



Capacitive Fingerprint Reader

用户手册

产品概述

- Capacitive Fingerprint Reader 是一款专用于二次开发集成应用的电容式指纹开发模块，高速度、快识别、高稳定性。
- Capacitive Fingerprint Reader 电容式指纹模块是以 STM32F105R8 高速数字处理器为核心，结合高安全性商用指纹算法，配高级半导体指纹传感器，并具有指纹录入、图像处理、特征值提取、模板生成、模板储存、指纹比对和搜索等功能的智能型集成模块；专业为科研单位，指纹产品生产企业，应用集成厂商提供标准二次开发指纹组件，快速、方便集成应用。

产品特性

- 使用方便，不需要了解指纹技术，模块内部结构和运算，只需简单的指令就能完成相应操作
- 采用商用算法，性能稳定，识别速度快
- 手指感应灵敏，手指只要轻轻地触碰采集窗就能快速识别，不需要用力按压
- 可以自由输入、输出指纹图片、指纹特征值文件及各种指纹操作
- 提供 UART 串口和 USB 双通讯方式
- 具有处理器掉电睡眠，手指感应唤醒功能，低功耗
- 提供完善的配套资料手册(配有 PC 机软件、SDK 开发板包、相关的工具和 Raspberry/Arduino/STM32 等示例程序)

应用案例

- 开发指纹锁，指纹保险箱，指纹门禁，指纹考勤

目 录

产品概述	1
产品特性	1
应用案例	1
使用说明	4
规格参数	4
接口说明	5
硬件连接	5
操作说明	6
正确按指纹	6
错误按指纹	6
传感器尺寸图	6
串口控制说明	7
实际操作	9
附加操作	11
USB 控制说明	14
示例程序	18
XNUCLEO-F103RB	19
硬件连接	19
下载程序	19
操作指令说明	20
UNO PLUS	20
硬件连接	20
下载程序	21
Raspberry Pi	21
硬件连接	21
树莓派串口配置	21
Python	22
wiringPi	22
程序运行结果	22

串口控制开发协议	23
USB SDK 开发包协议	40
1 枚举设备	40
2 打开设备	40
3 关闭设备	40
4 设置公司标记 KEY	41
5 检验公司标记 key	41
6 设备蜂鸣提示	42
7 打开 LED 灯提示	42
8 关闭 LED 灯	42
9 从位图中载入图像数据	43
10 将图像数据存储为位图	43
11 从设备中读取图像	44
12 检测某图像是否有指纹	44
13 指纹转成特征值	44
14 两个指纹特征值进行 1:1 比对	45
15 对输入的指纹特征值与指纹库内特征值进行快速 1 : N 比对	45
附录：应用方案	47

使用说明

规格参数

类别	参数
处理器 (CPU)	STM32F105
传感器	半导体传感器(电容式)
模块尺寸(mm)	45 x 30
传感器尺寸(mm)	33.4 x 20.4
图像	DPI 508
图像像素阵列	192 x 256
传感器感应面积(mm)	9.6 x 12.8
指纹容量	1000 枚 (可增容)
比对等级	默认 5, 可设置为 0-9, 越大越严格
比对时间	1 秒
认假率	0.00%
拒真率	0.01%

动态电流	<40mA
工作电压	3.3-5V
通讯接口	UART 或 USB
抗静电能力	空气放电 达到 $\pm 15KV$
	接触放电 达到 $\pm 8KV$

注：如果将存储芯片改为 W25Q32,模块将可扩展存储 4000 枚指纹。

接口说明

VCC ----- 3.3V 或 5V

GND ----- 接地

TXD(指纹模块串口发送) ----- RXD (PC 或单片机串口接收)

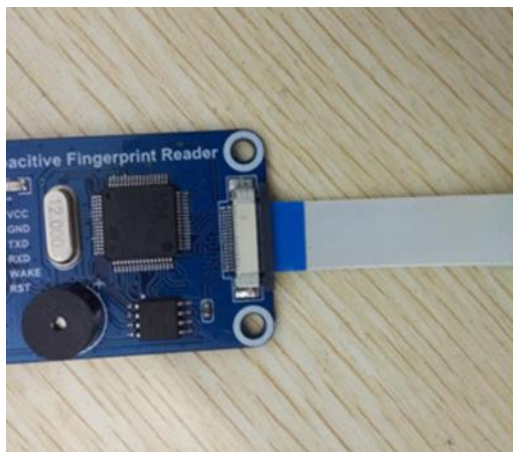
RXD(指纹模块串口接收) ----- TXD (PC 或单片机串口发送)

