
JM-101 Optical Fingerprint Module User manual

Version 1.8, Jan 2017

Sales and technical support

Address:

Post code:

Contact:

Telephone:

Version History

Version	Date	Revise content		
		Charter	person	Content
1.0	2016-01-11	Full text		original version
1.2	2016-02-20	4-6		Add the application example and one-stop agreement instruction
1.3	2016-04-26	5-3-8		Add read index table command
1.4	2016-05-18	5-3-2		Add automatic validation command pass function
1.5	2016-06-16	5-3-2		Automatic validation command update , sample byte 06 changed to 08
1.6	2016-07-18	5-3		Automatic registration and automatic authentication command update
1.7	2016-08-01	5.3.11		Add the command to get the chip's only ordinal number
1.8	2017-01-10	5.3.1-5.3.2		Update the one-stop command description

Contents

Sales and technical support.....	1
Version History	2
Contents	3
1 Overview	5
1.1 overview	5
1.2 working principle.....	5
1.3 Technical Parameter.....	6
2 Hardware interface.....	7
2.1 Hardware connection	8
2.2 Touch-sensitive.....	8
2.3 Serial protocol.....	8
2.4 Time of electrical delay	8
3 Mechanical structure.....	9
3.1 JM-101 Shell Size.....	9
4 Functional implementation examples	10
4.1 UART and USB Command packet processing procedure	10
4.2 UART data packet transmission.....	11
4.3 UART data packet reception	12
4.4 USB data transmission and reception	14
4.5 Automatic registration template processes.....	16
4.6 Automatic fingerprint verification process	17
5 Command format instruction.....	18
5.1 Command packet/data packet format	18
5.2 Command response.....	19
5.3 Module command set.....	21
5.3.1 Auto-registration template PS_AutoEnroll.....	21
5.3.2 Automatic fingerprint verification PS_AutoIdentify	25
5.3.3 Delete template PS_DeletChar	27
5.3.4 Empty fingerprint library PS_Empty	28
5.3.5 Cancel command PS_Cancel	28
5.3.6 Dormancy command PS_Sleep	29
5.3.7 Read the number of valid templates PS_ValidTemplateNum	30
5.3.8 Read the index table PS_ReadIndexTable (reservation)	30
5.3.9 Set password PS_SetPwd.....	31
5.3.10 Verify password PS_VfyPwd.....	32
5.3.11 Get the unique serial number of the chip	32
6 Application considerations.....	34
6.1 Power on	34
6.2 Power off	34

1 Overview

1.1 overview

JM-101 fingerprint module is a fingerprint processing module for integrating the light path and fingerprint processing part, has small volume, low power consumption, simple interface, high reliability, fast recognition and good adaptability for dry or wet finger, quick search speed of fingerprint.

When reading fingerprint image, it has sensitive reaction and judgment to dry or wet fingers, and obtains best image quality. Thus, it can be applied widely by most of people in the world.

It can also customize the adaptive function of learning and adjust the parameters according to the users' habits and climate change, so as to make a better match.

In the process of fingerprint identification, the new fingerprint feature can be extracted and integrated into the previous fingerprint feature, which can work better and better in practice.

The JM-101 module also has the function of sensing fingers, and when the finger presses on the fingerprint acquisition surface, the module Touch feet output high level.

The JM-101 module has USB and UART communication interface. This module is controlled by related command from the main equipment.

1.2 working principle

Through the optical imaging principle, the grain caused by uneven medial skin of fingers, can form a variety of fingerprint image. The skin texture on the pattern, breakpoints, and intersection are different, called "feature points" in information processing, the characteristics of each finger is different, that is to say, is the only, rely on this kind of uniqueness, we can match a person with his fingerprint , through his fingerprints and preserved one in advance, comparing the fingerprints for verifying his real identity.

The fingerprint identification system can collect, analyze and compare the fingerprint by means of special photoelectric conversion equipment and image processing technology. The system mainly

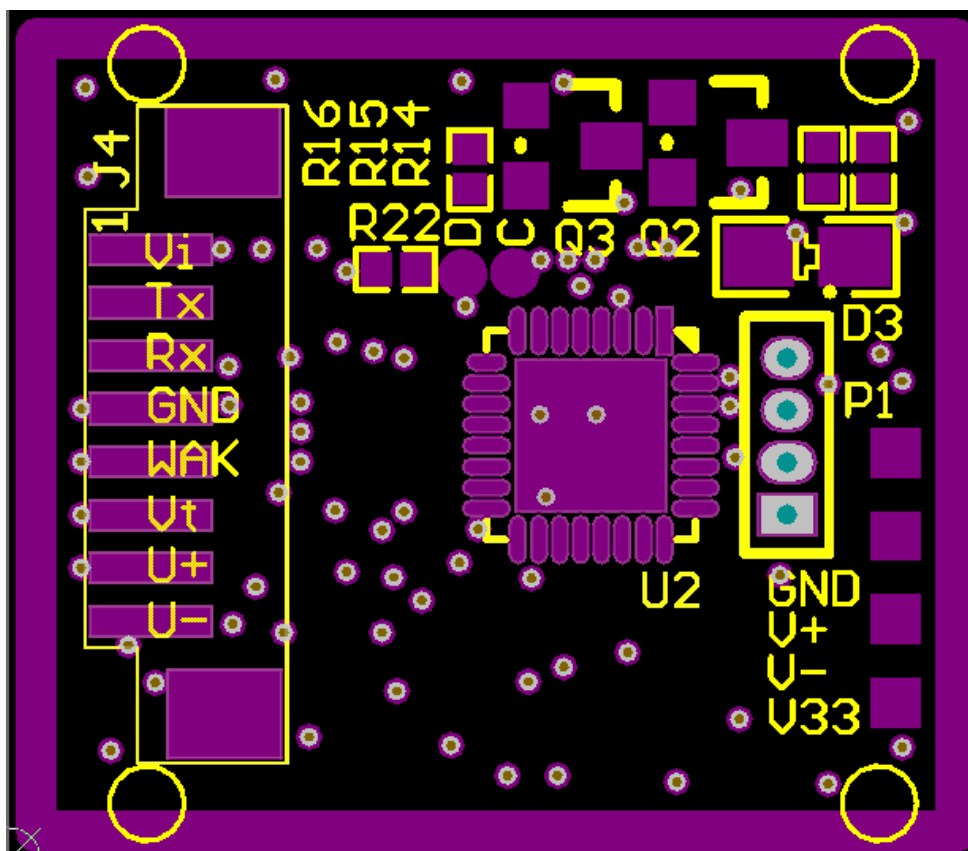
includes fingerprint image acquisition, fingerprint image processing, feature extraction, feature value comparison and matching.

