Fingerprint Identification Module User's Guide for Communication Protocol Stack B

(Ver 3.0)

Beijing IDWorld Technology Development Co., Ltd.

May 2020

Third Edition

Contents

1. Abstracts

This article describes the serial port parameters, communication process and communication command packet/data packet format of IDWorld Company's fingerprint identification module.

1.1 Serial Port Parameters, Communication Process, Transmission order in frame sequence

The parameters used for serial communication are as follows:

Start bit: 1 位（1bit）

Data bits: 8 位 （8bit）

Stop bit: 1 位（1bit）

Parity：No

Baud rate: 9600/19200/38400/57600/115200/230400/460800/921600 ， default value: 115200BPS

Communication Process

The sending and receiving of all commands must follow the principle of one sending and one receiving.

When the host (Host) does not receive a reply, it is not allowed to send commands to the target module (TARGET).

Transmission sequence in communication frame

Byte follows the rules LSB first transfer.

Word follows the rule of low byte first high byte transmitted later.

1.2 Main functions

Not only has the following general functions;

Fingerprint input： According to different models, the maximum fingerprint capacity that can be entered is 40/80/100/170/200/400/500/1000/1700/3000.

Fingerprint deletion: Fingerprint deletion: single deletion/specified number range deletion.

Fingerprint verification: Verify, 1:1 comparison

Fingerprint identification: Identify，1:N Search（within 1 - maximum capacity, the search range can be arbitrarily specified）

Also it has the following functions;

1. Upload/Download fingerprint template data

   Upload fingerprint template data to the host/download fingerprint template data to the module
2. Upload/Download fingerprint image

   Upload fingerprint image to host/download fingerprint image to module (fingerprint feature extraction/enroll/verify/identify with live fingerprint)
3. Check whether the fingerprint template data in the specified number range is damaged
4. Read the total number of registered users and user list in the module
5. Set/read fingerprint module parameters (security level, allow/prohibit self-learning, allow/prohibit fingerprint repeat check, baud rate, etc.)

6. Set/read the serial number of the fingerprint module
7. On-board firmware update
8. Read the software and hardware information of the fingerprint module information (firmware type and version, algorithm chip, fingerprint sensor) and other modules.

1.3 Description of obtained fingerprint module information

Use read module information command (CMD_DEVICE_INFO)

Or the information obtained by the "Get Module Information" button in the Demo program is as follows：

Module information composition rules：

ID_Protocol stack and algorithm name_ Algorithm chip model_ Collector model_ None or module hardware name_None or Inner (Fingerprint Capacity) Version number

ID: Indicates provided by IDWorld company

Communication protocol stack (consisting of at least 4 letters) and fingerprint identification algorithm name (none or a number)

The communication protocol stack supports the following:

SEOU: IDWORLD Communication Protocol A（Abbreviation：Old Protocol）

SEONU: IDWORLD Communication Protocol B（Abbreviation：New Protocol），Communication protocol document described in this manual

SEODU: IDWORLD Communication protocol A+B both protocol are supported（abbreviation：ID Dual Protocol）

SYNONU:IDWORLD Protocol B+SY protocol（abbreviation：ID+SY Dual Protocol）

Fingerprint identification algorithm names are classified as follows:

None: Conventional general algorithm, applicable to all fingerprint sensors, fingerprint template data size: 496 bytes

2: Referred to as the 160 new algorithm, it only supports the fingerprint sensor with 160*160 pixels, and the fingerprint template data size: 1008 bytes

5: Non-template algorithm, only supports FPC1021/1025 fingerprint sensor, fingerprint template data size: 2024 bytes

7: Referred to as 192 new algorithm, only supports 192*192 pixel fingerprint sensor, fingerprint template data size: 448 byte

Algorithm chip model：QS808/ID808/ID809/ID811

ID808: Cotex-M4 core，144Mhz main frequency，96K RAM+1MB FLASH，QFP48 Package(9mm*9mm)

ID809: Cotex-M4 core，144Mhz main frequency，128K RAM+1MB FLASH，QFN36 Package(6*6)

ID811: Cotex-M4 core，150/120Mhz main frequency，128K RAM+512KB FLASH, Built-in national secret algorithm，QFN32 Package(4*4)

Collector Model: The mainstream fingerprint sensors in the market (optical, various area/swipe capacitive sensors) almost all have corresponding firmware support.

Fingerprint capacity:(The number in brackets indicates the maximum fingerprint capacity of this module)

Notes:

This parameter is very important when using the module. If you are not sure, please consult our sales, or read the fingerprint capacity through commands or demo programs.

1. The ID number range specified in the identification command (CMD_SEARCH) cannot exceed the fingerprint capacity, otherwise the parameter error will be returned.

2. When using our demo program, choose the fingerprint capacity that matches this module, otherwise, the parameter error will be prompted during continuous identification. Example of acquired module information description:

Module information	Information description
ID_SEONU_ID809_GC0308_DORLO(3000fp) V1.0	Protocol stack: IDWORLD Command Set B，Algorithm: General collector old algorithm Algorithm Chip:ID809；Collector：Optical（GC0308） Module Model: DORLO；Fingerprint Capacity: 3000， Version Number: V1.0
ID_SEODU2_ID809_BF5325_PDI48_Inner(200fp) V1.1	Protocol stack: Dual IDWORLD Command Set A/B， Algorithm:160*160 new algorithm Algorithm Chip: ID809, Collector: BF5325 Module Model: PDI48，Fingerprint Capacity： 200， Version Number: V1.1
ID_SYNONU5_ID811A1_FPC1021_Inner(40fp) V1.3	Protocol stack: IDWORLD Protocol: Algorithm: FPC1021 Dedicated Algorithm Algorithm Chip: ID811,Collector: FPC0121；Fingerprint capacity: 40
ID_SYNONU7_ID808_FPC1020_Inner(500fp) V1.0	Protocol stack: IDWORLD Protocol: Algorithm: 192*192 New algorithm Algorithm Chip: ID811；Collector: FPC1020; Fingerprint Capacity: 500

1.4 The difference between different algorithm named in firmware

Algorithm name Compare items	Common algorithm for all collectors ex：SEONU	only applicable to 160*160 Collector's algorithm ex：SEONU2	only applicable to 192*192 Collector's Algorithm ex：SEONU7	FPC1021/1025 Collector's dedicated algorithm ex：SEONU5
Fingerprint template data size	498 byte	1008 byte	448 byte	2024 byte
Fingerprint duplicated verify function	Support	not support	Support	not support
Download the template and search and compare the fingerprint library in the module	support	Not support	support	Not support
Supported fingerprint collector	Any size	Any 160*160 pixel	Any 192*192 pixel	Only FPC1021/1025

Recommended collector	Optical, A288/A360 TCS1/2,FPC1011	BF5325/6632/5838 A176,FPC1021/1025	BF5333/5836 A192、TL192、 FPC1020/1024	FPC1021/1025

1.5 Quick Start for Application Development

Application developers can quickly complete the application development of the module by referring to the content of this chapter and related links.

1. **When the fingerprint capacity and algorithm version of the module are unknown:** Please use the command CMD_DEVICE_INFO (see §4.4 and §1.3 chapters)
2. **Fingerprint Enroll:** Refer to the enroll process in §6.1, and each commands used in each process step is attached, that is, execute the following commands: After executing {Capture fingerprint image(CMD_GET_IMAGE) ->Convert into template(CMD_GENERATE)} 3 times -> Merge fingerprint template(CMD_MERGE) ->Save fingerprint template (CMD_STORE_CHAR)
3. **Fingerprint Verify/Identify:** Refer to the verify/identify process in §6.2, that is to execute the following commands:

Capture fingerprint image CMD_GET_IMAGE -> Generate template CMD_GENERATE -> Verify CMD_VERIFY/ or Identify CMD_SEARCH

The chapter list corresponding to each command used above is as follows:

Command Name	Command Code	Reference Section	Description
CMD_GET_IMAGE	0x0020	§4.6	Capture fingerprint image and temporarily store it in ImageBuffer
CMD_GENERATE	0x0060	§4.20	From image in ImageBuffer generate template and temporarily store RamBuffer0/1/2
CMD_MERGE	0x0061	§4.21	Merge 3 templates temporarily stored in RamBuffer0/1/2 to generate a template to be used temporarily stored in the specified Ram Buffer.
CMD_STORE_CHAR	0x0040	§4.11	Save template data in Ram Buffer to the space of specified number in template library
CMD_SEARCH	0x0063	§4.23	Search and compare the template in the RamBuffer with the specified number in the fingerprint library. The search range is set by this command (for example, within the range of 1 - full capacity), and obtain the search result.
CMD_VERIFY	0x0064	§4.24	Compare the template in the RamBuffer of the specified number with a template set by this command in the fingerprint library and get the comparison result.

Table1.5: Commands list that, must be use for quick start of application development

Notes: For the result code and error code in the response packet after the command is executed, please refer to the response and error code table in Chapter 5.

Application developers can also refer to the test program source code provided by our company (NOEMHost_v3.18_20191230.7z).

Or directly use a serial port monitoring program (for example: monitoring software such as Bus Hound or AccessPort) to monitor the communication process between the fingerprint module and the test program, and get the command receiving/sending result, and quickly understand this manual according to the detailed description of the corresponding command in this manual.

2. Communication Protocol（Stack B）- Overview

After the module is powered on, the firmware BOOT needs time (i.e. hardware and algorithm initialization time), and the host must wait for the module to complete initialization before sending commands to the module

After the fingerprint module is powered on, the MCU's GPIO and UART ports are initialized successfully, and a byte of 0x55 will be sent through the UART as a handshake signal to notify the host.

When the host is waiting for the module to initialize after controlling the fingerprint module to be powered on, it can enter the working state in advance by receiving this handshake signal.

Notes: After the host control module is powered on, the communication process can be started by the following two methods.

