Shenzhen Hi-Link Electronic Co., Ltd.

Fingerprint module user communication protocol

Version1.1 2024.03

Statement

The following documents contain the private information of Hi-Link (hereinafter referred to as Hi-Link). The third party may use or disclose it at will without the formal permission of Hi-Link Any information may be used without authorization or special conditions. Copying and unauthorized modification of this information without any restriction or notification are infringements.

At any time, without notifying any party, Hi-Link has the right to make changes, additions, deletions, improvements and any other changes to the company's products and services. Hi-Link does not bear any responsibility or obligation in the use of our company's products; and third parties must not infringe any patents or other intellectual property rights in their use.

All products are sold subject to the Company's terms and conditions of sale in the order acknowledgment. The company uses testing, tools, quality control and other technical means to support a certain degree of assurance that the relevant performance of the product meets the required specifications. Beyond explicit written government requirements, it is not necessary to perform all parameter testing for each product. If the product is damaged or cannot be used normally due to improper use by the customer, the customer shall bear the responsibility.

Except for the logo design of Hi-Link all other trademarks or registered trademarks belong to their respective owners.

All rights reserved. Infringement will be prosecuted.

Revision Sheet

Version	Date	Modify content		
		Chapter	Revised by	Content
1.0	2022-04-16	All		Initial version
1.1	2024-03-16	All		Modify format

Content

Catalog of drawings

Table Directory

Abbreviations and Terminology

USB: Universal Serial Bus

UART: Universal Asynchronous Receiver/Transmitter

JTAG: Joint Test Action Group

FPM: Finger Print Module

1 Hardware interface description

1) UART

a) UART default baud rate is 57600 bps, data format: 8 data bits, 1 stop bit, no parity bit;

b) The UART baud rate can be set through instructions, ranging from 9600 to 115200;

c) If the main control is MCU (3.3V), connect directly to UART_TX and UART_RX; if the main control is PC, you need to connect the RS232 level conversion device.

d) Taking into account the uncertainty in the selection of subsequent algorithm chips (hereinafter referred to as MCU) and the phenomenon of current backflow in the IO of some MCUs currently on the market; therefore, the serial port power-on sequence is now unified as follows:

①: After the lock terminal (host) receives the FPM interrupt wake-up signal, it first controls the Vmcu (MCU power supply) to power on; and then initializes the serial port. It is forbidden to initialize the serial port first and then control Vmcu to power on to avoid abnormal MCU power-on due to serial port leakage.

②: After the lock board (host) and FPM complete all serial communication, before the lock board powers off the FPM, the serial port signal line needs to be pulled down first; and then the Vmcu is controlled to power off. Avoid other unknown abnormalities such as excessive power consumption caused by serial port leakage, abnormal MCU reset, etc. after power outage.

2 Software instructions

2.1 Parameters Table

The content of the parameter table is the basic parameters for the operation of the protocol and algorithm. All work of FPM will use the contents of the parameter table, so understanding and properly setting the parameter table is crucial to how to use fingerprint module products correctly;

Table 2-1 shows the structure of the parameter table:

Table 2-1 System parameter table

Type	Serial number	Chinese name	English name	Length (bytes)	Content and default values	Comment
PART1	1	Number of registrations	EnrollTimes	2		
	2	Fingerprint template size	TempSize	2		
	3	Fingerprint database size	DataBaseSize	2	Automatically determine according to FLASH type	
PART2	4	Score level	ScoreLevel	2	18	divided into 28 levels
	5	Device address	DeviceAddress	4	0xFFFFFFFF	Can be set via commands
	6	Packet size	CFG_PktSize	2	2	These 8 registers are the system configuration table
	7	Baud rate coefficient	CFG_BaudRate	2	6	
	8	Anti-fake fingerprints	ResSpitefullmg	2	1	

	9	Sensor	FPSensorPara	2		
	10	Encryption level	SecurLevel	2	0	
	11	Registration logic	EnrollLogic	2	0	
	12	Image format	ImageFormat	2	1	
	13	Serial port delay	DelayTime	2	0	
	14	Product number	ProductSN	8	ASCII code	Device descriptor
	15	Software version number	Software version	8	ASCII code	
	16	Manufacturer's name	Manufacturer	8	ASCII code	
	17	Sensor name	SensorName	8	ASCII code	
	18	Password	Password	4	00000000H	
	19	Jtag lock flag	Jtag Lock Flag	4	00000000H	
	20	reserve		2		
	21	reserve		2		
	22	reserve		54		
PART3	23	Parameter table valid flag	Para Table Flag	2	0x1234	

- Detailed explanation of parameter table:

1) Enroll Times

Reset Value：According to FLASH

Length:2 bytes

Attributes:read/write

Purpose: When registering, set the number of entry instructions

Read command: PS_ReadSysPara, see command description for details

Setting command: PS_WriteReg, please refer to the command description for details

2) **Temp Size**

Reset Value: According to FLASH

Length: 2 bytes

Properties: Read only

Purpose: Fingerprint template size indication

Read command: PS_ReadSysPara, see command description for details

3) Data Base Size

Reset Value: According to FLASH

Length: 2 bytes

Properties: Read only

Purpose: Fingerprint library capacity indication

Read command: PS_ReadSysPara, see command description for details

4) **Score Level**

Reset Value: 0x0012

Length: 2 bytes

Properties: Read and write

Purpose: Score level indication; the system sets the comparison threshold based on this value

Read command: PS_ReadSysPara, please refer to the command description for details

Setting command: PS_WriteReg, please refer to the command description for details

Twenty-eight levels:

1：Level 0

Lowest

2: Level 1

…

27: Level 26

28: Level 27

Highest

5) Device address

Reset Value: 0xFFFFFFFF

Length: 4 bytes

Properties: read/write

Purpose: The system only receives command packets/data packets with matching addresses

Read command: PS_ReadSysPara, see command description for details

Setting command: PS_SetChipAddr, please refer to the command description for details

6) **CFG_PktSize**

Reset Value: 0x0002

Length: 2 bytes

Properties: read/write

Purpose: When sending data, the system sets the length of a single data packet based on this value

Read command: PS_ReadSysPara, see command description for details

Setting command: PS_WriteReg, see command description for details

7) **CFG_BaudRate**

Reset Value: 0x0006

Length: 2 bytes

Properties: read/write

Purpose: Determine uart baud rate = this value * 9600

Read command: PS_ReadSysPara, see command description for details

Setting command: PS_WriteReg, see command description for details

8) Anti-fake fingerprint ResSpitefullmg

Reset Value: 1

Length: 2 bytes

Properties: Read

Purpose: Used to prevent false fingerprints from merging into fingerprint templates

Read command: PS_ReadSysPara, see command description for details

Note: This parameter is turned off by default and is not writable

9) FP Sensor Para

Reset Value: According to FLASH

Length: 2 bytes

Properties: Read

Purpose: Set relevant parameters for communication between the main control chip and sensor

Read command: PS_ReadSysPara, see command description for details

Note: Not configurable.

10) Secur Level

Reset Value: 0

Length: 2bytes

Purpose: Set the module encryption level. Changes are not allowed after setting.

Read command: PS ReadINFpage, please refer to the command description for details

0: Level 0 default state. All instructions except the safety instruction set are supported.

1: Level 1 has no security algorithm and does not support security instruction sets, upload templates, download templates and download images.

2: Level 2 SM4 (ECB) does not support uploading templates, downloading templates, downloading images, precise comparison, searching for fingerprints, storing templates, automatic registration, automatic verification and cancellation instructions, and supports security instruction sets.

3: Level 3 AES (128bits, ECB), does not support uploading templates, downloading templates, downloading images, precise comparison, searching for fingerprints, storing templates, automatic registration, automatic verification and cancellation instructions, and supports security instruction sets.

4: Level 4 3DES (16bytes, ECB), does not support uploading templates, downloading templates, downloading images, precise comparison, searching for fingerprints, storing templates, automatic registration, automatic verification and cancellation instructions, and supports security instruction sets.

5~19 reserved.

20: Level 20 RSA (1024bits), does not support uploading templates, downloading templates, downloading images, precise comparison, searching for fingerprints, storing templates, automatic registration, automatic verification and cancellation instructions, and supports security instruction sets.

21: Level 21 ECC (256bits), does not support uploading templates, downloading templates, downloading images, precise comparison, searching for fingerprints, storing templates, automatic registration, automatic verification and cancellation instructions, and supports security instruction sets.

twenty two~65535 Reserved.

11) Enroll Logic

Reset Value: 0

Length: 2bytes

Properties: read/write

Purpose: Enter the logic of fingers during registration

Read command: PS ReadINFpage, please refer to the command description for details

Setting command: PS_WriteReg, see command description for details

Logical way:

0: Mode 0 is the default state, no logic.

1: Mode 1 requires no correlation between the fingers entered.

2: Mode 2 requires that the fingers entered are related.

12) Image format

Reset Value: 1

Length: 2bytes

Properties: read/write

Purpose: Image format setting when taking pictures

Read command: PS_ReadINFpage, see the command description for details

Setting command: PS_WriteReg, see command description for details

Format type:

0: Format. The original image.

1: Format 1 default state, preprocessed image.

13) Serial Delay Time

Reset Value: 0

Length: 2bytes

Properties: read/write

Purpose: When transmitting data packets through the serial port, set the time interval between packets

Read command: PS_ReadINFpage, see the command description for details

Setting command: PS_WriteReg, see command description for details

Time range: 0~255ms

14) Product SN

Reset Value: initialization value after first power-on

Length: 8bytes

Properties: Read only

Purpose: Indicate product model

Read command: PS_ReadINFpage, see the command description for details

15) Software version

Reset Value: initialization value after first power-on

Length: 8bytes

Properties: Read only

Purpose: Indicate software version number

Read command: PS_ReadINFpage, see the command description for details

16) Manufacturer name

Reset Value: initialization value after first power-on

Length: 8 bytes

Properties: Read only

Purpose: Indicate manufacturer name

Read command: PS_ReadINFpage, see the command description for details

17) Sensor name

Reset Value: initialization value after first power-on

Length: 8 bytes

Properties: Read only

Purpose: Indicate sensor name

Read command: PS_ReadINFpage, see the command description for details

18) Password

Reset Value: 0x00000000

Length: 4 bytes

Properties: read/write

Purpose: Handshake password, the system must respond only after the password passes

Read command: PS_ReadINFpage, see the command description for details

Setting command: PS_SetPwd, please refer to the command description for details

19) JTAG lock flag

Reset Value: 0x00000000

Length: 4 bytes

Properties: Read only

Purpose: Writing a specific value on first power-up will close the JTAG port

Read command: PS_ReadINFpage, see the command description for details

20) Para Table Flag

Reset Value: 0x1234

Length: 2 bytes

Properties: Read only

Purpose: If the value of this field is 0x1234, it means that the parameter table has been initialized; if the value of this field is 0x0204, it means that the system only initializes the PART1 part of the parameter table; if this field is other values, the system will initialize the parameter table .

Read command: PS_ReadINFpage, see the command description for details

2.2 Number of registrations and fingerprint template size

There are differences in the number of registrations and fingerprint template sizes between fingerprint module products, such asTable2-1As shown in, the relevant information can be read through the PS_ReadSysPara instruction.

2.3 Power-on handshake signal

After the fingerprint module product is successfully initialized after power-on, it will send a 0x55 signal through Uart. While the host is waiting for FPM to initialize, it can enter the working state in advance by receiving the handshake signal.

If the host does not use the handshake signal to determine whether the FPM has been initialized, it is recommended that the FPM be delayed for 200ms after power-on before communicating to prevent the FPM from being initialized and causing abnormal communication between the host and the FPM.

3 Detailed explanation of command format

FPM is always in a slave position, and the master needs to use different instructions to let the module complete various functions. The main control instructions, module responses and data exchange are all carried out in accordance with the prescribed format of data packets. The master must encapsulate the instructions or data to be sent in the following format, and must also parse the received data packets in the following format.

3.1 Command packet/data packet format

Instructions/data packets are divided into three categories:

Package ID=01: command package.

Packet ID=02: Data packet, and there are subsequent packets.

Packet ID=08: The last data packet, that is, the end packet.

All data packets must add a header: 0xEF01, and the device address defaults to 0xFFFFFFFF.

●01 Command package format:

Table 3-1 Command format

Name	Header	Device address	Package ID	Packet length	Instruction	Parameter 1	…	Parameter N	Checksum
Number of bytes	2 bytes	4 bytes	1 byte	2 bytes	1 byte				2 bytes
Content	0xEF01	0xFFFFFFFF	01	N=					

●02 Data packet format:

Table 3-2 Packet format

Name	Header	Device address	Package ID	Packet length	Data	Checksum

Number of bytes	2 bytes	4bytes	1 byte	2 bytes	N bytes	2 bytes
Content	0xEF01	0xFFFFFFFF	02			

●08 End packet format:

Table 3-3 End packet format

Name	Header	Device address	Package ID	Packet length	Data	Checksum
Number of bytes	2 bytes	4bytes	1 byte	2 bytes	N bytes	2 bytes
Content	0xEF01	0xFFFFFFFF	08			

◆The data packet cannot enter the execution process alone and must follow the command packet or response packet.

◆The format of the downloaded or uploaded data packets is the same.

◆Packet length=The total number of bytes from the packet length to the checksum (command, parameter, or data), including the checksum but excluding the packet length itself.

◆The checksum is the sum of all bytes from the packet ID to the checksum, including the packet ID but not the checksum. Any carry exceeding 2 bytes is ignored.

◆The device address is the default 0xffffffff before it is generated. Once the master generates the device address through instructions, all data packets must be sent and received according to the generated address.

◆For multi-byte high byte first, low byte last (such as 2bytes 00 06 means 0006, not 0600)

3.2 Command response

The response is to report the relevant command execution status and results to the main control. The response contains parameters and can be followed by subsequent data packets. The main control is only in only after receiving the response packet can the packet receipt status and instruction execution status be confirmed.

