FD21 Face and Palm Vein Recognition Module Communication Protocol

Version 1.1, September 2024

Version History

Version Date		Modifications		
		chapter	Revised	content
1.0	2024-7-24	All	by bobo Initial version	
1.1	2024-9-25	14	bobo adds upload image command	

Table of contents

Machine Translated by Google

Communication Protocol

The module and the main controller communicate via serial port. This chapter will explain the serial port configuration and communication protocol.

1 Serial port configuration

The baud rate of the serial port communication between the module and the main control is 115200bps. The specific parameters are shown in Table 1. Other baud rates can be supported according to the requirements of the main control.

Rate.

Table 1

Configuration items	illustrate
Baud rate	Supports most baud rates, default is 115200
Hardware/software flow control	Not used
Data bits	8
Stop bits	1
Parity bit	Not used

2 Communication message format

The basic message format for communication between the master 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 2-1 (customized)

SyncWord	MsgID	Size	SequenceID	Data	ParityCheck
2 bytes	1 byte	2 bytes	4 bytes	N bytes	1 byte

Table 2-1 shows the data packet structure for adding SequenceID. It is added only when there are specific requirements. The data packet sent by the host computer contains

When SequenceID is specified, the response returned by the face module must also include SequenceID in order to determine which command it is responding to.

SequenceID = (seq_id[0] << 24) | (seq_id[1] << 16) | (seq_id[2] << 8) | (seq_id[3]);

Table 3 is a detailed description of each field.

Table 3

Field length		illustrate
SyncWord	2bytes	Fixed message start synchronization word 0xEF 0xAA
MsgID	1byte	Message ID (e.g. RESET)
Size	2bytes	Data size, in bytes
Data	N bytes of data corresponding to the message, such as the parameters corresponding to the command message.	65535>=N>=0, N=0 means this message has no parameters.
Parity Check	1 byte parity check code of the protocol, calculated by excluding the Sync Word from the entire protocol	After the first part is completed, the remaining bytes are bitwise XORed.

3 Communication Message Description

Each message processing has a corresponding timeout definition. Timeout refers to the maximum time the master waits for the module to process.

That is, the time when the module feedback processing is completed.

When the instruction processing times out, the master controller can adopt the following processing methods:

1. Send MID_GETSTATUS command

This message can quickly obtain the working status of the module. If it crashes, there will be no return and the main control can be powered off directly.

Send a RESET command

Stop all current processing, the module enters the standby state, and waits for new instructions from the master.

3. Think the module is frozen

The main control can adopt the power-off restart processing method.

For specific instruction format, please refer to [15.1].

4 Sending and receiving module messages

4.1 Message Receiving (H>>M)

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

Table 4

Name	SyncWord	MsgID	Size		Data	ParityCheck
Bytes:	2 bytes	1 byte	2 bytes		N bytes	1 byte
Contents	0xEFAA	command	high eight Bit	Low eight Bit	data	checksum

For detailed description of command and data, please refer to [15.1]

4.2 Message Sending (M>>H)

The module mainly sends three types of messages: REPLY, NOTE, and IMAGE.

1) Sending REPLY message

Reply is the module's response to the command sent by the master. The module will eventually reply to each command from the master.

Each command sent by the master to the module will eventually receive a MID_REPLY reply from the module, which contains the command sent by the master mid, the command execution result result, and the possible returned data data.

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

Table 5

Name	SyncWord	MsgID	Size		Data			ParityCheck
Bytes:	2 bytes	1 byte	2 bytes		N bytes			1 byte
Contents	0xEFAA	MID_REPLY (0x00)	high eight Bit	Low eight Bit	s_msg_reply_data			checksum
					mid (1byte)	result (1byte)	data[0] (n-byte)	

mid indicates the task that the module is currently processing. For example, when mid = MID_ENROLL (0x13), it means that the message is processed by the module.

The message you reply to after completing the enroll task.

result indicates the execution result of the command, as shown in Table 6.

Table 6

result	code	illustrate
MR_SUCCESS	0	Success
MR_REJECTED	1	The module rejects the command
MR_ABORTED	2	The entry/unlock algorithm has terminated
MR_FAILED4_CAMERA	4	Failed to open the camera
MR_FAILED4_UNKNOWNREASON	5	Unknown error
MR_FAILED4_INVALIDPARAM	6	Invalid parameter
MR_FAILED4_NOMEMORY	7	Insufficient memory
MR_FAILED4_UNKNOWNUSER	8	No users registered
MR_FAILED4_MAXUSER	9	The maximum number of users entered has exceeded
MR_FAILED4_FACEENROLLED	10	faces have been recorded
MR_FAILED4_LIVENESSCHECK	12	Liveness detection failed
MR_FAILED4_TIMEOUT	13	Entry or unlock timeout
MR_FAILED4_AUTHORIZATION	14	Encryption chip authorization failed
MR_FAILED4_READ_FILE	19	Failed to read file
MR_FAILED4_WRITE_FILE	20	Failed to write file
MR_FAILED4_NO_ENCRYPT	21	Communication protocol is not encrypted
MR_FAILED4_NO_RGBIMAGE	23	RGB image is not ready
MR_FAILED4_UNKNOWN_HANDUSER	239	Palm vein verification failed
MR_FAILED4_NOCAMERA	240	does not support color lens
MR_FAILED4_HANDENROLLED	241	Palmar vein duplication

Example: 0xEF 0xAA 0x00 0x00 0x04 0x13 0x00 0x00 0x01 0x16

name say	SyncWord	MsgID	Size	Data	Parity Check
Inside Allow	0xEFAA	0x00 (MID_RE	0x00 0x04	s_msg_reply_data 0x13 0x00 0x00 0x01	0x11

		PLY)			(MID_E (NROLL)	(MR_SU CCESS)	(use r_id _have)	(use r_id _leb)	

This message indicates that this is the REPLY message sent by the module to the main control. The data part occupies 4 bytes. The entry is successful and the user ID entered is

It is 1.

2) Sending NOTE messages

Note is the information that the module actively sends to the main controller. The message type and the adapted data structure are determined according to the note id.

MSG::NOTE is the information that the module actively reports to the main control. For example, when the module is powered on, it will actively send a NOTE to the main control to indicate that

It is READY and can receive commands from the master. After receiving the NOTE message, the master can start sending input or unlock commands.

