

UART Fingerprint Reader User Manual

Contents

UART Fingerprint Reader User Manual.....	1
Overview.....	2
Fingerprint Module Development Protocol.....	3
1. Communication.....	3
2. Description of each communication protocol command.....	4
A. Appendix.....	15
A.1 Add fingerprint process.....	15
A.2 Delete specified user processes.....	16
A.3 Delete all users process.....	16

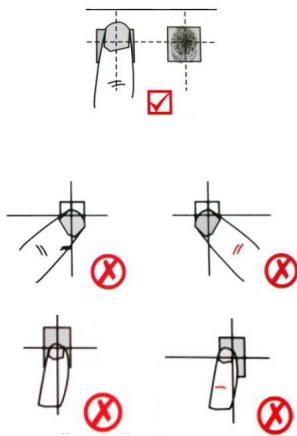
Overview

Note:

Module mentioned in this manual are using imported high-precision components, when gathering the fingerprint, your fingers only need to touch the acquisition area gently, no need to push it down, fingerprint module can be identified quickly.

If you want to power by battery, it is recommended to use alkaline batteries.

The correct use of the fingerprint module:



Hardware connection:

VCC ----- 3.3V or 5V

GND ----- GND

TXD (serial port of fingerprint module send) ----- RXD (serial port of PC or microcontroller receive)

RXD (serial port of fingerprint module receive) ----- TXD (serial port of PC or microcontroller send)

BL (fingerprint of backlight, connection is not a must) ----- IO port

RST (fingerprint module reset, connection is not a must) ----- IO port

After get the module, you can first use the test software (\UART-Fingerprint-Reader\software\Demo Software\ UART Fingerprint Reader.exe) to test the module.

After the hardware connection is completed, run (\UART-Fingerprint-Reader\software\Demo Software\Register Controls. bat) registration control first, and then open (\UART-Fingerprint-Reader\software\Demo Software\ UART Fingerprint Reader.exe)

For details, please refer to (\UART-Fingerprint-Reader\software\Demo Software\ UART Fingerprint Reader.exe))

After get a certain understanding of the module, the below development protocol can be used for secondary development.

Fingerprint Module Development Protocol

1. Communication

DSP module works as a slave device, the master device control it by sending related commands.
 Command interface: 19200bps; 1 start bits; 1 stop bits (parity bits: none)

Commands sent by master device and DSP module response can be divided into two categories according to data length:

1) = 8 bytes, data format as below:

Byte	1	2	3	4	5	6	7	8
Command	0xF5	CMD	P1	P2	P3	0	CHK	0xF5
Respond	0xF5	CMD	Q1	Q2	Q3	0	CHK	0xF5

Note:

CMD: Command / response type

P1, P2, P3: Command parameter

Q1, Q2, Q3: Response parameter

Q3 mainly used to return the effective operating information, there will be the following values:

```
#define ACK_SUCCESS    0x00    //Operation successfully
#define ACK_FAIL       0x01    // Operation failed
#define ACK_FULL       0x04    // Fingerprint database is full
#define ACK_NOUSER     0x05    //No such user
#define ACK_USER_EXIST 0x06    // User already exists
#define ACK_FIN_EXIST  0x07    // Fingerprint already exists
#define ACK_TIMEOUT    0x08    // Acquisition timeout
```

CHK: checksum value, XOR value for the second byte to the sixth byte

2) > 8 bytes, data includes two parts: data head + data package

Data header format:

Byte	1	2	3	4	5	6	7	8
Command	0xF5	CMD	Hi(Len)	Low(Len)	0	0	CHK	0xF5
Response	0xF5	CMD	Hi(Len)	Low(Len)	Q3	0	CHK	0xF5

Note:

CMD: Q3 definition as the above.

