R503/R503-M22 Fingerprint Module
User Manual

Hangzhou Grow Technology Co., Ltd.

2023.09 Ver: 1.4.1

Preface & Declaration

Thank you for you selection of R503/R503-M22 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 ensure you the correctness of the Manual. However, if you have any question or find error, feel free to contact us or the authorized agent. We would be very grateful.

www.hzgrow.com

Revised Version

Version Number	Date	Revise Content	Modifier
V1.2	2021.10	1. LED colors increased from three to seven, and the instruction formats were downward compatible. 2. Added 0x31 automatic registration template: the upper computer can automatically collect 6 images by sending only one command, and then generate templates for saving. 3. Added 0x32 automatic fingerprint verification: the upper computer can only send one command to realize image collection, generate features, search and compare the fingerprint database, and return the comparison results 4. Added template upload procedure flow 5. Added template download process	Grow Tech
V1.2.1	2022.03	1. Added Power Supply Requirements,Ripple noise 2. Update Command: The note of UpChar and DownChar	Grow Tech
V1.2.2	2022.10	1. Added Buffer contents	Grow Tech
V1.3	2023.06	1. Added Basic communication flow and general instruction communication flow 2. Updated Acknowledge package format of Read product information Command (0x3C) 3. Updated Automatic fingerprint verification Command (0x32) 4. Updated Automatic registration template Command (0x31)	Grow Tech
V1.4	2023.07	1. Added R503-M22 Size Version	Grow Tech
V1.4.1	2023.09	1. Updated Security level,Checksum 2. Added example instruction for AuraLedConfig, AutoEnroll, AutoIdentify	Grow Tech

Catalog

I Introduction

Power	DC 3.3V	Interface	UART(3.3V TTL logical level)
Working current (Fingerprint acquisition)	20mA	**Matching Mode** **Matching Time**	1:1 and 1:N 1:N<10ms/Fingerprint
Standby current (finger detection)	Typical touch standby voltage: 3.3V Average current: 2uA	**Characteristic value size**	512 bytes
Baud rate	(9600*N)bps, N=1～6 (default N=6）	**Template size**	1536 bytes
Image acquiring time	<0.2s	**Image resolution**	508dpi
Sensing Array	192*192 pixel	**Detection Area**	Diameter 15mm
Storage capacity	200	**Security level**	3 (1, 2, 3, 4, 5(highest))
FAR	<0.001%	**FRR**	<1%
Generate feature point time	< 500ms	**Starting time**	≤50ms
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

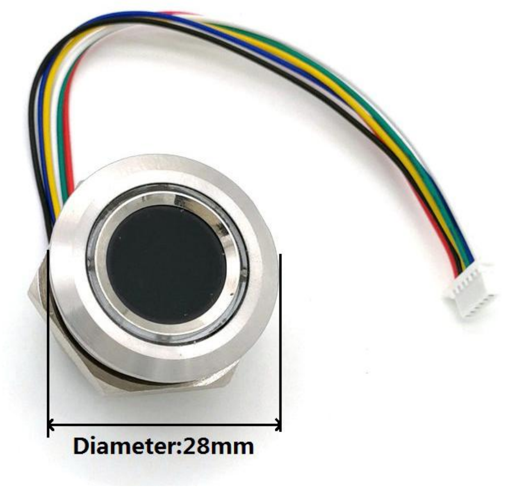
R503 Info

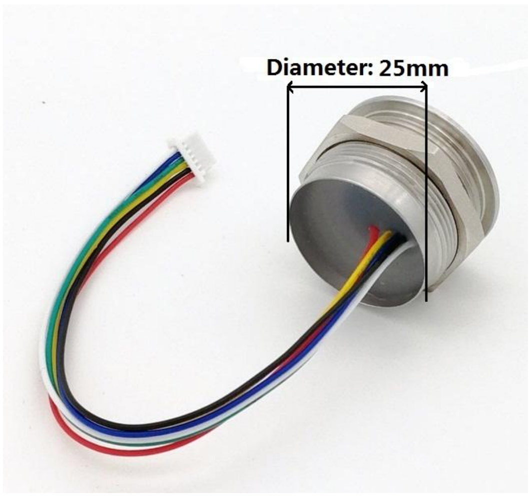
Connector: SH1.0--6P Thread:M25

Product external diameter: 28mm Inner diameter:25mm Height:19mm

Enclosure material: Zinc Alloy

(Standard height is 19mm, also have 15mm and 32mm height,or need black aluminium alloy enclosure, pls contact sales, support customized)



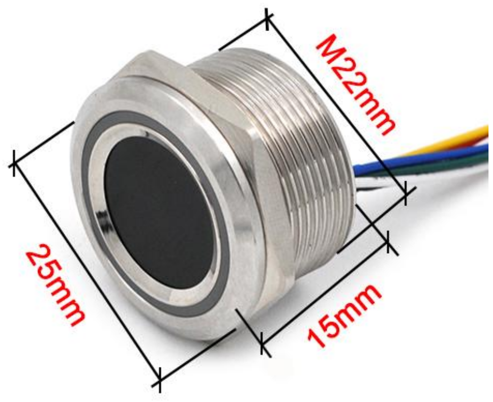


R503-M22 Info

Connector: SH1.0--6P Thread:M22

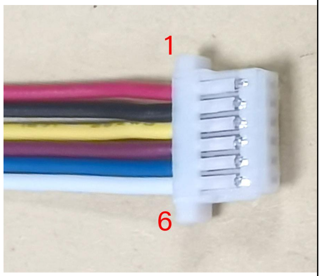
Enclosure material: Zinc Alloy

Product external diameter: 25mm Inner diameter:22mm Height:15mm



Serial Communication

Connector: SH1.0--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. Standby-high level, have finger-output low level.	
6	3.3VT	Touch induction power supply, DC3—5V	Note: The line order has nothing to do with color.

Hardware Connection

The RX of the module is connected with the TX of the upper computer, and the TX of the module is connected with the RX of the upper computer. The IRQ signal can be connected with the middle fracture or IO port of the upper computer.

To reduce the system standby power consumption,when the upper computer needs to use the fingerprint module,then power on the main power supply of the fingerprint module. At this time, the fingerprint module is powered on,and complete the corresponding instructions sent by the upper computer.When the upper computer does not need to use the fingerprint module, disconnect the fingerprint module from the main power supply.

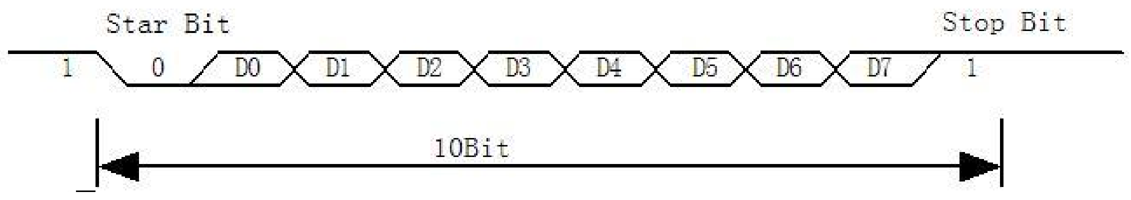
When the upper computer is in standby mode, in order to keep the finger touch detection, the touch power supply needs to be powered all the time. The working voltage of the touch power supply is 3V~5V, and the average current of the touch power supply is about 2uA. When there is no finger touch, the default touch sensing signal outputs high level; When a finger touches, the default touch sensing signal outputs low level. After detecting the touch sensing signal, the upper computer supplies power to the fingerprint module and the fingerprint module starts to work.

The maximum response time of the touch function is about 120mS @vt =3.3V. When the module is not touched, the recalibration period is about 4.0sec; the touch signal output is CMOS output, and the output voltage is roughly the same as the input voltage.

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.



Power-on delay time

At power on, it takes about 50ms for initialization. During this period, the module can't accept commands for upper computer.After completing the initialization, the module will immediately send a byte (0x55) to the upper computer, indicating that the module can work normally and receive instructions from the upper computer.

Power Supply Requirements

The power supply is DC +3.3V. The power input is allowed only after the R503/R503-M22 is properly connected.
Electrical components of the R503/R503-M22 may be damaged if you insert or remove the cable (with the electric plug) when the cable is live. Ensure that the power supply is switched off when you insert or remove the cable.
The R503/R503-M22 may not work properly due to poor power connections, short power off/on intervals, or excessive voltage drop pulses. So pls keep the power is stable. After the power is turned off, the power must be turned on at least two seconds later.

Ripple noise

Since the power input of R503/R503-M22 is directly supplied to the image sensor and decoding chip.
To ensure stable operation, pls use low ripple noise power input.
It is recommended that the ripple noise not exceed 50mV (peak-to-peak).

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.

The user can run the module address or random number command to configure the unique matching between the module and the system. That is, the system identifies only the unique module. If a module of the same type is replaced, the system cannot access the system.

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.

