THE USER'S MANUAL OF FINGERPRINT MODULE

M series

(V1.1.10)

September, 2020

contents

1. Summarize

M SERIES fingerprint module is a stable mass production product launched by our company.

M SERIES fingerprint module has a complete fingerprint processing capability, which can complete fingerprint input, image processing, fingerprint comparison, fingerprint feature storage and other functions without the involvement of upper computer.

Compared with similar fingerprint products, it has the following features:

✧ Higher integration, smaller size and lower power consumption.

✧ The module USES independently developed fingerprint identification algorithm and chip, and its performance indexes are better than similar products.

✧ Fingerprint recognition algorithm used by the module provides template splicing function to improve user experience.

✧ The module adopts the scheme of storing fingerprint features in the algorithm chip, which can prevent fingerprint features from being cracked violently and improve the product safety level.

✧ The product has the ability of anti-fog, anti-dust and anti-destruction, which effectively solves the influence of human body static electricity, improves the image quality of the product, and increases the durability of the product.

✧ Modular sensor surface can be customized color.

✧ The module has the function of touch and wake up, and can effectively identify the false fingers such as plastic fingers, silicone fingers, rubber fingers, finger dies and finger covers.

✧ The product has simple structure and modular design, which improves the stability and consistency of the product and facilitates mass production.

✧ Open interface code and command set, can achieve packaged secondary development, reduce the difficulty of customer development.

✧ Independent intellectual property technology can provide customers with efficient and flexible secondary development support; fully meet customer needs without

intellectual property disputes.

1.1 Technical supports

Our company has a mature technical team, most of the staff are professionals from the fingerprint industry, can provide good technical support and after-sales service for user development.

1.2 Applications

M SERIES fingerprint module is widely used and suitable for various fingerprint identification systems, such as:

✧ Fingerprint door locks, safes, gun boxes, finance and other security fields;

✧ Access control system, industrial control computer, POS machine, driving school training, attendance and other identity fields;

✧ Private clubs, management software, licensing and other management fields.

Customers can develop the corresponding fingerprint identification application system according to the technical data provided in this manual.

2. The main technical parameters

Please refer to the specifications of each module.

3. Hardware interface
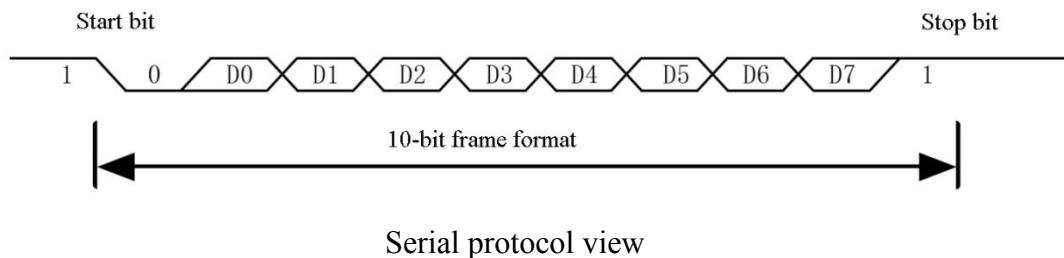
3.1 Module external interface definition

Please refer to the specifications of each module.

3.2 Serial protocol

With half-duplex asynchronous serial communication, the default baud rate is 115200bps.

The transmitted frame format is 10 bits, a 0-bit start bit, 8 data bits (lower bit first) and one stop bit, no parity bit.

Serial protocol view
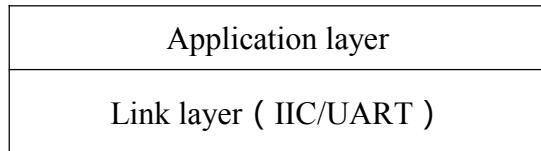
3.3 Power-on/reset delay

After the module is powered on, it takes about 200ms to perform initialization work. During this time, the module cannot respond to the host command.

4. Protocol

The communication protocol defines the rules for data exchange between the M series template and the host computer. The UART interface is adopted in the hardware, and the software implementation refers to the following communication protocol and instruction set.

This protocol is a two-tier structure: the application layer and the link layer. The application layer is mainly responsible for specific function definitions, and the link layer is used for data transmission. The 2 bytes or 4 bytes defined in the transport protocol use little endian, and the high order is after the first low order (for example, 0x12345678 actual transmission mode is 0x12 0x34 0x56 0x78, not 0x78 0x56 0x34 0x12).

Application layer
Link layer（IIC/UART）

4.1 Link layer description（UART）

The UART link layer is a half-duplex point-to-point mode. All commands must be initiated by the host computer and the fingerprint module responds.

UART data frame format:

Format	Frame header	Application layer data length	Frame header checksum	Application layer data
Length (bytes)	8	2	1	7+N

UART data frame format table

Description:

Frame header: The UART start header is defined. Whether the host computer sends data or the fingerprint module sends data, the frame header data is fixed to 0xF1 1F E2 2E B6 6B A8 8A;
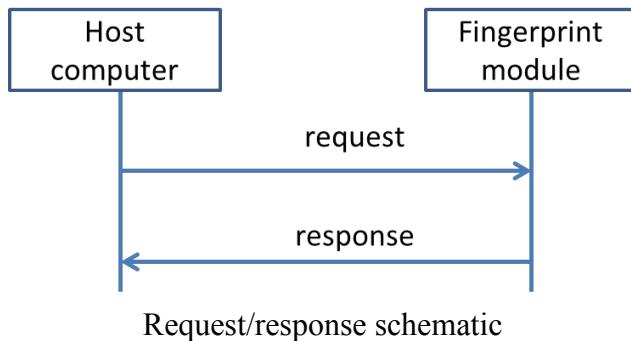
Application layer data length: describes the length of the actual data of the application

layer. This data length does not include the frame header, application layer data length, and frame header checksum.

Frame header checksum: It is the checksum of the frame header + application layer data length, which is used to check whether the data length is incorrect.

4.2 Application layer description

The application layer adopts the request/response communication protocol. All commands are initiated by the host computer and the fingerprint module responds, as shown in the following figure.

Request/response schematic

Considering the difficulty of development of the host computer, the application layer protocol uses a fixed frame length to communicate, use and verify the data to ensure data integrity. In practical applications, the data can be encrypted by modifying the protocol's check password, and will not be cracked by a third party. (Note: The verification password will take effect immediately after setting it. Please keep the password properly. After the password is lost, you need to return to the factory for initialization.)

According to the application layer protocol description, the request command and the response request are respectively described in two formats to describe the host computer data transmission and the fingerprint module information response.

Data transmission:

Format	Check password	Command	Data content	Checksum
Length (bytes)	4	2	N(0~n)	1

Data transmission form

Description:

The total length of the transmitted data packet is: (7+N) Bytes. The packet length is different according to different commands, but each instruction is a fixed length.

Check password: used to ensure data transmission security. The default password is 0x00000000. You can change this password through the application layer command.

Command:

Data content: The content actually sent.

Checksum: and checksum, in accordance with the BYTE operation. (See the appendix for specific checksum instructions)

Response request:

Format	Check password	Response command	Error code	Data content	Checksum
Length (bytes)	4	2	4	N(0~n)	1

Response request form

Description:

The total length of the received data packet is: (11 + N) Bytes. The packet length is different according to different commands, but each instruction is a fixed length.

Check password: used to ensure data transmission security. The default password is 0x00000000. You can change this password through the application layer command.

Response command: Corresponding to the command field sent by the host computer, indicating that the response has been received.

Error code: Returns the result of this command execution, 0x00000000 means no

error, other reference error information definition table.

Data content: Returns the result of the response.

Checksum: and checksum, in accordance with the BYTE operation. (See the appendix for specific checksum instructions)

5. Command set summary

Host					Fingerprint module			
Command type	Subcommand	CMD1	CMD2	Data (Len)	ACK1	ACK2	Error Code	Data (Len)
Fingerprint class	Fingerprint registration	0x01	0x11	1	0x01	0x11	Normal: 0x00000000 Error: return error code	0
Fingerprint class	Query registration results	0x01	0x12	0	0x01	0x12	Normal: 0x00000000 Error: return error code	3
Fingerprint class	Save fingerprint	0x01	0x13	2	0x01	0x13	Normal: 0x00000000 Error: return error code	0
Fingerprint class	Query save fingerprint results	0x01	0x14	0	0x01	0x14	Normal: 0x00000000 Error: return error code	0
Fingerprint class	Cancel fingerprint registration	0x01	0x15	0	0x01	0x15	Normal: 0x00000000 Error: return error code	0
Fingerprint class	Update storage feature values	0x01	0x16	2	0x01	0x16	Normal: 0x00000000 Error: return error code	0
Fingerprint class	View storage feature update results	0x01	0x17	0	0x01	0x17	Normal: 0x00000000 Error: return error code	0
Fingerprint class	Automatic registration	0x01	0x18	4	0x01	0x18	Normal: 0x00000000 Error: return error code	4
Fingerprint class	Fingerprint matching	0x01	0x21	0	0x01	0x21	Normal: 0x00000000 Error: return error code	0
Fingerprint	Query matching	0x01	0x22	0	0x01	0x22	Normal:	6

Host					Fingerprint module			
Command type	Subcommand	CMD1	CMD2	Data (Len)	ACK1	ACK2	Error Code	Data (Len)
class	results						0x00000000 Error: return error code	
Fingerprint class	Fingerprint matching (synchronization)	0x01	0x23	0	0x01	0x23	Normal: 0x00000000 Error: return error code	6
Fingerprint class	Fingerprint feature removal	0x01	0x31	N	0x01	0x31	Normal: 0x00000000 Error: return error code	0
Fingerprint class	Query fingerprint feature removal result	0x01	0x32	0	0x01	0x32	Normal: 0x00000000 Error: return error code	0
Fingerprint class	Query whether the fingerprint ID exists	0x01	0x33	2	0x01	0x33	Normal: 0x00000000 Error: return error code	3
Fingerprint class	Query fingerprint feature storage distribution	0x01	0x34	0	0x01	0x34	Normal: 0x00000000 Error: return error code	66
Fingerprint class	Query finger in position	0x01	0x35	0	0x01	0x35	Normal: 0x00000000 Error: return error code	1
Fingerprint class	Fingerprint feature clearing (synchronization)	0x01	0x36	N	0x01	0x36	Normal: 0x00000000 Error: return error code	0
Fingerprint class	Registration fingerprint confirmation	0x01	0x41	0	0x01	0x41	Normal: 0x00000000 Error: return error code	0
Fingerprint class	Query the registration fingerprint confirmation result	0x01	0x42	0	0x01	0x42	Normal: 0x00000000 Error: return error code	6
Fingerprint	Fingerprint feature	0x01	0x51	4	0x01	0x51	Normal:	0

