Multi\_Fingerprint Manual

Description：

USB communication uses the "FingerPrint\_usb.dll" dynamic library.

Steps for usage:

1. Initialize the dynamic library: int16\_t WINAPI USB\_lib\_int(void);
2. Open the device: int16\_t WINAPI USB\_open(uint32\_t vid,uint32\_t pid)；
3. Call the instruction function to implement specific functions.
4. End of the program, shut down the device: int16\_t WINAPI USB\_close(void)；

Note:

The fingerprint image size is 160\*160, 8-bit grayscale image;The length of the image data read is 160\*80,Image data is collected from left to right, top to bottom;The upper 4 bits of 2 pixels are combined into one byte. The upper 4 bits of each byte are the previous pixel data, and the lower 4 bits are the next pixel data, so the picture data needs to be converted before the picture is displayed.

For example, the read data is 0x4d, and is converted to 0x40 and 0xd0.

# Introduction to the use of interface functions

# View user information in the module

## 1.1To see how many users are entered in the module, call the function:

int16\_t WINAPI USB\_PS\_ValidTempleteNum(uint8\_t \* readbuff);

Total number of users=readbuff[0]\*256+readbuff[1];

## 1.2 View the use situation of the user number in the module, call the function:

int16\_t WINAPI USB\_PS\_ReadIndexTable(uint8\_t page\_number,uint8\_t \* readbuff);

Incoming parameters: page\_numberpage\_number=0；

Read data: readbuff[0]\*256+readbuff[1]=32,Indicates that there are 32 bytes of data behind;readbuff[2-33] A total of 32 bytes, each 1 bit represents an ID number, that is, 32 \* 8 = 256, indicating the use of 0--255 ID number, = 1 indicates that it has been used, and = 0 indicates that it is not used.

## 1.3 Get the first unused user number in the module, call the function:

int16\_t WINAPI USB\_PS\_GetDummyTempleteNo(uint8\_t \* readbuff);

First unused user number=readbuff[0]\*256+readbuff[1];

# Delete users

## 2.1 Delete the specified user, call the function:

int16\_t WINAPI USB\_PS\_DeletChar(uint16\_t pageID,uint16\_t num);

Incoming parameters: pageID-- Start user numberID(0--99)；num--The number of consecutive deletions(0<= pageID+num<=99).

## 2.2 Delete all users, call the function:

int16\_t WINAPI USB\_PS\_Empty(void);

# Add user

## 3.1Automatically add users, call the function:

int16\_t WINAPI USB\_PS\_AutoEnroll(uint16\_t ID)；

Incoming parameters: ID--The added user ID number(0--99).

## 3.2 Add users step by step, the process:

First, get the fingerprint image , call the function:

int16\_t WINAPI USB\_PS\_GetImage(void);

Second, upload the fingerprint image for the upper computer to display the fingerprint image. If the upper computer does not display the fingerprint image, ignore this step. When the fingerprint image is displayed, call the function:

int16\_t WINAPI USB\_PS\_UpImage(uint8\_t \* readbuff);

Read data: Number of data of the image=readbuff[0]\*256+readbuff[1]=160\*80，Image data is :readbuff[2-n]。

Third, generate a fingerprint feature value and call the function:

int16\_t WINAPI USB\_PS\_GetChar(uint8\_t BufferID);

Incoming parameters: BufferID=1。

Fourth, repeat the previous operation 3 times, but the BufferID of the third step is equal to 2, 3, 4 in turn.

Fifth, merge the feature values, call the function:

int16\_t WINAPI USB\_PS\_RegMB(void);

Sixth, store the user, call the function:

int16\_t WINAPI USB\_PS\_StorMB(uint8\_t BufferID,uint16\_t PageID);

Incoming parameters: BufferID=1。 PageID is the stored user number(0--99).

# verify fingerprint

## 4.1 Automatically verify the fingerprint, call the function:

int16\_t WINAPI USB\_PS\_Autoldentify(uint16\_t ID,uint8\_t \* readbuff)；

Incoming parameters:If ID=0xffff, a 1:n comparison is performed, that is, the entire fingerprint library is compared. If 0<=ID<=99, a 1:1 comparison is performed, that is, the fingerprint matching with the ID number specified in the fingerprint library.