- Response packet format:

Table 3-4 Response packet format

Name	Header	Device address	Package ID	Packet length	Confirmation code	Return parameters	Checksum
Number of bytes	2bytes	4 bytes	1 byte	2 bytes	1 byte	N bytes	2 bytes
Content	0xEF01	0xFFFFFFFF	07				

- Confirmation code definition:

00H: Indicates that the instruction execution is completed or OK;

01H: Indicates data packet reception error;

02H: Indicates there is no finger on the sensor;

03H: Indicates failure to enter fingerprint image;

04H: Indicates that the fingerprint image is too dry and too light to generate features;

05H: Indicates that the fingerprint image is too wet and muddy to generate features;

06H: Indicates that the fingerprint image is too chaotic to produce features;

07H: Indicates that the fingerprint image is normal, but there are too few feature points (or the area is too small) to generate features;

08H: Indicates fingerprint mismatch;

09H: Indicates that no fingerprint was found;

0aH: Indicates that feature merging failed;

0bH: Indicates that the address serial number exceeds the range of the fingerprint database when accessing the fingerprint database;

0cH: Indicates an error or invalidity in reading the template from the fingerprint database;

0dH: Indicates failure to upload features;

0eH: Indicates that the module cannot receive subsequent data packets;

0fH: Indicates failure to upload the image;

10H: Indicates failure to delete the template;

HH: Indicates that clearing the fingerprint database failed;

12H: Indicates that it cannot enter the low power consumption state;

13H: Indicates that the password is incorrect;

15H: Indicates that there is no valid original image in the buffer and the image cannot be generated;

16H: Indicates that the online upgrade failed;

17H: Indicates residual fingerprints or the finger has not moved between two collections;

18H: Indicates an error in reading and writing FLASH;

19H: Random number generation failed;

1aH: Invalid register number;

1bH: Register setting content error number;

1cH: Wrong specification of notepad page number;

1dH: Port operation failed;

1eH: Automatic registration (enroll) failed;

1fH: The fingerprint database is full;

20H: Device address error;

21H: The password is incorrect;

22H: The fingerprint template is not empty;

23H: The fingerprint template is empty;

24H: The fingerprint database is empty;

25H: The number of entries is set incorrectly;

26H: timeout;

27H: Fingerprint already exists;

28H: Fingerprint features are related;

29H: Sensor operation failed;

2AH: Module information is not empty;

2BH: Module information is empty;

2CH: OTP operation failed;

2DH: Key generation failed;

2EH: The secret key does not exist;

2FH: Security algorithm execution failed;

30H: The encryption and decryption results of the security algorithm are incorrect;

31H: Function and encryption level do not match

32H: The secret key is locked

33H: The image area is small

34H: Image not available

35H: Illegal data

36H：Reserve

3.3 Business instruction set

3.3.1 Universal instruction set

3.3.1.1 Verify using PS_Get Image to get the image

Function description: When verifying fingerprints, the finger is detected, and after detection, the fingerprint image is recorded and stored in the image buffer. The confirmation code returned means: successful entry, no fingers, etc.

Input parameters:none

Return parameters:confirmation word

Instruction code:01H

- Command package format:

Table 3-5 Command packet format

Header	Device address	Package ID	Packet length	Script code	Checksum
2bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	0xFFFFFFFF	01H	0003H	01H	0005H

- Response packet format:

Table 3-6 Response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	0xFFFFFFFF	07H	0003H	xxH	sum

Note: Confirmation code = 00H means the image acquisition is successful;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 02H means there is no finger on the sensor;

sum refers to the checksum.

3.3.1.2 Generate feature PS_GenChar

- Function Description:Generate a fingerprint feature file from the original image in the image buffer and store it in the template buffer.
- Input parameters:BufferID (positive integer)
- Return parameters:confirmation word
- Instruction code:02H
- Command package format:

Table 3-7 Generate feature instruction package format

Header	Device address	Package ID	Packet length	Script code	Buffer number	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes
0xEF01	0xFFFFFFFF	01H	0004H	02H	BufferID	Sum

Note: During the registration process, BufferID indicates the location where the extracted features are stored in the buffer; in other cases, BufferID has corresponding default values.

●Command package format:

Table 3-8 Generate feature command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	0xFFFFFFFF	07H	0003H	xxH	Sum

Note: Confirmation code = 00H means the feature generation is successful;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 06H means that the fingerprint image is too messy to generate features;

Confirmation code = 07H means the fingerprint image is normal, but there are too few feature points to generate features;

Confirmation code = 08H means there is no correlation between the current fingerprint feature and the previous feature; (this function is turned off by default)

Confirmation code = 0aH means the merge failed;

Confirmation code = 15H means that there is no valid original image in the image buffer and the image cannot be generated;

Confirmation code = 28H means there is a correlation between the current fingerprint feature and the previous feature; (this feature is turned off by default)

sum refers to the checksum.

3.3.1.3 Accurate comparison PS_Match

- ●Function Description:Accurately compare feature files or templates in the template buffer. This function is supported when the encryption level is set to 0 or 1 in Table 2-1.
- ●Input parameters:none
- ●Return parameters:Confirm word, score
- ●Instruction code:03H
- ●Command package format:

Table 3-9 Accurately compare two fingerprint feature instruction package formats

Header	Device address	Package ID	Packet length	Script code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	0xFFFFFFFF	01H	0003H	03H	0007H

- ●Response packet format:

Table 3-10 Accurately compare two signature command response packet formats

Header	Device address	Package ID	Packet length	Confirmation code	Score	Checksum
2bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes	2 bytes
0xEF01	0xFFFFFFFF	07H	0005H	xxH	xxH	sum

Note: Confirmation code = 00H means fingerprint matching;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 08H means the fingerprints do not match;

Confirmation code = 31H means the function does not match the encryption level;

sum refers to the checksum.

3.3.1.4 Search fingerprint PS_Search

●Function Description:Search the whole or part of the fingerprint library with signature files in the template buffer. If found, the page number is returned. likeTable2-1This feature is supported when the medium encryption level is set to 0 or 1.

●Input parameters:BufferID (default is 1), StartPage (start page), PageNum (number of pages)

● Return parameters:Confirmation word, page number (matching fingerprint template), score

●Instruction code:04H

●Command package format:

Table 3-11 Search fingerprint instruction packet format

Header	Device address	Package ID	Packet length	Script code	Buffer number	Parameter	Parameter	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes	2 bytes
0xEF01	0xFFFFFFFF	01H	0008H	04H	BufferID	StartPage	PageNum	Sum

Note: BufferID defaults to 1, and the entire or part of the fingerprint library is searched with the fingerprint template in the template buffer.

● Response packet format:

Table 3-12 Search fingerprint command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Page number	Score	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes	2 bytes	2 bytes

0xEF0l	0xFFFFFFFF	07H	07H	xxH	PageID	MatchScore	Sum

Note: Confirmation code = 00H means searched;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 09H means not found; at this time, the page number and score are 0;

Confirmation code = 17H means residual fingerprints or the finger has not moved between two collections;

Confirmation code = 18H means there is an error writing to FLASH;

Confirmation code = 31H means the function does not match the encryption level;

Confirmation code = 35H indicates illegal data;

sum refers to the checksum.

3.3.1.5 Merge feature PS_RegModel

- Function description: Fusion of feature files generates a template, and the result is stored in the template buffer.
- Input parameters: none
- Return parameters: confirmation word
- Instruction code: 05H
- Command package format:

Header	Device address	Package ID	Packet length	Script code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	0xFFFFFFFF	01H	0003H	05H	0009H

Table 3-13 Merge feature command packet format

●Command package format:

Table 3-14 Merge feature command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	0xFFFFFFFF	07H	0003H	xxH	Sum

Note: Confirmation code = 00H means the merge is successful;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 0aH means the merge failed;

sum refers to the checksum.

3.3.1.6 Store template PS_StoreChar

- ●Function Description:Save the template file in the template buffer to the flash database location of PageID number. This function is supported when the encryption level is set to 0 or 1 in Table 2-1.
- ●Input parameters:BufferID (default is 1), PageID (fingerprint library location number)
- ●Return parameters:confirmation word
- ●Instruction code:06H
- ●Command package format:

Table 3-15 Save template command package format

Header	Device address	Package ID	Packet length	Script code	Buffer number	Location number	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes
0xEF01	0xFFFFFFFF	01H	0006H	06H	BufferID	PageID	Sum

Note: BufferID defaults to 1. PageID starts from 0 by default.

●Response packet format:

Table 3-16 Save template command response packet format

Header	Device address	Package ID	Packet length	Confirmati on code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	0xFFFFFF FF	07H	0003H	xxH	sum

Note: Confirmation code = 00H indicates successful storage;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 0bH means that the PageID is outside the range of the fingerprint database;

Confirmation code = 18H means there is an error writing to FLASH;

Confirmation code = 31H means the function does not match the encryption level;

Confirmation code = 35H indicates illegal data;

sum refers to the checksum.

3.3.1.7 Read template PS_LoadChar

●Function Description:Read the fingerprint template with the specified ID number in the flash database into the template buffer.

●Input parameters:BufferID (default is 2), PageID (fingerprint library template number)

●Return parameters:confirmation word

●Instruction code:07H

●Command package format:

Table 3-17 Read template command packet format

Header	Device address	Package ID	Packet length	Script code	Buffer number	Page number	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes
0xEF0l	0xFFFFFFFF	01H	0006H	07H	BufferID	PageID	sum

Note: BufferID defaults to 2.

●Response packet format:

Table 3-18 Read template command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	0xFFFFFFFF	07H	0003H	xxH	sum

Note: Confirmation code = 00H indicates successful reading;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 0cH means there is an error in reading or the template is invalid;

Confirmation code = 0bH means that the PageID is outside the range of the fingerprint database;

Confirmation code = 18H means there is an error writing to FLASH;

Confirmation code = 35H indicates illegal data;

sum refers to the checksum.

3.3.1.8 Upload template PS_UpChar

●Function Description:Upload the template file saved in the template buffer to the main control. This function is supported when the encryption level is set to 0 in Table 2-1.

- ●Input parameters:BufferID (default)
- ●Return parameters:confirmation word
- ●Instruction code:08H
- ●Command package format:

Table 3-19 Upload template instruction package format

Header	Device address	Package ID	Packet length	Script code	Buffer number	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes
0xEF01	0xFFFFFFFF	01H	0004H	08H	BufferID	sum

Note: For templates collected and generated from sensors, the BufferID defaults to 1 when uploaded; for templates loaded from Flash, the BufferID defaults to 2 when uploaded.

- ●Response packet format:

Table 3-20 Upload feature or template command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	0xFFFFFFFF	07H	0003H	xxH	sum

Note: Confirmation code = 00H means that the data packet will be sent later;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 0dH means the instruction execution failed;

Confirmation code = 31H means the function does not match the encryption level;

sum refers to the checksum.

- ●Subsequent packets are sent after the reply.

Table 3-21 UART upload signature or template packet format

Header	Device address	Package ID	Packet length	data	Checksum
2 bytes	4 bytes	1 byte	2 bytes	N bytes	2 bytes
0xEF0l	0xFFFFFFFF	xxH	xxH	xxH	sum

Note: Package ID=02: data package, and there are subsequent packages.

Packet ID=08: The last data packet, that is, the end packet.

When the UART uploads a characteristic or template data packet, it is sent in packets according to the preset length.

3.3.1.9 Download template PS_DownChar

- ●Function Description:The master downloads the template to a template buffer of the module. This function is supported when the encryption level is set to 0 in Table 2-1.
- ●Input parameters:BufferlD (default 1)
- ●Return parameters:confirmation word
- ●Instruction code:09H
- ●Command package format:

Table 3-22 Download template instruction package format

Header	Device address	Package ID	Packet length	Script code	Buffer number	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes
0xEF01	0xFFFFFFFF	01H	0004H	09H	BufferlD	sum

Note: BufferID defaults to 1.

- ●Response packet format:

Table 3-23 Download template command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF0l	0xFFFFFFFF	07H	0003H	xxH	sum

Note: Confirmation code = 00H means subsequent data packets can be received;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 0eH means that subsequent data packets cannot be received;

Confirmation code = 31H means the function does not match the encryption level;

sum refers to the checksum.

●Receive subsequent packets after the reply:

Table 3-24 UART Download Characteristic or Template Packet Format

Header	Device address	Package ID	Packet length	data	Checksum
2 bytes	4 bytes	1 byte	2 bytes	N bytes	2 bytes
0xEF0l	0xFFFFFFFF	xxH	xxH	xxH	sum

Note: Package ID=02: data package, and there are subsequent packages.

Packet ID=08: The last data packet, that is, the end packet.

When UART downloads feature or template data packets, it receives them in packets according to the preset length.

3.3.1.10 Delete template PS_DeletChar

- Function Description:Delete N fingerprint templates starting from the specified ID number in the flash database.
- Input parameters:PageID (fingerprint database template number), N (number of deleted templates).
- Return parameters:confirmation word
- Instruction code:0CH
- Command package format:

Table 3-25 Delete template instruction package format

Header	Device address	Package ID	Packet length	Script code	page number	Delete number	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes	2 bytes	2 bytes
0xEF0l	0xFFFFFFFF	01H	0007H	0cH	PagelD	N	sum

- Response packet format:

Table 3-26 Delete template command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF0l	0xFFFFFFFF	07H	0003H	xxH	sum

Note: Confirmation code = 00H means the template is deleted successfully;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 10H means template deletion failed;

sum refers to the checksum.

3.3.1.11 Clear fingerprint database PS_Empty

- Function Description:Delete all fingerprint templates in the flash database.
- Input parameters:none
- Return parameters:confirmation word

- Instruction code:0DH
- Command package format:

Table 3-27 Clear fingerprint library command package format

Header	Device address	Package ID	Packet length	Script code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	0xFFFFFFFF	01H	0003H	0dH	0011H

- Response packet format:

Table3-28 Clear fingerprint database command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF0l	0xFFFFFFFF	07H	0003H	xxH	sum

Note: Confirmation code = 00H means clearing successfully;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 11H means clearing failed;

sum refers to the checksum.

3.3.1.12 Write system register PS_WriteReg

- Function Description:Write module register.Fingerprint module product user manual
- Input parameters:Register serial number, content
- Return parameters:confirmation word
- Instruction code:0EH
- Command package format:

Table 3-29 Write system register command packet format

Header	Device address	Package ID	Packet length	Script code	Register serial number	content	Checksu m
2 bytes	4bytes	1 byte	2 bytes	1 byte	1byte	1byte	2 bytes
0xEF0l	0xFFFFFF FF	01H	0005H	0eH	xxH	xxH	sum

●Response packet format:

Table 3-30 Write system register command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF0l	0xFFFFFFF F	07H	0003H	xxH	sum

Note 1: Confirmation code = 00H means OK;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 18H means there is an error in reading and writing FLASH;

Confirmation code = laH means the register serial number is incorrect;

Confirmation code = lbH indicates the error number of the register setting content;

sum refers to the checksum.

Note 2: When the write system register (PS_WriteReg) instruction is executed, the response is first made according to the original configuration. After the response, the system settings are modified and the configuration is recorded in FLASH.

Table 3-31 Register configuration table

Register number	Register name	Content description
0	Serial port delay	0~255ms
1	Number of registrations	EnrollTimes
2	Image format register	0: format 0 1: format 1
3	Register logic register	0: mode 01: mode 12: mode 2
4	Baud rate control register	Multiples N of 9600 (0<N<13)
5	Comparison threshold register	1:level 0 2： level 1 … 27: level 26 28: level 27
6	Packet size register	0: 32 bytes 1:64 bytes 2: 128 bytes 3: 256 bytes
7	encryption level register	0: level 0 1:level 1 2: level 2 3: level 3 4: level 4 5~19: Reserved 20: level 20 21： level 21 22-255: Reserved
8	Anti-fake fingerprint register	reserved

9	Sensor parameters	reserved Note: Please do not set it arbitrarily.

3.3.1.13 Read module basic parameters PS_ReadSysPara

- ●Function Description:Read the basic parameters of the module (baud rate, packet size, etc.). The first 16 bytes of the parameter table store the basic communication and configuration information of the module, which are called the basic parameters of the module.
- ●Input parameters:none
- ●Return parameters:Confirmation word, basic parameters (16 bytes)
- ●Instruction code:0FH
- ●Command package format:

Table 3-32 Read system basic parameter command packet format

Header	Device address	Package ID	Packet length	Script code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF0l	0xFFFFFFFF	01H	0003H	0fH	0013H

- ●Response packet format:

Table 3-33 Read system basic parameter command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Basic parameter list	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	16 bytes	2 bytes
0xEF0l	0xFFFFFFFF	07H	13H	xxH	structure seeTable3-34	sum

Note: Confirmation code = 00H means OK;

Confirmation code = 01H means there is an error in receiving the package;

sum refers to the checksum.