	Host				Fingerprint module			
Command type	Subcommand	CMD1	CMD2	Data (Len)	ACK1	ACK2	Error Code	Data (Len)
class	information download						0x00000000 Error: return error code	
Fingerprint class	Fingerprint feature data download	0x01	0x52	N	0x01	0x52	Normal: 0x00000000 Error: return error code	0
Fingerprint class	Fingerprint feature information upload	0x01	0x53	2	0x01	0x53	Normal: 0x00000000 Error: return error code	2
Fingerprint class	Fingerprint feature data upload	0x01	0x54	2	0x01	0x53	Normal: 0x00000000 Error: return error code	N
System class	password setting	0x02	0x01	4	0x02	0x01	Normal: 0x00000000 Error: return error code	0
System class	Reset module	0x02	0x02	0	0x02	0x02	Normal: 0x00000000 Error: return error code	0
System class	Get the number of fingerprint templates	0x02	0x03	0	0x02	0x03	Normal: 0x00000000 Error: return error code	2
System class	Gain gain	0x02	0x09	0	0x02	0x09	Normal: 0x00000000 Error: return error code	3
System class	Get matching threshold	0x02	0x0B	0	0x02	0x0B	Normal: 0x00000000 Error: return error code	2
System class	Sleep mode	0x02	0x0C	1	0x02	0x0C	Normal: 0x00000000 Error: return error code	0
System class	Set the maximum	0x02	0x0D	1	0x02	0x0D	Normal:	0

Host					Fingerprint module			
Command type	Subcommand	CMD1	CMD2	Data (Len)	ACK1	ACK2	Error Code	Data (Len)
	number of registrations						0x00000000 Error: return error code	
System class	Set LED control information	0x02	0x0F	5	0x02	0x0F	Normal: 0x00000000 Error: return error code	0
System class	Get system strategy	0x02	0xFB	0	0x02	0xFB	Normal: 0x00000000 Error: return error code	4
System class	Set system policy	0x02	0xFC	4	0x02	0xFC	Normal: 0x00000000 Error: return error code	0
Maintenance class	Read ID	0x03	0x01	0	0x03	0x01	Normal: 0x00000000 Error: return error code	16
Maintenance class	Heartbeat	0x03	0x03	0	0x03	0x03	Normal: 0x00000000 Error: return error code	0
Maintenance class	Set baud rate	0x03	0x04	4	0x03	0x04	Normal: 0x00000000 Error: return error code	0
Maintenance class	Set the communication password	0x03	0x05	4	0x03	0x05	Normal: 0x00000000 Error: return error code	0

Command set summary table

Command word description:

The M SERIES fingerprint module instruction set is optimized according to the usage scenario. The customer can support the fingerprint recognition function of different occasions and applications by simply integrating in the secondary development.

Fingerprint class:

It mainly contains some functional definitions of customers in actual use: fingerprint registration, matching, fingerprint feature deletion. When the host computer has other fingerprint algorithms, the fingerprint image acquisition function can be used to capture the fingerprint image.

The verification password and checksum of all the following examples are based on the default verification password 0x00000000. When the password verification function is used to modify the verification password, the new verification password and checksum are used here.

5.1 Fingerprint registration

(1) Send packet format (Byte):

REG_IDX indicates the number of times a fingerprint is pressed for registration. It is sent from 0x01. The return result only describes that the command has been received by the fingerprint module, and does not indicate that the fingerprint registration is successful. Whether the fingerprint registration is successful or not needs to be judged by querying the registration result. When the fingerprint progress PROC of the query is greater than 100, the fingerprint registration is completed. This function is a fingerprint splicing function, which is consistent with the application of fingerprint registration of mobile phones. It is recommended that users need to move their fingers properly when used to obtain a better unlocking user experience. The detailed process is described in the following operational process description. Recommendation: REG_IDX starts from 0x01 and increases to 0x06 (up to 6 times). When the registration result is found after 6 times and the progress PROC is not greater than 100, you can continue to increase the REG_IDX value to the appropriate number. Note: When the REG_IDX reaches the maximum number and the progress PROC is not greater than 100, the fingerprint template can also be saved. However, the

fingerprint template is incomplete and the user experience is not good.

Check password				CMD1	CMD2	REG_IDX	Checksum
0x00	0x00	0x00	0x00	0x01	0x11	0x01	0xED

Sending package example:

Number of registrations	Frame header							Data (Len)		Frame header check	Check password				Command		REG_IDX	Checksum
1	F1	1F	E2	2E	B6	6B	A8	8A	00	08	85	00	00	00	00	01	11	01	ED
2	F1	1F	E2	2E	B6	6B	A8	8A	00	08	85	00	00	00	00	01	11	02	EC
3	F1	1F	E2	2E	B6	6B	A8	8A	00	08	85	00	00	00	00	01	11	03	EB
4	F1	1F	E2	2E	B6	6B	A8	8A	00	08	85	00	00	00	00	01	11	04	EA
5	F1	1F	E2	2E	B6	6B	A8	8A	00	08	85	00	00	00	00	01	11	05	E9
6	F1	1F	E2	2E	B6	6B	A8	8A	00	08	85	00	00	00	00	01	11	06	E8

Fingerprint registration sending packet format table

(2) Response packet format (Byte):

Check password				CMD1	CMD2	Completion code				Checksum
0x00	0x00	0x00	0x00	0x01	0x11	0x00	0x00	0x00	0x00	0xEE

Answering package example:

Success (6 times the same):

Frame header								Data (Len)		Frame header check	Check password				Command		error code				Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	11	00	00	00	00	EE

Failure (system busy error code):

Frame header								Data (Len)		Frame header check	Check password				Command		error code				Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	11	00	00	00	04	EA

5.2 Query registration results

(1) Send packet format (Byte):

Check password				CMD1	CMD2	Checksum
0x00	0x00	0x00	0x00	0x01	0x12	0xED

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	01	12	ED

Query registration result sending package format table

(2) Response packet format (Byte): The actual ID value is returned when the registration is not completed, and the registration progress PROC value is less than 100:

Check password（4Byte）				CMD1	CMD2	Completion code				ID_H	ID_L	PROC	Checksum
0x00	0x00	0x00	0x00	0x01	0x12	0x00	0x00	0x00	0x00	0x00	0x01	0x14	0xD8

Answering package example:

Success (ID1 PROC16):

Frame header								Data (Len)		Frame header check	Check password				Command		error code				ID_H	ID_L	PROC	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0E	7F	00	00	00	00	01	12	00	00	00	00	00	01	10	DC

Success (ID1 PROC32):

Frame header								Data (Len)		Frame header check	Check password				Command		error code				ID_H	ID_L	PROC	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0E	7F	00	00	00	00	01	12	00	00	00	00	00	01	20	CC

Query registration result registration incomplete response packet format table

Failed (no finger pressure detected, timeout exit, etc.):

Frame header								Data (Len)		Frame header check	Check password				Command		error code				ID_H	ID_L	PROC	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0E	7F	00	00	00	00	01	12	00	00	00	08	00	00	00	E5

When the registration is completed, the actual ID value (ID is the current unused minimum value) is returned, and the registration progress PROC value is 100:

Check password (4Byte)				CMD1	CMD2	Completion code				ID_H	ID_L	PROC	Checksum
0x00	0x00	0x00	0x00	0x01	0x12	0x00	0x00	0x00	0x00	0x00	0x01	0x64	0x88

Answering package example:

Success (ID1 PROC100):

Frame header								Data (Len)		Frame header check	Check password				Command		error code				ID_H	ID_L	PROC	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0E	7F	00	00	00	00	01	12	00	00	00	00	00	01	64	88

Success (ID2 PROC100):

Frame header								Data (Len)		Frame header check	Check password				Command		error code				ID_H	ID_L	PROC	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0E	7F	00	00	00	00	01	12	00	00	00	00	00	02	64	87

Query registration result registration completion response packet format table

Get registration result error code			
No.	Error code	Description	Suggested treatment
1	0x00000000	normal	This registration is completed, you can check the registration progress.
2	0x00000003	Illegal command field	Check the registration start command and resend the registration.
3	0x00000004	The system is busy	The system is executing other commands, please delay the query and query the results.
4	0x00000005	If there is no request to send the command, the result will be queried.	The registration command needs to be resent and the number of times starts once.
5	0x00000006	System software reported error	It is possible that the feature data is too large

			during the splicing process and re-register this time.
6	0x00000007	Hardware error	Maybe the SPI can't communicate. Please try again. If it still doesn't work, please contact us.
7	0x00000008	No finger press detected, timeout	The current timeout period is 10S. No finger press is detected within 10s, and this time is re-registered.
8	0x00000009	Fingerprint extraction error, possible cause: poor image quality, etc.	Prompt fingerprint extraction error may result in poor image quality. Re-register this time
9	0x0000000B	Full storage space	The prompt storage space is full. Delete unwanted fingerprints and re-register
10	0x0000000C	Storage write failed	The prompt storage write failed. Target the issue and re-register
11	0x0000000D	Storage read failure	The prompt storage read failed. Target the issue and re-register
12	0x0000000E	The quality of the captured fingerprint image is not good	The image quality is poor. Fingers may be pressed too heavy, too light, or too moist, etc., and then re-register this time
13	0x00000010	Small area of drawing	Prompt the area of the drawing is small. Fingers may be pressed too lightly or too moist, etc., and then register this time again.
14	0x00000011	Excessive range of movement	The finger movement range is too large. Re-register this time
15	0x00000012	The range of movement is too small	The finger movement range is too small. Re-register this time
16	0x00000015	Forced exit	Please resend the start registration command
17	0x000000FF	Other errors	Normally, it will not appear. If it can appear, please try to re-register. Please contact us if it appears multiple times.

Get registration result error code table

5.3 Saving the fingerprint template

(1) Send packet format (Byte): The saved ID can be specified

Check password				CMD1	CMD2	ID_H	ID_L	Checksum
0x00	0x00	0x00	0x00	0x01	0x13	0x00	0x01	0xEB

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command	ID_H	ID_L	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	09	84	00	00	00	00	01 13	00	01	EB

Save fingerprint template to send package format table

(2) Response packet format (Byte):

Check password				CMD1	CMD2	Completion code				Checksum
0x00	0x00	0x00	0x00	0x01	0x13	0x00	0x00	0x00	0x00	0xEC

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		error code				Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	13	00	00	00	00	EC

Save fingerprint template response packet format table

5.4 Querying fingerprint save results

(1) Send packet format (Byte)

Check password				CMD1	CMD2	Checksum
0x00	0x00	0x00	0x00	0x01	0x14	0xEB

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	01 14	EB

Query fingerprint save result sending package format table

(2) Response packet format (Byte): The ID returned by the save is the saved ID, and the error code ID of 0x0F is the duplicate ID (the duplicate ID, the high digit is after, and the low digit is first).

