheck
		H	L	s_msg_algo_threshold_level		
				verify_threshold_level	liveness_threshold_level	
EFAA	D4H	0002H		xxH	xxH	xxH

指令包携带数据 s_msg_algo_threshold_level 数据结构体定义如下：

```
typedef struct {
```

```
    uint8_t verify_threshold_level;          // level 0~4, safety from low to high, default 2
    uint8_t liveness_threshold_level;        // level 0~4, safety from low to high, default 2
} s_msg_algo_threshold_level;
```

- 指令包示例：

  EFAA D4 0002 04 03 D1

  设置比对等级为较高等级，活体等级保持默认等级

- 应答包格式：

SyncWord	MsgID	Size		Data			ParityCheck
		H	L	mid	result	data	
EFAA	00H	0002H		D4H	xxH	无	xxH

- 应答包示例：

  EFAA 00 0002 D4 00 D6

声明：商汤推荐使用默认设置，安全等级过高易导致通过率降低；安全等级过低易导致误识问题，请慎用该设置。

1.12.16 图像抓拍

- 功能说明：图片抓拍，主控下达 MID_SNAPIMAGE 指令，模组会立刻进行抓拍，将抓拍到的图片按照起始编号依次向后命名并存于本地(jpg 格式)。最大支持本地存储的图片数量为 10 张，图片编号为 1-30。主控通过 MID_GETSAVEDIMAGE 和 MID_UPLOADIMAGE 指令获取抓拍的图像数据。

1.12.16.1 图像抓拍 MID_SNAPIMAGE

- 功能说明：图片抓拍，主控下达该指令，并携带需抓拍图片的数量以及抓拍图片的起始编号后，模组会立刻进行抓拍，将抓拍到的图片按照起始编号依次向后命名并存于本地(jpg 格式)。最大支持本地存储的图片数量为 10 张，图片编号为 1-30，当主控发送的编号与本地图片编号重复时，将覆盖原始图片。
- 输入参数：s_msg_snap_image_data
- 返回参数：无
- 指令代码：16H

- 指令包格式：

SyncWord	MsgID	Size		Data		ParityCheck
		H	L	s_msg_snap_image_data		
				image_counts	start_number	
EFAA	16H	0002H		xxH	xxH	xxH

- 参数说明：

  1) image_counts：抓拍图像的数量；

  2) start_number：抓取图像的名称编号。

- 指令包示例：

  EFAA 16 0002 05 03 12，抓取 5 张(05)图像，起始编号为 03，即保存图像名为：3.jpg、4.jpg、5.jpg、6.jpg、7.jpg 的 5 张图像。

  EFAA 16 0002 05 30 21，抓取 5 张(05)图像，起始编号为 30，即保存图像名为：30.jpg、1.jpg、2.jpg、3.jpg、4.jpg 的 5 张图像。

- 应答包格式：

SyncWord	MsgID	Size		Data			ParityCheck
		H	L	mid	result	data	
EFAA	00H	0002H		16H	xxH	无	xxH

- 应答包示例

  EF AA 00 00 02 16 00 14

  注：result=00H(MR_SUCCESS) 抓取图像成功；

  result=04H(MR_FAILED4_CAMERA) Camera 启动失败；

  result=05H(MR_FAILED4_UNKNOWNREASON) 未知错误；

  result=06H(MR_FAILED4_INVALIDPARAM) 指令包传参错误；

  result=14H(MR_FAILED4_WRITE_FILE) 写文件失败；

1.12.16.2 获取抓拍图像大小 MID_GETSAVEDIMAGE

- 功能说明：MID_SNAPIMAGE 抓拍的图片将存储在模组中，主控获取抓拍图片时，首先调用 MID_GETSAVEDIMAGE 命令获取待上传图片的大小，然后多次调用 MID_UPLOADIMAGE 来分包获取图片数据。
- 输入参数：s_msg_get_saved_image_data
- 返回参数：无

- 指令代码：17H
- 指令包格式：

SyncWord	MsgID	Size		Data	ParityCheck
		H	L	s_msg_get_saved_image_data	
				image_number	
EFAA	17H	0001H		xxH	xxH

- 参数说明：

  1) image_number：指定要获取的图像的编号。

- 指令包示例：

  EF AA 17 0001 03 15，获取编号为 03 即 3.jpg 的图像大小。

- 应答包格式：

SyncWord	MsgID	Size		Data			Parity Check
		H	L	mid	result	s_msg_reply_get_saved_image_data	
						image_size[4]	
EFAA	00H	xxxxH		17H	xxH	xxxxxxxxH	xxH