The user can read and write any buffer by instruction. CharBuffer can be used to store normal feature files or store template feature files.

When uploading or downloading images through the UART port, only the high four bits of pixel bytes are used to speed up the transmission, that is, use gray level 16, two pixels are combined into one byte. (The high four bits are a pixel, the low four bits are a pixel in the next adjacent column of the same row, that is, two pixels are combined into one byte and transmitted)

Since the image has 16 gray levels, when it is uploaded to PC for display (corresponding to BMP format), the gray level should be extended (256 gray levels, that is, 8bit bitmap format).

Fingerprint Library

System sets aside a certain space within Flash for fingerprint template storage, that's fingerprint library. The contents of the fingerprint database are protected by power-off, and the serial number of the fingerprint database starts from 0.

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.

System Configuration Parameters

The system allows the user to individually modify a specified parameter value (by parameter serial number) by command. Refer to *SetSysPara*. After the upper computer sets the system parameter instructions, the system must be powered on again so that the module can work according to the new configuration.

Baud rate control (Parameter Number: 4)

The Parameter controls the UART communication speed of the Module. Its value is an integer N, N= [1/2/4/6/12]. Corresponding 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 corresponding 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, after the module is powered on,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 *SetAddr*. 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 one 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 feature files in the synthesis template, the better the quality of the fingerprint template.

It is recommended to take at least four templates to synthesize features.

IV Communication Protocol

The protocol defines the data exchanging format when R503/R503-M22 series communicates with upper computer. The protocol and instruction sets apples for both UART communication mode. Baud rate 57600, data bit 8, stop bit 1, parity bit none.

Data package format

When communicating, the transferring and receiving of command/data/result are all wrapped in data package format. For multi-bytes, the high byte precedes the low byte (for example, a 2 bytes 00 06 indicates 0006, not 0600).

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 Table

Code	Identifier	Description	Code	Identifier	Description
01H	GenImg	Collect finger image	12H	SetPwd	To set password
02H	Genchar	To generate character file from image	13H	VfyPwd	To verify password
03H	Match	Carry out precise matching of two templates;	14H	GetRandomCode	to get random code
04H	Search	Search the finger library	15H	SetAdder	To set device address
05H	RegModel	To combine character files and generate template	16H	ReadInfPage	Read information page
06H	Store	To store template;	18H	WriteNotepad	to write note pad
07H	LoadChar	to read/load template	19H	ReadNotepad	To read note pad
08H	UpChar	to upload template	1DH	TempleteNum	To read finger template numbers
09H	DownChar	to download template	1FH	ReadIndexTable	Read-fingerprint template index table
0AH	UpImage	To upload image	0x28	GetImageEx	Fingerprint image collection extension command
0BH	DownImage	To download image	0x30	Cancel	Cancel instruction
0CH	DeletChar	to delete templates	0x40	HandShake	HandShake
0DH	Empty	to empty the library	0x36	Check Sensor	CheckSensor
0EH	SetSysPara	To set system Parameter	0x39	GetAlgVer	Get the algorithm library version
0FH	ReadSysPara	To read system Parameter	0x3C	ReadProdInfo	Read product information
0x3A	GetFwVer	Get the firmware version	0x35	Aura control	AuraLedConfig
0x3D	SoftRst	Soft reset	0x32	AutoIdentify	Automatic fingerprint verification
0x31	AutoEnroll	Automatic registration template			

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: command 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;
20h: the address code is incorrect;
21h: password must be verified;
22h: fingerprint template is empty;
24h: fingerprint library is empty;
26h: timeout
27h: fingerprints already exist;
29h: sensor hardware error;
1Ah: invalid register number;
1Bh: incorrect configuration of register;
1Ch: wrong notepad page number;
1Dh: fail to operate the communication port;
1Fh: fingerprint library is full;
FCh: unsupported command;
FDh: hardware error;
FEh: command execution failure;
others: system reserved;

V Module Instruction System

System-related instructions

Verify password VfyPwd

Description: Verify Module's handshaking password.
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 bytes	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Password	Checksum
0xEF01	xxxx	01H	0007H	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	0003H	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.
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 bytes	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Password	Checksum
0xEF01	xxxx	01H	0007H	12H	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	0003H	xxH	sum

Note: Confirmation code=00H: password setting complete;
Confirmation code=01H: error when receiving package;
Confirmation code=21H: have to verify password
Confirmation code=18H: error when write FLASH

Set Module address SetAdder

Description: Set Module address.

Input Parameter: Addr

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	0007H	15H	Addr	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	0003H	xxH	Sum

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

Confirmation code=01H: error when receiving package;

Confirmation code=18H: error when write FLASH

Set module system's basic parameter SetSysPara

Description: Operation parameter settings.

Input Parameter: Parameter number+Contents

Return Parameter: Confirmation code (1 byte)

Instruction code: 0eH

Command (or instruction) package format:

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

Acknowledge package format:

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

Confirmation code=18H: error when write FLASH

Name	Parameter number	Content
Baud rate	4	Data range:1, 2/4/6/12, indicates that baud rate is 9600 * N bps
Security level	5	Data range: 1, 2, 3, 4, 5
Packet content length	6	Data range: 0, 1, 2, 3 the corresponding lengths (bytes) are as follows: 32, 64, 128, 256

Read system Parameter ReadSysPara

Description: Read Module's status register and system basic configuration parameters;

Input Parameter：none

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

Instruction code: 0fH

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Checksum
0xEF01	Xxxx	01H	0003H	0fH	0013H

Acknowledge package format:

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

Note: Confirmation code=00H: read complete;

Confirmation code=01H: error when receiving package;

Confirmation code=18H: error when write FLASH

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

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

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	0005H	xxH	Num	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	Check-sum
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.

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)

Instruction 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	003DH

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	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)

Instruction 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	003EH

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	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)

Instruction 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	0040H

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	50 bytes	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
Other	4	System Reserved

Fingerprint-processing instructions

To collect finger image GetImg

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

The difference between GetImageEx and GetImage instruction:

GetImage: When the image quality is poor, return confirmation code 0x00 (the image is successfully captured).

GetImageEx: When image quality is poor, return confirmation code 0x07 (image quality is too poor).

Input Parameter: none

Return Parameter: Confirmation code (1 byte)

Instruction code: 01H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Checksum
0xEF01	Xxxx	01H	0003H	01H	0005H

Acknowledge package format:

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

To upload image UpImage

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

Input Parameter: none

Return Parameter: Confirmation code (1 byte)

Instruction code: 0aH

Command (or instruction) package format:

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

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
0xEF01	Xxxx	07H	0003H	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. The upper computer sends the command packet, the module sends the acknowledge packet first, and then sends several data packet.
3. Packet Bytes N is determined by Packet Length. The value is 128 Bytes before delivery.

Data package format:

2 bytes	4bytes	1 byte	2 bytes	N bytes	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 download the image DownImage

Description: to download image from upper computer to Img_Buffer.
Input Parameter: none
Return Parameter: Confirmation code (1 byte)
Instruction code: 0bH
Command (or instruction) package format:

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

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
0xEF01	Xxxx	07H	0003H	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.The upper computer sends the command packet, the module sends the acknowledge packet first, and then sends several data packet.
3.Packet Bytes N is determined by Packet Length. The value is 128 Bytes before delivery.

Data package format:

2 bytes	4bytes	1 byte	2 bytes	N bytes	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 GenChar

Description: to generate character file from the original finger image in ImageBuffer

Input Parameter: BufferID (character file buffer number)

Return Parameter: Confirmation code (1 byte)

Instruction code: 02H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Buffer number	Checksum
0xEF01	xxxx	01H	0004H	02H	CharBuffer ID	sum

CharBufferID: Character buffer number, range 1-6.

The R300-A module requires a minimum of four and a maximum of six fingerprint features for the generate template.

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	0003H	XxH	sum

Note: Confirmation code=00H: generate character file complete;

Confirmation code=01H: error when receiving package;

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

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

Confirmation code=15H: fail to generate the image for the lackness of valid primary image;

To generate template RegModel

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

Input Parameter：none

Return Parameter：Confirmation code (1 byte)

Instruction code: 05H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Checksum
0xEF01	xxxx	01H	0003H	05H	0009H

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	0003H	xxH	sum

Note: Confirmation code=00H: operation success;

Confirmation code=01H: error when receiving package;

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

To upload template UpChar

Description: Upload the data in the template buffer ModelBuffer to the upper computer.

Input Parameter: CharBufferID

Return Parameter: Confirmation code (1 byte)

Instruction code: 08H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Buffer number	Checksum
0xEF01	xxxx	01H	0004H	08H	CharBuffer ID	sum

Note: This command don't need to use the CharBufferID, so the CharBufferID can be any value between 1 and 6.

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	0003H	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;

Confirmation code=0fH: can not receive the following data packet

4. The upper computer sends the command packet, the module sends the acknowledge packet first, and then sends several data packet.