Check password				CMD1	CMD2	Completion code				ID_H	ID_L	Checksum
0x00	0x00	0x00	0x00	0x01	0x14	0x00	0x00	0x00	0x00	0x00	0x01	0xEB

Answering package example:

Success:

Frame header							Data (Len)		Frame header check	Check password				Command		error code				ID_H	ID_L	Checksum
F1	1F	E2	2E	B6	6B	A8	00	0B	82	00	00	00	00	01	14	00	00	00	00	00	01	EA

Failed (fingerprint duplicate ID is duplicate ID):

Frame header							Data (Len)		Frame header check	Check password				Command		error code				ID_H	ID_L	Checksum
F1	1F	E2	2E	B6	6B	A8	00	0B	82	00	00	00	00	01	14	00	00	00	0F	00	00	DC

Query fingerprint save result response packet format table

Get the fingerprint save result error code			
No.	Error code	Description	Suggested treatment
1	0x00000000	Normal	Successfully saved
2	0x00000004	The system is busy	The system is executing other commands, please delay the query and query the results.
3	0x00000005	If there is no request to send the command, the result will be queried.	Send the save fingerprint command first and then query the save result, then save again, and then query the save result.
4	0x00000006	System software reported error	Prompt system software to report an error. Re-save
5	0x00000007	Hardware error	The ID that may be saved is greater than the maximum that can be stored. Re-save
6	0x00000009	Fingerprint extraction error, possible cause: poor image quality, etc.	An error occurred in the fingerprint extraction. Re-register and save
7	0x0000000C	Storage write failed	The prompt storage write failed. Target the issue and save again
8	0x0000000D	Storage read failure	The prompt storage read failed. Locate the problem and re-query the saved results
9	0x0000000B	Full storage space	The prompt storage space is full. Delete unwanted fingerprints and save again
10	0x0000000E	The quality of the captured fingerprint image is not good	The fingerprint image collected by the prompt is of poor quality. Re-register and save
11	0x0000000F	Fingerprint template repeat	The fingerprint template is repeated, and the fingerprint has been entered.
12	0x000000FF	Other errors	May force the registration to stop, save again

5.5 Cancel fingerprint registration

(1) Send packet format (Byte):

Check password				CMD1	CMD2	Checksum
0x00	0x00	0x00	0x00	0x01	0x15	0xEA

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	01	15	EA

Cancel fingerprint registration send packet format table

(2) Response packet format (Byte):

Check password				CMD1	CMD2	Completion code				Checksum
0x00	0x00	0x00	0x00	0x01	0x15	0x00	0x00	0x00	0x00	0xEA

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		error code				Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	15	00	00	00	00	EA

Cancel the fingerprint registration response packet format table

5.6 Updating storage feature values

(1) Send packet format (Byte):

ID is the ID to be updated. Self-learning needs to be used with this command. After each match is successful, this command is sent to update the feature value.

Check password				CMD1	CMD2	ID_H	ID_L	Checksum
0x00	0x00	0x00	0x00	0x01	0x16	0x00	0x00	0xE9

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		ID_H	ID_L	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	09	84	00	00	00	00	01	16	00	00	E9

Update storage feature value send packet format table

(2) Response packet format (Byte):

Check password				CMD1	CMD2	Completion code				Checksum
0x00	0x00	0x00	0x00	0x01	0x16	0x00	0x00	0x00	0x00	0xE9

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		error code				Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	16	00	00	00	00	E9

5.7 Viewing feature update results

(1) Send packet format (Byte):

Check password				CMD1	CMD2	Checksum
0x00	0x00	0x00	0x00	0x01	0x17	0xE8

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	01	17	E8

Query feature update result sending packet format table

(2) Response packet format (Byte):

Check password				CMD1	CMD2	Completion code				Checksum
0x00	0x00	0x00	0x00	0x01	0x17	0x00	0x00	0x00	0x00	0xE8

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		error code				Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	17	00	00	00	00	E8

Query feature update result response packet format table

5.8 Automatic registration

(1) Send packet format (Byte):

The operation, extraction, splicing, saving, etc. can be automatically completed by the total number of incoming presses and the fingerprint ID number; the timeout period for returning data by the command is 8s; waiting for the finger mark is used for After registering the press, do you need to wait for the finger to lift? Set to 1 to indicate that you need to wait for the finger to lift and press again to press the next registration. If set to 0, you do not need to detect the finger to lift up, and always register to progress 100. Up to now; the number of presses can be set to 1~6 times; the user can specify the ID number of the current registered fingerprint by the incoming ID value (you can specify the ID number used in the current system), if the ID is 0xFFFF, it will be automatically assigned by the system. Fingerprint ID.

Note: This feature is not supported on all models.

Check password				Waiting for finger	Number of presses	CMD1	CMD2	Checksum
0x00	0x00	0x00	0x00	0x01	0x18	0x00	0x03	0xFF

Frame header								Data (Len)		Frame header check	Check password				Command		Waiting for finger	Number of presses	ID_H	ID_L	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	18	00	03	FF	FF	E6

(2) Response packet format (Byte):

The registration successfully returns the fingerprint ID. The response packet is returned multiple times, and each time the fingerprint is collected, the result is returned. When the finger detection flag is 1, the result is returned first and then the finger is raised, and the result is returned after saving. The number of times in the return packet parameter refers to the current number of acquisitions. When the number of times is 0xFF, the response of the fingerprint is saved. When the sending packet ID is 0xFFFF, the response packet ID is the smallest ID that can be used,

otherwise it is the ID specified by the user; For registration progress, the maximum is 100.

Check password				CMD1	CMD2	Completion code				frequency	ID_H	ID_L	schedule	Checksum
0x00	0x00	0x00	0x00	0x01	0x18	0x00	0x00	0x00	0x00	0x01	0x00	0x00	0x21	0xC5

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		error code				frequency	ID_H	ID_L	schedule	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0A	83	00	00	00	00	01	18	00	00	00	00	01	00	00	21	C5
F1	1F	E2	2E	B6	6B	A8	8A	00	0A	83	00	00	00	00	01	18	00	00	00	00	02	00	00	42	A3
F1	1F	E2	2E	B6	6B	A8	8A	00	0A	83	00	00	00	00	01	18	00	00	00	00	03	00	00	64	80
F1	1F	E2	2E	B6	6B	A8	8A	00	0A	83	00	00	00	00	01	18	00	00	00	00	FF	00	00	64	84

5.9 Fingerprint matching

(1) Send packet format (Byte):

Check password				CMD1	CMD2	Checksum
0x00	0x00	0x00	0x00	0x01	0x21	0xDE

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	01	21	DE

(2) Response packet format (Byte):

Check password				CMD1	CMD2	Completion code				Checksum
0x00	0x00	0x00	0x00	0x01	0x21	0x00	0x00	0x00	0x00	0xDE

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		error code				Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	21	00	00	00	00	DE

Fingerprint matching response packet format table

5.10 Querying matching results

(1) Send packet format (Byte):

Check password				CMD1	CMD2	Checksum
0x00	0x00	0x00	0x00	0x01	0x22	0xDD

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	01	22	DD

Query matching result sending packet format table

(2) The response packet format (Byte), which contains the matching result, the matching score, and the matching fingerprint id number:

Check password（4Byte）				CMD1	CMD2	Completion code（4Byte）			
0x00	0x00	0x00	0x00	0x01	0x22	0x00	0x00	0x00	0x00
Match result		Match score		Match fingerprint id number		Checksum			
0x00	0x01	0x27	0x0F	0x00	0x03	0xA3			

Answering package example:

Success:

Frame header								Data (Len)		Frame header check	Check password				Command		error code			
F1	1F	E2	2E	B6	6B	A8	8A	00	11	7C	00	00	00	00	01	22	00	00	00	00
Match result		Match score		Match fingerprint id number		Checksum														
00	01	27	0F	00	03	A3														

Failure (error in fingerprint matching, possible cause: poor image quality, etc.):

Frame header								Data (Len)		Frame header check	Check password				Command		error code			
F1	1F	E2	2E	B6	6B	A8	8A	00	11	7C	00	00	00	00	01	22	00	00	00	0A

Match result		Match score		Match fingerprint id number		Checksum
00	00	00	00	00	00	D3

Failure (system busy and other error conditions):

Frame header								Data (Len)		Frame header check	Check password				Command		error code				Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	22	00	00	00	04	D9

Query match result response packet format table

Query the error code of the matching result			
No.	error code	Description	Suggested treatment
1	0x00000000	normal	Successful match to see match ID and match score
2	0x00000004	The system is busy	The system is executing other commands, please delay the query and query the results.
3	0x00000005	If there is no request to send the command, the result will be queried.	Send the matching command first and then query the matching result.
4	0x00000006	System software reported error	Prompt system software to report an error. Rematch
5	0x00000007	Hardware error	Prompt for hardware error. Rematch
6	0x00000008	No finger press detected, timeout	Prompt the user not to press the finger. Rematch
7	0x00000009	Fingerprint extraction error, possible cause: poor image quality, etc.	Prompt fingerprint extraction error. Maybe the image quality is not good, then re-match
8	0x0000000A	Fingerprint template library is empty	Please register your fingerprint and match
9	0x0000000E	The quality of the captured fingerprint image is not good	The fingerprint image collected by the prompt is of poor quality. Maybe the finger is pressed too heavy, too light, or too moist, etc., to match
10	0x00000010	Small area of drawing	Prompt the area of the drawing is small. Finger pressure is too light or the pressure is not correct, rematch
11	0x000000FF	Other errors	Rematch

Query match result error code table

5.11 Fingerprint matching (synchronization)

(1) Send packet format (Byte): Not all modules support this feature.

