

# Fingerprint Identification Module

## User Manual

# I Introduction

## Operation Principle

Fingerprint processing includes two parts: fingerprint enrollment and fingerprint matching (the matching can be 1:1 or 1: N).

When enrolling, user needs to enter the finger two times. The system will process the two time finger images, generate a template of the finger based on processing results and store the template. When matching, user enters the finger through optical sensor and system will generate a template of the finger and compare it with templates of the finger library. For 1:1 matching, system will compare the live finger with specific template designated in the Module; for 1: N matching, or searching, system will search the whole finger library for the matching finger. In both circumstances, system will return the matching result, success or failure.

## II Main Parameters

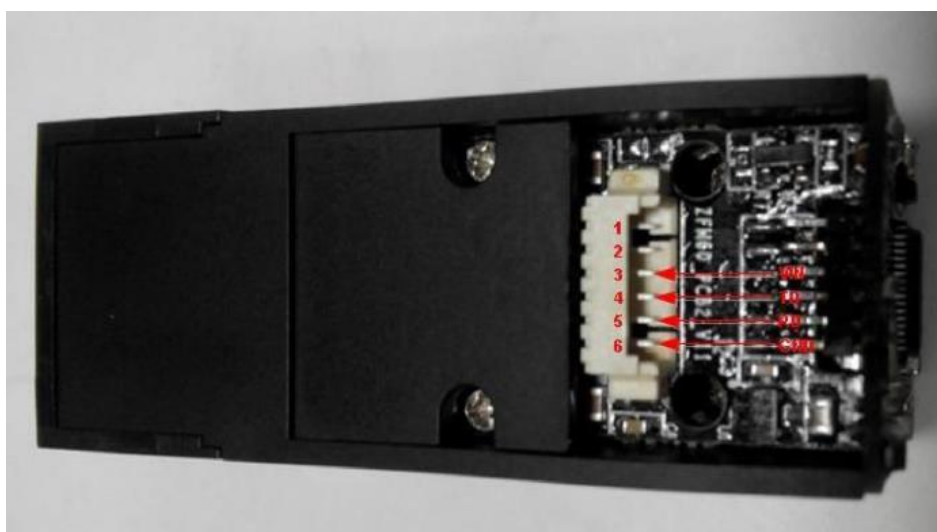
<b>Power</b>	DC 3.8V-7.0V	<b>Interface</b>	UART(TTL logical level)
<b>Working current</b>	Typical: <65mA Peak: <95mA	<b>Matching Mode</b>	1:1 and 1:N
<b>Baud rate</b>	(9600*N)bps, N=1~12 (default N=6)	<b>Character file size</b>	256 bytes
<b>Image acquiring time</b>	<1s	<b>Template size</b>	512 bytes
<b>Storage capacity</b>	1000	<b>Security level</b>	5 (1, 2, 3, 4, 5(highest))
<b>FAR</b>	<0.001%	<b>FRR</b>	<1.0%
<b>Average searching time</b>	< 1s (1:500)	<b>Window dimension</b>	14.5mm*19.4mm
<b>Working environment</b>	Temp: -20°C- +60°C	<b>Storage environment</b>	Temp: -40°C- +85°C
	RH: 40%-85%		RH: <85%
<b>Outline Dimention</b>	Integral type	54*20*20.5mm	

# III Hardware Interface

## 3.1 Connecting with upper computer

### 3.1.1 Serial Communication

Pin Nmuber	Name	Type	Function Description
1	Vtouch	in	Touch induction power input (cable color: blue)
2	Sout	out	Induction signal output (cable color:yellow)
3	Vin	in	Power input (cable color: red)
4	TD	out	Data output. TTL logical level (cable color: green)
5	RD	in	Data input. TTL logical level (cable color: white)
6	GND	—	Signal ground. Connected to power ground (cable color: black)



#### 3.1.1.1 Hardware connection

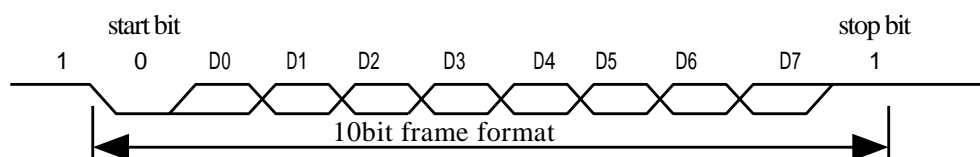
Via serial interface, the Module may communicate with MCU of 5V power: TD connects with RXD (receiving pin of MCU), RD connects with TXD (transferring pin of MCU).

Should the upper computer (PC) be in RS-232 mode, please add level converting circuit, like MAX232, between the Module and PC.

#### 3.1.1.2 Serial communication protocol

The mode is semiduplex synchronism serial communication. And the default baud rate is 57600bps. User may set the baud rate in 9600~115200bps.

Transferring frame format is 10 bit: the low-level starting bit, 8-bit data with the LSB first, and an ending bit. There is no check bit.



### 3.1.1.3 Reset time

At power on, it takes about 300ms for initialization. During this period, the Module can't accept commands for upper computer.

