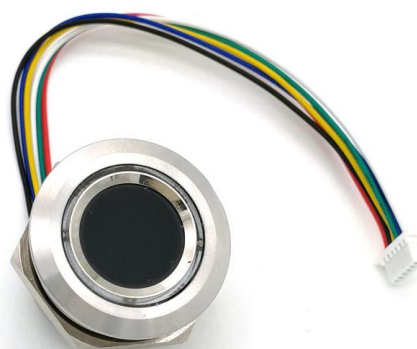


R503 Fingerprint Module User Manual



2019.6 Ver: 1.1

Preface & Declaration

Thank you for your selection of R503 Fingerprint Identification Module

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

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

I Introduction

Power	DC 3.3V	Interface	UART(3.3V TTL logical level)
Working current (Fingerprint acquisition)	20mA	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°C- +60°C	Storage environment	Temp: -40°C- +75°C
	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

Connector: MX1.0--6P Thread:M25

Product external diameter: 28mm Inner diameter:23.5mm Height:19mm



Serial Communication

Connector: MX1.0--6P

Pin	Name	Description	Pic
1	Power Supply	DC3.3V	
2	GND	Signal ground. Connected to power ground.	
3	TXD	Data output. TTL logical level	
4	RXD	Data input. TTL logical level	
5	WAKEUP	Finger Detection Signal	
6	3.3VT	Touch induction power supply, DC3—6V	

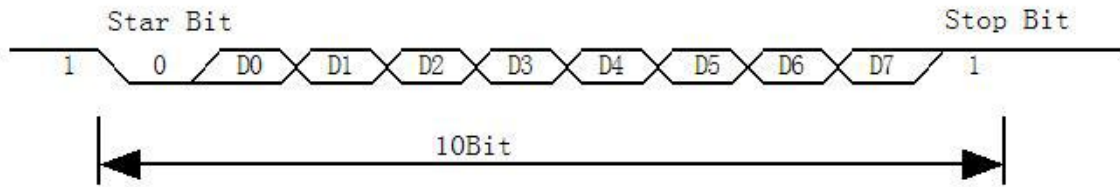
Hardware connection

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 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.



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

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.

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 has an image buffer and six feature file buffers, all buffer contents are not saved after power failure.

A template can be composed of 2-6 feature files, the more characteristic files the composite template has, the better the quality of the fingerprint template,

At least 3 template synthesis features are recommended.

IV Communication Protocol

The protocol defines the data exchanging format when R503 series communicates with upper computer. The protocol and instruction sets applies 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 multi 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 password 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 address 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 TemplateNum

Description: read the current valid template number of the Module

Input Parameter: none

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

Instruction code: 1dH

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Checksum
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=0x00: read success;

Confirmation code=0x01: error when receiving package;

Read fingerprint template index table ReadIndexTable (0x1F)

Description: Read the fingerprint template index table of the module, read the index table of the fingerprint template up to 256 at a time (32 bytes)

Input Parameter: Index page

Return Parameter: Confirmation code+Fingerprint template index table

Instruction code: 0x1F

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Index page	Checksum
0xEF01	xxxx	0x01	0x0004	0x1F	0/1/2/3	Sum

Index tables are read per page, 256 templates per page

Index page 0 means to read 0 ~ 255 fingerprint template index table

Index page 1 means to read 256 ~ 511 fingerprint template index table

Index page 2 means to read 512 ~ 767 fingerprint template index table

Index page 3 means to read 768 ~ 1023 fingerprint template index table

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	32 bytes	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Index page	Checksum
0xEF01	xxxx	0x07	0x0023	X	See the table below	sum

Note: Confirmation code=0x00: read complete;

Confirmation code=0x01: error when receiving package;

Index table structure: every 8 bits is a group, and each group is output starting from the high position.

transport order	The output is sequential from low byte to high byte, and each byte starts at a high byte.								
T[0]	Template number	7	6	5	4	3	2	1	0
	Index table data	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
T[1]	Template number	15	14	13	12	11	10	9	8
	Index table data	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
...	...								
T[31]	Template number	255	254	253	252	251	250	249	248
	Index table data	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

Data "0" in the index table means that there is no valid template in the corresponding position;"1" means that there is a valid template in the corresponding position.

Fingerprint-processing instructions

To collect finger image **GenImg**

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

Input Parameter: none

Return Parameter: Confirmation code (1 byte)

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 success;

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.

Data package format:

2 bytes	4bytes	1 byte	2 bytes	N byte	2 bytes
Header	Module address	Package identifier	Package length	Package content	Checksum
0xEF01	xxxx	0x02- have following packet 0x08 - end packet	N+2	Image data	sum

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.