1.3 Technical Parameter

Category	parameter		Min	Typical	Max	Remark
Module size	Length (mm)				23,3	
	Width (mm)				20.3	
	Height (mm)				48.1	
sensor	window	Length (mm)		18.2		
		Width (mm)		15.3		
	Valid pixel			256*288		
	Resolution (dpi)			500		
	Backlight color			blue		
arithmetic	Overall identification time(s)			<1		
	Search time(s)			<0.5		
	Storage capacity		0		150	
	FAR			<0.001%		
	FRR			<1%		
communication interface	UART	Baud rate bps		57600		
		Data bits		8		
		Stop bit		2		
		Check bit		None		
	USB			2.0FS		2.0 full speed
Electric parameter	Voltage (V)		3.0	3.3	3.6	
	Standby current (uA)		5		10	Sensing only part of the work
	Working current (mA)			40	60	
Working environment	Temperature (℃)		-20		60	
	Humidity (℃)		40%RH		85%RH	without condensation

2 Hardware interface

PIN No	Name	type	Function description
1	+3.3V	In	Module power positive input
2	TX	Out	Serial data output. TTL logic level.
3	RX	In	Serial data input. TTL logic level.
4	GND	—	Signal ground. Connected with power ground internally.
5	Touch	Out	sense signal output, high level effectively as default
6	TouchVin	In	Touch-sensitive power input, 3.3 v power supply
7	D+	—	USB D+
8	D-	—	USB D-



2.1 Hardware connection

It can communicate with the MCU of 3.3 V or other MCU through serial port. Considering the overall power consumption of the circuit, the fingerprint module only has two kinds of mode working and not working, no sleep or standby mode. When closing power input of fingerprint module, the fingerprint module will not work. When the MCU functions need to access the fingerprint module, input power to fingerprint module, fingerprint module works , complete the corresponding command. If you don't use the fingerprint module any more, cut off the power supply, the fingerprint module module turns into not working mode.

2.2 Touch-sensitive

The default access voltage of touch-sensitive input power supply is 3.3 V. The touch-sensitive circuit's overall power consumption is < 10 ua. The power supply of circuit can be sustained.

The high level of touch sensing signal output is effective. The touch signal output is low level when there is no finger put on acquisition window. When fingers are put on the acquisition window, the touch signal output changes into a high level. The signal can be connected to interrupt pin or IO port of MCU by a certain circuit. When the induction signal is received, the MCU is awoken to supply power for fingerprint module and then fingerprint module start to work (considering the overall power consumption of the circuit, the fingerprint module only has two kinds of status, work and not work, no sleep or standby mode.

Usually just supply power to touch circuit, the fingerprint module will not work when shutting down power supply. When touch signal is got, fingerprint module will be supplied power. Then the fingerprint module work for completing the corresponding command. If we don't use the fingerprint module any more, cut off the power supply of fingerprint module and then the fingerprint module get into not working mode.

2.3 Serial protocol

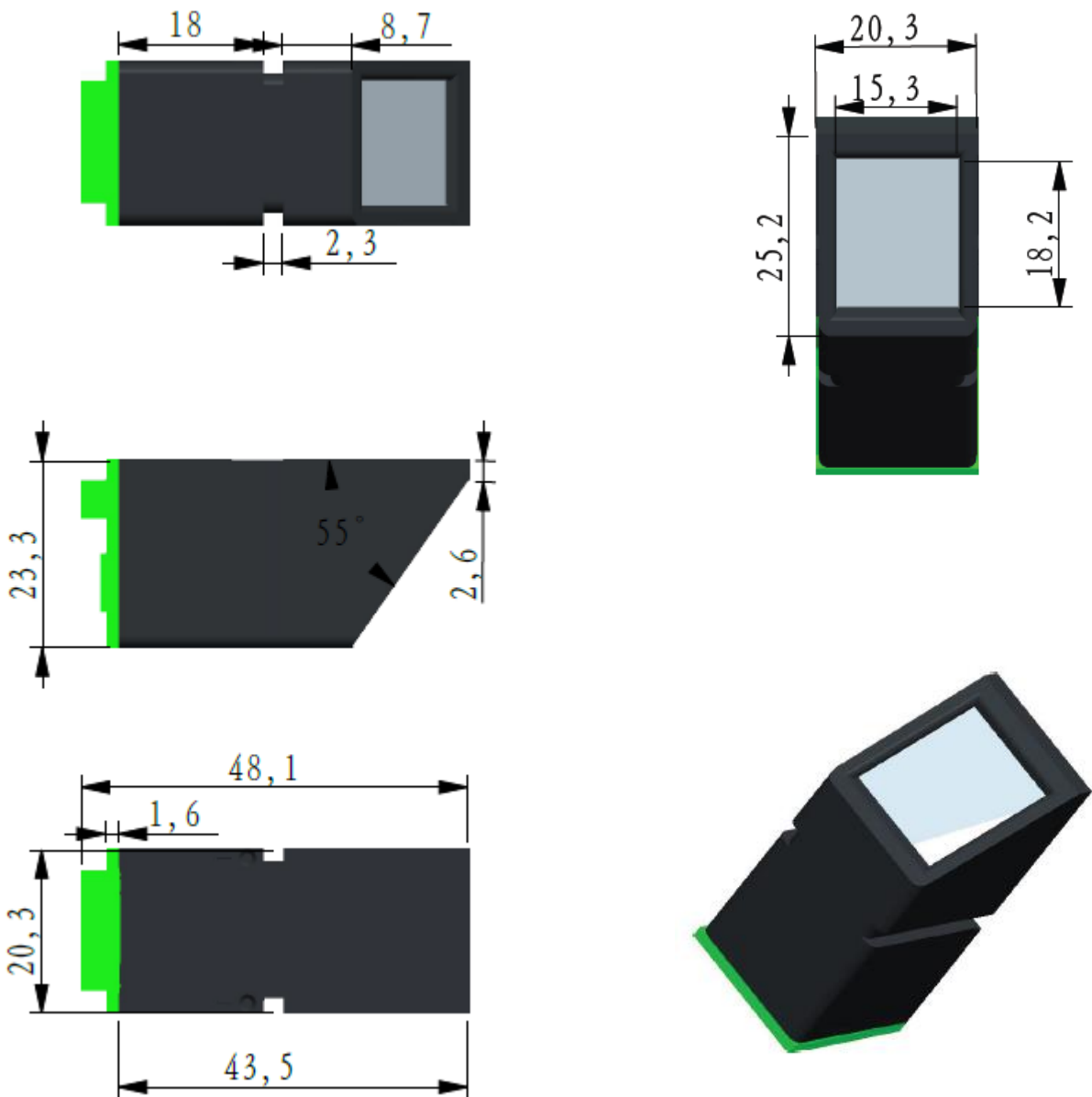
Semi-duplex asynchronous serial communication is adopted. The default baud rate is 57600bps, which can be customized to other baud rates. The transmitted frame format is 10 bits, a 0 level starting position, 8 data bits (low in front) and 2 stop bits, no check bits.

