SilkID Fingerprint Module Communication Protocols

Version: 2.3

Important Notice

Due to constant update of products, our company does not commit that actual products are always in compliance with this manual and will not be liable for any dispute caused by inconsistency between actual technical parameters and those in this manual. No notice will be given prior to changes.

Contents

1 Data Packet Protocols

1.1 Common Data Packet Transmission Protocol

In the data packet protocol that the fingerprint module complies with, one data packet contains 13 bytes, **and the host and the module communicate with each other using data packets of the same format.**

Data packet format:

Start	Command	Param	Size	Flag/Error	Checksum	End
1byte	**1byte**	**4bytes**	**4bytes**	**1byte**	**1byte**	**1byte**

1. Start code**: (1 byte), start flag of a data packet, which is always** 0x70**.**

2. Command**: (1 byte), a command. For details, see the** fingerprint module command table**.**

3. Param**: (4 bytes), user ID or system parameter.**

4. Size**: (4 bytes), length of data to be transmitted in the data packet plus the checksum length.**

5. Flag/Error**: (1 byte. When the host transmits data to the module, the value is** Flag**, indicating the transmitted data. When the module transmits data to the host, the value is** Error**, indicating the command execution result.**

6. Checksum**: 1 byte, used to check whether a data packet is valid. Checksum is the lower eight bits of the sum of bytes from Start code to Flag/Error. That is, checksum is the remainder of the sum of all the previous bytes divided by** 256**. (**Note**: Some commands may contain** 2-byte **or** 4-byte **checksum during transmission. The calculation of the checksum is the same regardless of the checksum length.)**

7. End code**: 1 byte, end flag of a data packet, which is always** 0x0A.

Note:

1. **The host and module communicate with each other using data packets of the same format. In general, the host sends a command and the module sends a response. Data packets sent from the host to the module are called** "request data packets" **and data packets sent from the module to the host are called** "response data packets." **In some conditions, however, two response data packets may be received for one request data packet, for example, the** MD_ENROLL_SCAN **command.**

2. **Data transmission adopts little endian. That is, the least significant byte is firstly transmitted. For example, when** 4-byte **data** 0x764308 **needs to be transmitted, the sequence for byte transmission is** 0x08 0x43 0x76 0x00**. The host and module must comply with this transmission rule.**

3. **Checksum is the remainder of the sum of all the previous bytes divided by** 256**. For example:**

Start	Command	Param	Size	Flag/Error	Checksum	End
0x70	0x15	0x03	0x200	0x70	0xFA	0x0A

0x70+0x15+0x03+0x00+0x00+0x00+0x00+0x02+0x00+0x00+0x70=0xFA

Checksum=0xFA&0xFF=0xFA

The transmission sequence of the data packet on the serial port is 0x70 0x15 0x03 0x00 0x00 0x00 0x00 0x02 0x00 0x00 0x70 0xFA 0x0A.

1.2 Extended Data Packet Transmission Protocol

The extended transmission protocol is recommended when a large amount of data needs to be transmitted between the module and the host. The extended transmission protocol is capable of segmenting a batch of data into several small data packets, thereby improving data transmission reliability and reducing data transmission errors.

Note:

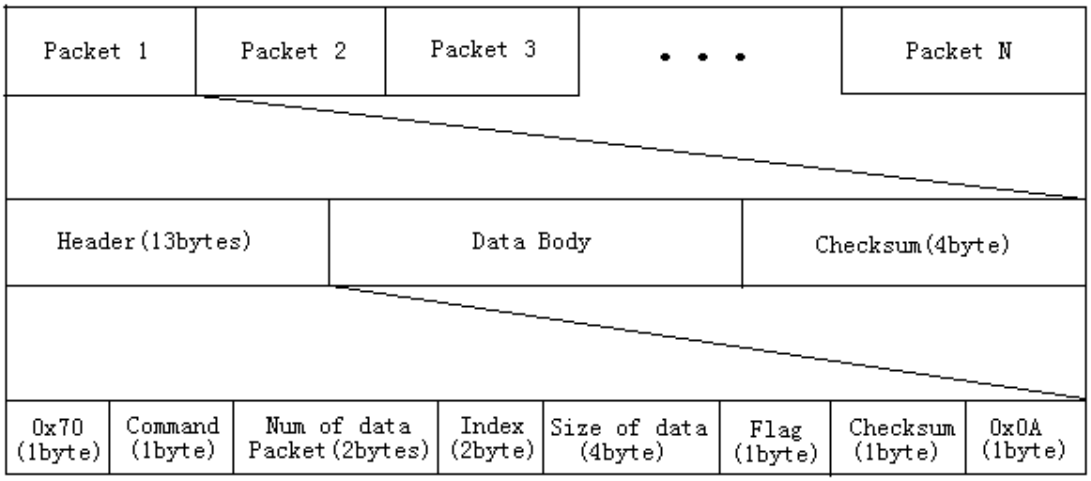
The fingerprint module can receive or transmit a data packet with the maximum length of 4 KB**. Therefore, the length of a data packet sent from the host to the fingerprint module cannot be larger than** 4 KB**.**

When the host reads batch data from the fingerprint module, it can specify the maximum length of a data packet to be transmitted (smaller than 4 KB) **in the request command. If the maximum length of a data packet to be transmitted is not specified, the fingerprint module transmits data with the default data packet length** (4 KB)**.**

Each data packet contains a 4-byte **checksum, which is the sum of bytes in the data body. If an error occurs on any data packet during transmission, the fingerprint module immediately exits the data receiving mode and enters the common mode.**

Data packet format:

Start	Command	PacketNum	Index	Size	Flag/Error	Checksum	End
1byte	1byte	4bytes		4bytes	1byte	1byte	1byte



For example, when the MD_ENROLL_IMAGE_X **command is used to transmit data of 16300 bytes to the module, with the maximum length of each data packet of 4096 bytes, the 16300-byte data is segmented into four small data packets for transmission, and the lengths of the four data packets are 4096 bytes, 4096 bytes, 4096 bytes, and 4012 bytes.**