Data package format:

2 bytes	4bytes	1 byte	2 bytes	N byte	2 bytes
Header	Module address	Package identifier	Package length	Package content	Checksum
0xEF01	xxxx	0x02- have following packet 0x08 - end packet	N+2	Image data	sum

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.

Data package format:

2 bytes	4bytes	1 byte	2 bytes	N byte	2 bytes
Header	Module address	Package identifier	Package length	Package content	Checksum
0xEF01	xxxx	0x02- have following packet 0x08 - end packet	N+2	Template data	sum

Download template DownChar

Description: upper computer download template to module buffer

Input Parameter: CharBufferID (Buffer number)

Return Parameter: Confirmation code (1 byte)

Instuction code: 09H

Command (or instruction) package format:

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

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

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
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: can not receive the following data packet

Data package format:

2 bytes	4bytes	1 byte	2 bytes	N byte	2 bytes
Header	Module address	Package identifier	Package length	Package content	Checksum
0xEF01	xxxx	0x02- have following packet 0x08 - end packet	N+2	Template data	sum

Note2: 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)

Instruction code: 06H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	buffer number	Location number	Checksum
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)

Instruction code: 07H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	buffer number	Page number	Checksum
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	Page	Score	Checksum
0xEF01	xxxx	07H	7	xxH	PageID	MatchScore	sum

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

Confirmation code=01H: error when receiving package;

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

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

Fingerprint image collection extension command **GetImageEx(0x28)**

Description: Detect the finger, record the fingerprint image and store it in ImageBuffer, return it and record the successful confirmation code; If no finger is detected, return no finger confirmation code (the module responds quickly to each instruction, therefore, for continuous detection, cycle processing is required, which can be limited to the number of cycles or the total time).

Differences between GetImageEx and the GetImage:

GetImage: return the confirmation code 0x00 when the image quality is too bad (image collection succeeded)

GetImageEx: return the confirmation code 0x07 when the image quality is too bad (poor collection quality)

Input Parameter: none

Return Parameter: Confirmation code

Instruction code: 0x28

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	0x01	0x0003	0x28	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	0x07	0x0003	X	sum

Note 1: Confirmation code=0x00: read success

Confirmation code=0x01: error when receiving package;

Confirmation code=0x02: no fingers on the sensor;

Confirmation code=0x03: unsuccessful entry

Confirmation code=0x07: poor image quality;

Cancel instruction **Cancel(0x30)**

Description: Cancel instruction

Input Parameter: none

Return Parameter: Confirmation code

Instruction code: 0x30

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	0x01	0x0003	0x30	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	0x07	0x0003	X	sum

Note 1: Confirmation code=0x00: cancel setting successful

Confirmation code=other: cancel setting failed

HandShake HandShake (0x40)

Description: Send handshake instructions to the module. If the module works normally, the confirmation code 0x00 will be returned. The upper computer can continue to send instructions to the module. If the confirmation code is other or no reply, it means that the device is abnormal.

Input Parameter: none

Return Parameter: Confirmation code

Instuction code: 0x40

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	0x01	0x0003	0x40	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	0x07	0x0003	X	sum

Note 1: Confirmation code=0x00: the device is normal and can receive instructions;

Confirmation code=other: the device is abnormal.

In addition, after the module is powered on, 0x55 will be automatically sent as a handshake sign. After the single-chip microcomputer detects 0x55, it can immediately send commands to enter the working state.

CheckSensor CheckSensor (0x36)

Description: Check whether the sensor is normal

Input Parameter: none

Return Parameter: Confirmation code

Instuction code: 0x36

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	0x01	0x0003	0x36	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	0x07	0x0003	X	sum

Note 1: Confirmation code=0x00: the sensor is normal;
Confirmation code=0x29: the sensor is abnormal.