Read data:Compare the same fingerprint ID =readbuff[0]\*256+readbuff[1]，Similarity score =readbuff[2]\*256+readbuff[3]。

## 4.2 Step-by-step verification of fingerprints, process:

First, get the fingerprint image and call the function:

int16\_t WINAPI USB\_PS\_GetImage(void);

Second, upload the fingerprint image for the upper computer to display the fingerprint image. If the upper computer does not display the fingerprint image, ignore this step. When the fingerprint image is displayed, call the function:

int16\_t WINAPI USB\_PS\_UpImage(uint8\_t \* readbuff);

Read data: Number of data of the image=readbuff[0]\*256+readbuff[1]=160\*80，Image data is :readbuff[2-n]。

Third, generate a fingerprint feature value and call the function:

int16\_t WINAPI USB\_PS\_GetChar(uint8\_t BufferID);

Incoming parameters: BufferID=1。

Fourth,1：n comparison，call the function:

int16\_t WINAPI USB\_PS\_SearchMB(uint8\_t BufferID,uint16\_t StartPage,uint16\_t PageNum, uint8\_t \* readbuff);

Incoming parameters: BufferID=1，StartPage=0，PageNum=99。

Read data:Compare the same fingerprint ID =readbuff[0]\*256+readbuff[1]，Similarity score =readbuff[2]\*256+readbuff[3]。

# USB library function interface

# General function

## 1.1 USB\_lib\_int

|  |  |
| --- | --- |
| Function name | USB\_lib\_int |
| Function prototype | int16\_t WINAPI USB\_lib\_int(void); |
| Functional description | Dynamic library initialization |
| Incoming parameters | no |
| Output parameters | no |
| return value | = 0 operation successfully；  < 0 operation failed； |
| prerequisites | no |
| Called function | no |

## 1.2 USB\_open

|  |  |
| --- | --- |
| Function name | USB\_open |
| Function prototype | int16\_t WINAPI USB\_open(uint32\_t vid,uint32\_t pid)； |
| Functional description | Open HID device |
| Incoming parameters | vid：VID number  pid：PID number |
| Output parameters | no |
| return value | = 0 operation successfully；  < 0 operation failed； |
| prerequisites | Dynamic library initialization succeeded（USB\_lib\_int） |
| Called function | no |

## 1.3 USB\_close

|  |  |
| --- | --- |
| Function name | USB\_close |
| Function prototype | int16\_t WINAPI USB\_close(void)； |
| Functional description | Turn off the HID device |
| Incoming parameters | no |
| Output parameters | no |
| return value | = 0 operation successfully；  < 0 operation failed； |
| prerequisites | Dynamic library initialization succeeded（USB\_lib\_int） |
| Called function | no |

# the fingerprint module instruction

## 2.1 USB\_PS\_AutoEnroll

|  |  |
| --- | --- |
| Function name | USB\_PS\_AutoEnroll |
| Function prototype | int16\_t WINAPI USB\_PS\_AutoEnroll(uint16\_t ID)； |
| Functional description | Automatically add fingerprints |
| Incoming parameters | ID：ID number（0--99）； |
| Output parameters | no |
| return value | = 0 operation successfully；  < 0 operation failed； |
| prerequisites | USB device is successfully opened（USB\_open） |
| Called function | no |

## 2.2 USB\_PS\_Autoldentify

|  |  |
| --- | --- |
| Function name | USB\_PS\_Autoldentify |
| Function prototype | int16\_t WINAPI USB\_PS\_Autoldentify(uint16\_t ID,uint8\_t \* readbuff)； |
| Functional description | Automatic verification of fingerprints |
| Incoming parameters | ID：ID number（0--99 Make a 1:1 comparison；0xffff --Make a 1:n comparison）； |
| Output parameters | Readbuff：Read data returned（readbuff[0-1]--ID number（=readbuff[0]\*256+readbuff[1]）；readbuff[2-3]--Score（=readbuff[2]\*256+readbuff[3]）） |
| return value | = 0 operation successfully；  < 0 operation failed； |
| prerequisites | USB device is successfully opened（USB\_open） |
| Called function | no |