WAKE (指纹头唤醒, 可不接) ----- IO 口

RST(指纹模块复位, 可不接) ----- IO 口

硬件连接

传感器和主控板之间通过异向排线连接, 接线如下：



用户拿到模块后，可先使用测试软件对模块进行测试。在对模块有了一定的了解后，可使用下面的开发协议进行二次开发。

模块可以通过 UART 串口控制或者通过主控板上的 Micro USB 接口控制。

操作说明

本模块采用高精密元件，在采集指纹时：

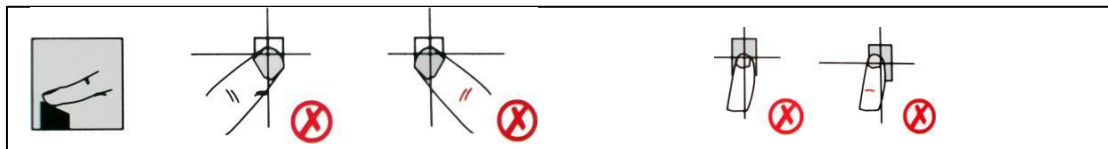
手指轻轻地触碰到采集窗口就能识别，不需要用力按压指纹采集窗

（感应灵敏、速度快、是本模块最大的特点）

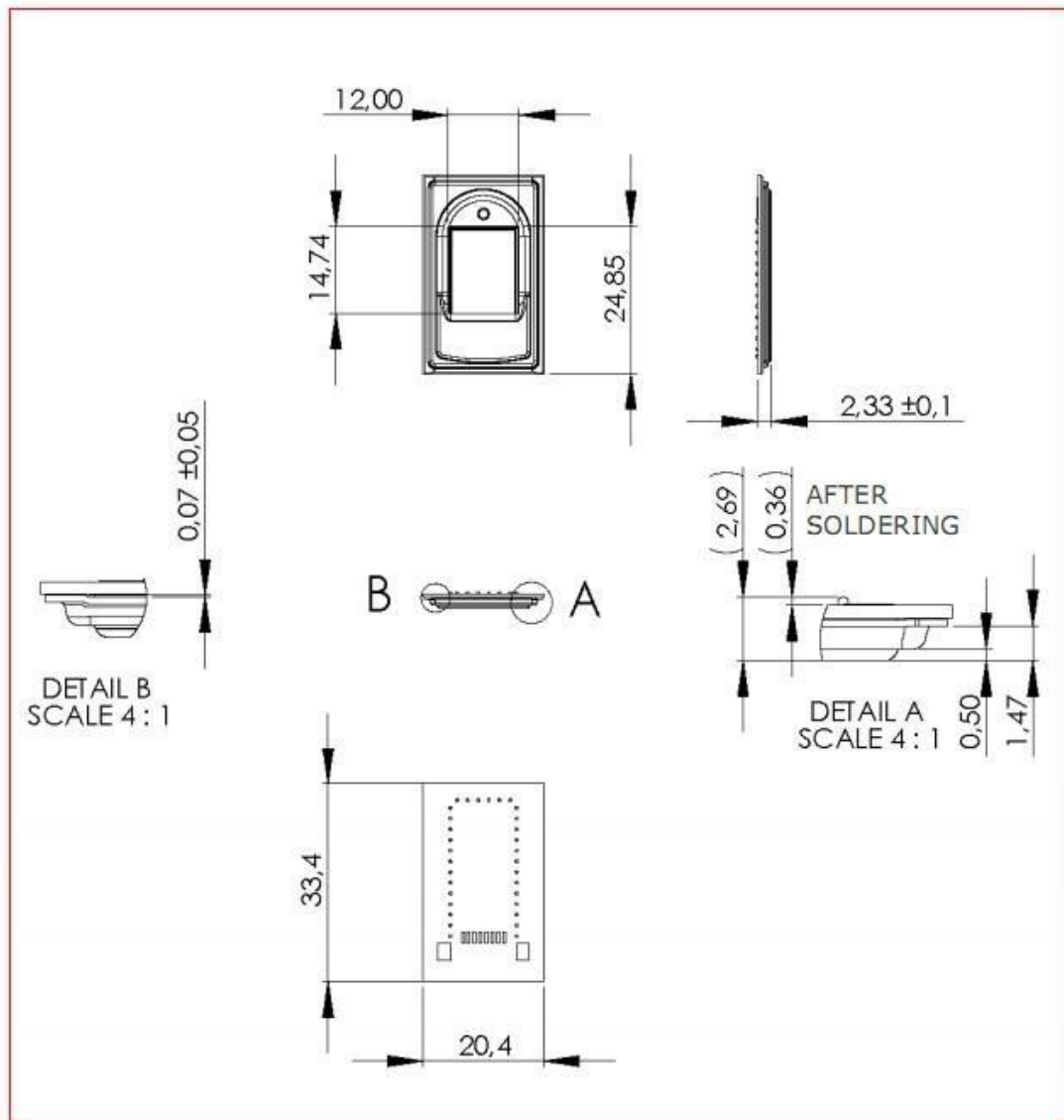
正确按指纹



错误按指纹

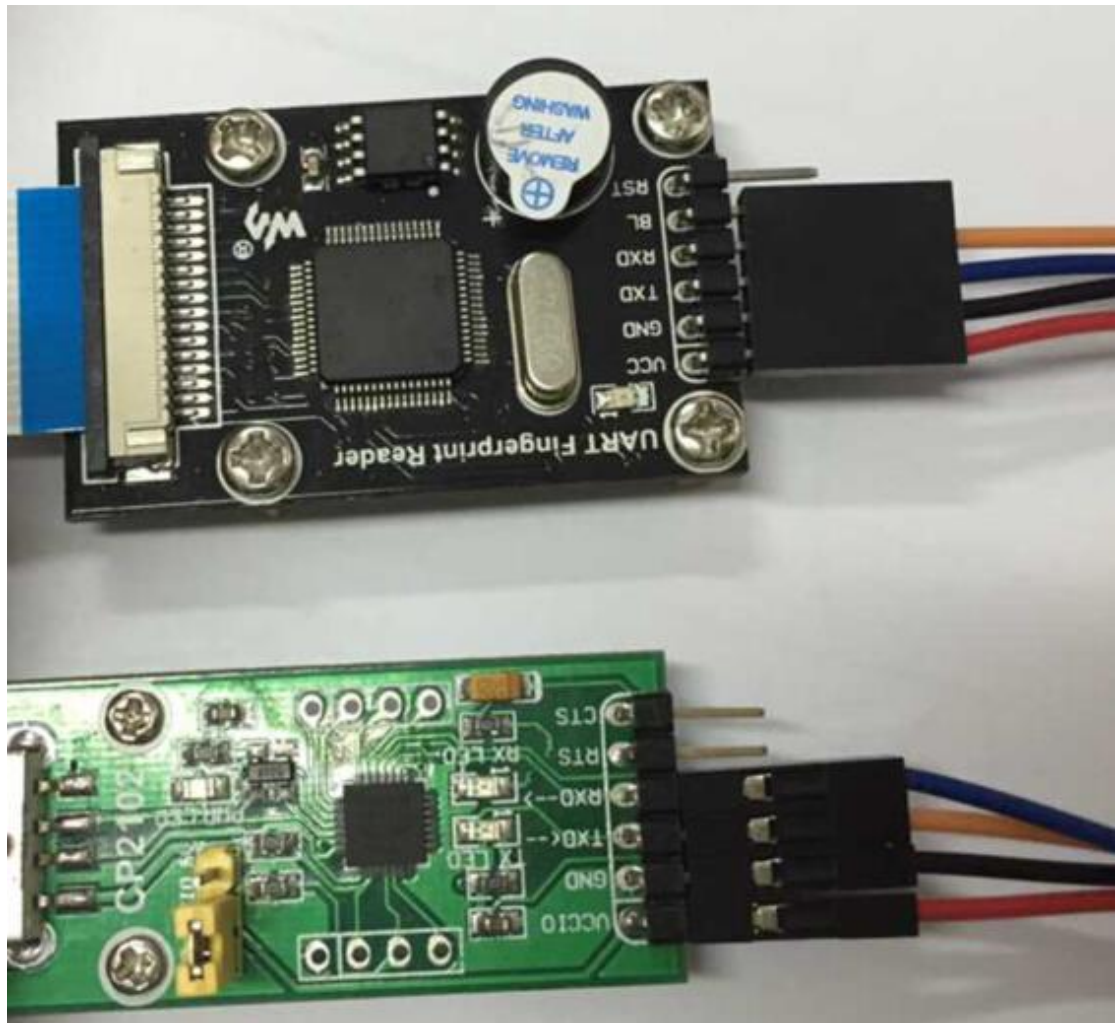


传感器尺寸图

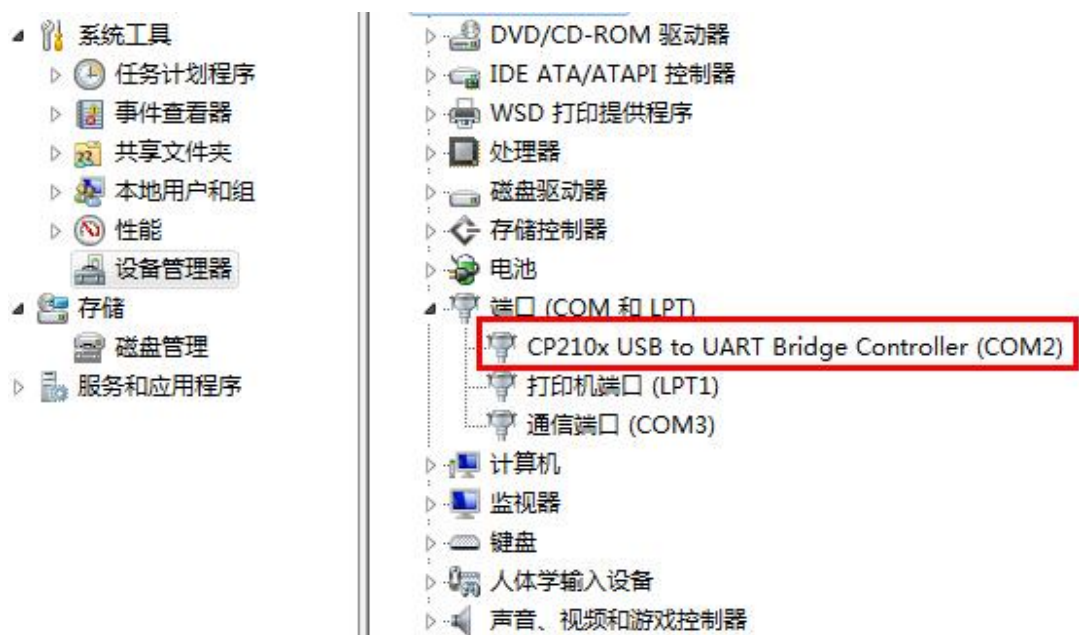


串口控制说明

1. 使用 CP2102 USB 转串口模块进行测试 (如用户使用其它 USB 转串口模块, 操作也类似, **模块需要另外购买**), 安装 [CP2102 驱动](#)
2. 按下图连接模块 (注意 RXD TXD 是交叉相连的) :



3. 将 CP2102 连接到 PC 机的 USB 接口，打开设备管理器，查看相应的 COM 口：



4. 下载测试软件：Capacitive-Fingerprint-Reader_(UART)，并注册控件：

注册 MSCOMM32.OCX，如提示：模块'mscomm32.ocx'已加载，但对 DllRegisterServer 的调用失败，错误代码为 0x8002801c

则按下面方法处理：

在 64 位系统中，需要把 MSCOMM32.OCX 复制到 C:\Windows\SysWOW64，再运行注册脚本。如果还是提示同样的错误，则需要将 MSCOMM32.OCX 复制到 C:\Windows\System32，再运行注册脚本（有时可能还要以管理员身份运行才可以注册成功！）。

打开 UART Fingerprint Reader,正常运行入下图所示（如果 USB 转串口默认不是 COM1，还会提示“Comm port error !”，选择确定即可）：

