



FP002 Facial and Palm Vein Recognition Module User Manual



Hangzhou Grow Technology Co., Ltd.

2025.10 Ver: 1.0

Preface & Declaration

Thank you for your selection of FP002 Facial and Palm Vein Recognition Module of GROW.

The Manual is targeted for hardware & software development engineer, covering module function, hardware and software interface etc. To ensure the developing process goes smoothly, it is highly recommended the Manual is read through carefully.

Because of the products constantly upgraded and improved, module and the manual content may be changed without prior notice. If you want to get the latest information, please visit our company website (www.hzgrow.com).

We have been trying our best to ensure you the correctness of the Manual. However, if you have any question or find error, feel free to contact us or the authorized agent. We would be very grateful.

The Manual contains proprietary information of Hangzhou Grow Technology Co., Ltd., which shall not be used by or disclosed to third parties without the permission of GROW, nor for any reproduction and alteration of information without any associated warranties, conditions, limitations, or notices.

No responsibility or liability is assumed by GROW for the application or use, nor for any infringements of patents or other intellectual property rights of third parties that may result from its use.

www.hzgrow.com

Revised Version

Version Number	Date	Revise Content	Modifier
V1.0	2025.10	Set up	Grow Tech

Catalog

1. Introduction of Module	1 -
1.1 Introduction	1 -
1.2 Technical Specification	2 -
1.3 Dimension&Pin Definition	3 -
1.3.1 Dimension(mm)	3 -
1.3.2 Pin Definition	3 -
2. Communication Interface	4 -
2.1 Serial Communication Interface	4 -
2.2 Communication Message Format	5 -
2.3 Communication Message Description	6 -
3. Sending and Receiving of Module Messages	7 -
3.1 Message Receiving	7 -
3.2 Message Sending	7 -
3.2.1 Sending of the REPLY Message	7 -
3.2.2 Sending of the NOTE Message	9 -
4. Communication Message Explanation	14 -
4.1 Command Data Format	14 -
4.2 Face Orientation Parameters	22 -
4.3 Message Instructions Format	24 -
4.3.1 Standby Instruction MID_RESET	24 -
4.3.2 Get Module Status MID_GETSTATUS	24 -
4.3.3 Face/Palm Vein Recognition MID_VERIFY	25 -
4.3.4 Single Frame Recording MID_ENROLL_SINGLE	29 -
4.3.5 Interactive Input MID_ENROLL	30 -
4.3.6 Capture Pictures and Store Locally MID_SNAPIMAGE	33 -
4.3.7 Get Uploaded Images MID_GETSAVEDIMAGE	33 -
4.3.8 Upload Local Storage Images To the Host MID_UPLOADIMAGE	34 -
4.3.9 Deleting a Single User MID_DELUSER	34 -
4.3.10 Delete All Users MID_DELALL	35 -
4.3.11 Get User Information For a Specified ID MID_GETUSERINFO	36 -
4.3.12 Get the ID of a Registered User MID_GET_ALL_USERID	37 -
4.3.13 Comprehensive Input MID_ENROLL_ITG	38 -
4.3.14 Get the Module Software Version MID_GET_VERSION	40 -
4.3.15 Set the Encryption Random Number MID_INIT_ENCRYPTION	41 -
4.3.16 Set the Encryption Rules MID_SET_RELEASE_ENC_KEY	42 -
4.3.17 Set the Security Level MID_SET_THRESHOLD_LEVEL	43 -
4.3.18 Demo Mode MID_DEMOMODE	44 -
4.3.19 LED Setting MID_LED CONTROL	45 -
4.3.20 Set the Upload Image Parameters MID_CAPTURE_PIC_TYPE	46 -
4.3.21 Upload User Template MID_UPLOADUSERCHAR	47 -
4.3.22 Download User Template MID_DOWNLOADUSERCHAR	47 -
4.3.23 Save User Template MID_SAVEDOWNLOADUSERCHAR	48 -
5. Application Process	49 -
5.1 Host Receiving Message Process	49 -

5.2 General Message Processing Flow	- 50 -
5.3 Power On and Off Process	- 51 -
5.4 Input Process	- 52 -
5.4.1 Face Interactive Input	- 52 -
5.4.2 Single Frame Input	- 54 -
5.5 Unlocking Process	- 54 -
5.5.1 Unlocking Process Explanation	- 54 -
5.6 Encrypted Communication Process	- 55 -
5.7 Upload Template Process	- 57 -
5.8 Download Template Process	- 58 -
5.8.1 Upload/Download Example	- 59 -
5.8.1.1 Upload Face User Template	- 59 -
5.8.1.2 Download Face User Template	- 63 -
5.8.1.3 Store the template data downloaded to the module cache into position 5 of the FLASH.	- 66 -

1. Introduction of Module

1.1 Introduction

The FP002 facial and palm vein recognition module is equipped with a bionic facial recognition algorithm and supports infrared detection, which not only ensures security but also provides a good user experience. It has the characteristics of low power consumption, low cost, fast recognition, ingenious product structure design, and easy installation. It uses a high-computing power chip to quickly complete the entire process of facial recognition and palm vein recognition.

Features include:

Supports live face recognition, offering simple operation, fast recognition, high accuracy, and strong adaptability.

Low-cost/high-performance: Using facial/palm vein recognition modules, this solution achieves both low cost and high performance.

Supports infrared detection, allowing for flexible adjustment of detection distance based on application scenarios.



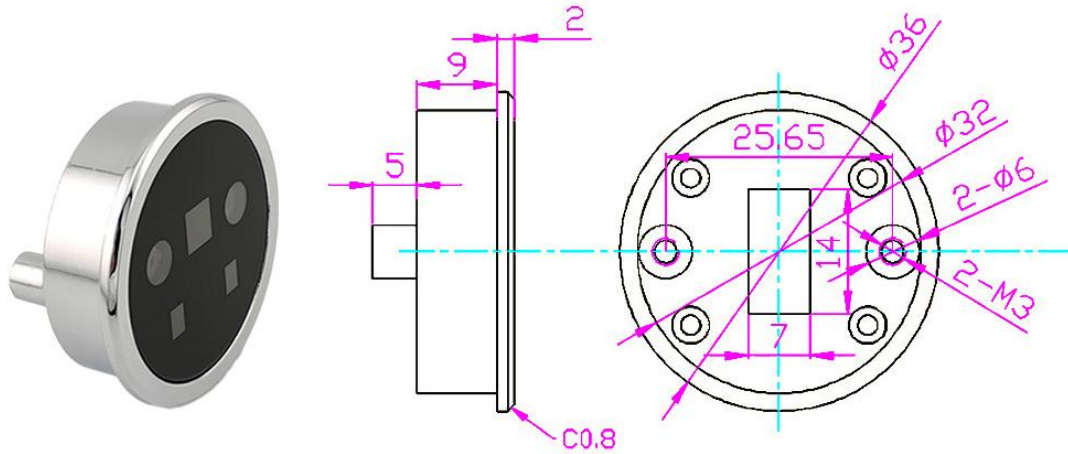
Note: When installing the module, the lens can only be kept horizontal with the face (as shown in the figure), and the lens direction cannot be tilted or reversed.

1.2 Technical Specification

Parameter	Performance
Product Size	Outer diameter 36mm, Height 11mm Mounting hole diameter 32mm
Capacity	Face: 500(ID:1-500) Palm Vein: 300 (ID501-800)
Reading Distance	Face registration distance: 20-90cm (optimal range: 45-65cm) Face search distance: 20-130cm Palm vein registration distance: 15-20cm Palm vein search distance: 10-20cm
Infrared Detection Distance	0-70cm (At 3.3V, infrared trigger standby average current 60uA peak power 88uA)
Indicator	Red,Green,White
Cold Start Speed	≤250ms
FRR	<0.1%
FAR	<0.001%
UART Baud Rate	115200bps
Interface	UART
Operating Voltage	DC 4.5-12V
Operating Current	200mA
Standby Current	38mA
Operating Temperature	-20°C ~ +50°C
Storage Temperature	-40°C ~ +70°C
Operating Humidity	<95%
Storage Humidity	<95%

1.3 Dimension&Pin Definition

1.3.1 Dimension(mm)



1.3.2 Pin Definition

Connector Size: MX 1.25mm 6pin

Pin	Name	Description	Pic
1	VCC	Power input 4.5-12V	
2	GND	Signal ground	
3	TXD	Data output. TTL logical level	
4	RXD	Data input. TTL logical level	
5	IR	Infrared sensing, outputs high level when receiving induction	
6	VCC	3-5V infrared power input	

2. Communication Interface

The module(M) and host(H) communicate via serial interface.

2.1 Serial Communication Interface

Baud rate: 115200bps Data bit:8 Stop bit:1 Parity bit: N

Parameters	Instructions
Baud rate	115200
Hardware/software flow control	N
Data bit	8
Stop bit	1
Parity bit	N
Signal level	3.3V

2.2 Communication Message Format

The basic message format for communication between the host and the module is shown in Table 2.

Table 2

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

Table 3 provides detailed description.

Table 3

Item	Length	Instruction
SyncWord	2bytes	Fixed message start synchronization word: 0xEF 0xAA
MsgID	1byte	Message ID (e.g:RESET)
Size	2bytes	Data size, unit: byte
Data	N bytes	Corresponding data, e.g.: Corresponding parameters for command message 65535>=N>=0,N=0 Indicates that this message has no parameters
Parity Check	1 byte	The parity check code of the protocol. The calculation method is to remove the Sync Word part of the entire protocol, and then perform an XOR operation on each bit of the remaining bytes.

Example: Get the number and ID of registered users

SyncWord	MsgID	Size		Data	ParityCheck
		H	L	fmt	
2bytes	1byte	1byte	1byte	1byte	1byte
EFAA	24H	0001H		02	27H

That is ^0x24 ^0x00 ^0x01 ^0x02 ParityCheck 0x27

2.3 Communication Message Description

Each message processing operation has a corresponding timeout defined. The timeout period refers to the maximum time the host is willing to wait for the module to complete the processing, which is the time when the module indicates that the processing is completed.

When command processing times out, the host can take the following actions:

1. Send the MID_GETSTATUS command

This message can quickly obtain the working status of the module. If it crashes, there will be no return and the host can cut off the power directly.

2. Send the MID_RESET command

Stop all current processing. The module enters the standby state and waits for new commands from the host.

3. Consider the module as crashed

The host can adopt the method of power-off restart.

3. Sending and Receiving of Module Messages

3.1 Message Receiving

The complete protocol format received by the module is shown in Table 4. The host should send commands to the module according to this format.

Table 4

Item	SyncWord	MsgID	Size		Data	ParityCheck
Bytes	2 bytes	1 byte	2 bytes		N bytes	1 byte
Content	0xEFAA	command	high 8 bits	low 8 bits	data	checksum

For detailed description of command and data, please refer to [Table 11]

3.2 Message Sending(M>>H)

The module mainly sends two types of messages: REPLY and NOTE.

3.2.1 Sending of the REPLY Message

REPLY is the response sent by the module to the command issued by the host. For each command issued by the host, the module will eventually send a REPLY.

Every command sent by the host to the module will eventually receive a MID_Reply reply from the module. This message contains the command “mid” sent by the host, the execution result of the command-”result”, as well as the possible returned data-”data”.

The complete protocol of the REPLY message sent by the module to the host is shown in Table 5.

Table 5

Item	SyncWord	MsgID	Size		Data			ParityCheck
Bytes	2bytes	1byte	2bytes		N bytes			1 byte
Content	0xEFAA	MID_REPLY (0x00)	high 8 bits	low 8 bits	Mid (1byte)	Result (1byte)	Data (n-byte)	checksum

mid represents the current task being processed by the module. For example, when mid = MID_ENROLL(0x13), it indicates that this message is the reply sent by the module after completing the enroll task.

result represents the execution result of this command. The details are shown in Table 6.

Table 6

result	code	Instruction
MR_SUCCESS	0 [0x00]	Successfully
MR_REJECTED	1 [0x01]	The module rejects the command
MR_ABORTED	2 [0x02]	The input/unlock algorithm has been terminated
MR_FAILED4_CAMERA	4 [0x04]	The camera failed to open.
MR_FAILED4_UNKNOWNREASON	5 [0x05]	Unknown error
MR_FAILED4_INVALIDPARAM	6 [0x06]	Invalid parameter
MR_FAILED4_NOMEMORY	7 [0x07]	Out of Memory
MR_FAILED4_UNKNOWNUSER	8 [0x08]	There are no registered users
MR_FAILED4_MAXUSER	9 [0x09]	The maximum number of users entered has exceeded
MR_FAILED4_FACEENROLLED	10 [0x0A]	The face has been recorded.
MR_FAILED4_LIVENESSCHECK	12 [0x0C]	Live detection failed
MR_FAILED4_TIMEOUT	13 [0x0D]	Input or unlock timeout
MR_FAILED4_AUTHORIZATION	14 [0x0E]	The authorization of the encryption chip failed
MR_FAILED4_READ_FILE	19 [0x13]	Failed to read the file
MR_FAILED4_WRITE_FILE	20 [0x14]	Failed to write the file
MR_FAILED4_NO_ENCRYPT	21 [0x15]	The communication protocol is not encrypted
MR_FAILED4_NO_RGBIMAGE	23 [0x17]	The RGB image is not ready
MR_FAILED4_UNKNOWN_HANDUSER	239 [0xEF]	Palm vein verification failed
MR_FAILED4_NOCAMERA	240 [0xF0]	Color lenses are not supported
MR_FAILED4_HANDENROLLED	241 [0xF1]	Palm vein duplication

For example: 0xEF 0xAA 0x00 0x00 0x04 0x13 0x00 0x00 0x01 0x16

Item	SyncWord	MsgID	Size		Data				ParityCheck
Bytes	2bytes	1byte	2bytes		N bytes				1 byte
Content		MID_ REPLY			MID- ENROLL	MR- scuess	User-id-heb	User-id-leb	checksum
	0xEFAA	0x00	0x00	0x04	0x13	0x00	0x00	0x01	0x16

This message indicates that this is a REPLY message sent by the module to the host.

The data section occupies 4 bytes, and the entry was successful and the user ID entered was 1.

3.2.2 Sending of the NOTE Message

NOTE is the information that the module actively sends to the host. The message type and corresponding data structure are determined based on the note ID.

NOTE is the information that the module actively reports to the host. For example, when the module is powered on, it will actively send a NOTE to the host, indicating that it is READY and can receive the commands from the host. After the host receives this NOTE message, it can start to send input or unlock commands.

Module is ready: EF AA 01 00 01 00 00

The main function of the NOTE message is to actively return some information to the host. Currently, the NOTE message is sent in three scenarios:

- When the module is powered on, it sends NID_READY to the host.
- During the input process, it sends facial or palm vein information to the host.
- During the unlocking process, it sends facial or palm vein information to the host.

The complete protocol of the NOTE message sent by the module to the host is shown in Table 7.

