

ZFM-70 Series
Fingerprint Identification Module
User Manual

Preface & Declaration

Thank you for your selection of ZFM-70 series Fingerprint Identification Module (Module) of Zhiantec Technologies Co., Ltd. (Zhiantec).

The Manual is targeted for hardware & software developing engineer, covering hardware interface, system resource, instruction system, installment information, etc. To ensure the developing process goes smoothly, it is highly recommended the Manual is read through carefully.

We will try our best to assure you the correctness of the Manual. However, should you find any problem or error with it, feel free to contact us or the sales representative of us. We would be very grateful.

Holding the principle of constantly improving and perfecting products, so both the module and contents of the Manual might subject to changes. Sorry for separate notice. You may visit our website or call us for the latest information.

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

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

All products are sold subject to Zhiantec's terms and conditions of sale supplied at the time of order acknowledgment. Testing, tool and other quality control techniques are used to the extent Zhiantec considers necessary to support the warranty of relevant performance of its products to the specifications, except as expressly agreed to in writing by government requirements, testing of all parameters of each product is not necessarily performed.

Copyright © 2010–2012 by Hangzhou Zhian Technologies Co., Ltd. All rights reserved.

Contact Information

<http://www.zhiantec.com>

Address: Suite C&D, Floor 8, West Building of International Garden,

No.160 Tianmushian RD., Hangzhou 310013, P.R.China

Phone: +86 571 88210122/88210133

Fax: +86 571 88210122(818)

Revisions

Version	Date	Modify	Modify content
V1.0	20131213		Base version

Content

Revisions	4
Content	5
I Introduction	7
1.1 characteristics	7
1.2 New features	7
1.3 Operation Principle	8
1.4 Order Information	8
II Main Parameters	10
III Hardware Interface	11
3.1 Serial Communication	11
3.1.1 Hardware connection	11
3.1.2 Serial communication protocol	12
3.1.3 Reset time	12
3.1.4 Electrical parameter (All electrical level takes GND as reference)	12
3.2 USB communication	13
IV System Resources	15
4.1 Buffer	15
4.1.1 Image buffer	15
4.1.2 Character file buffer	15
4.2 Fingerprint Library	15
4.3 System Configuration Parameter	15
4.3.1 Baud rate control (Parameter Number: 4)	15
4.3.2 Security Level (Parameter Number: 5)	15
4.3.3 Data package length (Parameter Number: 6)	16
4.4 System status register	16
4.5 Module password	16
4.6 Module address	16
4.7 Notepad	16
V Communication Protocol	18
5.1 Data package format	18
Instruction set summary	19
3.1.5 Classified by functions	19
3.1.6 Classified by instruction order	20
5.2 Check and acknowledgement of data package	20
5.3 Brief Work flow	21
VI Module Instruction System	23
6.1 System-related instructions	23
6.1.1 Verify password	23
6.1.2 Set password	23
6.1.3 Set Module address	24
6.1.4 Set system parameter	24

6.1.5	Read system Parameter	25
6.1.6	Read template index table	25
6.1.7	Read valid template number	26
6.2	Fingerprint-processing instructions	27
6.2.1	To capture finger image	27
6.2.2	Open backlight	27
6.2.3	Close backlight	28
6.2.4	Capture fingerprint no control light	28
6.2.5	Handshake	29
6.2.6	Auto Enroll fingerprint	29
6.2.7	Auto search	31
6.2.8	Search (with residual judge)	31
6.1.1	Upload image	32
6.2.9	Download the image	33
6.1.1	To generate character file from image	34
6.2.10	To generate template	34
6.2.11	To upload character or template	35
6.2.12	To download character file or template	36
6.2.13	To store template	36
6.2.14	To read template from Flash library	37
6.2.15	To delete template	38
6.2.16	To empty finger library	38
6.2.17	To carry out precise matching of two finger templates	39
6.2.18	To search finger library	39
6.3	Other instructions	40
6.3.1	Write notepad	40
6.3.2	Read notepad	40
	Appendix	42

I Introduction

1.1 characteristics

ZFM-70 series are separate fingerprint identification modules proposed by Hangzhou Zhian Technologies Co., Ltd., which takes Synochip DSP as the main processor and optical sensor with Zhian's own intellectual property rights. The module performs series of functions like fingerprint enrollment, image processing, fingerprint matching, searching and template storage. Compared to similar fingerprint products, the ZFM-70 module has the following characteristics:

- ◆ Touch sensing intelligent, more convenient operation

Acquisition window finger touch induction, works with low power consumption, more convenient. The ZFM-708 module with the function, the module of ZFM-706 do not have this feature.

- ◆ Independent intellectual property rights, image captured clearly

Hardware module of optical fingerprint sensor, all technologies, from Hangzhou zhian independent development, access to a number of national patents, optical path design is excellent, can gather to the clear fingerprint image.

- ◆ Finger match in range of $\pm 45^\circ$

When Fingerprint matching, support in range of $\pm 45^\circ$.

- ◆ Responsive, strong adaptability of fingerprints

Fingerprint image reading, has sensitive response and judgment of the dry and wet fingers, to obtain the optimal imaging quality, wide application of the crowd. Can also be self learning and adaptive function customization, adjust the parameters automatically according to the change of the habits of the users, such as climate, to better match.

- ◆ A particular highlight light green LED, anti-aging excellent performance

Using a specific green bright light source component, ultra low attenuation, longer service life, better performance and more durable.

- ◆ Easy to develop

Without the need for fingerprint recognition of professional knowledge can be applied, the ZFM-70 module provides rich control instruction, can develop powerful application of fingerprint identification system.

