Capacitive Fingerprint Reader Development Manual

CONTENT

DEVELOPMENT SERIAL PORT'S CONTROL PROTOCOL

COMMUNICATION MODE

DSP module as a slave device receives commands sent by master which controls the module

The commends sent by master and the length of data sent by DSP as a response divided on 2 categories:

1. =8 bytes with data format:

byte	1	2	3	4	5	6	7	8
command	0xF5	CMD	P1	P2	P3	0	CHK	0xF5
response	0xF5	CMD	Q1	Q2	Q3	0	CHK	0xF5

Description:

CMD: command/the type of response

P1, P2, P3: command arguments

Q1, Q2, Q3: response arguments

Q3: is usually used to return the efficiency of the result and can be one of following values:

```
#define   ACK_SUCCESS      0x00    // successful
#define   ACK_FAIL         0x01    // fail
#define   ACK_FULL         0x04    // the fingerprint database is full
#define   ACK_NOUSER       0x05    // there is no such user
#define   ACK_USER_EXIST   0x06    // the user exists already
#define   ACK_TIMEOUT      0x08    // timeout of scanning
```

CHK: checksum, XOR values of bytes 2-6

2. >8 bytes, data consists of two parts: header + packet

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

Description:

CMD, Q3 are same as described above

Len： 16-bits number containing the length of the packet divided on two parts:

Hi(Len)： High 8 bits of packet length's number

Low(Len)： Low 8 bits of packet length's number

CHK: checksum, XOR values of bytes 2-6

Data packet format:

byte	1	2…Len + 1	Len + 2	Len + 3
command	0xF5	Data	CHK	0xF5
response	0xF5	Data	CHK	0xF5

Description:

Len: count of bytes in Data

CHK: checksum based on bytes from 2 to Len – 2 and containing XOR values

The packet is sent immediately after the header.

COMMUNICATIONS PROTOCOL COMMAND DESCRIPTION

1. Switching the module to sleep mode (8 bytes command/response)

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. Setting/reading the mode of fingerprint adding (8 bytes command/response)

The fingerprint can be added in two modes: "Enable repeat mode" and "Disable repeat mode", in the "Disable repeat mode", only the same fingerprint can only be added one time. After adding a user, the second adding leads to an error message. After power-on the system is in the mode which forbids repeating.

byte	1	2	3	4	5	6	7	8
command	0xF5	0x2D	0	Byte5=0: 0: En. Repeat 1: Dis. Repeat	0: Set new adding mode	0	CHK	0xF5

				Byte5=1: 0	1: Reading current adding mode			
Response	0xF5	0x2D	0	Current adding mode	ACK_SUCCUSS ACK_FAIL	0	CHK	0xF5

3. Adding fingerprint (8 bytes command/response)

To be sure in efficiency of the adding, user has to input fingerprint 3 times and the host sends command to DSP module 3 times

1) The first time

byte	1	2	3	4	5	6	7	8
command	0xF5	0x01	user number (high 8 bits)	user number (Low 8 bits)	user permissions (1/2/3)	0	CHK	0xF5
response	0xF5	0x01	0	0	ACK_SUCCESS ACK_FAIL ACK_FULL AACK_TIMEOUT	0	CHK	0xF5

Description:

The user number is in the range 1-0xFFF

The user permissions are in range 1, 2 and 3, whose meaning should be defined by secondary developers.

2) The second time

byte	1	2	3	4	5	6	7	8
command	0xF5	0x02	user number (high 8 bits)	user number (low 8 bits)	user permissions (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 number (high 8 bits)	user number (low 8 bits)	user permissions (1/2/3)	0	CHK	0xF5
response	0xF5	0x03	0	0	ACK_SUCCESS ACK_FAIL ACK_USER_EXISTS ACK_TIMEOUT	0	CHK	0xF5

Comment: The user number and permission should be same in there times.