2.4 Time of electrical delay

After the module is supplied power, about 100mS time is required to initiate the work. During this time, the module can not respond to the command of upper machine .

3 Mechanical structure

3.1 JM-101 Shell Size



4 Functional implementation examples

4.1 UART and USB Command packet processing procedure

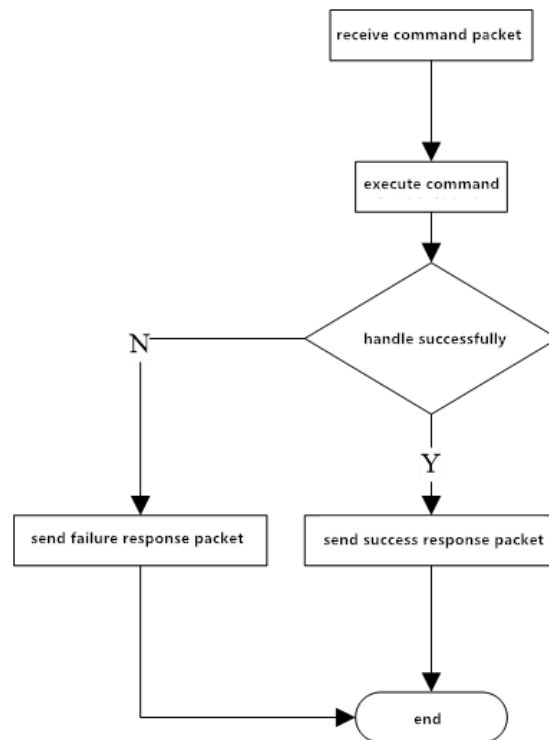


Figure 4-1 function implementation example 6: the processing of UART and USB command packages

4.2 UART data packet transmission

Before UART transmits data packets, it is first to receive the instruction packet of the data packet, then send the request package after the transmission is ready. Finally, begin to transmit the data packet. The data packets mainly include: packet header, chip address, packet identifier, packet length, data and checksum.

packet identifier of data packet is mainly divided into two types: 02H and 08H. 02H: data packet and follow-up packet . 08H: the last packet, the end packet. The data length is pre-set, mainly divided into: 32, 64, 128, and 256.

For example, the length of data to be transmitted is 1K bytes, and the data in the packet is 128 bytes in advance, so the data of 1K bytes is divided into 8 packets. Each packet includes: 2 bytes packet header, 4 bytes chip address, 1 bytes packet identifier, 2 bytes packet length, 128 bytes data, and 2 bytes checksum. The length of each packet is 139 bytes. In addition, in 8 packets, the packet identifier of first 7 packets is 02H, and the packet identifier of last packet is 08H. Finally, it is important to note that it will not be extended to 139 bytes in any other way if the length of packet does not reach 139 bytes.

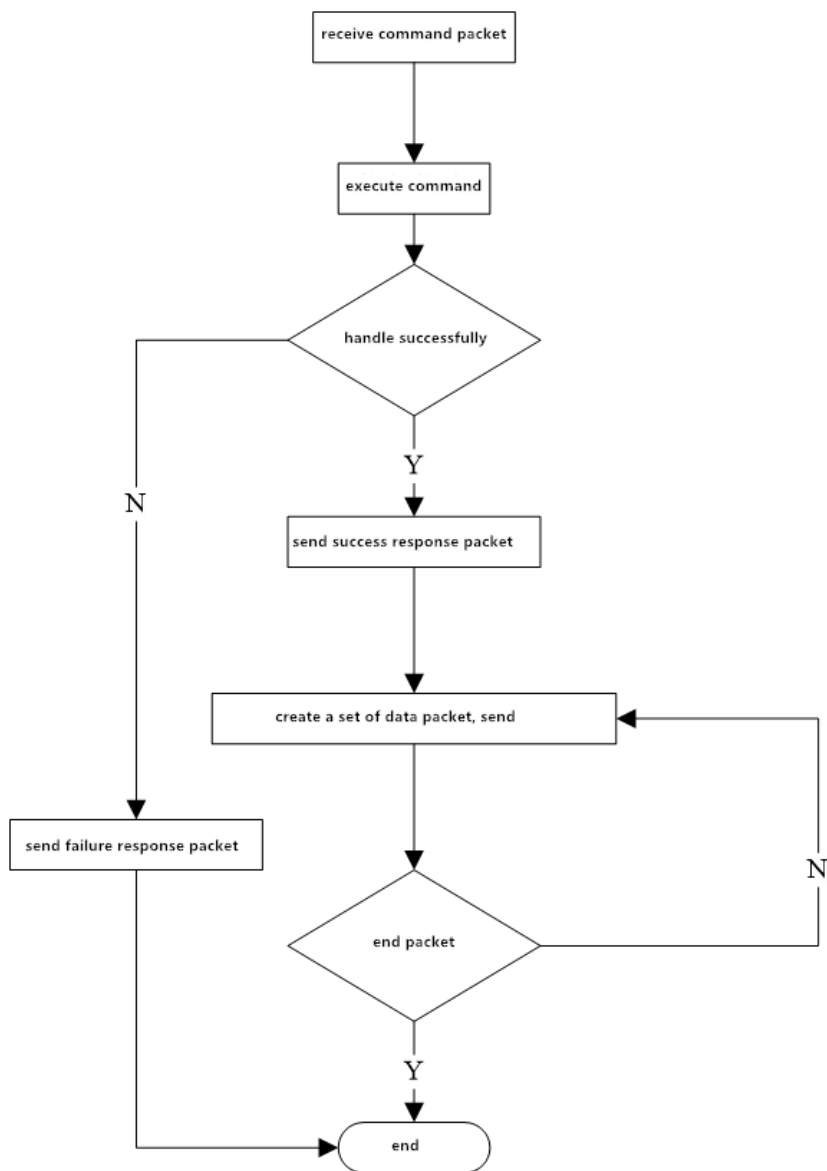


Figure 4-2 function implementation example 7: the sending process of UART packets

4.3 UART data packet reception

Before UART transmits data packets, it is first to receive the instruction packet of the data packet, then send the request package after the transmission is ready. Finally, begin to transmit the data packet. The data packets mainly include: packet header, chip address, packet identifier, packet length, data and checksum.

packet identifier is mainly divided into two types: 02H and 08H. 02H: data packet and follow-up packet . 08H: the last packet, the end packet. The data length is pre-set, mainly divided into: 32, 64, 128, and 256.

For example, the length of data to be transmitted is 1K bytes, and the data in the packet is 128 bytes in advance, so the data of 1K bytes is divided into 8 packets. Each packet includes: 2 bytes packet header, 4 bytes chip address, 1 bytes packet identifier, 2 bytes packet length, 128 bytes data, and 2 bytes checksum. The length of each packet is 139 bytes. In addition, in 8 packets, the packet identifier of first 7 packets is 02H, and the packet identifier of last packet is 08H. Finally, it is important to note that it will not be extended to 139 bytes in any other way if the length of packet does not reach 139 bytes.