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

a) When powered on, the module sends NID_READY to the main control

b) During the input process, the module sends facial information to the main control

c) During the unlocking process, the module sends facial information to the main control

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

Table 7

NameSyncWord		MsgID	Size		Data		ParityCheck
Bytes: 2 bytes		1 byte 2 bytes			N bytes		1 byte
Contents 0xEFAA		MID_NOTE (0x01)	high eight Bit	Low eight Bit	s_msg_note_data		checksum
					not (1byte)	data[0] (n-bytes)	

nid represents the execution result of the algorithm during the enroll process, as follows:

Table 8

nid (* indicates data is carried)	code	illustrate
NOT_READY*	0	Module is ready
NID_FACE_STATE*	1	The algorithm is executed successfully and returns the face information s_note_data_face

NOT_UNKNOWNERROR	2	Unknown error
NID_OTA_DONE*	3	OTA upgrade completed
NID_EYE_STATE	4	Eyes open or closed during unlocking

Date[0] of NID_READY is a single byte, indicating the module firmware type. The main controller can ignore it.

date[0] of NID_OTA_DONE is one byte. 0: OTA success, 1: OTA fail.

data[0] of NID_EYE_STATE is one byte.

12: Eyes open detected in closed eyes mode, 13: Eyes closed for one second, 14: Unable to confirm.

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

Table 9

structure	s_note_data_face							
Contents	state	left	top	right	bottom	yaw	pitch	roll
ÿÿ int16_t	int16_t int16_t int16_t	int16_t int16_t	int16_t int16_t					
Bytes 2 bytes	2 bytes 2 bytes			2bytes	2bytes 2bytes	2bytes		2bytes

State: indicates the current state of the face, mainly including the following situations:

Table 10

Face state	code	illustrate
FACE_STATE_NORMAL	0	Normal face
FACE_STATE_NOFACE	1	No face detected
FACE_STATE_TOOUP	2	The face is too close to the top edge of the image and cannot be recorded.
FACE_STATE_TOODOWN	3	The face is too close to the bottom edge of the image and cannot be recorded.
FACE_STATE_TOOLEFT	4	The face is too close to the left edge of the image and cannot be recorded.
FACE_STATE_TOORIGHT	5	The face is too close to the right edge of the image and cannot be recorded.
FACE_STATE_FAR	6	The face is too far away and cannot be recorded.
FACE_STATE_CLOSE	7	The face is too close and cannot be recorded.
FACE_STATE_EYEBROW_OCCLUSION	8	Eyebrows covering
FACE_STATE_EYE_OCCLUSION	9	Eye covering
FACE_STATE_FACE_OCCLUSION	10	Face occlusion

Face state	code	illustrate
FACE_STATE_DIRECTION_ERROR	11	Wrong face orientation entered
FACE_STATE_EYE_CLOSE_STATUS_ OPEN_EYE	12	Detecting open eyes in closed eyes mode
FACE_STATE_EYE_CLOSE_STATUS	13	Eyes closed
FACE_STATE_EYE_CLOSE_UNKNOW_ STATUS	14	Unable to determine whether eyes are open or closed during eye closure mode detection
FACE_STATE_HAND_NORMAL	128	Palmar vein detection
FACE_STATE_HAND_FAR	129	The palm is too far away and cannot be recorded.
FACE_STATE_HAND_CLOSE	130	The palm is too close to be recorded.
FACE_STATE_HAND_TOOUP	131	The palm is too close to the top edge of the image and cannot be recorded.
FACE_STATE_HAND_TOODOWN	132	The palm is too close to the bottom edge of the image and cannot be recorded.
FACE_STATE_HAND_TOOLEFT	133	The palm is too close to the left edge of the image and cannot be recorded.
FACE_STATE_HAND_TOORIGHT	134	The palm is too close to the right edge of the image and cannot be recorded.

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

left: The distance between the face frame and the leftmost side of the image (negative number means the face frame has exceeded the leftmost side of the image)

top: The distance between the face frame and the top of the image (negative numbers indicate that the face frame is beyond the top of the image)

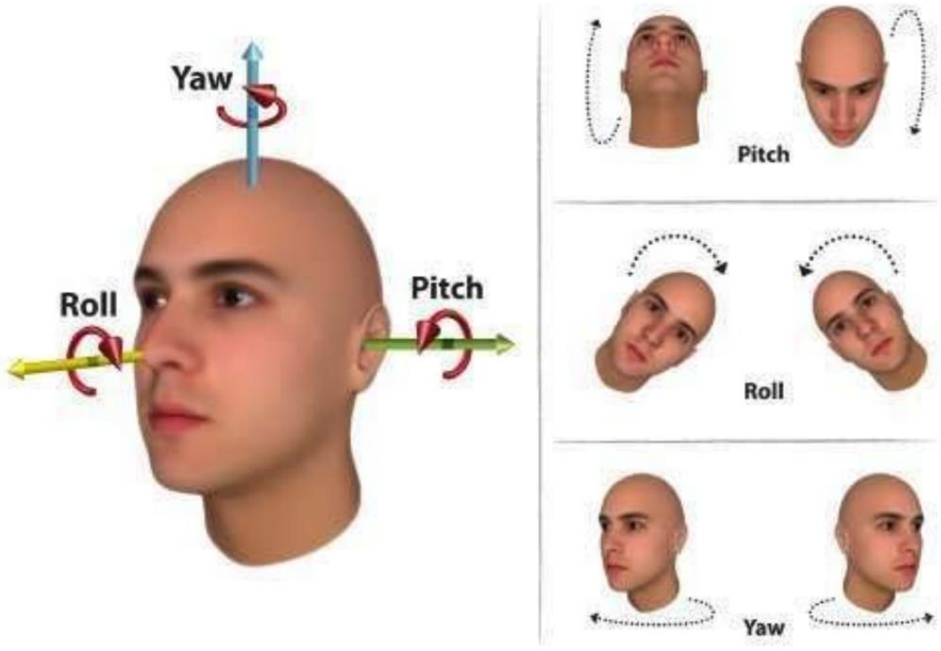
right: The distance between the face frame and the rightmost side of the image (negative numbers indicate the face frame has exceeded the rightmost side of the image)

bottom: The distance between the face frame and the bottom of the image (negative numbers indicate that the face frame has exceeded the bottom of the image)

The rotation directions represented by yaw, pitch, and roll are shown in the figure below. A negative yaw indicates a left turn, and a positive yaw indicates a right turn.