Packet 1 0x70 0x80 0x04 0x00 0x1000 0x00 0xD4 0x0A

Packet 2 0x70 0x80 0x04 0x01 0x1000 0x00 0xD5 0x0A

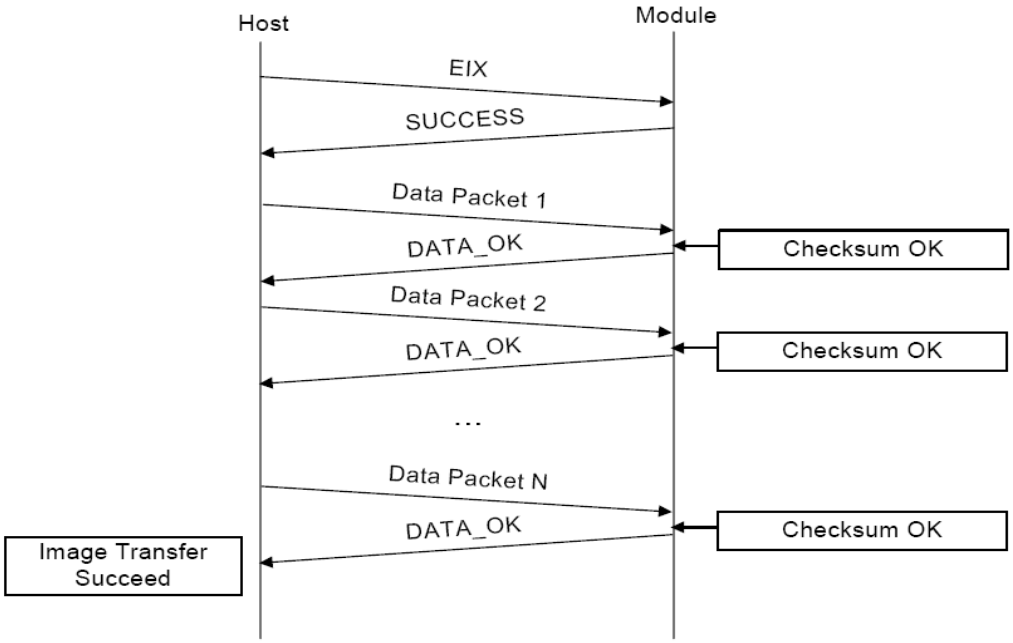
Packet 3 0x70 0x80 0x04 0x02 0x1000 0x00 0xD6 0x0A

Packet 4 0x70 0x80 0x04 0x03 0x0FAC 0x00 0x82 0x0A

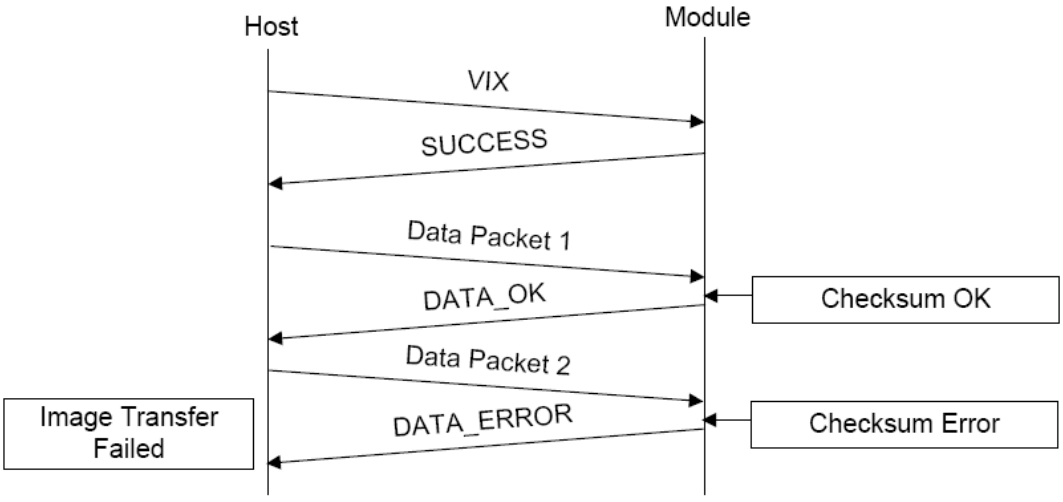
After successfully receiving a data packet, the module returns DATA_OK(0x83)**. Otherwise, the module returns** DATA_ERROR(0x82)**. If the data sender receives** DATA_ERROR(0x82)**, the data sender immediately exits the data transmission mode.**

Example:

1. **Data transmission process for the** MD_ENROLL_IMAGE_X **command.**



2. **Data transmission failure process for the** MD_IDENTIFY_IMAGE_X **command.**



2 Module Applications

The module supports three kinds of application mode: authentication mode, reader (image) mode**,** reader (template) mode**. Different mode can be chosen under different application.**

2.1 Authentication Mode

The default mode of the module is authentication mode. When the user presses the FP, the FP will be identified in 1:N and the result will be send to the Host. Template format supports ZKTeco format only in authentication mode.

2.2 Reader (image) Mode

The module outputs the FP image to the Host, and the host will have some related processing after receiving the image in this mode.

2.3 Reader (template) Mode

The module outputs the FP template to the Host, and the host will have some related processing after receiving the template. The format of template data is decided by the format set before.

3 System Configuration

System parameters are used to enable or disable a function, or modify the effective duration for executing a command. Each parameter has one parameter ID. If the value of a parameter needs to be read or changed, the parameter ID needs to be specified.

The value of each parameter can be read by using the MD_SYS_RP command. Some parameters are read/write while some parameters are read-only. All rewritable parameters can be rewritten by using the MD_SYS_WP command. To make modified system parameters take effect at the next restart, the MD_SYS_SP command needs to be executed to save the parameters. That is, if system parameter values need to be changed, change the system parameter values and run the MD_SYS_SP command to save the changes so that they take effect at the next restart. For details about the read/write process, see descriptions of the MD_SYS_WP and MD_SYS_RP commands in this document.

Table 1 System parameter setting table

For details about how to set system parameters, see 3.1, 3.2, and 3.3.