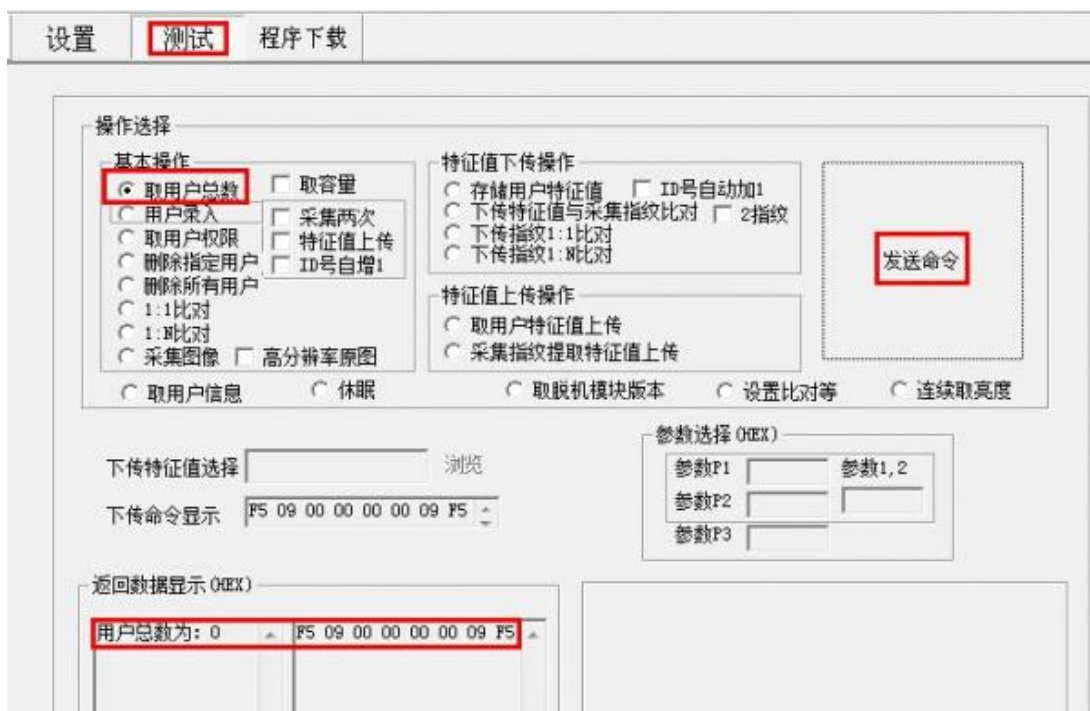


实际操作

1. 配置 COM 口：选择设置，在通信串口选择相应的串口号（默认波特率为 19200）



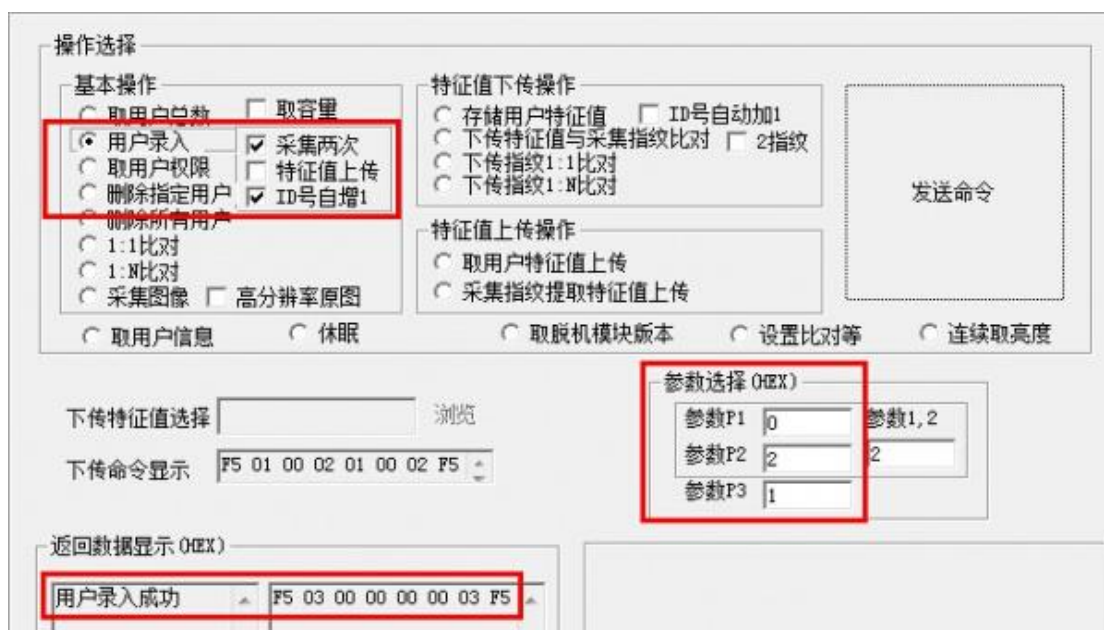
2. 检查通信：选择测试，在基本操作中选择“取用户总数”，点击发送命令，如在返回数据窗口中有相关内容显示，则说明模块与上位机通信成功。如无显示，请咨询查看前面几个步骤是否有误操作



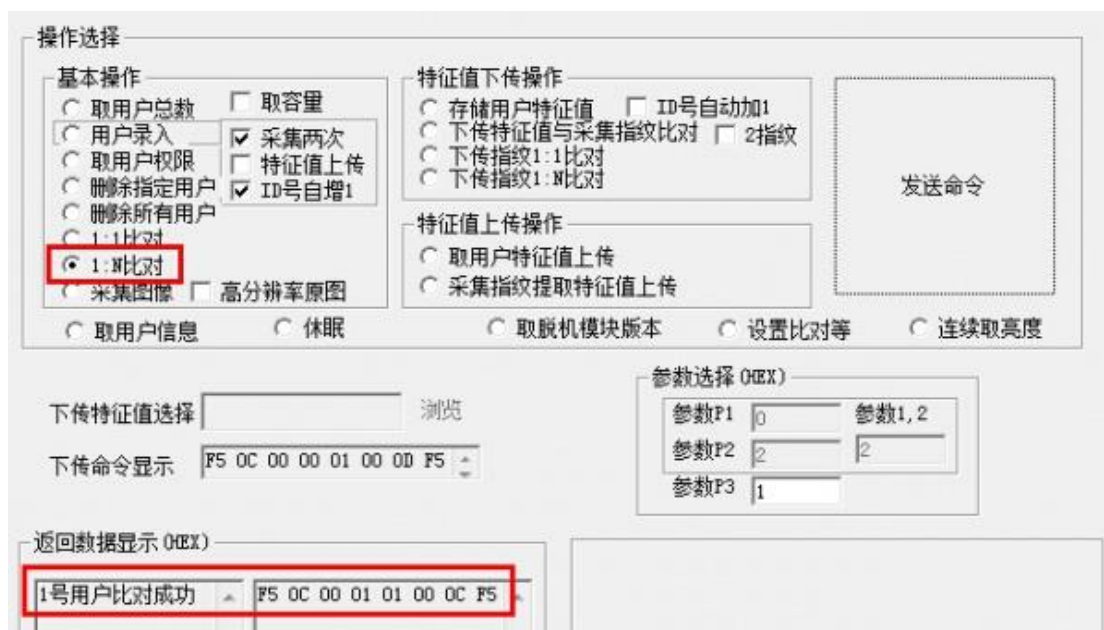
3. 用户录入：在基本操作中选择“用户录入”，勾选“采集两次”和“ID 号自增 1”，参数分辨设置为“011”（也可以设置其它，只要 ID 好不重复即可），点击发送，此时可以听到模块“嘀”一声，并且指纹头等亮起，手指触摸指纹头，录入两次后，正常情况下会提示“用户录入成功”

注意不可勾选“特征值上传”。如果选中该选项，特征值将上传到 PC 上而不写入指纹模块，会导致虽然显示添加指纹成功，但进行指纹匹配是无法匹配成功，且用户总数也

不会增加



4. 指纹对比：在基本操作中选择“1：N 比对”，点击发送，此时可以听到模块“滴”一声，并且指纹头灯亮起，手指触摸指纹头，如果该指纹已录入模块，则会提示比对成功，否则提示“无匹配指纹”



附加操作

注：以下涉及到修改波特率，如果你有改过波特率，导致下次使用时无法连接的情况，请先检查波特率是否正常！

1. 更改波特率：以 19200 比特率改为 115200 波特率为例：

选择设置，在通信串口选择相应的串口号，在更改模块波特率设置中选择“115200”，点

击发送命令即可



注：此时波特率已改为 115200，下次使用注意要用 115200 的波特率来配置上位机，否则将导致通信失败



如果忘记了模块的波特率，可以用以下方法测试出正确的波特率：

打开[串口调试助手](#)，选择相应的串口号，打开串口，勾选“HEX 显示”和“HEX 发送”，在发送窗口中输入“F5 09 00 00 00 00 09 F5”，点击发送，如果是正确的波特率，则在接收窗口转给你会显示相应的数据，类似：“F5 09 00 01 00 00 08 F5”，如果波特率不正确，则无显示，通过修改不同的波特率，重复上面的步骤，就可以得到正确的波特

率。



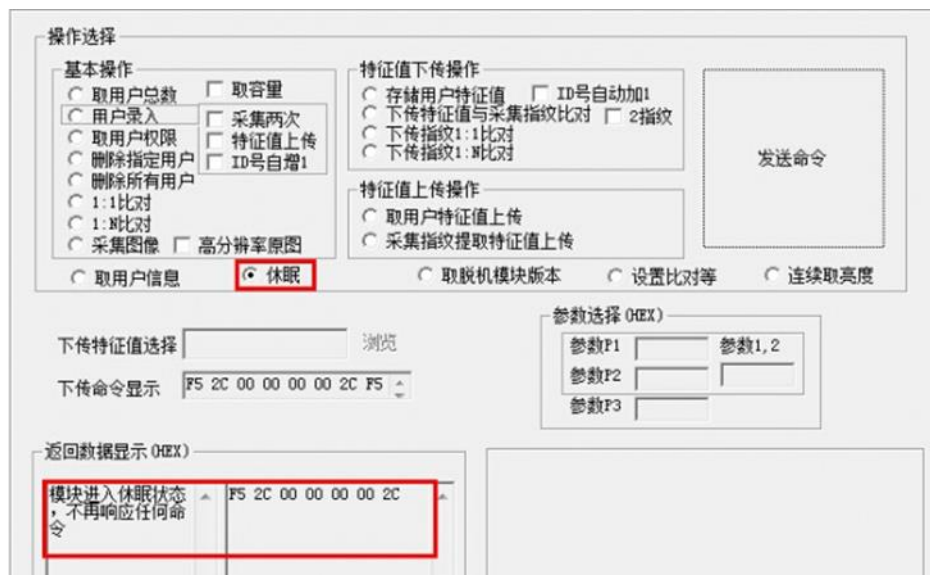
2. 采集图像：在基本操作中选择“采集图像”，点击发送，此时可以听到模块“滴”一声，并且指纹头灯亮起，手指触摸指纹头，大约 6 秒后上位机会显示采集到的图像

注：如果采集不到图像，或者采集到的图像不完整，则需要降低波特率，一般推荐在

9600 波特率下进行图像采集。改命令为测试命令，一般不推荐用于实际产品。



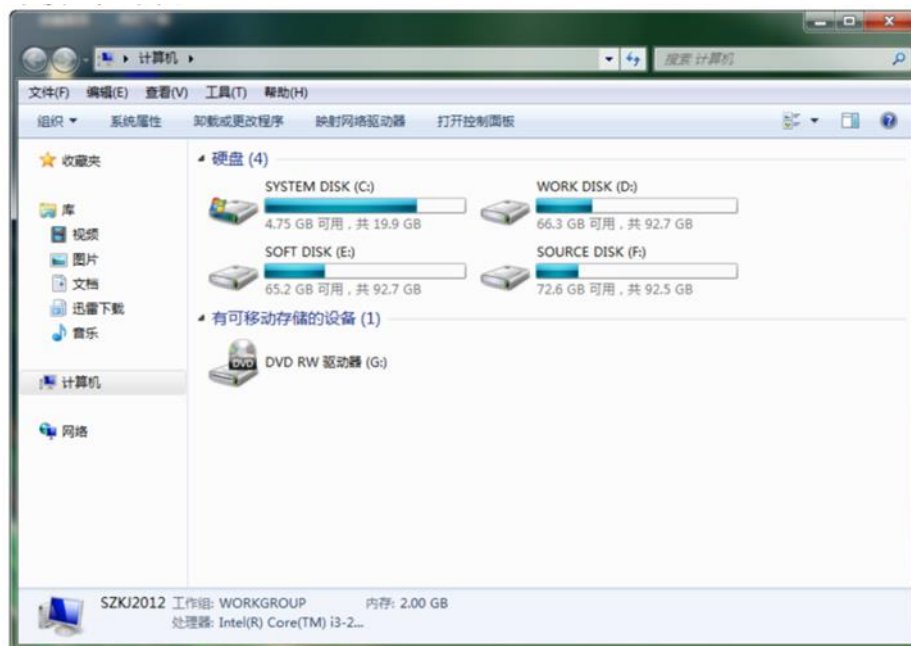
3. 休眠：在基本操作中选择“休眠”，点击发送，此时会提示“模块进入休眠状态，不再响应任何命令”，模块进入休眠后，只能通过重新上电进行唤醒。