- ◆ Flexible to set security level

For different application occasions or environment, users can set different security level 1 to level 5

1.2 New features

- ◆ Green LED backlight

ZFM-70 series uses green led backlight, more gentle

- ◆ Backlight long bright mode

6.1.1 ZFM-70 series provide the control interface of the LED backlight.

can keep lighting. (Realize by use the command "To capture finger image

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:

Header	Module address	Package identifier	Package length	Instruction code	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	Xxxx	01H	0003H	01H	05H

Acknowledge package format:

Header	Module address	Package identifier	Package 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: finger collection success;

Confirmation code=01H: error when receiving package;

Confirmation code=02H: can’t detect finger;

Confirmation code=03H: fail to collect finger;

”, “Close Led” and “GetImageFree”)

ZFM-70 series also retain the old command interface , use the old command, you can shark the light when capturing fingerprint (Realize by use the command “GetImg”).

◆ The faster response

By optimizing the image acquisition and pretreatment of the driver, the image acquisition and image preprocessing process can save more time, better image quality, faster response speed.

◆ Power on handshake

When ZFM-70 series power on, it will send a byte(0x55) to handshake, then upper computer can send other commands.

◆ single command to complete the enroll and search

ZFM-70 series add the commands “Autologin” and “AutoSearch”, in single command to complete the enroll and search

1.3 Operation Principle

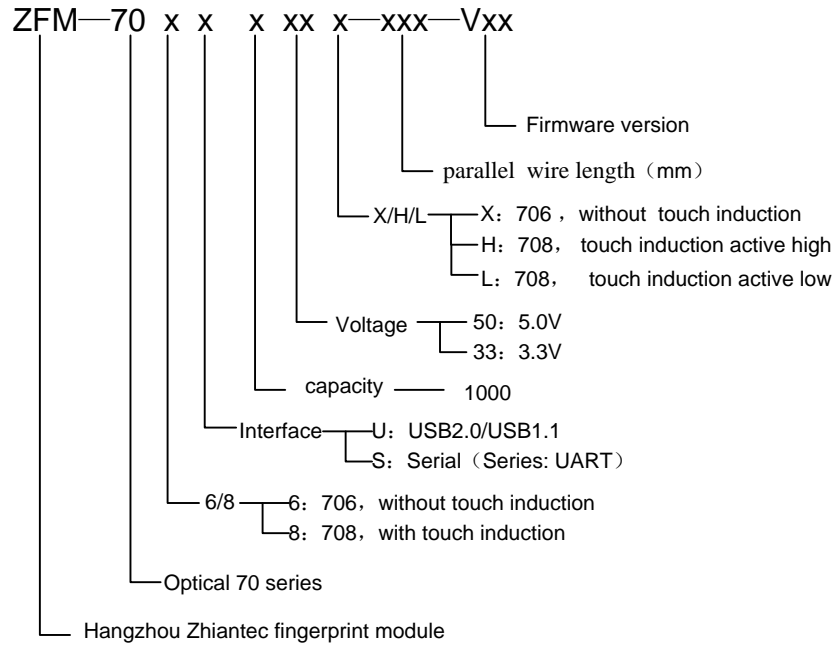
Fingerprint processing includes two parts: fingerprint enrollment and fingerprint matching (the matching can be 1:1 or 1:N).

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.

1.4 Order Information

Naming of our fingerprint modules follows the following rule. When placing order with us, please fill the correct the type name, so that we can provide better service.



Note: 1) Wire length means the length of parallel wire which connects optical sensor and main board. Uart communication mode is 150 mm, usb communication mode is 720 mm.

- 2) Software edition can be omitted in first order or neglected at all. By default, we take it as the latest edition.
- 3) finger library capacity support 1000

II Main Parameters

Power	DC 3.8V-7.0V(direct 3.3V)	Interface	UART(TTL logical level)/ USB 1.1
Back light	Green	Light type	bright/glnt
Working current	Typical: <65mA Peak: <95mA	Matching Mode	1:1 and 1:N
Baud rate	(9600*N)bps, N=1 ~ 12 (default N=6)	Character file size	256 bytes
Image acquiring time	<0.5s	Template size	512 bytes
Storage capacity	150/ 1000	Security level	5 (1, 2, 3, 4, 5(highest))
FAR	<0.001%	FRR	<1%
Average searching time	< 1s (1:500)	Window dimension	14.5mm*19.4mm
Working environment	Temp: -20℃- +60℃	Storage environment	Temp: -40℃- +85℃
	RH: 40%-85%		RH: <85%
Outline Dimention (L*W*H)	Integral type	54*20*20.5mm	

III Hardware Interface

Whether the interface is UART or USB (hardware setting is different when out of factory, please don't misuse), on PCB board the connector is same type, 6-pin connector with 1.25mm space between

3.1 Serial Communication

When the FP module communicates with user device, definition of pin is as follows:

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

Note: ZFM-706 there is not touch induction function, the pin 1 and pin 2 not connect.

ZFM-708 there is touch induction function, all pin is valid.

In type ,in means input to module, out means output from module.