Table 3-34 System basic parameter list

Name	Content description	Offset (bytes)	Size (bytes)
Number of registrations	When registering, enter the number of times	0	2
Fingerprint template size	Fingerprint template size	2	2
Fingerprint database size	Fingerprint database capacity	4	2
Score level	Score level code (1~28)	6	2
Device address	32-bit device address	8	4
Packet size	Packet size code: 0：32bytes 1: 62 bytes 2: 128 bytes 3: 256 bytes	12	2
Baud rate setting	N (baud rate 9600*N bps)	14	2

3.3.1.14 Read parameter page PS_ReadlNFpage

- Function Description:Read the parameter page where the FLASH Information Page is located (512 bytes).
- Input parameters:none
- Return parameters:confirmation word
- Instruction code:16H
- Command package format:

Table 3-35 Read flash information page command packet format

Header	Device address	Package ID	Packet length	Script code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF0l	0xFFFFFFFF	01H	0003H	16H	001Ah

●Response packet format:

Table 3-36 Read flash information page command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF0l	0xFFFFFFFF	07H	0003H	xxH	sum

Note: Confirmation code = 00H means that the data packet will be sent later;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 0DH means the instruction execution failed;

sum refers to the checksum.

●Subsequent packets are sent after the reply.

Table 3-37 UART upload information page packet format

Header	Device address	Package ID	Packet length	data	Checksum
2 bytes	4 bytes	1 byte	2 bytes	N bytes	2 bytes
0xEF0l	0xFFFFFFFF	xxH	xxH	xxH	sum

Note: Package ID=02: data package, and there are subsequent packages.

Packet ID=08: The last data packet, that is, the end packet.

When UART uploads parameter data packets, it is sent in packets according to the preset length.

3.3.1.15 Erase code PS_BurnCode

●Function Description:The main control sends an erasure code command, and the module will enter the upgrade mode after responding.

●Input parameters:Upgrade mode (default 1)

● Return parameters:confirmation word

●Instruction code:1AH

●Command package format:

Table 3-38 Programming on-chip FLASH command packet format

Header	Device address	Package ID	Packet length	Script code	Upgrade mode	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	1 bytes	2 bytes
0xEF0l	0xFFFFFFFF	01H	0004H	1Ah		sum

Note: Upgrade mode: default is 1;

●Response packet format:

Table 3-39 Programming on-chip FLASH command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF0l	0xFFFFFFFF	07H	0003H	xxH	sum

Note: Confirmation code = 00H means that subsequent data packets can be received;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 0EH means that subsequent data packets cannot be received;

sum refers to the checksum.

3.3.1.16 Read the number of valid templates PS_ValidTempleteNum

- Function Description:Read the number of valid templates.
- Input parameters:none
- Return parameters:Confirmation word, ValidN (number of valid templates)
- Instruction code:1DH
- Command package format:

Table 3-40 Read valid template number command packet format

Header	Device address	Package ID	Packet length	Script code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	0xFFFFFFFF	01H	0003H	1DH	0021H

- Response packet format:

Table 3-41 Read valid template number command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Number of valid templates	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes	2 bytes
0xEF0l	0xFFFFFFFF	07H	05H	xxH	ValidN	sum

Note: Confirmation code = 00H indicates successful reading;

Confirmation code = 01H means there is an error in receiving the package;

sum refers to the checksum.

3.3.1.17 Read index table PS_ReadlndexTable

- Function Description:Read the index table of the entry template.

- Input parameters:Index table page number, page number 0, 1, 2, 3... corresponds to the template index from 0-256, 256-512, 512-768, 768-1024..., each 1 bit represents a template, 1 represents the template of the corresponding storage area Already entered, 0 means not entered.
- Return parameters:Confirmation word, index table information
- Instruction code:1FH
- Command package format:

Table 3-42 Read index table command packet format

Header	Device address	Package ID	Packet length	Script code	page number	Checksum
2 bytes	4bytes	1 byte	2bytes	1 byte	1 byte	2 bytes
0xEF0l	0xFFFFFFFF	01H	0004H	1FH	xxH	xxxxH

- Response packet format:

Table 3-43 Read index table command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Index information	Checksum
2 bytes	4bytes	1byte	2bytes	1byte	32 bytes	2bytes
0xEF0l	0xFFFFFFFF	07H	0023H	xxH	Index	sum

Note: Confirmation code = 00H means OK;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 0BH means that the address serial number exceeds the range of the fingerprint database when asking the fingerprint database;

sum refers to the checksum.

3.3.1.18 Register to get the image PS_GetEnrolllmage

- Function Description:When registering a fingerprint, the finger is detected, and after detection, the fingerprint image is recorded and stored in the image buffer. Return confirmation code.
- Input parameters:none
- Indicates: successful entry, no fingers, etc.
- Return parameters:confirmation word
- Instruction code:29H
- Command package format:

Table 3-44 Input image command package format

Header	Device address	Package ID	Packet length	Script code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF0l	0xFFFFFFFF	01H	0003H	29H	002DH

- Response packet format:

Table 3-45 Input image command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF0l	0xFFFFFFFF	07H	0003H	xxH	sum

Note: Confirmation code = 00H means the image acquisition is successful;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 02H means there is no finger on the sensor;

sum refers to the checksum.

3.3.1.19 Sleep Command PS_Sleep

●Function Description:Set the sensor to enter sleep mode

●Input parameters:none

●Return parameters:confirmation word

●Instruction code:33H

●Command package format:

Table 3-46 Read index table command packet format

Header	Device address	Package ID	Packet length	Script code	Checksum
2 bytes	4 bytes	1 byte	2bytes	1 byte	2bytes
0xEF0l	0xFFFFFFFF	01H	0003H	33H	0037H

●Response packet format:

Table 3-47 Read index table command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1byte	2bytes	1byte	2bytes
0xEF0l	0xFFFFFFFF	07H	03	xxH	sum

Note: Confirmation code = 00H means the sleep setting is successful.

Confirmation code = 01H means there is an error in receiving the package.

Confirmation code = 29H means sensor operation failed.

sum refers to the checksum.

3.3.2 Module instruction set

3.3.2.1 Cancel command PS_Cancel

- Function Description:Cancel automatic registration template and automatic fingerprint verification. This function is supported when the encryption level is set to 0 or 1 in Table 2-1.
- Input parameters:none
- Return parameters:confirmation word
- Instruction code:30H
- Command package format:

Table 3-48 Read index table command packet format

Header	Device address	Package ID	Packet length	Script code	Checksum
2 bytes	4 bytes	1 byte	2bytes	1 byte	2bytes
0xEF01	0xFFFFFFFF	01H	0003H	30H	xxxxH

- Response packet format:

Table 3-49 Read index table command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1byte	2bytes	1byte	2bytes
0xEF01	0xFFFFFFFF	07H	03	xxH	sum

Note: Confirmation code = 00H means the setting is canceled successfully.

Confirmation code = 01H means the cancellation of the setting failed.

Confirmation code = 31H means the function does not match the encryption level;

sum refers to the checksum.

3.3.2.2 Automatic registration template PS_AutoEnroll

- Function Description:One-stop fingerprint registration, including functions such as fingerprint collection, feature generation, template combination, and template storage. likeTable2-1This feature is supported when the medium encryption level is set to 0 or 1.
- Input parameters:ID number, number of entries, parameters
- Return parameters:Confirmation word, parameter
- Instruction code:31H
- Command package format:

Table 3-50 Automatic registration template instruction package format

Header	Device address	Package ID	Packet length	Script code	ID number	Number of entries	parameter	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1byte	2 bytes	1byte	2byte	2 bytes
0xEF0l	0xFFFFFFFF	01H	0008H	31H	xxxxH	xxH	xxH	SUM

- Auxiliary instructions:

ID number: high byte first, low byte last. For example, if fingerprint No. 1 is entered, it will be 00 01H.

Number of entries: 1byte, if entered 2 times, it is 02H, if it is entered 4 times, it is 04H.

Parameter: The lowest bit is bit.

1) bit0: Image backlight control bit, 0-LED is always on, 1-LED is off after successfully acquiring the image;

2) bit1: Image preprocessing control bit, 0-disable preprocessing, 1-enable preprocessing;

3) bit2: During the registration process, whether the module is required to return to the current status in key steps, 0-return is required, 1-return is not required;

4) bit3: Whether it is allowed to overwrite the ID number, 0-not allowed, 1-allowed;

5) bit4: Allow repeated fingerprint registration control bit, 0-allowed, 1-not allowed;

6) bit5: When registering, during multiple fingerprint collection processes, whether the finger is required to leave before entering the next fingerprint image collection, 0-required to leave; 1-not required to leave;

7) bit6~bitl5: reserved.

●Response packet format:

Table 3-51 Automatic registration template command normal process response packet format

					Parameter 2 bytes			
Header	Device address	Package ID	Packet length	Confirmation code	Parameter 1	Parameter 2	Checksum	Remark
2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	1 byte	2 bytes	
0xEF0l	0xFFFFFFFF	07H	5	xxH	0H	0H	sum	Instruction legality check: legitimate/..
0xEF0l	0xFFFFFFFF	07H	5	xxH	01H	1	sum	Picture results: success/time out
0xEF0l	0xFFFFFFFF	07H	5	xxH	02H	1	sum	Generate feature results:

0xEF01	0xFFFFFFFF	07H	5	xxH	03H	1	sum	fingers away, The first entry is successful: success/time out
						…		
0xEF01	0xFFFFFFFF	07H	5	xxH	01H	n	sum	Picture results: success/time out
0xEF01	0xFFFFFFFF	07H	5	xxH	02H	n	sum	Generate feature results: success failure
0xEF01	0xFFFFFFFF	07H	5	xxH	04H	F0H	sum	Merge templates
0xEF01	0xFFFFFFFF	07H	5	xxH	05H	F1H	sum	Registered for testing
0xEF01	0xFFFFFFFF	07H	5	xxH	06H	F2H	sum	Template storage results

●Confirmation code, return value of parameter 1 and parameter 2

Table 3-52 Auto-registration template response packet interpretation cheat sheet

Confirmation code	Definition	Parameter 1	Definition	Parameter 2	Definition
00H	success	00H	Fingerprint legality	00H	Fingerprint legality detection
01H	fail	01H	Get image	F0H	Merge templates
07H	Failed to generate features	02H	Production characteristics	F1H	Check if the finger is registered

0aH	Failed to merge templates	03H	Judgment finger leaves	F2H	Store template
0bH	ID number out of range	04H	Merge templates	n	Currently entering the nth number of times
18H	Error reading and writing FLASH	05H	Registration inspection		
1fH	The fingerprint database is full	06H	Store template		
22H	Fingerprint template is not empty				
25H	The number of entries is incorrectly set				
26H	time out				
27H	Fingerprint already exists				
31H	Functions and encryption levels do not match;				
35H	illegal data				

●Instruction description:

1) If the specified ID number is invalid, the confirmation code, parameter 1 and parameter 2 are returned (the following is directly described as return): 0b 00 00H. Legality check:

■If the specified ID number is invalid, return: 0b 00 00H.

■If the number of entries is incorrectly configured, 25 00 00H will be returned. When the fingerprint is not covered, if the fingerprint database is full, 1F 00 00H will be returned;

■If the template with the specified ID number already exists, 22 00 00H will be returned.

■If the command legality check is successful, 00 00 00H will be returned and the first fingerprint entry will be entered.

2) Wait for successful image acquisition (return 00 01 0nH).

3) Wait for successful feature generation (00 02 0nH). If it fails (07 02 0nH), wait for successful image acquisition again.

4) Wait for the finger to leave. The first entry is successful (00 03 0nH). After the finger leaves, jump to step 2 and enter the next cycle until n is the set number of entries. Note: If the entry process is set so that the finger does not need to be removed, then it will directly return to the first successful entry and jump to step 2; the last time the fingerprint is collected, there is no response that the finger leaves the entry successfully.

5) Synthesize template, combine the previously acquired finger features into a finger template, return 00 04 F0H if successful, and 0A 04 F0H if failed.

6) Fingerprint duplicate check refers to matching the newly entered finger with the already stored finger (turn on or off this function by setting parameter bit4). If there are the same fingerprints, return to 27 05 F1H to end the process; if there are no same fingerprints , then return 00 05 F1H.

7) Register the template data. If the storage fails, it will return 01 06 F2H, ending the process; if it succeeds, it will return 00 06 F2H.

8) If the PS_Cancel command is received, terminate the command and return a response.

3.3.2.3 Automatically verify fingerprint PS_Autoldentify

- Function Description:Automatic fingerprint collection includes functions such as acquiring images, generating features, and searching for fingerprints. This function is supported when the encryption level is set to 0 or 1 in Table 2-1.
- Input parameters:Score level, ID number
- Return parameters:Confirmation word, page number (match fingerprint template)
- Instruction code:32H
- Command package format:

Table 3-53 Automatically verify fingerprint instruction packet format

Header	Device address	Package ID	Packet length	Script code	Score level	ID number	parameter	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes	2 bytes
0xEF01	0xFFFFFFFF	01H	0008H	32H	xxH	xxxxH	xxxxH	xxxxH

●Auxiliary instructions:

ID number: 2byte, big endian mode. For example, if fingerprint No. 1 is entered, it will be 0001H. If the ID number is 0xFFFF, then 1: N search will be performed; otherwise, 1:1 matching will be performed.

Parameter: The lowest bit is bit. .

1) bit0: Image backlight control bit, 0-LED is always on, 1-LED goes off after successfully acquiring the image;

2) bitl: image preprocessing control bit, 0-disable preprocessing, 1-enable preprocessing;

3) bit2: During the registration process, whether the module is required to return to the current status in key steps, 0-return is required, 1-return is not required;

4) bit3~bitl5: reserved

●Response packet format:

Table 3-54 Automatic verification fingerprint command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	parameter	ID number	Score	Checksum	Remark
2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes	2 bytes	
0xEF01	0xFFFFFFFF	07H	0008H	xxH	00H	xxxxH	xxxxH	sum	Instruction legality check: legitimate/..
0xEF01	0xFFFFFFFF	07H	0008H	xxH	01H	xxxxH	xxxxH	sum	Picture drawing result: Success/ time out
0xEF01	0xFFFFFFFF	07H	0008H	xxH	05H	xxxxH	xxxxH	sum	search results: success failure

●Confirmation code, return value of parameter 1 and parameter 2

Table 3-55 Automatic Verification Fingerprint Response Packet Interpretation Cheat Sheet

Confirmation code	Definition	parameter	Definition
00H	success	00H	Fingerprint legality detection

01H	fail	01H	Get image
07H	Failed to generate features	05H	Registered fingerprint comparison
09H	No fingerprint found		
0bH	ID number out of range		
17H	Residual fingerprints		
18H	Error reading and writing FLASH		
23H	Fingerprint template is empty		
24H	The fingerprint database is empty		
26H	time out		
27H	Indicates that the fingerprint already exists		
31H	Functions and encryption levels do not match;		
35H	illegal data		

●Instruction description:

1) If the fingerprint database is empty, the confirmation code and parameters are returned (the following is directly described as return): 24 00H. If the specified ID number is invalid, 0b 00H will be returned. If the registered Template does not exist, 23 00H will be returned.