Module initialized immediately after sending a byte (0x55) to the host computer, said module can already work normally and the receiving host computer instruction.

### 3.1.1.4 Electrical parameter (All electrical level takes GND as reference)

#### 1. Power supply

Item	Parameter			Unit	Note
	Min	Typ	Max		
Power Voltage (Vin)	3.8		7.0	V	Normal working value.
Maximum Voltage (Vin <sub>max</sub> )	-0.3		9.0	V	<b>Exceeding the Maximum rating may cause permant harm to the Module.</b>
Operation Current (I <sub>cc</sub> )	90	110	130	mA	
Peak Current (I <sub>peak</sub> )			130	mA	

#### 2. TD (output, TTL logic level)

Item	Condition	Parameter			Unit	Note
		Min	Typ	Max		
V <sub>OL</sub>	I <sub>OL</sub> =-4mA			0.4	V	Logic 0
V <sub>OH</sub>	I <sub>OH</sub> = 4mA	2.4		3.3	V	Logic 1

#### 3. RD (input, TTL logic level)

Item	Condition	Parameter			Unit	Note
		Min	Typ	Max		
V <sub>IL</sub>				0.6	V	Loigc 0
V <sub>IH</sub>		2.4			V	Logic 1
I <sub>IH</sub>	V <sub>IH</sub> =5V		1		mA	
	V <sub>IH</sub> =3.3V		30		uA	
V <sub>I<sub>max</sub></sub>		-0.3		5.5	V	<b>Maximum input voltage</b>

# IV System Resources

To address demands of different customer, Module system provides abundant resources at users use.

## 4.1 Notepad

The system sets aside a 512-bytes memory (16 pages\* 32 bytes) for user's notepad, where data requiring power-off protection can be stored. The host can access the page by instructions of PS\_WriteNotepad and PS\_ReadNotepad.

Note: when write on one page of the pad, the entire 32 bytes will be written in wholly covering the original contents.

## 4.2 Buffer

There are an image buffer and two 512-byte-character-file buffer within the RAM space of the module. Users can read & write any of the buffers by instructions.

Note: Contents of the above buffers will be lost at power-off.

### 4.2.1 Image buffer

ImageBuffer serves for image storage and the image format is 256\*288 pixels.

When transferring through UART, to quicken speed, only the upper 4 bits of the pixel is transferred (that is 16 grey degrees). And two adjacent pixels of the same row will form a byte before the transferring. When uploaded to PC, the 16-grey-degree image will be extended to 256-grey-degree format. That's 8-bit BMP format.

### 4.2.2 Character file buffer

Character file buffer, CharBuffer1, CharBuffer2, can be used to store both character file and template file.

## 4.3 Fingerprint Library

System sets aside a certain space within Flash for fingerprint template storage, that's fingerprint library. Contents of the library remain at power off.

Capacity of the library changes with the capacity of Flash, system will recognize the latter automatically. Fingerprint template's storage in Flash is in sequential order. Assume the fingerprint capacity N, then the serial number of template in library is 0, 1, 2, 3.....N-2, N-1. User can only access library by template number.

## 4.4 System Configuration Parameter

To facilitate users developing, Module opens part system parameters for use. And the basic instructions are SetSysPara & ReadSysPara. Both instructions take Parameter Number as parameter.

When upper computer sends command to modify parameter, Module first responses with original configurations, then performs the parameter modification and writes configuration record into Flash. At the next startup, system will run with the new configurations.

#### 4.4.1 Baud rate control (Parameter Number: 4)

The Parameter controls the UART communication speed of the Module. Its value is an integer N, N= [1, 12]. Corresponding baud rate is 9600\*N bps.

#### 4.4.2 Security Level (Parameter Number: 5)

The Parameter controls the matching threshold value of fingerprint searching and matching. Security level is divided into 5 grades, and corresponding value is 1, 2, 3, 4 and 5. At level 1, FAR is the highest and FRR is the lowest; however at level 5, FAR is the lowest and FRR is the highest.

#### 4.4.3 Data package length (Parameter Number: 6)

The parameter decides the max length of the transferring data package when communicating with upper computer. Its value is 0, 1, 2, 3, corresponding to 32 bytes, 64 bytes, 128 bytes, 256 bytes respectively.

### 4.5 System status register

System status register indicates the current operation status of the Module. Its length is 1 word, and can be read via instruction *ReadSysPara*. Definition of the register is as follows:

Bit Num	15	4	3	2	1	0
Description	Reserved		ImgBufStat	PWD	Pass	Busy

Note:

- Busy: 1 bit. 1: system is executing commands; 0: system is free;
- Pass: 1 bit. 1: find the matching finger; 0: wrong finger;
- PWD: 1 bit. 1: Verified devices handshaking password.
- ImgBufStat: 1 bit. 1: image buffer contains valid image.