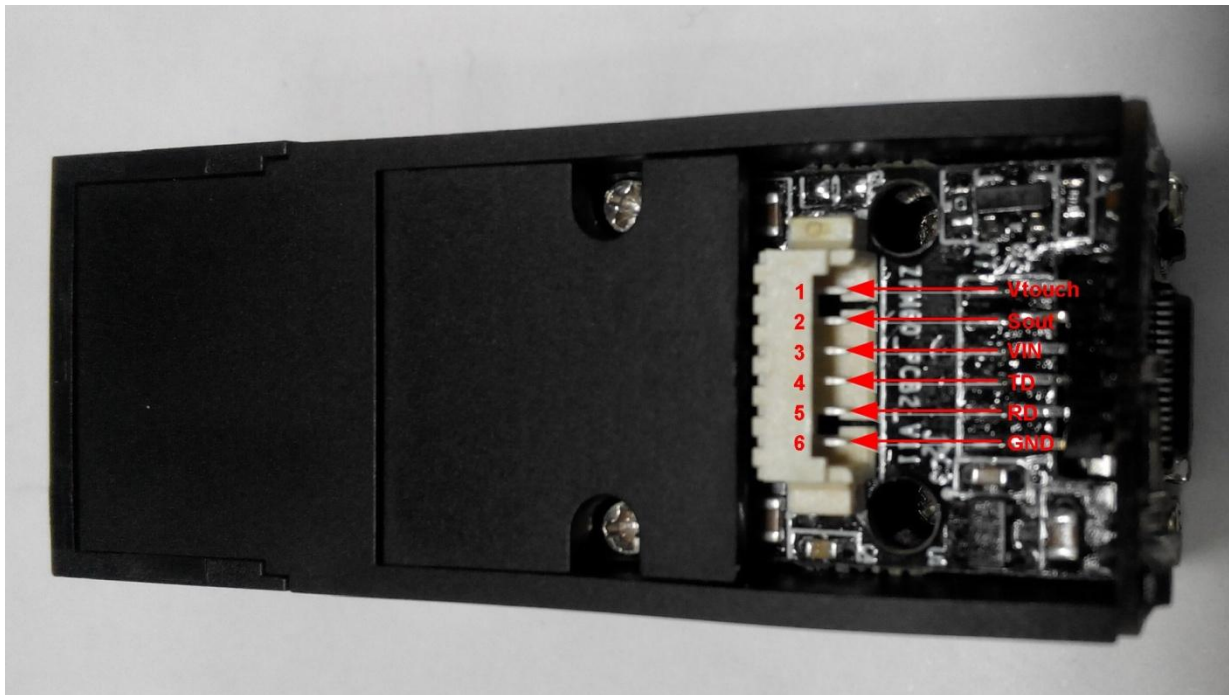


Figure 0.1 Serial communicates interface module

3.1.1 Hardware connection

Via serial interface, the Module may communicate with MCU of 3.3V or 5V power: TD (pin 2 of J1) connects with RXD (receiving pin of MCU), RD (pin 3 of J1) 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.

Considering that the power consumption of the whole module, a fingerprint module only work and not work two kinds of state, no hibernation or standby; normally closed power input of fingerprint module, the fingerprint module does not work; when the MCU (MCU) needs to access a fingerprint module function, provide a fingerprint module power supply input, fingerprint module work, complete the corresponding command, if you no longer use fingerprint module, the fingerprint module power supply is cut off, the fingerprint module into the not working state.

With ZFM-708 module touch induction (ZFM-706 does not have this feature), the default configuration output high level effectively. The finger when no put in the acquisition window, a touch signal output is low; when the finger on the acquisition window, a touch signal output change is high. The signal can be connected with the single chip computer (MCU) certain interrupt feet or IO port, after receive the induction signal, awaken the MCU (MCU), power supply to the fingerprint module, the fingerprint module begins to work.

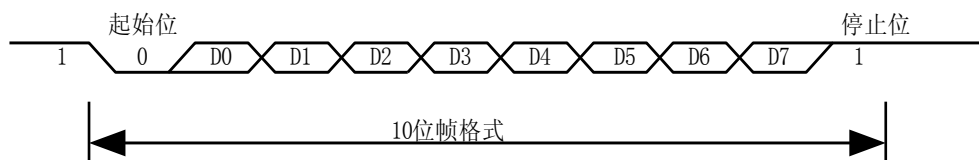
the default power supply voltage of touch induction is 3.3V-5V, can also be directly connected with the 4 battery, the touch induction circuit the overall power consumption current of <10uA; the circuit can continuously supply.

The touch sensing signal output, can be customized to output low level effectively, if you need this function, please inform our company

3.1.2 Serial communication protocol

The mode is semiduplex asynchronism 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.



3.1.3 Reset time

At power on, it takes about 300ms for initialization. During this period, the Module can't accept commands for upper computer. After reset complete, the module will send a byte (0x55).

3.1.4 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 (Vinmax)	-0.3		9.0	V	Exceeding the Maximum rating may cause permant harm to the Module.
Operation Current (Icc)	55	60	95	mA	
Peak Current (Ipeak)			95	mA	

2. TD (output, TTL logic level)

Item	Condition	Parameter			Unit	Note
		Min	Typ	Max		
VOL	IOL= -4mA			0.4	V	Logic 0
VOH	IOH= 4mA	2.4		3.3	V	Logic 1

3.RD (input, TTL logic level)

Item	Condition	Parameter			Unit	Note
		Min	Typ	Max		
VIL				0.6	V	Logic 0
VIH		2.4			V	Logic 1
IIH	VIH =5V		1		mA	
	VIH =3.3V		30		uA	
VImax		-0.3		5.5	V	Maximum input voltage

3.2 USB communication

When it's USB communication, definition is:

list 0.1 USB communication

Pin Num	Name	Type	Function Description
1	-	-	
2	-	-	
3	Vin	in	Power supply input (refer to 3.1.1.4 for electrical parameter)
4	DP+	In/Out	USB data
5	DP-	In/Out	USB data
6	GND	-	Signal ground. Connected to power ground.

Note: In type ,in means input to module, out means output from module.



图 0.2 USB communication interface

When use USB communication interface, all command called by API.

ZFM-70 series not need driver files when use in pc.

If you use in windows/linux/wince platform, you can contact our company for sdk

ZFM-70 series works in USB 2.0 full speed mode.

IV System Resources

To address demands of different customer, Module system provides abundant resources at user's use.