Parameter	Parameter ID	Description	Value (* representing default value)
Save verification log	0x36	Whether to save logs when fingerprint verification or identification is conducted.	*0x30: no 0x31: yes
User capacity	0x79	Maximum number of users who can be enrolled	A 4-byte integer
Log quantity	0x7C	Number of saved verification logs	A 4-byte integer
Log capacity	0x7B	Maximum number of verification logs that can be saved	A 4-byte integer
Fingerprint quantity	0x73	Number of enrolled fingerprint templates	A 4-byte integer
Fingerprint capacity	0x74	Maximum number of fingerprint templates that can be enrolled	A 4-byte integer

Timeout	0x62	Timeout time	0x30: never timeout 0x31: 1 second *0x3A: 10 seconds 0x44: 20 seconds
Baud rate	0x71	Baud rate	*0x35: 115200 bps
Automatic response	0x82	Whether to automatically send a response	*0x30: not sending a response 0x31: sending a response
Module ID	0x6D	Module ID	0 ~ 65535
Working mode	0×50	Authentication mode: FP identification in the module Reader(image): send the image to the Host and processing by host Reader(template): send the template to the Host and processing by host	*0x30:authentication mode 0x31: reader(image) 0x32: reader(template)
Template format	0×51	Set the template format, the module can identify the ZKTeco template format only	*0x30: ZK 0x32:Ansi 378 0x33:ISO 19794-2
Flashing light control	0×31	Flashing light control operation	Types of high 8-bit indication lamp: 0x80:green light , 0x40:red light , 0xC0:yellow light Display duration of low 8-bit indication light, range:1-255sec
Firmware version	0x6E	Firmware version	Four 1-byte characters
Creation date	0x89	Creation date	4-byte data

3.1 Save Verification Log

This parameter specifies whether to save logs when fingerprint verification or identification is conducted. Verification logs can be used as a basis for attendance. This parameter is useful when the module is used as attendance machine. When the module is used as an access control device **or** real-time monitoring **device,** enable **this parameter if verification logs need to be saved.**

Note:

The module does not **have a** clock chip**. Therefore, the module time is changed to zero after each restart. To ensure accuracy of verification logs, synchronize the clock** every time **the module is** connected **and synchronize the clock once** every other hour**.**

Parameter ID	Read/Write Type	Valid Value	Default Value
0x36	**Read/Write**	**0x30: Verification logs are not saved.** **0x31: Verification logs are saved.**	**0x30: Verification logs are not saved.**

3.2 User Capacity

This parameter specifies the maximum number of users that can be supported.

Note**: The maximum number of users that can be accommodated varies with the version.**

Parameter ID	Read/Write Type	Valid Value	Default Value
0x79	**Read**		

3.3 Log Quantity

This parameter indicates the number of stored logs.

Parameter ID	Read/Write Type	Valid Value	Default Value
0x7C	**Read**		

3.4 Log Capacity

This parameter specifies the maximum number of logs that can be accommodated. The maximum number of logs that can be accommodated varies with the version.

Parameter ID	Read/Write Type	Valid Value	Default Value

0x7B	Read		

3.5 Fingerprint Quantity

This parameter indicates the number of enrolled fingerprint templates.

Parameter ID	Read/Write Type	Valid Value	Default Value
0x73	Read		

3.6 Fingerprint Capacity

This parameter specifies the maximum number of fingerprint templates that can be enrolled. The maximum number of fingerprint templates that can be enrolled varies with the version.

Parameter ID	Read/Write Type	Valid Value	Default Value
0x74	Read		

3.7 Timeout

This parameter specifies the timeout time.

After the module sends a response command to the host during fingerprint enrollment, the module automatically exits the current enrollment state and sends a timeout response to the host if it does not receive a command requiring further action within the preset time.

When data is transmitted over the extended data transmission protocol, the waiting time between two data packets or continuous commands cannot exceed the value. Otherwise, the module automatically exits due to timeout and sends a timeout response to the host.

Parameter ID	Read/Write Type	Valid Value	Default Value
0x62	Read/Write	0x30: never timeout 0x31: 1 second *0x3A: 10 seconds 0x44: 20 seconds	0x3A: 10 seconds

3.8 Baud Rate

This parameter specifies the baud rate for RS232 **communication. The default value is** 115200 bps**.**

Parameter ID	Read/Write Type	Valid Value	Default Value
0x71	**Read**	***0x35: 115200 bps**	***0x35: 115200 bps**

3.9 Automatic Response

This parameter specifies whether the module needs to automatically send a response data packet to the host, even if the module does not receive any request command from the host.

Parameter ID	Read/Write Type	Valid Value	Default Value
0x82	**Read/Write**	***0x30: not sending a response** **0x31: sending a response**	***0x30: not sending a response**

3.10 Module ID

Each module has a default module ID. This parameter is very useful when multiple modules are connected to the system at the same time.

Parameter ID	Read/Write Type	Valid Value	Default Value
0x6D	**Read/Write**	**0 ~ 65535**	**1**

3.11 Firmware Version

A firmware version is represented by four characters. For example, the value of V1.0A **is** 0x56313041**, which indicates** V, 1, 0**, and** A **in** ASCII **code.**

Parameter ID	Read/Write Type	Valid Value	Default Value
0x6E	**Read**	**Four 1-byte characters**	

3.12 Creation Date

This parameter indicates the creation date of the firmware, which is a 4-byte value. For example, the value of the creation date 2008-12-30 is 0x08123000**.**

Parameter ID	Read/Write Type	Valid Value	Default Value

0x89	Read		

3.13 Working Mode

The module supports three kinds of working mode: authentication mode, reader(image) mode, reader(template)mode.

Parameter ID	Read/Write	Effective value	Default
0x50	Read/Write	*0x30: authentication mode 0x31: reader(image) 0x32: reader(template)	*0x30: authentication mode

3.14 Template Format

The module supports three kings of template format: ZK, Ansi, ISO.

Parameter ID	Read/Write	Effective value	Default
0x51	Read/Write	*0x30: ZK 0x32:Ansi 378 0x33:ISO 19794-2	*0x30: ZK

3.15 Flashing Light Control

Control the display of red light, yellow light or green light.

Parameter ID	Read/Write	Effective value	
0x31	Write	Types of high 8-bit indication lamp: 0x80:green light , 0x40:red light ,	

		0xC0:yellow light Display duration of low 8-bit indication light, range:1-255sec	

4 Module Operation Commands

Table 2 command table

Category	Name	Code	Description
System configurations	MD_SYS_WP	0x11	Writes data into a system parameter.
	MD_SYS_SP	0x12	Saves a system parameter.
	MD_SYS_RP	0x13	Reads a system parameter.
	MD_SYS_STATUS	0x14	Queries system status.
Enrollment	MD_ENROLL_SCAN	0x15	Enrolls a user fingerprint template by scanning a fingerprint three times.
	MD_ENROLL_IMAGE_X	0x80	Enrolls a fingerprint template by using a fingerprint image.
	MD_ENROLL_TMP	0x17	Registers a user by using a fingerprint template.
Verification	MD_VERIFY_SCAN	0x18	Checks whether a user exists by means of 1:1 scanning.
Identification	MD_IDENTIFY_IMAGE_X	0x81	Identifies a user using a fingerprint image.
	MD_IDENTIFY_FREE	0x2F	Identifies the fingerprint in the FreeScan mode.
Deletion	MD_DEL_ALL_TMP	0x27	Deletes all fingerprint templates.
	MD_DEL_TMP	0x26	Deletes the fingerprint template of a specified user.
Fingerprint template	MD_READ_TMP_X	0x89	Reads the fingerprint template of a specified user.
	MD_SCAN_TEMPLATE	0xFC	Scans the fingerprint template for the current finger.
User	MD_ADD_USER	0xF1	Adds a user.
	MD_READ_USER	0xF2	Reads data of a user.
	MD_DELETE_USER	0xF3	Deletes a user.
	MD_DEL_ALL_USER	0xF5	Deletes all users.
Fingerprint image	MD_SCAN_IMAGE_X	0x83	Reads a fingerprint image from the module.