Turn the head; a negative pitch indicates raising the head upward, a positive pitch indicates lowering the head downward; a negative roll indicates tilting the head to the right, a positive roll indicates

Tilt your head to the left.



Example: 0xEF 0xAA 0x01 0x00 0x01 0x00 0x00

name say	SyncWord	MsgID	Size		Data		ParityCheck
Inside Allow	0xEFAA	0x01 (MID_NO (IT)	0x00 0x01		s_msg_note_data		0x00
					0x00 (NOT_READY)	none	

This message indicates that this is a NOTE message sent by the module to the main control. The data length is 1 byte. The module is ready to receive other

Order.

3) IMAGE message sending

The module transmits the image to the main control.

The complete protocol of the IMAGE message sent by the module to the master is shown in Table 11.

Table 11

Name	SyncWord	MsgID	Size		Data	ParityCheck
Bytes:	2 bytes	1 byte	2 bytes		N bytes	1 byte
Contents	0xEFAA	MID_IMAGE (0x02)	high eight Bit	Low eight Bit	image data	checksum

When the face module transmits pictures, one frame of data supports a maximum of 4000 bytes of data transmission.

5 Master Controller Message Receiving Process

The message sent by the master receiving module can follow the process shown in Figure 1.

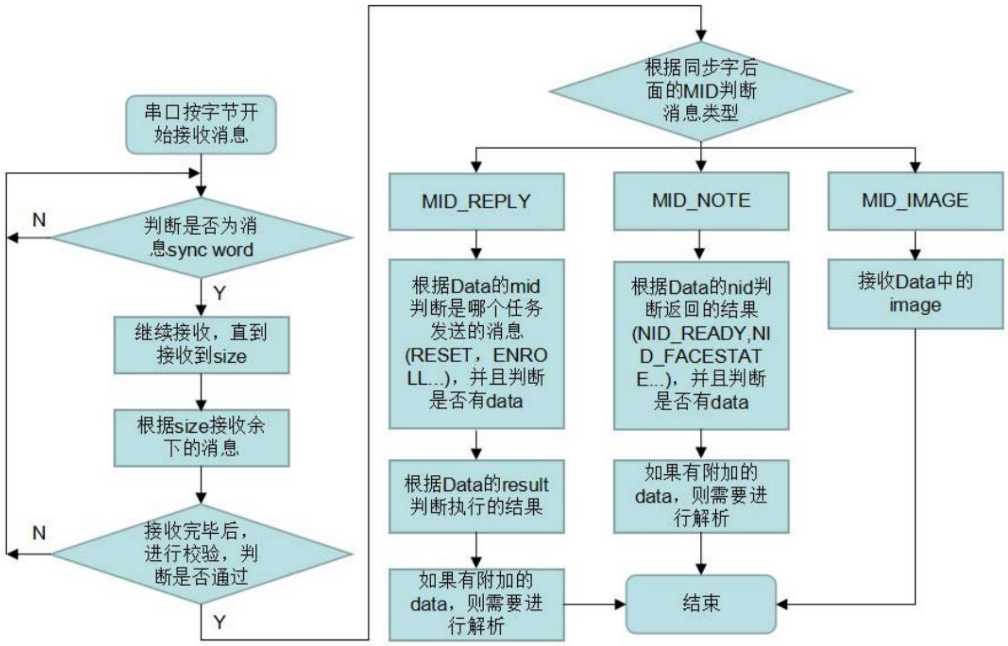
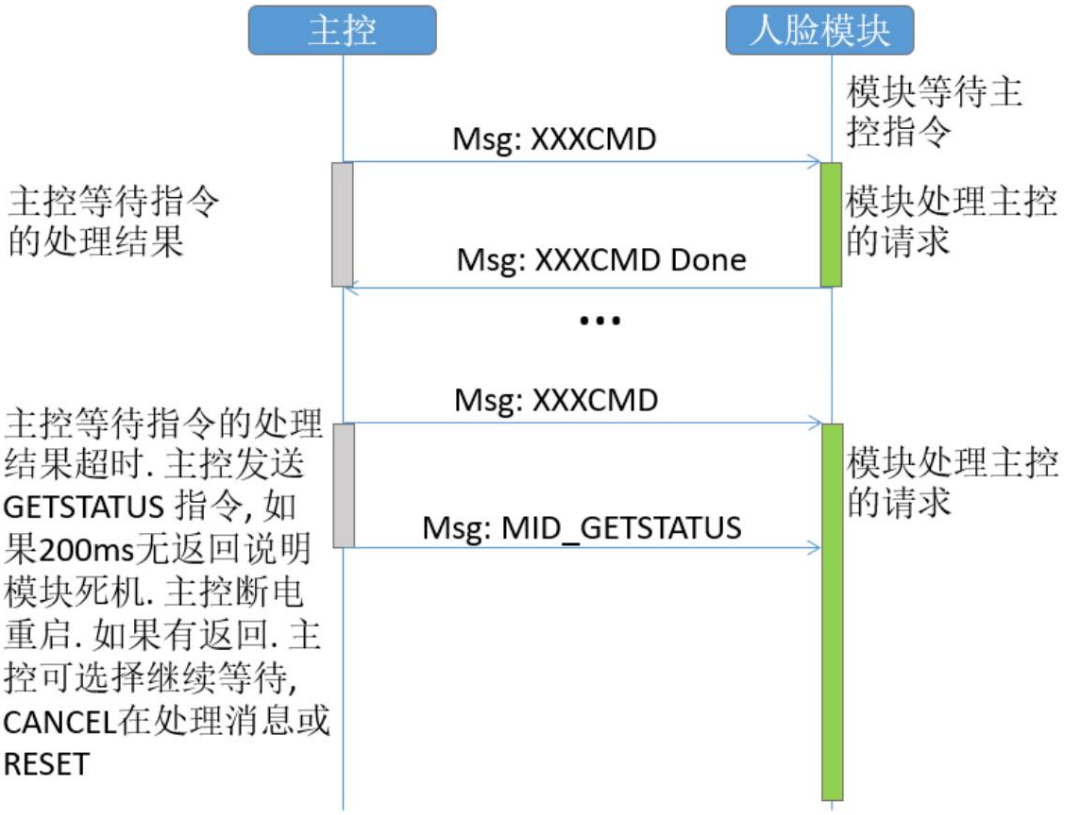