Len: Effective length of the data is 16 bits, consists two bytes

Hi (Len): Data packet length high 8-bit

Low (Len): Data packet length low 8-bit

CHK: checksum value, XOR value for the second byte to the sixth byte

Data packet format:

Byte	1	2...Len + 1	Len + 2	Len + 3
Command	0xF5	Data	CHK	0xF5
Response	0xF5	Data	CHK	0xF5

Note:

Len is the number of byte of the Data;

CHK: checksum value, XOR value for the second byte to the Len+1 byte

Send data packet immediately after send the data header.

2. Description of each communication protocol command

2.1 Enable the module into a dormant state (Both command and response are 8 bytes)

Command data format:

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x2C	0	0	0	0	CHK	0xF5

Response data format:

Byte	1	2	3	4	5	6	7	8
Response	0xF5	0x2C	0	0	0	0	CHK	0xF5

2.2 Set / read the fingerprint add mode (Both command and response are 8 bytes)

There are two modes for adding fingerprint: Allow repeat mode / prohibit repeat mode, in the "prohibit repeat mode", the same finger can add one user only, if forced to add a second user will return an error message. After power, the system is in prohibiting repeat mode.

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x2D	0	Byte5=0: 0: allow repeat 1: prohibit repeat Byte5=1: 0	0: set new add mode 1: read current add mode	0	CHK	0xF5
Response	0xF5	0x2D	0	Current add mode	ACK_SUCCUSS ACK_FAIL	0	CHK	0xF5

2.3 Add fingerprint (Both command and response are 8 bytes)

To ensure the effectiveness, user must input a fingerprint three times, the host is required to send command to the DSP module three times.

(1) The first time

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x01	User ID(high 8-bit)	User ID(low 8-bit)	User privilege(1/2/3)	0	CHK	0xF5
Response	0xF5	0x01	0	0	ACK_SUCCESS ACK_FAIL ACK_FULL ACK_TIMEOUT	0	CHK	0xF5

Note:

Range of user number is 1 - 0xFFF;

Range of User privilege is 1, 2, 3, its meaning is defined by secondary developers themselves.

(2) The second time:

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x02	User ID(high 8-bit)	User ID(low 8-bit)	User privilege(1/2/3)	0	CHK	0xF5
Response	0xF5	0x02	0	0	ACK_SUCCESS ACK_FAIL ACK_TIMEOUT	0	CHK	0xF5

(3) The third time:

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x03	User ID(high 8-bit)	User ID(low 8-bit)	user privilege(1/2/3)	0	CHK	0xF5
Response	0xF5	0x03	0	0	ACK_SUCCESS ACK_FAIL ACK_USER_EXIST ACK_TIMEOUT	0	CHK	0xF5

Note: User ID and user privilege should be in the same value in the three commands.

2.4 Delete specified user (Both command and response are 8 bytes)

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x04	User ID(high 8-bit)	User ID(low 8-bit)	0	0	CHK	0xF5
Response	0xF5	0x04	0	0	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5

2.5 Delete all users (Both command and response are 8 bytes)

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x05	0	0	0	0	CHK	0xF5
Response	0xF5	0x05	0	0	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5

2.6 Acquire the total number of users (Both command and response are 8 bytes)

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x09	0	0	0	0	CHK	0xF5
Response	0xF5	0x09	User number(high 8-bit)	User number(low 8-bit)	ACK_SUCCESS	0	CHK	0xF5

						ACK_FAIL			
--	--	--	--	--	--	----------	--	--	--

2.7 Compare 1:1 (Both command and response are 8 bytes)

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x0B	User ID(high 8-bit)	User ID(low 8-bit)	0	0	CHK	0xF5
Response	0xF5	0x0B	0	0	ACK_SUCCESS ACK_FAIL ACK_TIMEOUT	0	CHK	0xF5

2.8 Compare 1: N (Both command and response are 8 bytes)

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x0C	0	0	0	0	CHK	0xF5
Response	0xF5	0x0C	User ID(high 8-bit)	User ID(low 8-bit)	user privilege(1/2/3) ACK_NOUSER ACK_TIMEOUT	0	CHK	0xF5