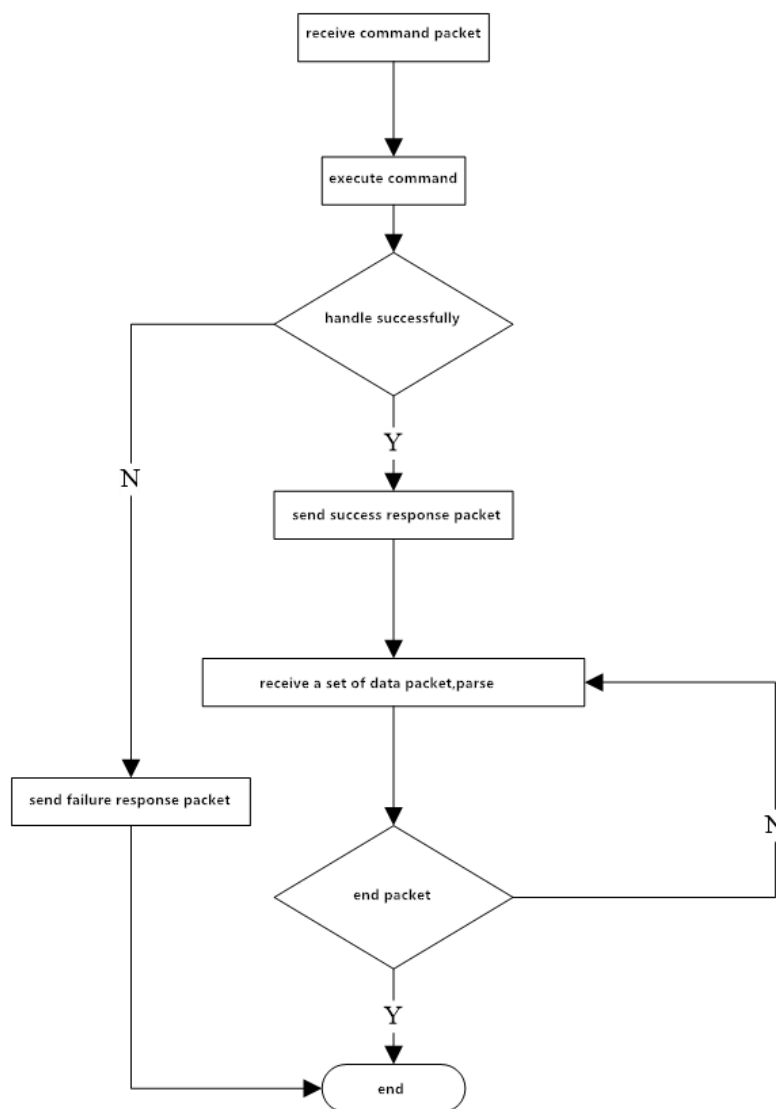


Figure 4-3 function implementation example 8: the receiving process of UART data packets

4.4 USB data transmission and reception

Before USB transmission, the instruction packet of data packet has to be received, and transmit successful request packet after the transmission is ready. USB data packets are only data, without packet header, chip address, packet identifier, packet length and checksum, and different from UART subcontract transmission, USB is the whole packet transmission.

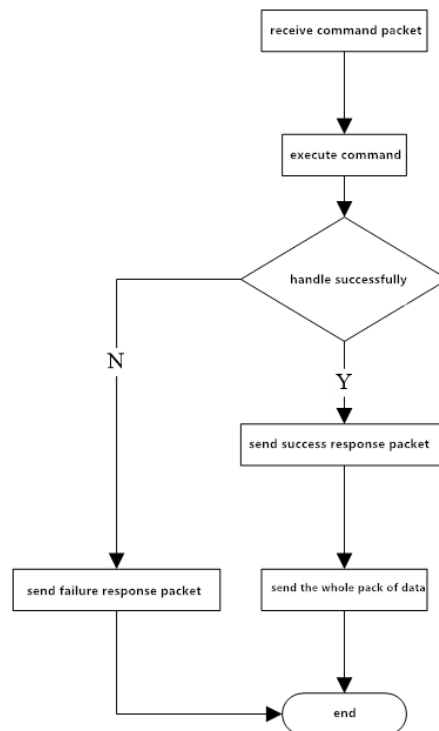


Figure 4-4 function implementation example 9: the sending process of a USB packet

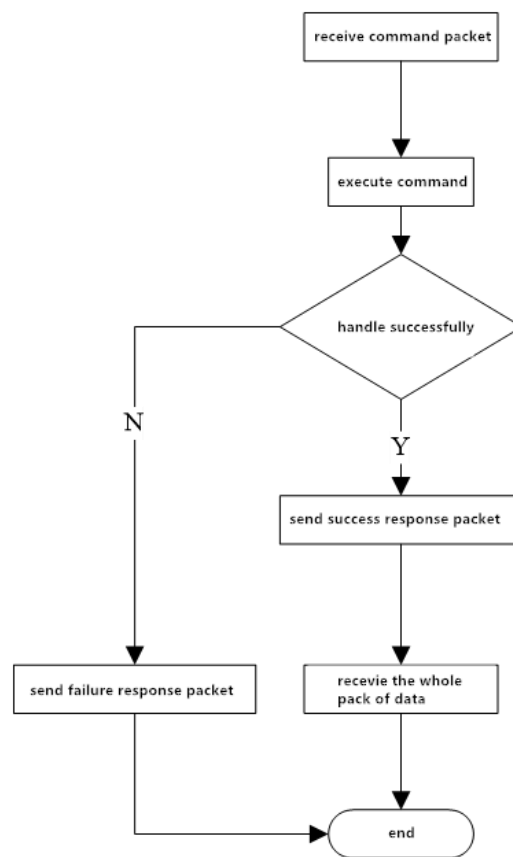


Figure 4-5 function implementation example 9: the sending process of a USB packet