4. Removing the specified user (8 bytes command/response)

byte	1	2	3	4	5	6	7	8
command	0xF5	0x04	user number (high 8 bits)	user number (low 8 bits)	0	0	CHK	0xF5
response	0xF5	0x04	0	0	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5

5. Removing all users (8 bytes command/response)

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

6. Getting the total count of users (8 bytes command/response)

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 bits)	user number (low 9 bits)	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5

7. Comparison 1:1 (8 bytes command/response)

byte	1	2	3	4	5	6	7	8

command	0xF5	0x0B	user number (high 8 bits)	user number (low 8 bits)	0	0	CHK	0xF5
response	0xF5	0x0B	0	0	ACK_SUCCESS ACK_FAIL ACK_TIMEOUT	0	CHK	0xF5

8. Comparison 1:N (8 bytes command/response)

byte	1	2	3	4	5	6	7	8
command	0xF5	0x0C	0	0	0	0	CHK	0xF5
response	0xF5	0x0C	user number (high 8 bits)	user number (low 8 bits)	user permission (1/2/3) ACK_NOUSER ACK_FAIL	0	CHK	0xF5

9. Getting user permission (8 bytes command/response)

byte	1	2	3	4	5	6	7	8
command	0xF5	0x0A	user number (high 8 bits)	user number (low 8 bits)	0	0	CHK	0xF5
response	0xF5	0x0A	0	0	user permission (1/2/3) ACK_NOUSER	0	CHK	0xF5

10. Getting DSP module version (8 bytes command and >8 bytes response)

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) Header:

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

2) Packet:

byte	1	2---Len+1	Len+2	Len+3
response	0xF5	Data version	CHK	0xF5

Note: this protocol is not available temporary

11. Setting/getting comparison level (8 bytes command/response)

byte	1	2	3	4	5	6	7	8
command	0xF5	0x28	0	Byte 5=0: new comparison level Byte5=1: 0	0: setup new comparison level 1: read current comparison level	0	CHK	0xF5
response	0xF5	0x28	0	current comparison level	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5

Note: Comparison level value is in range 0-9, bigger value is compared more strongly. It equals to 5 by defual

12. Obtaining and sending of image (8 bytes command and >8 bytes response)

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) 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) Packet:

byte	1	2---Len+1	Len+2	Len+3
response	0xF5	image data	CHK	0xF5

Description:

In DSP module, the image resolution is 200x264 pixels. Each grayscale pixel takes 8 bits. In order to reduce the amount of data during the data sending, in vertical and horizontal directions is skipped sampled, thus the resolution becomes 100x132. Low 4 bits of grayscale values are dropped as well, thus every byte of data include two pixels (the first pixel is stored in low 4 bits, and second one is stored in high 4 bits).

Transmission of image is performed row by row. The total data length is 100x132/2 bytes

The length of image data in bytes is 6600.

Depending on specific product model this value can be different. Please contact our technical staff.

13. Image obtaining and characteristics sending (8 bytes command and >8 bytes response)

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) Header:

byte	1	2	3	4	5	6	7	8
response	0xF5	0x24	Hi(Len)	Low(Len)	ACK_SUCCESS ACK_FAIL ACK_TIMOUT	0	CHK	0xF5

2) Packet:

byte	1	2	3	4	5---Len+1	Len+2	Len+3

command	0xF5	0	0	0	data characteristics	CHK	0xF5

Description: the length Len-3 of data characteristics equals to 193 bytes

14. Loading characteristics and obtaining fingerprint comparison (>8 bytes command and 8 bytes response)

Command data format:

1) Header:

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

2) Packet:

byte	1	2	3	4	5---Len+1	Len+2	Len+3
command	0xF5	0	0	0	data characteristics	CHK	0xF5

Description: the length Len-3 of data characteristics equals to 193.bytes.