Table 7

Item	SyncWord	MsgID	Size		Data		ParityCheck
Bytes	2bytes	1byte	2bytes		N bytes		1 byte
Content	0xEFAA	MID_NON E (0x01)	high 8 bits	low 8 bits	Nid (1byte)	Data (n-bytes)	checksum

"NID" represents the execution result of the algorithm during the enrollment process, as detailed below:

Table 8

nid (* indicates carrying data)	code	Instruction
NID_READY*	0 [0x00]	Module is ready
NID_FACE_STATE*	1 [0x01]	The algorithm executed successfully and returned the facial information. s_note_data_face
NID_UNKNOWNERROR	2 [0x02]	Unknown error
NID_OTA_DONE*	3 [0x03]	OTA upgrade has been completed.
NID_EYE_STATE	4 [0x04]	The state of opening and closing eyes during the unlocking process

The data[0] of NID_READY is a byte, indicating the type of module firmware. The host can ignore it.

The data[0] of NID_OTA_DONE is a byte. 0: OTA success, 1: OTA fail.

The data[0] of NID_EYE_STATE is a byte.

0x0C: Detected an open-eye state in the closed-eye mode.

0x0D: Kept in a closed-eye state for one second continuously.

0x0E: Undetermined state.

The data carried by NID_FACE_STATE mainly stores facial information, as detailed below:

Table 9

Structure	s_note_data_face							
Content	state	left	top	right	bottom	yaw	pitch	roll
Type	int16_t	int16_t	int16_t	int16_t	int16_t	int16_t	int16_t	int16_t
Bytes	2bytes	2bytes	2bytes	2bytes	2bytes	2bytes	2bytes	2bytes

state: Indicates the current state of the face, and mainly includes the following situations:

Table 10

Face status(state)	code	Instruction
FACE_STATE_NORMAL	0 [0x00]	The face is normal
FACE_STATE_NOFACE	1 [0x01]	No face was detected.
FACE_STATE_TOOUP	2 [0x02]	The face was too close to the top edge of the picture and could not be captured
FACE_STATE_TOODOWN	3 [0x03]	The face was too close to the bottom edge of the picture and could not be captured
FACE_STATE_TOOLEFT	4 [0x04]	The face was too close to the left edge of the picture and could not be captured
FACE_STATE_TOORIGHT	5 [0x05]	The face was too close to the right edge of the picture and could not be captured
FACE_STATE_FAR	6 [0x06]	The distance of the face was too far, so it could not be captured
FACE_STATE_CLOSE	7 [0x07]	The distance of the face was too close, so it could not be captured
FACE_STATE_EYEBROW_OCCLUSION	8 [0x08]	Eyebrow concealment
FACE_STATE_EYE_OCCLUSION	9 [0x09]	Eye concealment
FACE_STATE_FACE_OCCLUSION	10 [0x0A]	Facial concealment
FACE_STATE_DIRECTION_ERROR	11 [0x0B]	The direction of the facial recognition input is incorrect.
FACE_STATE_EYE_CLOSE_STATUS_OPEN_EYE	12 [0x0C]	The open-eyed state was detected in the closed-eye mode
FACE_STATE_EYE_CLOSE_STATUS	13 [0x0D]	Closed-eye state
FACE_STATE_EYE_CLOSE_UNKNOW_STATUS	14 [0x0E]	In the closed-eye mode detection, it is impossible to determine the state of eye
FACE_STATE_HAND_NORMAL	128 [0x80]	The palm vein has been detected.
FACE_STATE_HAND_FAR	129 [0x81]	The distance of the palm vein was too far, so it could not be captured
FACE_STATE_HAND_CLOSE	130 [0x82]	The distance of the palm vein was too close, so it could not be captured

FACE_STATE_HAND_TOOUP	131 [0x83]	The palm vein was too close to the top edge of the picture and could not be captured
FACE_STATE_HAND_TOODOWN	132 [0x84]	The palm vein was too close to the bottom edge of the picture and could not be captured
FACE_STATE_HAND_TOOLEFT	133 [0x85]	The palm vein was too close to the left edge of the picture and could not be captured
FACE_STATE_HAND_TOORIGHT	134 [0x86]	The palm vein was too close to the right edge of the picture and could not be captured

The specific meanings of other members in s_note_data_face are as follows:

left: The distance from the face box to the leftmost edge of the image (a negative value indicates that the face box has exceeded the leftmost edge of the image)

top: The distance from the face box to the topmost edge of the image (a negative value indicates that the face box has exceeded the topmost edge of the image)

right: The distance from the face box to the rightmost edge of the image (a negative value indicates that the face box has exceeded the rightmost edge of the image)

bottom: The distance from the face box to the bottommost edge of the image (a negative value indicates that the face box has exceeded the bottommost edge of the image)

The rotation directions represented by yaw, pitch, and roll are shown in the figure below.

Among them, a negative yaw indicates turning the head to the left, and a positive yaw indicates turning the head to the right; a negative pitch indicates raising the head upwards, and a positive pitch indicates lowering the head downwards; a negative roll indicates tilting the head to the right, and a positive roll indicates tilting the head to the left.

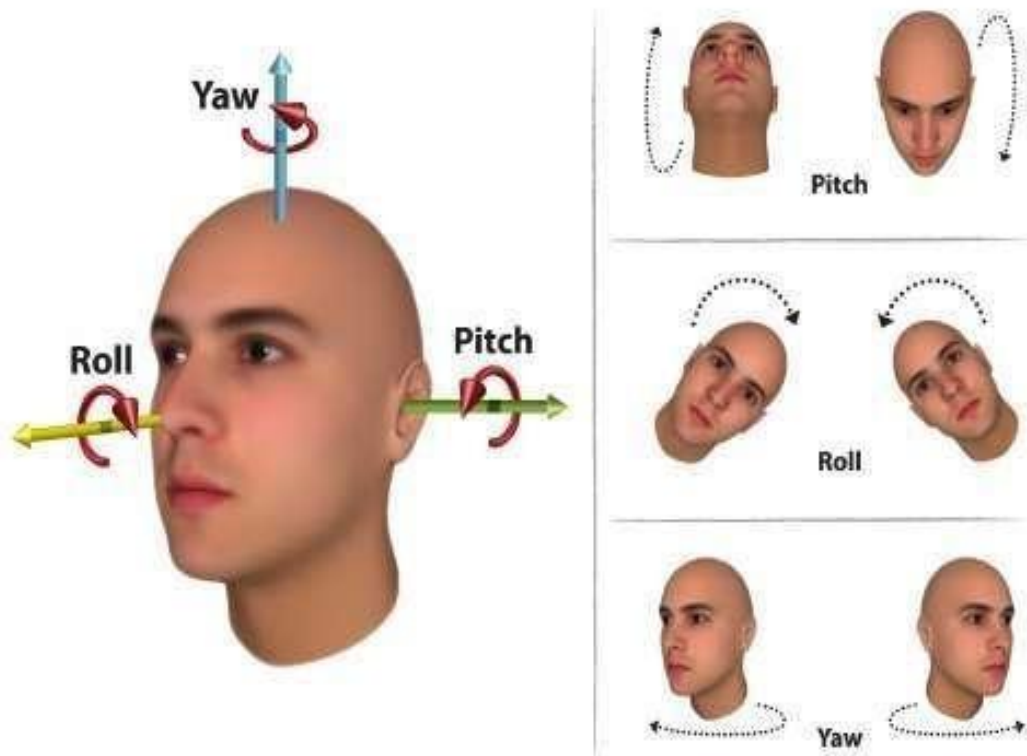


Figure 567

4. Communication Message Explanation

4.1 Command Data Format

All the instructions supported by the module are shown in Table 11.

The data received in the "Data" column is from the "data" section of Table 4.

The data sent in the "Data" column is from the "data" section of Table 5.

Table 11

MsgID/Timeout(ms)	Code	Data	
		Parameter(bytes)	Instructions
MID_RESET/200	0x10	Standby command: Stop all the current processing messages. The module enters the standby state. After the host sends this command, the module will cancel the previous commands being executed (such as inputting, unlocking, etc.) and return to the standby state.	
		Return data: None	
MID_GETSTATUS/200	0x11	Obtain module status: After the host sends this command, the module will return its current status.	
		status(1)	The status of the module includes: 0: MS_STANDBY (The module is in idle state and waiting for the host command) 1: MS_BUSY (The module is in working state) 2: MS_ERROR (The module has an error and cannot work normally) 3: MS_INVALID (The module has not been initialized) 4: MS_OTA (OTA upgrade is in progress)
		Return data: None	
MID_VERIFY	0x12	Face/ palm vein recognition unlocking	
		pd_rightaway(1)	Indicates whether to shut down immediately after unlocking; Whether to cut off power immediately after successful unlocking (yes: 1, no: 0) Deprecated now
		timeout(1)	Unlock timeout time (unit: s) [timeout] The unlock timeout time (unit: s), which can be set when sending the unlock command. The user can set the timeout time arbitrarily, with a maximum of no more than 255 seconds.
		Return data	

		user_id_heb(1)	Verified user ID The user_id is available only when the result is MR_SUCCESS
		user_id_leb(1)	
		user_name(32)	User name
		admin(1)	Whether an administrator
		unlockStatus(1)	The state of opening and closing eyes during the unlocking process 0xC8: Face verification successful 0xCC: Face verification successful, close eyes 0xFA: Palm vein verification successful
MID_ENROLL	0x13	Interactive input	
		admin(1)	Indicate that the person who made this entry is an administrator. Whether to set as an administrator (yes:1,no:0)
		user_name(32)	Enter the user's name
		face_direction(1)	The direction that the user needs to input, the specific parameters are shown in Table 12
		timeout(1)	Input timeout time (unit: s)
		Return data	
		user_id_heb(1)	The ID of the registered user During the input process, it is displayed as 0xFF. Only when the interactive input is successful will the user_id be available.
		user_id_leb(1)	
		face_direction(1)	The input status of faces from all directions, referred to Figure 10. 0xFD represents palm vein input 0x00 represents single-frame face input
MID_SNAPIMAGE	0x16	Capture pictures and save them locally	
		image_counts(1)	The number of captured images
		start_number(1)	The starting number for the captured images (1 - 20)
		Return data	
MID_GETSAVEDIMAGE	0x17	Get the size of the image to be uploaded	
		image_number(1)	Image ID to be transferred >0: Current image size (bytes) obtained by issuing the MID_SNAPIMAGE command
		Return data	
		image_size(4)	Size of the image to be uploaded, MSB
MID_UPLOADIMAGE	0x18	Upload the locally stored pictures to the host	

		upload_image_offset(4)	The offset of the image to be uploaded, MSB
		upload_image_size(4)	The size of the uploaded images (max 4K)
		Return data	
		data(n)	n: The host requires the actual size images to be uploaded.
MID_ENROLL_SINGLE	0x1D	Single-frame input (face/ palm vein input)	
		admin(1)	Whether set as an administrator(yes:1, no:0)
		user_name(32)	Enter the user's name
		face_direction(1)	Face: 0x00 Palm vein: 0xFD
		timeout(1)	Input timeout time (unit: s)
		Return data	
			The same as the response of MID_ENROLL command.
MID_DELUSER/100	0x20 (Face) 0x80 Palm vein	Delete a single registered user (face: 20H, palm vein: 80H)	
		user_id_heb(1)	The ID of the user to be deleted
		user_id_leb(1)	
		user_type(1)	User Type 0: Face 1: Palm vein
		Return data: None	
MID_DELALL	0x21	Delete all registered users	
		type(1)	Set the deletion method 0: Delete all users 1: Delete face recognition users 2: Delete palm vein recognition users 3: Delete general users 4: Delete users within a specified range,the range is [begin_user_id, end_user_id]
		begin_user_id_heb(1)	When Type = 4, the starting number of the user ID range
		begin_user_id_leb(1)	
		end_user_id_heb(1)	When Type = 4, the ending number of the user ID range
		end_user_id_leb(1)	
		Return data: None	

MID_GETUSERINFO/100	0x22 (Face)	Get the information of the user with the specified ID (Face: 22H, Palm vein: 83H)	
		user_id_heb(1)	Registered user's ID
	user_id_leb(1)		
	0x83 Palm vein	Return data	
		user_id_heb(1)	Registered user's ID
		user_id_leb(1)	
		user_name(32)	The name of the registered user
		admin(1)	Whether an administrator
MID_GET_ALL_USERID/1000	0x24	Get the number and IDs of all registered users	
		fmt(1)	Get the number and ID of all registered users and set the response format. 0: Facial user (response data format is long) 1: Facial user (response data format is short) 2: Palm vein user (response data format is short)
		Return data: fmt=0	
		user_counts_heb(1)	Number of registered users
		user_counts_leb(1)	
		users_id[MAX_USER_COUNTS*2]	All registered user IDs are stored using two consecutive bytes for each ID. The high eight bits are stored first. MAX_USER_COUNTS represents the maximum number of users.
		Return data: fmt=1,2	
		user_counts_heb(1)	Number of registered users
		user_counts_leb(1)	
		users_id[(MAX_USER_COUNTS + 7) / 8]	Registered user ID. Each bit represents a user, and MAX_USER_COUNTS represents the maximum number of facial user/ palm vein users. When the "user_id" is 0x08 0x03, the user numbers are 4, 9, and 10.
MID_ENROLL_ITG	0x26	Comprehensive input (integrated input method)	
		admin(1)	Whether set as an admin(yes:1 no:0)
		user_name(32)	Enter user name

		face_dir(1)	The direction that the user needs to enter, refer to Table 12. The enroll_type is only useful when it is 0. When registering palm vein, simply set face_dir to FACE_DIRECTION_HAND (0xFD).
		enroll_type(1)	Registration type: Interactive input or single-frame input 0: Interactive input (same as MID_ENROLL) 1: Single-frame input (same as MID_ENROLL_SINGLE)
		enable_duplicate(1)	Whether to allow duplicate entries 0: The same person cannot be entered, but the username can be repeated. 1: The same person can be entered repeatedly, and the username can be repeated. 2: The same person can be entered, but the username cannot be repeated.
		timeout(1)	Input timeout time (unit: s)
		reserved(3)	Keep it, just input 0.
		Return data	
		user_id_heb(1)	Registered user ID. The user_id exists only when the result is MR_SUCCESS
		user_id_leb(1)	
		face_direction(1)	Interactive input: Input status of faces in all directions Single-frame input: Face: 0x00 Palm vein: 0xFD
MID_GET_VERSION/1000	0x30	Get software version information	
		Receive: None	
		Return data	
		version_info(32)	Version information
MID_INIT_ENCRYPTION/1000	0x50	Set the encrypted random number	
		seed(4)	The host sends a random number

		mode(1)	Encryption mode 0: No encryption 1: AES encryption 2: SINGLE encryption
		Return data	
		device_id(20)	Device id information
MID_SET_RELEASE_ENC_KEY	0x52	Set encryption rules	
		enc_key_number(16)	Encryption Sequence
		Return data: None	
			Except for the initial setup, from the second setup onwards, the process is regarded as a failure.
MID_SET_DEBUG_ENC_KEY	0x53	Set the debug encryption key sequence, which will be lost if power is off	
		enc_key_number(16)	Encryption Sequence
		Return data: None	
MID_LED CONTROL	0x90	LED Indicator light	
		LED color(1)	Green: 0 Red: 1 White: 2
		on /off(1)	Turn on LED: 0 Turn off LED: 1
		Return data: None	
MID_GET_UID	0x93	Receive: None	
		Return data	
			Same as the response to the MID_INIT_ENCRYPTION command
MID_CAPTURE_PIC_TYPE	0x9A	Set the upload image parameters	
		pic_width(2)	Image width. Currently, only the supported aspect ratios are 640x480, 480x320, and 320x240.
		pic_height(2)	Image height. Currently, only the supported aspect ratios are 640x480, 480x320, and 320x240.
		capture_type(1)	Image type IR:0x00 RGB:0x01
		iompess_ratio(1)	Image compression ratio Compression quality: 30%-80% corresponds to settings of 30-80

		is cutout_face(1)	Whether to crop the face of the picture, not supported yet
		Return data: None	
MID_SET_THRESHOLD_LEVEL	0xD4	Set the security level	
		verify_threshold_level(1)	Data range: 0 - 4, Minimum: 0, Maximum: 4, Standard: 2
		liveness_threshold_level(1)	Data range: 0 - 4, Minimum: 0, Maximum: 4, Standard: 2
		Return data: None	
MID_DEMOMODE/100	0xFE	Demo Mode	
		After the host sends this command, the module enters demo mode. In this mode, feature comparison is not performed during the module recognition process, that is, all faces/palm veins can be unlocked, but liveness detection is still performed.	
		enable(1)	1: enable 0: disable
		Return data: None	
MID_DOWNLOADUSERCHAR	0xA0	Download user templates to cache	
		char_type(1)	Template Type: Face (0) Palm Vein (1) Face Template Size: 4128 Bytes Palm Vein Template Size: 12288 Bytes
		download char_offset(4)	Offset of the template to be downloaded, MSB
		download char_size(4)	Download template data size (range 1-4K)
		char_data(n)	n: Download template data download_char_size
		Return data: None	
MID_SAVEDOWNLOADUSERCHAR	0xA1	Save the user template to Flash	
		char_type(1)	Template Type: Face (0) Palm Vein (1)
		char save_local(2)	Template storage location Face range (1-500) Palm vein range (501-800)
		Return data: None	
MID_UPLOADUSERCHAR	0xA2	Upload user characteristic data	

		char_type(1)	Template Type: Face (0) Palm Vein (1) Face Template Size: 4128 Bytes Palm Vein Template Size: 12288 Bytes
		char_number(2)	The template number to be uploaded Face range (1-500) Palm vein range (501-800)
		upload_char_offset(4)	Offset of the template to be uploaded, MSB
		upload_char_size(4)	Upload template data size (range 1 - 4K)
		Return data	
		char_data(n)	n:The host requires the upload of template data-upload_char_size