5. Packet Bytes N is determined by Packet Length. The value is 128 Bytes before delivery.

6: The instruction doesn't affect buffer contents.

Data package format:

2 bytes	4bytes	1 byte	2 bytes	N bytes	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

To download template DownChar

Description: upper computer download template to module buffer

Input Parameter: CharBufferID (Buffer number)

Return Parameter: Confirmation code (1 byte)

Instruction code: 09H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Buffer number	Checksum
0xEF01	xxxx	01H	0004H	09H	CharBufferID	sum

Note: This command don't need to use the CharBufferID, so the CharBufferID can be any value between 1 and 6.

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	0003H	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 bytes	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

Note 2. The upper computer sends the command packet, the module sends the acknowledge packet first, and then sends several data packet.

3.Packet Bytes N is determined by Packet Length. The value is 128 Bytes before delivery.

4. 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: CharBufferID, ModelID

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	0006H	06H	CharBuffer ID	ModelID	sum

Note: CharBufferID is filled with 0x01

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	0003H	xxH	sum

Note: Confirmation code=00H: storage success;

Confirmation code=01H: error when receiving package;

Confirmation code=0bH: addressing ModelID 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: CharBufferID, ModelID

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	0006H	07H	CharBuffer ID	ModelID	sum

Note: CharBufferID is filled with 0x01

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	0003H	XxH	sum

Note: Confirmation code=00H: load success;

Confirmation code=01H: error when receiving package;

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

Confirmation code=0BH: addressing ModelID 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: StartID + Num

Return Parameter: Confirmation code (1 byte)

Instruction code: 0cH

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes	2bytes	2 bytes
Header	Module	Package	Package	Instruction	Page	number of	Checksum

	address	identifier	length	code	number	templates to be deleted	
0xEF01	Xxxx	01H	0007H	0cH	StartID	Num	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	0003H	xxH	sum

Note: Confirmation code=00H: delete success;
Confirmation code=01H: error when receiving package;
Confirmation code=10H: faile to delete templates;
Confirmation code=18H: error when write FLASH

To empty finger library Empty

Description: to delete all the templates in the Flash library
Input Parameter: none
Return Parameter: Confirmation code (1 byte)
Instruction code: 0dH
Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Checksum
0xEF01	Xxxx	01H	0003H	0dH	0011H

Acknowledge package format:

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

Note: Confirmation code=00H: empty success;
Confirmation code=01H: error when receiving package;
Confirmation code=11H: fail to clear finger library;
Confirmation code=18H: error when write FLASH

To carry out precise matching of two finger templates Match

Description: Compare the recently extracted character with the templates in the ModelBuffer, providing matching results.
Input Parameter: none
Return Parameter: Confirmation code (1 byte)，matching score.
Instruction code: 03H
Command (or instruction) package format:

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

0xEF01	Xxxx	01H	0003H	03H	0007H

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Matching score	Checksum
0xEF01	Xxxx	07H	0005H	XxH	MatchScore	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: CharBufferID + StartID + Num
Return Parameter: Confirmation code+ModelID(template number)+ MatchScore
Instruction code: 04H
Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	buffer number	Parameter	Parameter	Checksum
0xEF01	xxxx	01H	0008H	04H	CharBufferID	StartID	Num	sum

Note: CharBufferID is filled with 0x01

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	0007H	xxH	Model ID	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.

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	002CH

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	0034H

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

Instruction 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	0044H

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

Instruction 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	003AH

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.

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
Instruction 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	0041H

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
Instruction 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	0x04	0x05	0x06	0x07
Color	Red	Blue	Purple	Green	Yellow	Cyan	White

Number of cycles: 0- infinite, 1-255.
It is effective for with breathing light and flashing light.

Example of setting light: Cyan breathing light:
Send: EF 01 FF FF FF FF 01 00 07 35 01 50 06 00 00 94
Return: EF 01 FF FF FF FF 07 00 03 00 00 0A

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;

Automatic registration template AutoEnroll (0 x31)

When a fingerprint is recorded using an automatic registration template, the fingerprint image needs to be recorded six times for each fingerprint template. The blue light blinks when the fingerprint image is collected. The yellow light is on means the fingerprint image is collected successfully,the green light blinks means the fingerprint characteristic is generated successfully. If the finger is required to leave during image collection, the image will be collected again after the finger is lifted. During the process of waiting for the finger to leave, the white light flashes. After fingerprint images are collected for 6 times and features are generated successfully, features are synthesized and store fingerprint template. If the operation succeeds, the green light is on; if the operation fails, the red light is on. If the finger is away from the sensor for more than 10 seconds when in collecting the fingerprint image each time, it will automatically exits the automatic template registration process.

Input Parameter:ModelID- Fingerprint library location number
Config1: Whether to allow cover ID number
Config2: Whether to allow duplicate fingerprints
Config3: Whether the module return the status in the critical step
Config4: Whether to allow ask the finger to leave

Return Parameter:Confirmation code ModelID(Fingerprint library location number)
Instruction code: 0x31

Command (or instruction) package format:

2 bytes	4 bytes	1 byte	2 bytes	1 byte	1 byte	1 byte	1 byte	1 byte	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Location	Whether Duplicat	Whether Duplicate	Whether return	Whether ask finger	Check sum

					ID	e ID	Fingerprint	status	to leave	
0xEF 01	xxxx	0x01	0x0008	0x31	ID	Config1	Config2	Config 3	Config4	sum

Model ID: Location ID : 0-0xC7

0xC8-0xFF is automatic filling(The ID number is assigned by the system. The system will be starting from template 0 to searches the empty templates.)

Whether to allow cover ID number: 0:Not allowed 1: Allow
Whether to allow register duplicate fingerprints: 0:Not allowed 1: Allow
Whether to return to the critical step status during registration: 0:Not allowed 1: Allow
Whether the finger is required to leave during the registration process in order to enter the next fingerprint image collection: 0: don't need to leave 1: have to leave

Example of AutoEnroll:
Send: EF 01 FF FF FF FF 01 00 08 31 C8 00 01 01 01 01 05
Return: EF 01 FF FF FF FF 07 00 05 00 01 00 00 0D EF 01 FF FF FF FF 07 00 05 00 02 00 00 0E EF 01 FF FF FF FF 07 00 05 00 03 00 00 0F EF 01 FF FF FF FF 07 00 05 00 04 00 00 10 EF 01 FF FF FF FF 07 00 05 00 05 00 00 11 EF 01 FF FF FF FF 07 00 05 00 06 00 00 12 EF 01 FF FF FF FF 07 00 05 00 07 00 00 13 EF 01 FF FF FF FF 07 00 05 00 08 00 00 14 EF 01 FF FF FF FF 07 00 05 00 09 00 00 15 EF 01 FF FF FF FF 07 00 05 00 0A 00 00 16 EF 01 FF FF FF FF 07 00 05 00 0B 00 00 17 EF 01 FF FF FF FF 07 00 05 00 0C 00 00 18 EF 01 FF FF FF FF 07 00 05 00 0D 00 00 19 EF 01 FF FF FF FF 07 00 05 00 0E 00 00 1A EF 01 FF FF FF FF 07 00 05 00 0F **00** 00 1B (Fingerprint ID)

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 bytes	1 bytes	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Parameter 1	Parameter 2	Checksum
0xEF01	xxxx	0x07	0x0005	X	X	X	sum

Parameter 1: Step Process:

0x01: Collect image for the first time
0x02: Generate Feature for the first time
0x03: Collect image for the second time
0x04: Generate Feature for the second time
0x05: Collect image for the third time
0x06: Generate Feature for the third time
0x07: Collect image for the fourth time
0x08: Generate Feature for the fourth time
0x09: Collect image for the fifth time
0x0A: Generate Feature for the fifth time
0x0B: Collect image for the sixth time
0x0C: Generate Feature for the sixth time
0x0D: Repeat fingerprint check
0x0E: Merge feature
0x0F: Storage template

Parameter 2: fingerprint ID

Specific Acknowledge package format:

Header	Module address	Package identifier	Package length	Confirmation code	Step	Fingerprint ID	Check Sum	Note
2 Bytes	4 Bytes	1 Byte	2 Bytes	1 Byte	1 Byte	1 Byte	2 Bytes	
0xEF01	XXXX	0x07	0x0005	X	0x01	0x00	Sum	Collect image for the first time
0xEF01	XXXX	0x07	0x0005	X	0x02	0x00	Sum	Generate Feature for the first time
0xEF01	XXXX	0x07	0x0005	X	0x03	0x00	Sum	Collect image for the second time
0xEF01	XXXX	0x07	0x0005	X	0x04	0x00	Sum	Generate Feature for the second time
0xEF01	XXXX	0x07	0x0005	X	0x05	0x00	Sum	Collect image for the third time
0xEF01	XXXX	0x07	0x0005	X	0x06	0x00	Sum	Generate Feature for the third time
0xEF01	XXXX	0x07	0x0005	X	0x07	0x00	Sum	Collect image for the fourth time
0xEF01	XXXX	0x07	0x0005	X	0x08	0x00	Sum	Generate Feature for the fourth time
0xEF01	XXXX	0x07	0x0005	X	0x09	0x00	Sum	Collect image for the fifth time
0xEF01	XXXX	0x07	0x0005	X	0x0A	0x00	Sum	Generate Feature for the fifth time
0xEF01	XXXX	0x07	0x0005	X	0x0B	0x00	Sum	Collect image for the sixth time
0xEF01	XXXX	0x07	0x0005	X	0x0C	0x00	Sum	Generate Feature for the sixth time
0xEF01	XXXX	0x07	0x0005	X	0x0D	0x00	Sum	Repeat fingerprint check
0xEF01	XXXX	0x07	0x0005	X	0x0E	0x00	Sum	Merge feature
0xEF01	XXXX	0x07	0x0005	X	0x0F	modelID	Sum	Storage template

If the status of return key step is set to 0 during registration, only returned the the last acknowledge packet.