Response data format:

byte	1	2	3	4	5	6	7	8
response	0xF5	0x44	0	0	ACLK_SUCCESS ACK_FAIL ACK_TIMEOUT	0	CHK	0xF5

15. Loading fingerprint characteristics and DSO module database fingerprint comparison “1:1” (>8 bytes command and bytes response)

Command data format:

1) Header:

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

2) Packet:

byte	1	2	3	4	5---Len+1	Len+2	Len+3
command	0xF5	user number (high 8 bits)	user number (low 8 bits)	0	data characteristics	CHK	0xF5

Description: the length Len-3 of data characteristics equals to 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

16. Loading fingerprint characteristics and DSP module database fingerprint comparison "1:N" (>8 bytes command and 8 bytes response)

Command data format:

1) Header:

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

2) Packet:

byte	1	2	3	4	5---Len+1	7	8
command	0xF5	0	0	0	data characteristics	CHK	0xF5

Description: the length Len-3 of data characteristics equals to 193 bytes

Response data format:

byte	1	2	3	4	5	6	7	8
response	0xF5	0x43	user number (high 8 bits)	user number (low 8 bits)	user permission (1/2/3) ACK_NOUSER	0	CHK	0xF5

17. Uploading user characteristics from DSP module database (8 bytes command and >8 bytes response)

Command data format:

byte	1	2	3	4	5	6	7	8
command	0xF5	0x31	user number (high 8 bits)	user number (low 8 bits)	0	0	CHK	0xF5

Response data format:

1) Header:

byte	1	2	3	4	5	6	7	8
response	0xF5	0x31	Hi(Len)	Low(Len)	ACK_SUCCESS ACK_FAIL ACK_NOUSER	0	CHK	0xF5

2) Packet:

byte	1	2	3	4	5 --- Len+1	Len+2	Len+3
response	0xF5	user number (high 8 bits)	user number (low 8 bits)	user permission (1/2/3)	data characteristics	CHK	0xF5

Description: the length Len-3 of data characteristics equals to 193 bytes.

18. Loading characteristics and fingerprint by user number which are stored in DSP module database (>8 bytes command and 8 bytes response)

Command data format:

1) Header:

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

2) Packet:

byte	1	2	3	4	5 --- Len+1	Len+2	Len+3
command	0xF5	user number (Low 8 bits)	user number (Low 8 bits)	user permission (1/2/3)	data characteristics	CHK	0xF5

Description: the length Len-3 of data characteristics equals to 193 bytes

Response data format:

byte	1	2	3	4	5	6	7	8
response	0xF5	0x41	stored user number (high 8 bits)	stored user number (Low 8 bits)	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5

19. Getting all numbers and permissions of existed users (8 bytes command and >8 bytes response)

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) 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) Packet:

byte	1	2	3	4 --- Len+1	Len+2	Len+3
response	0xF5	user number (high 8 bits)	user number (low 8 bits)	user message data (user number and permission)	CHK	0xF5

Description: The length of “Len” of packet data equals to “3*user number+2”

User message data format:

byte	4	5	6	7	8	9	…
data	1st user number (high 8 bits)	1st user number (low 8 bits)	1st user permission (1/2/3)	2nd user number (high 8 bits)	2nd user number (low 8 bits)	2nd user permission (1/2/3)	…

20. Getting single record data (8 bytes command and > 8 bytes response)

Note: this module protocol is not provided temporary.

This protocol returns a record data from database of “recorded location)

Command data format:

byte	1	2	3	4	5	6	7	8
command	0xF5	0x38	Recorded location (high 8 bits)	Recorded location (low 8 bits)	0	0	CHK	0xF5

Response data format:

1) 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) Packet:

byte	1	2	3	4	5
response	0xF5	bits 7-1: year bit 0: month (3 bits)	bits 7-5: month (2-0 bits) 4-0 bits: day	bits 7-2: hour 1-0 bits: min (5-4 bits)	bits 7-4: min (3-0 bits) 3-0 bits: record number (21-18 bits)

byte	6	7	8	9	10	11
response	record number (10-17 bits)	record number (9-1 bits)	7-6 bits: record number (1-0 bits) 5-0 bits: user number (1308 bits)	user number (7-0 bits)	CHK	0xF5

Description: the length Len of data record equals to 8.

21. Getting new record data (8 bytes command and >8 bytes response)

Note: this module protocol is not provided temporary.