Check password				CMD1	CMD2	Checksum
0x00	0x00	0x00	0x00	0x01	0x23	0xDC

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	01	23	DC

Query matching result sending packet format table

(2) Response packet format (Byte):

Check password（4Byte）				CMD1	CMD2	Completion code（4Byte）			
0x00	0x00	0x00	0x00	0x01	0x23	0x00	0x00	0x00	0x00
Match result		Match score		Match fingerprint id number		Checksum			
0x00	0x01	0x27	0x0F	0x00	0x03	0xA2			

Answering package example:

Success:

Frame header								Data (Len)		Frame header check	Check password				Command		error code			
F1	1F	E2	2E	B6	6B	A8	8A	00	11	7C	00	00	00	00	01	23	00	00	00	00
Match result		Match score		Match id number			Checksum			
00	01	27	0F	00		03	A2			

Failure (error in fingerprint matching, possible cause: poor image quality, etc.):

Frame header								Data (Len)		Frame header check	Check password				Command		error code			
F1	1F	E2	2E	B6	6B	A8	8A	00	11	7C	00	00	00	00	01	23	00	00	00	0A
Match result		Match score		Match id number			Checksum			
00	00	00	00	00		00	D2			

5.12 Fingerprint feature removal

(1) Send packet format (Byte), CL_FLAG is the mode flag for clearing fingerprint features, 0x01 means to clear all fingerprints, 0x02 means to clear multiple fingerprints, 0x03 means to clear one fingerprint, and default 0x00 means to clear individual fingerprints. ID_H, ID_L is the fingerprint ID to be deleted (the ID value is assigned starting from 0):

Clear all or a single fingerprint feature to send the packet format:

Check password				CMD1	CMD2	CL_FLAG	ID_H	ID_L	Checksum
0x00	0x00	0x00	0x00	0x01	0x31	0x00	0x00	0x01	0xCD

Send package example:

Fingerprint feature clear (all):

Frame header								Data (Len)		Frame header check	Check password				Command		CL_FLAG	ID_H	ID_L	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0A	83	00	00	00	00	01	31	01	00	01	CC

Fingerprint feature removal (single):

Frame header								Data (Len)		Frame header check	Check password				Command		CL_FLAG	ID_H	ID_L	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0A	83	00	00	00	00	01	31	00	00	01	CD

Clear multiple fingerprint feature sending packet formats:

COUNT is the number of fingerprints to be deleted, and IDn is the fingerprint feature ID value to be deleted (the ID value is assigned from 0).

Check password				CMD1	CMD2	CL_FLAG	COUNT_H	COUNT_L
0x00	0x00	0x00	0x00	0x01	0x31	0x02	0xXX	0xXX
ID1_H	ID1_L	ID2_H	ID2_L	„ „		IDn_H	IDn_L	Checksum
0x00	0x01	0x00	0x02	„ „		0x00	0xXX	0xXX

Send package example:

Fingerprint feature clearing (multiple):

Frame header								Data (Len)		Frame header check	Check password				Command	
F1	1F	E2	2E	B6	6B	A8	8A	00	0E	7F	00	00	00	00	01	31

CL_FLAG	COUNT_H	COUNT_L	ID1_H	ID1_L	ID2_H	ID2_L	Checksum
02	00	02	00	01	00	02	C7

The block clear command CL_FLAG is 0x03, and you can enter the first ID and the last ID to be deleted.

Check password（4Byte）				CMD1	CMD2	CL_FLAG	FIRST ID_H	FIRST ID_L	LAST_ ID_H	LAST_ ID_H	Checksum
0x00	0x00	0x00	0x00	0x01	0x31	0x03	00	02	0x00	0x06	0xC3

Sending package example:

Fingerprint feature clearing (block deletion):

Frame header								Data (Len)		Frame header check	Check password				Command	
F1	1F	E2	2E	B6	6B	A8	8A	00	0C	81	00	00	00	00	01	31

CL_FLAG	FIRST ID_H	FIRST ID_L	LAST_ ID_H	LAST_ ID_H	Checksum		
03	00	02	00	06	C3		

Fingerprint feature clearing send packet format table

(2) Response packet format (Byte):

Check password(4Byte)				CMD1	CMD2	Completion code (4Byte)				Checksum
0x00	0x00	0x00	0x00	0x01	0x31	0x00	0x00	0x00	0x00	0xCE

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		error code				Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	31	00	00	00	00	CE

Fingerprint feature clear response packet format table

5.13 Querying fingerprint feature removal results

(1) Send packet format (Byte), no data field:

Check password（4Byte）				CMD1	CMD2	Checksum
0x00	0x00	0x00	0x00	0x01	0x32	0xCD

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	01	32	CD

Query fingerprint feature clear result sending packet format table

(2) Response packet format (Byte):

Check password(4Byte)				CMD1	CMD2	Completion code (4Byte)				Checksum
0x00	0x00	0x00	0x00	0x01	0x32	0x00	0x00	0x00	0x00	0xCD

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		error code				Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	32	00	00	00	00	CD

Query fingerprint feature clear result response packet format table

Get the deletion result error code			
No.	Error code	Description	Suggested treatment
1	0x00000000	normal	successfully deleted
2	0x00000001	Unrecognized command	Prompt unrecognized commands. Check if the command is sent correctly and resend again.
3	0x00000002	The command field length is illegal.	The command command field length is illegal. Check whether the length of the command field is correct and resend it again.
4	0x00000003	Illegal command field	The command command field is illegal. Check if the command field is correct and

5	0x00000004	The system is busy	resend it again.
			The system is executing other commands, please delay the query and query the results.

5.14 Check if fingerprint ID exists

(1) Send packet format (Byte), ID_H, ID_L is the fingerprint ID to be queried:

Check password（4Byte）				CMD1	CMD2	ID_H	ID_L	Checksum
0x00	0x00	0x00	0x00	0x01	0x33	0x00	0x01	0xCB

Send package example:

Frame header								Data (Len)	Frame header check	Check password				Command		ID_H	ID_L	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00 09	84	00	00	00	00	01	33	00	01	CB

Query whether the fingerprint ID exists in the format table of the sending packet

(2) The response packet format (Byte), the successful completion code is 0x00000000, and the others fail the query. If the fingerprint of the query exists, return STATE to 1, otherwise STATE is 0; the data returns ID_H, ID_L is the fingerprint ID of this operation query:

Check password（4Byte）				CMD1	CMD2	Complete code (4Byte)				STATE	ID_H	ID_L	Checksum
0x00	0x00	0x00	0x00	0x01	0x33	0x00	0x00	0x00	0x00	0x01	0x00	0x01	0xCA

Sample reply package:

Exist

Frame header								Data (Len)	Frame header check	Check password				Command		error code				STATE	ID_H	ID_L	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00 0E	7F	00	00	00	00	01	33	00	00	00	00	01	00	01	CA

Not exist

Frame header								Data (Len)	Frame header check	Check password				Command		error code				STATE	ID_H	ID_L	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00 0E	7F	00	00	00	00	01	33	00	00	00	00	00	00	01	CB

5.15 Query the storage distribution of fingerprint features

(1) Send packet format (Byte):

Check password（4Byte）				CMD1	CMD2	Checksum
0x00	0x00	0x00	0x00	0x01	0x34	0xCB

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	01	34	CB

(2) The response packet format (Byte), the successful completion code is 0x00000000, and the others fail the query. NUM_H, NUM_L is the total number of fingerprints stored, and INFO_0 ~ INFO_63 is the fingerprint storage state. The 0~7 fingerprint status corresponds to 0~7bit of INFO_0, the 8~15th fingerprint status corresponds to 0~7bit of INFO_1, and so on. If the fingerprint exists, the corresponding bit is set to 1, otherwise it is 0.

Check password				CMD1	CMD2	Complete code (4Byte)				NUM_H	NUM_L
0x00	0x00	0x00	0x00	0x01	0x34	0x00	0x00	0x00	0x00	0x20	0x00
INFO_0	INFO_1	INFO_2				......				INFO_63	Checksum
0xFF	0x00	0x00				......				0x00	chksum

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		error code			
F1	1F	E2	2E	B6	6B	A8	8A	00	4D	40	00	00	00	00	01	34	00	00	00	00
NUM_H	NUM_L	INFO_0	INFO_1	INFO_2	......	INFO_63	Checksum													
20	00	01	00	00		00	C8													

Query fingerprint feature storage distribution response packet format table

5.16 Querying the finger in position

(1) Send packet format (Byte):

Check password（4Byte）				CMD1	CMD2	Checksum
0x00	0x00	0x00	0x00	0x01	0x35	0xCA

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	01	35	CA

The format table of the sending packet for querying the state of the finger in place

(2) The response packet format (Byte), the successful completion code is 0x00000000, and the others fail the query. If the finger is in place, return STATE to 1, otherwise STATE is 0:

Check password（4Byte）				CMD1	CMD2	Complete code (4Byte)				STATE	check
0x00	0x00	0x00	0x00	0x01	0x35	0x00	0x00	0x00	0x00	0x01	0xC9

Answering package example:

In this position

Frame header								Data (Len)		Frame header check	Check password				Command		error code				STATE	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0C	81	00	00	00	00	01	35	00	00	00	00	01	C9

Not in position:

Frame header								Data (Len)		Frame header check	Check password				Command		error code				STATE	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0C	81	00	00	00	00	01	35	00	00	00	00	00	CA

Query finger in-position status response packet format table

5.17 Fingerprint feature clearing (synchronization)

(1) Send packet format (Byte): CL_FLAG is the mode flag for clearing fingerprint features. 0x01 means to clear all fingerprints, 0x02 means to clear multiple fingerprints, 0x03 means to clear one fingerprint, and default 0x00 means to clear individual fingerprints. ID_H, ID_L is the fingerprint ID to be deleted (ID value is assigned from 0): This function is only supported by the FPM030 and FPM031 modules.

Clear all or a single fingerprint feature to send the packet format:

Check password（4Byte）				CMD1	CMD2	CL_FLAG	ID_H	ID_L	Checksum
0x00	0x00	0x00	0x00	0x01	0x36	0x00	0x00	0x01	0xC8

Send package example:

Fingerprint feature clear (all):

Frame header								Data (Len)		Frame header check	Check password				Command		CL_ FLAG	ID_H	ID_L	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0A	83	00	00	00	00	01	36	01	00	01	C7

Fingerprint feature clear (single):

Frame header								Data (Len)		Frame header check	Check password				Command		CL_ FLAG	ID_H	ID_L	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0A	83	00	00	00	00	01	36	00	00	01	C8

Clear multiple fingerprint feature sending packet formats:

COUNT is the number of fingerprints to be deleted, and IDn is the fingerprint feature ID value to be deleted (the ID value is assigned from 0).

Check password（4Byte）				CMD1	CMD2	CL_FLAG	COUNT_H	COUNT_L
0x00	0x00	0x00	0x00	0x01	0x36	0x02	0xXX	0xXX
ID1_H	ID1_L	ID2_H	ID2_L	......		IDn_H	IDn_L	Checksum
0x00	0x01	0x00	0x02	......		0x00	0xXX	0xXX

Send package example:

Fingerprint feature clearing (multiple)

Frame header								Data (Len)		Frame header check	Check password				Command		CL_FLAG	COUNT_H	COUNT_L
F1	1F	E2	2E	B6	6B	A8	8A	00	0E	7F	00	00	00	00	01	36	02	00	02
ID1_H		**ID1_L**		**ID2_H**		**ID2_L**		**Checksum**											
00		01		00		02		C2											

The block clear command CL_FLAG is 0x03, enter the first ID and tail ID to be deleted.