USB 控制说明

1. 插入指纹识别仪，指纹仪采用仿真 CD 驱动接口，不需要安装驱动，直接插入既可以使用。

未接入如下图

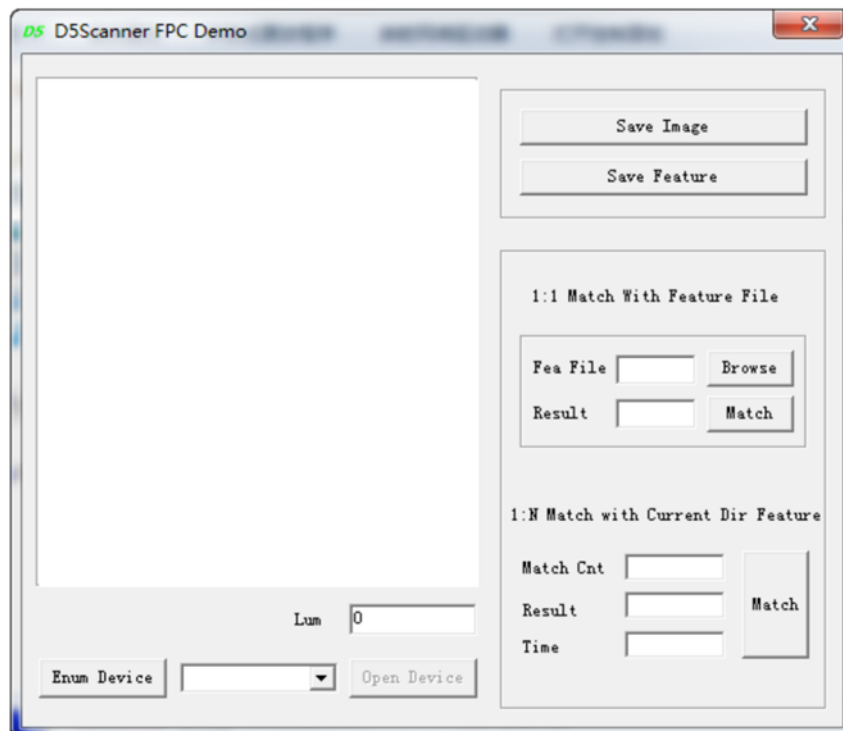


已接入，如下图：（都出来一个 CD 驱动器）

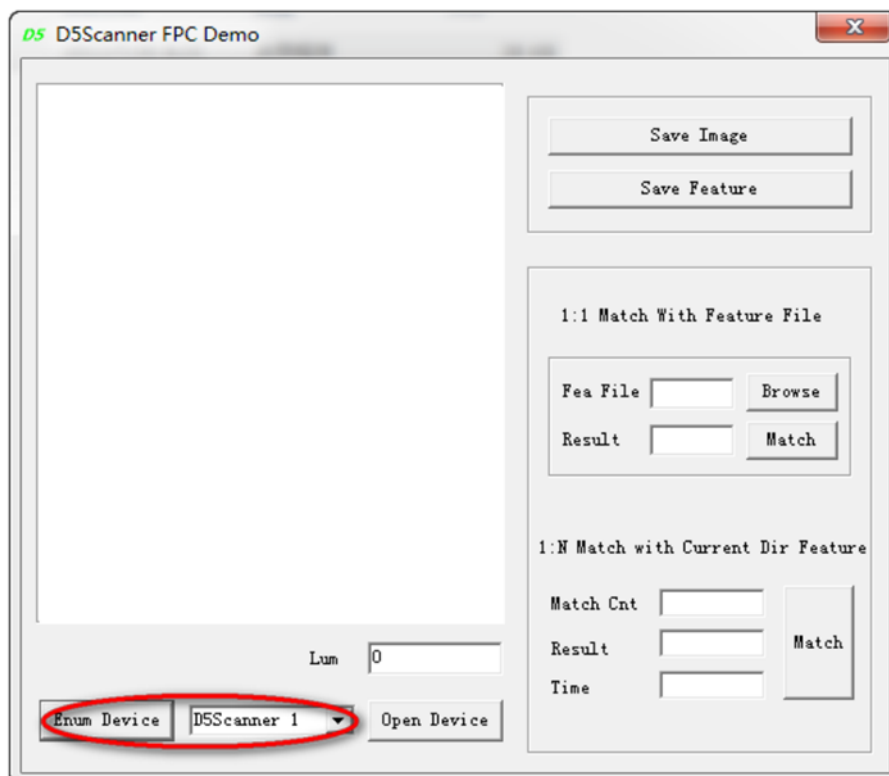


2. 打开测试软件

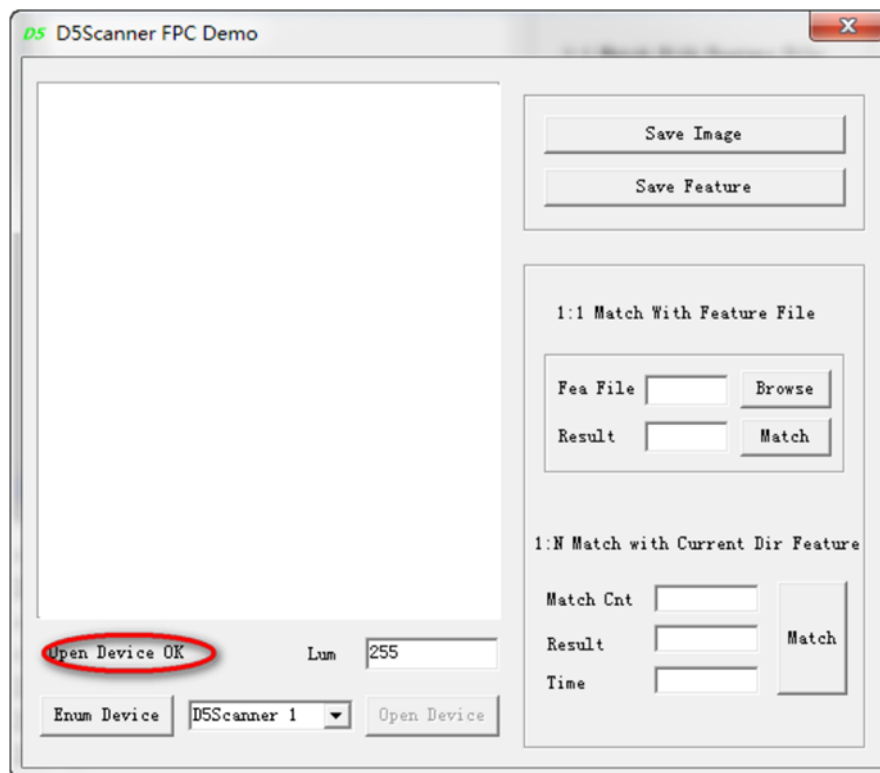
- 2.1. 下载测试软件：Capacitive-Fingerprint-Reader_(UART)，并解压，打开
“D5ScannerS77Demo\Release\D5ScannerS77Demo.exe”



- 2.2. 定义设备，点 Enum Device，枚举当前电脑下所接指纹仪数量，在下选择框中选中所要操作的指纹模块，如图：



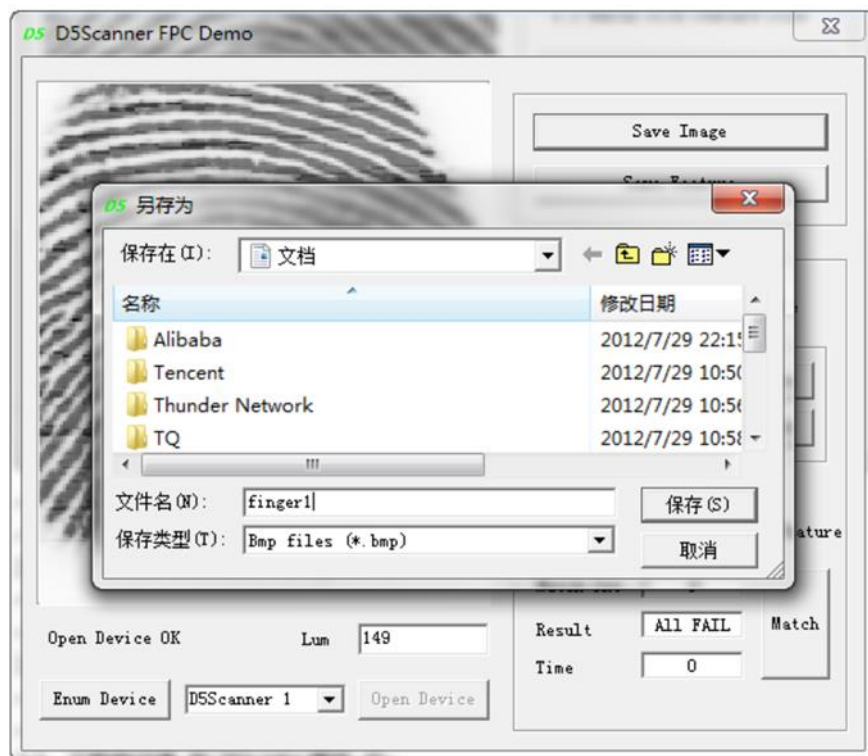
2.3. 打开设备，点：Open Device，提示：Open Device OK 或 Open Device Fail



2.4. 按手指采集指纹，按一次，嘀一声



2.5. 存图像，点：Save Image,保存成.BMP



2.6. 存特征值，点：Save Feature, 保存成*.fea 格式



示例程序

下面以接入微雪 XNUCLEO-F103RB、UNO PLUS 开发板、树莓派 3B 为例介绍模块的使用方法（使用 UART 串口通讯）。

XNUCLEO-F103RB

- 本历程使用的开发板主控芯片为 STM32F103RBT6
- 本例程基于 HAL 库，一次可以用使用 STM32CubeMX 移植程序到其他的 STM 芯片
- 本例程在 Keil v5 环境下编译通过

硬件连接

Capacitive Fingerprint Reader	XNUCLEO-F103RB
VCC	3.3V
GND	GND
TXD	D2 (PA10)
RXD	D8 (PA9)
WAKE	D3 (PB3)
RST	D4 (PB5)