### 4.6 Module password

The default is 0x00000000, If the default password is not modified; If be modified through UART communication or password, the first instruction is the host computer and the communication module must be verify the password, only the password verification through, module can enter the normal working state, receiving other instructions (That is, the serial communication must first perform a handshake signal processing) . Password modification, new password stored in Flash, power cuts are still preserved (The modified password cannot be acquired through communication instructions, such as accidentally forgetting the modules cannot communicate, please kindly with) .

### 4.7 Module address

Each module has an identifying address. When communicating with upper computer, each instruction/data is transferred in data package form, which contains the address item. Module system only responds to data package whose address item value is the same with its identifying address.

The address length is 4 bytes, and its default factory value is 0xFFFFFFFF. User may modify the address via instruction *SetAdder*. The new modified address remains at power off.

# V Communication Protocol

## 5.1 Data package format

When communicating, the transferring and receiving of command/data/result are all wrapped in data package format.

### Data package format

Header	Adder	Package identifier	Package length	Package content (instruction/data/Parameter)	Checksum
--------	-------	--------------------	----------------	--	----------

### Definition of Data package

Name	Symbol	Length	Description	
Header	START	2 bytes	Fixed value of EF01H; High byte transferred first.	
Adder	ADDR	4 bytes	Default value is 0xFFFFFFFF, which can be modified by command. High byte transferred first and at wrong adder value, module will reject to transfer.	
Package identifier	PID	1 byte	01H	Command packet;
			02H	Data packet; Data packet shall not appear alone in executing process, must follow command packet or acknowledge packet.
			07H	Acknowledge packet;
			08H	End of Data packet.
Package length	LENGTH	2 bytes	Refers to the length of package content (command packets and data packets) plus the length of Checksum (2 bytes). Unit is byte. Max length is 256 bytes. And high byte is transferred first.	
Package contents	DATA	—	It can be commands , data , command's parameters, acknowledge result, etc. (fingerprint character value, template are all deemed as data);	
Checksum	SUM	2 bytes	The arithmetic sum of package identifier, package length and all package contents. Overflowing bits are omitted. High byte is transferred first.	



## 5.2 Check and acknowledgement of data package

**Note: Commands shall only be sent from upper computer to the Module, and the Module acknowledges the commands.**

Upon receipt of commands, Module will report the commands execution status and results to upper computer through acknowledge packet. Acknowledge packet has parameters and may also have following data packet. Upper computer can't ascertain Module's package receiving status or command execution results unless through acknowledge packet sent from Module. Acknowledge packet includes 1 byte confirmation code and maybe also the returned parameter.

*Confirmation code definition is:*

1. 00h: command execution complete;
2. 01h: error when receiving data package;
3. 02h: no finger on the sensor;
4. 03h: fail to enroll the finger;
5. 06h: fail to generate character file due to the over-disorderly fingerprint image;
6. 07h: fail to generate character file due to weakness of character point or over-smallness of fingerprint image
7. 08h: finger doesn't match;
8. 09h: fail to find the matching finger;
9. 0Ah: fail to combine the character files;
10. 0Bh: addressing PageID is beyond the finger library;
11. 0Ch: error when reading template from library or the template is invalid;
12. 0Dh: error when uploading template;
13. 0Eh: Module can't receive the following data packages.
14. 0Fh: error when uploading image;
15. 10h: fail to delete the template;
16. 11h: fail to clear finger library;
17. 15h: fail to generate the image for the weakness of valid primary image;
18. 18h: error when writing flash;
19. 1Ah: invalid register number;

# VI Module Instruction System

## 6.1 System-related instructions

### 6.1.1 Verify password :VfyPwd

Description: The password verification module (Serial communication must be shake)

Input Parameter: control code 0

Return Parameter: confirmation code;

Instruction code: 13H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	4byte	2 bytes
Header	Chip address	Package identifier	Package length	Instruction code	Control code	Checksum
EF01H	xxxx	01H	0007H	13H	0	001BH

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Chip address	Package identifier	Package length	Confirmation code	Checksum
EF01H	xxxx	07H	0003H	xxH	sum

Note: Confirmation code=00H: Port operation complete;

Confirmation code=01H: error when receiving package;

Confirmation code=13H: fail to operate the communication port;

### 6.1.2 Set password : SetPwd

Description: Module password settings. (Refer to 4.6 for more information)

Input Parameter: PassWord;

Return Parameter: Confirmation code

Instruction code: 12H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	4byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	PassWord	Checksum
EF01H	xxxx	01H	0007H	12H	PassWord	sum

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1	2 bytes
Header	Module address	Package identifier	Package length	Confirmation	Checksum
EF01H	xxxx	07H	0003H	xx	Sum

Note: Confirmation code=00H: OK;

Confirmation code=01H: error when receiving package;

### 6.1.3 Set Module address: SetAdder

Description: Set Module address.