4.1 Buffer

There are an 72K-byte 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.

4.1.1 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.

When transferring through USB, the image is 8-bit pixel, that's 256 grey degrees.

4.1.2 Character file buffer

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

4.2 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. User can only access library by template number.

4.3 System Configuration Parameter

To facilitate user's 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.

4.3.1 Baud rate control (Parameter Number: 4)

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

4.3.2 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 cooresponding value is 1, 2, 3, 4, 5. At level 1, FAR is the highest and FRR is the lowest. For more information, please visit www.zhiantec.com

lowest; however at level 5, FAR is the lowest and FRR is the highest.

4.3.3 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.

4.4 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 device's handshaking password.
- ImgBufStat: 1 bit. 1: image buffer contains valid image.

4.5 Module password

At power-on reset, system first checks whether the handshaking password has been modified. If not, system deems upper computer has no requirement of verifying password and will enter into normal operation mode. That's, when Module password remains the default, verifying process can be jumped. The password length is 4 bytes, and its default factory value is 00H, 00H, 00H, 00H.

Should the password have be modified, *refer to instruction SetPwd*, then Module (or device) handshaking password must be verified before the system enter into normal operation mode. Or else, system will refuse to execute and command.

The new modified password is stored in Flash and remains at power off.

4.6 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.

4.7 Notepad

Module has an area of 512 bytes for notepad in Flash, The notepad is designed for 16 pages, and the size of page is 32 bytes. Via instruction *WriteNotepad* and *ReadNotepad* can write and read the notepad.

Note: When write one page, all 32 bytes will be written, and the old data will be overwrite . . .

V Communication Protocol

The protocol defines the data exchanging format when ZFM-70 series communicates with upper computer. The protocol and instruction sets apply for both UART and USB communication mode. For PC, USB interface is strongly recommended to improve the exchanging speed, especially in fingerprint scanning device.

5.1 Data package format

When communicating, the transferring and receiving of command/data/result are all wrapped in data package format. For multi byte, High byte in the former low byte after. For example, 00 06 means 0006.

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 0xEF01; High byte transferred first.
Adder	ADDER	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.
----------	-----	---------	--

Instruction set summary

3.1.5 Classified by functions

Classification by function, include system function, fingerprint processing and other function.

list 0.1 system instruction

number	instruction	function
1	0x13	Verify password
2	0x12	Set password
3	0x15	Set address
4	0x0e	Set system parameter
5	0x0f	Read system parameter
6	0x1f	Read fingerprint template index table
7	0x1d	Read valid template number

list 0.2 fingerprint processing instruction

number	instruction	function
1	0x01	Capture fingerprint image
2	0x0a	Upload fingerprint image
3	0x0b	Download fingerprint image
4	0x02	Generate Features from fingerprint image
5	0x05	Merge 2 Features to a template
6	0x08	Upload Features
7	0x09	Download Features
8	0x06	Store template
9	0x07	load template to buffer
10	0x0c	Delete template
11	0x0d	Empty template
12	0x03	Match template
13	0x04	Search
14	0x50	Open backlight
15	0x51	Close backlight
16	0x52	Capture fingerprint no control light
17	0x53	handshake
18	0x54	Auto enroll
19	0x55	Auto search
20	0x56	Search finger(with Residual judge)

list 0.3 other instruction

number	instruction	function
1	0x18	Write notepad
2	0x19	Read notepad

3.1.6 Classified by instruction order

list 0.4 by instruction order

number	instruction	function	number	instruction	function
0x01	GenImg	Capture fingerprint image	0x12	SetPwd	Set password
0x02	Img2Tz	Generate Features from fingerprint image	0x13	VfyPwd	Verify password
0x03	Match	Match template	0x14	GetRandomCode	Capture random number
0x04	Serach	Search	0x15	SetAddr	Set address
0x05	RegModel	Merge Features to template	0x18	WriteNotepad	Write notepad
0x06	Store	Store template	0x19	ReadNotepad	Read notepad
0x07	LoadChar	Load template to buffer	0x1d	TemplateNum	Read valid template number
0x08	UpChar	Upload Features	0x1f	ReadConList	Read fingerprint template index table
0x09	DownChr	Download Features	0x50	OpenLED	Open backlight
0x0a	UpImage	Upload fingerprint image	0x51	CloseLED	Close backlight
0x0b	DownImage	Download fingerprint image	0x52	GetImageFree	Capture fingerprint no control light
0x0c	DeletChar	Delete template	0x53	GetEcho	handshake
0x0d	Empty	Empty template	0x54	AutoLogin	Auto enroll
0x0e	SetSysPara	Set system parameter	0x55	AutoSearch	Auto search
0x0f	ReadSysPara	Read system parameter	0x56	SearchResBack	Search finger(with Residual judge)

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.

list 0.5 Confirmation code's definition

NO.	Confirmation code	description
1	0x00	commad execution complete
2	0x01	error when receiving data package
3	0x02	no finger on the sensor
4	0x03	fail to enroll the finger
5	0x06	fail to generate character file due to the over-disorderly fingerprint image
6	0x07	fail to generate character file due to lackness of character point or over-smallness of fingerprint image
7	0x08	finger doesn't match;
8	0x09	fail to find the matching finger
9	0x0a	fail to combine the character files;
10	0x0b	addressing PageID is beyond the finger library;
11	0x0c	error when reading template from library or the template is invalid;
12	0x0d	error when uploading template;
13	0x0e	Module can't receive the following data packages.
14	0x0f	error when uploading image
15	0x10	fail to delete the template;
16	0x11	fail to clear finger library
17	0x13	wrong password!
18	0x15	fail to generate the image for the lackness of valid primary image
19	0x18	error when writing flash;
20	0x1a	invalid register number;
21	0x20	Address error
22	0x21	Must verify password
23	Others	system reserved;

5.3 Brief Work flow

After completing initialization, waiting for instruction from upper computer, when received the instruction, then execute the instruction, and return the result.