1. **After the host receives the handshake signal 0x55 of the module, it can start sending commands to the module.**
2. **After the host control module is powered on, it can start to send commands to the module after a delay of 280ms.**

2.1 Communication process

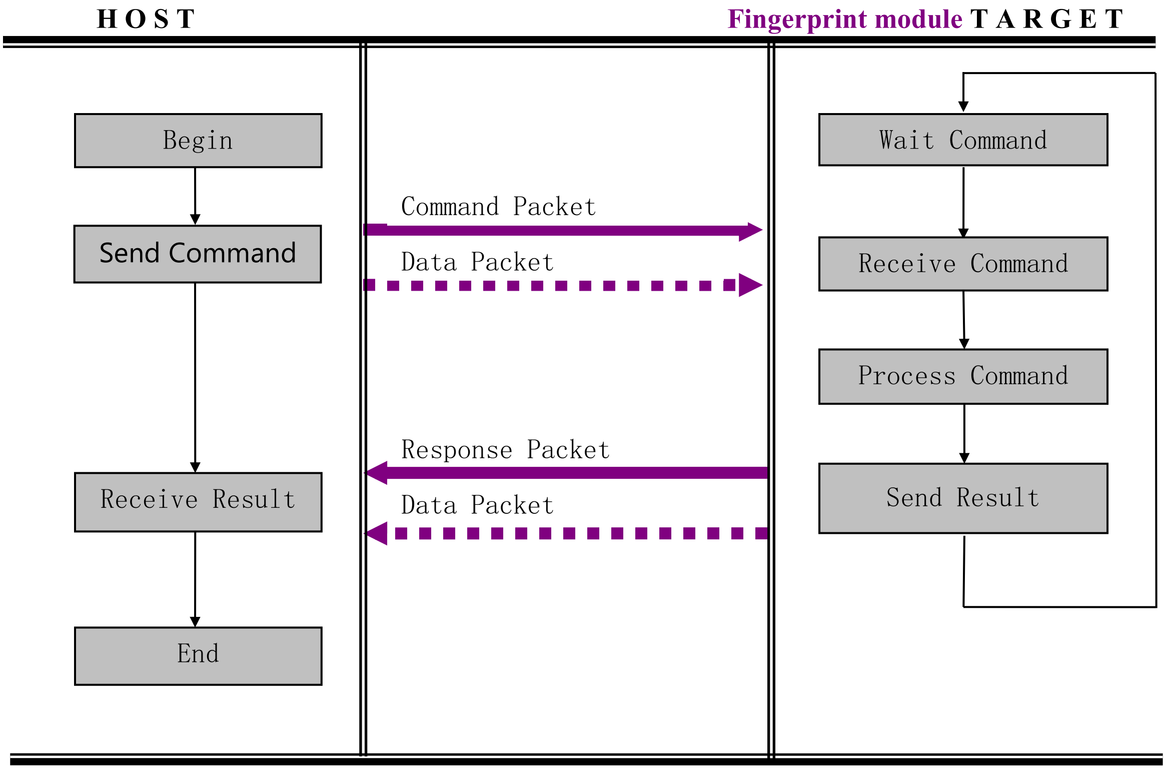

Fig 2-1. The communication process of protocol stack B

Notes:

In the communication process, the sending and receiving of all commands must follow the principle of one sending and one receiving.
Host cannot send commands to TARGET when it has not received the response from the module.

2.2 Classification of communication packet

2.2.1 Command packet

- The command packet describes the contents of the command from Host to Target.
- All commands sent from Host are transmitted through Command packet.
- The frame length of Command packet is **26 bytes**.

2.2.2 Response packet

- Response packet is to the response contents from Target to Host.
- All commands terminate their mission after receiving the corresponding processing result, which is Response packet.
- The length of the Response packet is **26 bytes**.

2.2.3 Data Packet of command/response

- When the length of the command parameter or response data is greater than 16 bytes, use the command/response data packet to transmit data.
- Before sending the command packet, the Host must inform the module Target of the length of the data packet by using the Command packet
- The maximum length of command parameters or corresponding data packets is 500 bytes.

2.3 Frame structure of communication packet

2.3.1 Communication packet identification code

The first 2byte of the communication packet is the identification code indicating the type of the communication packet, as shown in Table 2-1:

Table 2-1. Packet identification code

Packet type	Code packet type identifier
Command packet	0xAA55
Response packet	0x55AA
Command Data Packet	0xA55A
Response Data Packet	0x5AA5

2.3.2 Frame structure of command packet

PREFIX		SID	DID	CMD		LEN		DATA				CKS	
0x55	0xAA	源 ID	目标 ID	L	H	L	H	D0	D1	…	D15	L	H
0	1	2	3	4	5	6	7	8	9	…	23	24	25

Table 2-2. Structure of Command packet

OFFSET	FIELD	TYPE	SIZE	DESCRYPTION
0	PREFIX	WORD	2byte	Packet Identify code
2	SID	BYTE	1byte	Source Device ID
3	DID	BYTE	1byte	Destination Device ID
4	CMD	WORD	2byte	Command Code
6	LEN	WORD	2byte (=n, n < 16)	Length of DATA
8	DATA	Byte Array	16byte	Command Parameter (The actual data is n byte)
24	CKS	WORD	2byte	Check Sum: The lowest 2 bytes after arithmetic and operation of all data from PREFIX to DATA field

2.3.3 Frame structure of Response packet

PREFIX		SID	DID	RCM		LEN		RET		DATA				CKS	
0x55	0xAA	Source ID	Destination ID	L	H	L	H	L	H	D0	D1	…	D13	L	H
0	1	2	3	4	5	6	7	8	9	10	11	…	23	24	25

Table 2-3. Structure of Response packet

OFFSET	FIELD	TYPE	SIZE	DESCRYPTION
0	PREFIX	WORD	2byte	Packet Identify code
2	SID	BYTE	1byte	Source Device ID
3	DID	BYTE	1byte	Destination Device ID
4	RCM	WORD	2byte	Response Code
6	LEN	WORD	2byte(=n, n < 16)	Length of RET and DATA
8	RET	WORD	2byte	Result Code(0: Success, 1: Failure)
10	DATA	Byte Array	14byte	Response Data (Actually n-2 byte)
24	CKS	WORD	2byte	Check Sum: The lowest 2 bytes after arithmetic sum of all data from PREFIX to DATA field

2.3.4 Frame structure of Command Data Packet

PREFIX		SID	DID	CMD		LEN		DATA				CKS	
0x5A	0xA5	Source ID	Destination ID	L	H	L	H	D0	D1	…	Dn-1	L	H
0	1	2	3	4	5	6	7	8	9	…	8+n-1	8+n	8+n+1

Table 2-4. Structure of Command Data Packet

OFFSET	FIELD	TYPE	SIZE	DESCRYPTION
0	PREFIX	WORD	2byte	Packet Identify code
2	SID	BYTE	1byte	Source Device ID
3	DID	BYTE	1byte	Destination Device ID

4	CMD	WORD	2byte	Command Code
6	LEN	WORD	2byte(=n , n <500)	Length of DATA
8	DATA	Byte Array	nbyte	Command parameter
8+n	CKS	WORD	2byte	Check Sum: The lowest 2 bytes after arithmetic sum of all data from PREFIX to DATA field

The Host must transmit the command packet before sending the command data packet, so that the module Target enters the waiting state for command data packet reception.

In the DATA field of the command packet, the length of the command data packet to be transmitted must be set.

The Host shall transmit the Command Data Packet after confirming that the Target is in the waiting state for receiving the command packet.

2.3.5 Frame structure of Response data packet

PREFIX		SID	DID	RCM		LEN		RET		DATA				CKS	
0xA5	0x5A	Source ID	Destination ID	L	H	L	H	L	H	D0	D1	…	Dn-3	L	H
0	1	2	3	4	5	6	7	8	9	10	11	…	8+n-1	8+n	8+n+1

Table 2-5. Structure of Response Data Packet:

OFFSET	FIELD	TYPE	SIZE	DESCRYPTION
0	PREFIX	WORD	2byte	Packet Identify code
2	SID	BYTE	1byte	Source Device ID
3	DID	BYTE	1byte	Destination Device ID
4	CMD	WORD	2byte	Response Code
6	LEN	WORD	2byte(=n, n <500)	Length of result data (RET + DATA)
8	RET	WORD	2byte	Result code(0 : success, 1 : failure)
10	DATA	Byte Array	n-2 byte	Response data
8+n	CKS	WORD	2byte	Check Sum： The lowest 2 bytes after arithmetic sum of all data from PREFIX to DATA field

Notes: When transmitting more than 14 bytes of data from the Target to the Host, a response data packet is required

3. Introduction to Communication Command

3.1 Some concepts used in communication commands

The command execution needs to use the ImageBuffer to temporarily stores the fingerprint image and the RamBuffer which temporarily stores the fingerprint template data.

ImageBuffer: Used to temporarily store fingerprint images.

RamBuffer: Used to temporarily store fingerprint feature template data.

The module has opened three RamBuffer: RamBuffer0/RamBuffer1/RamBuffer2 。

Notes: In the event of a power failure, the data temporarily stored in ImageBuffer and RamBuffer will be lost.

TemplateLibrary: A collection of fingerprint templates stored in FLASH Memory,

The fingerprint template data can be written/read through CMD_STORE_CHAR /CMD_LOAD_CHAR。

3.2 Fingerprint Template Record generated by each version of the algorithm

Different versions of the algorithm generated fingerprint template data length is not the same, in the call upload fingerprint template (CMD_UP_CHAR)/download fingerprint template (CMD_DOWN_CHAR) command, attention should be paid to the difference in the length of fingerprint template data generated by each version of the algorithm (see the table below).

The structure of the fingerprint template data is as follows: fingerprint feature data (Feature Data) + 2-byte checksum (CheckSum)

Notes: CheckSum is the lowest 2 bytes after the arithmetic sum of each byte of Feature Data.

Table 3-2. Data length of the template record generated by each version of the algorithm

Algorithm's version	Length of template data
General algorithm, such as firmware SEODU	498 byte
160 new algorithm, such as firmware SEONU2	1008 byte
192 new algorithm, such as firmware SYNONU7	448 byte
Non-standard algorithm, such as firmware SYNONU5	2024 byte

3.3 Command List

No.	Command Name	Code	Function
1	CMD_TEST_CONNECTION	0x0001	Test command for communication connection with module
2	CMD_SET_PARAM	0x0002	Set the following parameters of the module， Notes: TimeOut is only available for swipe collectors （Device ID, Security Level, Baud rate, Duplication Check, Auto Learn， TimeOut）
3	CMD_GET_PARAM	0x0003	Get the following parameters of the module: (Device ID, Security Level, Baud rate, Duplication Check, Auto Learn， TimeOut)
4	CMD_GET_DEVICE_INFO	0x0004	Get module device information
5	CMD_ENTER_IAP_MODE	0x0005	Set the module to IAP (program in application) state
6	CMD_GET_IMAGE	0x0020	Collect fingerprint image from the collector and temporarily store them in ImageBuffer