Input Parameter: New Module address

Return Parameter: Confirmation code (1 byte)

Instruction code: 15H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	4 bytes	2 bytes
Header	Original Module address	Package identifier	Package length	Instruction code	New Module address	Checksum
EF01H	xxxx	01H	0007H	15H	xxxx	sum

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	New Module address	Package identifier	Package length	Confirmation code	Checksum
EF01H	xxxx	07H	0003H	xxH	Sum

Note: Confirmation code=00H: address setting complete;

Confirmation code=01H: error when receiving package;

### 6.1.4 Set module system's basic parameter: SetSysPara

Description: Operation parameter settings. (Refer to 4.4 for more information)

Input Parameter: Parameter number+ Contents;

Return Parameter: Confirmation code (1 byte)

Instruction code: 0eH

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1byte	1byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Parameter number	Contents	Checksum
EF01H	xxxx	01H	0005H	0eH	4/5/6	xx	sum

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
EF01H	xxxx	07H	0003H	xxH	Sum

Note: Confirmation code=00H: parameter setting complete;

Confirmation code=01H: error when receiving package;

Confirmation code=1aH: wrong register number;

Table6.1 Parameter number+ Contents

Name	Parameter number	Contents
Baud rate	4	N(N=1~12, Corresponding baud rate is 9600*N bps.)
Security level	5	N(1、 2、 3、 4、 5)
Data package length	6	N(0、 1、 2、 3, corresponding length is 32、 64、 128、 256 (bytes) )

### 6.1.5 Read system Parameter: ReadSysPara

Description: Read Module's status register and system basic configuration parameters; ( Refer to 4.4 for system configuration parameter and 4.5 for system status register) .

Input Parameter: none

Return Parameter: Confirmation code (1 byte) + basic parameter (16bytes)

Instruction code: 0Fh

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Checksum
EF01H	xxxx	01H	0003H	0fH	0013H

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	16 bytes	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Basic parameter list	Checksum
EF01H	xxxx	07H	0013H	xxH	See following Table6.2	sum

Note: Confirmation code=00H: read complete;

Confirmation code=01H: error when receiving package;

**Table6.2 system basic parameters**

Name	Description	Offset (word)	Size (word)
Status register	Contents of system status register	0	1
System identifier code	Fixed value: 0x0000	1	1
Finger library size	Finger library size	2	1
Security level	Security level (1, 2, 3, 4, 5)	3	1
Device address	32-bit device address	4	2
Data packet size	Size code (0, 1, 2, 3)	6	1
Baud settings	N (baud = 9600*N bps)	7	1

### 6.1.6 Read the fingerprint template index table: ReadConList

Description: Reading fingerprint template index table and each time the most read 256 fingerprint template.

Input Parameter: Index page=0~3.

Index page 0 representative read 0 ~ 255 fingerprint template index table

Index page 1 representative read 256 ~ 511 fingerprint template index table

Index page 2 representative read 512 ~ 767 fingerprint template index table

Index page 3 representative read 768 ~ 1024 fingerprint template index table

Return Parameter: Confirmation code (1 byte) + template index table

Instruction code: 1fH

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Index page	Checksum
EF01H	xxxx	01H	0004H	1FH	0/1/2/3	sum

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	32 bytes	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Index table	Checksum
EF01H	xxxx	07H	0023H	xxH	See following Table6.3	sum

- 1: Confirmation code =0x00: Read index table success;  
Confirmation code =0x01: error when receiving package;
- 2: Every time the most read 256 fingerprint template index data, the data is insufficient 256 bit "0".
- 3: Index table data structure: each of the 8 as a group, and each group consists of beginning a high output.

**Table6.3 Index table data structure**

Order of transmission	From low to high byte output, and each byte by beginning a high output.								
least significant byte	Template Number	7	6	5	4	3	2	1	0
	Template index table data	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
Template Number	Template Number	15	14	13	12	11	10	9	8
	Template index table data	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
...	...	...							
Most significant byte	Template Number	255	254	253	252	251	250	249	248
	Template index table data	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

Note: the index table data "0" on behalf of the corresponding position without a valid template; "1" represents the corresponding to the location of the effective template.

### 6.1.7 Read valid template number: TemplateNum

Description: read the current valid template number of the Module

Input Parameter: none

Return Parameter: Confirmation code (1 byte), template number N

Instruction code: 1dH

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Checksum
EF01H	xxxx	01H	0003H	1DH	0021H

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Template number	Checksum
EF01H	xxxx	07H	0005	xxH	N	sum

Note: Confirmation code=00H: read complete;

Confirmation code=01H: error when receiving package;

## 6.2 Fingerprint-processing instructions

### 6.2.1 To collect finger image: GenImg

Description: detecting finger and store the detected finger image in ImageBuffer while returning successful confirmation code; If there is no finger, returned confirmation code would be "can't detect finger".