Time and logs	MD_SET_TIME	0x4A	Sets current time.
	MD_GET_TIME	0x4B	Gets module time.
	MD_DEL_ALOG	0x9E	Deletes all logs.
	MD_LOAD_LOG_X	0xA4	Reads all log data.
Database	MD_LOAD_USER_X	0xA0	Reads all user data.
	MD_LOAD_TMP_X	0xA2	Reads data of all fingerprint templates.
	MD_DEL_DB	0xF8	Deletes all data.
	MD_WT_FILE_X	0xAA	Uploads the fingerprint template storage file.
	MD_WU_FILE_X	0x42	Uploads the user information storage file.
	MD_RT_FILE_X	0XAB	Downloads the fingerprint template storage file.
	MD_RU_FILE_X	0x43	Downloads the user information storage file.
Firmware Updating	MD_UPDATE_FW	0x72	Firmware Updating
Module Reset	MD_RESET	0xD0	Module Reset
Enable Device	MD_ENABLEDEVICE	0xFB	Enable Device
Disable Device	MD_DISABLEDEVICE	0xFA	Disable Device

Note:

The symbol # indicates a changeable value, NULL **indicates** 0**, and** N/A **indicates any value in the command description.**

4.1 Flag/Error Table

Name	Code	Description
BUSY	0x34	The system is busy.
SUCCESS	0x61	Succeeded
FAIL	0x63	Failed
TIME_OUT	0x64	Timeout

PARAM_ERROR	0x68	Parameter error
NOT_FOUND	0x69	Not found or not supported
MEM_FULL	0x6D	The storage capacity is full.
FP_LIMIT	0x72	The maximum number of fingerprint templates that can be enrolled (10) is exceeded.
INVALID_ID	0x76	Invalid ID
CANCELED	0x81	The command is cancelled.
DATA_ERROR	0x82	Transmission data error
EXIST_FINGER	0x86	The fingerprint already exists.

4.2 MD_SYS_WP

This command is used to change a system parameter value of the current module, but not to save the system parameter value. The original parameter value is retained upon the next power-on and restart. To save a changed parameter value and make it take effect upon restart, run the MD_SYS_SP command after running the MD_SYS_WP command.

The request command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x11	
Param	value	Valid Value
Size	NULL	
Flag	0x31 0x36 0x82 0x62 0x50 0x6D	SID_GPIO_LEVEL SID_SAVE_LOG SID_AUTO_ACK SID_TIMEOUT SID_MODULE_IDENTIFY SID_MODULE_ID
Checksum	#	
End code	0x0A	

The response command is as follows:

Data Field	Value	Description
Start code	0x70	

Data Field	Value	Description
Command	0x11	
Param	NULL	
Size	NULL	
Error	0x61 0x69 0x68 0x34	SUCCESS NOT_FOUND PARAM_ERROR BUSY
Checksum	#	
End code	0x0A	

Error code:

Error code	Description
SUCCESS	Writing data into a parameter is successful.
NOT_FOUND	This parameter is not found or is not supported.
BUSY	The module is busy in processing other commands.

4.3 MD_SYS_SP

This command is used to save a system parameter value to a file. After this command is executed, a parameter value changed by the MD_SYS_WP **command is effective upon restart.**

The request command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x12	
Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

The response command is as follows:

Data Field	Value	Description
Start code	0x70	

Command	0x12	
Param	NULL	
Size	NULL	
Error	0x61 0x34	SUCCESS BUSY
Checksum	#	
End code	0x0A	

4.4 MD_SYS_RP

This command is used to read the value of a system parameter based on the parameter ID.

The request command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x13	
Param	NULL	
Size	NULL	
Flag	0x36 0x82 0x62 0x6E 0x71 0x73 0x74 0x79 0x7C 0x7B 0x89 0x6D	SID_SAVE_LOG, SID_AUTO_ACK, SID_TIMEOUT, SID_FW_VER, SID_BAUDRATE, SID_ENROLL_FP, SID_FP_COUNT, SID_USER_COUNT, SID_LOG_NUM, SID_LOG_COUNT SID_BUILD_NUM SID_MODULE_ID
Checksum	#	
End code	0x0A	

The response command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x13	
Param	Value NULL	Value – the parameter value is returned successfully. NULL – the reading fails.
Size	NULL	
Error	0x61 0x69 0x34	SUCCESS NOT_FOUND BUSY
Checksum	#	
End code	0x0A	

4.5 MD_SYS_STATUS

This command is used to query the current status of the system that is, whether the system is idle or busy in processing other commands.

The request command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x14	
Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

The response command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x14	
Param	0x34	BUSY
Size	NULL	

Error	0x61	SUCCESS
Checksum	#	
End code	0x0A	

4.6 MD_ENROLL_SCAN

This command is used to enroll user fingerprint by means of scanning. After sending an enrollment command, the module waits until a user presses a finger on the fingerprint reader. It starts enrolling the user fingerprint after detecting that the user presses a finger.

A fingerprint template can be successfully enrolled only after a user presses a finger three times. When a fingerprint is successfully scanned each time, the module flashes yellow once and sends a notification command to the host. After fingerprint is successfully enrolled, the module flashes green once. If the enrollment fails, the module flashes red once.

The request command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x15	
Param	User ID	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

The response command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x15	
Param	User ID	
Size	Finger ID	(0~9)
Flag/Error	0x62 0x63 0x61 0x6D 0x72 0x76 0x86	SCAN_SUCCESS FAIL SUCCESS MEM_FULL FINGER_LIMIT INVALID_ID EXIST_FINGER

Checksum	#	
End code	0x0A	

Error code:

Error code	Description
SCAN_SUCCESS	Fingerprint is scanned successfully.
FAIL	Failed
SUCCESS	The enrollment is successful.
MEM_FULL	The module memory is full.
INVALID_ID	The user ID is invalid.
EXIST_FINGER	The fingerprint already exists.
FP_LIMIT	The enrolled fingerprint templates are out of range (10).