应答包携带数据 s_msg_reply_get_saved_image_data 定义如下：

```
typedef struct {
    uint8_t image_size[4];     // image size int -> [s1, s2, s3, s4]
} s_msg_reply_get_saved_image_data;
```

- 应答包示例

  EFAA 00 0006 17 00 00003B5B 71，获取编号为 03 的图像大小为 (0x00003B5B) 15195 字节。

  注：result=00H(MR_SUCCESS) 获取图像大小成功；

  result=13H(MR_FAILED4_READ_FILE) 读取图像文件失败；

1.12.16.3 上传抓拍图像数据 MID_UPLOADIMAGE

- 功能说明：根据 MID_GETSAVEDIMAGE 获取到的图像大小，通过串口分包上传抓拍图像给主控。
- 输入参数：s_msg_upload_image_data
- 返回参数：无
- 指令代码：18H

- 指令包格式：

SyncWord	Msg ID	Size		Data		Parity Check
		H	L	s_msg_upload_image_data		
				upload_image_offset[4]	upload_image_size[4]	
EFAA	18H	xxxx		xxxxxxxxH	xxxxxxxxH	xxH

指令包中携带的 s_msg_upload_image_data 结构体定义如下：

```
typedef struct {
    uint8_t upload_image_offset[4]; // upload image offset, int -> [o1 o2 o3 o4]
    uint8_t upload_image_size[4];     // upload image size, int -> [s1 s2 s3 s4]
} s_msg_upload_image_data;
```

- 参数说明：
  1) upload_image_offset[4]：表示该次上传图像的偏移量；
  2) upload_image_size[4]：表示该次上传图像的大小，最大不超过 4000 字节；
- 指令包示例：

  EFAA 18 0008 00000000 00000FA0 BF

  EFAA 18 0008 00000FA0 00000FA0 10

  EFAA 18 0008 00001F40 00000FA0 E0

  EFAA 18 0008 00002EE0 00000C7B A9

  模组上传大小为(0x3B5B)15195 字节的图像给主控，分 4 次发送，每次传送数据大小分别为 4000、4000、4000、3195 字节。
- 应答包格式：

SyncWord	MsgID	Size		Data			ParityCheck
		H	L	mid	result	data	
EFAA	02H	xxxxH		18H	xxH	无	xxH

- 应答包示例：

EF AA 02 0FA0 xxxxxx….xx

注：MsgID=02H(MID_IMAGE) 模组给主控传送图片

result=00H(MR_SUCCESS) 抓取图像成功；

result=06H(MR_FAILED4_INVALIDPARAM) 指令包传参错误；

1.12.17 获取断电保存的日志

- 功能说明： 模组断电前发送 MID_POWERDOWN 指令，模组收到该指令后，模组会保存简单的日志文件，主控通过 MID_GET_LOGFILE & MID_UPLOAD_LOGFILE 获取日志文件。

1.12.17.1 断电保存日志 MID_POWERDOWN

- 功能说明： 模组断电前发送该指令，模组收到该指令后，模组保存相应的 log。
- 输入参数：无
- 返回参数：无
- 指令代码：EDH
- 指令包格式：

SyncWord	MsgID	Size		Data	ParityCheck
		H	L		
EFAA	EDH	0000H		无	xxH

- 应答包格式：

SyncWord	MsgID	Size		Data			ParityCheck
		H	L	mid	result	data	
EFAA	00H	0002H		EDH	xxH	无	xxH

1.12.17.2 获取日志文件大小 MID_GET_LOGFILE

- 功能说明：MID_POWERDOWN 模组保存日志，主控获取模组 log，首先调用 MID_GET_LOGFILE 命令获取待上传 log 文件的大小，然后多次调用 MID_UPLOAD_LOGFILE 来分包获取 log 文件。使用方式与 MID_GETSAVDIMAGE & MID_UPLOADIMAGE 的使用方式完全相同。
- 输入参数：无
- 返回参数：s_msg_reply_get_log_data
- 指令代码：60H
- 指令包格式：

SyncWord	MsgID	Size	Data	ParityCheck

		H	L		
EFAA	60H	0000H		无	xxH

- 指令包示例：

  EFAA 60 0000 60

- 应答包格式：

SyncWord	MsgID	Size		Data			Parity Check
		H	L	mid	result	s_msg_reply_get_log_data	
						log _size[4]	
EFAA	00H	xxxxH		60H	xxH	xxxxxxxxH	xxH