- Since the face and palm vein template data is large, it needs to be divided into multiple transmissions. The maximum single transmission data of this module is 4K. Users can freely adjust the size according to the receiving capacity of their own devices.

Offset MSB: The starting address of the next template data transmission after the previous template data is cut and transmitted.

- Picture Capture (0x16): The host issues this command and along with the quantity of pictures to be captured and the starting number for the capture, the module will immediately start capturing, names them sequentially based on the starting image number, and stores them locally in jpg format. This function supports a maximum of 20 images stored locally, numbered 1-20. When the number sent by the host is the same as the local picture number, the original picture will be overwritten.
- Upload image size & Upload image to host(0x17 & 0x18): The captured image will be stored in the local path of the module. When the host obtains the captured image, it first calls the MID_GETSAVEDIMAGE command to obtain the size of the image to be uploaded, and then calls MID_UPLOADIMAGE multiple times according to the requirements to obtain the image. The data *upload_image_offset[4]* carried in the MID_UPLOADIMAGE instruction indicates the offset of the uploaded image.(for example, when the offset is 0, the transmission will start from the beginning of the image), and *upload_image_size[4]* represents the size of the uploaded image, with a maximum support of 4000 bytes.
- Upload Image Parameters (0x9A): The main function is to modify the parameters of the captured images. The parameters that can be modified include resolution, image type (whether it is an IR image or an RGB image), image compression ratio, and whether to crop the face, etc. After the command is sent, the next image captured by the module will be saved in the new format specified in the command and then sent back to the host.

4.2 Face Orientation Parameters

When using the MID_ENROLL_SINGLE command, the up, down, left, and right parameters are invalid.

When the parameter is set to FACE_DIRECTION_HAND, palm vein registration is performed.

When the parameter is set to FACE_DIRECTION_PICTURE, the face registration is performed through the direction parameter.

Table 12

st_face_dir (Face direction definition)	code	Instruction
FACE_DIRECTION_UP	0x10	Recording a face facing up
FACE_DIRECTION_DOWN	0x08	Recording a face facing down
FACE_DIRECTION_LEFT	0x04	Recording a face facing left
FACE_DIRECTION_RIGHT	0x02	Recording a face facing right
FACE_DIRECTION_MIDDLE	0x01	Recording a face facing forward
FACE_DIRECTION_UNDEFINE	0x00	Undefined, default is forward
FACE_DIRECTION_FACE_ONLY	0x40	Records forward-facing faces, only detects face registration methods (reserved)
FACE_DIRECTION_HAND	0xFD	Records palm vein

The "face_direction" in the responses of the MID_ENROLL and MID_ENROLL_ITG commands indicates the face enrollment status. The lower 5 bits of the data represent the face direction (up, down, left, right, and forward) from high to low, as shown in Figure 10 below. 1 indicates that the direction has been registered, and 0 indicates that the direction has not been registered.

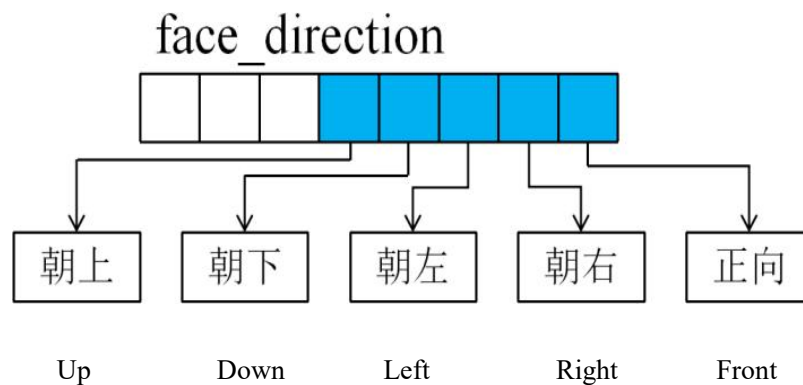


Figure 10

The angles for face recognition in all directions are as follows:

Table 13

Input direction	Deflection angle
Front	Facing the camera directly, no deflection
Up	Upward deflection: 5-55 degrees
Down	Downward deflection: 5-55 degrees
Right	Rightward deflection: 8-60 degrees
Left	Leftward deflection: 8-60 degrees

Among them, the upward and downward deflections are shown in Figure 5.



Figure 5

Left and right deflection are shown in Figure 6.

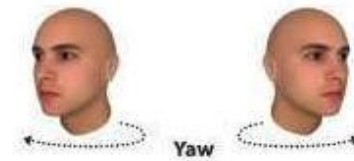


Figure 6

Please avoid turning your head as shown in Figure 7.



Figure 7

4.3 Message Instructions Format

4.3.1 Standby Instruction MID_RESET

- Description: After the host sends this command, the module will terminate the command being executed (such as input, unlock, etc.) and return to the standby state.
- Input Parameter: None
- Return Parameter: None
- Instruction code: 10H
- Command (or instruction) package format:

SyncWord	MsgID	Size		Data	ParityCheck
		H	L		
2bytes	1byte	1byte	1byte	0byte	1 byte
EFAA	10H	0000H		None	10H

- Instruction package example: EF AA 10 00 00 10
- Acknowledge package format:

SyncWord	MsgID	Size		Data			ParityCheck
		H	L	mid	result	data	
2bytes	1byte	1byte	1byte	1byte	1byte	0byte	1 byte
EFAA	00H	0002H		10H	xxH	None	xxH

- Acknowledge package example: EF AA 00 00 02 10 00 12

Note: result=00H indicates successful execution; others indicate failure.

4.3.2 Get Module Status MID_GETSTATUS

- Description: After the host sends this command, the module returns its current status.
- Input Parameter: None
- Return Parameter: status
- Instruction code: 11H
- Command (or instruction) package format:

SyncWord	MsgID	Size		Data	ParityCheck
		H	L		
2bytes	1byte	1byte	1byte	0 bytes	1 byte
EFAA	11H	0000H		None	11H

- Instruction package example: EF AA 11 00 00 11
- Acknowledge package format:

Sync Word	Msg ID	Size		Data			Parity Check
		H	L	mid	result	data	
						status	
2bytes	1byte	1byte	1byte	1byte	1byte	1byte	1byte
EFAA	00H	0003H		11H	xxH	xxH	xxH

- Acknowledge package example: EF AA 00 0003 11 00 **00** 12

Note:

status = 00H, the MS_STANDBY module is in idle state and is waiting for the host command;

status = 01H, the MS_BUSY module is in operation state;

status = 02H, the MS_ERROR module has an error and cannot work normally;

status = 03H, the MS_INVALID module has not been initialized.

4.3.3 Face/Palm Vein Recognition MID_VERIFY

- Description: The host sends this command to the module, and the module starts to recognize the face/palm vein for comparison.
- Input Parameter: pd_rightaway timeout
- Return Parameter: user_id, user_name ,admin , unlockStatus
- Instruction code: 12H
- Command (or instruction) package format:

SyncWord	MsgID	Size		Data		ParityCheck
		H	L	pd_rightaway	timeout	
2bytes	1byte	1byte	1byte	1byte	1byte	1byte
EFAA	12H	0002H		00	xxH	xxH

- Identify command parameter description:

1) pd_rightaway indicates whether to continue operating after unlocking. **Default setting: 00**

0: Power off immediately after unlocking. 1: No power off.

2) timeout represents the unlock timeout (Unit:s) The user can set the timeout, up to a maximum of 255 seconds. The timeout for the host to wait for the module's response packet should be greater than this parameter setting.

Command example: EF AA 12 00 02 00 0A 1A

- Acknowledge package format:

During the unlocking process, the module responds to two messages: NOTE and REPLY. During a recognition process, the module may respond with multiple NOTE messages (MSGID=01H) to indicate the face status, position, and posture of the current image frame, but ultimately only one REPLY message (MSGID=00H) represents the final result of the recognition.

- Response NOTE message format:

Sync Word	Msg ID	Size		Data		Parity Check
		H	L	nid	Data	
2bytes	1byte	1byte	1byte	1byte	16bytes	1byte
EFAA	01H	0011H		xxH	xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx	xxH

Note: nid=01H indicates NID_FACE_STATE, indicating that the data returned by the module is the face status of the current frame, as shown in Table 8.

Data definitions are shown in the table below:

Structure	s_note_data_face							
Content	state	left	top	right	bottom	yaw	pitch	roll
Type	int16_t	int16_t	int16_t	int16_t	int16_t	int16_t	int16_t	int16_t
Byte	2bytes	2bytes	2bytes	2bytes	2bytes	2bytes	2bytes	2bytes

state represents the current face state, with the low byte first and the high byte last. See Table 10 for details. l, t, r, and b represent the face's position in the image, representing left, top, right, and down, respectively. They are two bytes, with the low byte first and the high byte last.

y, p, and R represent the posture of the face, which are yaw (turn the head left and right) (Figure 5), pitch (raise the head up and down) (Figure 6), and roll (tilt the head left and right). They are two bytes, with the low byte first and the high byte last.

- Response NOTE message example:

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

The module returns the face status detected in the current image frame as **0700**, indicating that the face is too close.

- Identify instructions REPLY message format:

Sync Word	Msg ID	Size		Data						Parity Check
		H	L	mid	result	data				
						user_id	user_name	admin	unlock Stauts	
2bytes	1byte	1byte	1byte	1byte	1byte	2bytes	32bytes	1byte	1byte	1byte
EFAA	00H	0026H		12H	xxH	xxxx xxxx	xx.. xx	xxH	xxH	xxH

- Acknowledge package parameter description:

- 1) user_id: The ID of the successfully identified user;
- 2) user_name: The username of the successfully identified user;
- 3) admin: Whether the identified user is an administrator user;
- 4) unlockStatus: Feedback on successful identification(C8: Face recognition successful, CC: Face recognition successful with eyes closed, FA: Palm vein recognition successful)

- The return value of the identification result confirmation code is shown in Table 6

- Identify command return example:

[illegible]

identification successful 00, the identified user ID is 0001, the user name is 0.

This user is a regular user **00**, and the recognition result **C8** is a face user.

EF AA 00 0002 12 **0D** 1D Command explanation: Identification failed. The reason for failure is timeout
(0DH)

✧ Input instructions

Function description: There are three input instructions in the input instruction set.

- 1) Single frame recording MID_ENROLL_SINGLE (0x1D): This command only records one front face. The same face can be recorded multiple times to generate different face IDs.
- 2) Interactive Input MID_ENROLL (0x13): This command requires enrolling faces in five directions (up, down, left, right, and center) to complete an enroll and generate a face ID. The same face can be enrolled multiple times.
- 3) Comprehensive Input MID_ENROLL_ITG (0x26): This command combines single-frame and interactive input modes. It controls the enrolling mode and whether the same face can be enrolled repeatedly by passing parameters.

Sync Word	Msg ID	Size		Data				Parity Check
		H	L	mid	result	data		
						user_id	direction	
2bytes	1byte	1byte	1byte	1byte	1byte	2bytes	1byte	1byte
EFAA	00H	0005H		1DH	xxH	xxxxxxH	00H	xxH

- Direction confirmation result: 00: face, FD: palm vein
- The return value of the input result confirmation code is shown in Table 6

Note: User ID is automatically assigned by the system

- Single frame recording response packet command example:

EFAA 00 0005 1D 00 0003 00 1B, identification successful 00,

the user ID of the successful identification is 0003, the face is 00

EFAA 00 0002 1D 0D 12, identification failed. The reason for failure is recording timeout (0DH)

4.3.5 Interactive Input MID_ENROLL

- Description: Interactive input, complete the input in 5 directions (up, middle, down, left, and right), and finally generate a face ID
- Input Parameter: admin_name_direction_timeout
- Return Parameter: user_id_direction
- Instruction code: 13H
- Command (or instruction) package format:

SyncWord	MsgID	Size		Data				Parity Check
		H	L	admin	name	direction	timeout	
2bytes	1byte	1byte	1byte	1byte	32bytes	1byte	1byte	1byte
EFAA	13H	0023H		00H	xxxx...	xxH	xxH	xxH

- Parameter description:

1) admin: Sets the user being entered as an administrator;

2) user_name: the username of enter user

3) face_direction: The direction of the currently entered face;

face_direction = 10H (FACE_DIRECTION_UP) indicates that the face currently being recorded is facing upward.

face_direction = 08H (FACE_DIRECTION_DOWN) indicates that the face currently being

4.3.6 Capture Pictures and Store Locally MID_SNAPIMAGE

- Description: Capture user pictures and save them locally
- Input Parameter: image_counts, start_number
- Return Parameter: None
- Instruction code: 16H
- Command (or instruction) package format:

SyncWord	MsgID	Size		Data		ParityCheck
		H	L	image_counts	start_number	
2bytes	1byte	1byte	1byte	1byte	1byte	1byte
EFAA	16H	0002H		xxH	xxH	xxH

Image_counts: Number of captured images

start_number: Starting number of captured images (1-20)

4.3.7 Get Uploaded Images MID_GETSAVEDIMAGE

- Description: Get uploaded user pictures
- Input Parameter: image_number
- Return Parameter: image_size
- Instruction code: 17H
- Command (or instruction) package format:

SyncWord	MsgID	Size		Data	ParityCheck
		H	L	image_number	
2bytes	1byte	1byte	1byte	1byte	1byte
EFAA	17H	0001H		xxH	xxH

image_number: The number of the image to be transmitted > 0: Send the MID_SNAPIMAGE command to obtain the current image size (bytes).

