R502 Fingerprint Module
User Manual

Hangzhou Grow Technology Co., Ltd.

2019　Ver: 1.2

GROW

Preface & Declaration

Thank you for you selection of R502 Fingerprint Identification Module of GROW.

The Manual is targeted for hardware & software development engineer, covering module function, hardware and software interface etc. To ensure the developing process goes smoothly, it is highly recommended the Manual is read through carefully.

Because of the products constantly upgraded and improved, module and the manual content may be changed without prior notice. If you want to get the latest information, please visit our company website (www.hzgrow.com).

We have been trying our best to enssure you the correctness of the Manual. However, if you have any question or find errorst, feel free to contact us or the authorized agent. We would be very grateful.

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

No responsibility or liability is assumed by GROW 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.

www.hzgrow.com

I Introduction

| Power	DC 3.3V	Interface	UART(3.3V TTL logical level)
Working current (Fingerprint acquisition)	18mA	Matching Mode	1:1 and 1:N
Standby current (finger detection)	Typical touch standby voltage: 3.3V Average current: 2uA	Characteristic value size	384 bytes
Baud rate	(9600*N)bps, N=1～6 (default N=6）	Template size	768 bytes
Image acquiring time	<0.2s	Image resolution	508dpi
Sensing Array	192*192 pixel	Detection Area	Diameter 15mm
Storage capacity	200	Security level	5 (1, 2, 3, 4, 5(highest))
FAR	<0.001%	FRR	<1%
Generate feature point time	< 500ms	Starting time	<30ms
Working environment	Temp: -20℃- +60℃	Storage environment	Temp: -40℃- +75℃
	RH: 10%-85%		RH: <85%

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   Hardware Interface

Exterior Interface

Module size: diameter: 22mm. Connector: MX1.0--6P
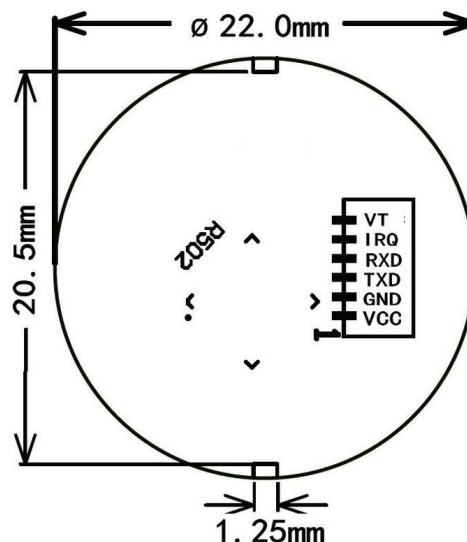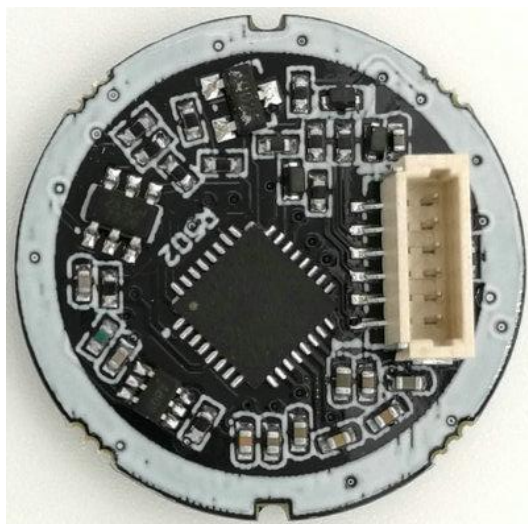
Serial Communication

When the FP module communicates with user device, definition of J1 is as follows:

|	Name	Type	Pin NmFunction Description
1	VCC		Power input DC 3.3V (Disconnect the power when standby)
2	GND		Signal ground. Connected to power ground
3	TXD	out	Data output. 3.3V TTL logical level
4	RXD	in	Data input. 3.3V TTL logical leve
5	IRQ	out	Finger detection signal
6	VT		Finger detection power (DC3.3V) (Often the power supply)

Hardware connection

Via serial interface, the Module may communicate with MCU of 3.3V or 5V power: TD (pin 3 of P1) connects with RXD (receiving pin of MCU), RD (pin 4 of P1) 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.