2) If the command legality check is successful, 00 00H will be returned, and fingerprint entry will be entered.

3) Within the set timeout period, if a complete fingerprint entry is not completed, 26 00H will be returned to end the process.

4) Check the correctness of the entered fingerprint image. If it is incorrect, wait for the next image acquisition.

5) If the fingerprint input is correct, 00 01H will be returned, that is, the fingerprint input and image acquisition are successful.

6) If the feature generation fails, 09 05H will be returned and the process will end.

7) After successfully generating features, compare the currently collected fingerprint template with the registered fingerprint template, and return the result. If the comparison fails, 09 05H is returned and the process ends; if the comparison is successful, 00 05H is returned, along with the correct ID number and score.

8) If the PS_Cancel command is received, terminate the command and return a response.

3.3.3 Safety instruction set

Some fingerprint module products based on security chips support secure registration and verification functions.

3.3.3.1 Get the secret key pair PS_GetKeyt

- ●Function Description:After receiving the command, the module clears the internal data (if it has been entered) and generates a set of key pairs. For example, if the encryption level is set to 0 and 1 in Table 2-1, this function is not supported.

- Input parameters:none
- Return parameters:confirmation word
- Instruction code: E0H
- Command package format:

Table 3-56 Get the key pair instruction package format

Header	Device address	Package ID	Packet length	Script code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	0xFFFFFFFF	01H	0003H	E0H	sum

- Response packet format:

Table 3-57 Get key pair command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF0l	0xFFFFFFFF	07H	0003H	xxH	sum

Note: Confirmation code = 00H means that subsequent data packets will be sent;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 18H means the Flash operation failed;

Confirmation code = 19H means the random number operation failed;

Confirmation code = 2EH means the secret key does not exist;

Confirmation code = 0fH means that subsequent data packets cannot be sent;

Confirmation code = 31H means the function does not match the encryption level;

Confirmation code = 32H means the secret key is locked;

sum refers to the checksum.

●Send subsequent data packets after the reply

Table 3-58 Get the key pair data packet format

Header	Device address	Package ID	Packet length	data	Checksum
2 bytes	4bytes	1 byte	2 bytes	N bytes	2 bytes
0xEF0l	0xFFFFFF FF	xxH	xxH		sum

Note: Package ID=02: data package, and there are subsequent packages.

Packet ID=08: The last data packet, that is, the end packet.

When UART uploads data packets, it is divided into packets and sent according to the preset length.

●Auxiliary instructions:

At encryption level 2, after receiving the command, the module clears the internal data (if there is information entered), generates a 32-byte random number (16-byte key A, 16-byte key B) and saves it inside the module, and then sends it After receiving the two pairs of secret key data, the main control saves A and B internally. The data length of the two pairs of secret keys is 32 bytes;

At encryption level 3, after receiving the command, the module clears the internal data (if there is information entered), generates a 32-byte random number (16-byte key A, 16-byte key B) and saves it inside the module, and then sends it After receiving the two pairs of secret key data, the main control saves A and B internally. The data length of the two pairs of secret keys is 32 bytes;

At encryption level 4, after receiving the command, the module clears the internal data (if there is information entered), generates a 32-byte random number (16-byte key A, 16-byte key B) and saves it inside the module, and then sends it After receiving the two pairs of secret key data, the main control saves A and B internally. The data length of the two pairs of secret keys is 32 bytes;

When the encryption level is 20, after receiving the command, the module clears the internal data (if there is any entered information), generates an RSA key pair (1024-bit key pair is generated 2 times/sec, private key 75 times/sec) and is stored inside the module. , then send the public key and module, and the master will save them internally after receiving them. The public key and module length are (4+128) bytes.

When the encryption level is 21, the module clears the internal data (if there is any entered information) after receiving the command, generates an ECC key pair (256-bit key pair is generated 100 times/second, and the private key is 80 times/second) and is stored inside the module. , then send the private key data, and the master saves the private key after receiving it. The private key data length is 32 bytes.

Note: You need to pay attention to security when exchanging secret keys. First, during the product production process, it is called to ensure the safety of the external environment; then, after product maintenance or upgrade, when the main control is powered on for the first time, it needs to verify whether the encryption level matches, and if it does not match, it will be called automatically.

3.3.3.2 Lock key pair PS_LockKeyt

- ●Function Description:After the module receives the command, it no longer supports the master to obtain a new key pair. Encryption level settings as shown in Table 2-1Cases 0 and 1 do not support this feature.
- ●Input parameters:none
- ●Return parameters:confirmation word
- ●Instruction code:E1H
- ●Command package format:

Table 3-59 Lock key pair command packet format

Header	Device address	Package ID	Packet length	Script code	Checksum

2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	0xFFFFFFFF	01H	0003H	E1H	sum

●Response packet format:

Table 3-60 Lock key pair command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	0xFFFFFFFF	07H	0003H	xxH	sum

Note: Confirmation code = 00H means that subsequent data packets will be sent;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 18H means the Flash operation failed;

Confirmation code = 2EH means the secret key does not exist;

Confirmation code = 31H means the function does not match the encryption level;

Confirmation code = 32H means the secret key is locked;

sum refers to the checksum.

3.3.3.3 Get the ciphertext random number PS_GetCiphertext

- Function Description:Get the ciphertext or random number from the module side. As shown in Table 2-1, the encryption level is set to. Cases 1 and 1 do not support this feature.
- Input parameters:none
- Return parameters:confirmation word
- Instruction code:E2H
- Command package format:

Table 3-61 Get ciphertext random number instruction packet format

Header	Device address	Package ID	Packet length	Script code	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	0xFFFFFFFF	01H	0003H	E2H	sum

●Response packet format:

Table 3-62 Get ciphertext random number command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF0l	0xFFFFFFFF	07H	0003H	xxH	sum

Note: Confirmation code = 00H means that subsequent data packets will be sent;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 18H means the Flash operation failed;

Confirmation code = 19H means the random number operation failed;

Confirmation code = 2EH means the secret key does not exist;

Confirmation code = 2FH indicates that the security algorithm execution failed

Confirmation code = 31H means the function does not match the encryption level;

sum refers to the checksum.

●Subsequent packets are sent after the reply.

Table 3-63 Get the ciphertext random number packet format

Header	Device address	Package ID	Packet length	data	Checksum

2 bytes	4bytes	1 byte	2 bytes	N bytes	2 bytes
0xEF01	0xFFFFFFFF	xxH	xxH		sum

Note: Package ID=02: data package, and there are subsequent packages.

Packet ID=08: The last data packet, that is, the end packet.

When UART uploads data packets, it is divided into packets and sent according to the preset length.

●Auxiliary instructions:

At encryption level 2, the master sends a command, and after receiving the command, the module generates a 16-byte random number R and adds it.And use key A to encrypt R to get Q, and return it to the main control. The ciphertext data length is 16 bytes.

When the encryption level is 3, the master sends a command, and after receiving the command, the module generates a 16-byte random number R and adds it.And use key A to encrypt R to get Q, and return it to the main control. The ciphertext data length is 16 bytes.

When the encryption level is 4, the master sends a command, and after receiving the command, the module generates a 16-byte random number R and adds it.And use key A to encrypt R to get Q, and return it to the main control. The ciphertext data length is 16 bytes.

When the encryption level is 20, the master sends a command, and after receiving the command, the module generates a 16-byte random number R, which is encryptedAnd use the private key to advance R to get Q, and return it to the main control. The ciphertext data length is 128 bytes.

When the encryption level is 21, the master sends a command and the module generates a 16-byte random number R after receiving the command. The data length isReturn to master. The ciphertext size is 16 bytes.

3.3.3.4 Security storage template PS_SecurityStoreChar

- ●Function Description:Save the template file in the template buffer to the flash database location of PageID number. For example, if the encryption level is set to 0 and 1 in Table 2-1, this function is not supported.
- ●Input parameters:BufferID (default is 1), PageID (fingerprint library location number)
- ●Return parameters:confirmation word
- ●Instruction code:E3H
- ●Command package format:

Table 3-64 Secure storage template instruction package format

Header	Device address	Package ID	Packet length	Script code	buffer number	location number	handshake signal	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	xx bytes	2 bytes
0xEF0l	0xFFFFFFFF	01H	xxH	E3H	BufferlD	PagelD		sum

Note: BufferID defaults to 1.

- ●Response packet format:

Table 3-65 Secure storage template command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	0xFFFFFFFF	07H	0003H	xxH	sum

Note: Confirmation code = 00H indicates successful storage;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 0bH means that the PageID exceeds the range of the fingerprint database;

Confirmation code = 18H means the Flash operation failed;

Confirmation code = 19H means the random number operation failed;

Confirmation code = 2EH means the secret key does not exist;

Confirmation code = 2FH indicates that the security algorithm execution failed

Confirmation code = 30H indicates that the encryption and decryption results of the security algorithm are incorrect.

Confirmation code = 31H means the function does not match the encryption level;

sum refers to the checksum.

●Auxiliary instructions:

At encryption level 2, after receiving Q, the master uses A to decrypt to get R, and then uses key B to encrypt R to get M. "The master sends M to the module, and the module uses key B to decrypt M to get R." Compare R==R" executes the template storage otherwise it will not be executed. The handshake signal data length is 16 bytes.

At encryption level 3, after receiving Q, the master uses A to decrypt to get R, and then uses key B to encrypt R to get Mo. The master sends M to the module, and the module uses key B to decrypt M to get R." Compare R ==R" executes template storage otherwise it will not be executed. The handshake signal data length is 16 bytes.

At encryption level 4, after receiving Q, the master uses A to decrypt to get R, and then uses key B to encrypt R to get Mo. The master sends M to the module, and the module uses key B to decrypt M to get R." Compare R ==R" executes template storage otherwise it will not be executed. The handshake signal data length is 16 bytes.

When the encryption level is 20, after receiving Q, the master uses the public key to decrypt and obtain R,. The master sends R to the module, and the module compares R==R" to execute template storage, otherwise it will not be executed. The handshake signal data length is 16 bytes.

When the encryption level is 21, after receiving R, the master uses the private key to sign R to obtain Q. The master sends Q to the module, and the module uses the public key to verify Q. The signature verification is stored through the execution template, otherwise it will not be executed. The handshake signal data length is 64 bytes.

3.3.3.5 Secure Search Fingerprint PS_SecuritySearch

- Function Description:Search the whole or part of the fingerprint library with signature files in the template buffer. If found, the page number is returned. likeTable2-1This feature is not supported when the medium encryption level is set to 0 and 1.
- Input parameters:BufferID (default is 1), StartPage (start page), PageNum (number of pages), rand (random number)
- Return parameters:Confirmation word, page number (matching fingerprint template), score
- Instruction code:E4H
- Command package format:

Table 3-66 Safe Search Fingerprint Instruction Package Format

Header	Device address	Package ID	Packet length	Script code	buffer number	parameter	parameter	parameter	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes	16 bytes	2 bytes
0xEF01	0xFFFFFFFF	01H	0018H	E4H	BufferID	StartPage	PageNum		sum

Note: BufferID defaults to 1, and the entire or part of the fingerprint library is searched with the fingerprint template in the template buffer.

- Response packet format:

Table 3-67 Safe search fingerprint command response packet format

Header	Device address	Package ID		Packet length	Confirmation code	Checksum
2 bytes	4bytes	1 byte		2 bytes	1 byte	2 bytes
0xEF0l	0xFFFFFFFF	07H		03H	xxH	sum

Note: Confirmation code = 00H means searched;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 18H means the Flash operation failed;

Confirmation code = 19H means the random number operation failed;

Confirmation code = 2EH means the secret key does not exist;

Confirmation code = 2FH indicates that the security algorithm execution failed

Confirmation code = 31H means the function does not match the encryption level;

sum refers to the checksum.

●Subsequent packets are sent after the reply.

Table 3-68 Safe Search Fingerprint Instruction Data Packet Format

Header	Device address	Package ID	Packet length	Data	Checksum
2 bytes	4bytes	1 byte	2 bytes	N bytes	2 bytes
0xEF0l	0xFFFFFFFF	xxH	xxH		sum

●Auxiliary instructions:

At encryption level 2, the master generates a 16-byte random number, recorded as R, uses key A to encrypt R to get Q, and sends it to the module. After receiving the command, the module uses key A to decrypt Q to obtain R, and performs a search. The search result is denoted as T, the search ID is denoted as I, the search score is denoted as S, P=T (1 byte)|1 (2 bytes)|S (2 bytes)|R, (lower 11 bytes), Use key B to encrypt P to get M, and send it to the master. The master uses key B to decrypt M to get . The response data length is 16 bytes.

At encryption level 4, the master generates a 16-byte random number, recorded as R, uses key A to encrypt R to obtain Q, and sends it to the module. After receiving the command, the module uses key A to decrypt Q to obtain R, and performs a search. The search result is denoted as T, the search ID is denoted as I, the search score is denoted as S, P=T (1 byte)|1 (2 bytes)|S (2 bytes)|R, (lower 11 bytes), Use key B to encrypt P to get M, and send it to the master. The master uses key B to decrypt M to get . The response data length is 16 bytes.

When the encryption level is 20, the main control generates a 16-byte random number, recorded as R, and sends it to the module. The module performs a search after receiving the command. The search result is recorded as T, the search ID is recorded as I, and the search score is recorded as S. P=T (1 byte) | 1 (2 bytes) | S (2 bytes) | R (lower 11 bytes), use the private key to encrypt P to get Q, send it to the master, the master uses the public key to decrypt Q and get X, compare X (lower 11 bytes) ==R (lower 11 bytes) to get X High 5 bytes of valid data. The response data length is 128 bytes.

When the encryption level is 21, the main control generates a 16-byte random number, recorded as R, and sends it to the module. The module performs a search after receiving the command. The search result is recorded as T, the search ID is recorded as I, and the search score is recorded as S. P=T (1 byte) | 1 (2 bytes) | S (2 bytes) | R (lower 11 bytes), use the public key to encrypt P to get Q, send it to the master, the master uses the private key to decrypt Q and get X, compare X (lower 11 bytes) ==R (lower 11 bytes) to get X High 5 bytes of valid data. The response data length is 128 bytes.

At encryption level 3, the master generates a 16-byte random number, recorded as R, uses key A to encrypt R to obtain Q, and sends it to the module. After receiving the command, the module uses key A to decrypt Q to obtain R, and performs a search.

The search result is denoted as T, the search ID is denoted as I, the search score is denoted as S, P=T (1 byte)|1 (2 bytes)|S (2 bytes)|R, (lower 11 bytes), Use key B to encrypt P to get M, and send it to the master. The master uses key B to decrypt M to get . The response data length is 16 bytes.