下载程序

1. 将模块连接到开发板后，接上 ST-Link 下载器，打开位于 MDK-ARM 目录下的 Capacitive Fingerprint Reader.uvprojx 文件。点击 Build 编译工程，点击 Download,把工程写入到芯片中
2. 用 USB 转串口线连接开发板和 PC，打开串口监视软件，选择正确的串口号，并设置如下：波特率：115200；数据位：8；停止位：1；校验位：None；控制流：None
3. 按下开发板的复位按键，便可以在串口助手看到如下的操作提示：

```
***** WaveShare Capacitive Fingerprint Reader Test *****
Compare Level: 5 (can be set to 0-9, the bigger, the stricter)
Number of fingerprints already available: 1
Use the serial port to send the commands to operate the module:
CMD1 : Query the number of existing fingerprints
CMD2 : Add fingerprint (Each entry needs to be read two times: "beep", put the finger on sensor, "beep", put up, "beep", put on again)
CMD3 : Fingerprint matching (Send the command, put your finger on sensor after "beep". Each time you send a command, module waits and matches once)
CMD4 : Clear fingerprints
CMD5 : Switch to sleep mode, you can use the finger Automatic wake-up function (In this state, only CMD6 is valid. When a finger is placed on the
sensor, the module is awakened and the finger is matched, without sending commands to match each time. The CMD6 can be used to wake up)
CMD6 : Wake up and make all commands valid
***** WaveShare Capacitive Fingerprint Reader Test *****
```

注：若串口助手窗口一直返回错误提示“***ERROR***”，则请确保模块供电为 3.3 或 5V、串口接线是否正确，模块波特率默认为 19200，最后断电，再重新上电

4. 从以上的操作提示可以知道，当前模块的匹配等级为 5，当前比对等级为：5（可设置为 0-9，数值越大越严格），当前模块中已存在的指纹数为 1，用户可以通过串口助手发送 CMD1-CMD6 来操作模块。注：不要勾选“发送新行”和“HEX 发送”，发送各种命令后，请留

意串口助手回应的各种操作提示！



操作指令说明

CMD1：查询已录入指纹个数

CMD2：指纹录入（每次录入需要读取 2 次，滴，把手指放上探头，滴，拿开，滴，再放上去，重复 2 次），注意尽量把指纹中心对准传感器窗口

CMD3：指纹匹配（发送命令，在滴一声后，把手指放上探头，每发送一次命令，模块会等待并匹配一次），注意尽量把指纹中心对准传感器窗口

CMD4：清空指纹库

CMD5：进入休眠模式，此时可以使用手指自动唤醒功能（发送命令后模块进入休眠状态，此时只有 CMD6 命令有效，当有手指放上探头会唤醒模块，并进行手指匹配，无须每次都发送命令来匹配。可发送 CMD6 退出休眠）

CMD6：切为启动模式，退出休眠并使所有指令有效

UNO PLUS

硬件连接

Capacitive Fingerprint Reader	Arduino UNO PLUS
VCC	3.3V or 5V
GND	GND
TXD	D10
RXD	D11
WAKE	D8
RST	D9

注：由于 Arduino UNO PLUS 只有一个硬件串口，并且已经用来连接 PC 下载程序和作串口调试工具用，因此使用 D10、D11 来模拟软件串口与 Capacitive Fingerprint Reader 通信

下载程序

1. 将示例程序包中的 Arduino\Capacitive_Fingerprint_Reader 文件夹复制到 Arduino 软件的安装目录 Arduino\libraries 下面。点击 File --> Examples --> Capacitive_Fingerprint_Reader--> Finger_Test 打开程序，并编译、下载。
2. 点击 Tools -> Port 选择 Arduino 开发板的串口号，用打开串口监视器，设置 No line ending, 115200baud。按下开发板的复位按键，便可以在串口监视器看到如下操作提示：

```

***** WaveShare Capacitive Fingerprint Reader Test *****
Compare Level: 5    (can be set to 0-9, the bigger, the stricter)
Number of fingerprints already available: 0
Use the serial port to send the commands to operate the module:
CMD1 : Query the number of existing fingerprints
CMD2 : Add fingerprint (Each entry needs to be read two times: "beep", put the finger on sensor, "beep", put up, "beep", put on again)
CMD3 : Fingerprint matching (Send the command, put your finger on sensor after "beep". Each time you send a command, module waits and matches once)
CMD4 : Clear fingerprints
CMD5 : Switch to sleep mode, you can use the finger Automatic wake-up function CMD6 : Wake up and make all commands valid
***** WaveShare Capacitive Fingerprint Reader Test *****

```

注：若串口助手窗口一直返回错误提示 "***ERROR***"，则请确保模块供电为 3.3 或 5V、串口线接线是否正确，模块波特率默认为 19200，最后断电再重新上电
具体的操作方法、命令说明、注意事项，请参考上面 XNUCLEO-F103RB 的使用介绍

RASPBERRY PI

硬件连接

这里使用 Raspberry Pi 3 Model B。引脚连接使用 BCM 管脚编码

Capacitive Fingerprint Reader	RaspberryPi 3B
VCC	3.3V or 5V
GND	GND
TXD	15 (BCM)
RXD	14 (BCM)
WAKE	23 (BCM)
RST	24 (BCM)

注：例程运行前需要安装必要的函数库（WiringPi、bcm2835、python 库），安装方法详见：
http://www.waveshare.net/wiki/Pioneer600_Datasheets

树莓派串口配置

- 由于树莓派串口默认用于终端调试，如需使用串口，则需要修改树莓派设置。执行如下命令进入树莓派配置：

```
sudo raspi-config
```

- 选择 Advanced Options -> Serial -> no, 关闭串口终端功能。(不同版本的 Raspbian 会有区别, 具体可以上网查)
- 打开/boot/config.txt 文件, 找到如下配置语句使能串口, 如果没有, 可添加在文件最后面:
enable_uart=1
- 重启生效。

注: 本例程使用树莓派 3B, 串口设备号为 ttyS0。使用 树莓派 2B/zero 的用户需要把例程中的串口设备号改为 ttyAMA0

PYTHON

把程序下载到树莓派中去, 并安装好需要的函数库, 并且配置好串口

执行命令: `sudo python3.5 main.py` 或 `sudo ./main.py` 运行程序

(程序基于 python3.5 及以上版本)

WIRINGPI

执行命令: `make`, 编译代码, 生成一个名为 `test` 的可执行文件。

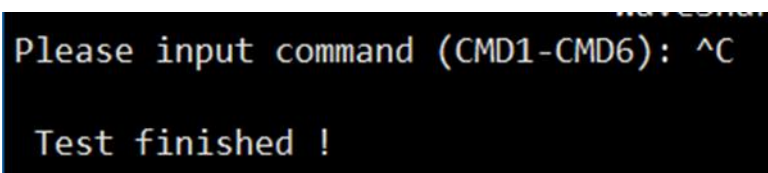
执行命令: `sudo ./test` 运行程序

程序运行结果

具体的操作方法、命令说明、注意事项, 请参考上面 XNUCLEO-F103RB 的使用介绍

```
pi@raspberrypi:~/Siuk/py_demo/Capacitive_Fingerprint_Reader $ python3.5 main.py
***** WaveShare Capacitive Fingerprint Reader Test *****
Compare Level: 5 (can be set to 0-9, the bigger, the stricter)
Number of fingerprints already available: 0
send commands to operate the module:
CMD1 : Query the number of existing fingerprints
CMD2 : Registered fingerprint (Each entry needs to be read two times: "beep", put the finger on sensor, "beep", put up, "beep", put on again)
CMD3 : Fingerprint matching (Send the command, put your finger on sensor after "beep". Each time you send a command, module waits and matches once)
CMD4 : Clear fingerprints
CMD5 : Switch to sleep mode, you can use the finger Automatic wake-up function (In this state, only CMD6 is valid. When a finger is placed on the sensor, the module is awakened and the finger is matched, without sending commands to match each time. The CMD6 can be used to wake up)
CMD6 : Wake up and make all commands valid
***** WaveShare Capacitive Fingerprint Reader Test *****
Please input command (CMD1-CMD6):
```

按下 Ctrl+C 可以退出程序



```
Please input command (CMD1-CMD6): ^C
Test finished !
```

注: 若串口助手窗口一直返回错误提示 "***ERROR***", 则请确保模块供电为 3.3 或 5V、串口线接线是否正确, 模块波特率默认为 19200, 最后断电再重新上电

串口控制开发协议

一、通信方式

DSP 模块作为从设备，由主设备发送相关命令对其进行控制。

命令接口：19200bps 1 起始位 1 停止位（无校验位）

主设备发送的命令及 DSP 模块的应答按数据长度可分为两类：

1) = 8 字节，数据格式如下：

字节	1	2	3	4	5	6	7	8
命令	0xF5	CMD	P1	P2	P3	0	CHK	0xF5
应答	0xF5	CMD	Q1	Q2	Q3	0	CHK	0xF5

说明：

CMD: 命令/应答类型

P1, P2, P3: 命令参数

Q1, Q2, Q3: 应答参数

Q3: 多用于返回操作的有效性信息，此时可有如下取值：

```

#define ACK_SUCCESS      0x00    //操作成功
#define ACK_FAIL         0x01    //操作失败
#define ACK_FULL         0x04    //指纹数据库已满
#define ACK_NOUSER       0x05    //无此用户
#define ACK_USER_EXIST   0x06    //用户已存在
#define ACK_TIMEOUT      0x08    //采集超时
  
```

CHK: 校验和，为第 2 字节到第 6 字节的异或值

2) > 8 字节，数据由两部分组成：数据头+数据包

数据头格式：

字节	1	2	3	4	5	6	7	8
命令	0xF5	CMD	Hi(Len)	Low(Len)	0	0	CHK	0xF5

应答	0xF5	CMD	Hi(Len)	Low(Len)	Q3	0	CHK	0xF5
----	------	-----	---------	----------	----	---	-----	------