Get the algorithm library version GetAlgVer (0x39)

Description: Get the algorithm library version

Input Parameter: none

Return Parameter: Confirmation code+AlgVer(algorithm library version string)

Instuction code: 0x39

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	0x01	0x0003	0x39	sum

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	32 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Random number	Checksum
0xEF01	xxxx	0x07	0x0023	X	AlgVer	sum

Note 1: Confirmation code=0x00: success;
Confirmation code=0x01: error when receiving package;

Get the firmware version GetFwVer (0x3A)

Description: Get the firmware version

Input Parameter: none

Return Parameter: Confirmation code+FwVer(Firmware version string)

Instuction code: 0x3A

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	0x01	0x0003	0x3A	sum

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	32 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Random number	Checksum
0xEF01	xxxx	0x07	0x0023	X	FwVer	sum

Note 1: Confirmation code=0x00: success;

Confirmation code=0x01: error when receiving package;

Read product information ReadProdInfo (0x3C)

Description: Read product information

Input Parameter: none

Return Parameter: Confirmation code+ProdInfo(product information)

Instuction code: 0x3C

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	0x01	0x0003	0x3C	sum

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	46 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Product information	Checksum
0xEF01	xxxx	0x07	0x0031	X	ProdInfo	sum

Note 1: Confirmation code=0x00: success;

Confirmation code=0x01: error when receiving package;

Product information: store in the following order.For Numbers, the high byte comes first.For a string, the insufficient part is 0x00.

Code	Bytes	Meaning
PARAM_FPM_MODEL	16	module type, ASCII
PARAM_BN	4	Module batch number, ASCII
PARAM_SN	8	Module serial number, ASCII
PARAM_HW_VER	2	For the hardware version, the first byte represents the main version and the second byte represents the sub-version
PARAM_FPS_MODEL	8	Sensor type, ASCII
PARAM_FPS_WIDTH	2	Sensor image width
PARAM_FPS_HEIGHT	2	Sensor image height
PARAM_TMPL_SIZE	2	Template size
PARAM_TMPL_TOTAL	2	Fingerprint database size

Soft reset SoftRst (0x3D)

Description: Send soft reset instruction to the module. If the module works normally, return confirmation code 0x00, and then perform reset operation.

Input Parameter: none

Return Parameter: Confirmation code

Instuction code: 0x3D

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	0x01	0x0003	0x3D	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	0x07	0x0003	X	sum

Note 1: Confirmation code=0x00: success;

Confirmation code=other: device is abnormal

After module reset, 0x55 will be automatically sent as a handshake sign. After the single-chip microcomputer detects 0x55, it can immediately send commands to enter the working state.

Aura control AuraLedConfig (0 x35)

Description: Aura LED control

Input Parameter: Control code:Ctrl; Speed; ColorIndex;Times

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	Speed	Color Index	Times	Checksum
0xEF01	xxxx	0x01	0x0007	0x35	Ctrl	Speed	Color Index	Count	sum

Control Code:

Control code	0x01	0x02	0x03	0x04	0x05	0x06
Function	breathing light	Flashing light	Light Always on	Light Always off	Light gradually on	Light gradually off

Speed: 0x00-0xff, 256 gears,Minimum 5s cycle.

It is effective for breathing lamp and flashing lamp,Light gradually on,Light gradually off

ColorIndex:

Code	0x01	0x02	0x03
Color	Red	Blue	Purple

Number of cycles: 0- infinite, 1-255.

It is effective for with breathing light and flashing light.

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;

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 read information page **ReadInfPage**

Description: read information page(512bytes)

Input Parameter: none

Return Parameter: Confirmation code (1 byte)

Instuction code: 16H

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	16H	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 1: Confirmation code=00H: ready to transfer the following data packet;
Confirmation code=01H: error when receiving package;
Confirmation code=0fH: can not transfer the following data packet;
2: Module shall transfer following data packet after responding to the upper computer.;
3: The instruction doesn't affect buffer contents.

Data package format:

2 bytes	4bytes	1 byte	2 bytes	N byte	2 bytes
Header	Module address	Package identifier	Package length	Package content	Checksum
0xEF01	xxxx	0x02- have following packet 0x08 - end packet	N+2	Information page	sum

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	others	20	04H	Search the finger library
	9	0AH	To upload image				
	10	0BH	To download image		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	16H	ReadInfPage	Read information page
09H	DownChr	to download template	17H	Control	Port control
0AH	UpImage	To upload image	18H	WriteNotepad	to write note pad
0BH	DownImage	To download image	19H	ReadNotepad	To read note pad
0CH	DeletChar	to delete tempates	1DH	TempleteNum	To read finger template numbers
0x1F	ReadIndexTable	Read fingerprint template index table	0x28	GetImageEx	Fingerprint image collection extension command
0x30	Cancel	Cancel instruction	0x40	HandShake	HandShake
0x36	CheckSensor	CheckSensor	0x39	GetAlgVer	Get the algorithm library version
0x3A	GetFwVer	Get the firmware version	0x3C	ReadProdInfo	Read product information
0x3D	SoftRst	Soft reset	0x35	Aura control	AuraLedConfig

VII Reference Circuit