串口按字节开始接收消息
判断是否为消息sync word
N
Y
继续接收，直到接收到size
根据size接收余下的消息
接收完毕后，进行校验，判断是否通过
N
Y
根据同步字后面的MID判断消息类型
MID_REPLY
MID_NOTE
MID_IMAGE
根据Data的mid判断是哪个任务发送的消息(RESET，ENROLL...)，并且判断是否有data
根据Data的nid判断返回的结果(NID_READY,NID_FACESTATE...)，并且判断是否有data
接收Data中的image
根据Data的result判断执行的结果
如果有附加的data，则需要进行解析
如果有附加的data，则需要进行解析
结束

Figure 1

6 General message processing flow



主控等待指令的处理结果超时. 主控发送GETSTATUS 指令, 如果200ms无返回说明模块死机. 主控断电重启. 如果有返回. 主控可选择继续等待, CANCEL在处理消息或RESET

Figure 2

7 Power On and Off Process

After the module is powered on, it takes about 1.1 seconds to initialize. During this period, the module cannot respond to the host computer command.

After initialization, the module will immediately send a status notification packet to the host computer, indicating that the module can work normally and receive commands from the host computer.

The main control actively controls the power on and off of the module, as shown in Figure 3. When the module is powered on, it enters the startup process and starts to enter the standby state.

The status will notify the master control MSG::NOTE::READY, the master control starts to send command messages, and the module processes and returns the results.

If the module does not respond within 2 seconds, the main control calls MSG::POWERDOWN, the module processes and returns, and the main control can be powered off.