No.	Command Name	Code	Function
7	CMD_FINGER_DETECT	0x0021	Check fingerprint input status (with/without finger)
8	CMD_UP_IMAGE	0x0022	Upload the fingerprint image stored in ImageBuffer to the HOST
9	CMD_DOWN_IMAGE	0x0023	Download the fingerprint image from the host (HOST) to the ImageBuffer of the module
10	CMD_SLED_CTRL	0x0024	Control the on/off of the backlight of the optical collector, or the LED light to prompt commands
11	CMD_STORE_CHAR	0x0040	Save the template data in the specified RamBuffer to the address of template of specified number in fingerprint library.
12	CMD_LOAD_CHAR	0x0041	Read the template data of the specified number and temporarily store it in the RamBuffer of the specified number (0/1/2).
13	CMD_UP_CHAR	0x0042	Upload the fingerprint template data temporarily stored in the specified RamBuffer to HOST.
14	CMD_DOWN_CHAR	0x0043	Download a fingerprint template from HOST to the specified RamBuffer of the module
15	CMD_DEL_CHAR	0x0044	Delete all fingerprint templates in the specified number range in the fingerprint library.
16	CMD_GET_EMPTY_ID	0x0045	Get the first template number that can be enrolled (not enrolled) in the specified range.
17	CMD_GET_STATUS	0x0046	Get the template enroll status with the specified number.
18	CMD_GET_BROKEN_ID	0x0047	Check whether all fingerprint templates in the specified number range are damaged.
19	CMD_GET_ENROLL_COUNT	0x0048	Get the number of enrolled templates in the specified number range.
20	CMD_GENERATE	0x0060	Generate template data from the fingerprint image temporarily stored in ImageBuffer and save it in the specified RamBuffer.
21	CMD_MERGE	0x0061	Merge two or three template data temporarily stored in RamBuffer into one template data Notes: The merged template data is temporarily stored in RamBuffer 0
22	CMD_MATCH	0x0062	1:1 comparison (match) between two fingerprint templates in the specified RamBuffer.
23	CMD_SEARCH	0x0063	1:N search (Identify) between the template in the specified RamBuffer and all the templates in the specified number range in the fingerprint library.
24	CMD_VERIFY	0x0064	1:1 comparison (verify) between the fingerprint template in the specified Ram Buffer and the fingerprint template with the specified number in the fingerprint library
25	CMD_SET_MODULE_SN	0x0008	Set the module serial number information (Module SN) in the device.
26	CMD_GET_MODULE_SN	0x0009	Get the module serial number (Module SN) of this device.
27	CMD_ADJUST_SENSOR	0x0025	Adjust the fingerprint collector parameters. Notes: Some models of modules do not support this command.

No.	Command Name	Code	Function
28	CMD_GET_ENROLLED_ID_LIST	0x0049	Get a list of enrolled User IDs
29	CMD_ENTER_STANDY_STATE	0x000C	Put the module into sleep state. Notes: Some models of modules do not support sleep function.

4. Command Reference

4.1 Check Connection (CMD_TEST_CONNECTION 0x0001)

[Function]

This command is used to check the status of connection between target and host. The host needs to send this command first to check the connection status with target. If it is unsuccessful, it can be considered that the connection with the target is not normal, or the target is not working properly, or the serial port baud rate is incorrect.

[Sequence]

If the connection is normal, ERR_SUCCESS will be returned.

[Command and Response]

Table 4-1. Command CMD_TEST_CONNECTION

PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID
CMD	0x0001
LEN	0
DATA	No Data
PREFIX	0x55AA
SID	Source Device ID
DID	Destination Device ID
RCM	0x0001
LEN	2
RET	Result Code (ERR_SUCCESS/ ERR_FAIL)
DATA	No Data

Example 4.1- The host sends the command CMD_TEST_CONNECTION and the module's response.

Host command: 55 AA 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01

Module response: AA 55 01 00 01 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 01

4.2 Set Parameters (CMD_SET_PARAM 0x0002)

[Function]

This command sets the value of the device parameters(Device ID, Security Level, Baud rate, Duplication Check, Auto Learn， FP Timeout) according to specified parameter type, and returns the result.

[Sequence]

① If the specified parameter type is invalid，ERR_INVALID_PARAM will be returned.

② If the specified parameter value is invalid，ERR_INVALID_PARAM will be returned.

③ According to the parameter type，sets the parameter value and returns the result.

[Command and Response]

Table4-2. Command CMD_SET_PARAM

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0002	
LEN	5	
DATA	1bytes	Parameter Type
	4bytes	Parameter Value
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0002	
LEN	2	
RET	Result Code (ERR_SUCCESS/ ERR_FAIL)	
DATA	No Data	

[Parameter Type]

Type	Description of Parameter Value
0	Device ID of this module. The range is 1~255.
1	Security level, default is 3. The larger the value, the lower the false recognition rate (FAR) is and the higher the false rejection rate (FRR) is. In other words, when the security level is higher, the probability to misunderstand will be lower, but it is not easy to pass the authentication. Conversely, when the security level is lower, it is easier to pass the authentication, but also easier to misunderstand.
2	Option for fingerprint Duplication Check. It can be set to 0 or 1. If it is 1, when processing the command CMD_STORE_CHAR, checks if there are fingerprints already enrolled. If it is 0, no duplication check is performed. Notes: The option of fingerprint duplicate check is valid only for firmware with the general algorithm (498-byte feature algorithm). The firmware with the other algorithms will not detect the fingerprints enrolled (even though set to 1, the duplication check will not be performed).
3	Index value of Baudrate. The index value can be set in the range 1~8, default is 5 (i.e. 115200bps). 1: 9600bps, 2: 19200bps, 3: 38400bps, 4: 57600bps, 5: 115200bps 6: 230400bps, 7: 460800bps, 8: 921600bps

4	Option for the fingerprint template Auto-Learn. It can be set to 0 or 1. If it is 1, the smart update is performed when command CMD_SEARCH and CMD_VERIFY are processed. If it is 0, the smart update is not performed.
5	Fingerprint TimeOut, it can be set 1 ~ 60 seconds. This parameter is used in the command CMD_GET_IMAGE to wait for fingerprint input within the FP TimeOUT. Notes: This parameter is valid only for the swipe fingerprint sensor module, default is 5s.

Example 4.2- ex1: Set the baud rate to 57600bps

Host command: 55 AA 00 00 02 00 05 00 03 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0d 01

Module response: AA 55 01 00 02 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 04 01

Example 4.2-ex2: Set security level = 5

Host command: 55 AA 00 00 02 00 05 00 01 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0c 01

Module response: AA 55 01 00 02 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 04 01

4.3 Read Parameters (CMD_GET_PARAM 0x0003)

[Function]

According to the specified Parameter Type, reads the module parameter values (Device ID, Security Level, Baudrate, Duplication Check, Auto Learn, and so on). About Parameter Type, refer to the above CMD_SET_PARAM.

[Sequence]

① If the specified Parameter Type is invalid, returns ERR_INVALID_PARAM.

② Returns the device parameters corresponding to the specified Parameter Type.

[Command and Response]

Table 4-3. Command CMD_GET_PARAM

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0003	
LEN	1	
DATA	1byte	Parameter Type
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0003	
LEN	Success : 6, Failure : 2	
RET	Result Code (ERR_SUCCESS/ ERR_FAIL)	

DATA	4bytes	On success: Parameter Value

Example 4.3- ex1: Read the current security level (Security level = 3)

Host command: 55 AA 00 00 03 00 01 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 04 01

Module response: AA 55 01 00 03 00 06 00 00 00 03 00 00 00 00 00 00 00 00 00 00 00 00 00 0C 01

Example 4.3- ex2: Read the current TimeOut value (TimeOut=5S) in the swipe fingerprint sensor module.

Host command: 55 AA 00 00 03 00 01 00 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 08 01

Module response: AA 55 01 00 03 00 06 00 00 00 05 00 00 00 00 00 00 00 00 00 00 00 00 00 0E 01

4.4 Read Device Information (CMD_DEVICE_INFO 0x0004)

[Function]

Gets the device information such as the version of the module (Device Information of Target). The format of device information is as follows: (please see "Composition Rules of Module Information").

ID_Protocol Stack and Algorithm Name_Algorithm Chip Model_Collector Model_No or Module Hardware Name_None or Inner (Fingerprint Capacity) Version Number

Among the module information, the keys are the algorithm name and the fingerprint capacity.

[Sequence]

① First, sends the length of the data packet to send at next time to the host using the command response packet.

② Using the data response packet, sends the device information.

[Command and Response]

Table 4-4. Command CMD_DEVICE_INFO

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0004	
LEN	0	
DATA	No Data	
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0004	
LEN	4	
RET	ERR_SUCCESS	
DATA	2bytes	Data length of data response packet
When Success		

PREFIX	0x5AA5
SID	Source Device ID
DID	Destination Device ID
RCM	0x0004
LEN	2 + Length of Device Information
RET	ERR_SUCCESS
DATA	Device Information

Example 4.4- Reading the module information using command CMD_DEVICE_INFO

Host command: 55 AA 00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 01

Module Response: AA 55 01 00 04 00 04 00 00 00 1A 00 00 00 00 00 00 00 00 00 00 00 00 00 22 01

Response data packet: A5 5A 01 00 04 00 1F 00 00 00 53 45 4F 4E 5F 47 44 5F 46 50 43 31 30 32 30 28 32 30 30 30 66 70 29 20 56 31 2E 30 00 2B08;

The data block highlighted in blue is the ASCII code in the device specification "SEON_GD_FPC1020 (2000fp) V1.0".

4.5 Set the module into IAP mode (CMD_ENTER_IAP_MODE 0x0005)

[Function]

Sets the device to the IAP mode.

[Sequence]

After receiving the command response packet, sets the device to the IAP state.

[Command and Response]

Table 4-5. Command CMD_ENTER_IAP_MODE

PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID
CMD	0x0005
LEN	0
DATA	No Data
PREFIX	0x55AA
SID	Source Device ID
DID	Destination Device ID
RCM	0x0005
LEN	2
RET	Result Code
DATA	No Data

Notes: The command CMD_ENTER_IAP_MODE will clear the firmware program, and this command needs to be executed when the firmware needs to be upgraded. After executing this command, the firmware must be re-programmed with USB. Please use this command with caution!!

4.6 Capture fingerprint image (CMD_GET_IMAGE 0x0020)

[Function]

Capture fingerprint images from the sensor and save them in ImageBuffer.

[Sequence]

Collect fingerprint images from the collector. If the captured image is correct, ERR_SUCCESS will be returned. Otherwise, an error code is returned.

For swipe **semiconductor fingerprint sensor:**

① If fingerprint is not detected within Fp TimeOut time, it will return ERR_TIME_OUT.

② If the CMD_FP_CANCEL command is received during the collection process or waiting for the fingerprint input, the operation of this command will be cancelled and ERR_FP_CANCEL will be returned.

[Command and Response]

Table 4-6. Command CMD_IMAGE

PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID
CMD	0x0020
LEN	0
DATA	No Data
PREFIX	0x55AA
SID	Source Device ID
DID	Destination Device ID
RCM	0x0020
LEN	2
RET	Result Code
DATA	0

Example 4.6- ex1: The command and response transferred when the module detects the finger after the fingerprint image is captured.