4.3.8 Upload Local Storage Images To the Host MID_UPLOADIMAGE

- Description: Upload user pictures to the host
- Input Parameter: upload_image_offset, upload_image_size
- Return Parameter: data
- Instruction code: 18H
- Command (or instruction) package format:

SyncWord	MsgID	Size		Data		ParityCheck
		H	L	upload_image_offset	upload_image_size	
2bytes	1byte	1byte	1byte	4bytes	4bytes	1byte
EFAA	17H	0008H		xxH	xxH	xxH

image_offset: The offset of the image to be uploaded

image_size: The size of the image to be uploaded (maximum 4K)

4.3.9 Deleting a Single User MID_DELUSER

- Description: By passing in the user ID, delete a single user with the specified ID
- Input Parameter: user_id, user_type
- Return Parameter: None
- Instruction code: 20H (face), 80H (palm vein)
- Command (or instruction) package format:

SyncWord	MsgID	Size		Data		ParityCheck
		H	L	user_id	user_type	
2bytes	1byte	1byte	1byte	2bytes	1byte	1byte
EFAA	xxH	0003H		xxH	xxH	xxH

user_type: 0: face, 1: palm vein

- Example of deleting a single user instruction package:

Deleting the face recognition user with ID 0006: EFAA **20** 0003 0006 **00** 25

Deleting the palm vein recognition user with ID 01F5: EFAA **80** 0003 01F5 **01** 76

- Acknowledge package format:

SyncWord	MsgID	Size		Data			ParityCheck
		H	L	mid	result	data	
2bytes	1byte	1byte	1byte	1byte	1byte	0byte	1byte
EFAA	00H	0002H		xxH	xxH	None	xxH

Note: The recommended timeout for the host waiting for a response is 5 seconds.

mid=20 Face

mid=80 Palm vein

result=00H (MR_SUCCESS) Delete successful;

result=05H (MR_FAILED4_UNKNOWNREASON) Unknown error;

result=08H (MR_FAILED4_UNKNOWNUSER) The deleted ID does not exist;

4.3.10 Delete All Users MID_DELALL

- Description: Delete all registered users
- Input Parameter: type ,begin user_id , end user_id
- Return Parameter: None
- Instruction code: 21H
- Command (or instruction) package format:

SyncWord	MsgID	Size		Data			ParityCheck
		H	L	type	begin user_id	end user_id	
2bytes	1byte	1byte	1byte	1byte	2bytes	2bytes	1byte
EFAA	21H	0005H		xxH	xxxxH	xxxxH	xxH

- Type deletion options: 0: Delete all users
 - 1: Delete face recognition users
 - 2: Delete palm vein recognition users
 - 3: Delete general users
 - 4: Delete users within a specified range (when type = 4, begin user_id ~ end user_id)
- Example of deleting a face user command: EFAA 21 00 05 **01** 00 00 00 00 25
- Example of deleting the face user range ID 100-200: EFAA 21 00 05 **04** 00 64 00 C8 8C
- Acknowledge package format:

SyncWord	MsgID	Size		Data			ParityCheck
		H	L	mid	result	data	
2bytes	1byte	1byte	1byte	1byte	1byte	0byte	1byte
EFAA	00H	0002H		21H	xxH	无	xxH

- Delete all user response examples:
- EFAA 00 0002 21 00 23, all face users deleted successfully.
- EFAA 00 0002 21 05 26, a brand new module with no previous face data entry, no face template file generated within the module, resulting in deletion failure. Alternatively, some other unknown error may have caused deletion failure.

Note: result = 00H (MR_SUCCESS) - Deletion successful;

result = 05H (MR_FAILED4_UNKNOWNREASON) - Unknown error;

4.3.11 Get User Information For a Specified ID MID_GETUSERINFO

- Description: The host specifies the registered user to be returned through the user id. After receiving the command, the module will return the information of the specified user.
- Input Parameter: user_id
- Return Parameter: None
- Instruction code: 22H (face), 83H (palm vein)
- Command (or instruction) package format:

SyncWord	MsgID	Size		Data	ParityC heck
		H	L	user_id	
2bytes	1byte	1byte	1byte	2bytes	1byte
EFAA	xxH	0002H		xxH	xxH

Example of a command packet to obtain the second face image: EF AA **22** 00 02 00 02 22

Example of a command packet to obtain the 501st palm vein image: EF AA **83** 00 02 01 F5 75

- Acknowledge package format:

SyncWord	MsgID	Size		Data					Parity Check
		H	L	mid	result	data			
						user_id	user_name	admin	
2bytes	1byte	1byte	1byte	1byte	1byte	2bytes	32bytes	1byte	1byte
EFAA	00H	0025H		xxH	xxH	xxH	xx..xxH	xxH	xxH

- Example of a response packet: EF AA 00 00 05 24 00 **00 06** 3F 18

Command Explanation: Successfully obtained the number of 6 faces registered by the module

- Acknowledge package long format:

SyncWord	MsgID	Size		Data				Parity Check
		H	L	mid	result	data		
						user_counts	MAX_user _counts*2	
2bytes	1byte	1byte	1byte	1byte	1byte	2byte	xx..bytes	1byte
EFAA	00H	xx xxH		24H	xxH	xxH	xx....xx	xxH

- Example of a response packet: EF AA 00 00 10 24 00 **00 06** 00 01 00 02 00 03 00 04 00 05 00 06 35

Command Explanation: Successfully obtained the number of 6 faces registered by the module, the 6 face IDs are 0001 0002 0003 0004 0005 0006 respectively.

The module can register up to 500 faces (ID1-500) and 300 palm veins (ID501-800). Please note the buffer size allocated by the host to save the response packet data.

4.3.13 Comprehensive Input MID_ENROLL_ITG

- Description: This command integrates two different input methods: the single-frame input command MID_ENROLL_SINGLE and the interactive input command MID_ENROLL. The method can be specified in the parameter structure, and whether the same face can be encoded repeatedly is also controlled.
- Input Parameter: admin,name,face_dir,enroll_type,enable,timeout,reserved
- Return Parameter: urse_id ,face_directiong
- Instruction code: 26H
- Command (or instruction) package format:

Sync Word	Msg ID	Size		Data							Parity Check
		H	L	admin	name	face_direction	enroll_type	enable_duplicate	timeout	reserved	
2bytes	1byte	1byte	1byte	1byte	32bytes	1byte	1byte	1byte	1byte	3bytes	1byte
EFAA	26H	0028H		xxH	x..xH	xxH	xxH	xxH	xxH	xx.. xxH	xxH

- Parameter description:

- 1) admin: Controls whether the entered user is an administrator user;
- 2) user_name: The username of the entered user;
- 3) face_direction: When using single-frame input, use 00 for face and FD for palm vein. For interactive

4.3.15 Set the Encryption Random Number MID_INIT_ENCRYPTION

- Description: This command, along with the `MID_SET_RELEASE_KEY` command, generates an encryption key and uses the specified encryption mode for encrypted communication. After receiving this command, the module enters encryption mode, encrypts the module's SN, and sends it back to the host. The host unit receives the data and decrypts it using the same key and decryption method to obtain the plaintext data.
- Input Parameter: seed , mode
- Return Parameter: device_id
- Instruction code: 50H
- Command (or instruction) package format:

Sync Word	Msg ID	Size		Data		Parity Check
		H	L	seed	mode	
2bytes	1byte	1byte	1byte	4bytes	1byte	1byte
EFAA	50H	0005H		xxxx xxxx	xx	xxH

- Parameter description:
 - 1) seed: Random number used to generate encryption keys.
 - 2) mode: Encryption mode. No encryption: 00H, AES encryption: 01H, SINGLE encryption: 02H
- Acknowledge package format:

Sync Word	Msg ID	Size		Data	Parity Check
		H	L	Device_id	
2bytes	1byte	1byte	1byte	20bytes	1byte
EFAA	50H	0014H		xxxx xxxx	xxH

4.3.16 Set the Encryption Rules MID_SET_RELEASE_ENC_KEY

- Description: Encrypted communication is required between the host and the module. This command and the `MID_INIT_ENCRYPTION` command work together to generate the encryption key. The module stores the data transmitted via this command and ensures that it is not lost during power failures.
- Input Parameter: `enc_key_number_data`
- Return Parameter: None
- Instruction code: 52H
- Command (or instruction) package format:

SyncWord	MsgID	Size		Data	ParityCheck
		H	L	enc_key_number	
2bytes	1byte	1byte	1byte	16bytes	1byte
EFAA	52H	0010H		xx...xx	xxH

- Example of command packet: EFAA 52 0010 **000102030405060708090A0B0C0D0E0F** 42
- Acknowledge package format:

SyncWord	MsgID	Size		Data			ParityCheck
		H	L	mid	result	data	
2bytes	1byte	1byte	1byte	1byte	1byte	0byte	1byte
EFAA	00H	0002H		52H	xxH	None	xxH

Note: Result = 00H (MR_SUCCESS) indicates the key was set successfully;

Result = 01H (MR_REJECTED) indicates the key has already been set.

Result = 06H (MR_FAILED4_INVALIDPARAM) indicates the parameter input is incorrect.

4.3.17 Set the Security Level MID_SET_THRESHOLD_LEVEL

- Description: Modify the algorithm security level. Different manufacturers and users have different understandings of security requirements, so the security level interface is opened to set the security level of liveness and verify. There are four levels that can be set: low, lower, normal, higher, and high.
- Input Parameter: verify_threshold , liveness_threshold
- Return Parameter: None
- Instruction code: D4H
- Command (or instruction) package format:

SyncWord	MsgID	Size		Data		ParityCheck
		H	L	verify_threshold_level	liveness_threshold_level	
2bytes	1byte	1byte	1byte	1byte	1byte	1byte
EFAA	D4H	0002H		xxH	xxH	xxH

- Example of command packet: EFAA D4 00 02 04 03 D1
Verify_threshold_level: 0-4, Standard 2
Liveness_threshold_level: 0-4, Standard 2
Level range: 0-4, 0 is the lowest, 4 is the highest.
- Acknowledge package format:

SyncWord	MsgID	Size		Data			ParityCheck
		H	L	mid	result	data	
2bytes	1byte	1byte	1byte	1byte	1byte	1byte	1byte
EFAA	00H	0002H		D4H	xxH	None	xxH

- Example of acknowledge packet: EFAA 00 00 02 D4 00 D6

Declaration:It is recommended to use the default settings.

A security level that is too high can result in a lower pass rate.

A security level that is too low can lead to false positives.Please use this setting with caution.

4.3.18 Demo Mode MID_DEMOMODE

- Description: Demo mode is for demonstration purposes only and is less secure. Anyone can unlock it. The data will not be saved when the power is off, and the module will return to normal mode after a restart.
- Input Parameter: enable
- Return Parameter: None
- Instruction code: FEH
- Command (or instruction) package format:

SyncWord	MsgID	Size		Data	ParityCheck
		H	L	enable	
2bytes	1byte	1byte	1byte	1byte	1byte
EFAA	FEH	0001H		01H	FEH

- Enable Mode: 1: Enter the demo, 0: Exit the demo
- Command example:

Enter demo mode: EFAA FE 00 01 01 FE

Exit demo mode: EFAA FE 00 01 00 FF

- Acknowledge package format:

SyncWord	MsgID	Size		Data			ParityCheck
		H	L	mid	result	data	
2bytes	1byte	1byte	1byte	1byte	1byte	0byte	1byte
EFAA	00H	0002H		FEH	xxH	无	xxH

- Example of acknowledge packet: EFAA 00 00 02 FE 00 FC

4.3.19 LED Setting MID_LED CONTROL

- Description: LED light control can be used for powering on, and for indicating status.
- Input Parameter: color, on/off
- Return Parameter: None
- Instruction code: 90
- Command (or instruction) package format:

SyncWord	MsgID	Size		Data		ParityCheck
		H	L	Color	On/off	
2bytes	1byte	1byte	1byte	1byte	1byte	1byte
EFAA	90H	0002H		xxH	xxH	xxH

1) Color Description: Green: 0, Red: 1, White: 2

2) Switch Description: On: 0, Off: 1

Green light example: Turn on the green light EF AA 90 00 02 00 00 92

Turn off the green light EF AA 90 00 02 00 01 93

- Acknowledge package format:

SyncWord	MsgID	Size		Data		ParityCheck
		H	L	mid	result	
2bytes	1byte	1byte	1byte	1byte	1byte	1byte
EFAA	00H	0002H		90H	xxH	xxH

- Example of acknowledge packet: EFAA 00 00 02 90 00 92

4.3.20 Set the Upload Image Parameters MID_CAPTURE_PIC_TYPE

- Description: Set the Upload Image Parameters
- Input Parameter:
- Return Parameter: None
- Instruction code: 9A
- Command (or instruction) package format:

SyncWord	MsgID	Size		Data					ParityCheck
		H	L	pic_width	pic_height	capture_type	iompress_ratio	is_cutout_face	
2bytes	1byte	1byte	1byte	2byte	2bytes	1byte	1byte	1byte	1byte
EFAA	9AH	0007H		xxH	xxH	xxH	xxH	xxH	xxH

pic_width: Image width. Currently, only supported resolutions are 640x480, 480x320, and 320x240.

pic_height: Image height. Currently, only supported resolutions are 640x480, 480x320, and 320x240.

capture_type: Image type: IR: 0, RGB: 1.

iompress_ratio: Image compression ratio/quality. 30%-80% (corresponding settings: 30-80).

is_cutout_face: Image cropping is not currently supported.

4.3.21 Upload User Template MID_UPLOADUSERCHAR

- Description: Upload face/palm vein template data
- Input Parameter: char_type,number,upload_char_offset,char_size
- Return Parameter: char_data
- Instruction code: A2
- Command (or instruction) package format:

SyncWord	MsgID	Size		Data				ParityCheck
		H	L	char_type	char_number	upload_char_offset	upload_char_size	
2bytes	1byte	1byte	1byte	1byte	2bytes	4bytes	4bytes	1byte
EFAA	A2H	0011H		xxH	xxH	xxH	xxH	xxH

- Parameter Description:
 - 1) char_type: 0: Face (template size 4128 bytes), 1: Palm vein (template size 12288 bytes)
 - 2) char_number: The uploaded template number (face range 1-500, palm vein range 501-800)
 - 3) char_offset: The uploaded template offset (MSB)
 - 4) char_size: The uploaded template data size (1-4KB)

4.3.22 Download User Template MID_DOWNLOADUSERCHAR

- Description: Download face/palm vein template data
- Input Parameter: char_type,number,upload_char_offset,char_size
- Return Parameter: None
- Instruction code: A0
- Command (or instruction) package format:

SyncWord	MsgID	Size		Data				ParityCheck
		H	L	char_type	download_char_offset	Download_char_size	char_data	
2bytes	1byte	1byte	1byte	1byte	4bytes	4bytes	N bytes	1byte
EFAA	A0H	xxH		xxH	xxH	xxH	xxH	xxH

- Parameter Description:

1) char_type: 0: Face (template size 4128 bytes), 1: Palm vein (template size 12288 bytes)

2) char_offset: Download template offset, MSB

3) char_size: Download template data size (1-4KB)

4) char_data: Download template data

4.3.23 Save User Template MID_SAVEDOWNLOADUSERCHAR

- Description: Save user template to Flash

- Input Parameter: char_type, char_save_local

- Return Parameter: None

- Instruction code: A1

- Command (or instruction) package format:

SyncWord	MsgID	Size		Data		ParityCheck
		H	L	char_type	char_save_local	
2bytes	1byte	1byte	1byte	1byte	2 bytes	1byte
EFAA	A1H	0003H		xxH	xxH	xxH

- Parameter Description:

1) char_type: 0: Face, 1: Palm vein

2) char_save_local: Template storage location: Face range 1-500, Palm vein range 501-800

5. Application Process

5.1 Host Receiving Message Process

The host receives the message sent by the module and can follow the process shown in Figure 1.