说明：

CMD, Q3 的定义同上

Len：数据包内有效数据长度，16 位，由两字节组成

Hi(Len)：数据包长度高 8 位

Low(Len)：数据包长度低 8 位

CHK：校验和，为第 2 字节到第 6 字节的异或值

数据包格式：

字节	1	2...Len + 1	Len + 2	Len + 3
命令	0xF5	Data	CHK	0xF5
应答	0xF5	Data	CHK	0xF5

说明：

Len：即为 Data 的字节数

CHK：校验和，为第 2 字节到第 Len - 2 字节的异或值

发送完数据头后紧接着发送数据包。

二. 各通讯协议命令说明

2.1 使模块进入休眠状态（命令/应答均为 8 字节）

命令数据格式：

字节	1	2	3	4	5	6	7	8
命令	0xF5	0x2C	0	0	0	0	CHK	0xF5

应答数据格式：

字节	1	2	3	4	5	6	7	8
应答	0xF5	0x2C	0	0	0	0	CHK	0xF5

2.2 设置/读取指纹添加模式（命令/应答均为 8 字节）

指纹添加分两种模式：允许重复模式/禁止重复模式，在“禁止重复模式”下，同一枚手指只能添加一个用户，若强行进行第二轮添加将返回错误信息。上电后系统处于禁止重复模式。

字节	1	2	3	4	5	6	7	8
命令	0xF5	0x2D	0	Byte5=0: 0: 允许重复 1: 禁止重复 Byte5=1: 0	0: 设置新的添加模式 1: 读取当前添加模式	0	CHK	0xF5
应答	0xF5	0x2D	0	当前添加模式	ACK_SUCCUSS ACK_FAIL	0	CHK	0xF5

2.3 添加指纹（命令/应答均为 8 字节）

为确保有效性，用户必须录入 3 次指纹，主机须向 DSP 模块发送 3 次命令

i) 第 1 次

字节	1	2	3	4	5	6	7	8
命令	0xF5	0x01	用户号 （高 8 位）	用户号 （低 8 位）	用户权限 （1/2/3）	0	CHK	0xF5
应答	0xF5	0x01	0	0	ACK_SUCCESS ACK_FAIL ACK_FULL ACK_TIMEOUT	0	CHK	0xF5

说明：

用户号的取值范围为 1 – 0xFFF；

用户权限取值范围为 1、 2、 3，其含义由二次开发者自行定义。

ii) 第 2 次

字节	1	2	3	4	5	6	7	8
命令	0xF5	0x02	用户号 (高 8 位)	用户号 (低 8 位)	用户权限 (1/2/3)	0	CHK	0xF5
应答	0xF5	0x02	0	0	ACK_SUCCESS ACK_FAIL ACK_TIMEOUT	0	CHK	0xF5

iii) 第 3 次

字节	1	2	3	4	5	6	7	8
命令	0xF5	0x03	用户号 (高 8 位)	用户号 (低 8 位)	用户权限(1/2/3)	0	CHK	0xF5
应答	0xF5	0x03	0	0	ACK_SUCCESS ACK_FAIL ACK_USER_EXIST ACK_TIMEOUT	0	CHK	0xF5

说明：三次命令中用户号与用户权限应为相同值。

2.4 删除指定用户（命令/应答均为 8 字节）

字节	1	2	3	4	5	6	7	8
命令	0xF5	0x04	用户号 (高 8 位)	用户号 (低 8 位)	0	0	CHK	0xF5

应答	0xF5	0x04	0	0	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5
----	------	------	---	---	-------------------------	---	-----	------

2.5 删除所有用户（命令/应答均为 8 字节）

字节	1	2	3	4	5	6	7	8
命令	0xF5	0x05	0	0	0	0	CHK	0xF5
应答	0xF5	0x05	0	0	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5

2.6 取用户总数（命令/应答均为 8 字节）

字节	1	2	3	4	5	6	7	8
命令	0xF5	0x09	0	0	0	0	CHK	0xF5
应答	0xF5	0x09	用户数 (高 8 位)	用户数 (低 8 位)	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5

2.7 比对 1： 1（命令/应答均为 8 字节）

字节	1	2	3	4	5	6	7	8
命令	0xF5	0x0B	用户号 (高 8 位)	用户号 (低 8 位)	0	0	CHK	0xF5
应答	0xF5	0x0B	0	0	ACK_SUCCESS ACK_FAIL ACK_TIMEOUT	0	CHK	0xF5

2.8 比对 1： N（命令/应答均为 8 字节）

字节	1	2	3	4	5	6	7	8
命令	0xF5	0x0C	0	0	0	0	CHK	0xF5
应答	0xF5	0x0C	用户号 (高 8 位)	用户号 (低 8 位)	用户权限(1/2/3) ACK_NOUSER ACK_TIMEOUT	0	CHK	0xF5

2.9 取用户权限（命令/应答均为 8 字节）

字节	1	2	3	4	5	6	7	8
命令	0xF5	0x0A	用户号 (高 8 位)	用户号(低 8 位)	0	0	CHK	0xF5
应答	0xF5	0x0A	0	0	用户权限 (1/2/3) ACK_NOUSER	0	CHK	0xF5

2.10 取 DSP 模块版本号（命令为 8 字节/应答>8 字节）

命令数据格式：

字节	1	2	3	4	5	6	7	8
命令	0xF5	0x26	0	0	0	0	CHK	0xF5

应答数据格式：

1) 数据头：

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

2) 数据包：

字节	1	2 --- Len + 1	Len + 2	Len + 3
应答	0xF5	版本数据	CHK	0xF5

说明：此协议暂不公开

2.11 设置/读取比对等级（命令/应答均为 8 字节）

字节	1	2	3	4	5	6	7	8
命令	0xF5	0x28	0	Byte5=0: 新比对等级; Byte5=1: 0	0: 设置新的比对等级 1: 读取当前比对等级	0	CHK	0xF5
应答	0xF5	0x28	0	当前比对等级	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5

说明：比对等级取值为 0-9，取值越大比对越严格，默认值为 5

2.12 采集图像并上传（命令为 8 字节/应答>8 字节）

命令数据格式：

字节	1	2	3	4	5	6	7	8
命令	0xF5	0x24	0	0	0	0	CHK	0xF5

应答数据格式：

1) 数据头：

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

2) 数据包：

字节	1	2 --- Len + 1	Len + 2	Len + 3
----	---	---------------	---------	---------

应答	0xF5	图像数据	CHK	0xF5
----	------	------	-----	------

说明：

在 DSP 模块中，指纹图像为 200*264 像素，每个像素灰度由 8 位表示。在上传过程中，为了减小数据量，在横/纵方向进行跳像素采样，这样图像变为 100*132，并取灰度的高 4 位，每两个像素合成一个字节传输（前一像素在低四位，后一像素在高四位）。

传输从第一行开始逐行进行，每一行从第一个像素开始，总共传输 100*132/2 个字节的的数据。

图像数据长度 Len 恒为 6600 字节。

// 具体产品型号不同，此数值可能有变动或差别，如有用到此命令，请与我公司技术人员联系！

2.13 采集图像并提取特征值上传（命令为 8 字节/应答>8 字节）

命令数据格式：

字节	1	2	3	4	5	6	7	8
命令	0xF5	0x23	0	0	0	0	CHK	0xF5

应答数据格式：

1) 数据头：

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

2) 数据包：

字节	1	2	3	4	5 --- Len + 1	Len + 2	Len + 3
应答	0xF5	0	0	0	特征值 数据	CHK	0xF5

说明：特征值数据长度 Len - 3 恒为 193 字节。

2.14 下传特征值与采集指纹比对（命令>8 字节/应答为 8 字节）

命令数据格式：

1) 数据头：

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

2) 数据包：

字节	1	2	3	4	5 --- Len + 1	Len + 2	Len + 3
命令	0xF5	0	0	0	特征值数据	CHK	0xF5

说明：特征值数据长度 Len - 3 恒为 193 字节。

应答数据格式：

字节	1	2	3	4	5	6	7	8
应答	0xF5	0x44	0	0	ACK_SUCCESS ACK_FAIL ACK_TIMEOUT	0	CHK	0xF5

2.15 下传指纹特征值与 DSP 模块数据库指纹比对 1： 1（命令>8 字节/应答为 8 字节）

命令数据格式：

1) 数据头：

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

2) 数据包：

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

命令	0xF5	用户号 (高 8 位)	用户号 (低 8 位)	0	特征值 数据	CHK	0xF5
----	------	----------------	----------------	---	-----------	-----	------

说明：特征值数据长度 $Len - 3$ 恒为 193 字节。

应答数据格式：

字节	1	2	3	4	5	6	7	8
应答	0xF5	0x42	0	0	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5

2.16 下传指纹特征值与 DSP 模块数据库指纹比对 1： N (命令>8 字节/应答为 8 字节)

命令数据格式：

1) 数据头：

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

2) 数据包：

字节	1	2	3	4	5 --- Len + 1	Len + 2	Len + 3
命令	0xF5	0	0	0	特征值 数据	CHK	0xF5

说明：特征值数据长度 $Len - 3$ 恒为 193 字节。

应答数据格式：

字节	1	2	3	4	5	6	7	8
应答	0xF5	0x43	用户号 (高 8 位)	用户号 (低 8 位)	用户权限 (1/2/3) ACK_NOUSER	0	CHK	0xF5

2.17 上传 DSP 模块数据库内指定用户特征值（命令为 8 字节/应答>8 字节）

命令数据格式：

字节	1	2	3	4	5	6	7	8
命令	0xF5	0x31	用户号（高 8 位）	用户号（低 8 位）	0	0	CHK	0xF5

应答数据格式：

1) 数据头：

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

2) 数据包：

字节	1	2	3	4	5 --- Len + 1	Len + 2	Len + 3
应答	0xF5	用户号（高 8 位）	用户号（低 8 位）	用户权限（1/2/3）	特征值数据	CHK	0xF5