Host command: 55 AA 00 00 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 1F 01

Module response: AA 55 01 00 20 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 22 01

Example 4.6- ex2: The command and response transferred when the result of capturing the swipe **fingerprint image is FP TimeOut**

CMD_GET_IMAGE: 55 AA 00 00 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 1F 01

ERR_TIME_OUT: AA 55 01 00 20 00 02 00 23 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 45 01

4.7 Detect finger (CMD_FINGER_DETECT 0x0021)

[Function]

Check the fingerprint input status at the time the command is received and return the result.

[Sequence]

At the time the command is received, it is return the fingerprint input state detected by sensor.

[Command and Response]

Table 4-7. Command CMD_FINGER_DETECT

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0021	
LEN	0	
DATA	No Data	
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0021	
LEN	Success : 3, Failure: 2	
RET	Result Code	
DATA	1byte	When successful: fingerprint input status (1: fingerprint input, 0: no fingerprint input)

Example 4.7- ex1: No fingerprint detected

Host command: 55 AA 00 00 21 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 20 01

Target response: AA 55 01 00 21 00 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 24 01

Example 4.7- ex2: Fingerprint detected

Host command: 55 AA 00 00 21 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 20 01

Target response: AA 55 01 00 21 00 03 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 25 01

4.8 Upload fingerprint image to host (CMD_UP_IMAGE_CODE 0x0022)

[Function]

According to the specified Image Type， the image stored in ImageBuffer is sent to Host.

If Image Type is 0: send full image:

(Optical collector and semiconductor collector: 242*266(202*258), swipe collector FPC1080：128*436).

If it is 1: Then send 1/4 image (get 1 point per 4 point).

(Optical collector and semiconductor collector: 121*133(101*129)，swipe collector FPC1080: 64*218).

[Sequence]

① If it is invalid the specified Image Type，return ERR_INVALID_PARAM.

② Using command response packet, send the size of the image to HOST.

③ According to Image Type, using response data packet, divide the image in 496bytes unit and send it to HOST.

[Command and Response]

Table 4-8. Command CMD_UP_IMAGE

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0022	
LEN	1	
DATA	1byte	Image Type（0：Full，1：Quarter）
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0022	
LEN	6/2	
RET	Result Code	
DATA	2bytes	Success: width of image Full image（242/202/128） Quarter image（121/101/64）
	2bytes	Success: Height of image Full image(266/258/436） Quarter image（133/129/218）
On success: Target sends a response data packet to HOST		
PREFIX	0x5AA5	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0022	
LEN	4+ Length of image data	
RET	ERR_SUCCESS	
DATA	Length of image data(2bytes) + Image data	

…

Continuously send response data packet

Notes:

1. Before calling this command, must call CMD_GET_IMAGE to save the fingerprint image in ImageBuffer.
2. High resolution mode (Full Mode) width * height: 242*266/202*258/128*436
3. Low resolution mode (Quarter Mode) width * height: 121*133/101*129/64*218

Example 4.8-ex1: Upload fingerprint image of 202*258 sensor in full resolution

Host command: 55 AA 00 00 22 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 22 01

Module response: AA 55 01 00 22 00 06 00 00 00 CA 00 02 01 00 00 00 00 00 00 00 00 00 00 F5 01

Notes: Width of full image=0xCA=202， Height of full image=0x102=258

Module response data packet: image data size is 202*258=52116 byte, divided into 105*496 byte + 1*36 byte.

A5 5A 01 00 22 00 F4 01 00 00 F0 01 496 bytes of data in this frame Checksum with 2 bytes

…A total of 105 response packets containing 496 bytes of image data.

A5 5A 01 00 22 00 28 00 00 00 24 00 36 bytes of data in the last frame Checksum with 2 bytes

The last response packet contains 36 bytes of image data.

Example 4.8- ex2: Upload fingerprint image of 202*258 sensor in quarter resolution

Host command： 55 AA 00 00 22 00 01 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 23 01

Module response: AA 55 01 00 22 00 06 00 00 00 65 00 81 00 00 00 00 00 00 00 00 00 00 00 0E 02

Notes: width of 1/4image=0x65=101， height of full image=0x81=129

Module response data packet: image data size is 202*258/4=13029 byte, divided into 26*496 byte + 1*133byte

A5 5A 01 00 22 00 F4 01 00 00 F0 01 496 byte data of this frame checksum with 2 byte.

… A total of 26 response packets containing 496 bytes of image data.

A5 5A 01 00 22 00 89 00 00 00 85 00 The 133 byte data of the last frame checksum with 2 byte

The last response data packet contain 133 bytes of image data.

4.9 Download Fingerprint Image to module (CMD_DOWN_IMAGE 0x0023)

[Function]

Save the image data received from the Host in ImageBuffer.

Host sends images to Target in units of 496bytes. At this time, the image data number is sent at the same time.

Notes: Fingerprint image requirement: Resolution: 500DPI， Grayscale： 8bit grayscale.
pixel size: Optical collector: 242*266；
Push-type semiconductor collector(such as FPC1011): 202*258；
swipe semiconductor collector（such as FPC1080): 128*436；

[Sequence]

① If the width and height of the image is invalid, return ERR_INVALID_PARAM.
② Using response packet, return ERR_SUCCESS.
③ Receive the command packet and save the image in ImageBuffer.

[Command and Response]

Table 4-9. Command CMD_DOWN_IMAGE

Command packet		
PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0023	
LEN	4	
DATA	2bytes	Width of image: 242/202/128
	2bytes	Height of image: 266/258/436
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0023	
LEN	2	
RET	Result Code	
DATA	0	
Command data packet		
PREFIX	0xA55A	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0023	
LEN	2 + Size of image daga	
DATA	Image data number(2bytes) + Image data	
Response data packet		
PREFIX	0x5AA5	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0023	
LEN	2	
RET	Result Code	
DATA	0	

…

Command data packet	
PREFIX	0xA55A
SID	Source Device ID
DID	Destination Device ID
CMD	0x0023
LEN	2 + Length of image data
DATA	Image data number(2bytes) + Image data
Response data packet	

PREFIX	0x5AA5
SID	Source Device ID
DID	Destination Device ID
RCM	0x0023
LEN	2
RET	Result Code
DATA	0

4.9 Example: Download fingerprint image to ImageBuffer.

Host command： 55 AA 00 00 23 00 04 00 CA 00 02 01 00 00 00 00 00 00 00 00 00 00 00 00 F3 01

Module response: AA 55 01 00 23 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 25 01

Command data packet: 5A A5 00 00 23 00 F2 01 00 00 496 byte data of this module checksum with 2 byte.

Response data packet: A5 5A 01 00 23 00 02 00 00 00 25 01: The length of the data response packet is 12 bytes because there is no data.

… A total of 105 command packets and response packets containing 496 bytes of image data.

Command data packet: 5A A5 00 00 23 00 26 00 69 00 The last data block containing 36bytes if umage data checksum with 2 byte.

Response data packet: A5 5A 01 00 23 00 02 00 00 00 25 01: Length of data response packet is 12 byte because there is no data.

4.10 Control the Collector Backlight On/Off (CMD_SLED_CTRL 0x0024)

[Function]

Control the on/off of the backlight of the optical collector, or the display status of the three-color indicator light of some models of fingerprint modules.

[Sequence]

Do nothing and return ERR_SUCCESS.

[Command and Response]

Table 4-10. Command CMD_SLED_CTRL

PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID
CMD	0x0024
LEN	2
DATA	LED state（1: Open，0: Close）
PREFIX	0x55AA
SID	Source Device ID
DID	Destination Device ID
RCM	0x0024
LEN	2

RET	ERR_SUCCESS
DATA	0

Example 4.10- Turn on the backlight (Backlight LED On)

Host command: 55 AA 00 00 24 00 02 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 26 01

Module response: AA 55 01 00 24 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 26 01

4.11 Save fingerprint template data to module fingerprint library (CMD_STORE_CHAR 0x0040)

[Function]

Store the fingerprint template temporarily stored in the specified Ram Buffer in the module fingerprint library with the specified number.

[Sequence]

① If the specified template number is invalid, it is return error code ERR_INVALID_TMPL_NO.

② if the specified Ram Buffer number is invalid, it is return error code ERR_INVALID_BUFFER_ID.

③ If Duplication Check is set OFF， directly register the fingerprint template data in the specified Ram Buffer in the fingerprint library with the specified number and return the result.

④ If Duplication Check is set ON， A 1:N comparison will be made between the Template in the specified Ram Buffer and all the Templates in the enrolled fingerprint library.

If there is template with successful verified, indicating that the fingerprint has been enrolled, it will return (RET):ERR_DUPLICATION_ID, and DATA will return the template number of the successful verify.

Otherwise, enroll the template in the fingerprint library with the specified Template number and return the result.

[Command and Response]

Table 4-11. Command CMD_STORE_CHAR

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0040	
LEN	4	
DATA	2bytes	Template Number
	2bytes	Ram Buffer Number
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0040	

LEN	LEN length is related to the result code: 2/or 4 When Result Code is ERR_DUPLICATION_ID: LEN=4; Otherwise: LEN=2	
RET	Result Code	
DATA	2bytes	The data is related to the result code: 0 or duplicated fingerprint number When the result code is ERR_DUPLICATION_ID: Duplicated fingerprint number; Otherwise: 0

Example 4.11- Save the template data in RamBuffer0 in the fingerprint data library into the specified number 1:

Host command packet: 55 AA 00 00 40 00 04 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 44 01

Module response packet: AA 55 01 00 40 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 42 01

4.12 Read the fingerprint in the module and temporarily store it in RamBuffer (CMD_LOAD_CHAR 0x0041)

[Function]

Take out the fingerprint template in the specified number in the fingerprint library and temporarily store it in the specified Ram Buffer.

[Sequence]

① If the specified template number is invalid，return ERR_INVALID_TMPL_NO.

② if there is no enrolled template in the specified template number, return the error code ERR_TMPL_EMPY.

③ if the specified Ram Buffer number is invalid, return the error code ERR_INVALID_BUFFER_ID.

④ Save the Template of the specified number in the specified Ram Buffer and return ERR_SUCCESS.

[Command and Response]

Table 4-12. Command CMD_LOAD_CHAR

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0041	
LEN	4	
DATA	2bytes	Template Number
	2bytes	Ram Buffer Number
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	

RCM	0x0041
LEN	2
RET	Result Code
DATA	0

Example 4.12- Read template data numbered 1 and temporarily store it in RamBuffer0:

Host command packet: 55 AA 00 00 41 00 04 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 45 01

Module response packet: AA 55 01 00 41 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 43 01

4.13 Upload the fingerprint template temporarily stored in RamBuffer to the host (CMD_UP_CHAR 0x0042)

[Function]

Send the Template in the specified Ram Buffer to Host.