2.9 Acquire user privilege (Both command and response are 8 bytes)

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x0A	User ID(high 8-bit)	User ID(low 8-bit)	0	0	CHK	0xF5
Response	0xF5	0x0A	0	0	user privilege(1/2/3) ACK_NOUSER	0	CHK	0xF5

2.10 Acquire DSP module version number (command = 8 bytes, and response > 8 bytes)

Command data format:

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x26	0	0	0	0	CHK	0xF5

Response data format:

1) Data header:

Byte	1	2	3	4	5	6	7	8
Respond	0xF5	0x26	Hi(Len)	Low(Len)	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5

2) Data packet:

Byte	1	2 --- Len + 1	Len + 2	Len + 3

Response	0xF5	Version data	CHK	0xF5
----------	------	--------------	-----	------

Note: This protocol won't public currently.

2.11 Set/ read comparison level (Both command and response are 8 bytes)

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x28	0	Byte5=0: New comparison level; Byte5=1: 0	0:Set new comparison level; 1:Read current comparison level	0	CHK	0xF5
Response	0xF5	0x28	0	Current comparison level	ACK_SUCCUSS ACK_FAIL	0	CHK	0xF5

Note: Range of the comparison level is 0-9, the greater the value, the more strict in comparison, default value is 5.

2.12 Acquire and upload images (Command = 8 bytes, response > 8 bytes)

Command data format:

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x24	0	0	0	0	CHK	0xF5

Response data format:

1) Data header:

Byte	1	2	3	4	5	6	7	8
Response	0xF5	0x24	Hi(Len)	Low(Len)	ACK_SUCCESS ACK_FAIL ACK_TIMEOUT	0	CHK	0xF5

2) Data packet:

Byte	1	2 --- Len + 1	Len + 2	Len + 3
Response	0xF5	Image data	CHK	0xF5

Note:

In DSP module, the fingerprint image is 248*296 pixels, grayness of each pixel is represented by 8 bits. During the upload process, in order to reduce the amount of data, jump pixel sampling in the horizontal / vertical direction, so that the image becomes 124*148, and take the grayness for high 4-bit, each two pixels composited into one byte for transferring (previous pixel low 4-bit, last pixel high 4-bit).

Transmission starts line by line from the first line, each line starts from the first pixel, totally transfer 124* 148/ 2 bytes of data.

Data length of image is fixed of 9176 bytes.

2.13 Upload acquired images and extracted eigenvalue (Command = 8 bytes, and response > 8 bytes)

Command data format:

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x23	0	0	0	0	CHK	0xF5

Response data format:

1) Data header:

Byte	1	2	3	4	5	6	7	8
Response	0xF5	0x23	Hi(Len)	Low(Len)	ACK_SUCCESS ACK_FAIL ACK_TIMEOUT	0	CHK	0xF5

2) Data packet:

Byte	1	2	3	4	5 --- Len + 1	Len + 2	Len + 3
Response	0xF5	0	0	0	Eigenvalues data	CHK	0xF5

Note: Eigenvalues data length Len-3 is fixed 193 bytes.

2.14 Download eigenvalues and acquire fingerprint comparison (Command > 8 bytes, response = 8 bytes)

Command data format:

1) Data header:

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x44	Hi(Len)	Low(Len)	0	0	CHK	0xF5

2) Data packet

Byte	1	2	3	4	5 --- Len + 1	Len + 2	Len + 3
Command	0xF5	0	0	0	Eigenvalues data	CHK	0xF5

Note: Eigenvalues data length Len-3 is fixed 193 bytes.

Response data format:

Byte	1	2	3	4	5	6	7	8
Response	0xF5	0x44	0	0	ACK_SUCCESS ACK_FAIL ACK_TIMEOUT	0	CHK	0xF5

2.15 Download the fingerprint eigenvalues and DSP module database fingerprint compare 1: 1 (command>8 bytes/response=8 bytes)

Command data format:

1) Data header:

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x42	Hi(Len)	Low(Len)	0	0	CHK	0xF5

2) Data packet:

Byte	1	2	3	4	5 --- Len + 1	Len + 2	Len + 3
Command	0xF5	User ID(high 8-bit)	User ID(low 8-bit)	0	Eigenvalues data	CHK	0xF5

Note: Eigenvalues data length Len-3 is fixed 193 bytes.