Serial communication protocol

The mode is semiduplex asychronism 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.
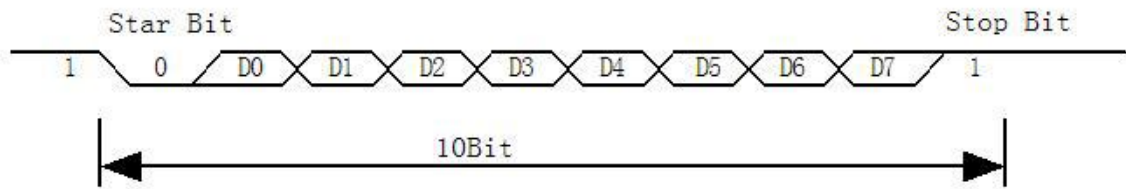
Reset time

At power on, it takes about 200ms for initialization. During this period, the Module can't accept commands for upper computer.

III   System Resources

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

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_Read Notepad.

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

Buffer

The module RAM resources are as follows:

An ImageBuffer: ImageBuffer

6 feature buffers: CharBuffer[1:6]

All buffer contents are not saved without power.

Fingerprint Library

Synstem 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.

Baud rate control (Parameter Number: 4)

The Parameter controls the UART communication speed of the Modul. Its value is an integer N, N= [1/2/4/6/12]. Cooresponding baud rate is 9600*N bps。

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

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.

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.

Module password

The default password of the module is 0x00000000. If the default password is modified, the first instruction of the upper computer to communicate with the module must be verify password. Only after the password verification is passed, the module will enter the normal working state and receive other instructions.

The new modified password is stored in Flash and remains at power off.(the modified password cannot be obtained through the communication instruction. If forgotten by mistake, the module cannot communicate, please use with caution)

Refer to instruction SetPwd and VfyPwd.

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.

Random number generator

Module integrates a hardware 32-bit random number generator (RNG) (without seed). Via instruction *GetRandomCode*, system will generate a random number and upload it.

Features and templates

The chip is equipped with a 72K byte image buffer and two character file buffers. The character file buffer is 768 bytes (each fingerprint character file size is 384 bytes).In this paper, the fingerprint character file of 384 bytes is taken as an example to narrate.The character file buffer can be used to store either normal character files or template character files.When uploading or downloading images through the UART port, in order to speed up, only the high four bits of pixel bytes are used, that is, two pixels are combined into one byte for transmission.USB port is the full 8 pixels.

Fingerprint character file size is 384 bytes, including character point information and overall information;The template size is 768 bytes, which is the sum of two identical fingerprint character.

Character file structure:

● The upper limit of the number of character points of a single character file is 50, and the character file takes up 384 bytes, among which the first 56 bytes are the file heads, which contain the overall information.The latter 328 bytes hold the information of 82 character points, each of which occupies 4 bytes.

● The file header format is as follows:

0~5 byte	6~39 byte	40~43 byte	44~55 byte
Symbol\Type\Character quality Character Number\Serial number	Background table: 34bytes	Two center coordinates	System reserved

Note:

1) Symbol: 1 byte. The character file logo is used to distinguish the character files generated by different sensors or different algorithms;The symbol field cannot be 0 when it is saved to the database. If it is 0, the character file is invalid or deleted.

2) Type: 1 byte.Refers to the character file type.

   0: The character file only contains file headers; 1: Simplified features;   2: Complete features.

3) Character quality: 1 byte. From 0 to 100, the higher the score, the higher the quality.

4) Number: 1 byte. From 5 to 82, refers to the number of character points.At least 5, at most 82.

5) Serial number: 2 bytes. Search assistance.

6) Background table: 34 bytes. Background compression table.

7) Singular point coordinates: 4 bytes. The X and Y coordinates of the two centers.

8) System reserved bytes: 12 bytes.

● Character unit structure

   Each feature unit takes up 4 bytes (32bits), as shown in the following table:

31~23 bit	22~14 bit	13~5 bit	4~1 bit	0 bit
x	y	Angle	Character point quality	Attribute

IV Communication Protocol

The protocol defines the data exchanging format when R502 series communicates with upper computer. The protocol and instruction sets apples for both UART and USB communication mode. For PC, USB interface is strongly recommended to improve the exchanging speed, especially in fingerprint scanning device.

4.1 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 (instuction/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 processs, 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 contens. Overflowing bits are omitted. high byte is transferred first.

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's definition is :

00h: commad execution complete;

01h: error when receiving data package;

02h: no finger on the sensor;

03h: fail to enroll the finger;

06h: fail to generate character file due to the over-disorderly fingerprint image;

07h: fail to generate character file due to lackness of character point or over-smallness of fingerprint image

08h: finger doesn't match;

09h: fail to find the matching finger;

0Ah: fail to combine the character files;

0Bh: addressing PageID is beyond the finger library;

0Ch: error when reading template from library or the template is invalid;

0Dh: error when uploading template;

0Eh: Module can't receive the following data packages.