[Sequence]

① If the specified RamBuffer Number is invalid, return ERR_INVALID_BUFFER_ID.

② Using command response packet, send the size of the template data to HOST.

③ Using response data packet, send the template data in the specified number to HOST.

[Command and Response]

Table 4-13. Command CMD_UP_CHAR

PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID
CMD	0x0042
LEN	2
DATA	Ram Buffer ID
PREFIX	0x55AA
SID	Source Device ID
DID	Destination Device ID
RCM	0x0042
LEN	4
RET	ERR_SUCCESS or ERR_FAIL
DATA	**Success: The data length of the next data response packet (Template Record Size + 2)** **Notes: The length of the fingerprint template data generated by each version of the algorithm is different, see Table 3-2 in Chapter 3** **Failure: error code**.
When successful, the module sends a response packet	
PREFIX	0x5AA5
SID	Source Device ID

DID	Destination Device ID
RCM	0x0042
LEN	**Template Record Size + 2** **Notes: The length of the fingerprint template data generated by each version of the algorithm is different, see Table 3-2 in Chapter 3**
RET	ERR_SUCCESS
DATA	Template Record Data

Notes:

1. **Before calling this command, first must call one of CMD_GENERATE, CMD_DOWN_CHAR, CMD_LOAD_CHAR, save the Template in a Ram Buffer.**
2. **The length of the fingerprint template data generated by each version of the algorithm is different, see Table 3-2 in Chapter 3.**

Example 4.13- Example of uploading template data in RamBuffer0 to HOST:

Ex1: General algorithm firmware (such as SEONU, fingerprint template data size is 498 byte)

Host command packet: 55 AA 00 00 42 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 43 01

Module response packet: AA 55 01 00 42 00 04 00 00 00 F2 01 00 00 00 00 00 00 00 00 00 00 00 00 39 02

Response data packet: A5 5A 01 00 42 00 F4 01 00 00 498 byte of this fingerprint template Checksum of 2 bytes

Ex2:160 new algorithm firmware (such as SEONU2, fingerprint template data size is 1008 byte)

Host command packet: 55 AA 00 00 42 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 43 01

Module response packet: AA 55 01 00 42 00 04 00 00 00 F0 03 00 00 00 00 00 00 00 00 00 00 00 00 39 02

Response data packet: A5 5A 01 00 42 00 F2 03 00 00 1008 byte of this fingerprint template Checksum of 2 bytes

Ex3: 192 new algorithm firmware (such as SEONU7, fingerprint template data size is 448 byte)

Host command packet: 55 AA 00 00 42 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 43 01

Module response packet: AA 55 01 00 42 00 04 00 00 00 C0 01 00 00 00 00 00 00 00 00 00 00 00 00 07 02

Response data packet: A5 5A 01 00 42 00 C2 01 00 00 448 byte of this fingerprint template Checksum of 2 bytes

Ex4: Non-standard algorithm(FPC1021 dedicated algorithm)firmware (such as SEONU5， fingerprint template data size is 2024 byte)

Host command packet: 55 AA 00 00 42 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 43 01

Module response packet: AA 55 01 00 42 00 04 00 00 00 E8 07 00 00 00 00 00 00 00 00 00 00 00 00 35 02

2024 byte template data is divided into five data packets (four 496 byte data packet plus one 40 byte data packet) is as follows:

Response data packet 1: A5 5A 01 00 42 00 F2 01 00 00 496 byte of this fingerprint template data Checksum of 2 bytes

Response data packet 2: A5 5A 01 00 42 00 F2 01 00 00 496 byte of this fingerprint template data Checksum of 2 bytes

Response data packet 3: A5 5A 01 00 42 00 F2 01 00 00 496 byte of this fingerprint template data Checksum of 2 bytes

Response data packet 4: A5 5A 01 00 42 00 F2 01 00 00 496 byte of this fingerprint template data Checksum of 2 bytes

Response data packet 5: A5 5A 01 00 42 00 2A 00 00 00 40 byte of this fingerprint template data Checksum of 2 bytes

4.14 Download the fingerprint template data to the RamBuffer specified by the module (CMD_DOWN_CHAR 0x0043)

[Function]

Receive fingerprint Template Data from Host and save it in the specified Ram Buffer.

[Sequence]

① Host sends command packet to make Target enter data (Ram Buffer + Template) receiving waiting state.

The DATA field of the command packet has set the length of the command data packet to be sent next time.

② Target checks the accuracy of the received command packet.
If it is incorrect, the error code is returned and the processing ends.
If the size of the data to be received is incorrect, return ERR_INVALID_PARAM.
If it is correct, it sends a response packet to the HOST, indicating that the module has entered the data (RAMbuffer number + template data) receiving waiting state, and has entered the data (RAMbuffer number + template data) receiving waiting state.

③ When the Host receives the response packet that the target has entered the data receiving waiting state, it uses the RamBuffer and Template data specified by the instruction packet and sends it to the Target.

④ After Target receives the command packet, if the Ram Buffer ID is invalid, return ERR_INVALID_BUFFER_ID.

⑤ Check the CheckSum of the received Template. If it is incorrect, return ERR_INVALID_TMPL_DATA.

⑥ Save the received Template in the specified Ram Buffer and return ERR_SUCCESS.

[Command and Response]

Table 4-14. Command CMD_DOWN_CHAR

Command packet	
PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID
CMD	0x0043
LEN	2
DATA	2 + Template Record Size
PREFIX	0x55AA
SID	Source Device ID
DID	Destination Device ID
RCM	0x0043
LEN	4
RET	Result Code
DATA	0

Command data packet	
PREFIX	0xA55A
SID	Source Device ID
DID	Destination Device ID
CMD	0x0043
LEN	2 + Template size(498)
DATA	Ram Buffer Number (2byte) + Template Data
PREFIX	0x5AA5
SID	Source Device ID
DID	Destination Device ID
RCM	0x0043
LEN	4
RET	Result Code
DATA	0

Notes: Save Template in RamBuffer2，if calling (CMD_SEARCH, CMD_VERIFY, CMD_GENERATE, CMD_STORE_CHAR, CMD_DEL_CHAR, CMD_GET_EMPTY_ID, CMD_GET_STATUS, GET_BROKEN_ID, CMD_GETN_ENROLL_COUNT) and other commands, it will be cleared. Try to avoid using RamBuffer2.

Example 4.14- Download fingerprint template data to RamBuffer0 in module.

Ex1, General algorithm firmware (such as SEONU，fingerprint template data size is 498 bytes)

Host command: 55 AA 00 00 43 00 02 00 F4 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 39 02

Command response: AA 55 01 00 43 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 45 01

Data block packet: 5A A5 00 00 43 00 F4 01 00 00 498 byte of this fingerprint template data Checksum of 2 bytes

Response: A5 5A 01 00 43 00 02 00 00 00 45 01 ; The length of the response packet is 12 bytes because there is no data.

Ex2, 160 new algorithm firmware (such as SEONU2，fingerprint template data size is 1008 bytes).

Host command: 55 AA 00 00 43 00 02 00 F3 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 39 02

Command response: AA 55 01 00 43 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 45 01

Data block packet: 5A A5 00 00 43 00 F2 03 00 00 1008 byte of this fingerprint template data Checksum of 2 bytes

Response packet: A5 5A 01 00 43 00 02 00 00 00 45 01 ; The length of the response packet is 12 bytes because there is no data.

Ex3, 192 new algorithm firmware (such as SEONU7，fingerprint template data size is 448 bytes).

Host command: 55 AA 00 00 43 00 02 00 C2 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 07 02

Command response: AA 55 01 00 43 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 45 01

Data block packet: 5A A5 00 00 43 00 C2 01 00 00 448 byte of this fingerprint template data Checksum of 2 bytes

Response packet: A5 5A 01 00 43 00 02 00 00 00 45 01 : The length of the response packet is 12 bytes because there is no data.

Ex4, Non-standard algorithm firmware (such as SEONU5), the 2024-byte template data is divided into five data packets for transmission, particularly described as follows:

Host command: 55 AA 00 00 43 00 02 00 EC 07 00 00 00 00 00 00 00 00 00 00 00 00 00 00 37 02

Command response: AA 55 01 00 43 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 45 01

The 2024 byte data is divided into five data packets (4*496+40), The 2024 byte data is divided into 5 data packets (4*496+40), the data block number of 2 bytes is added before the template data, and the transmission is as follows:

Data block packet 1: 5A A5 00 00 43 00 F4 01 00 00 00 00 496 byte of this fingerprint template data Checksum of 2 bytes

Response packet 1: A5 5A 01 00 43 00 02 00 00 00 45 01 ; The length of the response packet is 12 bytes because there is no data.

Data block packet 2: 5A A5 00 00 43 00 F4 01 00 00 01 00 496 byte of this fingerprint template data Checksum of 2 bytes

Response packet 2: A5 5A 01 00 43 00 02 00 00 00 45 01 ; The length of the response packet is 12 bytes because there is no data.

Data block packet 3: 5A A5 00 00 43 00 F4 01 00 00 02 00 496 byte of this fingerprint template data Checksum of 2 bytes

Response packet 3: A5 5A 01 00 43 00 02 00 00 00 45 01 ; The length of the response packet is 12 bytes because there is no data.

Data block packet 4: 5A A5 00 00 43 00 F4 01 00 00 03 00 496 byte of this fingerprint template data Checksum of 2 bytes

Response packet 4: A5 5A 01 00 43 00 02 00 00 00 45 01 ; The length of the response packet is 12 bytes because there is no data.

Data block packet 5: 5A A5 00 00 43 00 F4 01 00 00 04 00 40 byte of this fingerprint template data Checksum of 2 bytes

Response packet 5: A5 5A 01 00 43 00 02 00 00 00 45 01 ; The length of the response packet is 12 bytes because there is no data.

4.15 Delete Fingerprints In Specified Number Range (CMD_DEL_CHAR 0x0044)

[Function]

Delete all enrolled templates in the specified number range (starting template number ~ ending template number).

[Sequence]

① If the specified range is invalid, ERR_INVALID_PARAM will be returned.
② If there is no template enrolled in the specified range, ERR_TMPL_EMPTY will be returned.
③ Delete all enrolled templates in the specified range and return their results.

[Command and Response]

Table 4-15. Command CMD_DEL_CHAR

PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID
CMD	0x0044

LEN	4	
DATA	2bytes	Start Template Number
	2bytes	End Template Number

PREFIX	0x55AA
SID	Source Device ID
DID	Destination Device ID
RCM	0x0044
LEN	2
RET	Result Code
DATA	0

Example 4.15- ex1: Delete all fingerprints numbered 1~3000 in the database.