3.4 Maintenance instruction set

3.4.1 Upload image PS_Uplmage

●Function Description:Upload the data in the image buffer to the main control.

●Input parameters:none

●Return parameters:confirmation word

●Instruction code:0aH

●Command package format:

Table 3-69 Upload image command package format

Header	Device address	Package ID	Packet length	Script code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	0xFFFFFFFF	01H	0003H	0aH	000eH

●Response packet format:

Table 3-70 Upload image command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF0l	0xFFFFFFFF	07H	0003H	xxH	sum

Note: Confirmation code = 00H means that subsequent data packets will be sent;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 0fH means that subsequent data packets cannot be sent;

sum refers to the checksum.

●Subsequent packets are sent after the reply.

Table 3-71 UART upload image packet format

Header	Device address	Package ID	Packet length	data	Checksum
2 bytes	4 bytes	1 byte	2 bytes	N bytes	2 bytes
0xEF01	0xFFFFFFFF	xxH	xxH	xxH	sum

Note: Package ID=02: data package, and there are subsequent packages.

Packet ID=08: The last data packet, that is, the end packet.

When UART uploads image data packets, it is divided into packets and sent according to the preset length.

●One byte contains two pixels, and each pixel occupies 4 bits.

3.4.2 Download image PS_Downlmage

- Function Description:The main control downloads image data to the module. This function is supported when the encryption level is set to 0 in Table 2-1.
- Input parameters:none
- Return parameters:confirmation word
- Instruction code:0bH
- Command package format:

Table 3-72 Download image command package format

Header	Device address	Package ID	Packet length	Script code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	0xFFFFFFFF	01H	0003H	0bH	000fH

Note: After the preprocessing function is turned on, the collected images can be uploaded, but the download function is not supported, let alone the subsequent fingerprint algorithm function.

●Response packet format:

Table 3-73 Download image command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	0xFFFFFFFF	07H	0003H	xxH	sum

Note: Confirmation code = 00H means subsequent data packets can be received;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 0eH means that subsequent data packets cannot be received;

Confirmation code = 31H means the function does not match the encryption level;

sum refers to the checksum.

●Following the reply, subsequent packets are received.

Table 3-74 UART download image packet format

Header	Device address	Package ID	Packet length	data	Checksum
2 bytes	4 bytes	1 byte	2 bytes	N bytes	2 bytes
0xEF01	0xFFFFFFFF	xxH	xxH	xxH	sum

Note: Package ID=02: data package, and there are subsequent packages.

Packet ID=08: The last data packet, that is, the end packet.

When UART downloads image data packets, it receives them in packets according to the preset length.

●One byte contains two pixels, and each pixel occupies 4 bits.

3.4.3 Get the unique serial number of the chip PS_GetChipSN

- Function Description:Get the unique serial number of the chip.
- Input parameters:Reserved.
- Return parameters:Confirmation word, unique serial number
- Instruction code:34H
- Command package format:

Table 3-75 Get the chip's unique serial number instruction packet format

Header	Device address	Package ID	Packet length	Script code	parameter	Checksum
2 bytes	4bytes	1 byte	2bytes	1 byte	1 byte	2 bytes
0xEF01	0xFFFFFFFF	01H	0004H	34H	0	0039H

●Response packet format:

Table 3-76 Get the unique serial number of the chip command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	unique serial number	Checksum
2 bytes	4bytes	1byte	2bytes	1byte	32 bytes	2bytes
0xEF01	0xFFFFFFFF	07H	0023H	xxH	SN	sum

Note: Confirmation code = 00H means OK;

Confirmation code = 01H means there is an error in receiving the package;

sum refers to the checksum.

3.4.4 Handshake command PS_HandShake

- Function Description:Check whether the module is working properly.
- Input parameters:none.
- Return parameters:confirmation word
- Instruction code:35H
- Command package format:

Table 3-77 Handshake command packet format

Header	Device address	Package ID	Packet length	Script code	Checksum
2 bytes	4 bytes	1 byte	2bytes	1 byte	2bytes
0xEF01	0xFFFFFFFF	01H	0003H	35H	0039H

- Response packet format:

Table 3-78 Handshake command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1byte	2bytes	1byte	2bytes
0xEF01	0xFFFFFFFF	07H	0003H	xxH	sum

Note: Confirmation code = 00H means OK;

Confirmation code = 01H means there is an error in receiving the package;

sum refers to the checksum.

3.4.5 Check sensor PS_CheckSensor

- Function Description:Verify that the sensor is working properly.
- Input parameters:none.
- Return parameters:confirmation word
- Instruction code:36H
- Command package format:

Table 3-79 Verify sensor command packet format

Header	Device address	Package ID	Packet length	Script code	Checksum
2 bytes	4 bytes	1 byte	2bytes	1 byte	2bytes
0xEF0l	0xFFFFFFFF	01H	0003H	36H	003AH

●Response packet format:

Table 3-80 Verify sensor command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1byte	2bytes	1byte	2bytes
0xEF0l	0xFFFFFFFF	07H	0003H	xxH	sum

Note: Confirmation code = 00H means OK;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 29H indicates that the sensor operation failed;

sum refers to the checksum.

3.4.6 Restore factory settings PS_RestSetting

- Function Description:After receiving the command, the module clears the internal data (if it has been entered) and deletes the internal key pair. The master can reacquire the key pair. For example, if the encryption level is set to 0 and 1 in Table 2-1, this function is not supported.
- Input parameters:none
- Return parameters:confirmation word
- Instruction code:3BH
- Command package format:

Table 3-81 Delete key pair command packet format

Header	Device address	Package ID	Packet length	Script code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	0xFFFFFFFF	01H	0003H	3BH	sum

●Response packet format:

Table 3-82 Delete key pair command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	0xFFFFFFFF	07H	0003H	xxH	sum

Note: Confirmation code = 00H means that subsequent data packets will be sent;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 18H means the Flash operation failed;

Confirmation code = 2EH means the secret key does not exist;

sum refers to the checksum.

3.5 Customized instruction set

3.5.1 Set password PS_SetPwd

- Function Description:Set the module handshake password. The default password of the fingerprint module system is 0. If the default password has not been modified, the system does not require password verification during communication, and the main control can communicate directly with the module; if the password is modified, the first one between the main control and the device module The command must be to verify the password. Only after the password verification is passed, the module will receive other commands.

- Input parameters:Password
- Return parameters:confirmation word
- Instruction code:12H
- Command package format:

Table 3-83 Set password command packet format

Header	Device address	Package ID	Packet length	Script code	Password	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	4 bytes	2 bytes
0xEF01	0xFFFFFFFF	01H	0007H	12H	Password	sum

●Response packet format:

Table 3-84 Set password command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	0xFFFFFFFF	07H	0003H	xxH	sum

Note: Confirmation code = 00H means 0K;

Confirmation code = 01H means there is an error in receiving the package;

sum refers to the checksum.

3.5.2 Verification password PS_VfyPwd

●Function Description:Verify module password.

●Input parameters:Password

●Return parameters:confirmation word

●Instruction code:13H

●Command package format:

Table 3-85 Verify password command packet format

Header	Device address	Package ID	Packet length	Script code	Password	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	4 bytes	2 bytes
0xEF01	0xFFFFFFFF	01H	0007H	13H	Password	sum

●Response packet format:

Table 3-86 Verify password command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	0xFFFFFFFF	07H	0003H	xxH	sum

Note: Confirmation code = 00H means the password verification is correct;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 13H means the password is incorrect;

sum refers to the checksum.

3.5.3 Sampling random number PS_GetRandomCode

●Function Description:Let the module generate a random number and return it to the main control.

●Input parameters:none

●Return parameters:Confirmation word, random number

●Instruction code:14H

●Command package format:

Table 3-87 Sample random number command packet format

Header	Device address	Package ID	Packet length	Script code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	0xFFFFFFFF	01H	0003H	14H	0018H

●Response packet format:

Table 3-88 Sample random number command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	random number	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	4 bytes	2 bytes
0xEF0l	0xFFFFFFFF	07H	0007H	xxH	XXXX	sum

Note: Confirmation code = 00H means the generation is successful;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 19H means random number generation failed;

sum refers to the checksum.

3.5.4 Set device address PS_SetChipAddr

- Function Description:The default address of the module is 0xffffffff, which can be modified through this command. The address field of the command packet/data packet must match this address before it can be received by the fingerprint module.
- Input parameters:Device address
- Return parameters:confirmation word
- Instruction code:15H
- Command package format:

Table 3-89 Set device address command packet format

Header	Device address	Package ID	Packet length	Script code	Device address	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	4 bytes	2 bytes
0xEF0l	0xFFFFFFFF	01H	0007H	15H	XXXX	sum

●Response packet format:

Table 3-90 Set device address command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF0l	0xFFFFFFFF	07H	0007H	xxH	sum

Note: Confirmation code = 00H means the address is generated successfully;

Confirmation code = 01H means there is an error in receiving the package;

sum refers to the checksum.

●The command has been returned with a correct response for the time being, and the communication device address will not change after it is issued; but the system parameters will be updated.

3.5.5 Write Notepad PS_WriteNotepad

- Function Description:The module internally opens up 512bytes of FLASH space for users to store user data. This storage space is called user notepad. The notepad is logically divided into 16 pages. The write notepad command is used to write the user's 32bytes data to The specified notepad page. Note that when writing a certain page of Notepad, the entire 32-byte content of the page is written, and the original content is overwritten.
- Input parameters:Page number, user information
- Return parameters:confirmation word
- Instruction code:18H

- Command package format:

Table 3-91 Write notepad command package format

Header	Device address	Package ID	Packet length	Script code	page number	User Info	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	1byte	32 bytes	2 bytes
0xEF01	0xFFFFFFFF	01H	24H	18H	0~15	User content	sum

●Response packet format:

Table 3-92 Write Notepad command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	0xFFFFFFFF	07H	0003H	xxH	sum

Note: Confirmation code = 00H means 0K;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code =lcH means the notepad page number is incorrectly specified;

sum refers to the checksum.

3.5.6 Read Notepad PS_ReadNotepad

●Function Description:Read data from Notepad.

●Input parameters:page number

●Return parameters:Confirmation word, user information

●Instruction code:19H

●Command package format:

Table 3-93 Read notepad command packet format

Header	Device address	Package ID	Packet length	Script code	page number	Checksum

2 bytes	4 bytes	1 byte	2 bytes	1 byte	1byte	2 bytes
0xEF01	0xFFFFFFFF	01H	0004H	19H	0~15	xxxxH

●Response packet format:

Table 3-94 Read notepad command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	User Info	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	32 bytes	2 bytes
0xEF01	0xFFFFFFFF	07H	23H	xxH	User content	sum

Note: Confirmation code = 00H means OK;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code =lcH means the notepad page number is incorrectly specified;

sum refers to the checksum.

3.5.7 LED control light command PS_ControlBLN

- ●Function Description:Control light instructions are mainly divided into two categories: general indicator lights and colorful programming breathing lights.
- ●Input parameters:Function code, starting color, ending color, number of cycles
- ●Return parameters:confirmation word
- ●Instruction code:3CH
- ●Command package format:

Table 3-95 General indicator light command packet format

Header	Chip address	Package ID	Packet length	Script code	Function code	Starting color	End color	Cycles	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	1 bytes	1 bytes	1 bytes	2 bytes

0xEF01	0xFFFFFFFF	01H	0007H	3CH	xx	xx	xx	xx	sum

●Auxiliary instructions

Function code: LED light mode control bit, 1-normal breathing light, 2-flash light, 3-normally open light, 4-normally closed light, 5-gradually open light, 6-gradually close light. Other function codes are not suitable for this command package format;

Starting color: When set to the ordinary breathing light, the color from off to bright is limited to the ordinary breathing light (function code 01) function. When used for other functions, it is consistent with the end color. Among them, bit. It is the blue light control bit; bitl is the green light control bit; bit2 is the red light control bit. Set to 1 to turn on the light, and to set to 0 to turn off the light. For example, 0x0l—blue light is on, 0x02—green light is on, 0x04—red light is on, 0x06—red and green lights are on, 0x05—red and blue lights are on, 0x03—green and blue lights are on, 0x07—red, green, and blue lights are on, 0x00—all off;

End color: When set to the ordinary breathing light, the color from on to off is limited to the ordinary breathing light (function code 0x01). For other functions, it is consistent with the starting color. The setting method is the same as the starting color;

Number of cycles: Indicates the number of breaths or flashing lights. When set to 0, it represents an infinite loop; when set to other values, it represents a limited number of breaths. The number of cycles is applicable to breathing and flashing functions, and is invalid in other functions, such as normally open, normally closed, gradual opening and gradual closing.

●Response packet format:

Table 3-96 Breathing light command response packet format

Header	Chip address	Package ID	Packet length	Confirmation code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes

0xEF01	0xFFFFFF FF	07H	0003H	xxH	sum

Note: Confirmation code = 00H means the command is executed successfully;

Confirmation code = 01H means there is an error in receiving the package;

sum refers to the checksum.

3.5.8 Get image information command PS_Getlmagelnfo

●Function Description:After detection, the fingerprint image is recorded and stored in the image buffer, and the image information is returned.

●Input parameters:none

●Return parameters:Confirmation word, image area (percentage), image quality (0: qualified; others: unqualified)

●Instruction code:3DH

●Command package format:

Table 3-97 Get image information command packet format

Header	Device address	Package ID	Packet length	Script code	Checksum
2 bytes	4 bytes	1 byte	2bytes	1 byte	2bytes
0xEF01	0xFFFFFF FF	01H	0003H	3DH	0041H

●Response packet format:

Table 3-98 Obtain image information command response packet format

Header	Device address	Package ID	Packet length	Confirmati on code	Image area	Image Quality	Checksu m
2 bytes	4bytes	1byte	2bytes	1byte	1byte	1byte	2bytes

0xEF01	0xFFFFFFF F	07H	05	xxH	xxH	xxH	sum

Note: Confirmation code = 00H means the image acquisition is successful;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 02H means there is no finger on the sensor;

Confirmation code = 06H means that the image is too messy to generate a feature;

Confirmation code = 33H means the image area is small;

sum refers to the checksum.

3.5.9 Search current fingerprint command PS_SearchNow

- Function Description:Search the entire or part of the fingerprint library with the most recently extracted feature file in the template buffer. If found, the page number is returned. As shown in Table 2-1, the encryption level is set to. Or 1 case supports this feature.
- Input parameters:StartPage (start page), PageNum (number of pages)
- Return parameters:Confirmation word, page number (matching fingerprint template), score
- Instruction code:3EH
- Command package format:

Table 3-99 Search the current fingerprint instruction packet format

Header	Device address	Package ID	Packet length	Script code	paramete r	parameter	Checksu m
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes	2 bytes	2 bytes
0xEF01	0xFFFFFF FF	01H	0007H	3EH	StartPage	PageNum	sum

- Response packet format:

Table 3-100 Search the current fingerprint command response packet format

Header	Device address	Package ID	Packet length	Confirmation code	Page number	Score	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes	2 bytes	2 bytes
0xEF01	0xFFFFFFFF	07H	07H	xxH	PageID	MatchScore	sum

Note: Confirmation code = 00H means searched;

Confirmation code = 01H means there is an error in receiving the package;

Confirmation code = 09H means not found; at this time, the page number and score are 0;

Confirmation code = 17H means residual fingerprints or the finger has not moved between two collections;

Confirmation code = 31H means the function does not match the encryption level;

sum refers to the checksum.