说明：

特征值数据长度 Len - 3 恒为 193 字节。

2.18 下传特征值并按指定用户号存入 DSP 模块数据库（命令>8 字节/应答为 8 字节）

命令数据格式：

1) 数据头：

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

2) 数据包：

字节	1	2	3	4	5--- Len + 1	Len + 2	Len + 3
命令	0xF5	用户号 (高 8 位)	用户号 (低 8 位)	用户权限 (1/2/3)	特征值数据	CHK	0xF5

说明：特征值数据长度 Len - 3 恒为 193 字节

应答数据格式：

字节	1	2	3	4	5	6	7	8
应答	0xF5	0x41	存入 用户 号（高 8 位）	存入 用户 号（低 8 位）	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5

2.19 取已登录所有用户用户号及权限（命令为 8 字节/应答>8 字节）

命令数据格式：

字节	1	2	3	4	5	6	7	8
命令	0xF5	0x2B	0	0	0	0	CHK	0xF5

应答数据格式：

1) 数据头：

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

2) 数据包：

字节	1	2	3	4 --- Len + 1	Len + 2	Len + 3
----	---	---	---	---------------	---------	---------

应答	0xF5	用户数 (高 8 位)	用户数 (低 8 位)	用户信息 数据 (用 户号及权 限)	CHK	0xF5
----	------	----------------	----------------	-----------------------------	-----	------

说明:

数据包中数据长度 Len 恒为 "3 * 用户数 + 2"。

用户信息数据格式如下:

字节	4	5	6	7	8	9	...
数据	用户号 1(高 8 位)	用户号 1(低 8 位)	用户 1 权限 (1/2/3)	用户号 2(高 8 位)	用户号 2(低 8 位)	用户 2 权限 (1/2/3)	...

2.20 取单条记录数据 (命令为 8 字节/应答>8 字节) -- 注: 此协议模块内暂不提供
此协议返回记录库中由“记录位置”指定的记录数据。

命令数据格式:

字节	1	2	3	4	5	6	7	8
命令	0xF5	0x38	记录位 置 (高 8 位)	记录位 置 (低 8 位)	0	0	CHK	0xF5

应答数据格式:

1) 数据头:

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

2) 数据包:

字节	1	2	3	4	5
----	---	---	---	---	---

应答	0xF5	位 7-1: 年 位 0: 月 (位 3)	位 7-5: 月 (位 2-0) 位 4-0: 日	位 7-2: 时 位 1-0: 分 (位 5-4)	位 7-4: 分 (位 3-0) 位 3-0: 记录号(位 21-18)
----	------	-----------------------------	---------------------------------	---------------------------------	--

字节	6	7	8	9	10	11
应答	记录号 (位 17-10)	记录号 (位 9-2)	位 7-6: 记录号 (位 1-0) 位 5-0: 用户号(位 13-8)	用户号 (位 7-0)	CHK	0xF5

说明:

记录数据长度 Len 恒为 8;

2.21 取新记录数据 (命令为 8 字节/应答>8 字节) -- 注: 此协议模块内暂不提供

此协议返回记录库中记录号大于等于“最小记录号”的接续 50 条记录数据。

命令数据格式:

字节	1	2	3	4	5	6	7	8
命令	0xF5	0x39	位 7-6: 0 位 5-0: 最小记录号 (位 21-16)	最小记录号 (位 15-8)	最小记录号(位 7-0)	0	CHK	0xF5

应答数据格式:

1) 数据头:

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

2) 数据包:

字节	1	2---9	10-17	...	Len + 2	Len + 3
应答	0xF5	第 1 条记录	第 2 条记录	...	CHK	0xF5

说明:

数据包中的每条记录格式同 2. 20 中应答数据包中字节 2---字节 9。

数据长度 Len 恒为 (8 * 50 = 400 字节)。

2.22 清空记录数据 (命令/应答均为 8 字节) -- 注: 此协议模块内暂不提供

字节	1	2	3	4	5	6	7	8
命令	0xF5	0x3A	0	0	0	0	CHK	0xF5
应答	0xF5	0x3A	0	0	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5

2.23 设置模块时间 (命令>8 字节/应答为 8 字节) -- 注: 此协议模块内暂不提供命令数据格式:

1) 数据头:

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

2) 数据包:

字节	1	2	3	4	5	6	7	8	9	10
命令	0xF5	星期	年	月	日	时	分	秒	CHK	0xF5

说明：

时间数据长度 **Len** 恒为 7。

应答数据格式：

字节	1	2	3	4	5	6	7	8
应答	0xF5	0x48	0	0	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5

2.24 读取系统时间 -- 注：此协议模块内暂不提供

命令数据格式：

字节	1	2	3	4	5	6	7	8
命令	0xF5	0x3C	0	0	0	0	CHK	0xF5

应答数据格式：

1) 数据头：

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

2) 数据包：

字节	1	2	3	4	5	6	7	8	9	10
应答	0xF5	星期	年	月	日	时	分	秒	CHK	0xF5

说明：时间数据长度 **Len** 恒为 7。

2.25 设置/读取指纹采集等待超时时间（命令/应答均为 8 字节）

字节	1	2	3	4	5	6	7	8
命令	0xF5	0x2E	0	Byte5=0: 新超时时间;	0: 设置新的超时时间	0	CHK	0xF5

				Byte5=1: 0	1: 读取当前超 时时间			
应答	0xF5	0x2E	0	当前超时 时间	ACK_SUCCUSS ACK_FAIL	0	CHK	0xF5

说明:

指纹等待超时时间 (tout) 范围为 0-255。若此值为 0, 若无指纹按压则指纹采集过程将一直持续; 若此值非 0, 在 tout * T0 时间内若无指纹按压则系统将超时退出。

注: T0 为采集/处理一幅图像所需的时间, 一般为 0.2-0.3s。

USB SDK 开发包协议

```
#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 枚举设备

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

说明：

枚举已连接设备(每台电脑最多接 5 台)

参数：

pDeviceName: 已连接设备名称

返回：

0: 无连接设备

其它: 已连接设备数目

2 打开设备

long WINAPI D5OpenDevice (unsigned short uDeviceID);

说明：

打开设备(注:某设备如果已被打开,则需先关闭才能再打开)

参数：

uDeviceID: 设备号(0-4)

返回：

0: 打开成功

-1: 操作失败

3 关闭设备

long WINAPI D5CloseDevice (unsigned short uDeviceID);

说明：

关闭设备

参数：

uDeviceID: 设备号(0-4)

返回：

0: 关闭成功

-1: 操作失败

4 设置公司标记 KEY

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

说明：

设置公司标记

参数：

uDeviceID：设备号（0-4）

pMark：公司标记（8 字节）

返回：

0：设置成功

-1：操作失败

5 检验公司标记 key

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

说明：

检验公司标记

参数

uDeviceID：设备号（0-4）

pMark：公司标记（8 字节）

返回：

- 0：检验匹配
- 1：检验不匹配
- 1：操作失败

6 设备蜂鸣提示

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

说明：

设备蜂鸣

参数：

- uDeviceID：设备号（0-4）
- uMS：蜂鸣时间（毫秒为单位）

返回：

- 0：蜂鸣成功
- 1：操作失败

7 打开 LED 灯提示

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

说明：

打开 LED

参数：

- uDeviceID：设备号（0-4）
- uLEDS：LED 组合（红-Bit0/绿-Bit1/蓝-Bit2）

返回：

- 0：打开成功
- 1：操作失败

8 关闭 LED 灯

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

说明：

关闭 LED

参数：

uDeviceID：设备号（0-4）

uLEDs：LED 组合（红-Bit0/绿-Bit1/蓝-Bit2）

返回：

0：关闭成功

-1：操作失败

9 从位图中载入图像数据

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

说明：

从位图中载入图像数据

参数：

strFileName：位图文件名

pImage：图像数据

宽度 192，高度 256，数据按行顺序排列，每个像素用 1 字节表示灰度

返回：

0：载入成功

-1：文件无法打开或格式不对

10 将图像数据存储为位图

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

说明：

将图像数据存储为位图

参数：

strFileName：位图文件名

pImage：图像数据

宽度 192，高度 256，数据按行顺序排列，每个像素用 1 字节表示灰度

返回：

0：存储成功

-1：文件无法打开

11 从设备中读取图像

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

说明：

读取图像

参数：

wDeviceID：设备号（0-4）

pImage：图像数据

宽度 192, 高度 256, 数据按行顺序排列, 每个像素用 1 字节表示灰度

返回：

0：读取成功

-1：操作失败

12 检测某图像是否有指纹

```
bool WINAPI D5CheckFP (unsigned char *pImage);
```

说明：

检测某图像是否有指纹

参数：

pImage：图像数据

宽度 192, 高度 256, 数据按行顺序排列, 每个像素用 1 字节表示灰度

返回：

1：检测到指纹

0：无指纹

13 指纹转成特征值

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

说明：

对输入的指纹图像数据提取特征值

参数：

pImage : 输入的指纹图像数据

宽度 192, 高度 256, 数据按行顺序排列, 每个像素用 1 字节表示灰度

pFeature : 生成的指纹特征值 (256 非符号字节的数组)

返回 :

0 : 提取成功

-1 : 提取失败

-2 : 系统错误

14 两个指纹特征值进行 1:1 比对