Host command：55 AA 00 00 44 00 04 00 01 00 BB 08 00 00 00 00 00 00 00 00 00 00 00 00 0B 02
Module response: AA 55 01 00 44 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 46 01

Example 4.15- ex2: Delete all fingerprints numbered 1~1700 in the database.

Host command：55 AA 00 00 44 00 04 00 01 00 A4 06 00 00 00 00 00 00 00 00 00 00 00 00 F2 01
Module response: AA 55 01 00 44 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 46 01

Example 4.15- ex3：Delete all fingerprints numbered 1~500 in the database.

Host command：55 AA 00 00 44 00 04 00 01 00 F4 01 00 00 00 00 00 00 00 00 00 00 00 00 3D 02
Module response: AA 55 01 00 44 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 46 01

Example 4.15- ex4：Delete all fingerprints numbered 1~200 in the database.

Host command：55 AA 00 00 44 00 04 00 01 00 C8 00 00 00 00 00 00 00 00 00 00 00 00 00 10 02
Module response: AA 55 01 00 44 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 46 01

4.16 Get First Number That Can Be Enrolled In Specified Number Range (CMD_GET_EMPTY_ID 0x0045)

[Function]

Get the first template number that can be enrolled (without template enrollment) in the specified range (starting template number ~ ending template number).

[Sequence]

① If the specified range is invalid, ERR_INVALID_PARAM will be returned.
② Search for the first ID that can be enrolled in the specified range.
If it exists, return its value. Otherwise, it returns ERR_EMPTY_ID_NOEXIST.

[Command and Response]

Table4-16. Command CMD_GET_EMPTY_ID

PREFIX	0xAA55
SID	Source Device ID

DID	Destination Device ID	
CMD	0x0045	
LEN	4	
DATA	2bytes	Starting template number
	2bytes	Ending template number
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0045	
LEN	Success : 4, Failure : 2	
RET	Returm Code	
DATA	2bytes	Success: the first template number that can be enrolled

Example 4.16 - Get the first enrollable number in the number range of 1~2000 (0x1-0x07D0), and the returned number is 11.

Host command: 55 AA 00 00 45 00 04 00 01 00 D0 07 00 00 00 00 00 00 00 00 00 00 00 00 20 02

Module response: AA 55 01 00 45 00 04 00 00 00 0B 00 00 00 00 00 00 00 00 00 00 00 00 00 54 01

4.17 Check If Specified Number Has Been Registered (CMD_GET_STATUS 0x0046)

[Function]

Get the enrollment status of the template with the specified number.

[Sequence]

If the specified template number is invalid, ERR_INVALID_TMPL_NO will be returned.

If the template with the specified number has been enrolled, 1 is returned. Otherwise, 0 is returned.

[Command and Response]

Table4-17. Command CMD_GET_STATUS

PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID
CMD	0x0046
LEN	2
DATA	Specified template number
PREFIX	0x55AA
SID	Source Device ID

DID	Destination Device ID	
RCM	0x0046	
LEN	Success : 3, Failure : 2	
RET	Result Code	
DATA	1byte	Success: Enrollment status （1：already enrolled，0：not enrolled）

Example 4.17 - ex1：Get the enrollment status with ID number = 1 that can be enrolled.

Host command： 55 AA 00 00 46 00 02 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 48 01
Module response: AA 55 01 00 46 00 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 49 01

Example 4.17 - ex2：Get the enrollment status with ID number = 1 that was already enrolled.

Host command： 55 AA 00 00 46 00 02 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 48 01
Module response: AA 55 01 00 46 00 03 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 4A 01

4.18 Check If There Are Damaged Templates In Fingerprint Database Within Specified Range (CMD_GET_BROKEN_ID 0x0047)

[Function]

Check whether the enrolled templates in the specified range (starting template number ~ ending template number) are damaged.

During the Flash Write operation, the template may be damaged due to sudden power failure.

HOST uses this command at any time (for example, the initial startup of Target) to check the damage of the template.

The damaged template needs to be deleted and re-enrolled.

[Sequence]

① If the specified range is invalid, ERR_INVALID_PARAM will be returned.

② Check the damage of all enrolled templates within the specified range.

If there are damaged templates, return the number of damaged templates and the number of the first damaged template.

Otherwise, both the number of templates and the template number are 0.

[Command and Response]

Table4-18. Command CMD_GET_BROKEN_ID

PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID

CMD	0x0047	
LEN	4	
DATA	2bytes	Starting template number
	2bytes	Ending template number
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0047	
LEN	Success : 6, Failure : 2	
RET	Result Code	
DATA	2byte	Success: the number of damaged templates
	2byte	Success: the first damaged template number

Example 4.18 - Get ID numbers of damaged fingerprints in the range of 1~200

Host command: 55 AA 00 00 47 00 04 00 01 00 C8 00 00 00 00 00 00 00 00 00 00 00 00 00 13 02
Module response: AA 55 01 00 47 00 06 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 4D 01

4.19 Get Total Number of Enrolled Fingerprints In Specified Range (CMD_GET_ENROLL_COUNT 0x0048)

[Function]

Get the total number of fingerprints enrolled in the specified range (starting template number ~ ending template number).

[Sequence]

① If the specified range is invalid, ERR_INVALID_PARAM will be returned.
② Returns the number of fingerprints enrolled in the module.

[Command and Response]

Table 4-19. Command CMD_GET_ENROLL_COUNT

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0048	
LEN	4	
DATA	2bytes	The starting template number
	2bytes	The ending template number
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0048	
LEN	Success : 4, Failure : 2	

RET	Result Code	
DATA	2bytes	Number of enrolled templates

Example 4.19 - Get the total number of enrolled users in the range of 1~200 (0x0001~0x00C8), the total is 10 (0x000A)

Host command： 55 AA 00 00 48 00 04 00 01 00 C8 00 00 00 00 00 00 00 00 00 00 00 00 00 14 02

Module response: AA 55 01 00 48 00 04 00 00 00 0A 00 00 00 00 00 00 00 00 00 00 00 00 00 56 01

4.20 Generate Templates From Fingerprint Images Temporarily Stored In ImageBuffer (CMD_GENERATE 0x0060)

[Function]

Generates fingerprint template from fingerprint image in ImageBuffer and saves it in specified RamBuffer.

[Sequence]

① If the specified RamBuffer number is invalid, the error code ERR_INVALID_BUFFER_ID will be returned.

② Check the correctness of the image in ImageBuffer. If it is not correct, ERR_BAD_QUALITY is returned.

③ Save the generated template in the specified RamBuffer and return ERR_SUCCESS.

[Command and Response]

Table4-20. Command CMD_GENERATE

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0060	
LEN	2	
DATA	2bytes	Ram Buffer Number
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0060	
LEN	2	
RET	Result Code	
DATA	0	

Notes:

The command CMD_GENERATE generates the template from ImageBuffer. Therefore, before calling this command, you need to call the command CMD_GET_IMAGE in advance to save the image in the ImageBuffer.

Example 4.20 - ex1：Generate the template from ImageBuffer and save it in RamBuffer0

Host command：55 AA 00 00 60 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 61 01
Module response: AA 55 01 00 60 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 62 01

Example 4.20 - ex2：Generate the template from ImageBuffer and save it in RamBuffer1

Host command：55 AA 00 00 60 00 02 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 62 01
Module response: AA 55 01 00 60 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 62 01

Example 4.20 - ex3：Generate the template from ImageBuffer and save it in RamBuffer2

Host command：55 AA 00 00 60 00 02 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 63 01
Module response: AA 55 01 00 60 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 62 01

4.21 Merge Fingerprint Templates For Storage (CMD_MERGE 0x0061)

[Function]

Merges the templates temporarily stored in RamBuffer to generate a template and saves it in the specified RamBuffer.

The number to merge can be 2 or 3:

If it is 2, the templates of RamBuffer0 and RamBuffer1 are merged.

If it is 3: the templates of RamBuffer0, RamBuffer1 and RamBuffer2 are merged.

[Sequence]

① If the specified RamBuffer number is invalid, the error code ERR_INVALID_BUFFER_ID will be returned.

② If the number to merge is invalid, ERR_GEN_COUNT will be returned.

③ According to the number to merge, merges the templates and generates a template. If the merging fails, an error code is returned.

④ Saves the generated template in the specified RamBuffer and returns ERR_SUCCESS.

[Command and Response]

Table 4-21. Command CMD_MERGE

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0061	
LEN	3	
DATA	2bytes	Ram Buffer Number
	1byte	Number to merge (2/3)
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0061	

LEN	2
RET	Result Code
DATA	0

Example 4.21 – Merge 3 temporary fingerprint templates in RamBuffer into 1 fingerprint template.

Host command: 55 AA 00 00 61 00 03 00 00 00 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 66 01

Module response: AA 55 01 00 61 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 63 01

4.22 Template Matching Between Two Specified RamBuffers (CMD_MATCH 0x0062)

[Function]

Matches the templates in the two specified RamBuffers.

[Sequence]

① If the specified RamBuffer number is invalid, the error code ERR_INVALID_BUFFER_ID will be returned.

② Carries out the template matching between the two specified RamBuffer and returns the result. If the comparison is successful, RET returns ERR_SUCCESS and DATA returns the smart update result.

Otherwise, RET returns ERR_VERIFY.

[Command and Response]

Table4-22. Command CMD_MATCH

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0062	
LEN	4	
DATA	2bytes	The number of the first Ram Buffer to be matched
	2bytes	Number of the second Ram Buffer to be matched
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0062	
LEN	2	
RET	Result Code	
DATA	No Data	

Example 4.22 – Carry out the 1:1 matching the fingerprint template in RamBuffer0 and RamBuffer1, the result: the matching is successful.

Host command： 55 AA 00 00 62 00 04 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 66 01

Module response: AA 55 01 00 62 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 64 01

4.23 1:N Matching In The Specified Number Range (CMD_SEARCH 0x0063)

[Function]

1:N matching between the template in the specified RamBuffer and all the enrolled fingerprint templates in the specified search range (starting template number ~ ending template number) and returning the result.

[Sequence]

① If the specified RamBuffer number is invalid, the error code ERR_INVALID_BUFFER_ID will be returned.

② If the specified search range is invalid, the error code ERR_INVALID_BUFFER_ID will be returned.

③ If there is no enrolled template, the error code ERR_ALL_TMPL_EMPTY will be returned.

④ Carries out the matching between the template in the specified Ram Buffer and all the enrolled templates and returns the result.

If the search is successful, RET returns ERR_SUCCESS and the searched template number and smart update result are returned in the DATA field.

Otherwise, RET returns ERR_IDENTIFY.