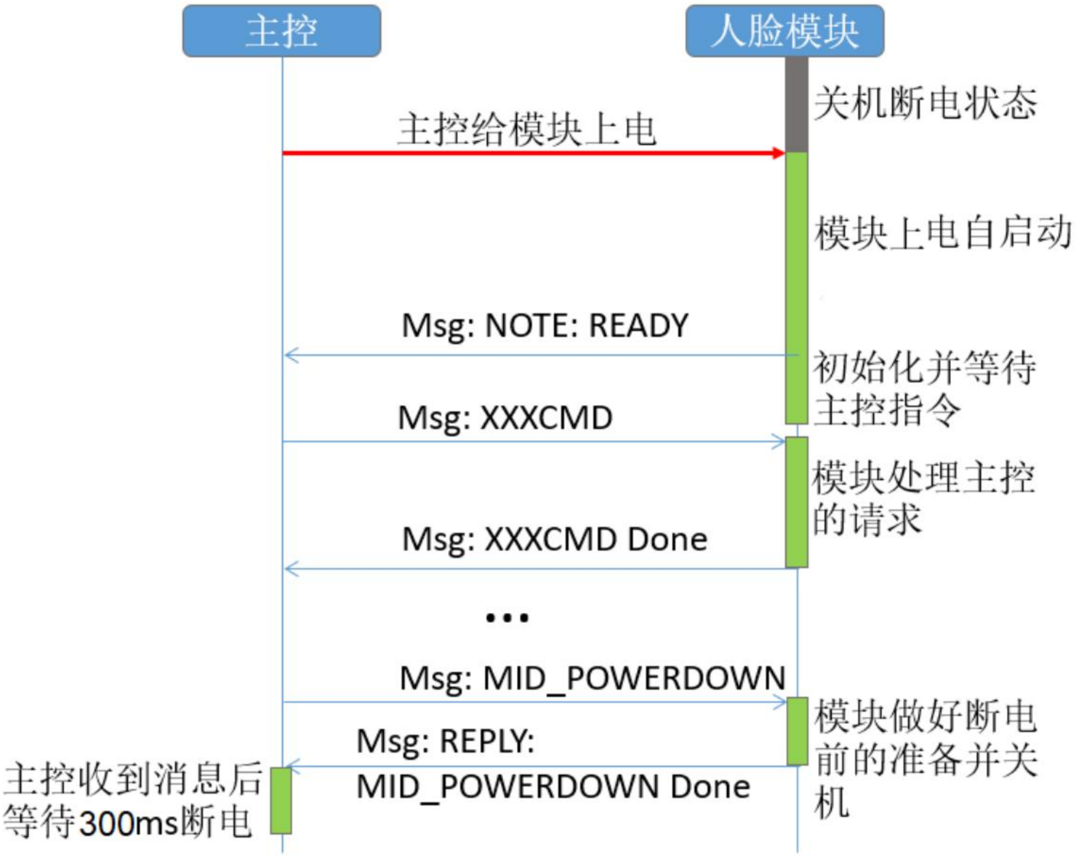


Figure 3

8 Entry Process

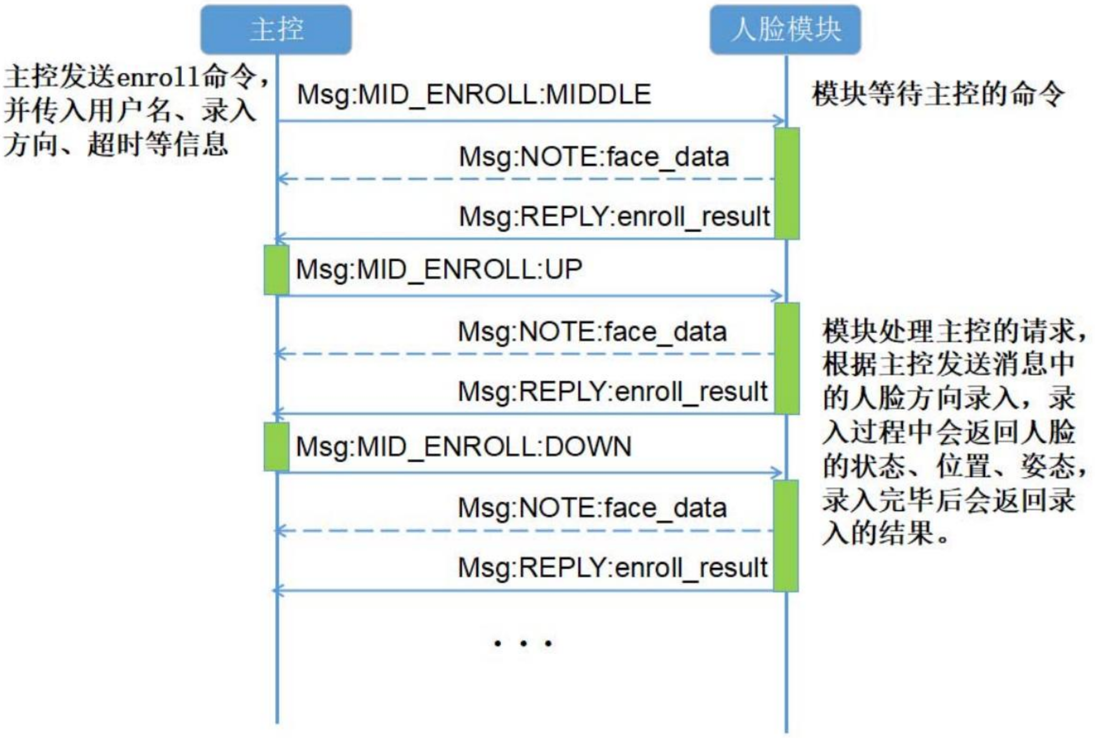


Figure 4

During the recording process, the face module does not limit the order of recording directions. Users can arbitrarily combine the order of recording directions. Figure 4 is

One of the sequences is used as an example for explanation.

When the main control issues a command to record a face in a certain direction, the module starts working and returns the face status, position, posture, and other information in real time.

The user can use this information to remind the inputter to adjust the posture and position.

If successful, the module will return the result of the entry in the form of a REPLY message. If the entry is unsuccessful, the module will also return the reason for the failure.

because.

It is worth noting that when the module successfully records the first frame of the picture it receives, it will directly return the result in the form of a REPLY message.

During the recording process, you can terminate the recording through the FACE RESET instruction, and the previous recording status will also be cleared.

If the power is suddenly cut off during the recording process, the previously recorded faces will not be saved.

9 Input Instructions

The angles for face recording in various directions are described as follows:

Table 12

Entry Direction	Deflection angle
Frontal	Facing the camera, no deflection angle
up	Face tilted upward 5~55 degrees
down	Face tilted downward 5 to 55 degrees
To the right	The face is tilted 8 to 60 degrees to the right
left	The face is tilted 8 to 60 degrees to the left

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

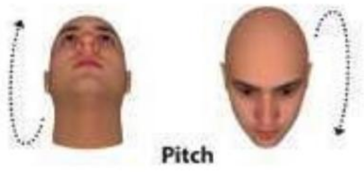


Figure 5

The left and right deflections are shown in Figure 6.

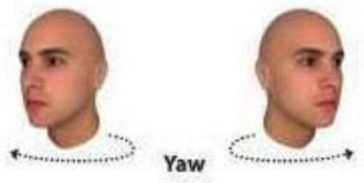


Figure 6

Please do not use the head rotation method shown in Figure 7.

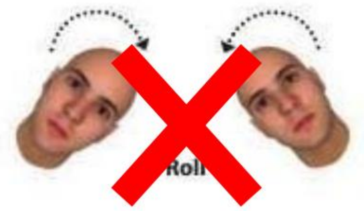


Figure 7

We recommend that users turn their heads slowly in a certain direction and keep turning during the actual input process. Do not turn your head too quickly or

The person immediately returns to the normal state.

10 Single frame recording

The main difference between single-frame recording and recording process is that single-frame recording only requires a positive face to be recorded successfully.

Complete registration five times.

11 Unlocking Process

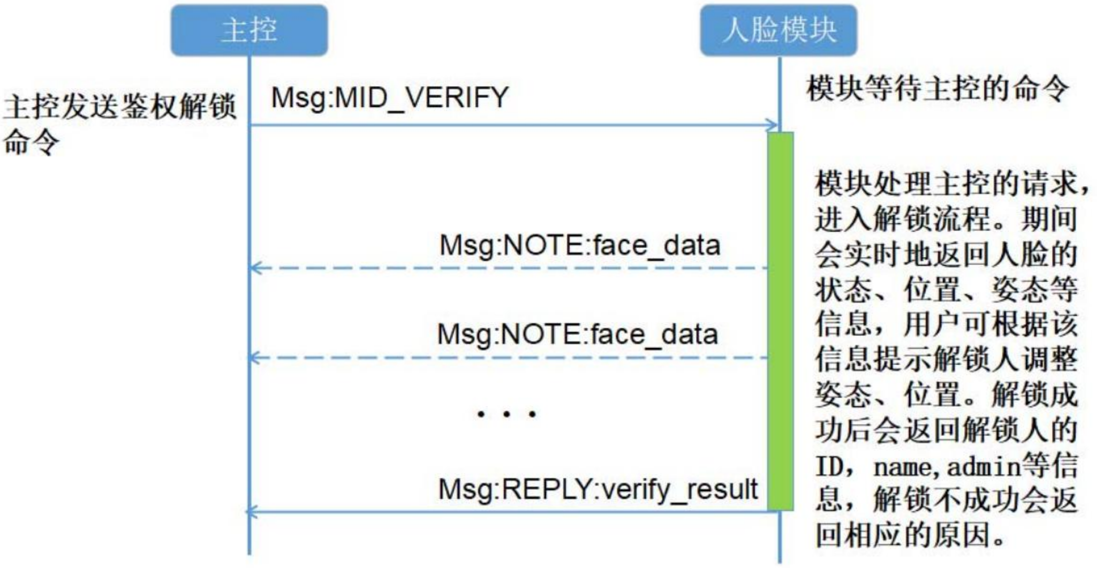

主控
人脸模块
主控发送鉴权解锁命令
Msg:MID_VERIFY
模块等待主控的命令
模块处理主控的请求，进入解锁流程。期间会实时地返回人脸的状态、位置、姿态等信息，用户可根据该信息提示解锁人调整姿态、位置。解锁成功后会返回解锁人的ID，name,admin等信息，解锁不成功会返回相应的原因。
Msg:NOTE:face_data
Msg:NOTE:face_data
• • •
Msg:REPLY:verify_result

Figure 8

12 Unlocking Process Description

The module processes the master's request and enters the unlocking process, during which it returns the face status, position, posture and other information in real time.