4 Function implementation example

4.1 Basic communication process

4.1.1 UART command packet processing process

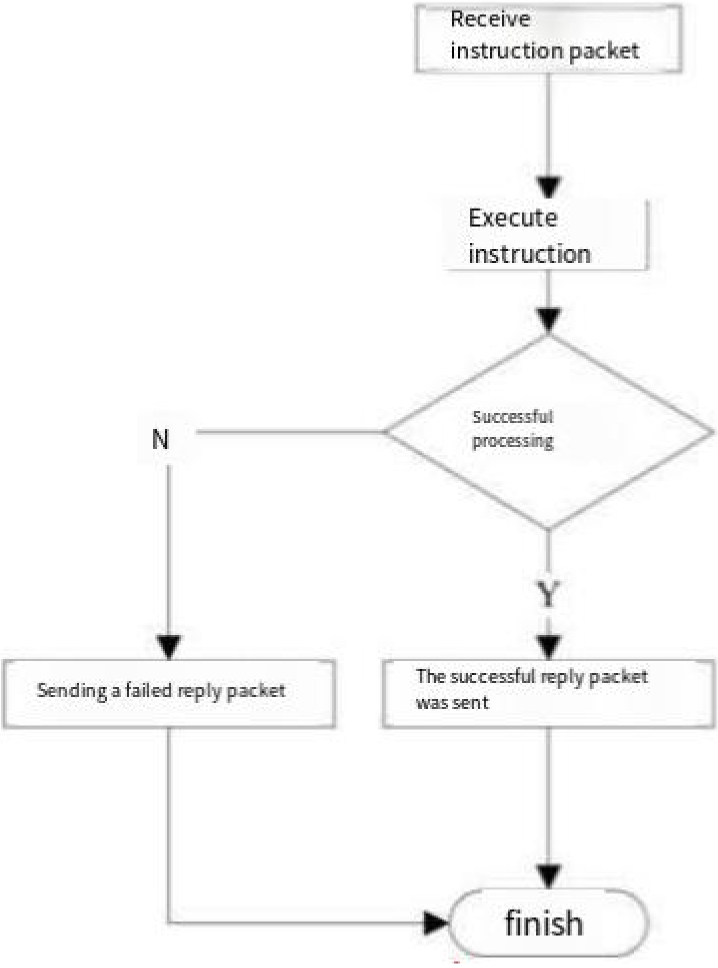
Receive instruction packet
Execute instruction
Successful processing
N
Y
Sending a failed reply packet
The successful reply packet was sent
finish

Figure4-1Function implementation example 1: UART command packet processing process

4.1.2 UART data packet sending process

Before UART transmits a data packet, it must first receive the command packet to transmit the data packet, send a successful response packet after preparing for transmission, and finally start transmitting the data packet. The data packet mainly includes: packet header, device address, packet identification, packet length, data and checksum.

There are two main types of packet identifiers for data packets: 02H and 08H. 02H: Data packet, and there are subsequent packets. 08H: The last data packet, the end packet. The data length is preset, mainly divided into four types: 32, 64, 128, and 256.

For example, if the data length to be transmitted is IK bytes and the preset data length in the data packet is 128 bytes, then the 1K bytes data must be divided into 8 data packets for transmission. Each data packet includes: 2bytes header, 4bytes device address, 1bytes packet identifier, 2bytes packet length, 128bytes data and 2bytes checksum. Each data packet length is 139bytes. In addition, among the 8 data packets, the report ID of the first 7 data packets is 02H, and the report ID of the last ending data packet is 08H. The last thing to note is that if the length of the end packet does not reach 139 bytes, it will be transmitted with the actual length and will not be expanded to 139 bytes in other ways.

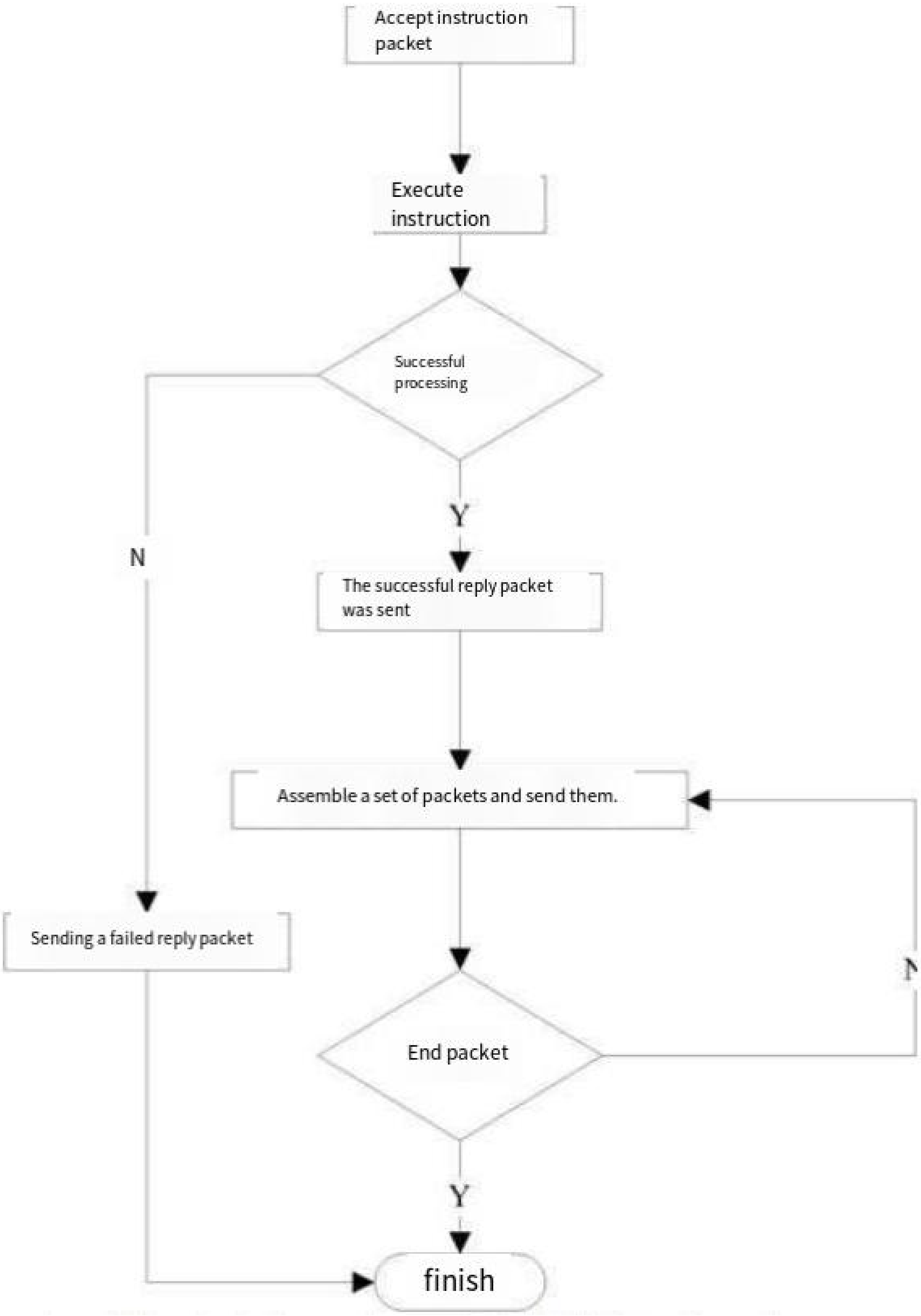
Accept instruction packet
Execute instruction
Successful processing
Y
N
The successful reply packet was sent
Assemble a set of packets and send them.
Sending a failed reply packet
N
End packet
Y
finish

Figure4-2 Function implementation example 2: UART data packet sending process

4.1.3 4UART data packet reception process

Before UART transmits a data packet, it must first receive the command packet to transmit the data packet, send a successful response packet after preparing for transmission, and finally start transmitting the data packet. The data packet mainly includes: packet header, device address, packet identification, packet length, data and checksum.

There are two main types of packet identifiers for data packets: 02H and 08H. 02H: Data packet, and there are subsequent packets. 08H: The last data packet, the end packet. The data length is preset, mainly divided into four types: 32, 64, 128, and 256.

For example, if the data length to be transmitted is IK bytes and the preset data length in the data packet is 128 bytes, then the data of 1K bytes must be divided into 8 data packets for transmission. Each data packet includes: 2 bytes packet header, 4 bytes device address, 1 bytes packet identification, 2bytes packet length, 128bytes data and 2bytes checksum. Each data packet length is 139bytes. In addition, among the 8 data packets, the report ID of the first 7 data packets is 02H, and the report ID of the last ending data packet is 08H. The last thing to note is that if the length of the end packet does not reach 139 bytes, it will be transmitted with the actual length and will not be expanded to 139 bytes in other ways.

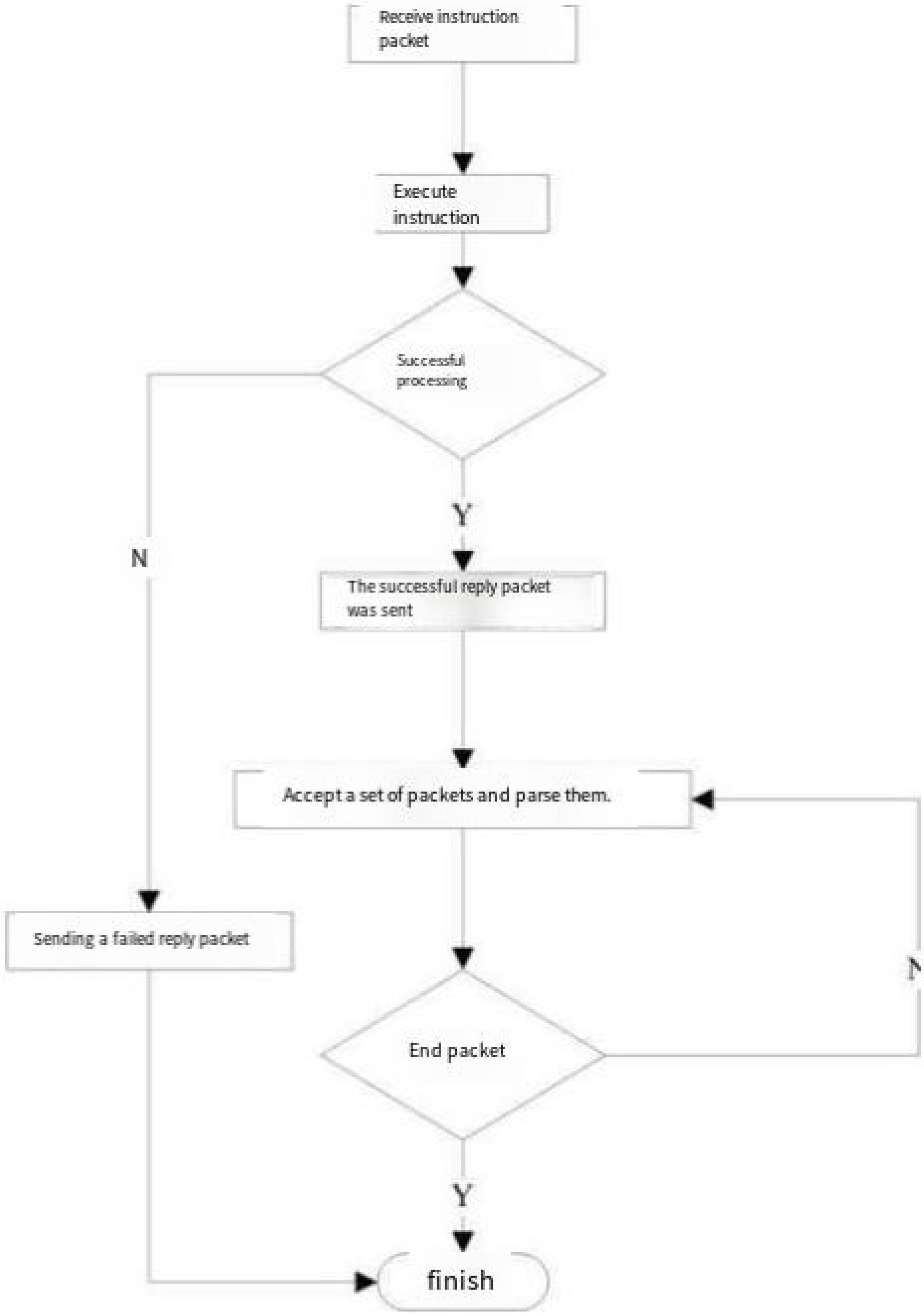
Receive instruction packet
Execute instruction
Successful processing
Y
N
The successful reply packet was sent
Accept a set of packets and parse them.
Sending a failed reply packet
N
End packet
Y
finish

Figure 4-3 Function implementation example 3: Receiving process of UART data packet

4.2 General command communication process

4.2.1 General instruction fingerprint registration process

The general instruction fingerprint registration process mainly includes: obtaining images for registration, generating features, merging features and storing templates. For example, for products such as XTL0605 and XTL0803, the default is N = 5 times for 160*160 pixels and N = 6 for below 120 pixels.