[Command and Response]

Table 4-23. Command CMD_SEARCH

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0063	
LEN	6	
DATA	2bytes	Ram Buffer ID
	2bytes	The starting template number to be searched
	2bytes	The ending template number to be searched
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0063	
LEN	Success : 5, Failure : 2	
RET	Result Code	
DATA	3bytes	Success： Template ID (2bytes) + SmartUpdate Result (1byte)

Example 4.23 Matching the templates temporarily stored in RamBuffer0 with the finger print with the number in range of 1~200, the result is ID = 8

Host command: 55 AA 00 00 63 00 06 00 00 00 01 00 C8 00 00 00 00 00 00 00 00 00 00 00 31 02

Module response: AA 55 01 00 63 00 05 00 00 00 08 00 01 00 00 00 00 00 00 00 00 00 00 00 71 01

4.24 Matching Specified RamBuffer With Template With Specified Number In Fingerprint Library (CMD_VERIFY 0x0064)

[Function]

Carries out the 1:1 matching between the specified template in the RamBuffer and the template with the specified number in the database is performed and returns the result.

[Sequence]

① If the specified template number is invalid, the error code ERR_INVALID_TMPL_NO will be returned.

② If the specified RamBuffer number is invalid, the error code ERR_INVALID_BUFFER_ID will be returned.

③ If there is no enrolled template with the specified number, the error code ERR_TMPL_EMPTY will be returned.

④ The template in the specified RamBuffer is matched with the template with the specified number and the result is returned.

If the matching is successful, RET returns ERR_SUCCESS and DATA returns the template number and smart update result.

Otherwise: RET returns ERR_VERIFY.

[Command and Response]

Table 4-24. Command CMD_VERIFY

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0064	
LEN	4	
DATA	2bytes	Template number to be matched
	2bytes	Ram Buffer ID
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0064	
LEN	Success : 5, Failure : 2	
RET	Result Code	
DATA	3bytes	Success: Template number (2bytes) + smart update result (1: smart update has been performed, 0: no update)

Example 4.24- 1:1 matching the fingerprint template in RamBuffer0 with the 8-th fingerprint in the database.

Host command: 55 AA 00 00 64 00 04 00 08 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 6F 01

Module response: AA 55 01 00 64 00 05 00 00 00 08 00 01 00 00 00 00 00 00 00 00 00 00 00 72 01

4.25 Set Module Serial Number (CMD_SET_MODULE_SN 0x0008)

[Function]

The module receives the module serial number (Module SN) sent from the host and saves it in the module. Module SN is 16 bytes.

[Sequence]

① The host sends the command packet to make the target enter the waiting mode for the data (Module SN) receiving. In the DATA field of the command packet, the length of the command data packet to be sent next time has been set.

② Target detects the correctness of the received command packet.

If not correct, returns an error code and ends processing.

If the size of the data to be received is incorrect, ERR_INVALID_PARAM is returned.

If it is correct, in order to inform that it has entered the waiting mode for the data (Module SN) receiving, it sends a response packet to the host and enters the waiting mode for the data (Module SN) receiving.

③ After the host receives the response packet, it sets the Module SN in the instruction packet and sends it to the target.

④ After the target receives the instruction packet, it sets Module SN in the module and returns the result.

[Command and Response]

Table4-25. Command CMD_SET_MODULE_SN

Command Response Packet	
PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID
CMD	0x0008
LEN	2
DATA	16 (data packet length of Module SN)
PREFIX	0x55AA
SID	Source Device ID
DID	Destination Device ID
RCM	0x0008
LEN	2
RET	Result Code
DATA	Not used

Command Data Packet	
PREFIX	0xA55A
SID	Source Device ID
DID	Destination Device ID
CMD	0x0008
LEN	16 (data packet length of Module SN)
DATA	Module SN(16bytes)
PREFIX	0x5AA5
SID	Source Device ID
DID	Destination Device ID
RCM	0x0008
LEN	2
RET	Result Code
DATA	Not used

Example 4.25- Set the module serial number to： IDWD2011-0123456

Host command： 55 AA 00 00 08 00 02 00 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 19 01

Module response: AA 55 01 00 08 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0A 01

Data packet： 5A A5 00 00 08 00 10 00 49 44 57 44 32 30 31 31 2D 30 31 32 33 34 35 36 95 04

Response End： A5 5A 01 00 08 00 02 00 00 00 0A 01

4.26 Read Module Serial Number (CMD_GET_MODULE_SN 0x0009)

[Function]

Send the Module SN stored in the module to the Host.

[Sequence]

① In the form of a command response packet, specify the size of the Module SN to be received by the HOST as the data transmission request.

② The Module SN stored in the module is sent using the response data packet.

[Command and Response]

Table4-26. Command CMD_GET_MODULE_SN

PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID
CMD	0x0009
LEN	0
DATA	Not used
PREFIX	0x55AA
SID	Source Device ID

DID	Destination Device ID
RCM	0x0009
LEN	4
RET	Result Code
DATA	Success: The data length of the next data response packet (Module SN Size(16)) Failure: error code

Response packet on success	
PREFIX	0x5AA5
SID	Source Device ID
DID	Destination Device ID
RCM	0x0009
LEN	Module SN Size(16)
RET	ERR_SUCCESS
DATA	Module SN(16bytes)

Example 4.26- Read the module serial number, the serial number obtained is：IDWD2011-0123456

Host command：55 AA 00 00 09 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 08 01

Module response: AA 55 01 00 09 00 04 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00 00 00 1D 01

The module serial number packet received by the host:

A5 5A 01 00 09 00 12 00 00 00 49 44 57 44 32 30 31 31 2D 30 31 32 33 34 35 36 99 04

4.27 Get Enrolled ID List (CMD_GET_ENROLLED_ID_LIST 0x0049)

[Function]

Send the ID list information registered in the module to the host.

The ID list information structure is as follows:

Each bit of each byte represents the enrollment status of the x-th fingerprint (x = byte number (starting from 0) * 8 + bit number (starting from 0)).

If it is 0, it means it was not enrolled. If it is 1, it means it has been already enrolled.

For example, assuming that the second byte of ID list information is 0100 0001 (binary), the meaning of each bit is as follows:

0-th bit from right (1): 8*2+0 = 16 (the 16-th fingerprint was enrolled)

1-th bit from right (0): 8*2+1 = 17 (the 17-th fingerprint was not enrolled)

…

6-th bit from right (1): 8*2+6 = 22 (the 22-th fingerprint was enrolled)

7-th bit from right (0): 8*2+7 = 23 (the 23-th fingerprint was not enrolled)

[Sequence]

① Set the size of the ID list information to be received by the host in the form of a command response packet as the response to the data transmission request.

② Send the enrolled ID list information in the module with response data packet.

[Command and Response]

Table 4-27. Command CMD_GET_ENROLLED_ID_LIST

PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID
CMD	0x0049
LEN	0
DATA	Not used
PREFIX	0x55AA
SID	Source Device ID
DID	Destination Device ID
RCM	0x0049
LEN	4
RET	Return Code
DATA	Success：data length of the next data response packet (ID List Information Size) Fail: Error code
Response packet on success	
PREFIX	0x5AA5
SID	Source Device ID
DID	Destination Device ID
RCM	0x0049
LEN	ID List Information Size
RET	ERR_SUCCESS
DATA	ID List Information

Example 4.27 – Command to get the enrolled User List within ID = 1~490.

Host command： 55 AA 00 00 49 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 48 01

Module response： AA 55 01 00 49 00 04 00 00 00 41 00 00 00 00 00 00 00 00 00 00 00 00 00 8E 01

Command data packet：

A5 5A 01 00 49 00 43 00 00 00 FE FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF 07 00 00 00 55 3E

In User ID List:

Byte 0: 0xFE = 1111 1110, ID = 0 not enrolled，ID = 1~7 already enrolled

Bit 0 (0*8+0) = 0，ID=0 not enrolled

Bit 1 (0*8+1) = 1，ID=1 already enrolled

…

Bit 7 (0*8+7) = 1，ID=7 already enrolled

Byte 1: 0xFF = 1111 1111, ID = 8~15already enrolled

Bit 8 (1*8+0) = 1，ID=8 already enrolled

Bit 9 (1*8+1) = 1，ID=9 already enrolled

…

Bit 15 (1*8+7) = 1，ID=15 already enrolled

…

Byte 61: 0x07 = 0000 0111:

Bit 488 (61*8+0=488) = 1，ID=488 already enrolled

Bit 489 (61*8+1=489) = 1，ID=489 already enrolled

Bit 490 (61*8+2=490) = 1，ID=490 already enrolled

Bit 491 (61*8+3=491) = 0，ID=491 not enrolled

…

Bit 495 (61*8+7=495) = 0，ID=495 not enrolled

…

Byte 62: 0x00 = 0000 0000:

Bit 496 (62*8+0=496) = 0，ID = 496 not enrolled

Bit 497 (62*8+1=497) = 0，ID = 497 not enrolled

Bit 498 (62*8+2=498) = 0，ID = 498 not enrolled

Bit 499 (62*8+3=497) = 0，ID = 499 not enrolled

Bit 500 (62*8+4=500) = 0，ID = 500 not enrolled

…

4.28 Enter Standby State (CMD_ENTER_STANDBY_STATE 0x000C)

[Function]

Put the module into standby state.

[Sequence]

After the module receives the command packet, it returns ERR_SUCCESS and enters the sleep state.

[Command and Response]

Table 4-28 Command CMD_ENTER_STANDBY_STATE

PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID
CMD	0x000C
LEN	0
DATA	No data
PREFIX	0x55AA

SID	Source Device ID
DID	Destination Device ID
RCM	0x000C
LEN	Success: 2
RET	0
DATA	No data

Example 4.28- Put the module into sleep state

Host command：55 AA 00 00 0C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0B 01

Module response：AA 55 01 00 0C 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0E 01

Notes：

1. **It is recommended to send this command (CMD_ENTER_STANDBY_STATE) before turning off the power of the module to make the module enter the standby state and then turn off the power of the module.**
2. **Some models of modules do not support the sleep function.**

4.29 Automatically adjust the fingerprint sensor. (**CMD_ADJUST_SENSOR 0x0025**)

[Function]

Automatically adjust the register parameters of the collector to obtain a fingerprint image with better results.

[Sequence]

Adjust the sensor and return its result.

[Command and Response]

Table 4-29 Command CMD_ADJUST_SENSOR

PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID
CMD	0x0025
LEN	0
DATA	No data
PREFIX	0x55AA
SID	Source Device ID
DID	Destination Device ID
RCM	0x0025
LEN	Success: 2
RET	0
DATA	No data

Note: The sensors modules of some models do not support this operation, and execution of this command has no effect.

Example 4.29- Automatically adjust the fingerprint sensor.