## 2.3 USB\_PS\_DeletChar

|  |  |
| --- | --- |
| Function name | USB\_PS\_DeletChar |
| Function prototype | int16\_t WINAPI USB\_PS\_DeletChar(uint16\_t pageID,uint16\_t num); |
| Functional description | Delete template |
| Incoming parameters | pageID：page number（0--99）. num：Delete number(0<= pageID+num<=99) |
| Output parameters | no |
| return value | = 0 operation successfully；  < 0 operation failed； |
| prerequisites | USB device is successfully opened（USB\_open） |
| Called function | no |

## 2.4 USB\_PS\_Empty

|  |  |
| --- | --- |
| Function name | USB\_PS\_Empty |
| Function prototype | int16\_t WINAPI USB\_PS\_Empty(void); |
| Functional description | Clear fingerprint library |
| Incoming parameters | no |
| Output parameters | no |
| return value | = 0 operation successfully；  < 0 operation failed； |
| prerequisites | USB device is successfully opened（USB\_open） |
| Called function | no |

## 2.5 USB\_PS\_Cancel

|  |  |
| --- | --- |
| Function name | USB\_PS\_Cancel |
| Function prototype | int16\_t WINAPI USB\_PS\_Cancel(void); |
| Functional description | Cancel instruction |
| Incoming parameters | no |
| Output parameters | no |
| return value | = 0 operation successfully；  < 0 operation failed； |
| prerequisites | USB device is successfully opened（USB\_open） |
| Called function | no |

## 2.6 USB\_PS\_Sleep Not supported yet

|  |  |
| --- | --- |
| Function name | USB\_PS\_Sleep |
| Function prototype | int16\_t WINAPI USB\_PS\_Sleep(void); |
| Functional description | Enter sleep |
| Incoming parameters | no |
| Output parameters | no |
| return value | = 0 operation successfully；  < 0 operation failed； |
| prerequisites | USB device is successfully opened（USB\_open） |
| Called function | no |

## 2.7 USB\_PS\_ValidTempleteNum

|  |  |
| --- | --- |
| Function name | USB\_PS\_ValidTempleteNum |
| Function prototype | int16\_t WINAPI USB\_PS\_ValidTempleteNum(uint8\_t \* readbuff); |
| Functional description | Read the number of valid templates (ie the total number of users) |
| Incoming parameters | no |
| Output parameters | Readbuff：Read data returned（readbuff[0-1]--Number of valid templates（=readbuff[0]\*256+readbuff[1]）） |
| return value | = 0 operation successfully；  < 0 operation failed； |
| prerequisites | USB device is successfully opened（USB\_open） |
| Called function | no |

## 2.8 USB\_PS\_ReadIndexTable

|  |  |
| --- | --- |
| Function name | USB\_PS\_ReadIndexTable |
| Function prototype | int16\_t WINAPI USB\_PS\_ReadIndexTable(uint8\_t page\_number,uint8\_t \* readbuff); |
| Functional description | Read index table |
| Incoming parameters | page\_number：page number（0--Corresponding template index from 0-255） |
| Output parameters | Readbuff：Read data returned（readbuff[0--1]--Index information data length（=readbuff[0]\*256+readbuff[1])  ；readbuff[2--n]--Index information (each 1 bit represents a template, =1 represents the mode corresponding to the storage area Version has been entered, =0 means not entered）） |
| return value | = 0 operation successfully；  < 0 operation failed； |
| prerequisites | USB device is successfully opened（USB\_open） |
| Called function | no |

## 2.9 USB\_PS\_SetPwd

|  |  |
| --- | --- |
| Function name | USB\_PS\_SetPwd |
| Function prototype | int16\_t WINAPI USB\_PS\_SetPwd(uint8\_t \* PassWord); |
| Functional description | Set password |
| Incoming parameters | PassWord：password（PassWord[0-3]--Password 4 bytes） |
| Output parameters | no |
| return value | = 0 operation successfully；  < 0 operation failed； |
| prerequisites | USB device is successfully opened（USB\_open） |
| Called function | no |

## 2.10 USB\_PS\_VfyPwd

|  |  |
| --- | --- |
| Function name | USB\_PS\_VfyPwd |
| Function prototype | int16\_t WINAPI USB\_PS\_VfyPwd(uint8\_t \* PassWord); |
| Functional description | Verify password |
| Incoming parameters | PassWord：password（PassWord[0-3]--Password 4 bytes） |
| Output parameters | no |
| return value | = 0 operation successfully；  < 0 operation failed； |
| prerequisites | USB device is successfully opened（USB\_open） |
| Called function | no |