Input Parameter: none

Return Parameter: Confirmation code (1 byte)

Instruction code: 01H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Checksum
EF01H	xxxx	01H	0003H	01H	0005H

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
EF01H	xxxx	07H	0003H	xxH	Sum

Note: Confirmation code=00H: finger collection success;  
Confirmation code=01H: error when receiving package;  
Confirmation code=02H: can't detect finger;  
Confirmation code=03H: fail to collect finger;

### 6.2.2 Open the fingerprint lighting background LED : OpenLED

Description: Open the fingerprint lighting background LED

Input Parameter: none

Instruction code: 50H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Checksum
EF01H	xxxx	01H	0003H	50H	0054H

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
EF01H	xxxx	07H	0003H	xxH	sum

Confirmation code=00H: operation success;  
Confirmation code =others: operation failed

### 6.2.3 Close the fingerprint lighting background LED : CloseLED

Description: Close the fingerprint lighting background LED

Input Parameter: none

Instruction code: 51H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Checksum
EF01H	xxxx	01H	0003H	51H	0055H

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
EF01H	xxxx	07H	0003H	xxH	sum

Confirmation code=00H: operation success;  
Confirmation code =others: operation failed

## 6.2.4 Fingerprint get image free lighting : GetImageFree

Description: Fingerprint get image free lighting

Input Parameter: none

Instruction code: 52H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Checksum
EF01H	xxxx	01H	0003H	52H	0056H

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
EF01H	xxxx	07H	0003H	xxH	sum

Confirmation code =00H: finger collection success;

Confirmation code =01H: error when receiving package;

Confirmation code =02H: sensor has no finger;

Confirmation code =03H: fail to collect finger

## 6.2.5 Handshake : GetEcho

Description: Instructions to the module to send to shake hands, if the module is working correctly, will return to the confirmation code 0x55, the computer can continue to send instructions to the module; if the confirmation code for other or no response, said equipment abnormal.

Input Parameter: none

Instruction code: 53H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Checksum
EF01H	xxxx	01H	0003H	53H	0057H

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
EF01H	xxxx	07H	0003H	xxH	sum

Confirmation code=55H: The equipment is normal, can receive the command;

Confirmation code = Other or no response, said equipment abnormal.

In addition, module after power on automatically send 0x55 as handshake marks, MCU detected after 0x55, can immediately send the command to enter the working state.

## 6.2.6 Auto-login : AutoLogin

Description: Send the instruction, can make the module automatically complete the image acquisition, generation characteristics, synthetic template and the stored template work, To the four command line " To collect finger image (GenImg)", "To generate character file from image (Img2Tz)", "To generate template (RegModel) ", "To store template (Store)" into an instruction to complete.

Input Parameter: Fingerprint wait time + Number of times for pressing the fingerprint +Stored sequence number

Instruction code: 54H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Fingerprint wait time	Number of times for pressing the fingerprint	Stored sequence number	Repeated registration mark	Checksum
EF01H	xxxx	01H	0003H	54H	xxH	2/3	xxxx	0/1	sum

1. The fingerprint wait time is to wait the longest finger presses each image acquisition, if there is no finger presses on this parameter setting time, is that there is no finger. The domain values range from 1 to 255, the higher the value, the more time. The 70 series, usually taken as 54 (36H), corresponding to a time of 3.5 seconds, the other time intervals are listed in the following table:

Fingerprint wait time	Real time interval (s)	Fingerprint wait time	Real time interval (s)
31(1fH)	2	62(3eH)	4
38(26H)	2.5	69(45H)	4.5
46(2eH)	3	77(4dH)	5
54(36H)	3.5	85(55H)	5.5

2. Number of times for pressing the fingerprint is to Press the number of fingers to confirm the registration fingerprint, only the value of 2 or 3. Value of 2 represents the two press fingerprint recognition, value of 3 represents the 3 press fingerprint recognition.

3. Number of times for pressing the fingerprint is 2, the command will gather two fingerprint to register as a template, if collect the fingerprint success for the first time will sending a response code 56H (PS\_AUTOLOGIN\_OK1), then continue the second fingerprint process. Number of times for pressing the fingerprint is 3, the command will gather three fingerprint to register as a template, if collect the fingerprint success for the first time will sending a response code 56H (PS\_AUTOLOGIN\_OK1), and if collect the fingerprint success for the second time will sending a response code 57H (PS\_AUTOLOGIN\_OK2), then continue the third fingerprint process.

4. Repeated registration mark is set whether to allow repeated registration. 0 representative does not allow duplicate registration, i.e. if the current registration finger has been registered in the fingerprint database, then this will no longer register. 1 representative allows repeated registration, namely the current registration finger regardless of the fingerprint database whether registered, the registration of all.

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
EF01H	xxxx	07H	0003H	xxH	sum

Confirmation code =00H: auto Login success

Confirmation code =02H: sensor has no finger;

Confirmation code=06H: fail to generate character file due to the over-disorderly fingerprint image;

Confirmation code=07H: fail to generate character file due to lackness of character point or over-smallness of fingerprint image;

Confirmation code =0aH: fail to combine the character files. That's, the character files don't belong to one finger;

Confirmation code =0bH: the stored sequence number exceeds the effective range;

Confirmation code =56H: the first finger collection success;

Confirmation code =57H: the Second finger collection success;

Confirmation code =24H: failure due to repeated registration (That is, the current registered fingerprint in fingerprint database already exists)



## 6.2.7 Auto-Search : AutoSearch

**Description:** Send the instruction, can make the module automatically complete the image acquisition, generation characteristics and search fingerprint in the fingerprint template library work. To the three command line " To collect finger image (GenImg)", "To generate character file from image (Img2Tz)", " To search finger library (Search) " into an instruction to complete.