Host command：55 AA 00 00 25 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 24 01

Module response：AA 55 01 00 25 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 27 01

4.30 Communication error return value (Incorrect Command)

[Function]

If the module receives an incorrect command due to communication errors, the response is sent to the HOST.

[Command and Response]

Table 4-30 Incorrect Command return packet

PREFIX	0x55AA
SID	Source Device ID
DID	Destination Device ID
RCM	0x00FF
LEN	2
RET	ERR_SUCCESS
DATA	-

4.31 Notes

① CMD_GENERATE command is generate Template Data from ImageBuffer.
Therefore, before calling this command, you need to call the CMD_GET_IMAGE command in advance to save the image in the ImageBuffer.

② Call CMD_VERIFY, CMD_SEARCH, CMD_GENERATE, CMD_MERGE, CMD_MATCH commands,
The image saved in ImageBuffer will be cleared.

③ The Template stored in Ram Buffer2 will be cleared by calling CMD_SEARCH, CMD_VERIFY, CMD_GENERATE, CMD_STORE_CHAR, CMD_DEL_CHAR, CMD_GET_EMPTY_ID, CMD_GET_STATUS, GET_BROKEN_ID, CMD_GETN_ENROLL_COUNT commands.

Therefore， except enroll, do not use Ram Buffer2.

5. Response & Error Code Table

No	Response & error code	Value	Description
1	ERR_SUCCESS	0x00	Command process success.
2	ERR_FAIL	0x01	Command process failure
3	ERR_VERIFY	0x10	Template 1:1 verify failure in specified number range
4	ERR_IDENTIFY	0x11	1:1 Verify completed, but no exist matched template
5	ERR_TMPL_EMPTY	0x12	No exist enrolled template in specified number range
6	ERR_TMPL_NOT_EMPTY	0x13	No exist template in specified number range
7	ERR_ALL_TMPL_EMPTY	0x14	No exist enrolled template 。
8	ERR_EMPTY_ID_NOEXIST	0x15	No enroll able template ID
9	ERR_BROKEN_ID_NOEXIST	0x16	No damaged template.
10	ERR_INVALID_TMPL_DATA	0x17	Invalid specified template data。
11	ERR_DUPLICATION_ID	0x18	Already enrolled fingerprint
12	ERR_BAD_QUALITY	0x19	The fingerprint image quality is not good
13	ERR_MERGE_FAIL	0x1A	Template merge failure
14	ERR_NOT_AUTHORIZED	0x1B	No communication password confirmation. Notes： ①If the communication password has been set, but CMD_VERIFY_DEVPASS has not been called for confirmation, all commands except CMD_TEST_CONNECTION and CMD_VERIFY_DEVPASS will return this error code. ② If no communication password is set, all commands can be used without confirming the password.
15	ERR_MEMORY	0x1C	External flash write error
16	ERR_INVALID_TMPL_NO	0x1D	Invalid Specified template number
17	ERR_INVALID_PARAM	0x22	Used incorrect parameters
18	ERR_GEN_COUNT	0x25	Invalid fingerprint merge number
19	ERR_TIME_OUT	0x23	No fingerprint during TimeOut time
20	ERR_INVALID_BUFFER_ID	0x26	Incorrect buffer ID value
21	ERR_FP_NOT_DETECTED	0x28	No fingerprint on collector
22	ERR_FP_CANCEL	0x41	Canceled command

6. Fingerprint Enroll & Verify Process

6.1 Fingerprint Enroll Process

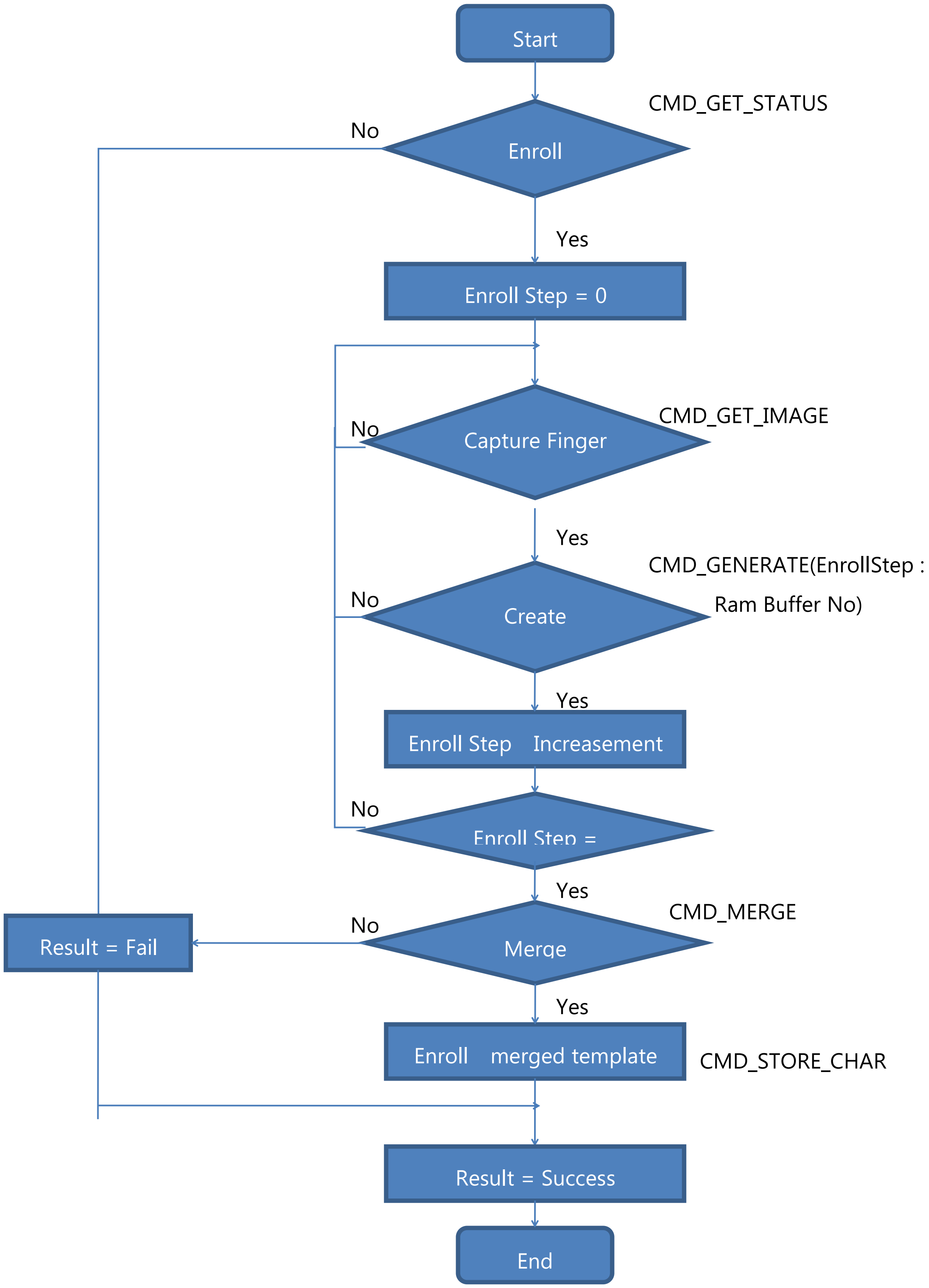

Fig.6-1: Fingerprint enroll process

6.2 Fingerprint Verify & Identify Process

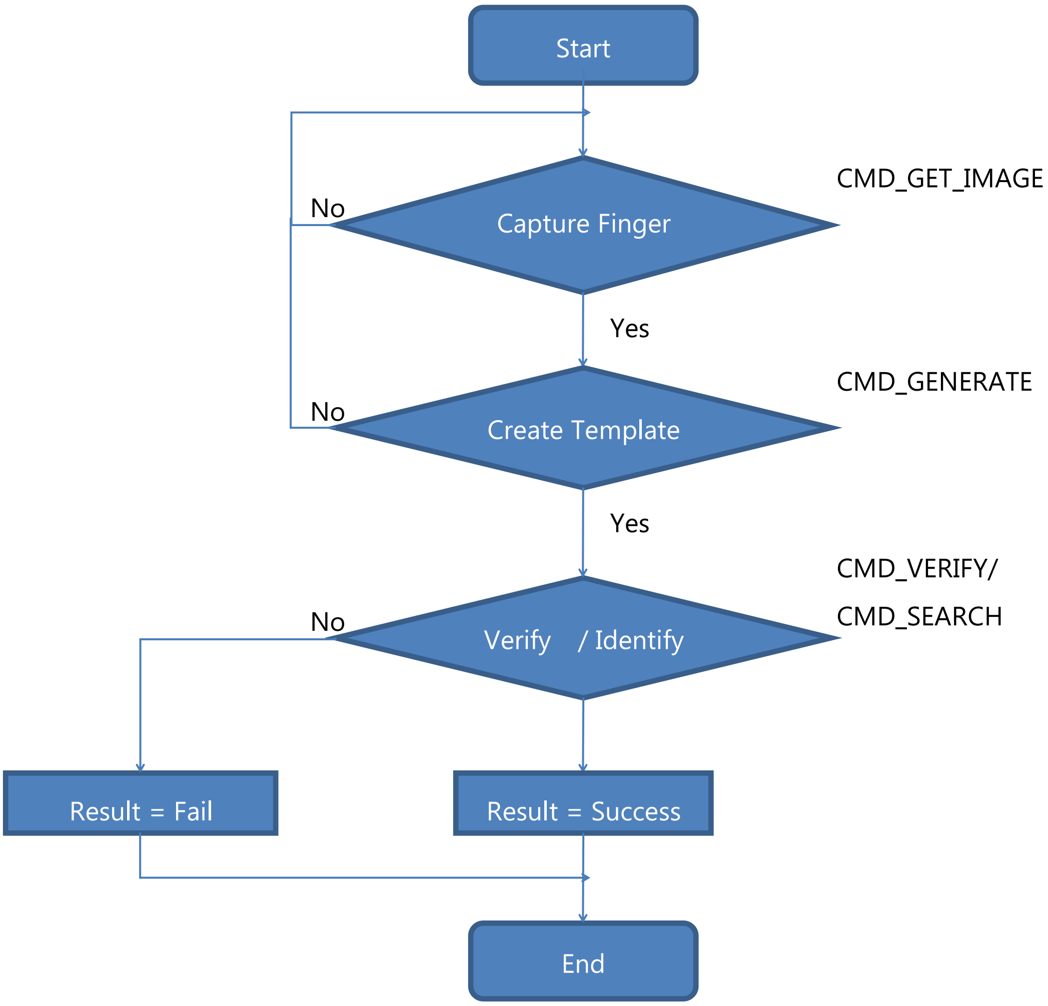

Fig.6-2: Fingerprint verify & identify process