## 2.11 USB\_PS\_GetImage

|  |  |
| --- | --- |
| Function name | USB\_PS\_GetImage |
| Function prototype | int16\_t WINAPI USB\_PS\_GetImage(void); |
| Functional description | Get image |
| Incoming parameters | no |
| Output parameters | no |
| return value | = 0 operation successfully；  < 0 operation failed； |
| prerequisites | USB device is successfully opened（USB\_open） |
| Called function | no |

## 2.12 USB\_PS\_GetRnrollImage

|  |  |
| --- | --- |
| Function name | USB\_PS\_GetRnrollImage |
| Function prototype | int16\_t WINAPI USB\_PS\_GetRnrollImage(void); |
| Functional description | Registration to get images |
| Incoming parameters | no |
| Output parameters | no |
| return value | = 0 operation successfully；  < 0 operation failed； |
| prerequisites | USB device is successfully opened（USB\_open） |
| Called function | no |

## 2.13 USB\_PS\_GetChar

|  |  |
| --- | --- |
| Function name | USB\_PS\_GetChar |
| Function prototype | int16\_t WINAPI USB\_PS\_GetChar(uint8\_t BufferID); |
| Functional description | Generating features |
| Incoming parameters | BufferID：Buffer number (1--4 during the registration process, BufferID indicates the first few fingers). |
| Output parameters | no |
| return value | = 0 operation successfully；  < 0 operation failed； |
| prerequisites | USB device is successfully opened（USB\_open） |
| Called function | no |

## 2.14 USB\_PS\_RegMB

|  |  |
| --- | --- |
| Function name | USB\_PS\_RegMB |
| Function prototype | int16\_t WINAPI USB\_PS\_RegMB(void); |
| Functional description | Merge template |
| Incoming parameters | no |
| Output parameters | no |
| return value | = 0 operation successfully；  < 0 operation failed； |
| prerequisites | USB device is successfully opened（USB\_open） |
| Called function | no |

## 2.15 USB\_PS\_StorMB

|  |  |
| --- | --- |
| Function name | USB\_PS\_StorMB |
| Function prototype | int16\_t WINAPI USB\_PS\_StorMB(uint8\_t BufferID,uint16\_t PageID); |
| Functional description | Storage template |
| Incoming parameters | BufferID：Buffer number（1--4，The default is 1）；  PageID：User number（0--99） |
| Output parameters | no |
| return value | = 0 operation successfully；  < 0 operation failed； |
| prerequisites | USB device is successfully opened（USB\_open） |
| Called function | no |

## 2.16 USB\_PS\_SearchMB

|  |  |
| --- | --- |
| Function name | USB\_PS\_SearchMB |
| Function prototype | int16\_t WINAPI USB\_PS\_SearchMB(uint8\_t BufferID,uint16\_t StartPage,uint16\_t PageNum, uint8\_t \* readbuff); |
| Functional description | Search template(1:n Comparison) |
| Incoming parameters | BufferID：Buffer number (1--4, default is 1);  StartPage：Start page（default is 0）；  PageNum：Number of pages（default is 99）。 |
| Output parameters | Readbuff：Read data returned（  readbuff[0--1]--page number（=readbuff[0]\*256+readbuff[1]）； readbuff[2--3]--Score（=readbuff[2]\*256+readbuff[3]）） |
| return value | = 0 operation successfully；  < 0 operation failed； |
| prerequisites | USB device is successfully opened（USB\_open） |
| Called function | no |

## 2.17 USB\_PS\_GetChipUID Not supported yet

|  |  |
| --- | --- |
| Function name | USB\_PS\_GetChipUID |
| Function prototype | int16\_t WINAPI USB\_PS\_GetChipUID(uint8\_t \* readbuff); |
| Functional description | Get the chip serial number |
| Incoming parameters | no |
| Output parameters | Readbuff：Read data returned（readbuff[0--1]--UID Information data length（=readbuff[0]\*256+readbuff[1]）；readbuff[2--n]--UID Information data ） |
| return value | = 0 operation successfully；  < 0 operation failed； |
| prerequisites | USB device is successfully opened（USB\_open） |
| Called function | no |