Check password（4Byte）				CMD1	CMD2	CL_FLAG	FIRST ID_H	FIRST ID_L
0x00	0x00	0x00	0x00	0x01	0x36	0x03	0x00	0x02
LAST_ ID1_H	**LAST_ ID1_L**	**Checksum**						
0x00	0x06	0xBE						

Sending package example:

Fingerprint feature clearing (block deletion):

Check password（4Byte）				CMD1	CMD2	CL_FLAG	FIRST ID_H	FIRST ID_L
F1	1F	E2	2E	B6	6B	A8	8A	00
LAST_ ID1_H	**LAST_ ID1_L**	**Checksum**						
00	06	BE						

Fingerprint feature clearing send packet format table

(2) Response packet format (Byte):

Check password（4Byte）				CMD1	CMD2	Complete code (4Byte)				Checksum
0x00	0x00	0x00	0x00	0x01	0x36	0x00	0x00	0x00	0x00	0xC9

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		error code				Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	36	00	00	00	00	C9

Query fingerprint feature clear result response packet format table

5.18 Registration fingerprint confirmation

This command is used to confirm the successful entry after the fingerprint is successfully registered, and is used according to the industry application customer requirements.

(1) Send packet format (Byte):

Check password（4Byte）				CMD1	CMD2	Checksum
0x00	0x00	0x00	0x00	0x01	0x41	0xBE

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	01	41	BE

Register fingerprint confirmation send package format table

(2) Response packet format (Byte):

Check password（4Byte）				CMD1	CMD2	Complete code (4Byte)				Checksum
0x00	0x00	0x00	0x00	0x01	0x41	0x00	0x00	0x00	0x00	0xBE

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		error code				Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	41	00	00	00	00	BE

Register fingerprint confirmation response packet format table

5.19 Querying the registration fingerprint confirmation result

(1) Send packet format (Byte):

Check password（4Byte）				CMD1	CMD2	Checksum
0x00	0x00	0x00	0x00	0x01	0x42	0xBD

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	01 42	BD

Query the registration fingerprint confirmation result sending packet format table

(2) Response packet format (Byte), including matching results, matching scores, matching id number (reserved) three items:

Check password（4Byte）				CMD1	CMD2	Completion code（4Byte）			
0x00	0x00	0x00	0x00	0x01	0x42	0x00	0x00	0x00	0x00
Match result		Match score		Id (reserved)		Checksum			
0x00	0x01	0x27	0x0F	0x00	0x00	0x86			

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		error code			
F1	1F	E2	2E	B6	6B	A8	8A	00	11	7C	00	00	00	00	01	42	00	00	00	00
Match result		Match score		Id (reserved)			Checksum													
00	01	27	0F	00		00	86													

Query registration fingerprint confirmation result response packet format table

5.20 Fingerprint feature information download

(1) Send packet format (Byte). ID_H, ID_L is the fingerprint ID to be downloaded (the ID value is assigned from 0), and INFO is the size of the total data to be downloaded (maximum 4096 bytes).

Check password（4Byte）				CMD1	CMD2	ID_H	ID_L	INFO_H	INFO_L	Checksum
0x00	0x00	0x00	0x00	0x01	0x51	0x00	0x00	0x07	0xEC	0xBB

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		ID_H	ID_L	INFO_H	INFO_L	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	51	00	00	07	EC	BB

Fingerprint feature information download and send packet format table

(2) Response packet format (Byte):

Check password（4Byte）				CMD1	CMD2	Complete code (4Byte)				Checksum
0x00	0x00	0x00	0x00	0x01	0x51	0x00	0x00	0x00	0x00	0xAE

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		error code				Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	51	00	00	00	00	AE

Query registration fingerprint confirmation result response packet format table

5.21 Fingerprint feature data download

(1) Send packet format (Byte): FRAME is the current frame (the first frame starts from 0), the number of DATA data is not fixed, and the length of data sent by the user is determined (len-9, the maximum is not More than 128byte). Download all the data of INFO.

Check password（4Byte）				CMD1	CMD2	FRAME_H	FRAME_L	...	DATAn	Checksum
0x00	0x00	0x00	0x00	0x01	0x52	0xXX	0xXX	...	0xXX	0xXX

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		FRAME_H	FRAME_L	...	DATAn	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	89	04	00	00	00	00	01	52	0xXX	0xXX	...	0xXX	0xXX

Fingerprint feature data download and transmission package format table

(2) Response packet format (Byte)

Check password（4Byte）				CMD1	CMD2	Complete code (4Byte)				Checksum
0x00	0x00	0x00	0x00	0x01	0x52	0x00	0x00	0x00	0x00	0xAD

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		error code				Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	52	00	00	00	00	AD

Fingerprint feature data download response packet format table

5.22 Uploading fingerprint feature information

(1) Send packet format (Byte), ID_H, ID_L is the fingerprint ID to be uploaded:

Check password（4Byte）				CMD1	CMD2	ID_H	ID_L	Checksum
0x00	0x00	0x00	0x00	0x01	0x53	0x00	0x00	0xAC

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		ID_H	ID_L	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	09	84	00	00	00	00	01	53	00	00	AC

Fingerprint feature information upload and send packet format table

(2) Response packet format (Byte), the size of the data uploaded by INFO.

Check password（4Byte）				CMD1	CMD2	INFO_H	INFO_L	Checksum
0x00	0x00	0x00	0x00	0x01	0x53	0x00	0x00	0x00

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Error code				INFO_H	INFO_L	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0D	80	00	00	00	00	01	53	00	00	00	00	07	EC	B9

Fingerprint feature information upload response packet format table

5.23 Fingerprint feature data upload

(1) Send packet format (Byte), NUM_H, NUM_L is the number of uploads or frames, and the module uploads 128 bytes each time:

Check password（4Byte）				CMD1	CMD2	NUM_H	NUM_L	Checksum
0x00	0x00	0x00	0x00	0x01	0x54	0x00	0x00	0xAB

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		ID_H	ID_L	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	09	84	00	00	00	00	01	54	00	00	AB

Fingerprint feature data upload and send packet format table

(2) Response packet format (Byte):

Check password（4Byte）				CMD1	CMD2	Complete code (4Byte)				DATA1	...	DATA130	Checksum
0x00	0x00	0x00	0x00	0x01	0x54	0x00	0x00	0x00	0x00	0xXX	...	0xXX	0xXX

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Error code				DATA1	...	DATA130	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	BD	00	00	00	00	00	01	54	00	00	00	00	XX	...	XX	XX

Fingerprint feature data upload response packet format table

5.24 Password Setting Command

(1) Send packet format (Byte): If the setup password is 0x12345678, PSW0~PSW3 are 0x12, 0x34, 0x56, 0x78 respectively. Do not write flash, power down recovery.

Check password（4Byte）				CMD1	CMD2	PSW0	PSW1	PSW2	PSW3	Checksum
0x00	0x00	0x00	0x00	0x02	0x01	0x12	0x34	0x56	0x78	0xE9

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		PSW0	PSW1	PSW2	PSW3	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	02	01	12	34	56	78	E9

Password setting command to send the package format table

(2) Response packet format (Byte):

Check password（4Byte）				CMD1	CMD2	Complete code (4Byte)				Checksum
0x12	0x34	0x56	0x78	0x02	0x01	0x00	0x00	0x00	0x00	0xE9

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Error code				Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	12	34	56	78	02	01	00	00	00	00	E9

Password setting command response packet format table

(3) The password is set successfully. If you re-issue the fingerprint module ID command, the format is as follows.

Check password（4Byte）				CMD1	CMD2	Checksum
0x12	0x34	0x56	0x78	0x03	0x01	0xE8

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	12	34	56	78	03	01	E8

Password setting success check packet format table

5.25 Resetting the fingerprint module

(1) Send packet format (Byte):

Check password（4Byte）				CMD1	CMD2	Checksum
0x00	0x00	0x00	0x00	0x02	0x02	0xFC

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	02	02	FC

Reset fingerprint module send packet format table

(2) Response packet format (Byte):

Check password				CMD1	CMD2	Complete code				Checksum
0x00	0x00	0x00	0x00	0x02	0x02	0x00	0x00	0x00	0x00	0xFC

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Error code				Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	02	02	00	00	00	00	FC

Reset fingerprint module response packet format table

5.26 Obtaining the number of fingerprint templates

(1) Send packet format (Byte):

Check password（4Byte）				CMD1	CMD2	Checksum
0x00	0x00	0x00	0x00	0x02	0x03	0xFB

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	02	03	FB

Get the number of fingerprint templates to send the package format table

(2) Response packet format (Byte): Returns the number of registered modules.

Check password				CMD1	CMD2	Complete code				NUM_H	NUM_L	Checksum
0x00	0x00	0x00	0x00	0x02	0x03	0x00	0x00	0x00	0x00	0x00	0x04	0xF7

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Error code				NUM_H	NUM_L	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0D	80	00	00	00	00	02	03	00	00	00	00	00	04	F7

Get the number of fingerprint template response packet format table

5.27 Gaining gain

(1) Send packet format (Byte):

Check password（4Byte）				CMD1	CMD2	Checksum
0x00	0x00	0x00	0x00	0x02	0x09	0xF5

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	02	09	F5

Get Gain Send Packet Format Table

(2) Response packet format (Byte):

Check password				CMD1	CMD2	Complete code				Shift	Gain	PxlCtrl	Checksum
0x00	0x00	0x00	0x00	0x02	0x09	0x00	0x00	0x00	0x00	0x1F	0x00	0x04	0xD2

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Error code				Shift	Gain	PxlCtrl	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0E	7F	00	00	00	00	02	09	00	00	00	00	1F	00	04	D2

Get the gain response packet format table

5.28 Obtaining matching thresholds

(1) Send packet format (Byte):

Check password				CMD1	CMD2	Checksum
0x00	0x00	0x00	0x00	0x02	0x0B	0xF3

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	02	0B	F3

Get matching threshold sending packet format table

(2) Response packet format (Byte):

Check password				CMD1	CMD2	Complete code				Threshold_H	Threshold_L	Checksum
0x00	0x00	0x00	0x00	0x02	0x0B	0x00	0x00	0x00	0x00	0x21	0x34	0x9E

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Error code				Threshold_H	Threshold_L	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0D	80	00	00	00	00	02	0B	00	00	00	00	21	34	9E

Get matching threshold response packet format table

5.29 Setting Sleep Mode

(1) Send packet format (Byte): The MODE field is 1Byte, 00 means to enter normal sleep mode, 01 means to enter deep sleep mode.

Check password				CMD1	CMD2	MODE	Checksum
0x00	0x00	0x00	0x00	0x02	0x0C	0x00	0xF2

Send package example:

Normal sleep:

Frame header								Data (Len)		Frame header check	Check password				Command		MODE	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	08	85	00	00	00	00	02	0C	00	F2

Deep sleep:

Frame header								Data (Len)		Frame header check	Check password				Command		MODE	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	08	85	00	00	00	00	02	0C	01	F1

Set sleep mode send packet format table

(2) Response packet format (Byte):

Check password				CMD1	CMD2	Complete code				Checksum
0x00	0x00	0x00	0x00	0x02	0x0C	0x00	0x00	0x00	0x00	0xF2

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Error code				Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	02	0C	00	00	00	00	F2

Set the sleep mode response packet format table

Note: When the module enters the normal sleep mode, the module can be woken up by pressing the fingerprint sensor. The serial port has no response; the serial port can be accessed after waking up. When the module enters deep sleep mode, an external input reset signal is required to wake up the module, and the serial port does not respond.