The unlocking person can adjust his posture and position based on this information.

If the unlocking is successful, the unlocker's ID, name, admin and other information will be returned. If the unlocking is unsuccessful, the corresponding reason will be returned.

13 Encrypted Communication Process

The encrypted communication process is shown in Figure 9.

1. The main control unit powers on the face module

2. The face module sends READY (plain text) to the main control

3. When the module is powered on for the first time, you need to call the MID_SET_RELEASE_ENC_KEY command to set the 16-byte sequence required by the private protocol.

The master controller uses random numbers to generate a random sequence and sends it to the face module (4 bytes)

4. The face module and the main control both use a custom private protocol to generate a 16-byte password based on the random sequence.

The same algorithm generates the same password. Both parties will use this password for encryption and decryption in this conversation.

5. The face module returns the status and uses a random password AES to encrypt and send the product serial number to the main control

6. The master controller decrypts and identifies the device ID. After confirming the identity, it begins processing the required instructions, such as input or unlocking.

7. The master controller uses a randomly generated password and AES encryption to encrypt the instructions and data and send them to the face module.

8. The face module decrypts the password obtained in step 4, determines the legitimacy of the instruction and processes it.

9. The face module encrypts the command processing results and data with the password decrypted in step 4 and sends it to the main control

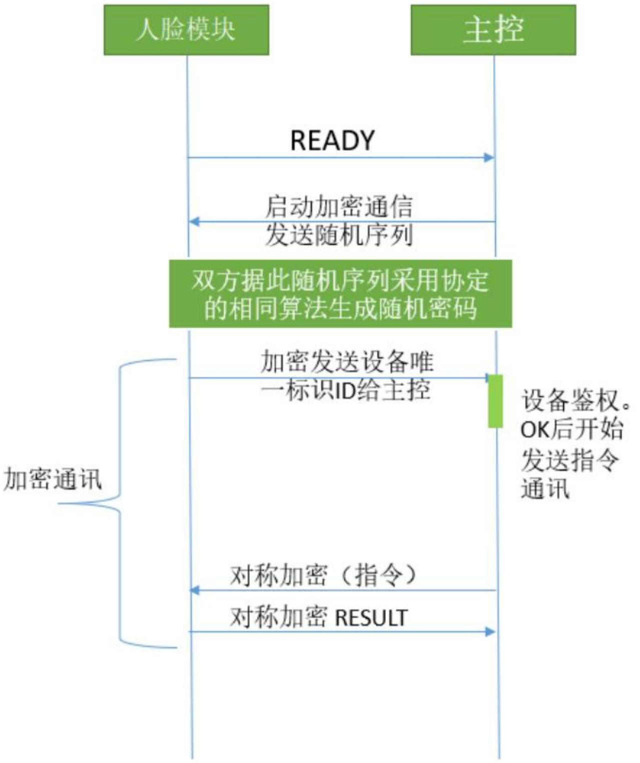


Figure 9

14 Supplementary Notes

14.1 Command Data Format

All the commands supported by the module are shown in Table 13.

The data received in the Data column is the Data part of Table 4.

The data sent in the Data column is the data portion of Table 5: MID_REPLY instruction: s_msg_reply_data.

Table 13

MsgID/Timeout(ms)	Code	Data	
		Parameters (bytes)	illustrate
MID_RESET/200	0x10	Stop all currently processed messages and the module enters the standby state After the master sends this command, the module will cancel the previously executed command (such as recording Enter, unlock, etc.), and return to STANDBY state.	
		Receive: No	
		Send: None	
MID_GETSTATUS/200	0x11	After the master sends this command, the module returns to its current state	
		Receive: No	
		Send: s_msg_reply_getstatus_data	
		status(1)	The module status includes: 0: MS_STANDBY (The module is in idle state, waiting Waiting for master control command) 1: MS_BUSY (module is in working state) 2: MS_ERROR (Module error, cannot work normally do) 3: MS_INVALID (module not initialized) 4: MS_OTA (OTA upgrade in progress)
MID_VERIFY	0x12	Authentication unlock, Timeout is the timeout for the verification instruction execution	
		Receive: s_msg_verify_data	
		pd_rightaway(1)	Indicates whether to shut down the device immediately after unlocking; Whether to power off immediately after unlocking successfully (yes: 1, no: 0) Deprecated now
		timeout(1)	Unlock timeout (unit: s) [timeout] is the unlock timeout (unit s), the default is 10s, you can send the unlock command Time setting, the timeout time can be set arbitrarily by the user The maximum value is no more than 255s.

		verify_mode(1) is reserved, currently The firmware no longer uses this parameter	Registration Mode 0: Standard (facial and palm vein) 1: Palmar vein 2: 3:
		Send: s_msg_reply_verify_data	
		user_id_heb(1)	Authenticated user's ID Only when the result is MR_SUCCESS Only user_id The user ID when recognizing the QR code is 0
		user_id_leb(1)	
		user_name(32)	Username When the module recognizes the QR code, the content here is recognized The QR code string that comes out. When the length of the QR code to be recognized is greater than 32, The length of user_name is also expanded to 256 bytes.
		admin(1)	Is this an administrator?
		unlockStatus(1)	Eyes open or closed during unlocking 200: Face verification successful 204: Face verification successful, close your eyes 250: Palm vein verification successful
MID_ENROLL	0x13 New user entry, Timeout means the registration command has timed out, and repeated registration is not allowed		
		Receive: s_msg_enroll_data	
		admin(1)	Indicates that the person entering the entry is the administrator Whether to set as administrator (yes: 1, no: 0)
		user_name(32)	Enter user name
		face_direction(1)	The user needs to enter the direction, the specific parameters are as shown in the table 12 shown
		timeout(1)	Input timeout (unit: s)
		Send: s_msg_reply_enroll_data	
		user_id_heb(1)	Registered user ID Only when the result is MR_SUCCESS Only user_id When determining duplicate registration, return the registered user ID (only available upon specific request)
		user_id_leb(1)	
		face_direction(1)	Recording status of faces in various directions

