



UART Fingerprint Sensor (C)

用户手册

目录

产品简介.....	2
特性.....	2
参数.....	2
硬件说明.....	3
尺寸.....	3
接口说明.....	3
串口通信协议.....	4
通信方式.....	4
命令类型.....	5
通信协议操作流程示例.....	13
添加指纹.....	13
删除指定用户.....	14
删除全部用户.....	14
采集图像并提取特征值上传.....	15
使用说明.....	16
连接 PC 使用.....	16
硬件准备和连接.....	16
硬件测试.....	16
连接 XNUCLEO-F103RB 使用.....	18
连接树莓派使用.....	18

产品简介

UART Fingerprint Sensor (C)是一款高度集成的一体式电容指纹膜识别模块。将指纹识别主板和电容式指纹模块传感器集成在一体，极大的缩小了产品体积。圆形设计，支持 360°旋转识别。

采用高性能 Cortex 内核的主控，集成了高安全性商用指纹算法，支持指纹录入，图像采集，特征值提取，模板生成，模板存储和指纹比对。

特性

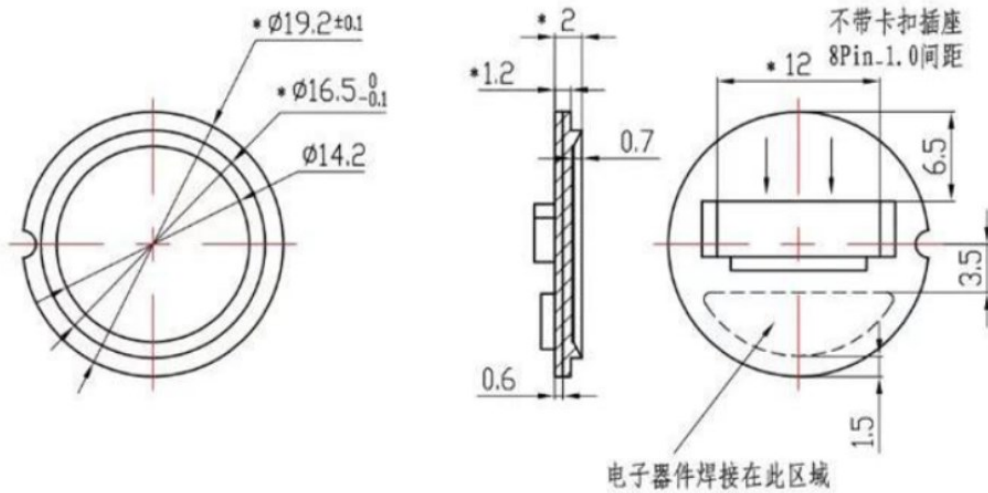
- 使用方便，无需了解指纹算法和内部结构，只需简单的指令就能完成相应操作
- 内置商用算法，支持指纹录入、指纹比对、提取指纹图像和上传指纹特征值等功能，性能稳定，识别速度快
- 电容式指纹识别，感应灵敏，手指只要轻轻地触碰采集窗就能快速识别
- 硬件集成度高，主控和指纹传感器集成一体，可方便嵌入到各种小体积应用
- 不锈钢窄边圆形结构，指纹传感器接触面积大，支持 360 度旋转识别
- 内置人体感应器件，具有掉电睡眠和触摸唤醒功能，功耗更低
- 板载 UART 接口，方便接入 STM32 和 Raspberry Pi 等硬件平台

参数

传感器类型	电容式触摸式传感器
分辨率	508DPI
图像像素阵列	192x192
图像灰度等级	8 级灰度
传感器尺寸	R15.5mm
指纹容量	500 枚
比对时间	<500ms (1:N, 且 N≤100)
认假率	<0.001%
拒真率	<0.1%
工作电压	2.7~3.3v
工作电流	<50mA
睡眠电流	<16uA
静电测试	接触放电 8KV/空气放电 15KV
通信接口	UART
通信波特率	19200 bps
工作环境	温度： -20°C~70°C
	湿度： 40%RH~85%RH (无凝霜)
存储环境	温度： -40°C~70°C
	湿度： <85%RH (无凝霜)
使用寿命	一百万次