应答包携带数据 s_msg_reply_get_log_data 定义如下：

```
typedef struct {
    uint8_t log_size[4];    // log size int -> [s1, s2, s3, s4]
} s_msg_reply_get_log_data;
```

- 应答包示例：

  EF AA 00 0006 60 00 00002F18 51，获取的日志文件大小为：00002F18(12056 Bytes)

1.12.17.3 获取日志文件 MID_UPLOAD_LOGFILE

- 功能说明：MID_POWERDOWN 模组保存日志，主控获取模组 log，首先调用 MID_GET_LOGFILE 命令获取待上传 log 文件的大小，然后多次调用 MID_UPLOAD_LOGFILE 来分包获取 log 文件。上传上来的 log 文件是一个 tar 包， 需要先解包再查看里面的内容。使用方式与 MID_GETSAVDIMAGE & MID_UPLOADIMAGE 的使用方式完全相同。
- 输入参数：无
- 返回参数：s_msg_upload_logfile_data
- 指令代码：61H
- 指令包格式：

SyncWord	Msg ID	Size		Data		Parity Check
		H	L	s_msg_upload_logfile_data		
				upload_image_offset[4]	upload_image_size[4]	

EFAA	61H	xxxx	xxxxxxxxH	xxxxxxxxH	xxH

指令包中携带的 s_msg_upload_ logfile_data 结构体定义如下：

```
typedef struct {
    uint8_t upload_logfile_offset[4]; // upload logfile offset, int -> [o1 o2 o3 o4]
    uint8_t upload_logfile_size[4];     // uploade logfile size, int -> [s1 s2 s3 s4]
} s_msg_upload_logfile_data;
```

- 参数说明：
  3) upload_ logfile _offset[4]：表示该次上传日志的偏移量；
  4) upload_ logfile _size[4]：表示该次上传日志的大小，最大不超过 4000 字节；
- 指令包示例：

  EF AA 61 0008 00000000 00000FA0 C6

  EF AA 61 0008 00000FA0 00000FA0 69

  EF AA 61 0008 00001F40 00000FA0 99

  EF AA 61 0008 00 002EE0 00000038 9F

  模组上传大小为 12056 字节的日志文件给主控，分 4 次发送，每次传送数据大小分别为 4000、4000、4000、56 字节。
- 应答包格式：

SyncWord	MsgID	Size		Data			ParityCheck
		H	L	mid	result	data	
EFAA	00H	xxxxH		18H	xxH	无	xxH

- 应答包示例：

  EF AA 00 0FA0 xxxxxx….xx

1.12.18 获取售后 debug 日志

- 功能说明：售后使用该功能，在终端用户配合下进入 debug 模式然后复现用户出现的问题，通过 MID_GET_DEBUG_INFO 和 MID_UPLOAD_DEBUG_INFO 指令获取相应的日志和图像数据，退出 debug 模式时会删除所有 debug 模式中保存的数据。

1.12.18.1 进入 debug 模式 MID_DEBUG_MODE

- 功能说明：进入/退出 debug 模式
- 输入参数：无
- 返回参数：无
- 指令代码：F0H
- 指令包格式：

SyncWord	MsgID	Size		Data	ParityCheck
		H	L	s_msg_debug_mode_data	
				mode	
EFAA	F0H	0001H		xxH	xxH

指令包中携带的 s_msg_debug_mode_data 结构体定义如下：

```
typedef struct {
    uint8_t enable; // enable debug mode or not
} s_msg_debug_mode_data;
```

- 指令包示例:

  进入 debug 模式：EFAA F0 0001 01 F0

  退出 debug 模式：EFAA F0 0001 00 F1

- 应答包格式：

SyncWord	MsgID	Size		Data			ParityCheck
		H	L	mid	result	data	
EFAA	00H	xxxxH		F0H	xxH	无	xxH

- 应答包示例：

  EF AA 00 0002 F0 00 F2

1.12.18.2 获取 debug 数据 MID_GET_DEBUG_INFO& MID_UPLOAD_DEBUG_INFO

- 功能说明：售后使用该功能，在终端用户配合下打开 debug 模式然后复现用户出现的问题，通过 MID_GET_DEBUG_INFO 和 MID_UPLOAD_DEBUG_INFO 指令获取相应的日志和图像数据，退出 debug 模式时会删除所有 debug 模式中保存的数据。这两条指令的使用方式与 MID_GETSAVDIMAGE & MID_UPLOADIMAGE 的使用方式完全相同，请参考。