You can enroll and search to complete finger match. Also you can uploading and downloading feature to complete finger match.

Instruction Decomposition

For example: Capture fingerprint image

Ef 01 ff ff ff ff 01 00 03 01 00 05 (01 is package Identification, it means command package)

Note : 01 is instruction , it means to execute capture fingerprint image.

After received the command, the module will execute the command ,and return the result of command.

Ef 01 ff ff ff ff 07 00 03 02 00 0c (07 is package Identification, it means response

package)

Note: 02 Confirmation code, means no finger on sensor

If the Confirmation code is 00, it means ok.

Ef 01 ff ff ff ff 07 00 03 00 00 0a

Then execute generate features instruction

Ef 01 ff ff ff ff 01 00 04 02 01 00 08(02 is instruction, it means to execute generate features instruction.)

Note: 01 is parameter, it means the features store to buffer 1

VI Module Instruction System

ZFM-70 series provide many instructions. Through combination of different instructions, application program may realize multi finger authentication functions. All commands/data are transferred in package format. Refer to 5.1 for the detailed information of package

6.2 System-related instructions

6.2.1 Verify password

Description: Verify Module's handshaking password. (Refer to 4.6 for details)

Input Parameter: PassWord (4 bytes)

Return Parameter: Confirmation code (1 byte)

Instruction code: 13H

Command (or instruction) package format:

Header	Module address	Package identifier	Package Length	Instruction code	Password	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	4 byte	2 bytes
0xEF01	xxxx	01H	0007H	13H	PassWord	sum

Acknowledge package format:

Header	Module address	Package identifier	Package 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: Correct password;

Confirmation code = 01H: error when receiving package;

Confirmation code = 13H: Wrong password;

6.2.2 Set password

Description: Set Module's handshaking password. (Refer to 4.6 for details)

Input Parameter: PassWord (4 bytes)

Return Parameter: Confirmation code (1 byte)

Instruction code: 12H

Command (or instruction) package format:

Header	Module address	Package identifier	Package length	Instruction code	Password	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	4 byte	2 bytes
0xEF01	xxxx	01H	0007H	12H	PassWord	sum

Acknowledge package format:

Header	Module address	Package identifier	Package 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: password setting complete;

Confirmation code=01H: error when receiving package;

6.2.3 Set Module address

Description: Set Module address. (Refer to 4.7 for address information)

Input Parameter: None;

Return Parameter: Confirmation code (1 byte)

Instruction code: 15H

Command (or instruction) package format:

Header	Original Module address	Package identifier	Package length	Instruction code	New Module address	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	4 bytes	2 bytes
0xEF01	xxxx	01H	0007H	15H	xxxx	sum

Acknowledge package format:

Header	New Module address	Package identifier	Package 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: address setting complete;

Confirmation code=01H: error when receiving package;

6.2.4 Set system parameter

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

Input Parameter: Parameter number;

Return Parameter: Confirmation code (1 byte)

Instruction code: 0eH

Command (or instruction) package format:

Header	Module address	Package identifier	Package length	Instruction code	Parameter number	Contents	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	1byte	1byte	2 bytes
0xEF01	Xxxx	01H	0005H	0eH	4/5/6	xx	sum

Acknowledge package format:

Header	Module address	Package identifier	Package 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: parameter setting complete;

Confirmation code=01H: error when receiving package;

Confirmation code=1aH: wrong register number;

6.2.5 Read system Parameter

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:

Header	Module address	Package identifier	Package length	Instruction code	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	Xxxx	01H	0003H	0fH	sum

Acknowledge package format:

Header	Module address	Package identifier	Package length	Confirmation code	Basic parameter list	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	16 bytes	2 bytes
0xEF01	xxxx	07H	0013H	xxH	See following table	sum

Note: Confirmation code=00H: read complete;

Confirmation code=01H: error when receiving package;

Name	Description	Offset (word)	Size (word)
Status register	Contents of system status register	0	1
System identifier code	Fixed value: 0x0009	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

6.2.6 Read template index table

Description: read the template NO. list and only support 256 list at most each time.

Input Parameter: list page

Page 0: read the 0-255 finger template list;

Page 1: read the 256-511 finger template list;
 Page 2: read the 512-767 finger template list;
 Page 3: read the 768-1023 finger template list;

Return parameter: confirmation code(1 byte), template lists;

Instruction code: 1fH

Command (or instruction) package format:

Header	Module address	Package identifier	Package length	Instruction code	List page	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes
0xEF01	Xxxx	01H	0004H	0fH	0/1/2/3	sum

Acknowledge package format:

Header	Module address	Package identifier	Package length	Confirmation code	Con List	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	32 bytes	2 bytes
0xEF01	xxxx	07H	0023H	xxH	See following table	sum

Note: 1、confirmation code = 00h: read template list success;

Confirmation code = 01h: package receiving wrong;

2、Each time it can read 256 list data; if the data bit less than 256 it should defined as 0;

3、list data structure: 1 group maintains 8 bit and can be output by the high bit firstly.

For details please see below table:

Transfer sequence	Output from low byte to hig byte in order and every byte should be output by the highest bit								
The lowest byte	Template NO.	7	6	5	4	3	2	1	0
	List data	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
The second lowest byte	Template NO.	15	14	13	12	11	10	9	8
	List data	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
.....							
The highest byte	Template NO.	255	254	253	252	251	250	249	248
	List data	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

Note: list data 0 means there is no template in this place while 1 means exists.