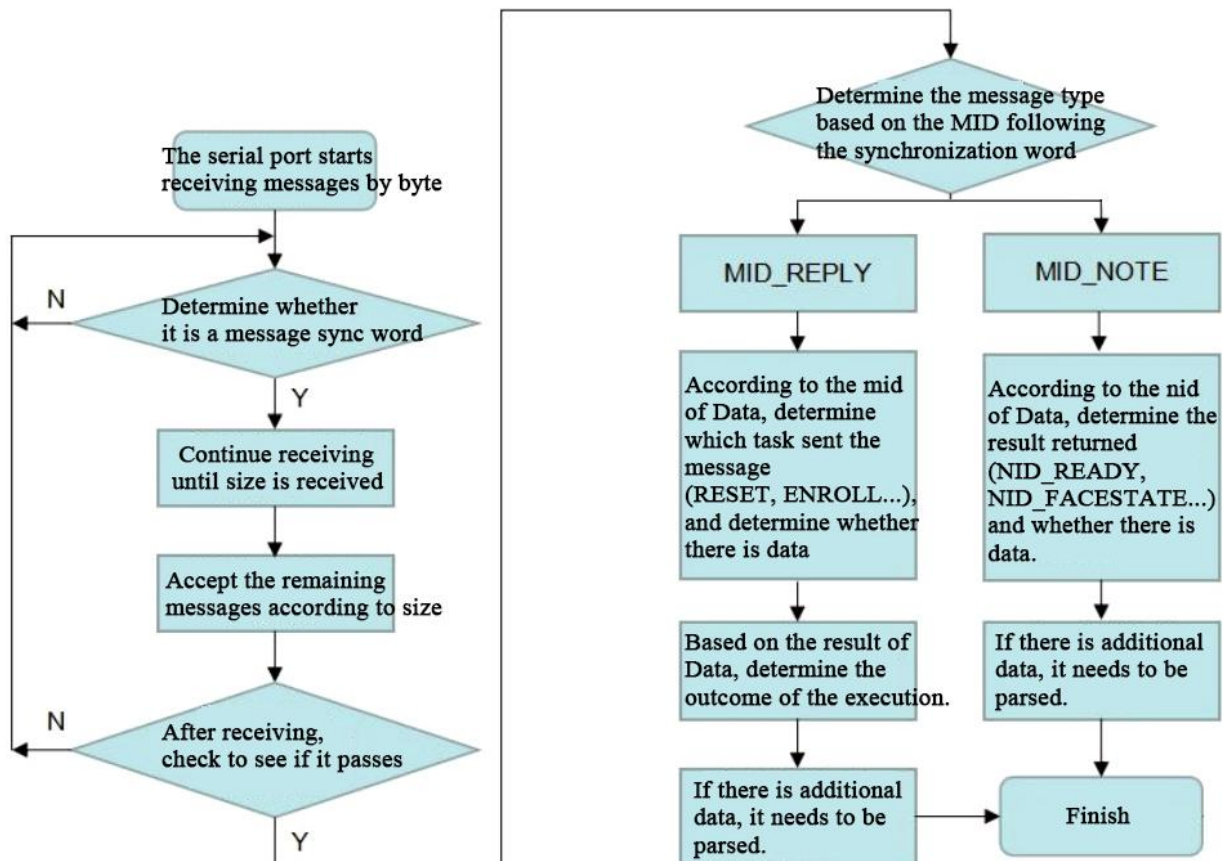


Figure 1

5.2 General Message Processing Flow

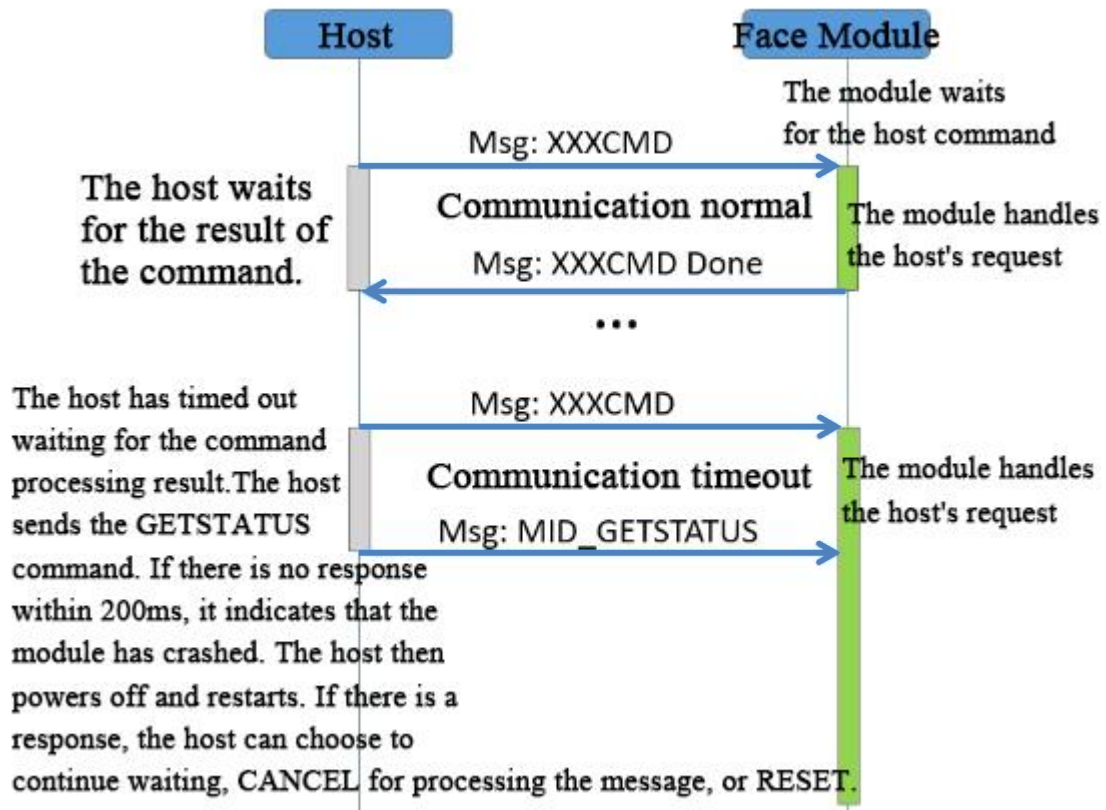


Figure 2

5.3 Power On and Off Process

After the module is powered on, it takes approximately 250ms to complete the initialization process. During this period, the module cannot respond to commands from the host computer. Once the initialization is completed, the module will immediately send a status notification package to the host computer, indicating that the module is now ready to work normally and can receive commands from the host computer.

The power-on and power-off of the host active control module are shown in Figure 3. When the module is powered on, it enters the startup process. When it transitions to the standby state, it will notify the host. The instruction is: EF AA 01 00 01 00 00. The host then starts to send command messages, and the module processes and returns the results.

If there is no response from the host after it issues an instruction, the host can directly cut off power to the module.

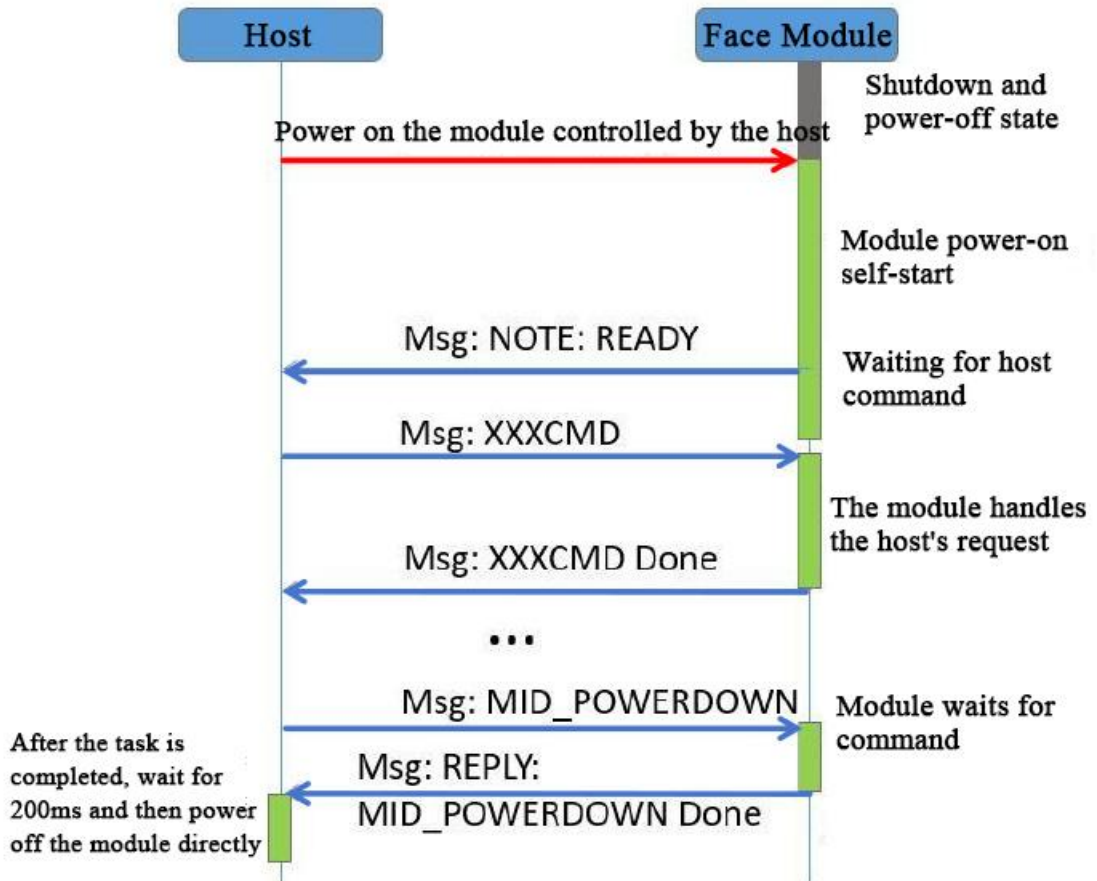


Figure 3

5.4 Input Process

5.4.1 Face Interactive Input

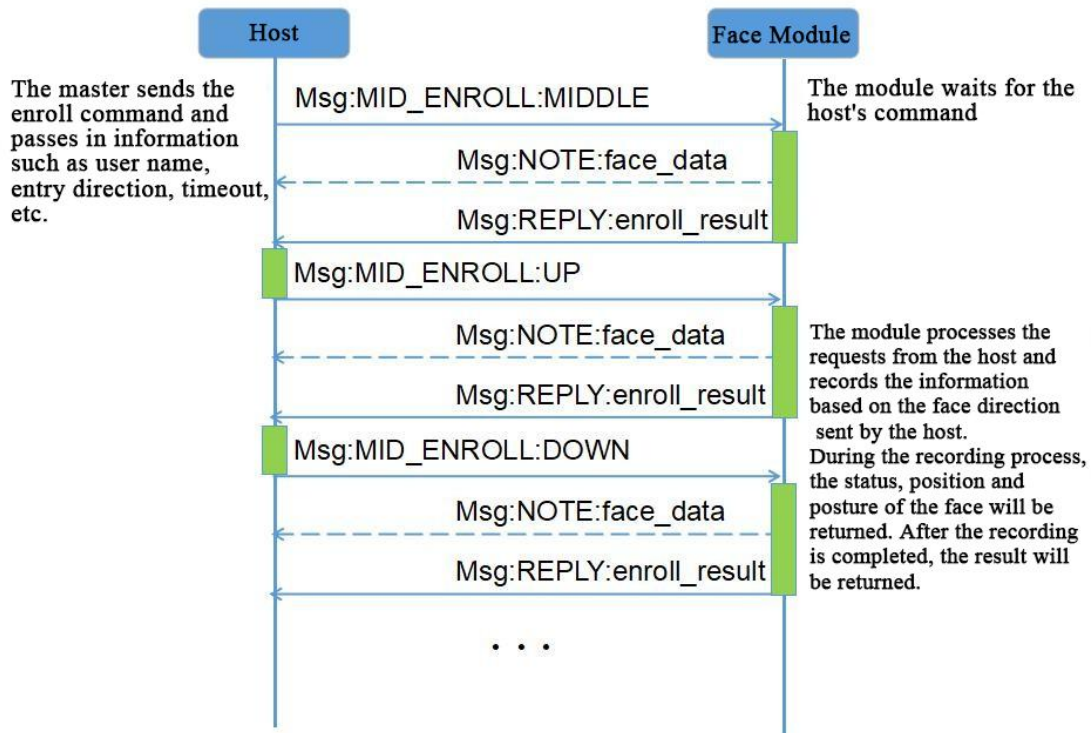


Figure 4

During the interactive input process, the face module does not restrict the order of input directions. Users can arbitrarily combine the order of input directions. Figure 4 uses one of the orders as an example to illustrate.

When the main control issues a command to record a face in a certain direction, the module starts working and returns real-time information such as the face's status, position, and posture (returned in the form of NOTE messages). The user can use this information to remind the registrant to adjust the posture and position. When the registration is successful, the module will return the registration result in the form of a REPLY message. If the registration is unsuccessful, the module will also return the reason for the failure accordingly.

It's worth noting that if the module successfully records the first frame it receives, it will return the result directly in the form of a REPLY message with a position ID of 0xFF FF. After recording in all five directions, the final position ID will be provided. During the recording process, the FACE RESET command can be used to terminate the recording process, and the previous recording status will be reset.

If the power is suddenly lost during the recording process, the face recording process will not be saved.

➤ **Input Instructions**

It is recommended that users slowly turn their heads in a certain direction and maintain the deflection during the actual data entry process. Please do not turn your head too quickly or return to the front face position immediately.

The angles for face recognition in all directions are as follows:

Table 13

Input direction	Deflection angle
Front	Facing the camera directly, no deflection
Up	Upward deflection: 5-55 degrees
Down	Downward deflection: 5-55 degrees
Right	Rightward deflection: 8-60 degrees
Left	Leftward deflection: 8-60 degrees

Among them, the upward and downward deflections are shown in Figure 5.

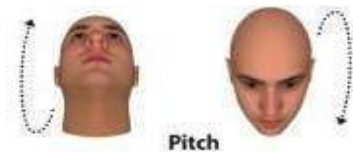


Figure 5

Left and right deflection are shown in Figure 6.



Figure 6

Please avoid turning your head as shown in Figure 7.