4.5 Automatic registration template processes

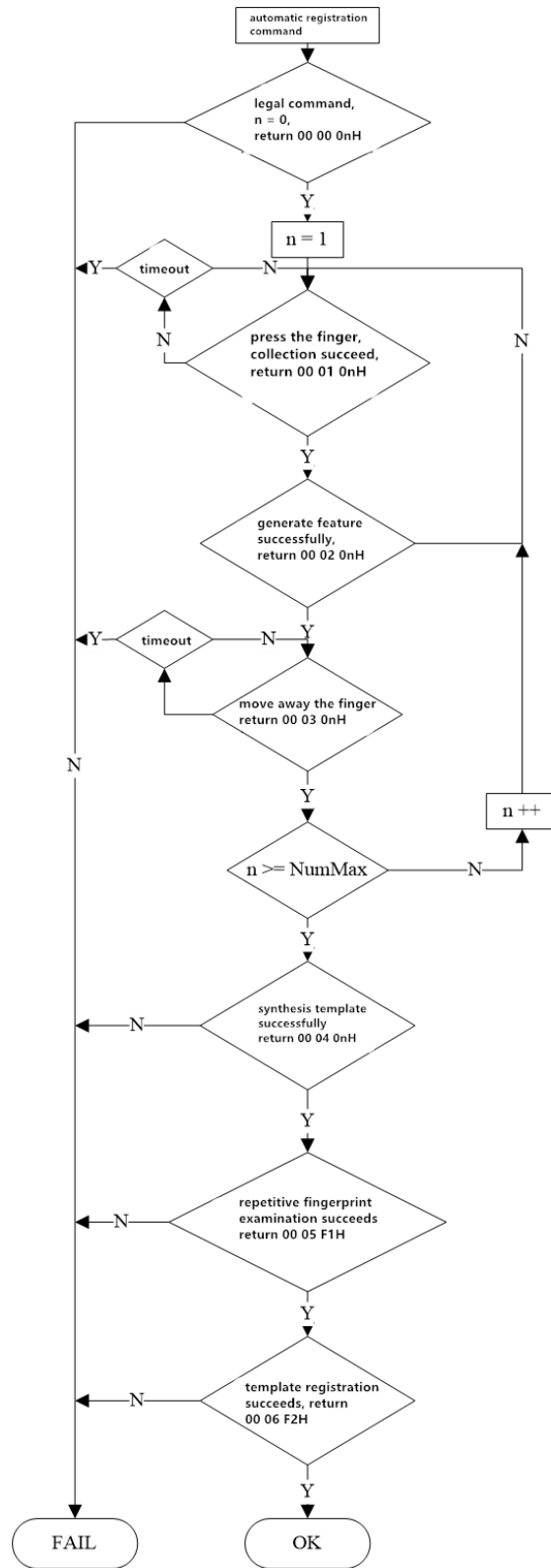


Figure 4-6 function implementation example 10: automatic registration template process

4.6 Automatic fingerprint verification process

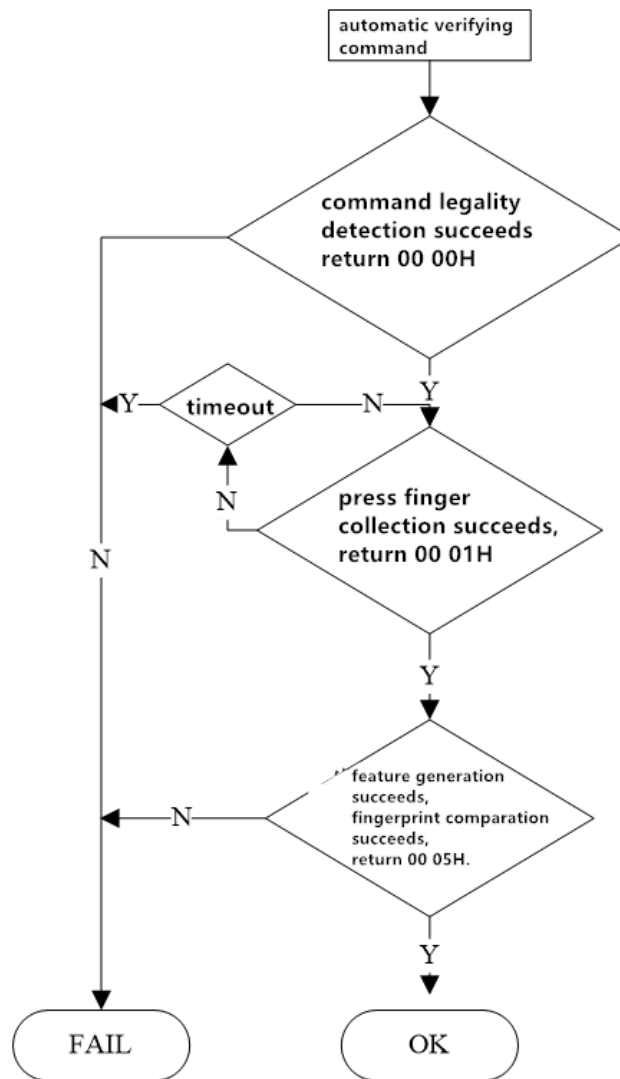


Figure 4-7 function implementation example 11: automatically validates the fingerprint process

5 Command format instruction

JM - 101 optical fingerprint module can form a complete fingerprint identification module after hooking up the necessary peripheral circuit (sensor, power supply, etc.). The module is always in a subordinate position (Slave mode), the host can ask the module to execute different kinds of function through different command. The instructions of the host, the response of the module, and the data exchange are all arranged in a specified format. The host must encapsulate the instructions or data to be sent in the following format, and you must also parse the received packets as followings.

5.1 Command packet/data packet format

The instructions/data packets fall into three categories:

Package id = 01: command packet.

packet identifier = 02: packets, and there are subsequent packets.

packet identifier = 08: the last packet, the end packet.

All packets should be added the packet header : 0xEF01.

- 01 command packet format :

Table 5-1 command package format

Name	Packet header	Chip address	packet identifier	Packet length	instruction	Parameter 1	...	Parameter N	Checksum
bytes	2 bytes	4 bytes	1 byte	2 bytes	1 byte				2 bytes
content	0xEF01	xxxx	01	N=					

- 02 data packet format:

Table 5-2 data packet format

Name	Packet header	Chip address	packet identifier	Packet length	Data	Checksum
bytes	2bytes	4bytes	1 byte	2 bytes	N bytes	2 bytes
Content	0xEF01	xxxx	02			

- 08 end packet format:

Table 5-3 end packet format

Name	Packet header	Chip address	packet identifier	Packet length	Data	Checksum
bytes	2bytes	4bytes	1 byte	2 bytes	N bytes	2 bytes
Content	0xEF01	xxxx	08			

◆ packets can't separate into execution process, must be followed instruction packet or reply packet.

◆ The data packet format of download and upload of is the same.

◆ packet length = The total bytes from packet length to the checksum (instructions, parameters or data), containing checksum, but not containing the bytes of the packet length itself.

◆ The checksum is all the bytes from the packet identifier to checksum. The carry that is more than 2 bytes will be ignored.

◆ The default chip address is 0xffffffff before it is generated. Once the upper computer generate chip addresses through the instruction, then all the data packets must transmit and receive in accordance with the generated address. The chip will reject the packet with the wrong address.

◆ Regard to multibyte, high byte first low byte last (such as 2 bytes, 00 06 means 0006, not 0600).

5.2 Command response

The response is to report the execution of command to the upper computer. And the response contains parameters and can be followed the subsequent packets. The upper computer can only confirm the SOC collection condition and the instruction execution condition after receiving the response packet of the SOC.