**Input Parameter:** Fingerprint wait time + Start page number + Search number.

**Instruction code:** 55H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Fingerprint wait time	Start page number	Search number	Checksum
EF01H	xxxx	01H	0008H	55H	xxH	xxxx	xxxx	sum

The fingerprint wait time is to wait the longest finger presses each image acquisition, if there is no finger presses on this parameter setting time, is that there is no finger. The domain values range from 1 to 255, the higher the value, the more time. The 70 series, usually taken as 54 (36H), corresponding to a time of 3.5 seconds, the other time intervals are listed in the following table:

Fingerprint wait time	Real time interval (s)	Fingerprint wait time	Real time interval (s)
31(1fH)	2	62(3eH)	4
38(26H)	2.5	69(45H)	4.5
46(2eH)	3	77(4dH)	5
54(36H)	3.5	85(55H)	5.5

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes	2 bytes	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Number	score	Checksum
EF01H	xxxx	07H	0007H	xxH	xxxx	xxxx	sum

Confirmation code =00H: search success;

Confirmation code =09H: search failed;

Confirmation code =02H: sensor has no finger;

Confirmation code=06H: fail to generate character file due to the over-disorderly fingerprint image;

Confirmation code=07H: fail to generate character file due to lackness of character point or over-smallness of fingerprint image;

Confirmation code =22H: residual fingerprint;

Confirmation code =23H: The specified interval does not exist an effective fingerprint template

## 6.2.8 Search fingerprints (With residual judgment) : SearchResBack

**Description:** to search the whole or part finger library for the template that matches the one in CharBuffer1 or CharBuffer2. When found, PageID will be returned. This command with Search (Instruction code = 04H) difference is for the remaining fingerprint return code is different, SearchResBack detected residual return code 22H, and the Search command detection residual return code 09H

**Input Parameter:** BufferID + StartPage + PageNum

**Return Parameter:** Confirmation code + Number (Match the fingerprint template)

**Instruction code:** 56H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Buffer number	StartPage	Page number	Checksum
EF01H	xxxx	01H	0008H	56H	BufferID	StartPage	PageNum	sum

Note: BufferID of CharBuffer1 and CharBuffer2 are 1h and 2h respectively.

#### Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes	2 bytes	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Page number	Score	Checksum
EF01H	xxxx	07H	0007H	xxH	PageID	MatchScore	sum

Confirmation code =00H, search success;  
 Confirmation code =01H, error when receiving package;  
 Confirmation code =09H, search failed;  
 Confirmation code =22H, residual fingerprint

#### 6.2.9 Upload image: UpImage

Description: to upload the image in ImageBuffer to upper computer.

Input Parameter: none

Return Parameter: Confirmation code (1 byte)

Instruction code: 0aH

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Checksum
EF01H	xxxx	01H	0003H	0AH	000EH

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
EF01H	xxxx	07H	0003H	xxH	sum

Data package format (have subsequent packet) :

2 bytes	4bytes	1 byte	2 bytes	N bytes	2 bytes
Header	Module address	Package identifier	Package length	Contents	Checksum
EF01H	xxxx	02H	N+2	Image data	Sum

End package format (have not subsequent packets) :

2 bytes	4bytes	1 byte	2 bytes	N bytes	2 bytes
Header	Module address	Package identifier	Package length	Contents	Checksum
EF01H	xxxx	08H	N+2	Image data	sum

Note 1: Confirmation code=00H: ready to transfer the following data packet;

Confirmation code=01H: error when receiving package;

Confirmation code=0fH: fail to transfer the following data packet;

2: Module shall transfer the following data packet and end packet after responding to the upper computer. And data packet and end packet no reply packet.

3.The value of N(number of bytes of the packet content) is determined by the length of the packet content, factory package content length is set to 128 bytes.

#### 6.2.10 Download the image: DownImage

Description: to download image from upper computer to ImageBuffer. The image must be 256\*288 size in BMP format.

Input Parameter: none

Return Parameter: Confirmation code (1 byte)

Instruction code: 0bH

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Checksum
EF01H	xxxx	01H	0003H	0bH	000FH

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
EF01H	xxxx	07H	0003H	xxH	sum

Data package format (have subsequent packet) :

2 bytes	4bytes	1 byte	2 bytes	N bytes	2 bytes
Header	Module address	Package identifier	Package length	Contents	Checksum
EF01H	xxxx	02H	N+2	Image data	Sum

End package format (have not subsequent packets) :

2 bytes	4bytes	1 byte	2 bytes	N bytes	2 bytes
Header	Module address	Package identifier	Package length	Contents	Checksum
EF01H	xxxx	08H	N+2	Image data	sum

Note: 1: Confirmation code=00H: ready to transfer the following data packet;

Confirmation code=01H: error when receiving package;

Confirmation code=0eH: fail to transfer the following data packet;

2: Module shall transfer the following data packet and end packet after responding to the upper computer.