```
long WINAPI D5Match (unsigned char *pFeature1,unsigned char *pFeature2,  
unsigned short uRotate = 60,unsigned short uLevel = 5);
```

说明 :

对输入的两个指纹特征值进行比对

参数 :

pFeature1 : 指纹特征值 1

pFeature2 : 指纹特征值 2

uRotate : 旋转角度 (1-180)

uLevel : 匹配等级 (0-9)

返回 :

0 : 匹配成功

-1 : 匹配失败

-2 : 系统错误

15 对输入的指纹特征值与指纹库内特征值进行快速 1 : N 比对

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

说明 :

对输入的指纹特征值与指纹库内特征值进行快速 1 : N 比对

参数 :

pFeatureIn: 需比对的特征值

pFeatureLib : 指纹特征值库数组

lFingernum：指纹库指纹数，数目不限

uRotate：旋转角度（1-180）

uLevel：匹配等级（0-9）

返回：

-1：匹配不成功

-2：系统错误

其他：匹配成功的指纹库内指纹下标

附录：应用方案

一、典型应用方案：

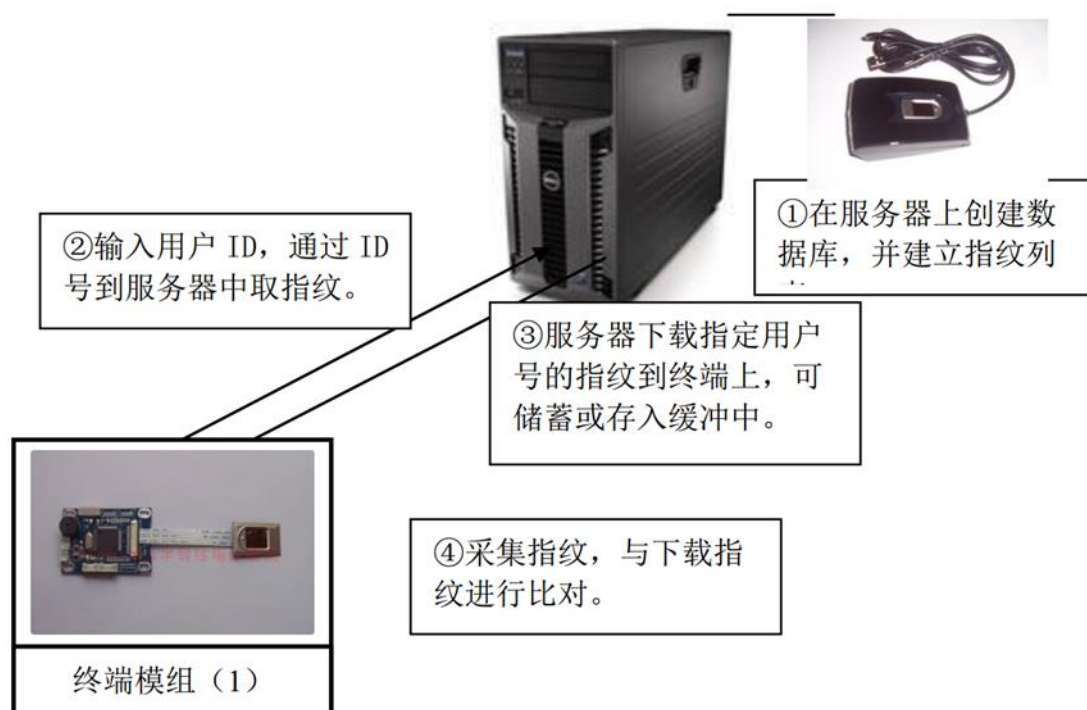
指纹采集应用方案之一：终端采集，终端比对



指纹采集应用方案之一：终端采集，服务器比对



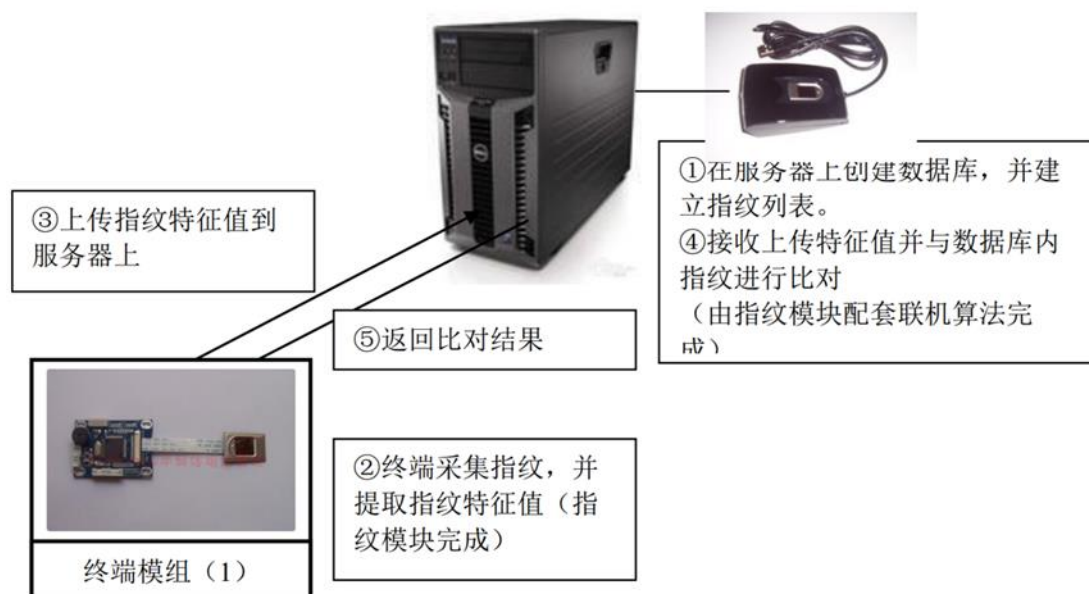
联机比对应用方案之一：终端输入 ID，取指纹数据库中指定指纹，终端比对



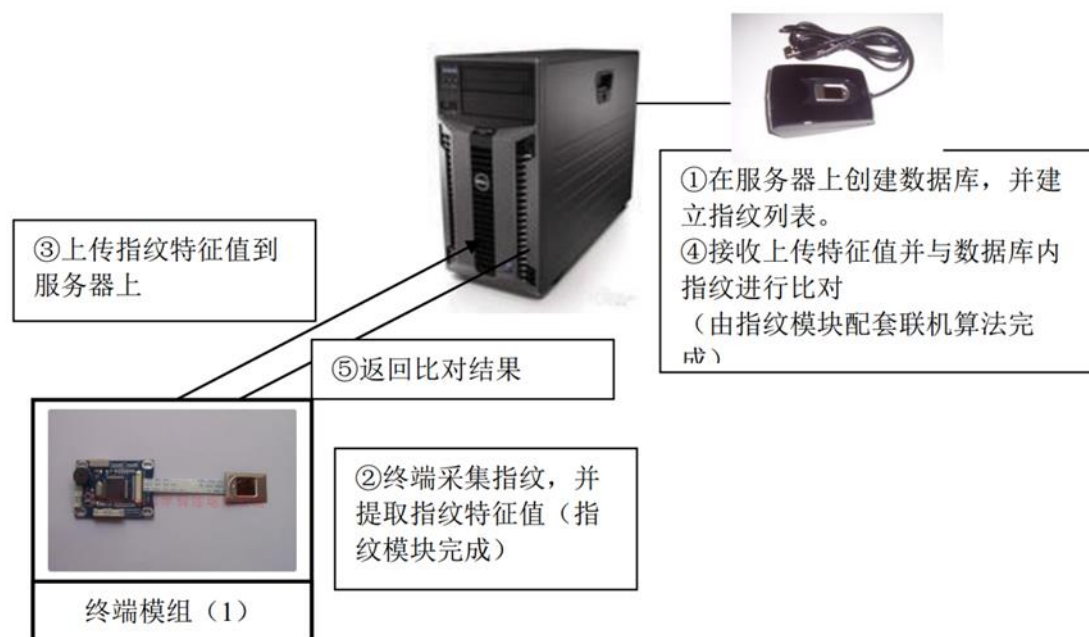
联机比对应用方案之二：分类（批）下载指纹到终端，终端比对



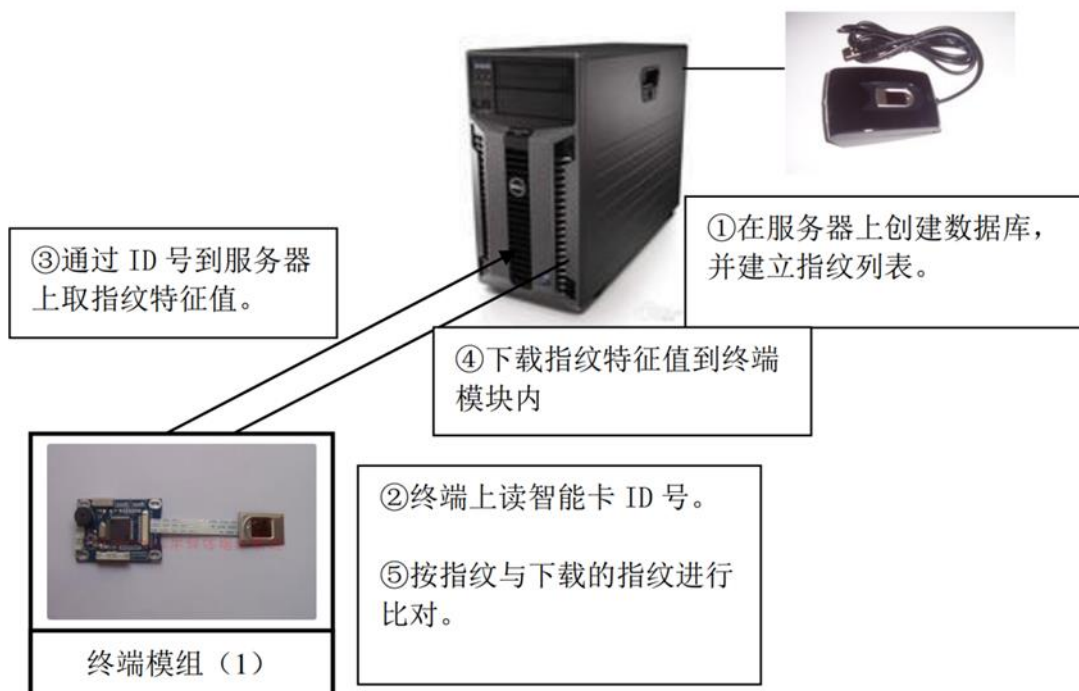
联机比对应用方案之三：终端采集指纹，上传服务器 1:1 比对



联机比对应用方案之四：终端采集指纹，上传服务器 1：N 比对



指纹卡应用方案之一：卡号取数据库指纹，下载终端与采集指纹比对



指纹卡应用方案之二：指纹存于卡内，下载卡内指纹与采集指纹对比