6.2.7 Read valid template number

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:

Header	Module address	Package identifier	Package length	Instruction code	Checksum
--------	----------------	--------------------	----------------	------------------	----------

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	xxxx	01H	0003H	1dH	0021H

Acknowledge package format:

Header	Module address	Package identifier	Package length	Confirmation code	Template number	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes	2 bytes
0xEF01	xxxx	07H	0005h	xxH	N	sum

Note: Confirmation code=00H: read complete;

Confirmation code=01H: error when receiving package;

6.3 Fingerprint-processing instructions

6.3.1 To capture finger image

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:

Header	Module address	Package identifier	Package length	Instruction code	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	Xxxx	01H	0003H	01H	05H

Acknowledge package format:

Header	Module address	Package identifier	Package 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: finger collection success;

Confirmation code=01H: error when receiving package;

Confirmation code=02H: can’t detect finger;

Confirmation code=03H: fail to collect finger;

6.3.2 Open backlight

Description: open backlight for capture fingerprint

Input Parameter: none

Return Parameter: Confirmation code (1 byte)

Instruction code:50H

Command (or instruction) package format:

Header	Module address	Package identifier	Package length	Instruction code	Checksum
2 bytes	4 bytes	1 bytes	2 bytes	1 bytes	2 bytes
0xEF01	XXXX	0x01	0x0003	0x50	0x54

Acknowledge package format:

Header	Module address	Package identifier	Package length	Confirmation code	Checksum
2 bytes	4 bytes	1 bytes	2 bytes	1 bytes	2 bytes
0xEF01	XXXX	0x07	0x0003	XX	XX

Note: Confirmation code=00H: success

Others: fail

6.3.3 Close backlight

Description: Close backlight

Input Parameter: none

Return Parameter: Confirmation code (1 byte)

Instruction code: 51H

Command (or instruction) package format:

Header	Module address	Package identifier	Package length	Instruction code	Checksum
2 bytes	4 bytes	1 bytes	2 bytes	1 bytes	2 bytes
0xEF01	XXXX	0x01	0x0003	0x51	0x55

Acknowledge package format:

Header	Module address	Package identifier	Package length	Confirmation code	Checksum
2 bytes	4 bytes	1 bytes	2 bytes	1 bytes	2 bytes
0xEF01	XXXX	0x07	0x0003	XX	XX

Note: Confirmation code=00H: success

Others: fail

6.3.4 Capture fingerprint no control light

Description: Auto capture fingerprint, not need to control light

Input Parameter: none

Return Parameter: Confirmation code (1 byte)

Instruction code: 52H

Command (or instruction) package format:

Header	Module address	Package identifier	Package length	Instruction code	Checksum
--------	----------------	--------------------	----------------	------------------	----------

2 bytes	4 bytes	1 bytes	2 bytes	1 bytes	2 bytes
0xEF01	XXXX	0x01	0x0003	0x52	0x56

Acknowledge package format:

Header	Module address	Package identifier	Package length	Confirmation code	Checksum
2 bytes	4 bytes	1 bytes	2 bytes	1 bytes	2 bytes
0xEF01	XXXX	0x07	0x0003	XX	XX

Note: Confirmation code=00H: success

Confirmation code=01H, fail to receive package

Confirmation code=02H, no finger on sensor

Confirmation code=03H, capture fail

6.3.5 Handshake

Description: send handshake instruction to module, if module is ok, it will send 0x55 back. If no data return, the module is work wrong.

Input Parameter: none

Return Parameter: Confirmation code (1 byte)

Instruction code: 53H

Command (or instruction) package format:

Header	Module address	Package identifier	Package length	Instruction code	Checksum
2 bytes	4 bytes	1 bytes	2 bytes	1 bytes	2 bytes
0xEF01	XXXX	0x01	0x0003	0x53	0x57

Acknowledge package format:

Header	Module address	Package identifier	Package length	Confirmation code	Checksum
2 bytes	4 bytes	1 bytes	2 bytes	1 bytes	2 bytes
0xEF01	XXXX	0x07	0x0003	XX	XX

Note:

Confirmation code=55H, it means the module is ok, it can receive instruction.

Confirmation code= other or none, the module work wrong.

By the way, when the module power on, it will send 0x55 to upper computer. The upper computer can send other command to module.

6.3.6 Auto Enroll fingerprint

Description: Send the instruction, the module will complete a serial of operation. include: capture fingerprint, generate features, merge two features to one template and store the template to flash.

Input Parameter: wait time for press finger + times for press finger + store number in flash

Return Parameter: Confirmation code (1 byte)

Instruction code: 54H

Command (or instruction) package format:

Header	Module address	Package identifier	Package length	Instruction code	wait time for press	times for press	store number	Repeat flag	Checksum
2 bytes	4 bytes	1 bytes	2 bytes	1 bytes	1 bytes	1 bytes	2 bytes	1byte	2 bytes
0xEF01	XXXX	0x01	0x0008	0x54	XX	2/3	xxxx	0/1	xxxx

1. Wait time for press finger is the max wait time for waiting . if no finger press inside the wait time, it will return no finger. The value range from 1 to 255, The higher the value, the more time, for ZFM-70 serials Usually use 54 (0x36) The corresponding time is 3.5seconds, the other time is below:

Value	Wait time(seconds)	Value	Wait time(seconds)
31(0x1f)	2	62(0x3e)	4
38(0x26)	2.5	69(0x45)	4.5
46(0x2e)	3	77(0x4d)	5
54(0x36)	3.5	85(0x55)	5.5

2. Times for press means the times to press finger to enroll one template, the value is 2 or 3, 2 means press two times, 3 means press three times.