Response data format:

Byte	1	2	3	4	5	6	7	8
Response	0xF5	0x42	0	0	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5

2.16 Download the fingerprint eigenvalues and DSP module database fingerprint compare 1: N (command>8 bytes / response=8 bytes)

Command data format:

1) Data header:

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x43	Hi(Len)	Low(Len)	0	0	CHK	0xF5

2) Data packet:

Byte	1	2	3	4	5 --- Len + 1	Len + 2	Len + 3
Command	0xF5	0	0	0	Eigenvalues data	CHK	0xF5

Note: Eigenvalues data length Len-3 is fixed 193 bytes.

Response data format:

Byte	1	2	3	4	5	6	7	8
Response	0xF5	0x43	User ID(high 8-bit)	User ID(low 8-bit)	User privilege(1/2/3)	0	CHK	0xF5
					ACK_NOUSER			

2.17 Upload the DSP module database specified user eigenvalue (command = 8 bytes, response > 8 bytes)

Command data format:

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x31	User ID(high 8-bit)	User ID(low 8-bit)	0	0	CHK	0xF5

Response data format:

1) Data header:

Byte	1	2	3	4	5	6	7	8
Response	0xF5	0x31	Hi(Len)	Low(Len)	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5

					ACK_NOUSER			
--	--	--	--	--	------------	--	--	--

2) Data packet:

Byte	1	2	3	4	5 --- Len + 1	Len + 2	Len + 3
Response	0xF5	User ID(high 8-bit)	User ID(low 8-bit)	User privilege(1/2/3)	Eigenvalues data	CHK	0xF5

Note: Eigenvalues data length Len-3 is fixed 193 bytes.

2.18 Download the eigenvalue and save to the DSP module database according to the specified user number (command>8 bytes/response=8 bytes)

Command data format:

1) Data header:

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x41	Hi(Len)	Low(Len)		0	CHK	0xF5

2) Data packet:

Byte	1	2	3	4	5--- Len + 1	Len + 2	Len + 3
Command	0xF5	User ID(high 8-bit)	User ID(low 8-bit)	User privilege(1/2/3)	Eigenvalues data	CHK	0xF5

Note: Eigenvalues data length Len-3 is fixed 193 bytes.

Response data format:

Byte	1	2	3	4	5	6	7	8
Response	0xF5	0x41	User ID(high 8-bit)	User ID(low 8-bit)	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5

2.19 Acquire all logged in user numbers and user privilege (command = 8 bytes, response > 8 bytes)

Command data format:

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x2B	0	0	0	0	CHK	0xF5

Response data format:

1) Data header:

Byte	1	2	3	4	5	6	7	8
Response	0xF5	0x2B	Hi(Len)	Low(Len)	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5

2) Data packet:

Byte	1	2	3	4 --- Len + 1	Len + 2	Len + 3
------	---	---	---	---------------	---------	---------

Response	0xF5	User number(high 8-bit)	User number(low 8-bit)	User information data(User ID and privilege)	CHK	0xF5
----------	------	-------------------------	------------------------	---	-----	------

Note:

Data length Len in the Data packet is fixed “3”, the user number is fixed “+2”.

User information data format as below:

Byte	4	5	6	7	8	9	...
Data	User ID1(high 8-bit)	User ID1(high 8-bit)	User ID1(high 8-bit)	User ID2(high 8-bit)	User ID2(high 8-bit)	User ID2(high 8-bit)	...

2.20 Acquire a single record data (command = 8 bytes, response >8 bytes)—Note: The module won’t provide the protocol currently.

This protocol return data records in the library which is designated by the "record location".