Confirmation code=0x00 set successfully

Confirmation code=0x01 set fails

Confirmation code=0x07 failed to generate a feature

Confirmation code=0x0a failed to merge templates

Confirmation code=0x0b the ID is out of range

Confirmation code=0x1f fingerprint library is full

Confirmation code=0x22 fingerprint template is empty

Confirmation code=0x26 times out

Confirmation code=0x27 fingerprint already exists

Automatic fingerprint verification AutoIdentify (0 x32)

When the automatic fingerprint verification command is used to search and verify a fingerprint, the system automatically collects a fingerprint image and generates features, and compares the image with the fingerprint template in the fingerprint database. If the comparison is successful, the system returns the template ID number and the comparison score. If the comparison fails, the system returns the corresponding error code.
When obtaining the fingerprint image, the fingerprint head will light up with a white breathing light. After the image collection is successful, the yellow light will light up, and the green light will light up after the comparison is successful. If there is a fingerprint image collection error or no fingerprint search, the red light will be on to prompt.
If the system does not detect the finger for more than 10 seconds after sending the command or collecting the fingerprint image again after reporting an error, it will automatically exit the command.

Input Parameter:
SafeGrade (1-5 level)
StartID
Num -Number of searches
Config1 Whether the module returns to the status in key steps
Config2 Number of fingerprint search error

Return Parameter:Confirmation code ModelID MarchScore
Instruction code: 0x32
Command (or instruction) package format:

2 bytes	4 bytes	1 byte	2 bytes	1 byte	1 byte	1 byte	1 byte	1 byte	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Security Level	Start Position	Number of searches	Whether return key step	Number of fingerprint search error	Check sum
0xEF01	xxxx	0x01	0x0008	0x32	Safe Grade	Start ID	num	Config 1	Config2	sum

Security level: 1-5
Starting position: 0-199
End position: 1-200
Return search steps: 0: not allowed 1: Allowed
Fingerprint search error times:
0-0xFF: 0: the operation of image collection and feature search is carried out all the time. If the same feature ID number is found in the fingerprint database, the operation will exit (0 means the cycle continues until the matching ID is found or the power is cut off).
1-0xFF: The generated features and fingerprint search were performed on the collected images. If the match is successful, the ID number and score of the match will be returned, and exit this instruction at the same time. If the match fails, repeat the previous operations for 1-0xff times. Exit after

corresponding error times. No matter what way, if the system does not detect the finger after sending the command or collecting the image again for more than 10 seconds, it will exit the command.

Example of AutoIdentify:

Send: EF 01 FF FF FF FF 01 00 08 32 03 00 C8 01 01 01 08

Return: EF 01 FF FF FF FF 07 00 08 00 01 00 00 00 00 00 10 EF 01 FF FF FF FF 07 00 08 00 02 00 00 00 00 00 11 EF 01 FF FF FF FF 07 00 08 00 03 **00 00** **00 3D** 00 4F (Fingerprint ID and score)

Acknowledge package format:

2 bytes	4 bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes	2 bytes	Note
Header	Module address	Package identifier	Package length	Confirmation code	Step	Position Number	Score	Check Sum	
0xEF01	xxxx	0x07	0x0008	X	1	00	00	Sum	Collect Image
0xEF01	xxxx	0x07	0x0008	X	2	00	00	Sum	Generate Feature
0xEF01	xxxx	0x07	0x0008	X	3	Model ID	Match score	Sum	Search

Confirmation code=0x00 set successfully

Confirmation code=0x01 set fails

Confirmation code=0x09 failed to search fingerprint

Confirmation code=0x0b the ID is out of range

Confirmation code=0x22 fingerprint template is empty

Confirmation code=0x24 fingerprint library is empty

Confirmation code=0x26 times out

Other instructions

To generate a random code GetRandomCode

Description: to command the Module to generate a random number and return it to upper computer;

Input Parameter: none

Return Parameter: Confirmation code (1 byte)+RandomCode

Instruction 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	0003H	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	0007H	xxH	RandomCode	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)
Instruction 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	0003H	16H	001AH

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	0003H	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. The upper computer sends the command packet, the module sends the acknowledge packet first, and then sends several data packet.
3.Packet Bytes N is determined by Packet Length. The value is 128 Bytes before delivery.
4: The instruction doesn't affect buffer contents.

Data package format:

2 bytes	4bytes	1 byte	2 bytes	N bytes	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.Also see R**eadNotepad;**
Input Parameter: NotePageNum, user content (or data content)
Return Parameter: Confirmation code (1 byte)
Instruction code: 18H
Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1byte	32 bytes	2 bytes

Header	Module address	Package identifier	Package length	Instruction code	Page number	Data content	Checksum
0xEF01	xxxx	01H	0x0024	18H	0x00-0x0F	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	0003H	xxH	sum

Note: Confirmation code=00H: write success;

Confirmation code=01H: error when receiving package;

Confirmation code=18H: error when write FLASH

To read note pad ReadNotepad

Description: to read the specified page's data content;Also see **WriteNotepad**.

Input Parameter: NotePageNum

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

Instruction code: 19H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Page number	Checksum
0xEF01	xxxx	01H	0004H	19H	0x00-0x0F	Sum

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	0x0023	xxH	User content	sum

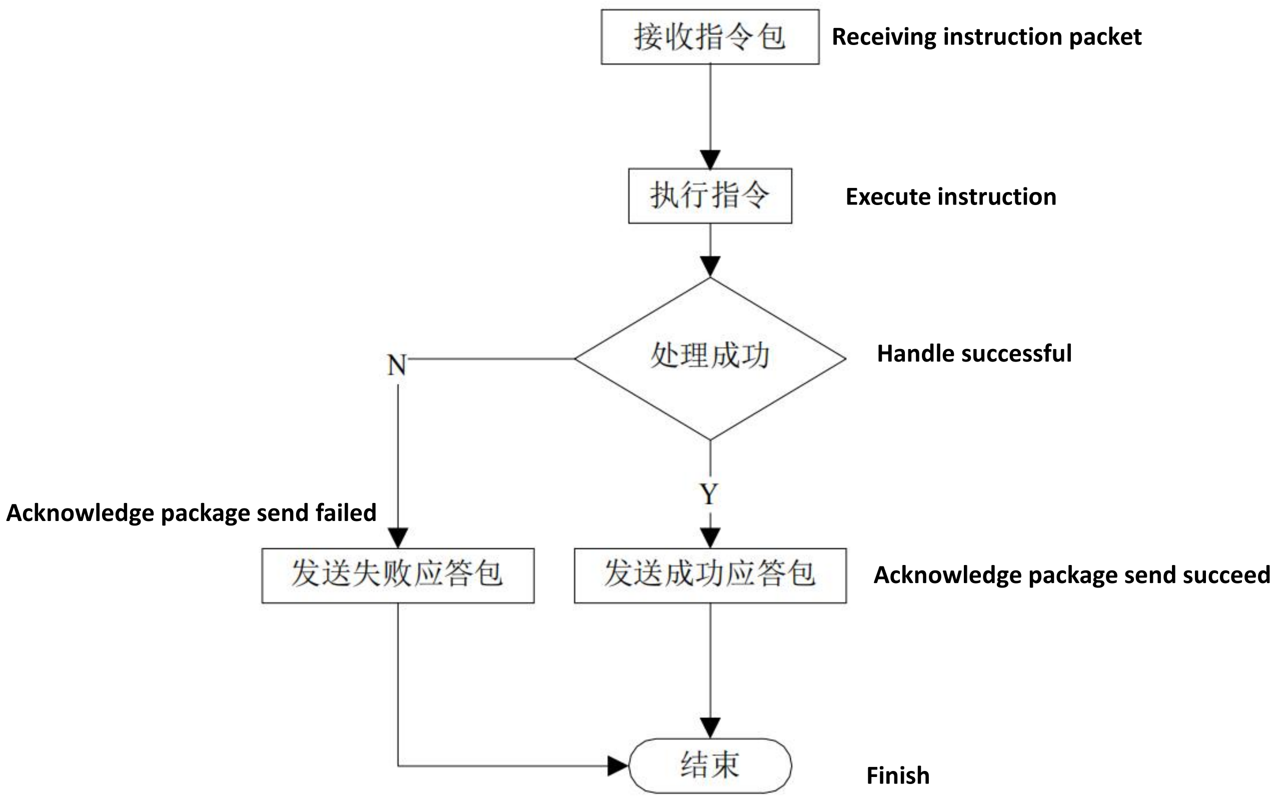
Note: Confirmation code=00H: read success;