This protocol returns record from database which number equals and is greater " the smallest record number" of 50 recorded data.

byte	1	2	3	4	5	6	7	8
command	0xF5	0x39	bits 7-6: 0 bits 5-0: the smallest record number (21-16)	the smallest record number (15-8)	the smallest record number (7-0)	0	CHK	0xF5

Response data format:

1) 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) Packet:

byte	1	2 --- 9	10-17	…	Len+2	Len+3
response	0xF5	1st record	2nd record	…	CHK	0xF5

Description: in each packet record with a format as in **20.** the packet response data are in bytes 2-9.

The data length Len equals to "8*50 = 400 bytes".

22. Record data cleaning (8 bytes command/response)

Note: this module protocol is not provided temporary.

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

23. Module time setting (>8 bytes command and 8 bytes response)

Note: this module protocol is not provided temporary

Command data format:

1) Header:

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

2) Packet:

byte	1	2	3	4	5	6	7	8	9	10
command	0xF5	week	year	month	day	hour	min	sec	CHK	0xF5

Description: the length Len of time data equals to 7

Response data format:

byte	1	2	3	4	5	6	7	8
response	0xF5	0x48	0	0	ACK_SUCCESS ACK_FAIL	024.	CHK	0xF5

24. System time getting:

Note: this module protocol is not provided temporary.

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) 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) Packet:

byte	1	2	3	4	5	6	7	8	9	10
response	0xF5	week	year	month	day	hour	min	sec	CHK	0xF5

Description: The length Len of time data equals to 7.

25. Timeout period of fingerprint's obtaining setting/getting (8 bytes command/response)

byte	1	2	3	4	5	6	7	8
command	0xF5	0x2E	0	byte5=0: new timeout period byte5=1: 0	0: new timeout period setting 1: current timeout period getting	0	CHK	0xF5
response	0xF5	0x2E	0	current timeout period	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5

Description: fingerprint timeout period (tout) is in range. If it is 0 and the fingerprint isn't pressed, the process of scanning continues. If it is more than 0, and the fingerprint isn't pressed, the process finishes after timeout period "tout * T0".

Note: T0 is a scanning or image processing time and is in range 0.2-0.3 sec

USB SDK DEVELOPING PROTOCOL

```
#ifndef   _D5_SCANNER_H_
#define   _D5_SCANNER_H_
#define   _DEV_MAX_NUM   5
#define   LED_R     0x01
#define   LED_G     0x02
#define   LED_B     0x04
```

1 DEVICE ENUMERATION

unsigned short WINAPI D5EnumDevice (char pDeviceName[DEV_MAX_NUM][128]);

Description:

Enumeration of already connected devices (maximum 5 touches at each computer)

Arguments:

pDeviceName: title of already connected device.

Return:

0: there aren't any connected devices

otherwise: count of connected devices

2 DEVICE OPENING

long WINAPI D5OpenDevice (unsigned short uDeviceID);

Description:

Device opening (Note: if some is already opened, it should be closed firstly to open again)

Arguments:

uDeviceID: device number (0-4)

Return:

0: is opened successfully;

-1: operation is failed

3 DEVICE CLOSING

long WINAPI D5CloseDevice (unsigned short uDeviceID);

Description:

Device closing

Arguments:

uDeviceID: device number (0-4)

Return:

0: is closed successfully

-1: operation is failed

4 SETTING COMPANY MARK KEY

long WINAPI D5SetMark (unsigned short uDeviceID, unsigned char *pMark);

Description:

setting company mark.

Arguments:

uDeviceID: device number (0-4)

pMark: company mark (8 bytes)

Return:

0: is set successfully

-1: operation is failed

5 CHECKING KEY OF COMPANY MARK

long WINAPI D5CheckMark (unsigned short wDevID, unsigned char *pMark);

Description:

it checks logo if company

Arguments:

uDeviceID: device number (0-4)

pMark: company mark (8 bytes)

Return:

0: match

1: doesn't match

-1: operation is failed

6 BEEP SOUD REMINDER

long WINAPI D5Beep (unsigned short uDeviceID, unsigned short uMS)

Description:

Set beep sound

Arguments:

uDeviceID: device number (0-4);

uMS: beep sound duration (ms)

Return:

0: beep sound set successfully

-1: operation is failed

7 LED INDICATOR OPENING

long WINAPI D5OpenLED (unsigned short uDeviceID, unsigned short uLEDS);

Description:

LED opening

Arguments:

uDeviceID: device number (0-4);

uLEDS: LED bits combination (red-Bit0/Green-Bit1/blue-Bit2)

Return:

0: LED is opened successfully;

-1: operation is failed

8 LED INDICATOR CLOSING

long WINAPI D5CloseLED (unsigned short uDeviceID, unsigned short uLEDS);

Description:

LED closing.