Command data format:

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x38	Record location(high 8-bit)	Record location(low 8-bit)	0	0	CHK	0xF5

Response data format:

1) Data header:

Byte	1	2	3	4	5	6	7	8
Response	0xF5	0x38	Hi(Len)	Low(Len)	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5

2) Data packet:

Byte	1	2	3	4	5
Response	0xF5	Digit 7-1:Year Digit:0: Month(Digit 3)	Digit:7-5:Month(Digit2-0) Digit 4-0:Date	Digit 7-2: Hour Digit 1-0: Minute(Digit 5-4)	Digit 7-4: Minute(Digit 3-0) Digit 3-0: Record No.(Digit 21-18)

Byte	6	7	8	9	10	11
Response	Record No.(Digit 17-10)	Record No. (Digit 9-2)	Digit 7-6: Record No.(Digit 1-0) Digit 5-0: User ID(Digit 13-8)	User ID(Digit 7-0)	CHK	0xF5

Note:

Record data length Len is fixed 8;

2.21 Acquire new record data (command = 8 bytes, response > 8 bytes) – Note: The module won’t provide the protocol currently.

This protocol returns 50 continuous record data which is greater or equal to "minimum record number" in the record library.

Command data format:

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x39	Digit 7-6:0 Digit 5-0:Minimum Record No.(Digit 21-16)	Minimum Record No.(Digit 15-8)	Minimum Record No.(Digit 7-0)	0	CHK	0xF5

Response data format:

1) Data header:

Byte	1	2	3	4	5	6	7	8
Response	0xF5	0x39	Hi(Len)	Low(Len)	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5

2) Data packet:

Byte	1	2---9	10-17	...	Len + 2	Len + 3
Response	0xF5	The first record	The second record	...	CHK	0xF5

Note:

Format of each record in the data packet is the same as byte 2-byte 9 in response data packet in above 2.20.

Data length Len is fixed (8*50=400 bytes).

2.22 Wipe the data record (Both command and response are 8 bytes)—Note: The module won't provide the protocol currently.

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x3A	0	0	0	0	CHK	0xF5
Response	0xF5	0x3A	0	0	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5

2.23 Set Module time (command > 8 bytes, response = 8 bytes) -- Note: The module won't provide the protocol currently.

1) Data header:

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x48	Hi(Len)	Low(Len)	0	0	CHK	0xF5

3) Data packet:

Byte	1	2	3	4	5	6	7	8	9	10
Command	0xF5	Week	Year	Month	Date	Hour	Minute	second	CHK	0xF5

Note:

Time data length Len is fixed 7.

Response data format:

Byte	1	2	3	4	5	6	7	8
Response	0xF5	0x48	Hi(Len)	Low(Len)	0	0	CHK	0xF5

Response	0xF5	0x48	0	0	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5
----------	------	------	---	---	-------------------------	---	-----	------

2.24 Read system time -- Note: The module won't provide the protocol currently.

Command data format:

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x3C	0	0	0	0	CHK	0xF5

Response data format:

1) Data header:

Byte	1	2	3	4	5	6	7	8
Response	0xF5	0x3C	Hi(Len)	Low(Len)	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5

2) Data packet:

Byte	1	2	3	4	5	6	7	8	9	10
Response	0xF5	Week	Year	Month	Date	Hour	Minute	Second	CHK	0xF5

Note:

Time data length Len is fixed 7.

2.25 Set/read fingerprint capture timeout value (Both command and response are 8 bytes)

Byte	1	2	3	4	5	6	7	8
Command	0xF5	0x2E	0	Byte5=0: New timeout value; Byte5=1: 0	0: Set new timeout value 1: Read current timeout value	0	CHK	0xF5
Response	0xF5	0x2E	0	Current timeout value	ACK_SUCCUSS ACK_FAIL	0	CHK	0xF5