0Fh: error when uploading image;

10h: fail to delete the template;

11h: fail to clear finger library;

13h: wrong password!

15h: fail to generate the image for the lackness of valid primary image;

18h: error when writing flash;

19h: No definition error;

1Ah: invalid register number;

1Bh: incorrect configuration of register;

1Ch: wrong notepad page number;

1Dh: fail to operate the communication port;

others: system reserved;

V Module Instruction System

R30X series provide 23 instructions. Through combination of different instructions, application program may realize muti finger authentication functions. All commands/data are transferred in package format. *Refer to 5.1 for the detailed information of package.*

System-related instructions

Verify passwoard   VfyPwd

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:

2 bytes	4bytes	1 byte	2 bytes	1 byte	4 byte	2 bytes
Header	Module address	Package identifier		Instruction code	Password	Checksum
0xEF01	xxxx	01H	07H	13H	PassWord	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
0xEF01	xxxx	07H	03H	xxH	sum

Note: Confirmation code = 00H: Correct password;

Confirmation code = 01H: error when receiving package;

Confirmation code = 13H: Wrong password;

Set password   SetPwd

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:

2 bytes	4bytes	1 byte	2 bytes	1 byte	4 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Password	Checksum
0xEF01	xxxx	01H	07H	12H	PassWord	sum

Acknowledge package format:

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

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

Confirmation code=01H: error when receiving package;

Set Module address   SetAdder

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

Input Parameter: None;

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
0xEF01	xxxx	01H	07H	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
0xEF01	xxxx	07H	07H	xxH	Sum

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

Confirmation code=01H: error when receiving package;

Set module system's basic parameter   SetSysPara

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:

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
0xEF01	Xxxx	01H	05H	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
0xEF01	Xxxx	07H	03H	xxH	Sum

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

Confirmation code=01H: error when receiving package;

Confirmation code=1aH: wrong register number;

Port Control   Control

Description:

For UART protocol, it control the "on/off" of USB port;

For USB protocol, it control the "on/off" of UART port;

Input Parameter: control code

Control code "0" means turns off the port;

Control code "1" means turns on the port;

Return Parameter: confirmation code;

Instruction code: 17H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1byte	2 bytes
Header	Chip address	Package identifier	Package length	Instruction code	Control code	Checksum
0xEF01	xxxx	01H	04H	17H	0/1	sum

Acknowledge package format:

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

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

Confirmation code=01H: error when receiving package;

Confirmation code=1dH: fail to operate the communication port;

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）

Instuction 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
0xEF01	Xxxx	01H	03H	0fH	sum

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
0xEF01	xxxx	07H	3+16	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

Read valid template number   TempleteNum

Description: read the current valid template number of the Module

Input Parameter: none

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

Instuction 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
0xEF01	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
0xEF01	xxxx	07H	5	xxH	N	sum

Note: Confirmation code=00H: read complete;

Confirmation code=01H: error when receiving package;

Fingerprint-processing instructions

To collect finger image   GenImg

Description: detecting finger and store the detected finger image in ImageBuffer while returning successfull 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)

Instuction 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
0xEF01	Xxxx	01H	03H	01H	05H

Acknowledge package format:

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

Note: Confirmation code=00H: finger collection successs;

Confirmation code=01H: error when receiving package;

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

Confirmation code=03H: fail to collect finger;

Upload image   UpImage

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)

Instuction 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
0xEF01	Xxxx	01H	03H	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
0xEF01	Xxxx	07H	03H	xxH	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.

Download the image   DownImage

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)

Instuction 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
0xEF01	Xxxx	01H	03H	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
0xEF01	Xxxx	07H	03H	xxH	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。

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)

Instuction 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
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:

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

To generate template   RegModel

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)

Instuction 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
0xEF01	xxxx	01H	03H	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
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.

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)

Instuction 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
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:

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

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)

Instuction 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
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:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes

Header	Module address	Package identifier	Package length	Confirmation code	Checksum
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.