5.30 Setting the number of fingerprint registration stitches

(1) Send packet format (Byte): ENROLL_NUM Setting range (1~6)

Check password				CMD1	CMD2	Complete code				ENROLL_NUM	Checksum
0x00	0x00	0x00	0x00	0x02	0x0D	0x06	0x00	0x00	0x00	0x00	0xEB

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		ENROLL_NUM	Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	08	85	00	00	00	00	02	0D	06	EB

Set fingerprint registration splicing send packet format table

(2) Response packet format (Byte):

Check password				CMD1	CMD2	Complete code				Checksum
0x00	0x00	0x00	0x00	0x02	0x0D	0x00	0x00	0x00	0x00	0xF1

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Error code				Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	02	0D	00	00	00	00	F1

Set the fingerprint registration splicing response packet format table

5.31 Setting LED Control Information

(1) Sending packet format (Byte): Some models have no blue light

Check password				CMD1	CMD2	Control mode	Light color	Parameter 1	Parameter 2	Parameter 3	Checksum
0x00	0x00	0x00	0x00	0x02	0x0F	0x03	0x03	0x64	0x00	0x32	0x53

Parameter Description:

Parameter Item	Control Mode	Light Color		Parameter 1	Parameter 2	Parameter 3
Parameter Description	0: Turn off the LED light 1: Turn on the LED light 2: Automatically illuminate the LED when the finger touches 3: PWM control LED light (breathing light) 4: flashing LED light	0: no color control 1: green 2: red 3: red + green 4: blue 5: red + blue 6: green + blue 7: red + blue + green	PWM mode	Maximum duty cycle (Range: 0~100)	Minimum duty cycle (Range: 0~100)	Duty cycle rate per second (unit: 1%/s, range: 0~100)
			Flashing mode	LED lighting duration (Unit: 10ms)	LED off time (Unit: 10ms)	Number of flash cycles

Send package example:

Turn off the LED light:

Frame header								Data (Len)		Frame header check	Check password				Command	
F1	1F	E2	2E	B6	6B	A8	8A	00	0C	81	00	00	00	00	02	0F

Control mode	Light color	Parameter 1	Parameter 2	Parameter 3	Checksum
00	00	00	00	00	EF

Set to the breathing light mode (the traffic lights are lit at the same time, the maximum duty cycle is 100, the minimum duty ratio is 0, and the duty cycle is changed by 50% per second):

Frame header								Data (Len)		Frame header check	Check password				Command	
F1	1F	E2	2E	B6	6B	A8	8A	00	0C	81	00	00	00	00	02	0F

Control mode	Light color	Parameter 1	Parameter 2	Parameter 3	Checksum
03	03	64	00	32	53

Set to flashing mode (green light flashes, lights up 200ms, extinguishes 200ms, cycles 5 times):

Frame header								Data (Len)		Frame header check	Check password				Command	
F1	1F	E2	2E	B6	6B	A8	8A	00	0C	81	00	00	00	00	02	0F

Control mode	Light color	Parameter 1	Parameter 2	Parameter 3	Checksum
04	01	14	14	05	BD

Set LED light control information transmission package format table

(2) Response packet format (Byte):

Check password				CMD1	CMD2	Complete code				Checksum
0x00	0x00	0x00	0x00	0x02	0x0F	0x00	0x00	0x00	0x00	0xEF

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Error code				Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	02	0F	00	00	00	00	EF

Set LED control information response packet format table

5.32 Obtaining system policies

(1) Send packet format (4Byte):

Check password				CMD1	CMD2	Complete code				Checksum
0x00	0x00	0x00	0x00	0x02	0xFB	0x00	0x00	0x00	0x00	0x03

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	02	FB	03

Get System Policy Send Format Table

(2) Response packet format (Byte): Bit0: Internal use. Bit1: Repeat fingerprint check. Bit2: Self-learning function. Bit3: Internal use. Bit4: 360 degree recognition. Bit5: Internal use. Others for reservations.

Check password				CMD1	CMD2	Policy0	Policy1	Policy2	Policy3	Checksum
0x00	0x00	0x00	0x00	0x02	0xFB	0x00	0x00	0x00	0x16	0xEC

Answering packet example: Strategy 0x00000016 for repeated fingerprint detection, self-learning, 360 degrees

Frame header								Data (Len)		Frame header check	Check password				Command		Error code			
F1	1F	E2	2E	B6	6B	A8	8A	00	0F	7E	00	00	00	00	02	FB	00	00	00	00

Policy0	Policy1	Policy2	Policy3	Checksum
00	00	00	16	ED

Get System Policy Response Format Table

5.33 Setting System Policy

(1) Send packet format (4Byte): Bit0: Internal use. Bit1: Repeat fingerprint check. Bit2: Self-learning function. Bit3: Internal use. Bit4: 360 degree recognition. Bit5: Internal use. Others for reservations.

Check password				CMD1	CMD2	Policy0	Policy1	Policy2	Policy3	Checksum
0x00	0x00	0x00	0x00	0x02	0xFC	0x00	0x00	0x00	0x16	0xEC

Sending package example: Strategy 0x00000016 for repeated fingerprint detection,

self-learning, 360 degrees

Frame header								Data (Len)		Frame header check	Check password				Command	
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	02	FC

Policy0	Policy1	Policy2	Policy3	Checksum
00	00	00	16	EC

Set the system policy send format table

(2) Response packet format (Byte):

Check password				CMD1	CMD2	Complete code				Checksum
0x00	0x00	0x00	0x00	0x02	0xFC	0x00	0x00	0x00	0x00	0x02

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Error code				Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	02	FC	00	00	00	00	02

Set the system policy response format table

5.34 Obtaining the fingerprint module ID

(1) Send packet format (Byte):

Check password				CMD1	CMD2	Checksum
0x00	0x00	0x00	0x00	0x03	0x01	0xFC

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	03	01	FC

(2) Response packet format (Byte): The return field length is 16, as follows; return "ML-FPM001-01-101" as an example.

Check password				CMD1	CMD2	Completion code				ID0
0x00	0x00	0x00	0x00	0x03	0x01	0x00	0x00	0x00	0x00	0x4D
ID1	**ID2**	**ID3**	**ID4**	**ID5**	**ID6**	**ID7**	**ID8**	**ID9**	**ID10**	**ID11**
0x4C	0x2D	0x46	0x50	0x4D	0x30	0x30	0x31	0x2D	0x30	0x31
ID12	**ID13**	**ID14**	**ID15**	**Checksum**						
0x2D	0x31	0x30	0x31	0x75						

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Error code				ID0
F1	1F	E2	2E	B6	6B	A8	8A	00	1B	72	00	00	00	00	03	01	00	00	00	00	4D
ID1	ID2	ID3	ID4	ID5	ID6	ID7	ID8	ID9	ID10	ID11	ID12	ID13	ID14	ID15	Checksum						
4C	2D	46	50	4D	30	30	31	2D	30	31	2D	31	30	31	75						

Get fingerprint module ID response packet format table

5.35 Heartbeat command

(1) Send packet format (Byte): Generally used to detect communication with the module.

Check password				CMD1	CMD2	Checksum
0x00	0x00	0x00	0x00	0x03	0x03	0xFA

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	03	03	FA

Heartbeat command send packet format table

(2) Response packet format (Byte):

Check password				CMD1	CMD2	Complete code				Checksum
0x00	0x00	0x00	0x00	0x03	0x03	0x00	0x00	0x00	0x00	0xFA

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Error code				Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	03	03	00	00	00	00	FA

Heartbeat command response packet format table

5.36 Setting the baud rate

(1) Send packet format (4Byte): If the baud rate is set to 9600, Baud0~Baud3 are 0x00002580 respectively.

Check password				CMD1	CMD2	Baud0	Baud1	Baud2	Baud3	Checksum
0x00	0x00	0x00	0x00	0x03	0x04	0x00	0x00	0x25	0x80	0x54

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command	
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	03	04

Baud0	Baud1	Baud2	Baud3	Checksum
00	00	25	80	54

The baud rate is 115200:

Frame header								Data (Len)		Frame header check	Check password				Command	
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	03	04

Baud0	Baud1	Baud2	Baud3	Checksum
00	01	C2	00	36

Set the baud rate send packet format table

(2) Response packet format (Byte):

Check password				CMD1	CMD2	Complete code				Checksum
0x00	0x00	0x00	0x00	0x03	0x03	0x00	0x00	0x00	0x00	0xFA

Answering package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Error code				Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	03	04	00	00	00	00	F9

Set the baud rate response packet format table

5.37 Setting the communication password

(1) Send packet format (Byte): If the setup password is 0x12345678, PSW0~PSW3 are 0x12, 0x34, 0x56, 0x78 (write flash) respectively.

Check password				CMD1	CMD2	PSW0	PSW1	PSW2	PSW3	Checksum
0x00	0x00	0x00	0x00	0x03	0x05	0x12	0x34	0x56	0x78	0xE4

Send package example:

Set the password to: 0x12345678

Frame header								Data (Len)		Frame header check	Check password				Command	
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	03	05

PSW0	PSW1	PSW2	PSW3	Checksum
12	34	56	78	E4

Password recovery default 0x00000000:

Frame header								Data (Len)		Frame header check	Check password				Command	
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	12	34	56	78	03	05

PSW0	PSW1	PSW2	PSW3	Checksum
00	00	00	00	E4

Set communication password transmission packet format table

(2) Response packet format (Byte):

Check password				CMD1	CMD2	Complete code				Checksum
0x12	0x34	0x56	0x78	0x03	0x05	0x00	0x00	0x00	0x00	0xE4

Send package example:

Set the password to: 0x12345678

Frame header								Data (Len)		Frame header check	Check password				Command		Error code				Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	12	34	56	78	03	05	00	00	00	00	E4

The password is: 0x00000000

Frame header								Data (Len)		Frame header check	Check password				Command		Error code				Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	03	05	00	00	00	00	F8

Set communication password response packet format table

(3) If you re-issue the fingerprint module ID command, the format is as follows.