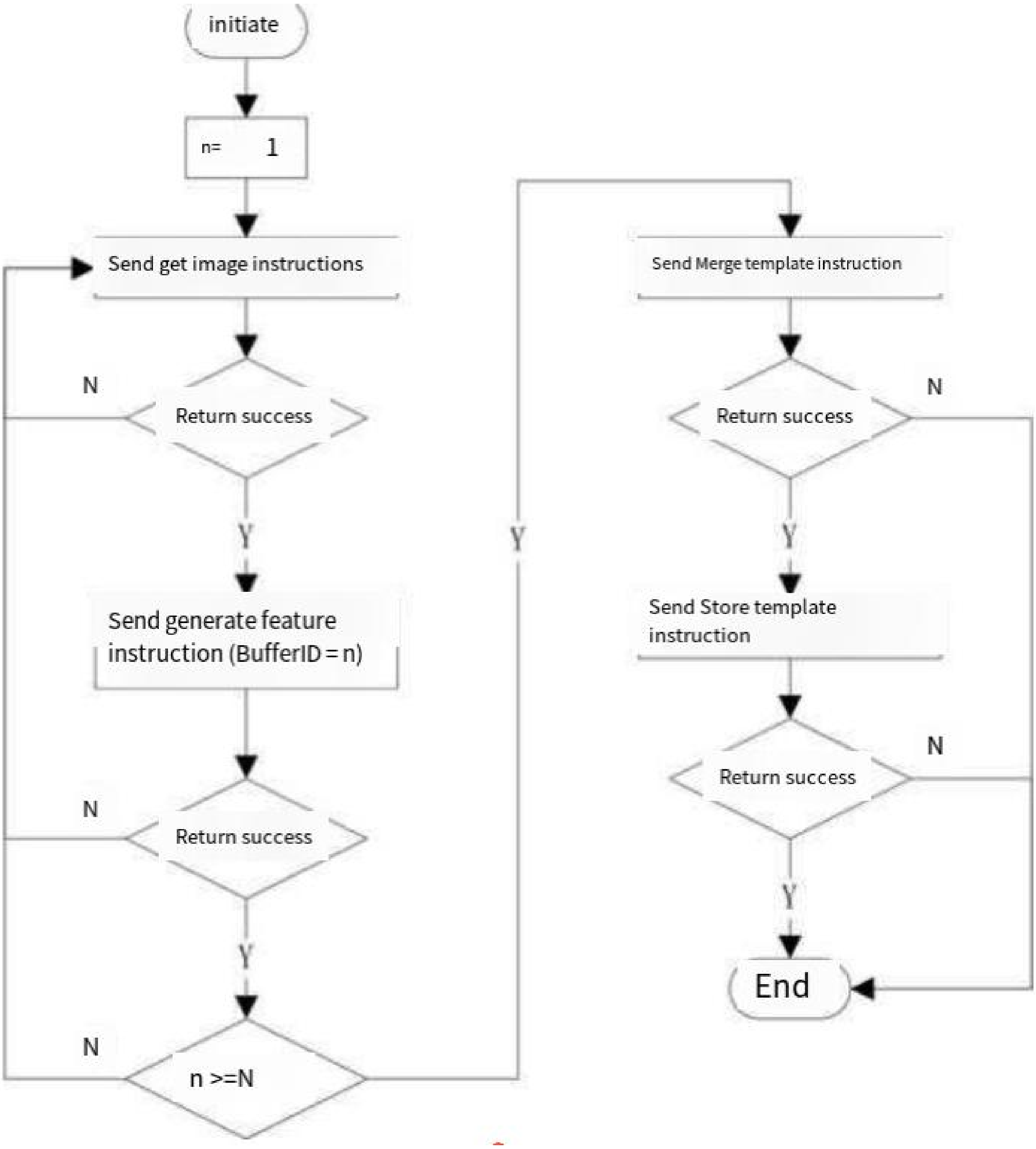
initiate
n= 1
Send get image instructions
N
Return success
Y
Send generate feature instruction (BufferID = n)
N
Return success
Y
N
n >=N
Y
Send Merge template instruction
Return success
N
Y
Send Store template instruction
Return success
N
Y
End

Figure 4-4 Function Implementation Example 4: General Instruction Registration Process

likeTable2-1In , when the registration logic is set to 1, the fingerprint is registered. If the currently collected fingerprint is similar to the fingerprint that has been collected before, the confirmation code in the response packet of the feature command will not show success, but will return 28H, indicating that the current fingerprint feature is different from that of the previous fingerprint. There was a correlation between the previous features. It should be noted that the mutual comparison correlation is limited to the fingerprints included in this registration process and will not be compared with the fingerprints in the fingerprint database.

Like Table2-1 In , when the registration logic is set to 2, the fingerprint is registered. If the currently collected fingerprint is not similar to the fingerprint that has been collected before, the confirmation code in the response packet of the feature generation command will not show success, but will return 08H, indicating the current fingerprint feature. No correlation with previous features. It should be noted that the mutual comparison correlation is limited to the fingerprints included in this registration process and will not be compared with the fingerprints in the fingerprint database.

Regardless of whether 28H or 08H is returned, the current fingerprint features have been extracted successfully. You can re-draw the image and generate features without changing the BufferID, or you can skip the BufferID of this round and collect the next round of fingerprints.

4.2.2 General instruction verification fingerprint process

The general instruction fingerprint verification process mainly includes: obtaining images for verification, generating features and searching for fingerprints. When sending generated features and search fingerprints, BufferID is set to the default value of 1.

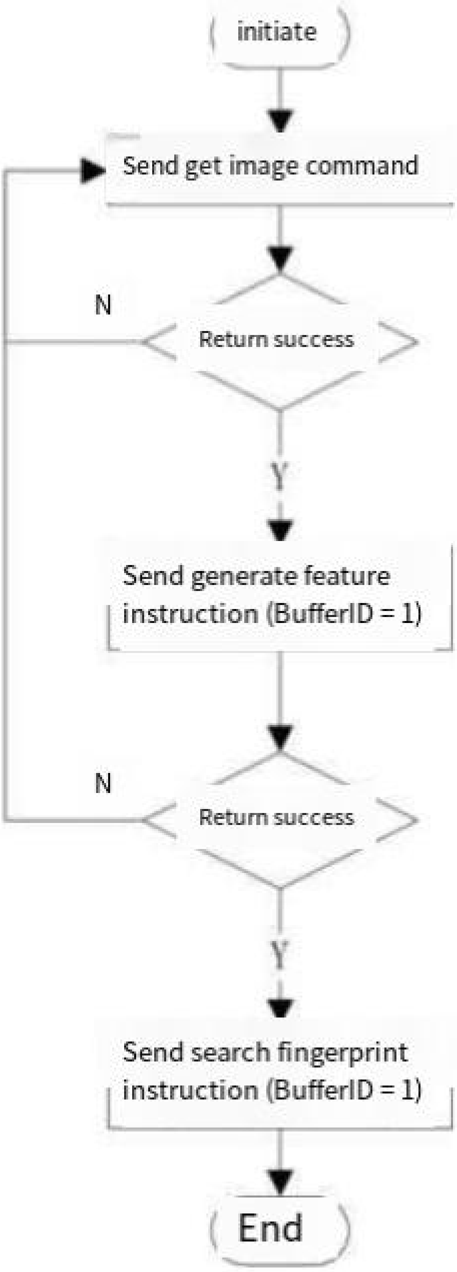
initiate
Send get image command
N
Return success
Y
Send generate feature instruction (BufferID = 1)
N
Return success
Y
Send search fingerprint instruction (BufferID = 1)
End

Figure4-5 Function Implementation Example 5: General Instruction Verification Process

4.2.3 Read a specified template from the flash fingerprint library and upload it

The entire process mainly includes: reading out templates and uploading templates. When sending readout templates and uploading features, BufiferlD is set to a default value of 2. This function is supported when the encryption level is set to 0 in Table 2-1.

It should be noted that what is uploaded at this time is a template, not a feature file.

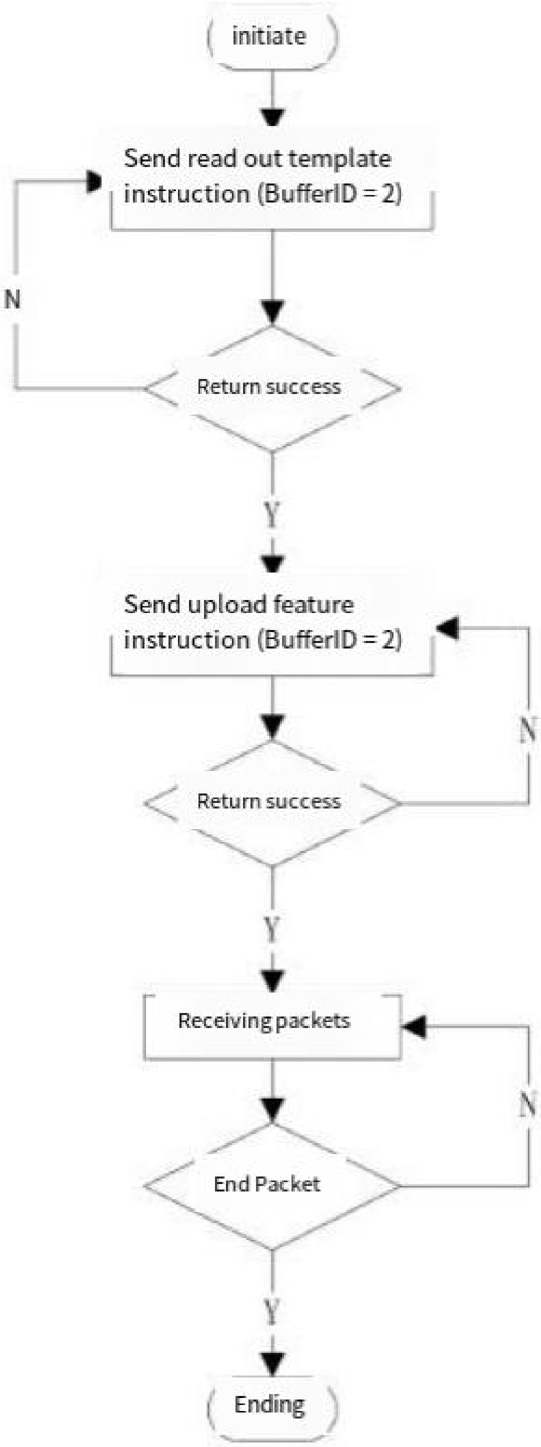
initiate
Send read out template instruction (BufferID = 2)
N
Return success
Y
Send upload feature instruction (BufferID = 2)
N
Return success
Y
Receiving packets
N
End Packet
Y
Ending

Figure4-6Function implementation example 6: Read a specified template from the flash fingerprint library and upload it

4.3 Module command communication process

4.3.1 Automatic registration template process

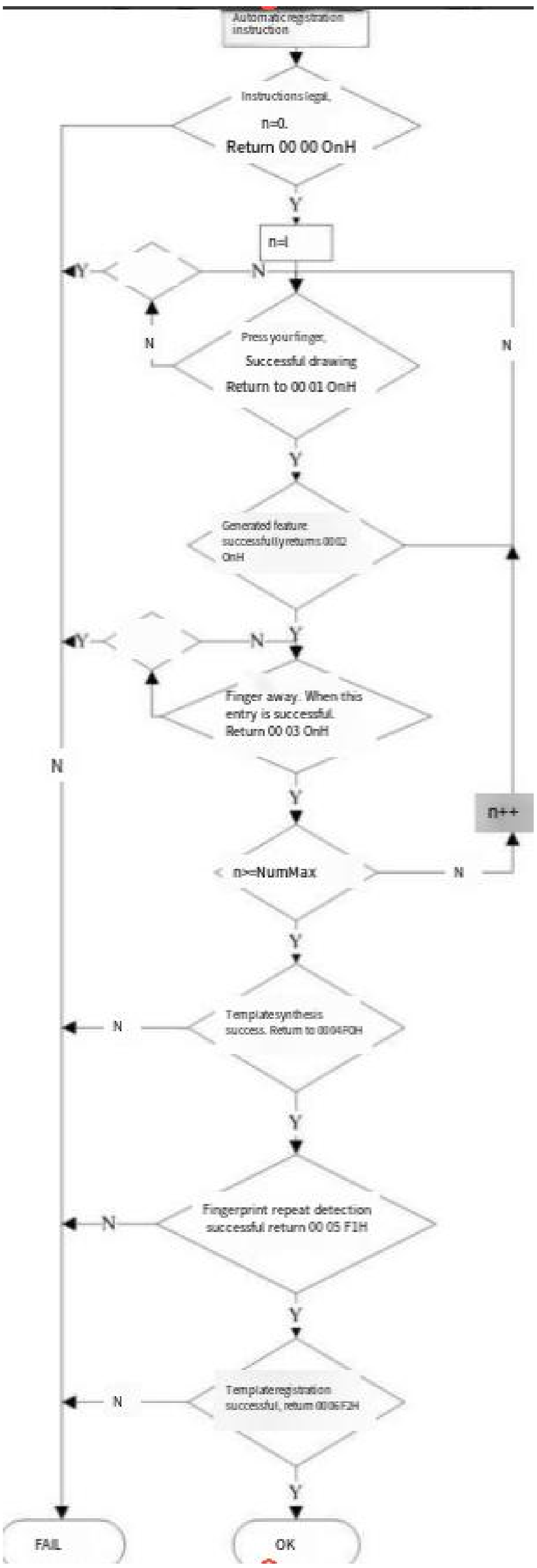
Automatic registration instruction
Instructions legal,
n=0.
Return 00 00 OnH
Y
n=l
Y
N
N
Press your finger,
Successful drawing
Return to 00 01 OnH
N
Y
Generated feature successfully returns 00 02 OnH
Y
N
Y
Finger away. When this entry is successful.
Return 00 03 OnH
N
Y
n++
n>=NumMax
N
Y
N
Templatesynthesis success. Return to 00 04 F0H
Y
N
Fingerprint repeat detection successful return 00 05 F1H
Y
N
Templateregistration successful, return 00 06 F2H
Y
FAIL
OK

Figure 4-7 Function Implementation Example 7: Automatic Registration Template Process

Like Table2-1 In , when the registration logic is set to 1, the fingerprint is registered. If the currently collected fingerprint is similar to the fingerprint that has been collected before, the confirmation code in the response packet of the feature command will not show success, but will return 28H, indicating that the current fingerprint feature is different from that of the previous fingerprint. There was a correlation between the previous features. It should be noted that the mutual comparison correlation is limited to the fingerprints included in this registration process and will not be compared with the fingerprints in the fingerprint database. In addition, it should be noted that unlike the general instruction fingerprint registration process, the number of correlation comparisons in the automatic registration template process is limited. Once the number limit is exceeded, it will no longer be compared.

Compare current fingerprint features with previous features.

Like Table2-1 In , when the registration logic is set to 2, the fingerprint is registered. If the currently collected fingerprint is not similar to the fingerprint that has been collected before, the confirmation code in the response packet of the feature generation command will not show success, but will return 08H, indicating the current fingerprint feature. No correlation with previous features. It should be noted that the mutual comparison correlation is limited to the fingerprints included in this registration process and will not be compared with the fingerprints in the fingerprint database. In addition, it should be noted that unlike when the automatic registration template process registration logic is set to 1, there is no limit to the number of correlation comparisons.