Figure 7

5.4.2 Single Frame Input

- ① The main difference between single-frame input process and the interactive input process is that single-frame face input only requires one positive face to successfully register, rather than requiring five interactive steps to complete the registration.
- ② Palm vein input: During the input process, naturally spread out your palm and place it directly in front of the camera at a distance of 7 to 50 cm. Move it slowly back and forth and wait until the input is successful.

5.5 Unlocking Process

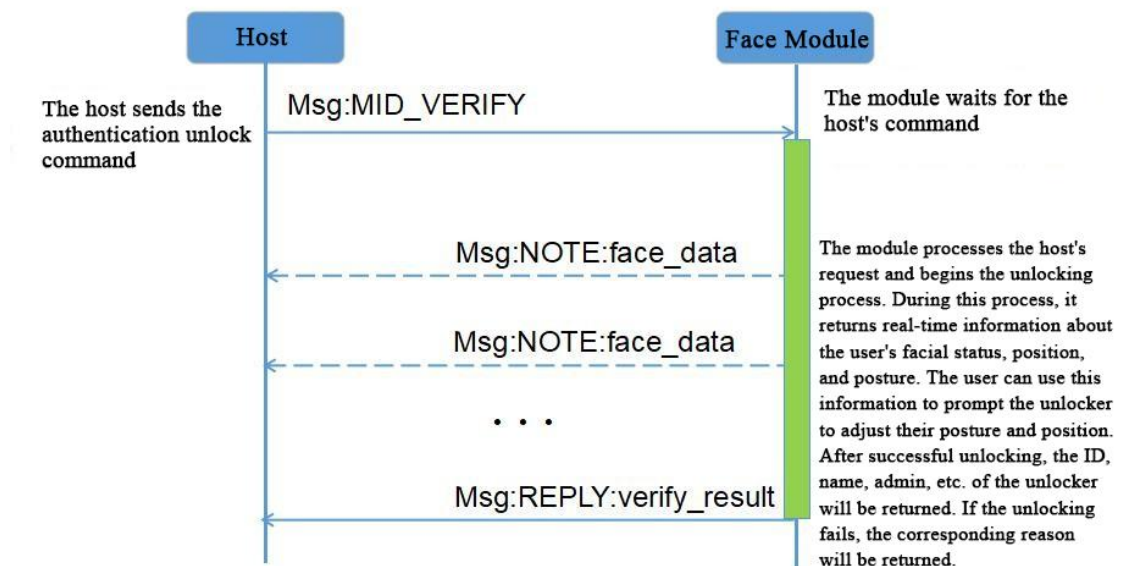


Figure 8

5.5.1 Unlocking Process Explanation

The module processes the master controller's request and begins the unlocking process. During this process, real-time information such as the status, position, and posture of the face or palm vein will be returned. Users can adjust the posture and position of the unlocker based on this information.

After successful unlocking, the ID, name, and admin information of the unlocker will be returned. If the unlocking fails, the corresponding reason will be returned.

5.6 Encrypted Communication Process

The encrypted communication process is shown in Figure 9.

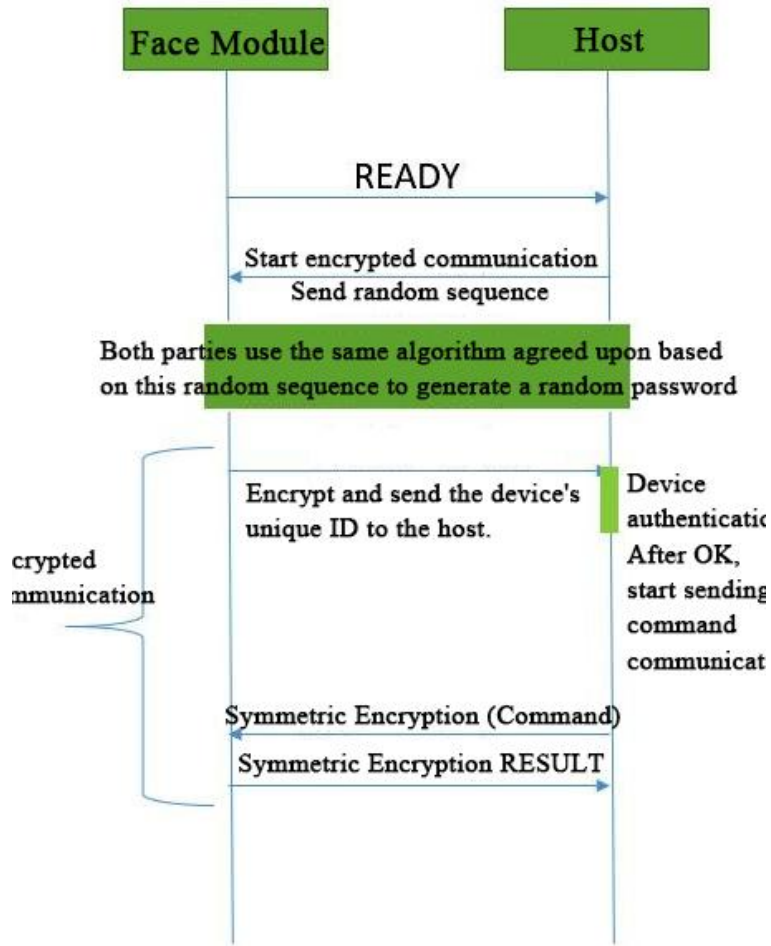
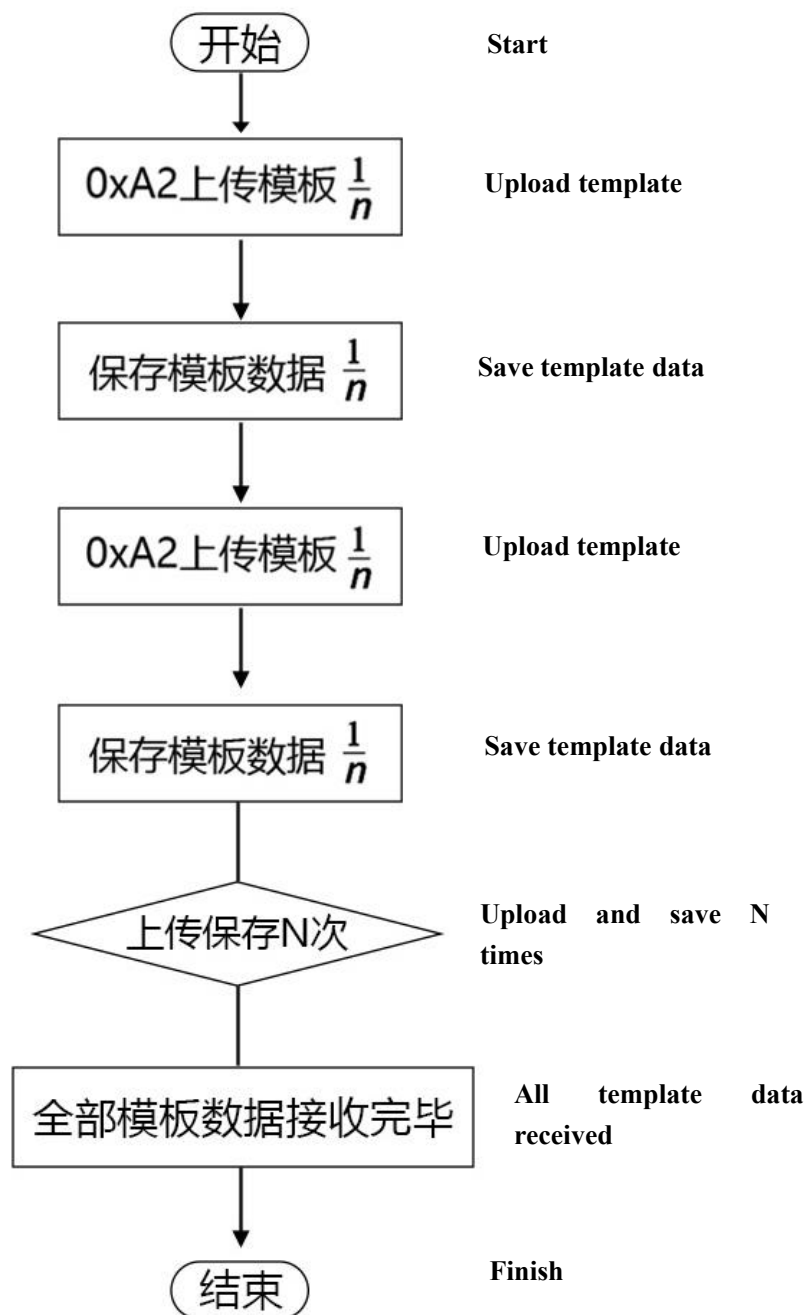


Figure 9

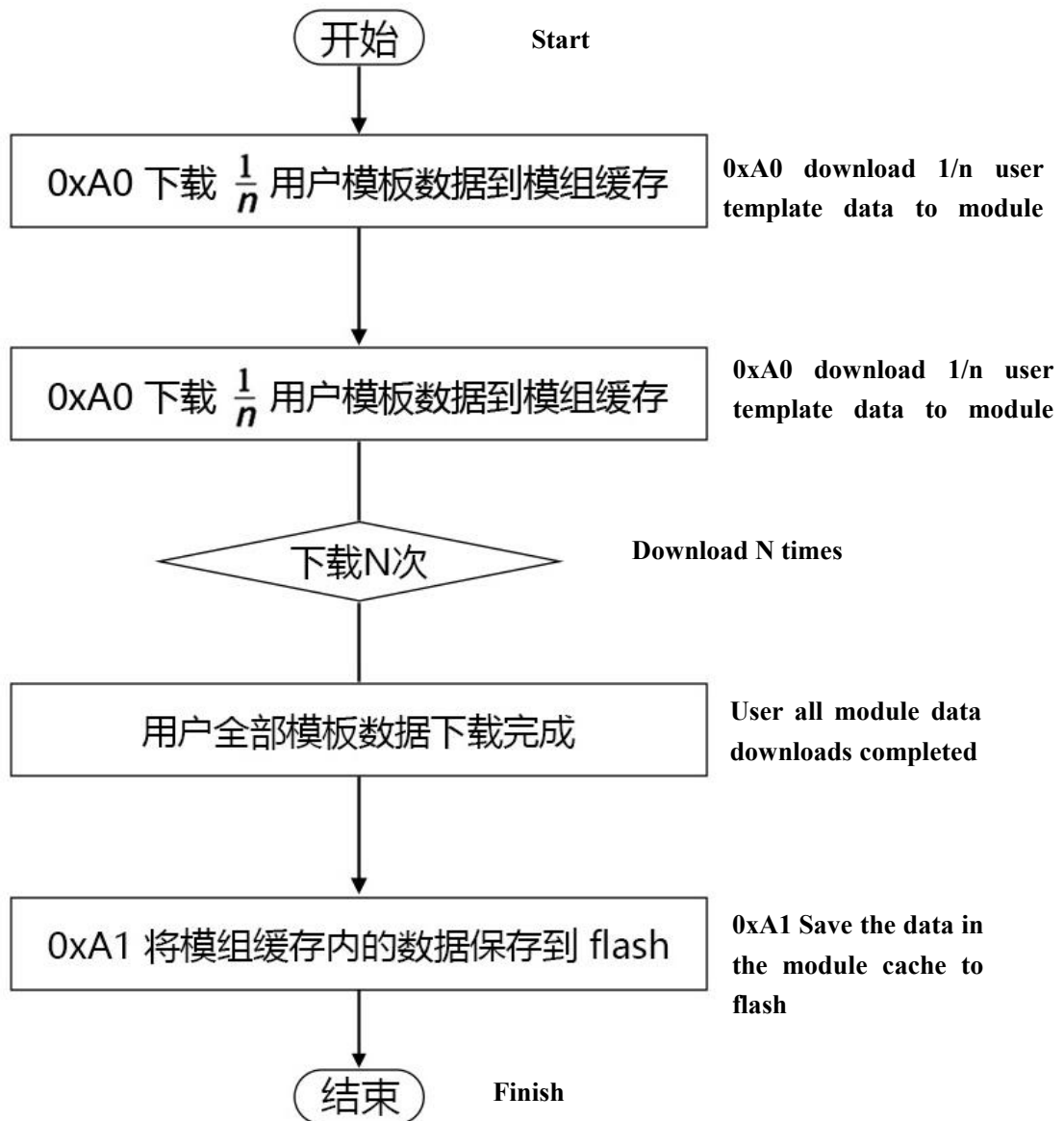
1. The host powers on the face module.
2. The face module sends READY (in plain text) to the host.
3. When the module is powered on for the first time, call the MID_SET_RELEASE_ENC_KEY command to set the 16-byte sequence required by the private protocol. The host generates a random sequence using a random number and sends it to the face module (4 bytes).
4. The face module and the host, based on this random sequence, use a custom private protocol to generate a 16-byte password. Both parties use the same algorithm to generate the same password. Subsequently, both parties use this password for encryption and decryption for this session.
5. The face module returns the status and uses a random password AES to encrypt and send the product serial number to the host.

6. The host decryption. After verifying the device ID and confirming the identity, it begins to process the demand command, such as inputting or unlocking.
7. The host uses a randomly generated password and encrypts the command and data using AES encryption method and then sends them to the face module.
8. The face module decrypts the command using the password decrypted in step 4, verifies the command's legitimacy, and processes it.
9. The face module encrypts the command processing results and data using the password decrypted in step 4 and sends them to the host.