- Response packet format:

Table 5-4 response packet format

Name	Packet header	Chip address	packet identifier	Packet length	Confirmation code	Return parameter	Checksum
bytes	2bytes	4bytes	1 byte	2 bytes	1 byte	N bytes	2 bytes
Content	0xEF01		07				

◆ Confirmation code definition:

00H: the instructions are executed or OK;

01H: packet reception is error;

02H: no finger on the sensor;

03H: the fingerprint image is failed;

04H: the fingerprint image is too dry or too weak to be characteristic;

05H: the fingerprint image is too wet or too blurry to be characteristic;

06H: the fingerprint image is too messy to be characteristic;

07H: the fingerprint image is normal, but the feature is too few (or the area is too small) to be characteristic;

08H: the fingerprint does not match;

09H: can not find any fingerprint;

0aH: the feature merger fails;

0bH: When accessing to fingerprint database, the address sequence number exceeds the scope of the fingerprint library;

0cH: error or invalidation when reading the template from fingerprint library;

0dH: fail when uploading feature;

0eH: the module cannot receive the subsequent packets;

0fH: fail to upload image.

10H: the deletion of template failed;

11H: fail to clear the fingerprint database.

12H: no access to low-power status;

13H: wrong password.

14H: the system reset failed;

15H: There is no effective original picture for generating image in the buffer zone;

16H: online upgrade failed;

17H: the finger was not moved between or there is residual fingerprints between the two collections;

18H: error to read FLASH.

F0H: There are instructions for subsequent packet. When it is received correctly, reply 0xf0.

F1H: instructions for subsequent data packets, command envelope 0xf1 response;

f2H: When burning internal FLASH, checksum is error.

f3H: When burning internal FLASH, packet identifier is error.

f4H: When burning internal FLASH, packet length is error.

f5H: When burning internal FLASH, code length is error.

f6H: When burning internal FLASH, fail to burn.

19H: Undefined error.

1aH: Invalid register number;

-
- 1bH: Register setting content error number;
 - 1cH: Specified Error in notepad page code;
 - 1dH: Port operation failed;
 - 1eH: Automatic registration (enroll) fails;
 - 1fH: Full of fingerprint database ;
 - 20H—efH: Reserved。

The command can only be given to the module by the upper computer, and then the module reply.

The system will firstly check whether the default device handshake password has been changed or not after Power-on Reset. If it has not been modified, then the system think that it is not necessary to verify password, SOC directly enter into the normal working status. If it has been modified, you must first verify the device handshake password. SOC can work normally only after the verification is passed.

5.3 Module command set

5.3.1 Auto-registration template PS_AutoEnroll

- function description: one-stop fingerprint registration, containing fingerprint collection, generating characteristics, combination of template, storage of templates and so on.
- input parameters: ID number, input frequency, parameter
- returning parameters: confirm word, parameter
- instruction code: 31H
- instruction packet formats:

55 automatic registration template instruction packet format

Packet header	Chip address	packet identifier	Packet length	Command code	ID number	Input frequency	Parameter	Checksum
2 bytes	4bytes	1 byte	2 bytes	1byte	2 byte	1byte	2byte	2 bytes
0xEF01	xxxx	01H	0008H	31H	xxxxH	xxH	xxH	SUM

- Assist instruction:

ID number: High byte in front, low byte in the back. For example, input No. 1 fingerprint, it is 0001H.

Input times: 1byte, input 2 times, then it is 02H, input 4 times then it is 04H.

Parameter: the lowest bit is bit0.

Help:

ID number: the high byte first, and the low byte last. Such as fingerprint input 1, is 0001 h.

Input number: 1 byte, input 2 times, is 02 h, input 4 times is 4 h.

Parameters: the lowest bit is bit0.

1) bit0: backlight control bit of collecting image, 0 - LED always keeps on, 1 - LED turns off after the image is obtained successful.

2) bit1: preprocess control bits of collecting image , 0 to close the preprocess, 1 - to open the preprocess.

3) bit2: registration process, whether to ask module to return the current status in the key steps, 0 - require to return, 1 - does not require to return;

1) bit3: is it allowed to cover ID number, 0 - don't allow, 1 - allow;

2) bit4: if it is allowed the fingerprint to register control bits repeatedly, 0 - allowed, 1 - not allowed;

3) bit5: when registering, fingerprint acquisition process for many times, whether to ask finger to leave and then enter another fingerprint image acquisition, 0 - asked to leave. 1 - not asked to leave.

4) bit6~ bit15: reserved.

● Response packet format

Table 5-6 automatic registration template instruction normal process response packet format

Packet header	Chip address	packet identifier	Packet length	Confirmation code	Parameter 2 byte		Checksum	Remark
					Parameter 1	Parameter 2		
2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	1 byte	2 bytes	
0xEF01	xxxx	07H	5	xxH	0H	0H	sum	Directive legality detection: Legal /..
0xEF01	xxxx	07H	5	xxH	01H	1	sum	Acquisition results: Success/timeout
0xEF01	xxxx	07H	5	xxH	02H	1	sum	Result of generation characteristics: Success/failure
0xEF01	xxxx	07H	5	xxH	03H	1	sum	The finger goes away, 1st input success: Success/timeout

Packet header	Chip address	packet identifier	Packet length	Confirmation code	Parameter 2 byte		Checksum	Remark
					Parameter 1	Parameter 2		
						...		
0xEF01	xxxx	07H	5	xxH	01H	n	sum	Acquisition results: Success/timeout
0xEF01	xxxx	07H	5	xxH	02H	n	sum	Result of generation characteristics: Success/failure
0xEF01	xxxx	07H	5	xxH	04H	F0H	sum	Merge template
0xEF01	xxxx	07H	5	xxH	05H	F1H	sum	Registered test
0xEF01	xxxx	07H	5	xxH	06H	F2H	sum	Template storage result

- confirmation code, the return value of parameter 1 and 2

Table 5-7 automatic registration template response packet paraphrase check list

Confirmation code	paraphrase	Parameter 1	paraphrase	Parameter 2	paraphrase
00H	success	00H	Fingerprint validity detection	00H	Fingerprint validity detection
01H	failure	01H	acquisition of image	F0H	Merge template
07H	Failure to generate feature	02H	generate feature	F1H	Verify if the finger has been registered
0aH	failure to merge template	03H	Judge if finger leaves or not	F2H	store template
0bH	ID exceeds range	04H	Merge template	n	Current input in No. n
1fH	Fingerprint library is full	05H	registration check		
22H	Fingerprint	06H	store template		

	template is not empty	H			
25H	Incorrect input number set				
26H	timeout				
27H	Fingerprints already exist				