Check password				CMD1	CMD2	Checksum
0x78	0x56	0x34	0x12	0x03	0x01	0xE8

Send package example:

Frame header								Data (Len)		Frame header check	Check password				Command		Checksum
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	12	34	56	78	03	01	E8

Password setting success check packet format table
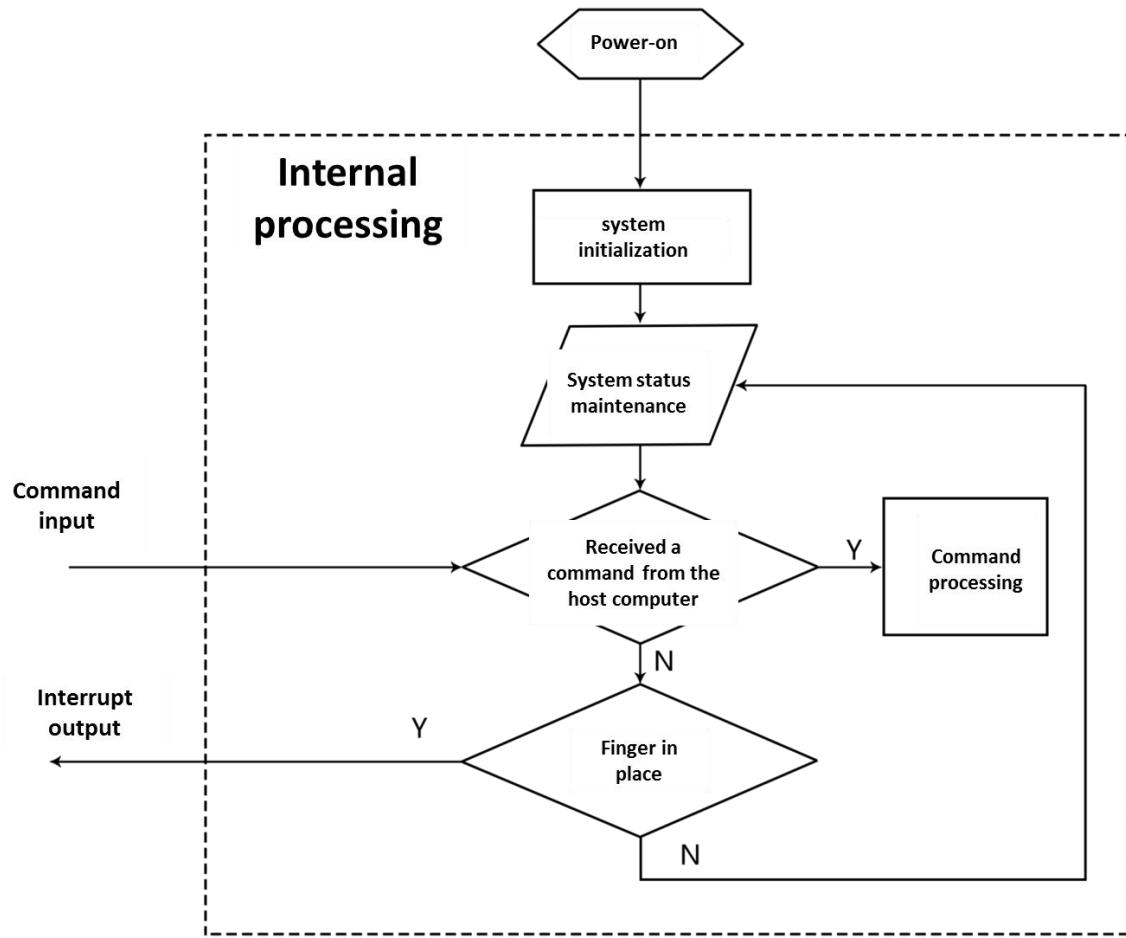
6. Appendix

6.1 Error Information Definition Table

No.	Error code	Description
1	0x00000000	Normal completion
2	0x00000001	Unrecognized command
3	0x00000002	The command data length is illegal.
4	0x00000003	Command field data is illegal
5	0x00000004	The system is busy and cannot execute the current command
6	0x00000005	If there is no request to send the command, the result will be queried.
7	0x00000006	System software reported error
8	0x00000007	Hardware error
9	0x00000008	No finger press detected, timeout
10	0x00000009	Fingerprint extraction error
11	0x0000000A	Fingerprint matching error (fingerprint template library is empty)
12	0x0000000B	Fingerprint data storage space is full
13	0x0000000C	Storage write failed
14	0x0000000D	Storage read failure
15	0x0000000E	The quality of the captured fingerprint image is not good
16	0x0000000F	Fingerprint repetition
17	0x00000010	The area of the drawing is too small, and the contact surface between the finger and the sensor is too small.
18	0x00000011	Finger movement range is too large when taking pictures
19	0x00000012	Finger movement range is too small when taking pictures
20	0x00000013	Fingerprint ID is occupied
21	0x00000014	Module acquisition failed
22	0x00000015	Command forced interrupt
23	0x00000016	Fingerprint feature data does not need to be updated
24	0x00000017	Invalid fingerprint ID
25	0x00000018	Gain adjustment failed
26	0x00000019	Data buffer overflow
27	0x000000FF	Other errors

6.2 Power-on operation process

As shown in the figure below, after the fingerprint module is powered on, the following process is executed. After the PC is powered on or reset, it needs to wait for about 100ms to wait for the fingerprint module to complete the initialization.
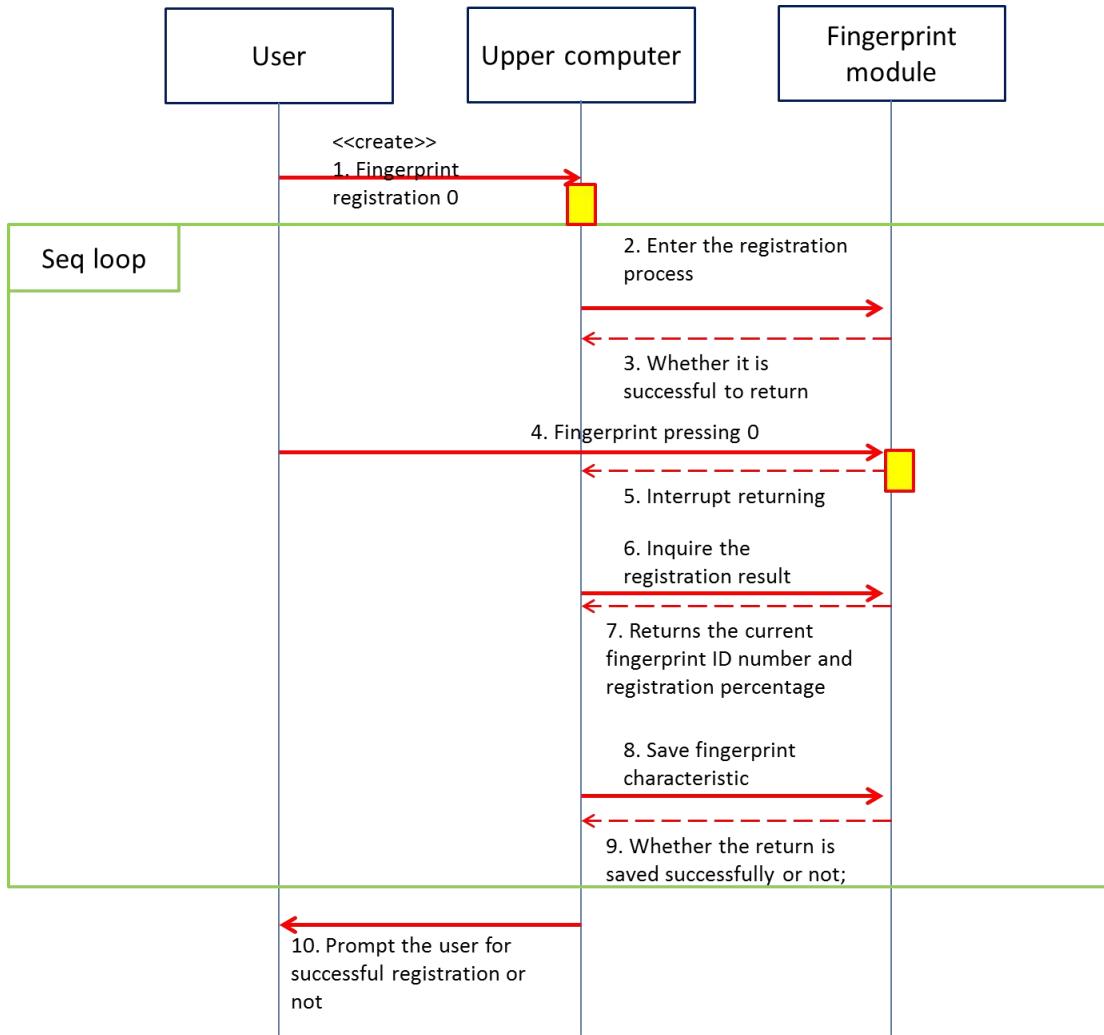
Power-on operation flow chart

Description:

When the finger is pressed on the fingerprint sensor, the initialization of the fingerprint module is not affected, and after the initialization is completed, the interrupt signal of the fingerprint in place is immediately output.

6.3 Registration process

The fingerprint registration process is mainly used to increase the fingerprint feature information of the user for subsequent fingerprint verification. The fingerprint registration process is shown below:

the remigration process figure

Firstly, the user conducts the fingerprint registration operation on the upper computer. When being operated by the user, the upper computer will send the registration process Command to the fingerprint module; the fingerprint module will respond when receiving the message and enter the registration process. After entering the registration process, if the inquired result Command is not send exceeding 10S, the Command entering the registration process will be sent again. When interval time for sending the inquired registration result does not exceed 10S, if not, the entering

registration process will be sent again.