For example, when fingerprint needs to be enrolled for a user with the user ID of 0x023, the request data packet is shown in the following table.

Start	Command	Param	Size	Flag	Checksum	End
0x70	0x15	0x0023	0x00	0x00	0xA8	0x0A

4.7 MD_ENROLL_IMAGE_X

This command is used to enroll fingerprint for a specified user by using the fingerprint image transmitted from the host. Fingerprint images are transmitted over the extended transmission protocol.

Note:

1. During enrollment, a fingerprint template may fail to be enrolled but, normally the specified user will be created.

2. A fingerprint image must be an 8-bit gray image in BMP format. It is recommended that images read by the module be used for enrollment as the identification rate may be affected if other images are used. In addition, only the bitmap data excluding the BMP headers and palette data, rather than the entire BMP image is transmitted.

The request command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x80	

Param	User ID	
Size	Image size	Size of a fingerprint image
Flag	NULL	
Checksum	#	
End code	0x0A	

The response command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x80	
Param	User ID	
Size	NULL	
Error	0x61 0x63 0x6C 0x6D 0x72 0x76	SUCCESS FAIL TIME_OUT MEM_FULL FP_LIMIT INVALID_ID
Checksum	#	
End code	0x0A	

4.8 MD_IDENTIFY_IMAGE_X

This command is used to identify a user by using fingerprint images in the 1:N manner. Data packets are transmitted over the extended transmission protocol.

The request command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x81	
Param	Image size	Size of a fingerprint image
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

The response command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x81	
Param	Param NULL	Param – when the identification is successful, the user ID is 0xFFFF & Param and the fingerprint index are (Param >> 16) & 0x0F. NULL – the identification fails.
Size	NULL	
Error	0x61 0x63 0x6C 0x69	SUCCESS FAIL TIME_OUT NOT_FOUND
Checksum	#	
End code	0x0A	

4.9 MD_ENROLL_TMP

This command is used to enroll fingerprint for a user by using the fingerprint template transmitted from the host. The fingerprint template transmitted here is real fingerprint template data rather than the entire structure data of the fingerprint template.

Note: The data length of the transmitted fingerprint template cannot be larger than the maximum fingerprint template length supported by the module. Otherwise, PARAM_ERROR is returned.

The request command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x17	
Param	User ID	
Size	Template size	
Flag	0 0x84	0-None 0x84-CHECK_FINGER
Checksum	#	

End code	0x0A	
Data	Template Data	Template data
Checksum of data	#	Checksum of template data

The response command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x17	
Param	User ID	
Size	NULL	
Error	0x61 0x6D 0x72 0x76 0x63 0x86	SUCCESS MEM_FULL FINGER_LIMIT INVALID_ID FAIL EXIST_FINGER
Checksum	#	
End code	0x0A	

4.10 MD_VERIFY_SCAN

This command is used to check whether a user exists by scanning a fingerprint template in the 1:1 manner.

If the "save verification log" parameter is enabled, the module automatically saves the log and returns to the current log status.

The request command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x18	
Param	User ID	
Size	NULL	
Flag	0x61	
Checksum	#	
End code	0x0A	

The response command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x18	
Param	User ID	
Size	NULL	
Error	0x63 0x6C 0x61	FAIL TIME_OUT SUCCESS
Checksum	#	
End code	0x0A	

4.11 MD_IDENTIFY_FREE

This command is used to identify a user in the 1:N manner. When a user presses a finger on the fingerprint reader, the module automatically identifies the finger and sends the comparison result to the host.

The module sends the following command:

Data Field	Value	Description
Start code	0x70	
Command	0x2F	
Param	NULL	User ID: The identification is successful. Error code: The identification fails.
Size	NULL	
Error	0x61 0x63	0x61-SUCCESS 0x63-FAIL
Checksum	#	
End code	0x0A	

4.12 MD_DEL_ALL_TMP

This command is used to delete all the fingerprint templates as well as all the user data.

The request command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x27	
Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

The response command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x27	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
Checksum	#	
End code	0x0A	

4.13 MD_DEL_TMP

This command is used to delete only the fingerprint templates of a user.

The request command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x26	
Param	User ID, finger index	The data of the Param field are (user ID & 0xFFFF) | ((fingerprint index & 0x0F) << 16).
Size	NULL	
Flag	NULL (default) DELETE_ONLY_ONE (0x77)	Deletes a user and the user's fingerprint.

		Deletes the fingerprint of a user.
Checksum	#	
End code	0x0A	

The response command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x26	
Param	Param	
Size	NULL	
Error	0x61 0x63	SUCCESS FAIL
Checksum	#	
End code	0x0A	

4.14 MD_READ_TMP_X

This command is used to download a fingerprint template from the module to the host based on the specified user ID and fingerprint index.

The request command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x89	
Param	User ID+Index	
Size	NULL	
Flag	0 1	0 – Ignores the finger ID. 1 – Downloads a fingerprint template based on the fingerprint index.
Checksum	#	
End code	0x0A	

The response command is as follows:

Data Field	Value	Description

Data Field	Value	Description
Start code	0x70	
Command	0x89	
Param	NULL	
Size	NULL	
Error	0x61 0x63 0x69 0x6C 0x82	SUCCESS FAIL NOT_FIND TIME_OUT DATA_ERROR
Checksum	#	
End code	0x0A	

Note: **The length of each fingerprint template transmitted may not be the same, but; the total length of multiple fingerprint templates transmitted at a time is the same.**

Example 1: **When one fingerprint template of a user is read, the user ID is** 1, **finger ID is** 1, **and the length of the data packet is** 2048 **bytes.**

The request data packet is shown in the following table.

Start	Command	Param	Size	Flag	Checksum	End
0x70	0x89	0x01	0x10800	0x01	0x04	0x0A

Example 2**: When all fingerprint templates of a user are read, the user ID is** 1 **and the length of the data packet is** 2048 **bytes. The request data packet is shown in the following table.**

Start	Command	Param	Size	Flag	Checksum	End
0x70	0x89	0x01	0x800	0x00	0x02	0x0A

4.15 MD_ADD_USER

This command is used to add a user to a module file or modify information about an existing user.