		face_data(n)	The registration direction of unidirectional registration (0x1D) is FACE_DIRECTION_RENT(0xFC) When saving the registration data here, the data format Same as server feature data conversion LIB, To use MID_TRANS_FILE_PACKET(0x90) Instructions to transfer data for registration.
MID_SNAPIMAGE	0x16	Capture pictures and store them locally	
		Receive: s_msg_snap_image_data	
		image_counts(1)	Number of captured images
		start_number(1)	The starting number of the captured image (1-20)
		Send: None	
MID_GETSAVEDIMAGE	0x17	Get the size of the image to be uploaded	
		Receive: s_msg_get_saved_image_data	
		image_number(1)	The number of the image to be transferred >0: Send MID_SNAPIMAGE command to obtain the current Current image size (bytes).
		Send: s_msg_reply_get_saved_image_data	
		image_size(4)	Size of the image to be uploaded, MSB
MID_UPLOADIMAGE	0x18	Upload the locally stored image to the main control	
		Receive: s_msg_upload_image_data	
		upload_image_offset(4)	Offset of the image to be uploaded, MSB
		upload_image_size(4)	The size of the uploaded image (maximum 4K), MSB
		Send: image_data	
		data(n)	n: The size that the master requires to upload Actual pictures
MID_ENROLL_SINGLE	0x1D	Single frame recording, need to wait for the timeout set by the host computer registration command, and allow to repeat Re-registration	
		Receive: s_msg_enroll_data	
		admin(1)	Whether to set as administrator (yes: 1, no: 0)
		user_name(32)	Enter user name

		face_direction(1)	The user needs to enter the direction, the specific parameters are as shown in the table 12 shown. In addition to the five directions of the face, there are two more parameters FACE_DIRECTION_HAND is set to FACE_DIRECTION_HAND, palm vein injection book FACE_DIRECTION_RENT(0xFC), the registration number The data is transmitted to the lock board, only the data is transmitted, and the module itself No registration is performed.
		timeout(1)	Input timeout (unit: s)
		Send: s_msg_reply_enroll_data	
			The response is the same as that of the MID_ENROLL command.
MID_DELUSER/100	0x20	Delete a registered user	
		Receive: s_msg_deluser_data	
		user_id_heb(1)	ID of the user to be deleted
		user_id_leb(1)	
		user_type(1)	User Type 0: Face 1: Palmar vein
		Send: None	
MID_DELALL	0x21	Delete all registered users	
		Receive: s_msg_del_all_data	
		type(1)	Set deletion method 0: Delete all users 1: Delete face user 2: Delete palm vein user 3: Delete ordinary users 4: Delete users within the set 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 end number of the user ID
		end_user_id_leb(1)	
		Send: None	
MID_GETUSERINFO/100	0x22	Get the information of a registered user	

		Receive: s_msg_getuserinfo_data	
		user_id_heb(1)	Registered user ID
		user_id_leb(1)	
		Send: s_msg_reply_getuserinfo_data	
		user_id_heb(1)	Registered user ID
		user_id_leb(1)	
		user_name(32)	Name of registered user
		admin(1)	Is this an administrator?
MID_GET_ALL_USERID/1000 0x24		Get the number and IDs of all registered users	
		Receive: s_msg_get_all_userid_data	
		fmt(1)	Get the number and IDs of all registered users Set the response format 0: Face user (response data format is long) 1: Face user (response data format is short) 2: Palm vein user (response data format is short)
		Send: s_msg_reply_all_userid_data, fmt=0	
		user_counts(1)	Number of registered users
		users_id[MAX_USER_COUNTS*2]	All registered user IDs, use two consecutive Bytes store an ID, with the high eight bits stored first MAX_USER_COUNTS is the maximum number of users
		Send: s_msg_reply_all_userid_data_fmt1, fmt=1,2	
		user_counts(1)	Number of registered users
		users_id[(MAX_USER_COUNTS + 7) / 8]	Registered user ID. One bit represents one user, MAX_USER_COUN TS is the maximum number of face users/palm vein users. The user_id column is 0x08 0x03. The numbers are 4, 9, and 10.
MID_ENROLL_ITG	0x26	Integrate support and expand all entry methods Need to wait for the timeout set by the host computer registration command	
		Receive: s_msg_enroll_itg	
		admin(1)	Is it set as administrator (yes:1 no:0)
		user_name(32)	Enter user name

		face_dir(1)	The user needs to enter the direction, the specific parameters are as shown in the table 12 shown Only valid when enroll_type is 0. When registering palm vein, set face_dir to FACE_DIRECTION_HAND(0xFD) will do.
		enroll_type(1)	Registration type: interactive entry or single frame entry 0: Interactive entry (same method as MID_ENROLL) Mode) 1: Single frame entry (same as MID_ENROLL_SINGLE similar method)
		enable_duplicate(1)	Is it repeated entry? 0: The same person cannot enter, but user name can be repeated. 1: The same person can be entered repeatedly, user name can also be repeated. 2: The same person can enter, but user The name cannot be repeated (only when registering RGB effect).
		timeout(1)	Input timeout (unit: s)
		reserved(3)	Reserved, will be removed later. Currently enter 0 That's it.
		Send: s_msg_reply_enroll_data	
		user_id_heb(1)	Registered user ID Only when the result is MR_SUCCESS Only user_id
		user_id_leb(1)	
		face_direction(1)	Recording status of faces in various directions
MID_GET_VERSION/1000	0x30	Get software version information	
		Receive: No	
		Send: s_msg_reply_version_data	
		version_info(32)	Version Information
MID_INIT_ENCRYPTION/1000 0x50		Set the encryption random number	
		Receive: s_msg_init_encryption_data	
		seed(4)	The master sends a random number

		mode(1)	Encryption Mode 0: No encryption 1: AES encryption 2: SINGLE encryption
		Send: s_msg_reply_init_encryption_data	
		device_id(20)	Device ID information
		Send: None	
MID_SET_RELEASE_ENC_KEY 0x52		Set the encryption key sequence for mass production, which will be saved after power failure	
		Receive: s_msg_enc_key_number_data	
		enc_key_number(16)	Encryption Sequence
		Send: None	
			In addition to the initial setup, processing starts from the second setup For failure.
MID_SET_DEBUG_ENC_KEY	0x53	Set the debug encryption key sequence, which will be lost if power is off	
		Receive: s_msg_enc_key_number_data	
		enc_key_number(16)	Encryption Sequence
		Send: None	
MID_GET_UID	0x93	Receive: No	
		Send: s_msg_reply_init_encryption_data	
			Response to the MID_INIT_ENCRYPTION command Sample
		Send: None	
MID_CAPTURE_PIC_TYPE	0x9A	Set the upload image parameters	
		Receive: s_msg_capture_pic_type	
		pic_width(2)	Image width, currently only aspect ratio is supported 640x480, 480x320, 320x240 Three resolutions
		pic_height(2)	Image width, currently only aspect ratio is supported 640x480, 480x320, 320x240 Three resolutions