3. When the press time is 2, the command will capture two fingerprint to merge a template, and the first capture, the module will send Confirmation code 0x56(PS_AUTOLOGIN_OK1), then you can continue to capture fingerprint second

When the press time is 3 ,the command will capture three fingerprint to merge a template, and the first capture, the module will send Confirmation code 0x56(PS_AUTOLOGIN_OK1),the second capture, the module will send Confirmation code 0x57(PS_AUTOLOGIN_OK2). then you can continue to capture fingerprint third.

4. Repeat flag means whether still enroll if the template exists. The value 0 means cannot repeat, if the template exists, it will not enroll. The value 1 means can repeat, even if the template exists, it can enroll also.

Acknowledge package format:

2 bytes	4 bytes	1 bytes	2 bytes	1 bytes	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
0xEF01	XXXX	0x07	0x0003	XX	XX

Note: Confirmation code=00H: success

Confirmation code=01H, fail to receive package

Confirmation code=02H, no finger on sensor

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;

Confirmation code =0bH, addressing Page ID is beyond the finger library;

Confirmation code =56H, The first acquisition was successful

Confirmation code =57H, The second acquisition was successful

Confirmation code =24H, repeat to enroll(the template is exist)

6.3.7 Auto search

Description: Send the instruction, the module will complete a serial of operation. include: capture fingerprint, generate features, and search

Input Parameter: wait time for press+ start ID+ search numbers

Return Parameter: Confirmation code (1 byte)

Instruction code: 55H

Command (or instruction) package format:

Header	Module address	Package identifier	Package length	Instruction code	wait time for press	Start id	Search numbers	Checksum
2 bytes	4 bytes	1 bytes	2 bytes	1 bytes	1 bytes	2 bytes	2 bytes	2 bytes
0xEF01	XXXX	0x01	0x0008	0x55	XX	xxxx	xxxx	xxxx

Wait time for press finger is the max wait time for waiting . if no finger press inside the wait time, it will return no finger. The value range from 1 to 255, The higher the value, the more time, for 70serials Usually use 54 (0x36) The corresponding time is 3.5seconds, the other time is below:

Value	Wait time(seconds)	Value	Wait time(seconds)
31(0x1f)	2	62(0x3e)	4
38(0x26)	2.5	69(0x45)	4.5
46(0x2e)	3	77(0x4d)	5
54(0x36)	3.5	85(0x55)	5.5

Acknowledge package format:

Header	Module address	Package identifier	Package length	Confirmation code	Find page	score	Checksum
2 bytes	4 bytes	1 bytes	2 bytes	1 bytes	2 bytes	2 bytes	2 bytes
0xEF01	XXXX	0x07	0x0007	XX	xxxx	xxxx	XX

Note: Confirmation code =00H, success to search

Confirmation code =09H, No search

Confirmation code =02H, no finger on sensor

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 = 0x22 , residual finger

Confirmation code = 0x23 , there is not valid template

6.3.8 Search (with residual judge)

Description: use the feature of CharBuffer1 or CharBuffer2 to search. If searched, then return the id matched.

The difference between the command and Search(instruction code is 04H) is that the return code is different.

SearchResBack return 22H when find residual ,but Search return 09H.

Input Parameter: Buffer ID + Start Page +Page number

Return Parameter: Confirmation code (1 byte)+find number

Instruction code:56H

Command (or instruction) package format:

2 bytes	4 bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Buffer id	Start page	Page number	Checksum
0xef01	XXXX	0x01	0x0008	0x56	BufferID	StartPage	PageNum	Sum

Note: ID of Buffer CharBuffer1、CharBuffer2 is 01H and 02H

Acknowledge package format:

2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes	2 bytes	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Find page	score	Checksum
0xef01	XXXX	0x07	0x007	X	PageID	MatchScore	Sum

Note: Confirmation code =00H,success

Confirmation code =01H,fail to receive package

Confirmation code =09H, No search

Confirmation code =22H , residual finger

6.3.9 Upload image

Description: to upload the image in Img_Buffer to upper computer. Refer to 1.1.1 for more about image buffer.

Input Parameter: none

Return Parameter: Confirmation code (1 byte)

Instruction code: 0aH

Command (or instruction) package format:

Header	Module address	Package identifier	Package length	Instruction code	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	Xxxx	01H	0003H	0aH	000eH

Acknowledge package format:

Header	Module address	Package identifier	Package length	Confirmation code	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	Xxxx	07H	0003H	xxH	sum

Data Package format(data package stills will be transferred later):

Header	Module address	Package identifier	Package length	Package content	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	Xxxx	02H	N+2	N	sum

End package format:

Header	Module address	Package identifier	Package length	Package content	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	Xxxx	08H	N + 2	N	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 after responding to the upper computer.

6.3.10 Download the image

Description: to download image from upper computer to Img_Buffer. Refer to 1.1.1 for more about the image buffer.

Input Parameter: none

Return Parameter: Confirmation code (1 byte)

Instruction code: 0bH

Command (or instruction) package format:

Header	Module address	Package identifier	Package length	Instruction code	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	Xxxx	01H	03H	0bH	000fH

Acknowledge package format:

Header	Module address	Package identifier	Package length	Confirmation code	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	Xxxx	07H	03H	xxH	sum

Data Package format(data package stills will be transferred later):

Header	Module address	Package identifier	Package length	Package content	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	Xxxx	02H	N+2	N	sum

End package format:

Header	Module address	Package identifier	Package length	Package content	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	Xxxx	08H	N + 2	N	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 after responding to the upper computer. Data package length must be 64, 128, or 256.