The user data structure is as follows (using 1-byte memory alignment mode):

```
typedef struct _User_
{
  U16 ID;              /* User ID                    */
  U8   Privilege;      /* User permissions           */
```

```
    U8    FpNum;              /* Number of enrolled fingerprint templates of the user (automatic
maintenance)          */

    U16   SecLevel;           /* User encryption level (not supported in this version) */

    U32   PIN2;               /* User No. (not supported in this version)          */

    Char  Name[8];            /* User name                                         */

    Char  Password[5];    /* Password (not supported in this version)              */

    U8    Card[5];            /* Card ID (not supported in this version)           */

}
```

The request command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0xF1	
Param	NULL	
Size	Length of data to be transmitted	User structure length + check data size
Flag	0x61	
Checksum	#	
End code	0x0A	
Data	User Data	User data
Checksum of data	#	Checksum of user data

The response command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0xF1	
Param	NULL	
Size	NULL	
Error	0x61 0x63 0x6D 0x68	SUCCESS FAIL MEM_FULL PARAM_ERROR
Checksum	#	
End code	0x0A	

4.16 MD_READ_USER

This command is used to read the user's information in a module file.

The request command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0xF2	
Param	User ID	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

The response command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0xF2	
Param	NULL	
Size	Length of data to be transmitted	User structure length + verification data size
Error	0x61 0x69	SUCCESS NOT_FIND
Checksum	#	
End code	0x0A	
Data	User Data	User data
Checksum of data	#	Checksum of user data

4.17 MD_DELETE_USER

This command is used to delete a user, including the user and the user's fingerprint.

The request command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0xF3	

Data Field	Value	Description
Param	**User ID**	
Size	**NULL**	
Flag	**NULL**	
Checksum	**#**	
End code	**0x0A**	

The response command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0xF3	
Param	User ID	
Size	NULL	
Error	0x61 0x63	SUCCESS FAIL
Checksum	#	
End code	0x0A	

Example: **Delete a user with the user ID of** 1**.**

The request data packet is as follows:

Start	Command	Param	Size	Flag	Checksum	End
0x70	0XF3	0x01	0x00	0x00	0x64	0x0A

The response data packet is as follows:

Start	Command	Param	Size	Flag	Checksum	End
0x70	0XF3	0x01	0x00	0x61	0XC5	0x0A

4.18 MD_DEL_ALL_USER

This command is used to delete all users and their fingerprint data from module files. All fingerprint templates will also be deleted when all users are deleted.

The request command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0xF5	
Param	NULL	
Size	NULL	
Flag	NULL	

Checksum	#	
End code	0x0A	

The response command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0xF5	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
Checksum	#	
End code	0x0A	

4.19 MD_SCAN_IMAGE_X

This command is used to get a fingerprint image from the module. After this command is transmitted, a user needs to press a finger on the module reader. After successfully scanning the fingerprint, the module sends the fingerprint image to the host over the extended transmission protocol.

The data structure of a fingerprint image is as follows:

```
typedef struct _Image_
{
    int width;          /*  Image width                     */
    int height;         /*  Image height                    */
    int compressed;     /*  This value is not used currently.   */
    int encrypted;      /*  Encrypted image, which is not used currently. */
    int binary;         /*  Data format of the image, which is not used
currently. The default value is 0.*/
    int img_len;        /*  Image length                    */
    int template_len;   /*  Length of the fingerprint template generated using
the image, which is not used currently.     */
    };
```

The request command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x83	
Param	Data Size	Data packet size, 4 KB by default
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

The response command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x83	
Param	NULL	
Size	Image size	
Error	0x61 0x6C 0x6D 0x63 0x81	SUCCESS TIME_OUT MEM_FULL MD_FAIL CANCELED
Checksum	#	
End code	0x0A	

Note：to allocating a buffer zone in size of 400*400 for storage when receive the FP image by Host.

4.20 MD_SET_TIME

This command is used to write the current time to the module. The clock of the fingerprint module must be set by the host as the fingerprint module has no clock chip**. The host can set the module time every other hour.**

The year in Data **is the current year minus 2000. For example, if the current year is 2008, the year value in** Data **is 8.**

The request command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x4A	
Param	(DD<<16) | (MM<<8) | YY	Date
Size	8	Data size + verification data size
Flag	NULL	
Checksum	#	
End code	0x0A	
Data	(ss<<16) | (mm<<8) | hh	Time
Checksum of data	#	Checksum of data

The response command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x4A	
Param	NULL	
Size	NULL	
Error	0x61 0x63	SUCCESS FAIL
Checksum	#	
End code	0x0A	

4.21 MD_GET_TIME

This command is used to read the current module time.

The request command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x4B	
Param	NULL	
Size	NULL	

Flag	NULL	
Checksum	#	
End code	0x0A	

The response command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x4B	
Param	(DD<<16) | (MM<<8) | YY	Date
Size	8 NULL	8 – data size when reading is successful. NULL – the reading fails.
Error	0x61 0x63	SUCCESS FAIL
Checksum	#	
End code	0x0A	
Data	ss<<16 | mm<<8 | hh	Time
Checksum of data	#	

Note: The year must be plus 2000 when the date is displayed in the format of YYYY/MM/DD.

4.22 MD_DEL_ALOG

This command is used to delete all the verification logs from the module.

The request command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x9E	
Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

The response command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0x9E	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
Checksum	#	
End code	0x0A	

4.23 MD_LOAD_LOG_X

This command is used to read all verification logs in the module over the extended transmission protocol. Verification logs are stored as specific data structures in the form of files in the module.

```
Typedef struct    _Log_{
                BYTE    Event;
                BYTE    verified;
                U32     Date;
                U32     Time;
                U16     userID;
                Char    Reserved[4];
};
```

Data Field	Number of Bytes	Description
Event	1 byte	Stores events occurring during recording. See the following event table.
Verified	1 byte	Verification mode, fingerprint mode or other modes. This field can be ignored and it will be used in later versions of the module.
Date	4 byte	YY MM DD
Time	4 byte	hh mm ss
User ID	2 byte	User ID

Data Field	Number of Bytes	Description
Reserved	**4 byte**	**This field can be read or set by using the** MD_LC **command.**

Note:

Year is YY **plus 2000. For example, if the year is 2008, the value of** YY **is** 8**. If the module time is not set, the value of** YY **is** 0 **and** Time **is an incorrect value.**

Code table of recorded events:

Event	Code	Description
LOG_FREE	**0**	**Verification logs generated using the** Free Scan **command**
LOG_SV	**1**	**Verification logs stored when the** Scan Verify **command is used**
LOG_TV	**2**	**Verification logs stored when the** template verify **command is used**
LOG_IV	**3**	**Verification logs stored when the** image verify **command is used**
LOG_SI	**4**	**Verification logs stored when the** scan identify **command is used**
LOG_TI	**5**	**Verification logs stored when the** template identify **command is used**
LOG_II	**6**	**Verification logs stored when the** image identify **command is used**

The request command is as follows:

Data Field	Value	Description
Start code	**0x70**	
Command	**0xA4**	
Param	**Data size**	**Data size, 4 KB at the maximum**
Size	**NULL**	
Flag	**NULL**	
Checksum	**#**	
End code	**0x0A**	

The response command is as follows:

Data Field	Value	Description

Data Field	Value	Description
Start code	**0x70**	
Command	**0xA4**	
Param	**NULL**	

Size	NULL	
Error	0x61 0x63 0x6C 0x69 0x82	SUCCESS FAIL TIME_OUT NOT_FIND DATA_ERROR
Checksum	#	
End code	0x0A	

4.24 MD_LOAD_USER_X

This command is used to read all user data stored as user structures in the fingerprint module and to send the user structure of each user to the host over the extended transmission protocol.

The request command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0xA0	
Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

The response command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0xA0	
Param	NULL	
Size	NULL	
Error	0x61 0x6C 0x63 0x69 0x82	SUCCESS TIME_OUT FAIL NOT_FOUND DATA_ERROR
Checksum	#	

Data Field	Value	Description
End code	**0x0A**	

4.25 MD_LOAD_TMP_X

This command is used to read all fingerprint templates stored as fingerprint template structures in the fingerprint module and send relevant data in each fingerprint template structure to the host in a certain cycle.

```
typedef struct _Template_
{
    U16    Size;          /* Actual length of a fingerprint template          */
    U16    ID             /* User ID, that is, user PIN                       */
    BYTE   FingerID;      /* Fingerprint ID (Sub_ID or Finger ID)             */
    BYTE   Valid;         /* Whether a fingerprint template is valid. 1: valid; 0: invalid */
    BYTE   Template[1664]; /* Fingerprint template data, with the maximum length of 1664 bytes */
};
```

Note:

The preceding structure uses the 1-byte memory alignment mode.

The actual fingerprint template length is not fixed and varies with the finger enrolled.

In the transmission of a fingerprint template, in order to speed it up, the length of transmitted fingerprint data is not the length of the fingerprint template structure but Size+ID+FingerID+Valid+Template (actual length)**.**

For example, when the following two fingerprint templates need to be transmitted:

Fingerprint template 1: Size=1024, ID=1, Finger ID=0, Valid=1, Template=(1024-byte template data)

Fingerprint template 2: Size=1660, ID=2, Finger ID=1, Valid=1, Template=(1660-byte template data)

When the ReadAllTemplates command is used to transmit fingerprint template data, the transmitted data packets in hexadecimal notation are as follows:

00 04 01 00 00 01 (1024-byte template data of fingerprint 1) 7C 06 02 00 01 01 (1660-byte template data of fingerprint 2)

The total length of transmitted fingerprint template data is: (6 bytes + 1024 bytes) + (6 bytes + 1660 bytes) = 2696 bytes.

Note:

A very long time is required if many fingerprint templates need to be transmitted. See the precautions section in this document.

The request command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0xA2	
Param	Packet size	Size of the transmitted data packet
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

The response command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0xA2	
Param	NULL	
Size	NULL	
Error	0x61 0x6C 0x63 0x69 0x82	SUCCESS TIME_OUT FAIL NOT_FOUND DATA_ERROR
Checksum	#	
End code	0x0A	

4.26 MD_DEL_DB

This command is used to delete all the data from the fingerprint module, including user data, fingerprint template data, and verification log data.

The request command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0xF8	
Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

The response command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0xF8	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
Checksum	#	
End code	0x0A	

4.27 MD_SCAN_TEMPLATE

This command is used to scan a fingerprint template. After a user presses a finger on the fingerprint reader, the host reads the fingerprint template from the module. The maximum length of a fingerprint template is 2 KB. Before this command is executed, the value of 0x50 **in the command needs to be changed to** 0x30**. Then, the module waits for a user to press a finger. The host needs to always call this command to read fingerprint templates.**

Note: **The module supports two fingerprint identification methods: The first method is identifying fingerprint directly in the module; the second method is identifying fingerprint on the host after finger templates are generated in the module.**

The request command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0xFC	
Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	
nd code	0x0A	

The response command is as follows:

Data Field	Value	Description
Start code	0x70	
Command	0xFC	
Param	NULL	
Size	size NULL	Size – length of a fingerprint template, 2 KB at the maximum NULL – no fingerprint template
Error	0x61	SUCCESS
Checksum	#	
End code	0x0A	
Data	Template	Template data
Checksum of data	#	

Note: **The module always responds with success of this command.**

4.28 MD_DISCONNECT

Disconnection. **Module supports both 232-TTL communication and USB communication. Module only can communicate with the Host by one communication, so when changing the communication type, you have to disconnect the old one or else the new one can't be connected.**

Command requesting:

Data field	Numerical value	Description
Start code	0x70	
Command	0xCC	
Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

Command answering

Data field	Numerical value	Description
Start code	0x70	
Command	0xCC	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
Checksum	#	
End code	0x0A	

4.29 MD_RESET

Module reset.

Command requesting:

Data field	Numerical value	Description
Start code	0x70	
Command	0xD0	
Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

Command answering:

Data field	Numerical value	Description
Start code	0x70	

Command	0xD0	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
Checksum	#	
End code	0x0A	

4.30 MD_ENABLEDEVICE

Enable device**. When this command is called, the module will send Host the FP identification result under authentication mode.**

Command requesting:

Data field	Numerical value	description
Start code	0x70	
Command	0xFB	
Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

Command answering:

Data field	Numerical value	description
Start code	0x70	
Command	0xFB	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
Checksum	#	
End code	0x0A	

4.31 MD_DISABLEDEVICE

Disable device. When this command is called, the module will not send Host the FP identification result under authentication mode.