The specification of the registered process is as follows:

1 Send the fingerprint registration Command (0x01 0x11) (Suggestion: REG_IDX is started from 0x01 and increased to 0x06)

2 Inquire the registration result (0x01 0x12) If the completion code is 0x00000000, it shows that this registration is finished;

It can check whether PROC value is increased or not; if so, the registration is successful; Moreover, it needs to check PROC value; if PROC<100; the ③ step is conducted; if PROC >= 100, it shows that the whole registration process is successfully ended; the ④ step is conducted; If the completion code is 0x00000004, it means that the fingerprint data is not processed by the sensor; it will be delayed suitably (200ms); and then the result is obtained; for other reasons (such as: poor image quality, heavy, light or damp finger pressing, continue toregister), the ① step is conducted;

3 Inquire the finger state (0x01 0x35) If the completion code is not 0x00000000 or STATE is 1;

It means that the inquiry is error or the user finger is onwards; it hints the user's fingers are away from sensor to press again; (suggested time delay: 200ms), the finger state is inquired again; when the completion code is 0x00000000 and STATE is 0, the ①step is conducted;
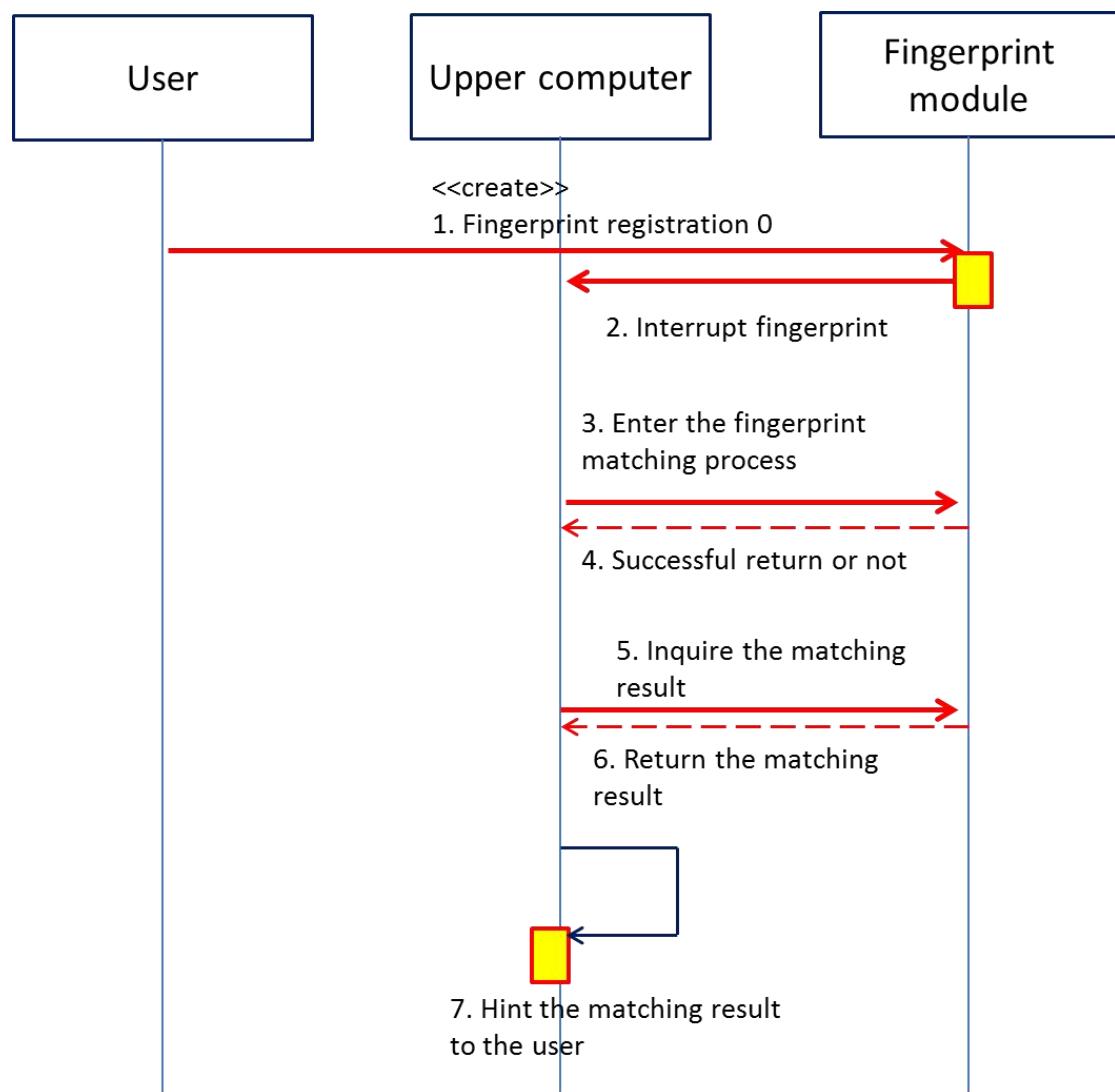
4 Save fingerprint (0x01 0x13)

5 Inquire the fingerprint save result (0x01 0x14) If the completion code is 0x00000000, it shows that the save is finished;If the completion code is 0x00000004, it shows that it is not saved; the time is delayed suitably (200ms); and the result is obtained again; If the completion code is 0x0000000B, flash storage may be full. For the specific reason, the user can refer to the contrast error information definition table of completion code.

6.4 Verification process

The fingerprint verification process is mainly used for verifying the authority of the user; and fingerprint must be confirmed whether it is the registered fingerprint; 10S delay time cannot exceed when entering the fingerprint matching process and inquired machining result; if not, it will be resent;

The verification process is as follows:

verification process figure

The specification of verification process is as follows:
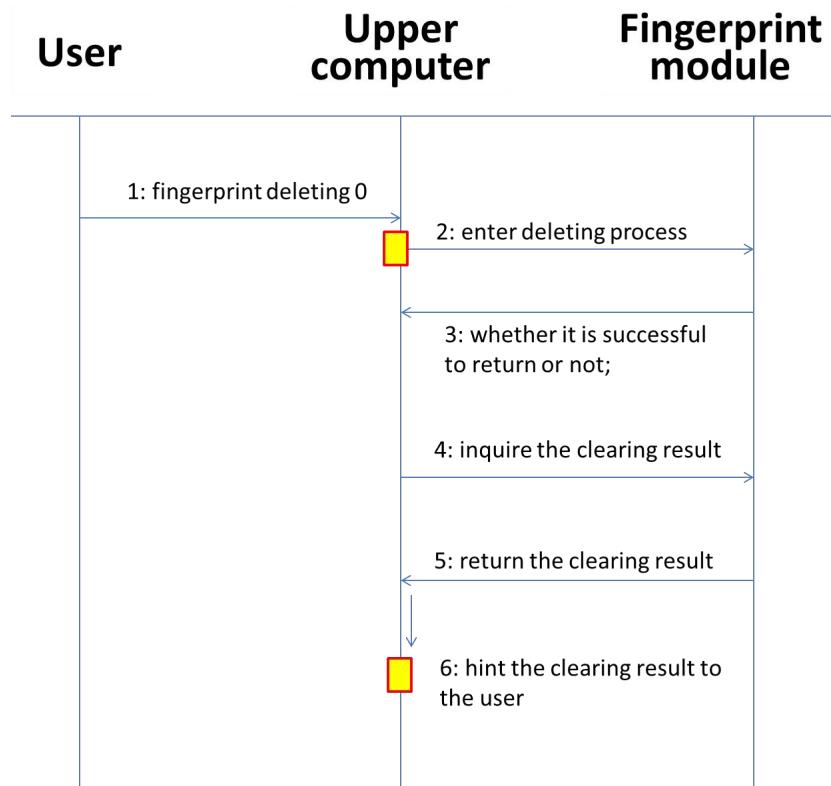
1 Send the fingerprint matching Command (0x01 0x21) Hint the finger pressing before sending the fingerprint matching Command

2   Inquire the matching result (0x01 0x22) If the completion code is 0x00000000, it shows that the matched action is finished; At this time, the matching can be examined;

Result, matching grade and matched fingerprint ID number If the matching result is 1, it shows the fingerprint has been matched; if not, the fingerprint has been not matched; If the completion code is 0x00000004, the matching is not finished; the time is delayed (200ms); and then the result is obtained again; For other reason (such as poor image quality, 0x0000000A, heavy, light or damp finger pressing, etc), please match fingerprint again, and the 1 step is conducted;

6.5 Deleting process

The deleting process is mainly used for deleting the unneeded fingerprint of user.

The deleting process is as follows:

The specification of deleting process is as follows:

1 Send the fingerprint characteristic clearing Command (0x01 0x31) CL_ALL is the

mark whether to clear all fingerprint characteristics;0x01 shows to clear all fingerprints; 0x00 is defaulted to clear single fingerprint; ID_H and ID_L are fingerprint ID to be deleted; The user can delete all, single or multiple fingerprints according to the demands (specific sending format can refer to the sending example

package for clearing the fingerprint characteristics);
2 Inquire the clearing result of fingerprint characteristics (0x01 0x32); if the completion code is 0x00000000, it shows successful deleting; If the completion code is 0x00000004, it shows incomplete deleting; the time is delayed suitably (200ms); and then the result is obtained again; For other errors (such as inexistence of fingerprint ID, error flash hardware, etc), the user can examine the error information contrast table according to the completion code and send the right fingerprint characteristic clearing Command again;

6.6 Example of inspection and function codes

Frame header checksum   = ~ (inspection of frame header +application layer data length) + 1

Sending checksum = ~ (check password+ Command+ data content) +1

Receiving checksum = ~ (check password+ respond Command+ error code+ data content) +1

```
static U8Bit FP_protocol_get_checksum(U8Bit * data, U32Bit length)

{
U32Bit i = 0;
S8Bit sum = 0;

for(i = 0; i < length; i++){
sum += data[i];
}
return (U8Bit)((~sum)+1);

}
```

6.7. Reset function

Purpose: Wake-up function (when the fingerprint head is not touched, the reset can be

used to wake up). Operation mode: Pull the RESET pin of the module down for 10ms and then pull it high.

6.8. Self-learning function

Self-learning updates a small number of feature values that have not been registered after the matching is successful, thereby increasing the success rate of the matching. The process is described as follows:

1 Send a match command (0x01 0x21). Press your finger.

2 Query the fingerprint matching result (0x01 0x22). If the completion code is 0x00000000 and the match result is 0x0001, the match is successful. Go to step 3. Otherwise it fails and there is no need to update the feature value.

3 Update the storage feature value (0x01 0x16). The ID of the match after the match is successfully passed.

4 Query the updated eigenvalue result (0x01 0x17). Whether the query is updated successfully or not.

7. Revision record

Version	Revisions	Date	Modified by
V1.0.1	1.First draft.	2016-01-13	Roy
V1.0.2	1.Modify Command	2016-04-26	Roy
V1.0.3	1.Modify Query Fingerprint ID	2016-04-29	Roy
V1.0.5	Fix bug	2016-05-11	Roy/Jack
V1.0.6	Add Low Power Model	2016-06-02	Jack/lidong
V1.0.7	optimize	2017-03-20	lidong/Alex
V1.0.8	optimize	2017-07-03	lidong/Alex
V1.0.9	Add Command/optimize	2017-09-01	Alex
V1.1.0	Add Command	2017-11-27	lidong
V1.1.1	Add Command/optimize	2018-01-10	Alex
V1.1.2	Add Command/optimize	2018-03-26	Alex
V1.1.3	optimize	2018-04-25	Alex
V1.1.5	Add Command/optimize	2018-06-21	Alex
V1.1.6	Add Command	2018-08-20	Lance
V1.1.7	Add Command	2018-10-08	Lance
V1.1.8	Optimize	2018-11-16	Lance
V1.1.9	Fix bug	2019-12-25	Phager
V1.1.10	1. Definitions of hardware interface removed (please be referred to the module SPEC for them). 2. Layout optimized.	2020-09-11	Lidong