3: The value of N (number of bytes of the packet content) is determined by the length of the packet content, factory package content length is set to 128 bytes.

## 6.2.11 To generate character file from image: Img2Tz

Description: to generate character file from the original finger image in ImageBuffer and store the file in CharBuffer1 or CharBuffer2.

Input Parameter: BufferID (character file buffer number)

Return Parameter: Confirmation code (1 byte)

Instruction code: 02H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Buffer number	Checksum
EF01H	xxxx	01H	0004H	02H	BufferID	sum

Note: BufferID of CharBuffer1 and CharBuffer2 are 1h and 2h respectively. Other values (except 1h, 2h) would be processed as CharBuffer2.

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
EF01H	xxxx	07H	0003H	xxH	sum

Note: Confirmation code=00H: generate character file complete;  
Confirmation code=01H: error when receiving package;  
Confirmation code=06H: fail to generate character file due to the over-disorderly fingerprint image;  
Confirmation code=07H: fail to generate character file due to lackness of character point or over-smallness of fingerprint image;  
Confirmation code=15H: fail to generate the image for the lackness of valid primary image;

### 6.2.12 To generate template: RegModel

Description: To combine information of character files from CharBuffer1 and CharBuffer2 and generate a template which is stored back in both CharBuffer1 and CharBuffer2.

Input Parameter: none

Return Parameter: Confirmation code (1 byte)

Instruction code: 05H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Checksum
EF01H	xxxx	01H	0003H	05H	09H

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
EF01H	xxxx	07H	0003H	xxH	sum

Note: Confirmation code=00H: operation success; Confirmation code=01H: error when receiving package;  
Confirmation code=0aH: fail to combine the character files. That's, the character files don't belong to one finger.

### 6.2.13 To upload character or template: UpChar

Description: to upload the character file or template of CharBuffer1/CharBuffer2 to upper computer;

Input Parameter: BufferID (Buffer number)

Return Parameter: Confirmation code (1 byte)

Instruction code: 08H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Buffer number	Checksum
EF01H	xxxx	01H	0004H	08H	BufferID	sum

Note: BufferID of CharBuffer1 and CharBuffer2 are 1h and 2h respectively.

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
EF01H	xxxx	07H	0003H	xxH	sum

Data package format (have subsequent packet) :

2 bytes	4bytes	1 byte	2 bytes	N bytes	2 bytes
Header	Module address	Package identifier	Package length	Contents	Checksum
EF01H	xxxx	02H	N+2	Template data	Sum

End package format (have not subsequent packets) :

2 bytes	4bytes	1 byte	2 bytes	N bytes	2 bytes
Header	Module address	Package identifier	Package length	Contents	Checksum
EF01H	xxxx	08H	N+2	Template data	sum

Note 1: Confirmation code=00H: ready to transfer the following data packet;

Confirmation code=01H: error when receiving package; Confirmation code=0dH: error when uploading template;

2: Module shall transfer following data packet after responding to the upper computer.;

3.The value of N(number of bytes of the packet content) is determined by the length of the packet content, factory package content length is set to 128 bytes.

4: The instruction doesn't affect buffer contents.

## 6.2.14 To download character file or template: DownChar

Description: to download character file or template from upper computer to the specified buffer of Module;

Input Parameter: BufferID (buffer number)

Return Parameter: Confirmation code (1 byte)

Instruction code: 09H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	buffer number	Checksum
EF01H	xxxx	01H	0004H	09H	BufferID	sum

Note: BufferID of CharBuffer1 and CharBuffer2 are 1h and 2h respectively.

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
EF01H	xxxx	07H	0003H	xxH	sum

Data package format (have subsequent packet) :

2 bytes	4bytes	1 byte	2 bytes	N bytes	2 bytes
Header	Module address	Package identifier	Package length	Contents	Checksum
EF01H	xxxx	02H	N+2	Template data	Sum

End package format (have not subsequent packets) :

2 bytes	4bytes	1 byte	2 bytes	N bytes	2 bytes
Header	Module address	Package identifier	Package length	Contents	Checksum
EF01H	xxxx	08H	N+2	Template data	sum

Note 1: Confirmation code=00H: ready to transfer the following data packet;

Confirmation code=01H: error when receiving package;

Confirmation code=0eH: fail to receive the following data packages.

2: Module shall transfer the following data packet after responding to the upper computer.

3. The value of N(number of bytes of the packet content) is determined by the length of the packet content, factory package content length is set to 128 bytes.

### 6.2.15 To store template: Store

Description: to store the template of specified buffer (Buffer1/Buffer2) at the designated location of Flash library.

Input Parameter: BufferID(buffer number)+PageID (Flash location of the template, two bytes with high byte front and low byte behind)

Return Parameter: Confirmation code (1 byte)

Instruction code: 06H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	buffer number	Location number	Checksum
EF01H	xxxx	01H	06H	0006H	BufferID	PageID	sum

Note: BufferID of CharBuffer1 and CharBuffer2 are 1h and 2h respectively.