Command requesting:

Data field	Numerical value	description
Start code	**0x70**	
Command	**0xFA**	
Param	**NULL**	
Size	**NULL**	
Flag	**NULL**	
Checksum	**#**	
End code	**0x0A**	

Command answering:

Data field	Numerical value	description
Start code	**0x70**	
Command	**0xFA**	
Param	**NULL**	
Size	**NULL**	
Error	**0x61**	**SUCCESS**
Checksum	**#**	
End code	**0x0A**	

4.32 MD_UPDATE_FW

Firmware updating**. In the updating process, sending the size of firmware to module, host will send the data in extend data format to module when answering YES.**

Command requesting:

Data field	Numerical value	Description
Start code	**0x70**	
Command	**0x72**	
Param	**firmware size**	**Size of firmware data**
Size	**NULL**	
Flag	**NULL**	
Checksum	**#**	
End code	**0x0A**	

Command answering:

Data field	Numerical value	description
Start code	**0x70**	
Command	**0x72**	
Param	**NULL**	
Size	**NULL**	
Error	**0x61** **0x63** **0x6C**	**SUCCESS** **FAIL** **TIME_OUT**
Checksum	**#**	
End code	**0x0A**	

4.33 MD_WT_FILE_X

This command is used to upload the fingerprint template storage file. The size of the fingerprint template storage file is first sent to the module during writing. After receiving the confirmation from the module, the host will transfer data in the extensible data format.

Note: If the user ID in a fingerprint template is inconsistent with the user ID in the user data, the user data does not match the fingerprint template data and an error will occur during fingerprint verification. Therefore, fingerprint template data transferred must be consistent with user data.

The format of the request command formats is shown below:

Data Field	Value	Description
Start code	0x70	
Command	0xAA	
Param	templates size	Size of the fingerprint template storage file
Size	NULL	

Flag	NULL	
Checksum	#	
End code	0x0A	

Response command

Data Field	Value	Description
Start code	0x70	
Command	0xAA	
Param	NULL	
Size	NULL	
Error	0x61 0x63 0x6C	SUCCESS FAIL TIME_OUT
Checksum	#	
End code	0x0A	

4.34 MD_WU_FILE_X

This command is used to upload the user information storage file. The size of the user information storage file is first sent to the module during writing. After receiving the confirmation from the module, the host will transfer data in the extensible data format.

Note: If the user ID in a fingerprint template is inconsistent with the user ID in user data, the user data does not match the fingerprint template data and an error will occur during fingerprint verification. Therefore, fingerprint template data transferred must be consistent with user data.

The format of the request command is shown below:

Data Field	Value	Description
Start code	0x70	
Command	0x42	
Param	Data size	User information storage file
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

Response command

Data Field	Value	Description
Start code	0x70	
Command	0x42	
Param	NULL	
Size	NULL	
Error	0x61 0x63 0x6C	SUCCESS FAIL TIME_OUT
Checksum	#	
End code	0x0A	

4.35 MD_RT_FILE_X

This command is used to download the template storage file in the module to the host. The data will be transferred via an extensible protocol during download.

The format of the request command is shown below:

Data Field	Value	Description
Start code	0x70	
Command	0xAB	
Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

Response command

Data Field	Value	Description
Start code	0x70	
Command	0xAB	
Param	NULL	
Size	NULL	
Error	0x61 0x63 0x6C	SUCCESS FAIL TIME_OUT

Checksum	#	
End code	0x0A	

4.36 MD_RU_FILE_X

This command is used to download the user information storage file in the module to the host. The data will be transferred via an extensible protocol during download.

The format of the request command is shown below:

Data Field	Value	Description
Start code	0x70	
Command	0x43	
Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

Response command

Data Field	Value	Description
Start code	0x70	
Command	0x43	
Param	NULL	
Size	NULL	

Error	0x61 0x63 0x6C	SUCCESS FAIL TIME_OUT
Checksum	#	
End code	0x0A	

4.37 MD_IDENTIFY_FREE

This command is used to identify fingerprint based on the fingerprint template comparison result in the free scan mode. The module sends the real-time comparison data to the host.

Note: The host only receives data rather than transfer commands.

Response command

Data Field	Value	Description
Start code	0x70	
Command	0x2F	
Param	UserID/0x6A	0x61: ID of a user with a successful fingerprint comparison result 0x63: No user with the matched fingerprint is found.
Size	NULL	
Error	0x61 0x63	SUCCESS FAIL
Checksum	#	

End code	0x0A	

5 Precautions

5.1 Data Packet

By default, the module adopts data packets represented in hexadecimal **notation.**

The maximum length of a data packet transmitted by the serial port cannot be larger than 4 KB**.**

5.2 User ID

A user ID, that is, PIN, is a 2-byte **value of the unsigned short data type ranging from** 1 **to** 65534**. The maximum user capacity supported by the module cannot be larger than** 65534**.**

5.3 Fingerprint Template Length

The maximum fingerprint template length is 2048 **bytes. The actual length of each fingerprint template cannot be larger than the maximum value.**

The lengths of fingerprint templates enrolled using different fingers or the same finger may be different, but the actual length of a fingerprint template is definitely smaller than the maximum fingerprint template length.

The value of Size **in a fingerprint template structure is the actual length of the template in the structure.**

5.4 Verification Logs

By default, the module does not store verification logs. To save verification logs, enable the system parameter "save verification log" (parameter ID: 0x36**).**

Note**: The module does not have a clock chip and the accuracy of its internal RTC clock is not high. The time is changed to zero upon each system restart. Therefore, after the parameter "save verification log" (parameter ID:** 0x36**) is enabled, ensure that the clock is synchronized after the module is started or before the module works normally. In this case, stored verification logs are meaningful. Otherwise, stored verification logs are meaningless. The module clock needs to be synchronized every other hour during working.**

5.5 Time Consumption of Commands

Most commands take a very short time in processing. Nevertheless, if an operation involves a fingerprint template file or user data file, the required time varies with the quantities of users and fingerprint templates, which is similar to the power-on and startup duration.

Commands requiring a long time are as follows:

- MD_ENROLL_IMAGE_X
- MD_SCAN_IMAGE_X
- MD_READ_TMP_X
- MD_LOAD_TMP_X
- MD_LOAD_USER_X
- MD_IDENTIFY_IMAGE_X
- MD_WT_FILE_X
- MD_WU_FILE_X
- MD_RT_FILE_X
- MD_RU_FILE_X

5.6 Deletion Command

Whenever a user is deleted, the user's fingerprint templates are also deleted. However, when a user's fingerprint is deleted, the user may not be deleted. For example, if a user has multiple fingerprint templates and one or more fingerprint templates are deleted, the user is not deleted if the user still has fingerprint templates.

6 Appendix

6.1 Data Transmission Reference Document

Refer to SilkID Command Description.doc.