Arguments:

uDeviceID: device number (0-4);

uLEDS: LED bits combination (red-Bit0/green-Bit1/blue-Bit2).

Return:

0: LED is closed successfully;

-1: operation is failed

9 LOADING IMAGE DATA FROM BITMAP FILE

long WINAPI D5LoadBMPFile (char *strFileName, unsigned char *pImage);

Description:

loading image data from the bitmap file

Arguments:

strFileName: bitmap file name

pImage: image data; (width 192 pixels, height 256 pixels. Data order: row by row. Each grayscale pixel takes 1 byte)

Return:

0: file is loaded successfully

-1: there are no opening permission or wrong file format

10 SAVING IMAGE DATA TO BITMAP FILE

long WINAPI D5SaveBMPFile (char *strFileName, unsigned char *pImage)

Description:

saving image data to the bitmap file

Arguments:

strFileName: bitmap file name

pImage: image data (width 192 pixels, height 256 pixels. Data order: row by row. Each grayscale pixel takes 1 byte)

Return:

0: file is saved successfully

-1: file is not saved

11 IMAGE READING FROM DEVICE

long WINAPI D5GetImage (unsigned short wDeviceID, unsigned char *pImage);

Description:

image reading

Arguments:

wDeviceID: device number (0-4)

pImage: image data (width 192 pixels, height 256 pixels. Data order: row by row. Each grayscale pixel takes 1 byte)

Return:

0: image is read successfully;

-1: operation is failed

12 FINGERPRINT IAMGE DETECTING

bool WINAPI D5CheckFP (unsigned char *pImage);

Description:

detects fingerprint image from image data

Arguments:

pImage: image data. (width 192 pixels, height 256 pixels. Data order: row by row. Each grayscale pixels 1 byte

Return:

1: fingerprint is detected

0: fingerprint isn't detected

13 EXTRACTING CHARACTERISTICS FROM FINGERPRINT

long WINAPI D5Process (unsigned char *pImage, unsigned char *pFeature);

Description:

extracting characteristics from fingerprint

Arguments:

pImage: fingerprint's image data; (width 192 pixels, height 256 pixels. Data order: row by row. Each grayscale pixels takes 1 byte)

pFeature: existed fingerprint characteristics (256-item non-symbol byte array)

Return:

0: match

-1: doesn't match

-2: system error

14 1:1 FINGERPRINT CHARACTERISTICS COMPARISON

long WINAPI D5Match (unsigned char *pFeature1, unsinged char *pFeatuer2, unsigned short uRotate=60, unsigned short uLevel=5);

Description:

comparison of two fingerprint characteristics

Arguments:

pFeature1: 1st fingerprint characteristics

pFeature2: 2nd fingerprint characteristics

uRotate: rotation angle (1-180)

uLevel: matching level (0-9)

Return:

0: match

-1: doesn't match

-2: system error

15 1:N FINGERPRINT CHARACTERISTICS COMPARISON

long WINAPI D5MatchN (unsigned char *pFeatureIn, unsigned char pFeatureLib[][256], unsigned long lFingernum, unsigned short uRotate = 60, unsigned short uLevel = 5);

Description:

Fingerprint characteristics comparison with fingerprint stored in database of fingerprints.

Arguments:

pFeatureIn: compared characteristics:

pFeatureLib: fingerprint characteristics database

lFingernum: fingerprint number in database, the number isn't limited

uRotate: rotation angle (1-180)

uLevel: matching level (0-9)

Return:

-1: matching fail;

-2: system error

otherwise: scanned fingerprint and fingerprint from database are matched successfully.