5.7 Upload Template Process



5.8 Download Template Process



5.8.1 Upload/Download Example

5.8.1.1 Upload Face User Template

Send upload user characteristic command

Send 1:

Template Type: 00 (Face, template size 4128 bytes)

Template number: 00 01

Template offset: 00 00 00 00 Offset: 0

Uploaded data size: 00 00 0F A0 Upload data size: 4000

EF AA A2 00 0B 00 00 01 00 00 00 00 00 0F A0 07

Send 2:

Template Type: 00 (Face, template size 4128 bytes)

Template number: 00 01

Template offset: 00 00 0F A0 Offset: 4000

Uploaded data size: 00 00 0F A0 Upload data size: 128

EF AA A2 00 0B 00 00 01 00 00 0F A0 00 00 00 80 87

Received data 1:

EF AA 02 0F A0 67 FF E6 FE C0 01 E7 02 A7 FF 4C 00 59 00 DA 02 4D FE C0 01 8D 00 F4 01 8D FE
67 FF 27 FF 73 01 5A FF 1A 01 B3 00 80 FF 9A 01 B3 01 F3 00 C0 FF F3 FE 73 FF 73 00 DA FF 73
FF A7 FF 4C FD 73 00 8D FE DA FE F4 FF 4D FF F4 FF 5A 01 F4 01 66 01 B4 FF 40 01 33 FF F4 FF
66 FE B4 FF 5A 02 99 00 B3 FD F3 FE A7 FF DA FE B4 FF DA FF B4 FF 33 00 DA FF 00 01 27 FF
0D FF 59 00 5A FF A7 FF 00 01 DA FE 73 FF C0 FF 0C 00 1A 01 4C FD 99 00 19 FE 99 00 4D FE C0
FF E6 00 8D FE 99 00 66 01 B3 00 99 00 8D 00 F3 FE 73 FD 33 FE 8D 00 DA FF 4D FE C0 FF 8D FF
C0 FE C0 FF A6 FE CD FF 40 01 E7 FF 59 FE 8D 01 A7 FF 00 00 66 00 99 00 8D FF 0D FF 59 00 CD
01 A6 00 CD FF 73 FF C0 FE E6 FD 5A 02 DA FF E7 02 26 00 DA FE DA FF B3 FE 99 00 59 FE 66
FE 00 00 99 00 A6 FE 59 00 67 FF 4D 01 9A FE 00 FE 4D 02 A7 FF 19 00 99 00 DA FF A6 FE E7 FF
99 00 8D 00 D9 00 33 01 9A FF 0D FF 26 FE F3 FD 67 02 19 00 A6 FE 9A FF 5A FF 8D 02 19 FE 40
00 8D FF F3 FE 9A FF C0 00 C0 FF A7 01 F3 FE DA FF 8D 00 B3 FE 8D FF 99 FD F4 FF CD 02 1A
03 C0 00 80 00 4C 00 B4 FF 4D 01 E6 FD 67 FF 9A FE CD FF C0 FE 5A 01 9B 03 33 00 1A 02 80 FF
F3 00 00 FF 4D FF B3 FD 80 00 5A FF B3 01 F4 FF 33 FF C0 FF 26 00 B3 00 F3 00 00 00 5A 01 A6
00 0C 00 E7 FF 0C 00 B3 01 73 00 00 00 A7 FF 27 FF 8D FE 4D 01 4D FF 8D 00 40 00 1A 02 C0 FF
9A 01 F4 FF 33 FF C0 FF 73 FF 1A FF 73 FD F3 FD 27 FF DA FE 5A 01 3F FD 9A FF 26 01 00 02 66
00 59 FE DA FE 67 FF 33 FF 27 FF 33 FF 8D 00 1A 01 1A FF F3 FE A7 FF 00 00 40 01 26 01 E6 FD
33 01 26 01 27 FF A7 01 CD 01 B3 FE 9A FF 80 00 27 FF E6 FE 4D 02 B4 FF 01 30 00 00 D5 FE 0E
00 0E 01 00 02 64 FF E3 00 63 00 72 01 B9 FE 80 01 B9 FF 39 01 39 FF F2 FF E4 FE 1C 00 64 FF E4
FF 1C 00 39 FE C7 00 1C 01 38 00 8E 00 2B FF 55 FE 80 01 00 00 0E FE 55 FE 9C FE 47 00 2B FF
2A 00 80 FE 0E 00 2B FF AA 00 E4 01 D6 FF 72 FF 2A 00 38 00 9D FF 0E FE 1C FE 39 01 2A 00 72

FE C7 FE 0E FF 72 FF 63 00 2A 00 71 00 71 00 72 FF 55 00 55 FE 00 01 80 00 AB FF AB FF F2 00
E4 FE 0E FF D6 FF D5 00 63 00 C7 FD 2B 01 39 FE AB 01 E3 FC 8E FE AB FF 47 00 AB FF 9D 01
80 00 C7 01 0E 00 AB FE 72 FF 80 FE 00 01 B9 FF 0E FE 80 00 55 00 0E FF 63 00 00 FF 39 FF 00 01
9C 00 63 FC 47 01 63 00 80 FF 55 00 1C 00 80 FE 8E 00 55 00 00 01 AB FF 71 00 C8 FF E3 FD 9C
FD AA 00 E4 FF D6 02 39 FE D5 FE D5 FE 72 FF B9 FF 72 FE 00 FF 72 FF E3 00 39 FE 55 01 0E 00
39 01 E3 FD 71 FD F2 01 1D FF 47 00 47 00 F2 FF 64 FF F2 FE 1C 00 9D 01 9D 01 E4 01 55 00 39
FF E3 FD 80 FF AB 01 8F FF 00 FF 9D FF 2B FF C7 01 55 FD 39 01 F1 FD 0E FF 56 FF D6 FF 55 01
1C 01 F1 FD 80 00 B9 00 B8 FD C8 FF 2A FE B9 FF D5 00 2B 02 8E 01 8E 00 9D FF AB FE 1C 01
8E FD AB FF B9 FE 8E 00 2A FE E3 00 B9 01 63 00 F3 03 C8 FF 2B 01 0E FF 8F FF B9 FE D5 00 0E
00 39 02 47 00 2A 00 C8 FF 8E 00 AB FE 38 00 C7 00 9D 01 2A 00 F2 00 2B 01 39 01 AB 01 E3 00
55 00 0E 00 F2 FF F2 FE 71 00 AB FF 0E 00 47 01 0F 02 AB FE 0E 01 0E FF AA 00 80 FF 72 01 F2
FE F1 FD 00 FF D6 FF 63 FE D5 00 9C FC F2 FF 38 00 64 02 63 00 55 FE 1C 00 9C FE 8E 00 8F FF
B9 FE D6 FF 1C 01 E4 FE 2B FF 71 00 D6 FF 47 00 2A 00 63 FE 0E 01 C7 01 39 FF 38 00 00 01 00
FE D5 FE 00 01 E4 FF F2 00 64 03 E4 FF 01 30 00 00 00 00 35 FF 05 01 31 01 65 00 F7 00 2B 00 36
02 44 FF 9F 00 57 00 E8 00 EC FE 2F FE 35 FF BC 00 91 00 9F 00 D5 FF 09 FF 22 01 71 02 91 00 79
01 21 FE D5 FF 74 00 9B FF CF FE EC FE B2 FE 14 01 6F FF 82 00 35 FF F2 FF 0E 00 3A 00 05 01
44 FF 8C FF 6B 01 8C FF 87 FE C6 FF E3 FF F7 00 74 00 9E FD F5 FD 2F FE 5B FE C6 FF E8 00 78
FE DA 00 4C FE F2 FF 7E FF 18 FF 44 FF A9 FF 3A 00 DA 00 3E FE 52 FF A9 FF 05 01 8C FF 1B
FD 4E 01 44 FF 19 02 9E FD 8C FF 57 00 61 FF 44 FF DA 00 82 00 0E 00 F2 FF 6A FE 7E FF CA FD
6F FF CF FE 87 FE 18 FF 65 00 48 00 EE 01 26 FF E3 FF 3F 01 E8 00 95 FE A5 01 2B 00 26 FF 74 00
5C 01 78 FE D5 FF C6 FF F7 00 82 00 C6 FF E3 FF CF FE DE FE 45 02 A9 FF 53 02 48 00 35 FF 44
FF FB FE 57 00 8C FF 35 FF 79 01 DA 00 EC FE E8 00 3F 01 7E FF 64 FD 55 FD 02 03 CA FD E8 00
E8 00 F2 FF A4 FE 74 00 E8 00 BC 00 88 01 BC 00 35 FF 26 FF F5 FD 09 FF C8 02 61 FF 26 FF B8
FF 6F FF 9F 00 04 FE BC 00 E7 FD 91 00 6A FE 3F 01 BC 00 F7 00 81 FD DA 00 2B 00 FB FE 6A
FE 44 FF 8C FF 10 03 5C 01 62 02 44 FF 82 00 21 FE 45 02 4C FE 00 00 2B 00 52 FF 47 FD 74 00 3F
01 3A 00 C2 01 35 FF DF 01 44 FF 9B FF CA FD 00 00 DE FE 7F 02 12 FE 0B 02 6F FF 91 00 09 FF
AE 00 AE 00 48 00 14 01 CB 00 36 02 1D 00 14 01 F2 FF E3 FF B8 FF 82 00 B8 FF C6 FF 8C FF 82
00 62 02 F7 00 87 FE A5 01 1D 00 35 FF 9F 00 DA 00 CF FE 18 FF 04 FE 26 FF 6A FE 5C 01 87 FE
DE FE 18 FF 79 01 AE 00 CF FE C6 FF 6F FF 7E FF B2 FE 57 00 2B 00 B9 02 A7 FC E3 FF D5 FF
9B FF 0B 02 6B 01 A4 FE 7E FF 4E 01 61 FF AE 00 57 00 E3 FF 82 00 CB 00 7E FF BC 00 B4 01 87
FE 01 30 00 00 26 FF C2 00 55 00 18 00 0B 02 62 FF 55 00 9E 00 87 FF 18 00 91 00 17 01 A0 FD 93
FF 57 FD 79 01 85 00 94 FE 7B FF 57 FE D0 FF 0B 01 87 FF 23 01 3E FF 9F FF F3 00 6F FE D0 FE
57 FE F5 FD 87 FF E7 00 85 00 B8 FF 55 00 B6 00 48 01 54 01 0B 01 23 01 FF 00 87 FE DD FD D0
FF C2 00 0B 01 6D 00 A0 FD 57 FE 32 FF 26 FF 18 00 23 01 DA 00 24 00 3E FE 0D FF 0C 00 23 01
1A FE 01 FF 6F FF 79 01 3E FE E8 FF D0 FE 6C 01 91 00 DD FE 9E 00 DD FD 6C 01 E9 FE 6D 00
93 FF 61 00 0E FE 3C 01 D0 FF B6 00 7B FF B8 FF E9 FD 70 FD 00 00 AB FF D0 FF 4B FE F4 FF
B8 FE DC FF 0D FF 7B FE E7 00 3E FF A1 FC E7 00 DC FF 7B FF D0 FF 9D 01 88 FD 48 00 E7 00
30 00 9F FF E9 FE 48 00 87 FF 87 FF 6C 01 85 00 78 03 E8 FF 6F FE CE 00 DA 01 FF 00 6F FE 7B
FF 79 00 85 00 63 FD B6 00 A9 01 A9 01 0D FF 3E FE 60 02 AC FE F5 FE 18 00 48 00 63 FD 18 00
CE 00 DA 00 3C 01 85 00 DD FE E8 FF 63 FD 01 FF 48 01 30 00 B6 00 01 FF 26 FE C2 01 02 FE 26
FF 94 FE F4 FF FF 00 FF 00 60 01 0B 02 AC FD DD FE A9 01 C5 FD F5 FD B8 FE AC FE 60 02 17
02 00 00 C2 01 DC FF A0 FE B5 01 D1 FD 0E FE B8 FE E8 FF E9 FD DC FF C1 02 B6 00 6D 00 6F
FF 30 00 E8 FF 30 00 DD FE 0C 00 AB FF 79 00 48 01 B8 FF DC FF 00 00 48 01 3C 00 60 01 9D 01
00 00 DC FF 60 01 D0 FE C2 00 B6 00 B6 00 55 00 85 00 24 00 62 FF DC FF 0C 00 FF 00 AA 00 C4
FE 85 01 55 00 DA 00 DC FF 6D 00 6F FE 1A FE 6F FE 9E 00 62 FF 0B 01 0E FE 54 01 24 00 F5 FE
CE 00 4A FF E8 FF F4 FF B6 00 87 FF 0D FF DA 00 C2 00 C4 FE 32 FE 56 FF E9 FE E7 00 79 00 4B
FE 3E FF C2 00 C5 FD B6 00 47 02 56 FF 6D 00 55 00 62 FF 85 00 17 02 9F FF 01 30 00 00 6A FF 4B



[illegible]**Receive 2:**[illegible]

The face template data has been received.

5.8.1.2 Download Face User Template

Save the face template data to the module cache:

Send the command to download user characteristic:

Template Type: 00 (Face: Template size 4128 bytes)

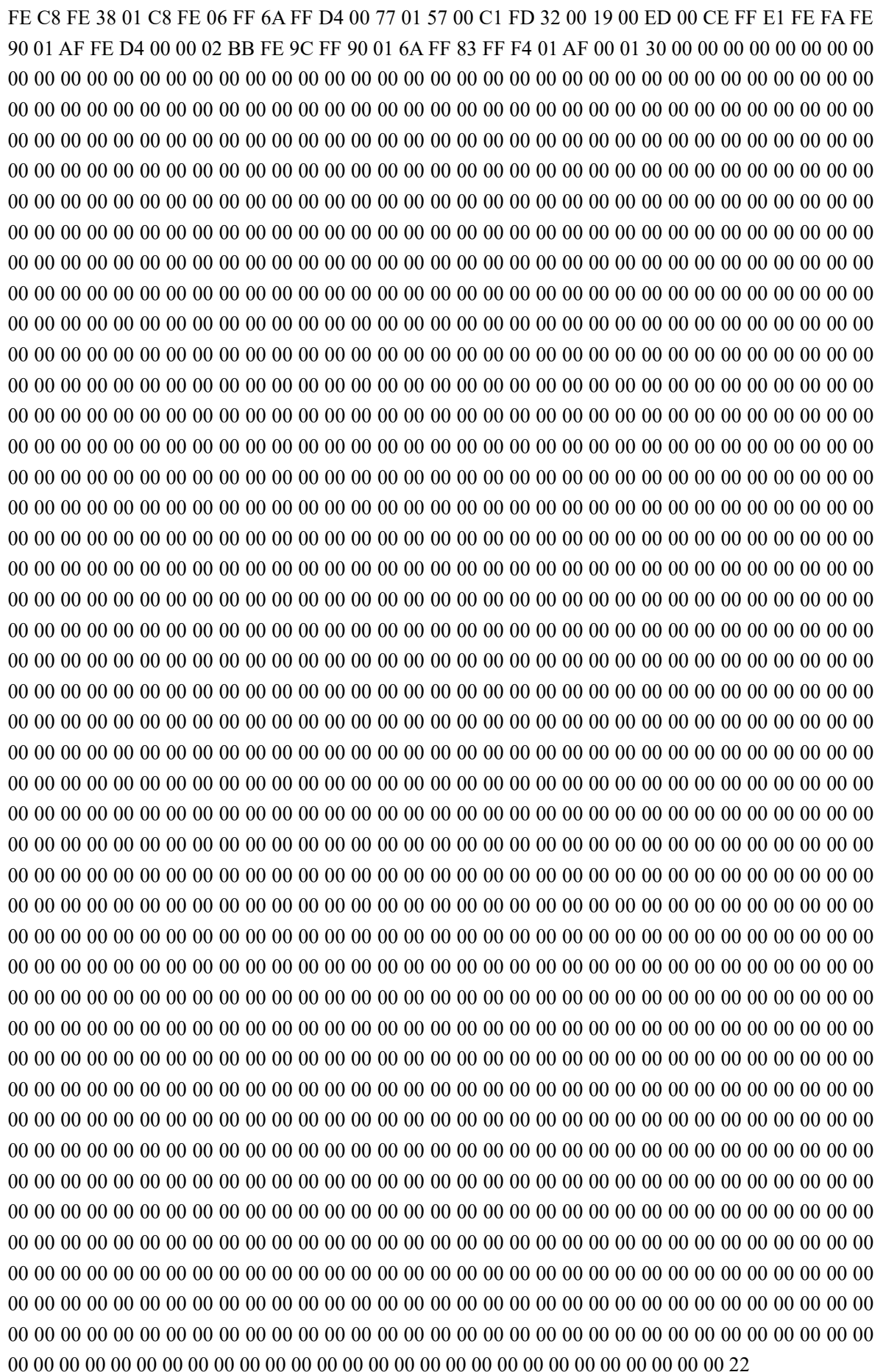
Pending upload template offset: 00 00 00 00

Upload data size: 00 00 0F A0 (Upload data size:4000bytes)