Confirmation code=01H: error when receiving package;

VI Operation Process

6.1 Basic communication flow

6.1.1 Process of the UART command package



功能实现示例 1：UART命令包的处理过程

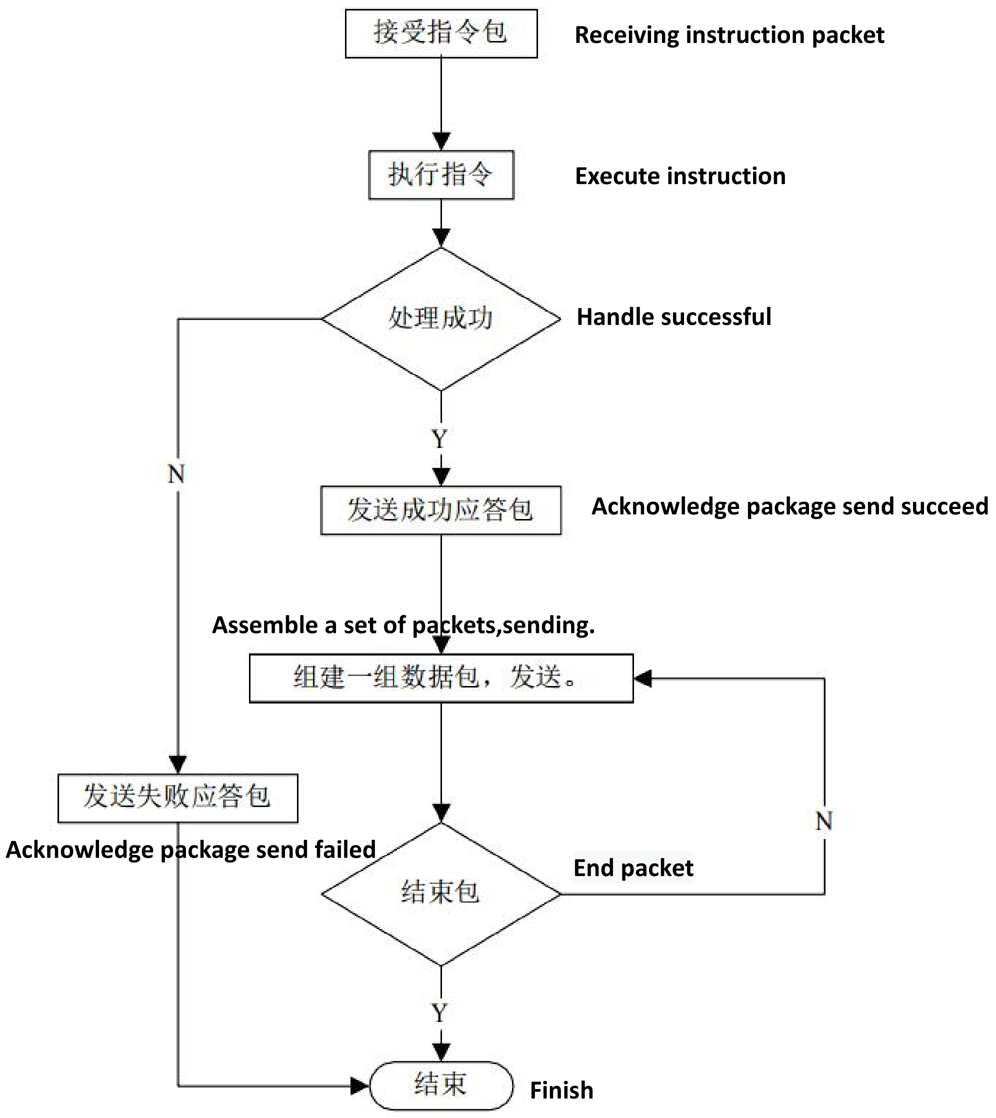
6.1.2 UART Packet Sending Process

Before transmitting data packets, the UART should be received the instruction packet for transmitting data packets first, makes preparations for transmission, then sends a successful response packet, and finally starts transmitting the data packets. Packet mainly includes: packet header, chip address, packet identity, packet length, data and checksum.

There are two types of packet identifiers: 02H and 08H. 02H: indicates the data packet and subsequent packets. 08H: indicates the last packet, that is, the end packet. Data length is pre-set, mainly divided into: 32, 64, 128, and 256 four types.

For example, if the length of the data to be transmitted is 1K bytes and the preset length of the data packet is 128 bytes, the 1K bytes of data must be divided into eight data packets. Each packet includes: 2 bytes header, 4 bytes chip address, 1 bytes packet identifier, 2 bytes packet length, 128 bytes data and 2 bytes check sum, each packet length is 139 bytes.

In addition, of the eight packets, the packet ID of the first seven packets is 02H and the packet ID of the last end data packet is 08H. Finally, note that if the end packet does not reach 139 bytes in length, it is transmitted at the actual length and is not otherwise expanded to 139 bytes.



功能实现示例 2：UART 数据包的发送过程

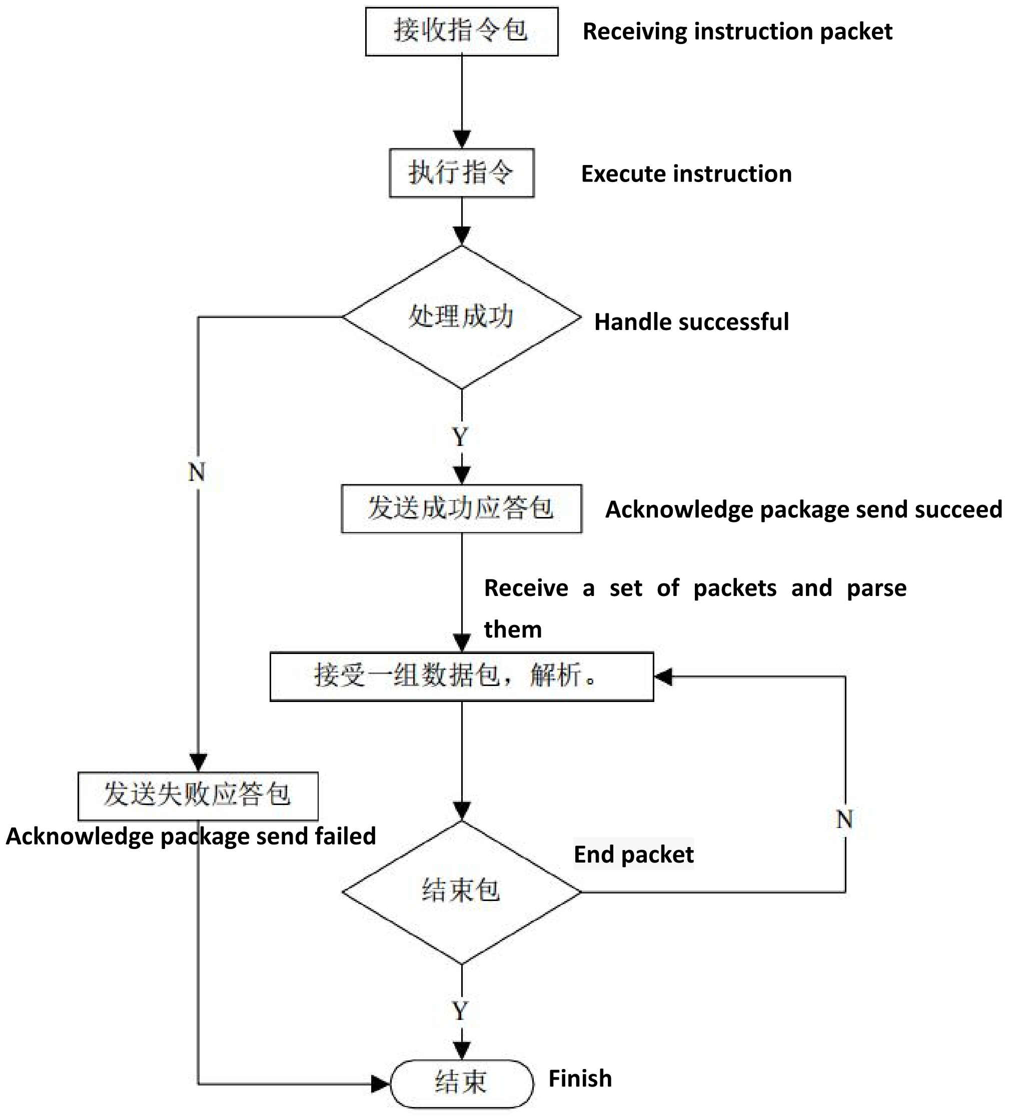
6.1.3 UART packet receiving process

Before transmitting data packets, the UART should be received the instruction packet for transmitting data packets first, makes preparations for transmission, then sends a successful response packet, and finally starts transmitting the data packets. Packet mainly includes: packet header, chip address, packet identity, packet length, data and checksum.