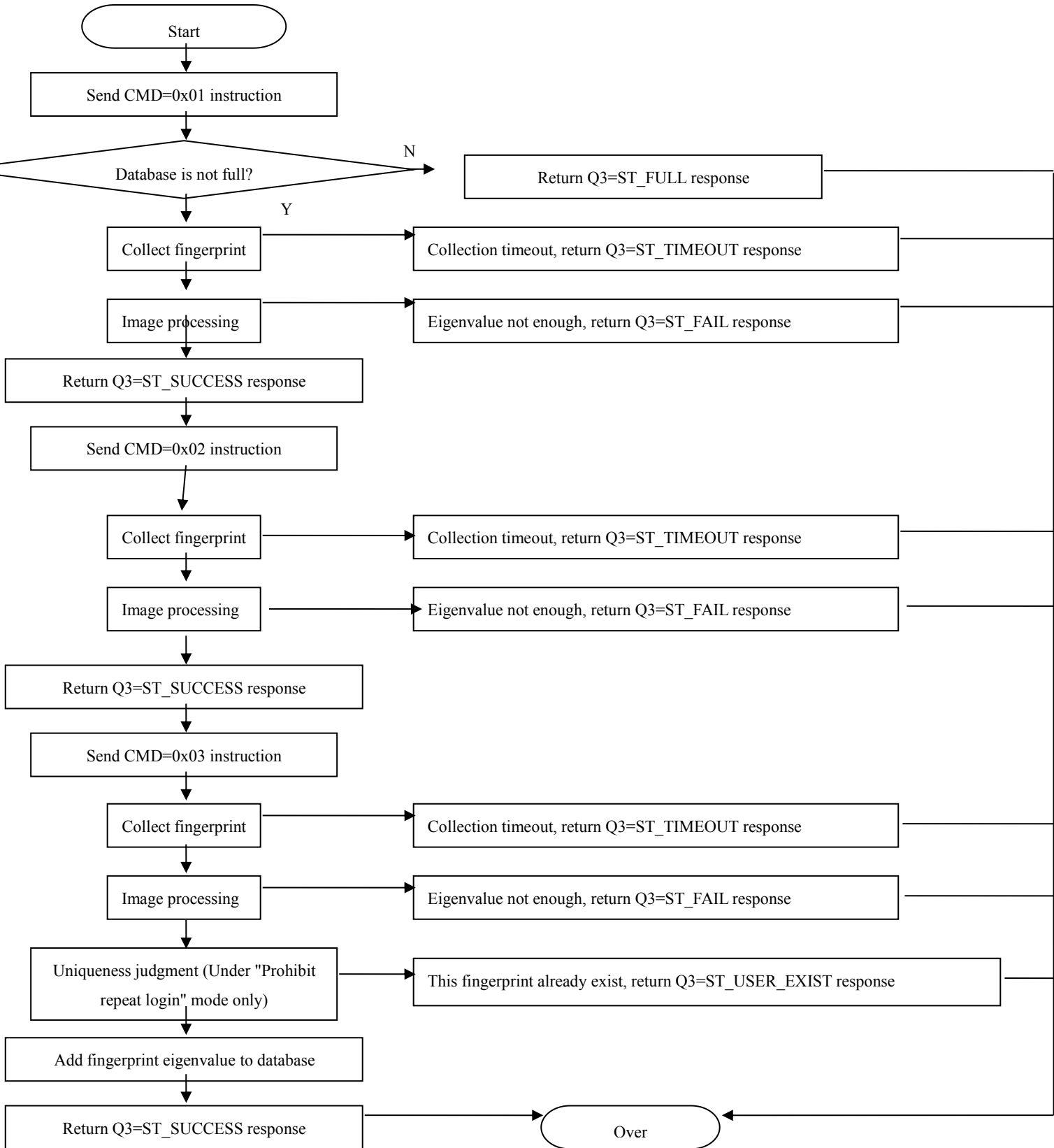
Note:

Range of fingerprint waiting timeout (tout) value is 0-255. If the value is 0, the fingerprint acquisition process will keep continue if no fingerprints press on; If the value is not 0, the system will exist for reason of timeout if no fingerprints press on in time tout * T0.