● Instruction explanation

- 1) If the specified ID number is invalid, then the confirmation code, parameter 1, and 2 return (directly described as return) : 0b 00 00H. Validity detection:
 - If the specified ID number is invalid, return: 0b 00 00H.
 - If input number configuration errors, then return 25 00 00H. If the fingerprint is not covered, it will return 1f 00 00H if the fingerprint database is full;
 - If the specified ID number exists template, then return 22 00 00H.
 - instruction validity test is successful, return 00 00 00H, and then enter into the fingerprint input for the first time.
- 2) wait for collecting image successfully (return 0001 0nH).
- 3) wait for generating feature successfully (00 02 0nH), and if it fails (0020nh), wait for collecting image successfully again.
- 4) wait for the finger to move away, if the first input succeeds (00 03 0nH), jump to step 2 after the finger is moved away, enter the next cycle, until n is the maximum number of times set for input. Note: if the fingers are not set to be moved away when inputting, then directly return the first time to input successfully , and skip to step 2. The last fingerprinted, no response of successful input after finger is moved away.
- 5) compose template, combining previously acquired finger features into a finger template, return 00 04 F0H if it succeeds. If it fails, then return 0A 04 F0H.
- 6) Examination of repeated fingerprint, check if a fingerprint just input is matched with already stored one or not (by setting the Parameter bit4 open or not). If there is same fingerprint, then return 27 05 F1H, end the process. If there is no the same fingerprint, return 00 05 F1H.
- 7) register the template data, if it is failure to store, then return 01 06 F2H, end the process. If it succeeds, return 00 06 F2H.
- 8) If the PS_Cancel command is received, terminate the command and return the response.

5.3.2 Automatic fingerprint verification PS_AutIdentify

- function description: automatic fingerprint collection, search target template or whole fingerprint template in the whole fingerprint database, and then return search results. If the score is higher than the maximum threshold when the target template is compared with the current collected template, and the target template is not complete, then the collected new feature will be updated to the blank area of the target template. One-stop search includes capturing images, generating features, searching fingerprints and so on.

- Input Parameter: security level, ID number
- Return Parameter: confirm word, page number (Matching fingerprint template)
- command code: 32H
- command packet format:

Table 5-1 automatic verifying fingerprint command packet format

Packet header	Chip address	Packet identifier	Packet length	Command code	Security level	ID number	Parameter	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 byte	2 byte	2 bytes
0xEF01	xxxx	01H	0008H	32H	xxH	xxxxH	xxxxH	xxxxH

- Assist instructions:

ID number : 2byte, big-endian. For example, when inputting No.1 fingerprint, it is 0001H. If the ID number is 0xFFFF, then do a 1: N search. Or do a 1:1 search.

Parameter: The lowest level is bit0.

- 1) bit0: control bit of the collection backlight, 0-LED is always on, and 1-LED turns off when the image is successfully obtained.
- 2) bit1: pretreatment control bit of collecting image, 0- close pretreatment, 1- open pretreatment.
- 3) bit2: in the process of registration, whether the module is required to return current status in the key step, 0- request return, 1- no return is requested.
- 4) bit3~bit15: reservation

- response packet format:

Table 5-2 automatic verifying fingerprint command response packet format

Packet header	Chip address	Packet identifier	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	xxxx	07H	0008H	xxH	00H	xxxxH	xxxxH	sum	Directive legality

Packet header	Chip address	Packet identifier	Packet length	confirmation code	Parameter	ID number	score	Checksum	Remark
									detection: Legal /..
0xEF01	xxxx	07H	0008H	xxH	01H	xxxxH	xxxxH	sum	Collection results: Success/timeout
0xEF01	xxxx	07H	0008H	xxH	05H	xxxxH	xxxxH	sum	Search results: Success/failure

- confirmation code、Parameter1 and Parameter2 return value

Table 5-10 automatic verification of fingerprint response packet paraphrase check list

confirmation code	paraphrase	Parameter	paraphrase
00H	Succeed	00H	Fingerprint validity detection
01H	Failure	01H	acquire image
07H	Failure to generate feature	05H	registered fingerprint comparison
09H	No fingerprints were found		
0bH	ID number exceeds range		
17H	Residual fingerprint		
23H	The fingerprint template is empty		
24H	Fingerprint library is empty		
26H	Timeout		
27H	Indicates fingerprint		

	already exists		
--	----------------	--	--

- Command explanation:

- 1) If the fingerprint library is empty, then confirmation code and parameter return (the followings directly describe as return) : 24 00H。 If specified ID number is invalid, then return 0b 00H。 If the registered template doesn't exist, then return 23 00H.
- 2) The command validity is detected successfully, return 00 00H, and enter into the fingerprint input.
- 3) During the timeout period set, if none of complete fingerprint input is finished, then return 26 00H and the process end.
- 4) Check the validity of fingerprint image input. If it is incorrect, wait for the next image collection.
- 5) If the fingerprint input is correct, then return 00 01H, i.e. the image is acquired successfully after the input of fingerprint.
- 6) If it is failure to generate the feature, then return 09 05H, the process end.
- 7) After the feature are generated successfully, the current collected fingerprint template is compared with the registered fingerprint template and return the results.If the comparison fails, then return 09 05H and the process ends. If the comparison is successful, return 00 05H, the correct ID number and the correct score.
- 8) If the FpCancel command is received, terminate the command and return response.

5.3.3 Delete template PS_DeletChar

- function description: delete N fingerprint templates starting from the specified ID number of flash database.
- Input Parameter: PageID (fingerprint database template number) , N (the number of deleted templates) .
- Return Parameter: confirm word
- command code: 0cH
- command packet format:

Table 5-3 delete template command packet format

Packet header	Chip address	Packet identifier	Packet length	Command code	page number	The number of deletion	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes	2bytes	2 bytes
0xEF01	xxxx	01H	0007H	0cH	PageID	N	sum

- response packet format:

Table 5-4 delete template command response packet format

Packet header	Chip address	Packet identifier	Packet length	confirmation code	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	xxxx	07H	0003H	xxH	sum

Note: confirmation code=00H indicates delete the template successfully.

confirmation code=01H indicates error to receive the packet.

confirmation code=10H indicates failure to delete the template.

Sum indicates Checksum.

5.3.4 Empty fingerprint library PS_Empty

- function description: delete all the fingerprint templates of the flash database.
- Input Parameter: none
- Return Parameter: confirm word
- command code: 0dH
- command packet format:

Table 5-5 empty fingerprint library command packet format

Packet header	Chip address	Packet identifier	Packet length	Command code	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	xxxx	01H	0003H	0dH	0011H

- response packet format:

Table 5-6 empty fingerprint library command response packet format

Packet header	Chip address	Packet identifier	Packet length	confirmation code	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	xxxx	07H	0003H	xxH	sum

Note: confirmation code=00H indicates empty successfully.

confirmation code=01H indicates error to receive the packet.

confirmation code=11H indicates failure to empty.

Sum indicates Checksum.