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
EF01H	xxxx	07H	0003H	xxH	sum

Note: Confirmation code=00H: storage success;

Confirmation code=01H: error when receiving package;

Confirmation code=0bH: addressing PageID is beyond the finger library;

Confirmation code=18H: error when writing Flash.

### 6.2.16 To read template from Flash library: LoadChar

Description: to load template at the specified location (PageID) of Flash library to template buffer CharBuffer1/CharBuffer2

Input Parameter: BufferID(buffer number)+PageID (Flash location of the template, two bytes with high byte front and low byte behind).

Return Parameter: Confirmation code (1 byte) Instruction code: 07H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	buffer number	Page number	Checksum
EF01H	xxxx	01H	0006H	07H	BufferID	PageID	sum

Note: BufferID of CharBuffer1 and CharBuffer2 are 1h and 2h respectively.

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
EF01H	xxxx	07H	0003H	xxH	sum

Note: Confirmation code=00H: load success;

Confirmation code=01H: error when receiving package;

Confirmation code=0cH: error when reading template from library or the readout template is invalid;

Confirmation code=0BH: addressing PageID is beyond the finger library;

### 6.2.17 To delete template: DeletChar

Description: to delete a segment (N) of templates of Flash library started from the specified location (or PageID);

Input Parameter: PageID (template number in Flash)+ N (number of templates to be deleted)

Return Parameter: Confirmation code (1 byte)

Instruction code: 0cH

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes	2bytes	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Page number	number of templates to be deleted	Checksum
EF01H	xxxx	01H	0007H	0cH	PageID	N	sum

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
EF01H	xxxx	07H	0003H	xxH	sum

Note: Confirmation code=00H: delete success;

Confirmation code=01H: error when receiving package;

Confirmation code=10H: failed to delete templates;

### 6.2.18 To empty finger library: Empty

Description: to delete all the templates in the Flash library

Input Parameter: none

Return Parameter: Confirmation code (1 byte)

Instruction code: 0dH

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Checksum
EF01H	xxxx	01H	0003H	0dH	0011H

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum

EF01H	xxxx	07H	0003H	xxH	sum
-------	------	-----	-------	-----	-----

Note: Confirmation code=00H: empty success;  
Confirmation code=01H: error when receiving package;  
Confirmation code=11H: fail to clear finger library;

### 6.2.19 To carry out precise matching of two finger templates: Match

Description: to carry out precise matching of templates from CharBuffer1 and CharBuffer2, providing matching results.

Input Parameter: none

Return Parameter: Confirmation code (1 byte) + matching score.

Instruction code: 03H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Checksum
EF01H	xxxx	01H	0003H	03H	0007H

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Matching score	Checksum
EF01H	xxxx	07H	0005H	xxH	xxH	sum

Note 1: Confirmation code=00H: templates of the two buffers are matching;  
Confirmation code=01H: error when receiving package;  
Confirmation code=08H: templates of the two buffers aren't matching;  
2: The instruction doesn't affect the contents of the buffers.

### 6.2.20 To search finger library: Search

Description: to search the whole or part finger library for the template that matches the one in CharBuffer1 or CharBuffer2. When found, PageID will be returned.

Input Parameter: BufferID+StartPage (searching start address)+ PageNum (searching numbers)

Return Parameter: Confirmation code (1 byte)+PageID (matching templates location)

Instruction code: 04H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	buffer number	Start page number	Number	Checksum
EF01H	xxxx	01H	0008H	04H	BufferID	StartPage	PageNum	sum

Note: BufferID of CharBuffer1 and CharBuffer2 are 1h and 2h respectively.

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes	2 bytes	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Page	Score	Checksum
EF01H	xxxx	07H	0007H	xxH	PageID	MatchScore	sum

Note 1: Confirmation code=00H: found the matching finger;  
Confirmation code=01H: error when receiving package;



Confirmation code=09H: No matching in the library (both the PageID and matching score are 0);

2: The instruction doesn't affect the contents of the buffers.

## 6.3 Other instructions

### 6.3.1 To write note pad: WriteNotepad

Description: for upper computer to write data to the specified Flash page;

Input Parameter: NotePageNum, user content (or data content)

Return Parameter: Confirmation code (1byte)

Instruction code: 18H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1byte	32 bytes	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Page number	Data content	Checksum
EF01H	xxxx	01H	0024H	18H	0~15	content	sum

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
EF01H	xxxx	07H	0003H	xxH	sum

Note: Confirmation code=00H: write success;

Confirmation code=01H: error when receiving package;

### 6.3.2 To read note pad: ReadNotepad

Description: to read the specified page's data content;

Input Parameter: none

Return Parameter: Confirmation code (1 byte) + data content

Instruction code: 19H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Page number	Checksum
EF01H	xxxx	01H	0004H	19H	0~15	xxH

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	32bytes	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	User content	Checksum
EF01H	xxxx	07H	0023H	xxH	content	sum

Note: Confirmation code=00H: read success;

Confirmation code=01H: error when receiving package;

# Dimensions

Dimensions of integral type Module (unit: mm)