There are two types of packet identifiers: 02H and 08H. 02H: indicates the data packet and subsequent packets. 08H: indicates the last packet, that is, the end packet. Data length is pre-set, mainly divided into: 32, 64, 128, and 256 four types.

For example, if the length of the data to be transmitted is 1K bytes and the preset length of the data packet is 128 bytes, the 1K bytes of data must be divided into eight data packets. Each packet includes: 2 bytes header, 4 bytes chip address, 1 bytes packet identifier, 2 bytes packet length, 128 bytes data and 2 bytes check sum, each packet length is 139 bytes.

In addition, of the eight packets, the packet ID of the first seven packets is 02H and the packet ID of the last end data packet is 08H. Finally, note that if the end packet does not reach 139 bytes in length, it is transmitted at the actual length and is not otherwise expanded to 139 bytes.

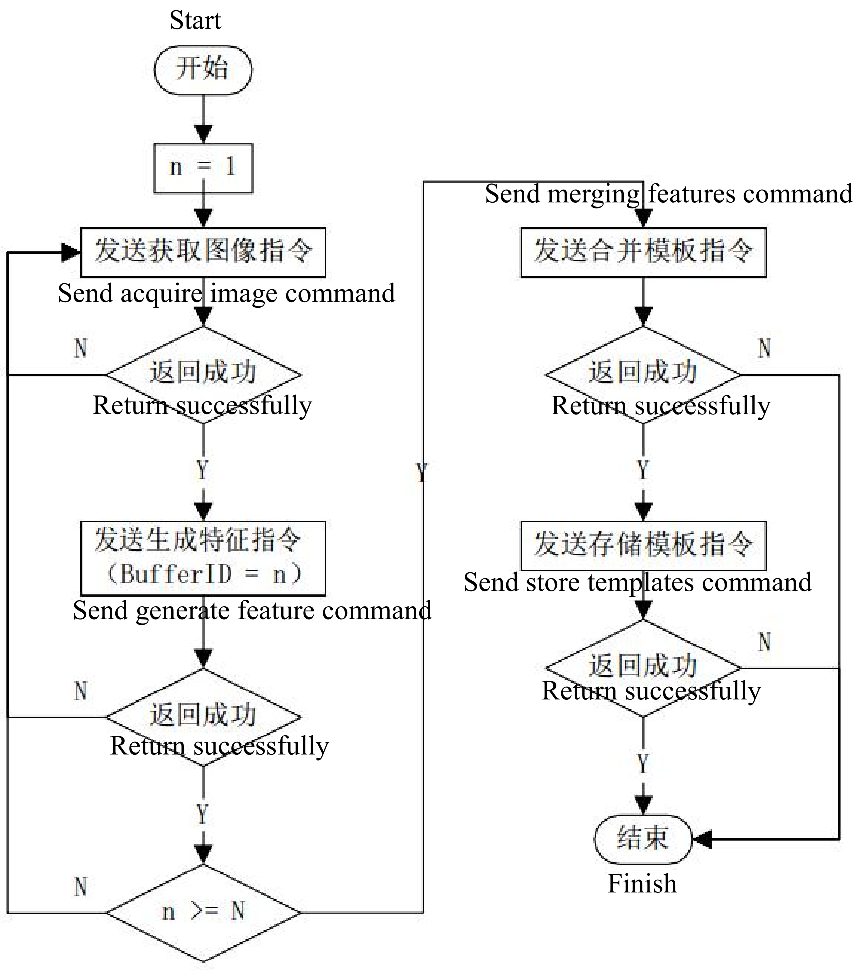


功能实现示例 3：UART 数据包的接收过程

6.2 General instruction communication flow

6.2.1 General instruction register fingerprint process

The fingerprint registration process mainly includes: obtaining images for registration, generating features, merging features and storing templates. Usually N = 2 times.



功能实现示例 4：通用指令注册流程

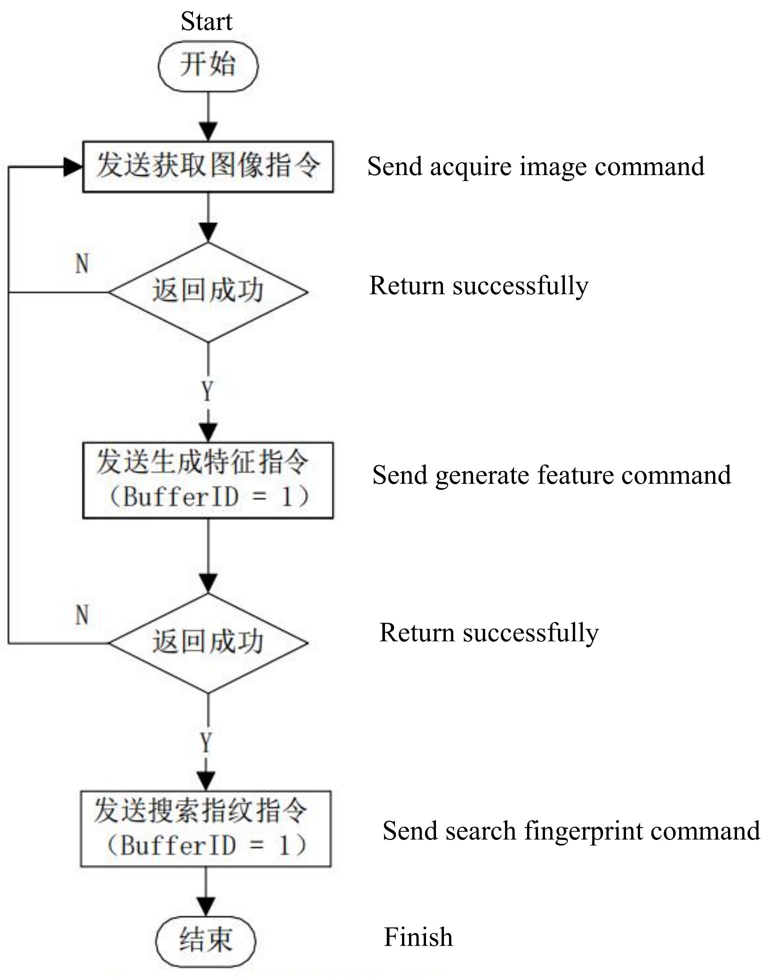
When the registration logic is set to 1, register fingerprint. If the current fingerprint is similar to the fingerprint that has been included before,the confirmation code in the response packet that generates the feature command does not show success, but returns 28H, indicating that there is a correlation between the current fingerprint feature and the previous feature. It should be noted that the mutual comparison correlation is limited to the fingerprints included in this registration process, and will not be compared with the fingerprints in the fingerprint library.

When the registration logic is set to 2, register fingerprint. If the current fingerprint is not similar to the fingerprint that has been included before,the confirmation code in the response packet that generates the feature command does not show success, but returns 08H, indicating that there is no correlation between the current fingerprint feature and the previous feature. It should be noted that the mutual comparison correlation is limited to the fingerprints included in this registration process, and will not be compared with the fingerprints in the fingerprint library.

Whether it returns 28H or 08H, the current fingerprint feature has been successfully extracted, you can take a new map and generate features without changing the BufferID, or you can skip the current BufferID and include the next round of fingerprints.

6.2.2 General instruction verity fingerprint process

The fingerprint verification process of general instructions mainly includes: obtaining images for verification, generating features and searching fingerprints. When sending generated features and searching for fingerprints, BufferID is set to 1 by default.