4.3.2 Automatic fingerprint verification process

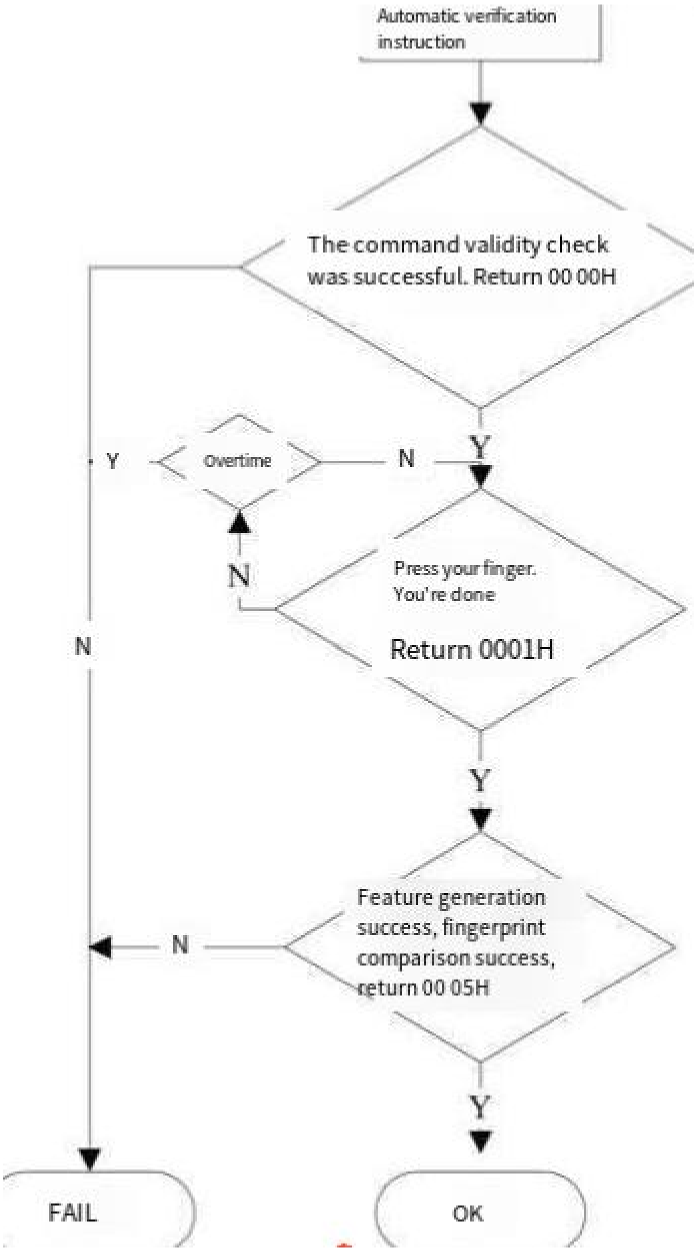
Automatic verification instruction
The command validity check was successful. Return 00 00H
Y
Overtime
N
Y
Press your finger. You're done
Return 0001H
N
N
Y
Feature generation success, fingerprint comparison success, return 00 05H
N
Y
FAIL
OK

Figure 4-8 Function Implementation Example 8: Automatic Fingerprint Verification Process

4.4 Safety command communication process

4.4.1 Security instruction fingerprint registration process

Before calling the security instruction set registration, you need to call the write system register PS_WriteReg instruction to set the encryption level, and then call the PS_GetKeyt instruction to obtain and store the secret key.

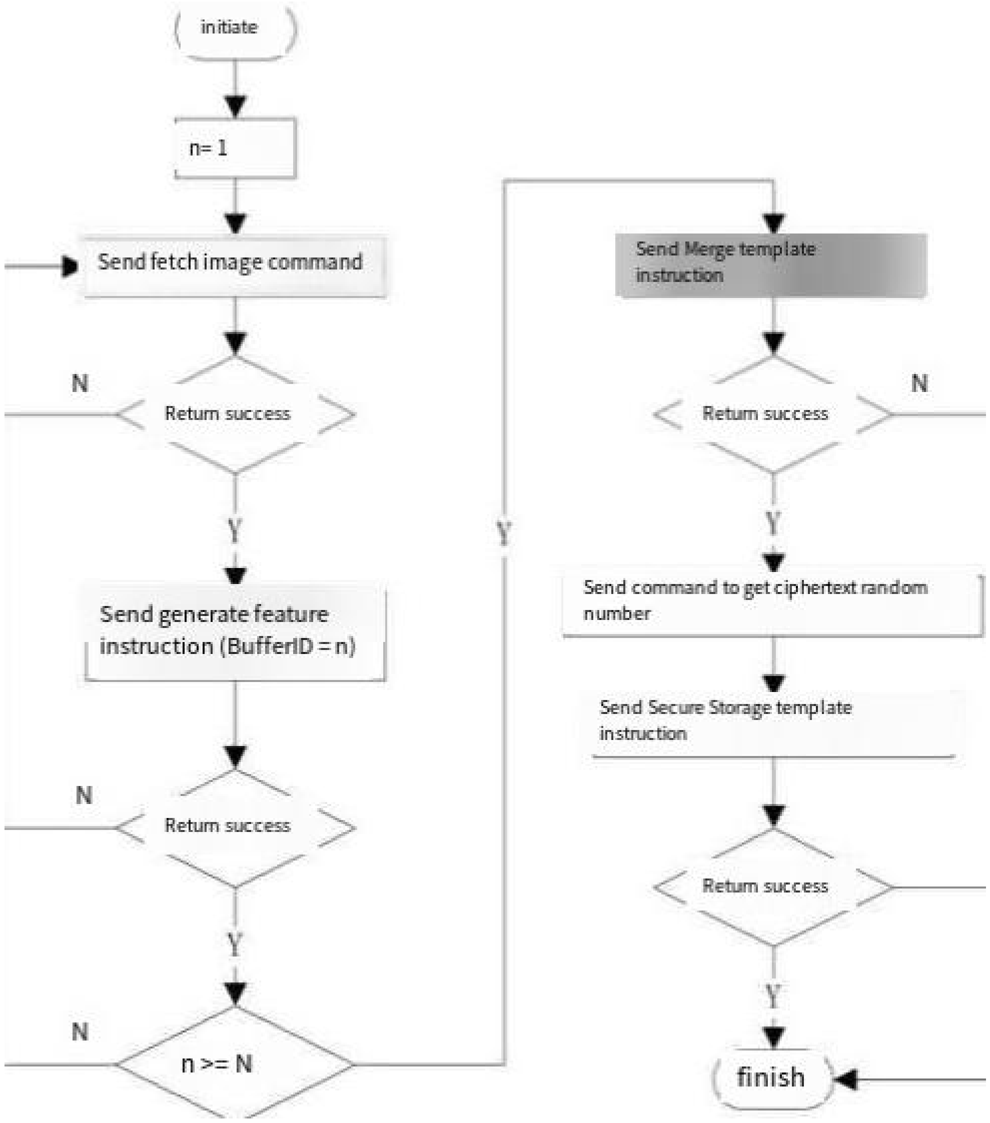
initiate
n= 1
Send fetch image command
N
Return success
Y
Send generate feature instruction (BufferID = n)
N
Return success
Y
N
n >= N
Y
Send Merge template instruction
Return success
N
Y
Send command to get ciphertext random number
Send Secure Storage template instruction
Return success
Y
finish

Figure 4-9 Function Implementation Example 9: Security Instruction Registration Fingerprint Process

4.4.2 Security command verification fingerprint process

Before calling the security instruction set verification, you need to call the PS_WriteReg instruction to write the system register to set the encryption level, and then call the PS_GetKeyt instruction to obtain and store the secret key.

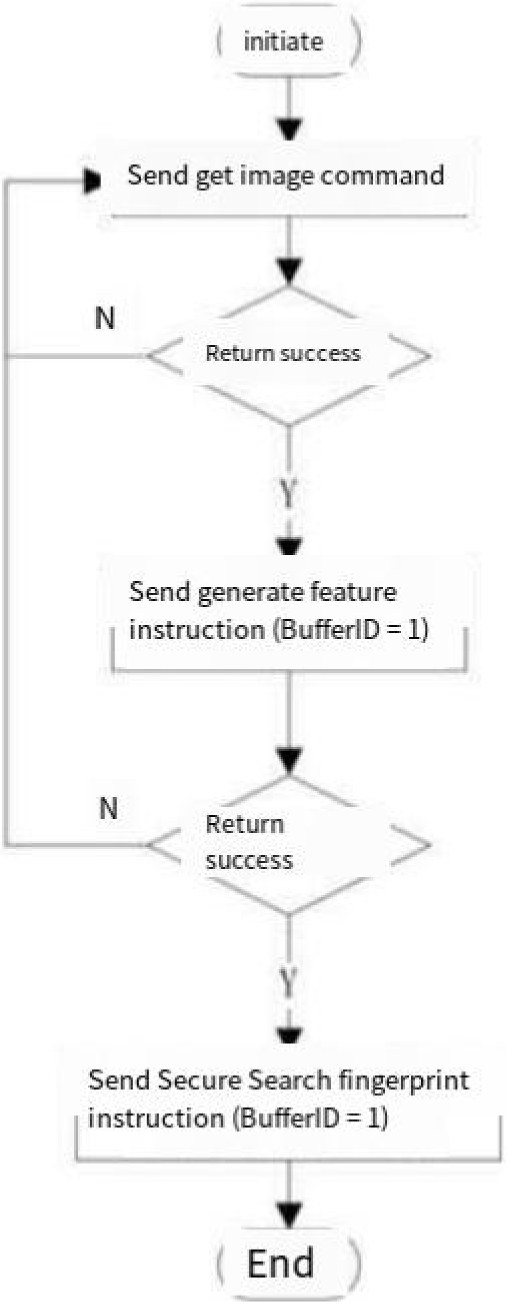
initiate
Send get image command
N
Return success
Y
Send generate feature instruction (BufferID = 1)
N
Return success
Y
Send Secure Search fingerprint instruction (BufferID = 1)
End

Figure 4-10 Function Implementation Example 10: Security Instruction Verification Fingerprint Process

4.5 Sleep wake-up process

It should be noted that the touch feedback pin signal of the fingerprint module is only effective after successfully entering the sleep process.

Starting from 2020, our company will no longer manufacture and produce external trigger fingerprint modules.

4.5.1 self-triggering process

Below is an example process for a self-triggered product without a touch chip.

It should be noted that if you want to enter the self-triggered mode into low power consumption, you must first send a sleep command to the module, and then cut off the module fingerprint chip power after receiving a successful response.

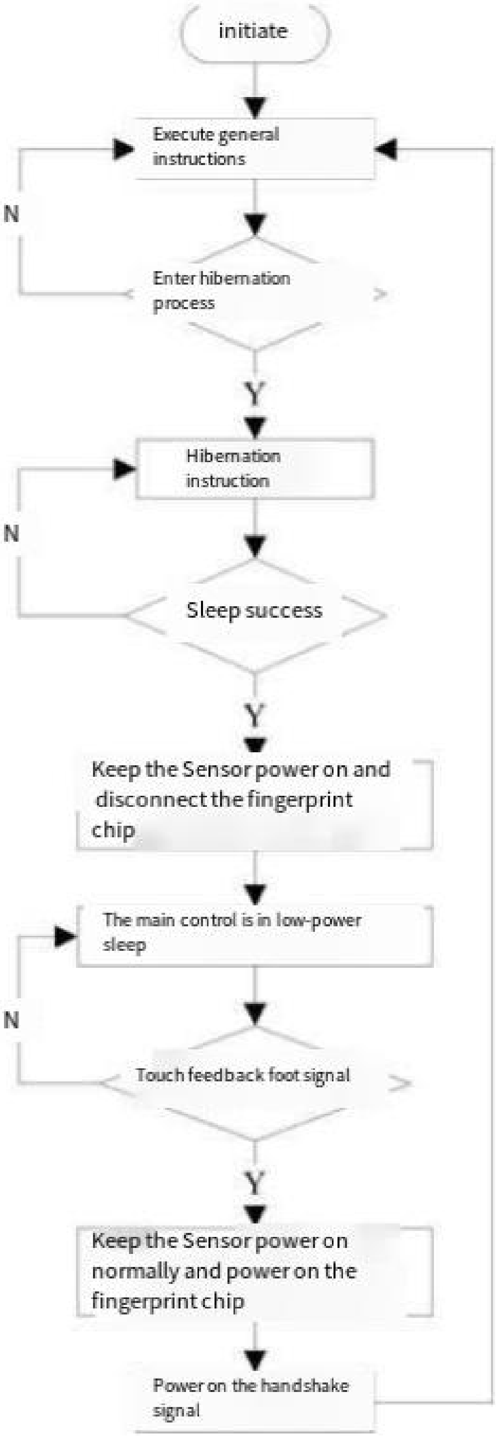
initiate
Execute general instructions
N
Enter hibernation process
Y
Hibernation instruction
N
Sleep success
Y
Keep the Sensor power on and disconnect the fingerprint chip
The main control is in low-power sleep
N
Touch feedback foot signal
Y
Keep the Sensor power on normally and power on the fingerprint chip
Power on the handshake signal

Figure 4-11Function implementation example 11: Self-triggering process

4.5.2 External trigger process

The following is an example process for an externally triggered product with a touch chip.

It should be noted that if you want the external trigger to enter low power consumption, just cut off the power supply of the module fingerprint chip.

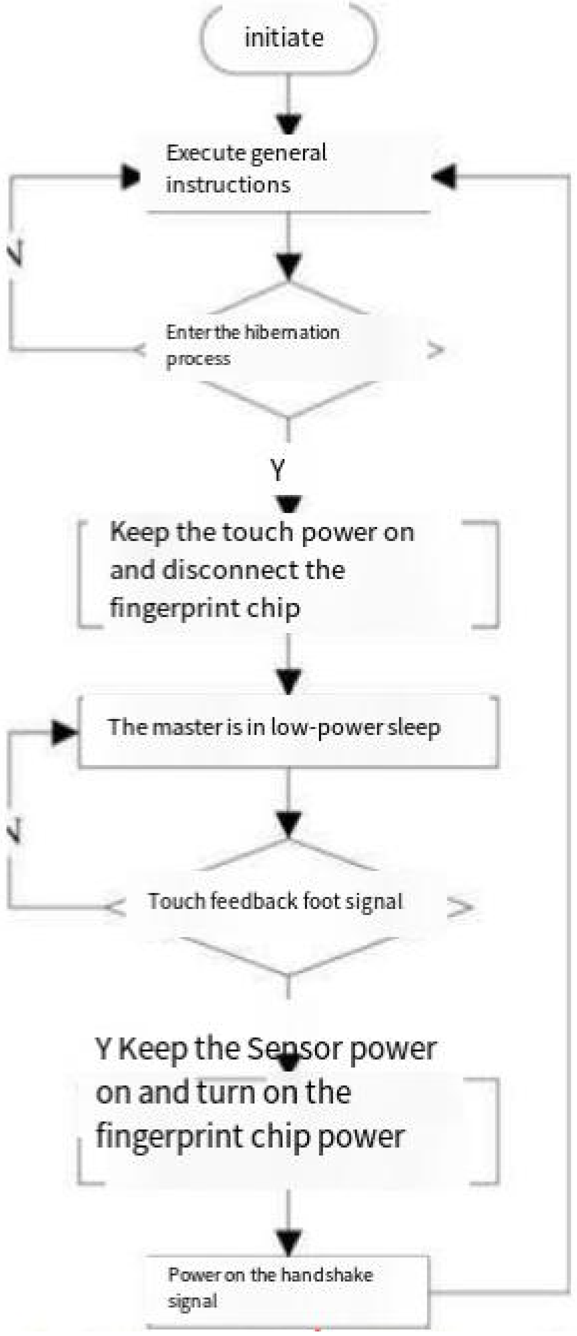
initiate
Execute general instructions
N
Enter the hibernation process
Y
Keep the touch power on and disconnect the fingerprint chip
The master is in low-power sleep
N
Touch feedback foot signal
Y Keep the Sensor power on and turn on the fingerprint chip power
Power on the handshake signal

Figure 4-12 Function implementation example 12: external trigger process

Technical support and contact information

Shenzhen Hi-Link Electronic Co.,Ltd

Address: 1705, 17/F, Building E, XingheWORLD, Minle Community, Minzhi Street,
Longhua District, Shenzhen
Tel：0755-23152658/83575155
E-mail: sales@hlktech.com
Website: https://www.hlktech.net/