Upload data: n_datas 4000 bits

```
EF AA A0 0F A9 00 00 00 00 00 00 00 0F A0 67 FF E6 FE C0 01 E7 02 A7 FF 4C 00 59 00 DA 02 4D
FE C0 01 8D 00 F4 01 8D FE 67 FF 27 FF 73 01 5A FF 1A 01 B3 00 80 FF 9A 01 B3 01 F3 00 C0 FF
F3 FE 73 FF 73 00 DA FF 73 FF A7 FF 4C FD 73 00 8D FE DA FE F4 FF 4D FF F4 FF 5A 01 F4 01
66 01 B4 FF 40 01 33 FF F4 FF 66 FE B4 FF 5A 02 99 00 B3 FD F3 FE A7 FF DA FE B4 FF DA FF
B4 FF 33 00 DA FF 00 01 27 FF 0D FF 59 00 5A FF A7 FF 00 01 DA FE 73 FF C0 FF 0C 00 1A 01 4C
FD 99 00 19 FE 99 00 4D FE C0 FF E6 00 8D FE 99 00 66 01 B3 00 99 00 8D 00 F3 FE 73 FD 33 FE
8D 00 DA FF 4D FE C0 FF 8D FF C0 FE C0 FF A6 FE CD FF 40 01 E7 FF 59 FE 8D 01 A7 FF 00 00
66 00 99 00 8D FF 0D FF 59 00 CD 01 A6 00 CD FF 73 FF C0 FE E6 FD 5A 02 DA FF E7 02 26 00
DA FE DA FF B3 FE 99 00 59 FE 66 FE 00 00 99 00 A6 FE 59 00 67 FF 4D 01 9A FE 00 FE 4D 02 A7
FF 19 00 99 00 DA FF A6 FE E7 FF 99 00 8D 00 D9 00 33 01 9A FF 0D FF 26 FE F3 FD 67 02 19 00
A6 FE 9A FF 5A FF 8D 02 19 FE 40 00 8D FF F3 FE 9A FF C0 00 C0 FF A7 01 F3 FE DA FF 8D 00
B3 FE 8D FF 99 FD F4 FF CD 02 1A 03 C0 00 80 00 4C 00 B4 FF 4D 01 E6 FD 67 FF 9A FE CD FF
C0 FE 5A 01 9B 03 33 00 1A 02 80 FF F3 00 00 FF 4D FF B3 FD 80 00 5A FF B3 01 F4 FF 33 FF C0
FF 26 00 B3 00 F3 00 00 00 5A 01 A6 00 0C 00 E7 FF 0C 00 B3 01 73 00 00 00 A7 FF 27 FF 8D FE
4D 01 4D FF 8D 00 40 00 1A 02 C0 FF 9A 01 F4 FF 33 FF C0 FF 73 FF 1A FF 73 FD F3 FD 27 FF DA
FE 5A 01 3F FD 9A FF 26 01 00 02 66 00 59 FE DA FE 67 FF 33 FF 27 FF 33 FF 8D 00 1A 01 1A FF
F3 FE A7 FF 00 00 40 01 26 01 E6 FD 33 01 26 01 27 FF A7 01 CD 01 B3 FE 9A FF 80 00 27 FF E6
FE 4D 02 B4 FF 01 30 00 00 D5 FE 0E 00 0E 01 00 02 64 FF E3 00 63 00 72 01 B9 FE 80 01 B9 FF 39
01 39 FF F2 FF E4 FE 1C 00 64 FF E4 FF 1C 00 39 FE C7 00 1C 01 38 00 8E 00 2B FF 55 FE 80 01
00 00 0E FE 55 FE 9C FE 47 00 2B FF 2A 00 80 FE 0E 00 2B FF AA 00 E4 01 D6 FF 72 FF 2A 00 38
00 9D FF 0E FE 1C FE 39 01 2A 00 72 FE C7 FE 0E FF 72 FF 63 00 2A 00 71 00 71 00 72 FF 55 00 55
FE 00 01 80 00 AB FF AB FF F2 00 E4 FE 0E FF D6 FF D5 00 63 00 C7 FD 2B 01 39 FE AB 01 E3
FC 8E FE AB FF 47 00 AB FF 9D 01 80 00 C7 01 0E 00 AB FE 72 FF 80 FE 00 01 B9 FF 0E FE 80 00
55 00 0E FF 63 00 00 FF 39 FF 00 01 9C 00 63 FC 47 01 63 00 80 FF 55 00 1C 00 80 FE 8E 00 55 00
00 01 AB FF 71 00 C8 FF E3 FD 9C FD AA 00 E4 FF D6 02 39 FE D5 FE D5 FE 72 FF B9 FF 72 FE
00 FF 72 FF E3 00 39 FE 55 01 0E 00 39 01 E3 FD 71 FD F2 01 1D FF 47 00 47 00 F2 FF 64 FF F2 FE
1C 00 9D 01 9D 01 E4 01 55 00 39 FF E3 FD 80 FF AB 01 8F FF 00 FF 9D FF 2B FF C7 01 55 FD 39
01 F1 FD 0E FF 56 FF D6 FF 55 01 1C 01 F1 FD 80 00 B9 00 B8 FD C8 FF 2A FE B9 FF D5 00 2B 02
8E 01 8E 00 9D FF AB FE 1C 01 8E FD AB FF B9 FE 8E 00 2A FE E3 00 B9 01 63 00 F3 03 C8 FF
2B 01 0E FF 8F FF B9 FE D5 00 0E 00 39 02 47 00 2A 00 C8 FF 8E 00 AB FE 38 00 C7 00 9D 01 2A
00 F2 00 2B 01 39 01 AB 01 E3 00 55 00 0E 00 F2 FF F2 FE 71 00 AB FF 0E 00 47 01 0F 02 AB FE
0E 01 0E FF AA 00 80 FF 72 01 F2 FE F1 FD 00 FF D6 FF 63 FE D5 00 9C FC F2 FF 38 00 64 02 63
00 55 FE 1C 00 9C FE 8E 00 8F FF B9 FE D6 FF 1C 01 E4 FE 2B FF 71 00 D6 FF 47 00 2A 00 63 FE
0E 01 C7 01 39 FF 38 00 00 01 00 FE D5 FE 00 01 E4 FF F2 00 64 03 E4 FF 01 30 00 00 00 00 35 FF
05 01 31 01 65 00 F7 00 2B 00 36 02 44 FF 9F 00 57 00 E8 00 EC FE 2F FE 35 FF BC 00 91 00 9F 00
```

D5 FF 09 FF 22 01 71 02 91 00 79 01 21 FE D5 FF 74 00 9B FF CF FE EC FE B2 FE 14 01 6F FF 82
00 35 FF F2 FF 0E 00 3A 00 05 01 44 FF 8C FF 6B 01 8C FF 87 FE C6 FF E3 FF F7 00 74 00 9E FD
F5 FD 2F FE 5B FE C6 FF E8 00 78 FE DA 00 4C FE F2 FF 7E FF 18 FF 44 FF A9 FF 3A 00 DA 00
3E FE 52 FF A9 FF 05 01 8C FF 1B FD 4E 01 44 FF 19 02 9E FD 8C FF 57 00 61 FF 44 FF DA 00 82
00 0E 00 F2 FF 6A FE 7E FF CA FD 6F FF CF FE 87 FE 18 FF 65 00 48 00 EE 01 26 FF E3 FF 3F 01
E8 00 95 FE A5 01 2B 00 26 FF 74 00 5C 01 78 FE D5 FF C6 FF F7 00 82 00 C6 FF E3 FF CF FE DE
FE 45 02 A9 FF 53 02 48 00 35 FF 44 FF FB FE 57 00 8C FF 35 FF 79 01 DA 00 EC FE E8 00 3F 01
7E FF 64 FD 55 FD 02 03 CA FD E8 00 E8 00 F2 FF A4 FE 74 00 E8 00 BC 00 88 01 BC 00 35 FF 26
FF F5 FD 09 FF C8 02 61 FF 26 FF B8 FF 6F FF 9F 00 04 FE BC 00 E7 FD 91 00 6A FE 3F 01 BC 00
F7 00 81 FD DA 00 2B 00 FB FE 6A FE 44 FF 8C FF 10 03 5C 01 62 02 44 FF 82 00 21 FE 45 02 4C
FE 00 00 2B 00 52 FF 47 FD 74 00 3F 01 3A 00 C2 01 35 FF DF 01 44 FF 9B FF CA FD 00 00 DE FE
7F 02 12 FE 0B 02 6F FF 91 00 09 FF AE 00 AE 00 48 00 14 01 CB 00 36 02 1D 00 14 01 F2 FF E3 FF
B8 FF 82 00 B8 FF C6 FF 8C FF 82 00 62 02 F7 00 87 FE A5 01 1D 00 35 FF 9F 00 DA 00 CF FE 18
FF 04 FE 26 FF 6A FE 5C 01 87 FE DE FE 18 FF 79 01 AE 00 CF FE C6 FF 6F FF 7E FF B2 FE 57 00
2B 00 B9 02 A7 FC E3 FF D5 FF 9B FF 0B 02 6B 01 A4 FE 7E FF 4E 01 61 FF AE 00 57 00 E3 FF 82
00 CB 00 7E FF BC 00 B4 01 87 FE 01 30 00 00 26 FF C2 00 55 00 18 00 0B 02 62 FF 55 00 9E 00 87
FF 18 00 91 00 17 01 A0 FD 93 FF 57 FD 79 01 85 00 94 FE 7B FF 57 FE D0 FF 0B 01 87 FF 23 01
3E FF 9F FF F3 00 6F FE D0 FE 57 FE F5 FD 87 FF E7 00 85 00 B8 FF 55 00 B6 00 48 01 54 01 0B
01 23 01 FF 00 87 FE DD FD D0 FF C2 00 0B 01 6D 00 A0 FD 57 FE 32 FF 26 FF 18 00 23 01 DA 00
24 00 3E FE 0D FF 0C 00 23 01 1A FE 01 FF 6F FF 79 01 3E FE E8 FF D0 FE 6C 01 91 00 DD FE 9E
00 DD FD 6C 01 E9 FE 6D 00 93 FF 61 00 0E FE 3C 01 D0 FF B6 00 7B FF B8 FF E9 FD 70 FD 00
00 AB FF D0 FF 4B FE F4 FF B8 FE DC FF 0D FF 7B FE E7 00 3E FF A1 FC E7 00 DC FF 7B FF D0
FF 9D 01 88 FD 48 00 E7 00 30 00 9F FF E9 FE 48 00 87 FF 87 FF 6C 01 85 00 78 03 E8 FF 6F FE
CE 00 DA 01 FF 00 6F FE 7B FF 79 00 85 00 63 FD B6 00 A9 01 A9 01 0D FF 3E FE 60 02 AC FE F5
FE 18 00 48 00 63 FD 18 00 CE 00 DA 00 3C 01 85 00 DD FE E8 FF 63 FD 01 FF 48 01 30 00 B6 00
01 FF 26 FE C2 01 02 FE 26 FF 94 FE F4 FF FF 00 FF 00 60 01 0B 02 AC FD DD FE A9 01 C5 FD F5
FD B8 FE AC FE 60 02 17 02 00 00 C2 01 DC FF A0 FE B5 01 D1 FD 0E FE B8 FE E8 FF E9 FD DC
FF C1 02 B6 00 6D 00 6F FF 30 00 E8 FF 30 00 DD FE 0C 00 AB FF 79 00 48 01 B8 FF DC FF 00 00
48 01 3C 00 60 01 9D 01 00 00 DC FF 60 01 D0 FE C2 00 B6 00 B6 00 55 00 85 00 24 00 62 FF DC
FF 0C 00 FF 00 AA 00 C4 FE 85 01 55 00 DA 00 DC FF 6D 00 6F FE 1A FE 6F FE 9E 00 62 FF 0B 01
0E FE 54 01 24 00 F5 FE CE 00 4A FF E8 FF F4 FF B6 00 87 FF 0D FF DA 00 C2 00 C4 FE 32 FE 56
FF E9 FE E7 00 79 00 4B FE 3E FF C2 00 C5 FD B6 00 47 02 56 FF 6D 00 55 00 62 FF 85 00 17 02
9F FF 01 30 00 00 6A FF 4B 00 45 FF 19 00 C2 01 32 00 FA 00 0D 02 DB FF 13 01 4B 00 0C 00 64
FE 45 FF B5 FD D4 00 5E FF 0C 00 C8 00 3E FE 77 FF 38 01 64 FE 3E 00 AF 00 F4 FF 51 01 9C FF
1F FF 00 00 F3 FD 45 FF F4 FF E1 00 06 FF AF FE 1F 01 32 00 DB 01 3E 00 1F FF ED 00 A2 FE ED
FE C8 FE 77 01 51 01 2C 01 1F FF 96 FE E7 FF 45 01 ED FE 45 FF 83 01 B5 FF 96 FE BB FE ED 00
7D 00 06 FF 2C FF ED FE 8A 02 51 FF 89 FE FA FE BB 00 E1 00 C7 FC 3E 00 96 FE 6A 01 96 FE 1F
FF D4 FE 7D FE 57 FE 2C 01 70 00 3E 00 E1 00 ED FE 19 FE ED FE 0C 00 32 00 90 FF 3E 00 4B FE
13 FF C2 FF 70 00 76 FD AF 02 7D FE 5D FD D4 00 64 00 CE FF A2 00 70 00 AF FE AF 00 51 FF 19
00 38 FF 4B 00 CE FF C2 FF 13 FF 0D 02 00 00 4B 02 38 01 E1 FE 38 FF 77 FF F4 FF E7 FF 51 FF
CE FF BB 00 AF FE 13 01 AF 00 45 03 D4 FE 76 FD 13 01 06 FF 5E FF A2 00 13 01 00 FE 89 00 19
00 64 02 2C 01 A2 00 13 FF 00 00 1F FD 96 FE AF 02 F4 FF 3E 00 E7 FF F3 FD A9 01 00 FE 3E 00
96 FE 32 00 DB FF BB 00 90 01 B5 01 E7 FF 57 00 6A 01 A8 FD 06 FF CE FD 25 00 96 02 A3 02 CE
FF E7 FF A2 00 83 FF 6A 01 76 FD 06 FF B5 FF 45 FF 57 FE 7D 00 4B 02 45 01 B5 01 F4 FF DA FD
C2 FF 0C 00 77 FF C8 00 4B 00 FA 00 71 02 C8 00 ED FE E7 FF 7D 00 19 00 ED 00 5E 01 3E 00 C8
00 E7 01 C8 00 3F 02 9C 01 A9 FF C2 FF B5 FF ED FE 96 00 C8 FE E1 00 70 00 00 02 90 FF 32 00
A2 00 C2 FF E7 FF A2 00 19 00 1F FD F3 FD E1 00 38 FF 25 00 1F FF 06 01 13 01 32 00 F4 FF 89





Received return data 1: EF AA 00 00 02 A0 00 A2

Send 2:

Template Type: 00 (Face: Template size 4128 bytes)

Pending upload template offset: 00 00 0F A0 data offset: 4000

Upload data size: 00 00 00 80 (Upload data size: 128bytes)

Upload data: n_datas 128 bits

[illegible]

Received return data 2: EF AA 00 00 02 A0 00 A2

The face template data has been downloaded.

5.8.1.3 Store the template data downloaded to the module cache into position 5 of the FLASH.

Template Type: 00 Face

Template storage location: 00 05

EF AA A1 00 03 00 00 05 A7

Received return data:

EF AA 00 00 02 A1 00 A3

Face template stored