6.3.11 To generate character file from image

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:

Header	Module address	Package identifier	Package length	Instruction code	Buffer number	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes
0xEF01	xxxx	01H	04H	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:

Header	Module address	Package identifier	Package length	Confirmation code	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	xxxx	07H	03H	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;

6.3.12 To generate template

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

Input Parameter: none

Return Parameter: Confirmation code (1 byte)

Instruction code: 05H

Command (or instruction) package format:

Header	Module address	Package identifier	Package length	Instruction code	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	xxxx	01H	03H	05H	09H

➤ Acknowledge package format:

Header	Module address	Package identifier	Package length	Confirmation code	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	xxxx	07H	03H	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.

6.3.13 To upload character or template

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:

Header	Module address	Package identifier	Package length	Instruction code	Buffer number	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes
0xEF01	xxxx	01H	04H	08H	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:

Header	Module address	Package identifier	Package length	Confirmation code	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	xxxx	07H	03H	xxH	sum

Data Package format(data package stills will be transferred later):

Header	Module address	Package identifier	Package length	Package content	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	Xxxx	02H	N+2	N	sum

End package format:

Header	Module address	Package identifier	Package length	Package content	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	Xxxx	08H	N + 2	N	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 instruction doesn't affect buffer contents.

6.3.14 To download character file or template

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:

Header	Module address	Package identifier	Package length	Instruction code	buffer number	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes
0xEF01	xxxx	01H	04H	09H	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:

Header	Module address	Package identifier	Package length	Confirmation code	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	xxxx	07H	03H	xxH	sum

Data Package format(data package stills will be transferred later):

Header	Module address	Package identifier	Package length	Package content	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	Xxxx	02H	N+2	N	sum

End package format:

Header	Module address	Package identifier	Package length	Package content	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	Xxxx	08H	N + 2	N	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.

6.3.15 To store template

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:

Header	Module address	Package identifier	Package length	Instruction code	buffer number	Location number	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes
0xEF01	xxxx	01H	06H	06H	BufferID	PageID	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:

Header	Module address	Package identifier	Package length	Confirmation code	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	Xxxx	07H	03H	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.

6.3.16 To read template from Flash library

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:

Header	Module address	Package identifier	Package length	Instruction code	buffer number	Page number	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes
0xEF01	xxxx	01H	06H	07H	BufferID	PageID	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:

Header	Module address	Package identifier	Package length	Confirmation code	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	xxxx	07H	03H	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;

6.3.17 To delete template

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:

Header	Module address	Package identifier	Package length	Instruction code	Page number	number of templates to be deleted	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes	2bytes	2 bytes
0xEF01	Xxxx	01H	07H	0cH	PageID	N	sum

Acknowledge package format:

Header	Module address	Package identifier	Package length	Confirmation code	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	Xxxx	07H	03H	xxH	sum

Note: Confirmation code=00H: delete success;

Confirmation code=01H: error when receiving package;

Confirmation code=10H: fail to delete templates;

6.3.18 To empty finger library

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:

Header	Module address	Package identifier	Package length	Instruction code	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	Xxxx	01H	03H	0dH	0011H

Acknowledge package format:

Header	Module address	Package identifier	Package length	Confirmation code	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	Xxxx	07H	03H	xxH	sum

Note: Confirmation code=00H: empty success;

Confirmation code=01H: error when receiving package;

Confirmation code=11H: fail to clear finger library;

6.3.19 To carry out precise matching of two finger templates

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:

Header	Module address	Package identifier	Package length	Instruction code	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
0xEF01	Xxxx	01H	03H	03H	07H

Acknowledge package format:

Header	Module address	Package identifier	Package length	Confirmation code	Matching score	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes	2 bytes
0xEF01	Xxxx	07H	05H	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.

6.3.20 To search finger library

Description: to search the whole 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:

Header	Module address	Package identifier	Package length	Instruction code	buffer number	Parameter	Parameter	Checksum
2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes	2 bytes
0xEF01	xxxx	01H	08H	04H	BufferID	StartPage	PageNum	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:

Header	Module	Package	Package	Confirmation	page	score	Checksum
--------	--------	---------	---------	--------------	------	-------	----------

	address	identifier	length	code			
2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes	2 bytes	2 bytes
0xEF01	xxxx	07H	7	xxH	PageID	MatchScore	sum

Note 1: Confirmation code=00H: found the matching finer;

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.

6.4 Other instructions

6.4.1 Write notepad

Description: write 32 bytes user content to notepad page

Input Parameter: Page Number, user content

Return Parameter: Confirmation code (1 byte)

Instruction code:18H

Command (or instruction) package format:

Header	Module address	Package identifier	Package length	Instruction code	Page	User Info	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	1 byte	32 bytes	2 bytes
0xef01	XXXX	0x01	0x0024	0x18	0x00-0x0e	32 bytes	Sum

Acknowledge package format:

Header	Module address	Package identifier	Package length	Confirmation code	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	2 bytes
0xef01	XXXX	0x07	0x0003	X	Sum

Note: Confirmation code=00H: success

Others: fail

6.4.2 Read notepad

Description: read user content from notepad page

Input Parameter: page

Return Parameter: Confirmation code (1 byte)+32 bytes user content

Instruction code:19H

Command (or instruction) package format:

Header	Module	Package	Package	Instruction	Page	Checksum
--------	--------	---------	---------	-------------	------	----------

	address	identifier	length	n code		
2 bytes	4 bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes
0xef01	XXXX	0x01	0x0004	0x19	0x00-0x0e	sum

Acknowledge package format:

Header	Module address	Package identifier	Package length	Confirmation code	User content	Checksum
2 bytes	4 bytes	1 byte	2 bytes	1 byte	32 bytes	2 bytes
0xef01	XXXX	0x07	0x0023	X	content	Sum

Note: Confirmation code=00H: success

Others: fail

Appendix

Dimensions of integral type Module (mm)