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)

Instuction 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
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:

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

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)

Instuction 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
0xEF01	Xxxx	01H	07H	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
0xEF01	Xxxx	07H	03H	xxH	sum

Note: Confirmation code=00H: delete success;

Confirmation code=01H: error when receiving package;

Confirmation code=10H: faile to delete templates;

To empty finger library　　　　Empty

Description: to delete all the templates in the Flash library

Input Parameter: none

Return Parameter: Confirmation code (1 byte)

Instuction 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
0xEF01	Xxxx	01H	03H	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
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;

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.

Instuction 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
0xEF01	Xxxx	01H	03H	03H	07H

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
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.

To search finger library　　　Search

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）

Instuction 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	Parameter	Parameter	Checksum
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:

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

LED Control　　　　　LedConfig

Description: LED control

Input Parameter: Control code:Ctrl; ColorIndex

Return Parameter: Confirmation code

Instuction code: 0x35

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	1 byte	1 byte	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Control code		Color Index		Checksum
0xEF01	xxxx	0x01	0x0007	0x35	Ctrl	0x01	Color Index	0x01	sum

Control Code:

Control code	0x03	0x07
Function	Light on	Light off

ColorIndex:

Code	0x01	0x02	0x03
Color	Red	Blue	Purple

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
0xEF01	xxxx	0x07	0x0003	X	sum

Note 1: Confirmation code=0x00: success;

Confirmation code=0x01:error when receiving package;

Note: this instruction may conflict with the status of the module automatic control LED lamp, which needs to be adjusted by yourself.

Other instructions

To generate a random code   GetRandomCode

Description: to command the Module to generate a random number and return it to upper computer;Refer to 4.8 for more about Random Number Generator;

Input Parameter: none

Return Parameter: Confirmation code (1 byte)

Instuction code: 14H

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
0xEF01	xxxx	01H	03H	14H	0018H

Acknowledge package format:

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

Note: Confirmation code=00H: generation success;

Confirmation code=01H: error when receiving package;

To write note pad   WriteNotepad

Description: for upper computer to write data to the specified Flash page (refer to 4.1 for more about Note pad). Also see **ReadNotepad;**

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

Return Parameter: Confirmation code (1 byte)

Instuction 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
0xEF01	xxxx	01H	36	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
0xEF01	xxxx	07H	03H	xxH	sum

Note: Confirmation code=00H: write success;

Confirmation code=01H: error when receiving package;

To read note pad   ReadNotepad

Description: to read the specified page's data content; Refer to 4.1 for more about user note pad. Also see **WriteNotepad**.

Input Parameter: none

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

Instuction 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
0xEF01	xxxx	01H	04H	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
0xEF01	xxxx	07H	3+32	xxH	User content	sum

Note: Confirmation code=00H: read success;

Confirmation code=01H: error when receiving package;

Instruction Table

Classified by functions

type	num	code	description	Type	num	Code	description
System-related	1	13H	To verify password	Fingerprint processing	13	08H	to upload template
	2	12H	To set password		14	09H	To download template
	3	15H	To set device address		15	06H	To store template;
	4	0EH	To set system Parameter		16	07H	to read/load template
	5	17H	Port control		17	0CH	to delete tempates
	6	0FH	To read system Parameter		18	0DH	to empty the library
	7	1DH	To read finger template numbers		19	03H	Carry out precise matching of two templates;
Fingerprint processing	8	01H	Collect finger image		20	04H	Search the finger library
	9	0AH	To upload image				
	10	0BH	To download image	others	21	14H	to get random code
	11	02H	To generate character file from image		22	18H	to write note pad
	12	05H	To combine character files and generate template		23	19H	To read note pad

Classified by instruction code

code	identifier	Description	Code	Identifier	Description
01H	GenImg	Collect finger image	0DH	Empty	to empty the library
02H	Img2Tz	To generate character file from image	0EH	SetSysPara	To set system Parameter
03H	Match	Carry out precise matching of two templates;	0FH	ReadSysPara	To read system Parameter
04H	Serach	Search the finger library	12H	SetPwd	To set password
05H	RegModel	To combine character files and generate template	13H	VfyPwd	To verify password
06H	Store	To store template;	14H	GetRandomCode	to get random code
07H	LoadChar	to read/load template	15H	SetAdder	To set device address
08H	UpChar	to upload template	17H	Control	Port control
09H	DownChr	to download template	18H	WriteNotepad	to write note pad
0AH	UpImage	To upload image	19H	ReadNotepad	To read note pad
0BH	DownImage	To download image	1DH	TempleteNum	To read finger template numbers
0CH	DeletChar	to delete tempates	0x35	LedConfig	LED control

Ⅶ Reference Circuit