5.3.5 Cancel command PS_Cancel

- function description: cancel command
- Input Parameter: none

- Return Parameter: confirm word
- command code: 30H
- command packet format:

Table 5-7 cancel command packet format

Packet header	Chip address	Packet identifier	Packet length	Command code	Checksum
2 bytes	4bytes	1 byte	2bytes	1 byte	2bytes
0xEF01	xxxx	01H	0003H	30H	xxxxH

- response packet format:

Table 5-8 cancel command response packet format

Packet header	Chip address	Packet identifier	Packet length	confirmation code	Checksum
2 bytes	4bytes	1byte	2bytes	1byte	2bytes
0xEF01	xxxx	07H	03	xxH	sum

Note: confirmation code=00H indicates cancel the setting successfully.

confirmation code=01H indicates failure to cancel setting.

Sum indicates Checksum.

5.3.6 Dormancy command PS_Sleep

- function description: set sensor into dormancy mode.
- Input Parameter: None
- Return Parameter: confirm word
- command code: 33H
- command packet format:

Table 5-9 dormancy command packet format

Packet header	Chip address	Packet identifier	Packet length	Command code	Checksum
2 bytes	4bytes	1 byte	2bytes	1 byte	2bytes
0xEF01	xxxx	01H	0003H	33H	xxxxH

- response packet format:

Table 5-10 dormancy response packet format

Packet header	Chip address	Packet identifier	Packet length	confirmation code	Checksum
2 bytes	4bytes	1byte	2bytes	1byte	2bytes
0xEF01	xxxx	07H	03	xxH	sum

Note: confirmation code=00H indicates dormancy set successfully.

confirmation code=01H indicates failure to set dormancy.

Sum indicates Checksum.

5.3.7 Read the number of valid templates PS_ValidTemplateNum

- function description: read Number of valid templates。
- Input Parameter: none
- Return Parameter: confirm the word, Number of valid templates
- command code: 1dH
- command packet format:

Table 5-11 Read the number of valid templates command packet format

Packet header	Chip address	Packet identifier	Packet length	Command code	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	xxxx	01H	0003H	1dH	0021H

- response packet format:

Table 5-12 Read the number of valid templates command response packet format

Packet header	Chip address	Packet identifier	Packet length	confirmation code	Number of valid templates	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes	2 bytes
0xEF01	xxxx	07H	05H	xxH	ValidN	sum

Note: confirmation code=00H indicates reading successfully.

confirmation code=01H indicates error to receive the packet.

Sum indicates Checksum.

5.3.8 Read the index table PS_ReadIndexTable (reservation)

- function description: Read the index table of input template.
- Input Parameter: index table page, the page number 0,1,2,3 corresponds template from 0-256,256-512,512-768,768-1024 index, each one represents a template, 1 indicates that the template of corresponding storage area is already input, 0 means no input.
- Return Parameter: Confirm the word + index table information
- command code: 1fH
- command packet format:

Table 5-21 reading index table command packet format

Packet header	Chip address	Packet identifier	Packet length	Command code	page number	Checksum
---------------	--------------	-------------------	---------------	--------------	-------------	----------

Packet header	Chip address	Packet identifier	Packet length	Command code	page number	Checksum
2 bytes	4bytes	1 byte	2bytes	1 byte	1 byte	2bytes
0xEF01	xxxx	01H	0004H	1fH	0~3	xxxxH

- response packet format:

Table 5-22 reading index table command response packet format

Packet header	Chip address	Packet identifier	Packet length	confirmation code	index information	Checksum
2 bytes	4bytes	1byte	2bytes	1byte	32bytes	2bytes
0xEF01	xxxx	07H	23H	xxH	Index	sum

Note: confirmation code=00H indicates OK.

confirmation code=01H indicates error to receive packet.

Sum indicates Checksum.

5.3.9 Set password PS_SetPwd

- function description: set module handshake password。
- Input parameter: PassWord
- Return Parameter: confirm word
- command code: 12H
- command packet format:

Table 5-23 Set password command packet format

Packet header	Chip address	Packet identifier	Packet length	Command code	password	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	4 byte	2 bytes
0xEF01	xxxx	01H	0007H	12H	PassWord	sum

Note: the default value of module address is 0.

- response packet format:

Table 5-24 set password command response packet format

Packet header	Chip address	Packet identifier	Packet length	confirmation code	Checksum
2 bytes	4 byte	1 byte	2 bytes	1 byte	2 bytes
0xEF01	xxxx	07H	0003H	xxH	sum

Note: confirmation code=00H indicates OK.

confirmation code=01H indicates that error to received the packet.

Sum indicates Checksum。

5.3.10 Verify password PS_VfyPwd

- function description: validating module handshake password .
- input Parameter: PassWord
- return Parameter: confirm the word
- command code: 13H
- command packet format:

Table 5-25 validating password command packet format

Packet header	Chip address	Packet identifier	Packet length	Command code	password	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	4 byte	2 bytes
0xEF01	xxxx	01H	0007H	13H	PassWord	sum

- Response packet format:

Table 5-26 validating password command response packet format

Packet header	Chip address	Packet identifier	Packet length	confirmation code	Checksum
2 bytes	4 byte	1 byte	2 bytes	1 byte	2 bytes
0xEF01	xxxx	07H	0003H	xxH	sum

Note: confirmation code =00H indicates correct password verification;
Confirmation code =01H indicates that the collection is wrong;
Confirmation code =13H indicates that the password is incorrect;
Sum indicates Checksum.

5.3.11 Get the unique serial number of the chip

- function description: obtain chip unique serial number.
- input parameter: reserved.
- return parameter: confirm the word + unique serial number
- command code: 34H
- command packet formats:

Table 5-21 Obtaining the chip's only serial number command packet format

Packet header	Chip address	Packet identifier	Packet length	Command code	Parameter	Checksum
2 bytes	4bytes	1 byte	2bytes	1 byte	1 byte	2bytes
0xEF01	xxxx	01H	0004H	34H	0	0039H

- Response packet format:

Table 5-22 obtaining the chip's only sequence number command response packet format

Packet header	Chip address	packet identifier	Packet length	confirmation code	Unique sequence number	checksum
2 bytes	4bytes	1byte	2bytes	1byte	32bytes	2bytes
0xEF01	xxxx	07H	23H	xxH	SN	sum

Note: Confirmation code=00H means OK.

Confirmation code=01H means error to receive packet.

Sum means checksum.

6 Application considerations

6.1 Power on

When supplying power to the module, you must ensure that pins of UART_RX and UART_TX are low level, otherwise the module will not work properly. In practice, the lock chip should firstly supply power to the module, wait for 10ms, and then initialize UART, so that UART_RX and UART_TX become high level.

6.2 Power off

When cutting off the power of module, the pins of UART_RX and UART_TX should be set to output low level, wait for 10ms, and then the power of module will be down.