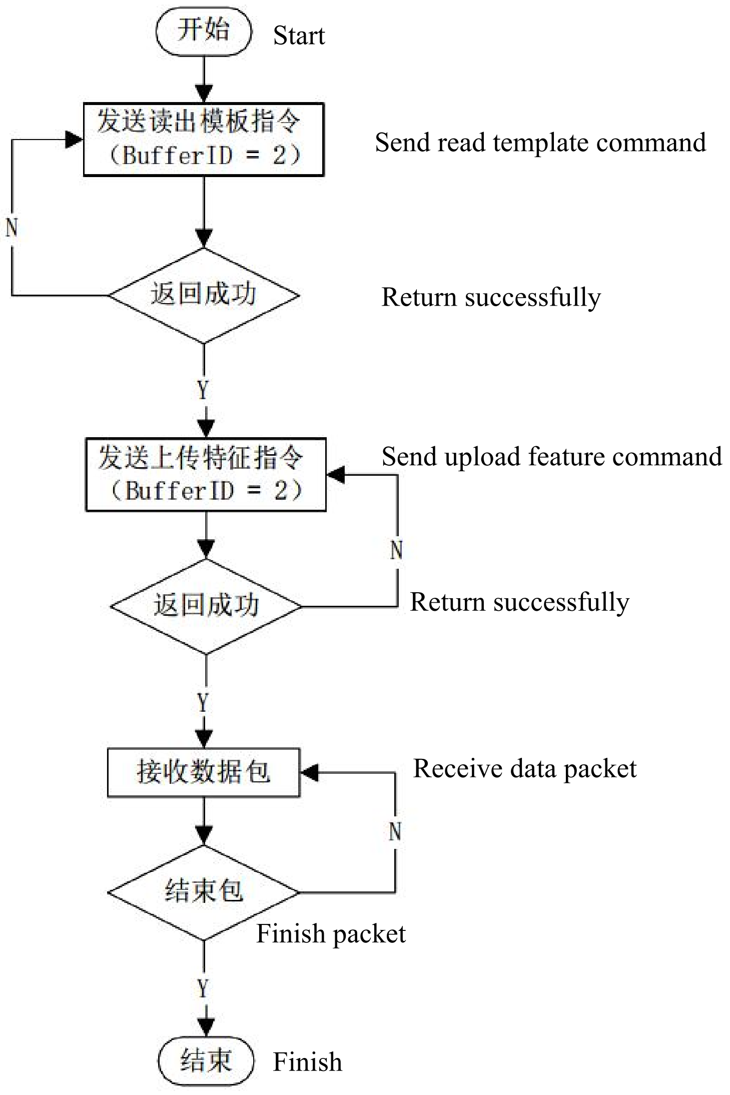

Start
开始
发送获取图像指令
Send acquire image command
N
返回成功
Return successfully
Y
发送生成特征指令 (BufferID = 1)
Send generate feature command
N
返回成功
Return successfully
Y
发送搜索指纹指令 (BufferID = 1)
Send search fingerprint command
结束
Finish

功能实现示例 5：通用指令验证流程

6.2.3 Read a specified template upload to Flash Fingerprint Database

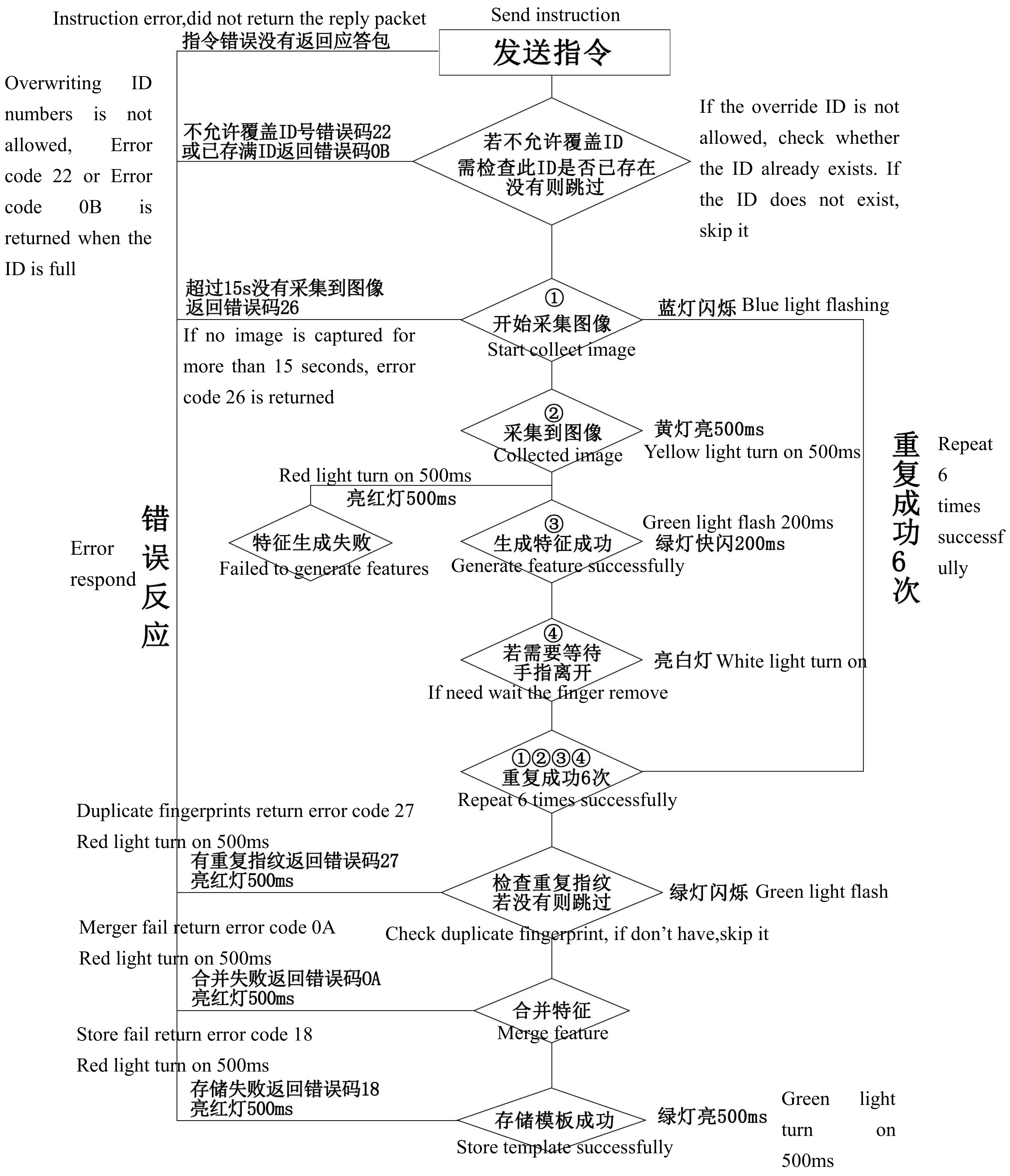
The whole process mainly includes: read template and upload templates.

BufferID is set to the default value 2 when reading template and uploading feature.