		capture_type(1)	Image Type IS:0x00 RGB:0x01
		iompress_ratio(1)	Image compression ratio Compression quality 30%-80% corresponds to setting 30-80
		is_cutout_face(1)	Whether to crop the face of the picture, not supported yet
		Send: None	
MID_SET_THRESHOLD_LEVEL 0xD4		Set the algorithm security level	
		Receive: s_msg_algo_threshold_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
		Send: None	
MID_DEMOMODE/100	0xFE	Enter the demonstration mode. After the master sends this command, the module enters the demonstration mode. In the module authentication process, no feature comparison will be performed, that is, anyone can unlock the module. However, liveness detection will still be performed.	
		Receive: s_msg_demomode_data	
		enable(1)	1: enable 0: disable
		Send: None	

The face direction parameters are shown in Table 14.

When using the MID_ENROLL_SINGLE instruction, the up, down, left, and right parameters are invalid. When the parameter is set to FACE_DIRECTION_HAND, note that When the palm vein is registered and the parameter is set to FACE_DIRECTION_PICTURE, the photo data sent by MID_START_OTA is used for human identification. Face registration.

Table 14

st_face_dir (Face direction definition)	code	illustrate
FACE_DIRECTION_UP	0x10	Recording an upward-facing face
FACE_DIRECTION_DOWN	0x08	Recording a face facing downwards

FACE_DIRECTION_LEFT	0x04	Recording a left-facing face
FACE_DIRECTION_RIGHT	0x02	Recording a face facing right
FACE_DIRECTION_MIDDLE	0x01	Recording a positive face
FACE_DIRECTION_UNDEFINE	0x00	Undefined, default is forward
FACE_DIRECTION_FACE_ONLY	0x40	Enter positive face, only detect face registration method (reserved)
FACE_DIRECTION_RENT	0xFC	Single frame registration transmits registration data to the host computer
FACE_DIRECTION_HAND	0xFD	Palm vein recording
FACE_DIRECTION_PICTURE	0xFE	Remote registration of face recognition for rental housing project (received via MID_START_OTA command) After taking the picture, use this command to register the face)

The face_direction in the response of MID_ENROLL and MID_ENROLL_ITG commands indicates the face entry status. The lower 5 bits of the data are

From high to low, they represent the face direction upward, downward, left, right and forward, as shown in the figure below. 1 means the direction has been entered, 0 means the direction has been entered.

Indicates that the direction has not been entered.

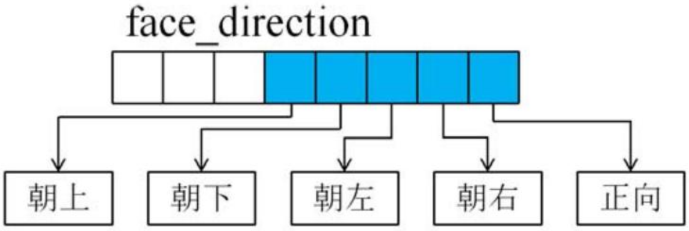


Figure 10

14.2 MID_SNAPIMAGE

This command is for capturing pictures. When the master controller issues this command and carries the number of pictures to be captured and the starting number of the captured pictures, the module

The block will immediately capture the pictures, name them in order from the starting number and save them in the local (jpg format).

The number of pictures supported for local storage is 20, and the pictures are numbered from 1 to 20. When the number sent by the master is the same as the local picture number,

Overwrite the original image.

14.3 MID_GETSAVEDIMAGE & MID_UPLOADIMAGE

The captured pictures will be stored in the local path of the module. When the main control obtains the captured pictures, it first calls the MID_GETSAVEDIMAGE command.

Get the size of the image to be uploaded, and then call MID_UPLOADIMAGE multiple times to get the image as needed.

The data upload_image_offset[4] carried in the instruction indicates the offset of the uploaded image (for example, when the offset is 0, the image will be uploaded from the

upload_image_size[4] indicates the size of the uploaded image, with a maximum support of 4000 bytes.

14.4 MID_CAPTURE_PIC_TYPE

This command is used to modify the parameters of the captured image, including resolution, image type (IR or RGB), image compression ratio, and whether to crop faces. After the command is sent, the next image captured by the module will be saved in the new format specified in the command parameter and transmitted back to the main control unit.

14.5 MID_SET_RELEASE_ENC_KEY

The master controller must send this command when it is powered on for the first time. The module will encrypt and save the data carried by the command. The data is encrypted as described in [13 Encrypted Communication Process]. The data is not lost when power is off.

14.6 MID_INIT_ENCRYPTION

The master controller sends a random number, and the module generates a password after receiving the random number, and sends a reply message according to the new password.

In addition to the random number, the structure also contains the encryption mode and the crttime array for setting the system time. The stored contents represent the number of seconds in the current time zone (1970 to present).

The structure of the random number sent by the master is as follows:

```
typedef struct {
        uint8_t seed[4];
        uint8_t mode;
        uint8_t crttime[4];
} s_msg_init_encryption_data;
```

The data structure returned after the module encryption is completed is as follows:

```
typedef struct {
        uint8_t device_id[20];
} s_msg_reply_init_encryption_data;
```

14.7 MID_SET_INTERACTIVATE

After the main control is turned on, it sends the angle range thresholds for each direction used during interactive entry. If this interface is not called, the default threshold is used for registration.

The structure sent by the master control is as follows:

```
typedef struct {
        uint8_t range_left_min;
        uint8_t range_left_max;
        uint8_t range_right_min;
        uint8_t range_right_max;
        uint8_t range_up_min;
        uint8_t range_up_max;
        uint8_t range_down_min;
        uint8_t range_down_max;
    } s_msg_interactivate_param;
```