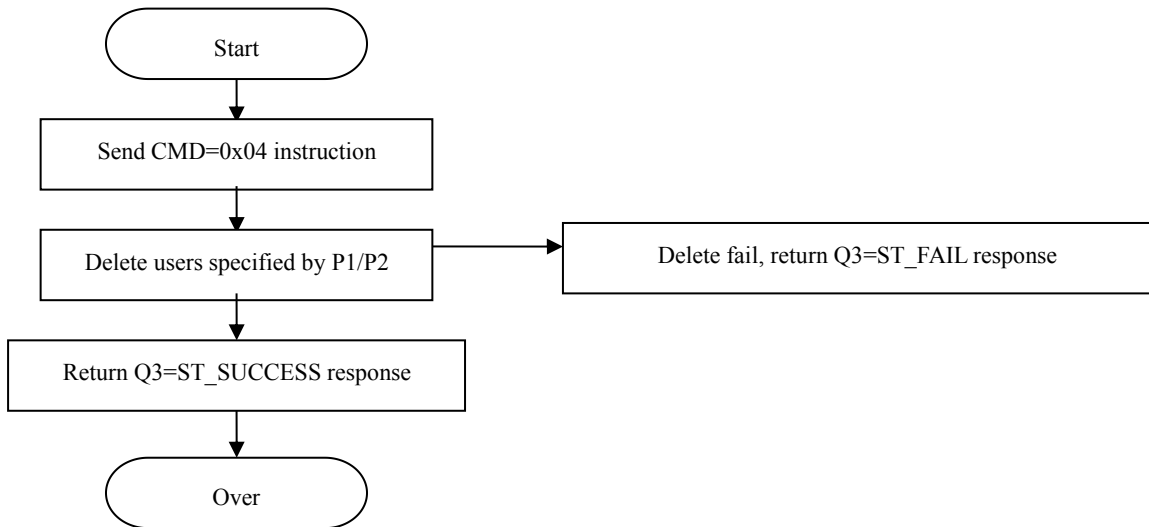
Note: T0 is the time required for collecting/processing an image, usually 0.2- 0.3 s.

A. Appendix

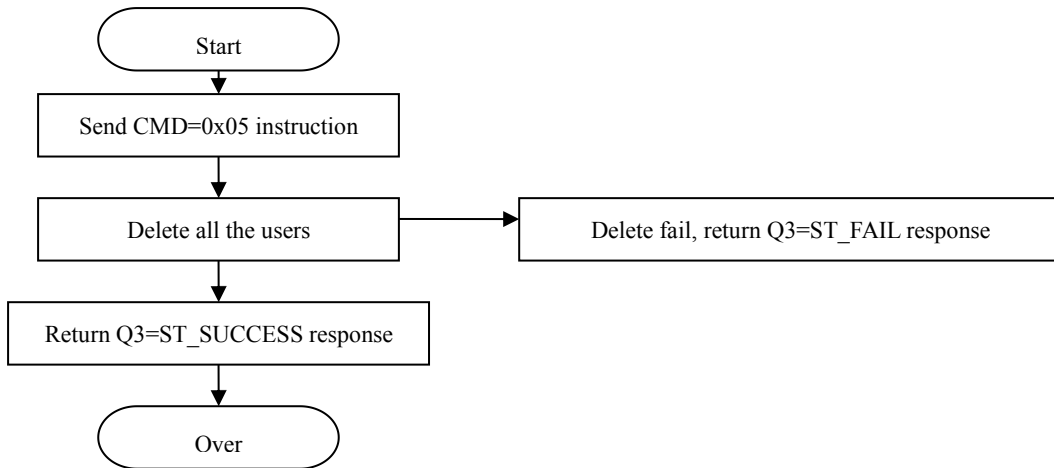
A.1 Add fingerprint process



A.2 Delete specified user process



A.3 Delete all users process



A.4 Upload collected image and extracted eigenvalue process