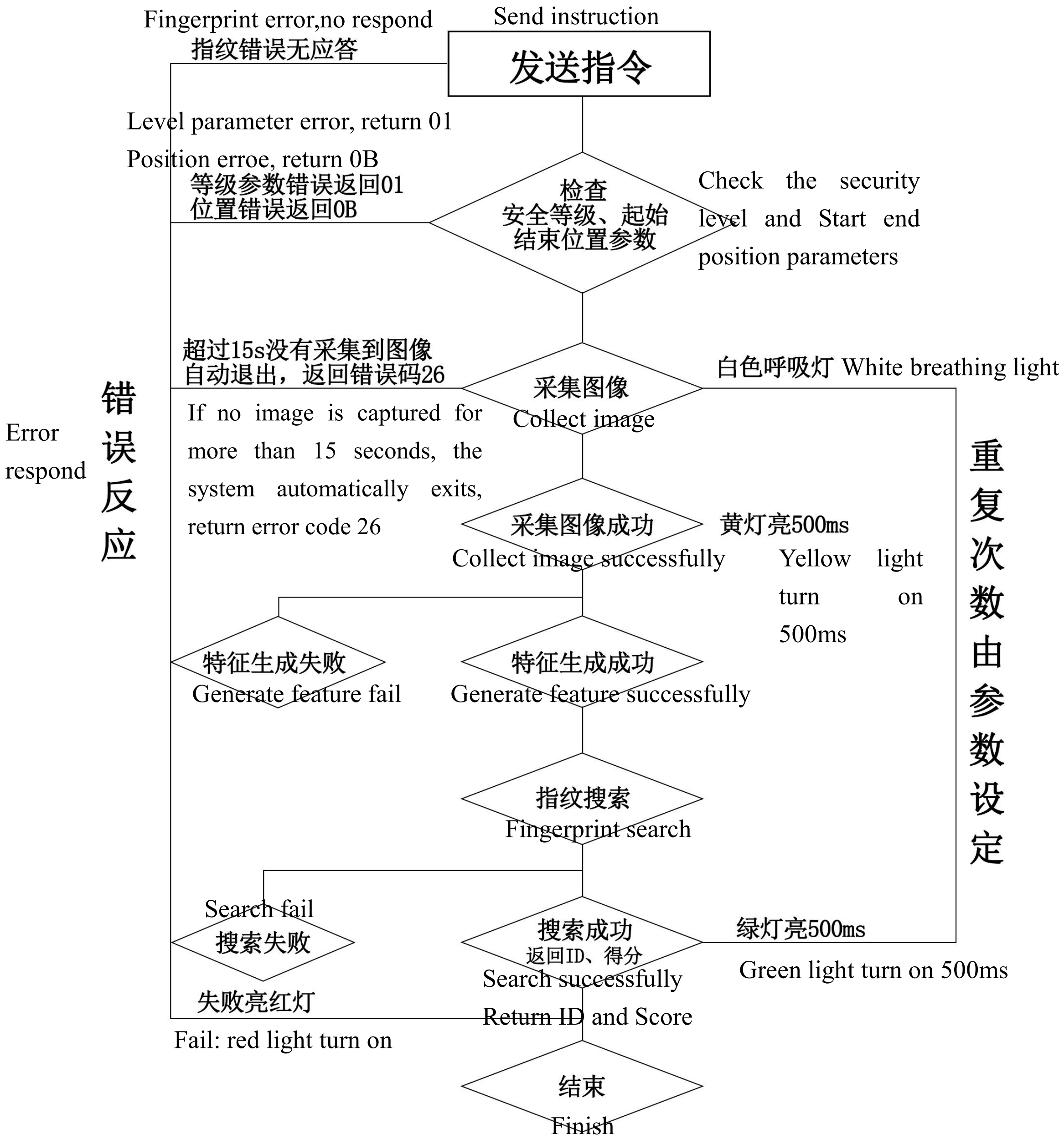
功能实现示例 7：从 flash 指纹库中读取一个指定的模板上传

6.3 Automatic Register Fingerprint



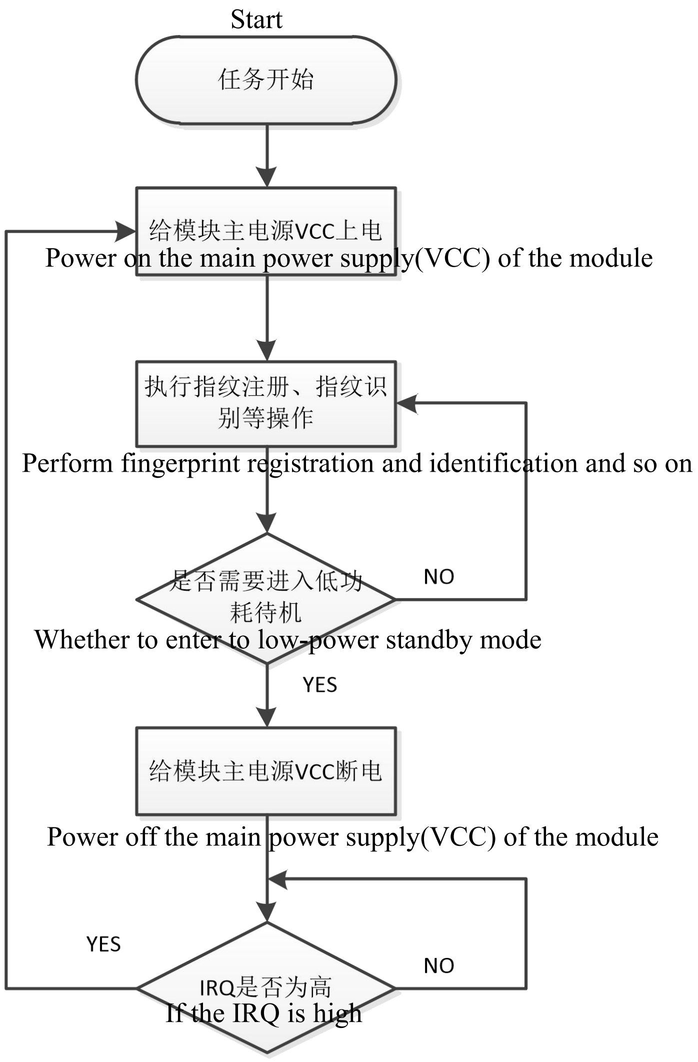
Instruction error,did not return the reply packet
指令错误没有返回应答包
Send instruction
发送指令
Overwriting ID numbers is not allowed, Error code 22 or Error code 0B is returned when the ID is full
不允许覆盖ID号错误码22
或已存满ID返回错误码0B
若不允许覆盖ID
需检查此ID是否已存在
没有则跳过
If the override ID is not allowed, check whether the ID already exists. If the ID does not exist, skip it
超过15s没有采集到图像
返回错误码26
If no image is captured for more than 15 seconds, error code 26 is returned
①
开始采集图像
Start collect image
蓝灯闪烁 Blue light flashing
②
采集到图像
Collected image
黄灯亮500ms
Yellow light turn on 500ms
Red light turn on 500ms
亮红灯500ms
特征生成失败
Failed to generate features
③
生成特征成功
Generate feature successfully
Green light flash 200ms
绿灯快闪200ms
Error respond
错误反应
重复成功6次
Repeat 6 times successfully
④
若需要等待
手指离开
If need wait the finger remove
亮白灯 White light turn on
①②③④
重复成功6次
Repeat 6 times successfully
Duplicate fingerprints return error code 27
Red light turn on 500ms
有重复指纹返回错误码27
亮红灯500ms
检查重复指纹
若没有则跳过
Check duplicate fingerprint, if don't have,skip it
绿灯闪烁 Green light flash
Merger fail return error code 0A
Red light turn on 500ms
合并失败返回错误码0A
亮红灯500ms
合并特征
Merge feature
Store fail return error code 18
Red light turn on 500ms
存储失败返回错误码18
亮红灯500ms
存储模板成功
Store template successfully
绿灯亮500ms
Green light turn on 500ms

6.4 Automatic Fingerprint Verification(Search)



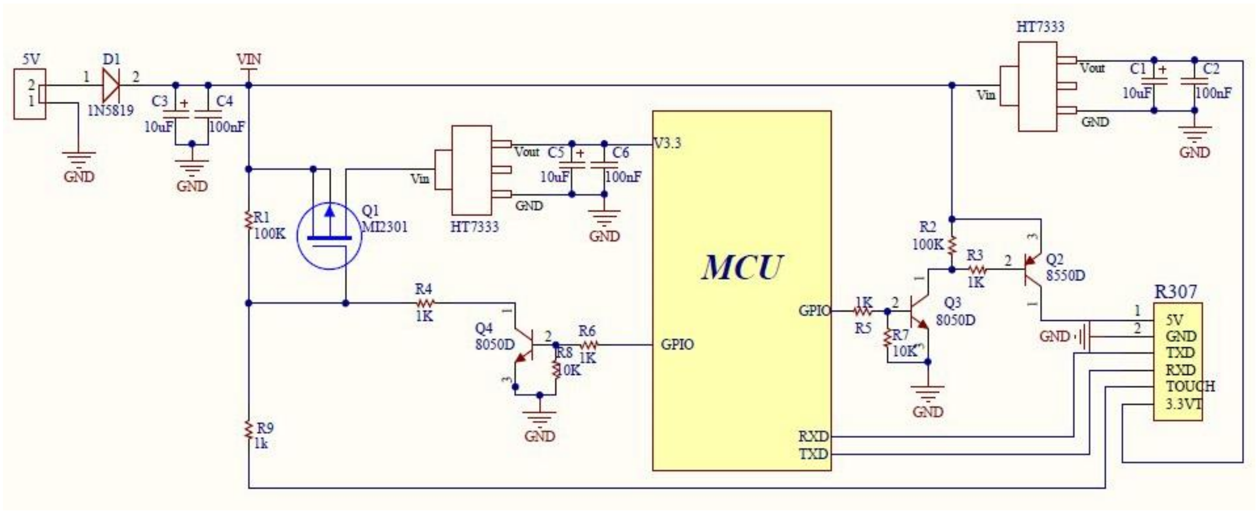
6.5 Low power standby

For low-power standby scenarios, the host can cut off the main power supply of the module (can not cut off the touch-sensitive power supply). Once the module detects a finger, it outputs a signal in the IRQ signal. Then the host can power on the module to perform fingerprint identification.


Start
任务开始
给模块主电源VCC上电
Power on the main power supply(VCC) of the module
执行指纹注册、指纹识别等操作
Perform fingerprint registration and identification and so on
是否需要进入低功耗待机
Whether to enter to low-power standby mode
NO
YES
给模块主电源VCC断电
Power off the main power supply(VCC) of the module
YES
NO
IRQ是否为高
If the IRQ is high

VII Reference Circuit

In low-power supply mode, the whole circuit is normally powered off. Use the finger detection function of the module to power on the whole machine. Please refer to the circuit form of **R307 (R307 is 5V power supply).**



5V
D1
1N5819
C3
10uF
C4
100nF
VIN
GND
R1
100K
Q1
MI2301
HT7333
Vin
Vout
C5
C6
10uF
100nF
V3.3
MCU
R4
1K
Q4
8050D
R6
1K
R8
10K
GPIO
R9
1k
RXD
TXD
GPIO
R5
1K
R7
10K
Q3
8050D
R2
100K
R3
1K
Q2
8550D
C1
C2
R307
5V
GND
TXD
RXD
TOUCH
3.3VT