## 2.18 USB\_PS\_GetChipEcho Not supported yet

|  |  |
| --- | --- |
| Function name | USB\_PS\_GetChipEcho |
| Function prototype | int16\_t WINAPI USB\_PS\_GetChipEcho(void) |
| Functional description | Handshake command |
| Incoming parameters | no |
| Output parameters | no |
| return value | = 0 operation successfully；  < 0 operation failed； |
| prerequisites | USB device is successfully opened（USB\_open） |
| Called function | no |

## 2.19 USB\_PS\_AutoCaiSensor Not supported yet

|  |  |
| --- | --- |
| Function name | USB\_PS\_AutoCaiSensor |
| Function prototype | int16\_t WINAPI USB\_PS\_AutoCaiSensor(void); |
| Functional description | Calibration sensor |
| Incoming parameters | no |
| Output parameters | no |
| return value | = 0 operation successfully；  < 0 operation failed； |
| prerequisites | USB device is successfully opened（USB\_open） |
| Called function | no |

## 2.20 USB\_PS\_UpImage

|  |  |
| --- | --- |
| Function name | USB\_PS\_UpImage |
| Function prototype | int16\_t WINAPI USB\_PS\_UpImage(uint8\_t \* readbuff); |
| Functional description | Upload image |
| Incoming parameters | no |
| Output parameters | Readbuff：Read data returned（readbuff[0--1]--Image data length（=readbuff[0]\*256+readbuff[1]）； readbuff[2--n]--Image data） |
| return value | = 0 operation successfully；  < 0 operation failed； |
| prerequisites | USB device is successfully opened（USB\_open）  Get image successfully（USB\_PS\_GetChar） |
| Called function | no |

Image data description:

The image size is 160\*160; the size of the read image data is 160\*160/2 bytes; one byte contains two pixels, the previous pixel occupies high 4 bits, and the latter pixel occupies low 4 bits.

Image data conversion reference method: Convert one byte image data into two-byte image data, such as 0x6d becomes 0x60 and 0xd0. This becomes image data of 160\*160 size. This processing method is for reference only.

## 2.21 USB\_PS\_GetDummyTempleteNo

|  |  |
| --- | --- |
| Function name | USB\_PS\_GetDummyTempleteNo |
| Function prototype | int16\_t WINAPI USB\_PS\_GetDummyTempleteNo(uint8\_t \* readbuff); |
| Functional description | Get blank storage |
| Incoming parameters | no |
| Output parameters | Readbuff：Read data returned（readbuff[0--1]--Storage number（=readbuff[0]\*256+readbuff[1]）） |
| return value | = 0 operation successfully；  < 0 operation failed； |
| prerequisites | USB device is successfully opened（USB\_open） |
| Called function | no |

# Error code

|  |  |
| --- | --- |
| Parameter | Description |
| 0 | Operation Successful |
| -994 | USB library load failed |
| -995 | Open the USB device failed |
| -996 | Close the USB device failed |
| -1000 | Time out |
| -1001 | Send data failed |
| -1002 | Read data failed |
| -1003 | The specified ID number is invalid |
| -1004 | Incorrect number of entries |
| -1005 | The fingerprint library is full |
| -1006 | The specified user ID already exists |
| -1007 | Capture images failed |
| -1008 | Generate feature values failed |
| -1009 | Composite template failed |
| -1010 | Fingerprint template already exists |
| -1011 | Store template failed |
| -1012 | Automatic registration of fingerprint failed |
| -1013 | Did not find the fingerprint |
| -1014 | ID number is out of range |
| -1015 | Residual fingerprint |
| -1016 | Fingerprint template is empty |
| -1017 | The fingerprint library is empty |
| -1018 | Automatic verification of fingerprint failed |
| -1019 | Delete template failed |
| -1020 | Fingerprint device is incorrectly received |
| -1021 | Clear fingerprint library failed |
| -1022 | Cancel instruction failed |
| -1023 | The fingerprint is too messy to generate eigenvalues |
| -1024 | The fingerprint image is normal, but the feature points are too few to be born. |
| -1025 | There is no valid original image in the image buffer and no image is generated. |
| -1026 | Unable to upload image data |
| -1028 | Incorrect password |