硬件说明

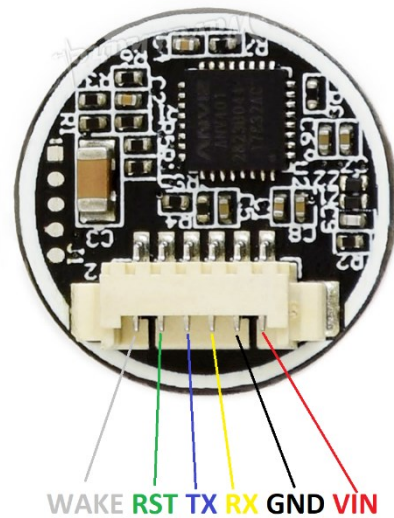
尺寸



接口说明

注意：实际配送的排线颜色可能会跟下方图片有差异，使用的时候请对照引脚定义连接，不要单纯依赖线材颜色。

- VIN：电源 3.3V
- GND：地
- RX：串口数据输入（TTL 电平）
- TX：串口数据输出（TTL 电平）
- RST：休眠控制接口：
 - 高电平：模块供电-启动
 - 低电平：模块掉电-休眠
- WAKE：唤醒引脚，当 RST 处于低电平（休眠）时，手指触摸采集器表面会激活指纹模块，此引脚输出高电平



串口通信协议

通信方式

模块作为从设备，由主设备发送相关指令对其进行控制。指令接口：UART 19200bps 8 位数据为，1 位停止位（无校验）

主设备发送的指令及 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，此时可以有如下取值：

```
#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_OCCUPIED    0x06 //用户已存在

#define ACK_FINGER_OCCUPIED  0x07 //指纹已存在

#define ACK_TIMEOUT          0x08 //采集超时
```

CHK: 校验和，为第 1 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+1 字节的异或值

发送完数据后之后紧接着发送数据包。

命令类型

1. 修改模块序列号 (命令/应答均为 8 字节)

字节	1	2	3	4	5	6	7	8
命令	0xF5	0x08	新序列号 (位 23-16)	新序列号 (位 15-8)	新序列号 (位 7-0)	0	CHK	0xF5
应答	0xF5	0x08	旧序列号 (位 23-16)	旧序列号 (位 15-8)	旧序列号 (位 7-0)	0	CHK	0xF5

说明:

序列号为 24 位的一个常数, 每个 DSP 模块不一样, 可以用于区别不同的 DSP 模块

2. 取 DSP 模块内部序列号 (命令/应答均为 8 字节)

字节	1	2	3	4	5	6	7	8
命令	0xF5	0x2A	0	0	0	0	CHK	0xF5
应答	0xF5	0x2A	序列号 (位 23-16)	序列号 (位 15-8)	序列号 (位 7-0)	0	CHK	0xF5

3. 使模块进入休眠状态 (命令/应答均为 8 字节)

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

说明: 此状态下, 手指触摸触发 WAKE 引脚的唤醒方法无效

4. 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

5. 添加指纹 (命令/应答均为 8 字节)

为确保有效性, 用户需录入 3 次指纹, 主机需向 DSP 模块发送 3 次命令

a) 第一次

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_USER_OCCUPIED ACK_FINGER_OCCUPIED ACK_TIMEOUT	0	CHK	0xF5

说明:

用户号的取值范围为 1~0xFFF;

用户号权限取值范围为 1,2,3, 其含义由用户自行定义

b) 第二次

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

c) 第三次

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_TIMEOUT	0	CHK	0xF5

说明: 3 次命令中用户号与用户权限必须是相同的

6. 添加用户并将特征值上传 (命令为 8 字节/应答>8 字节)

此命令的流程与”5.添加指纹”命令类似, 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。

a) 第一次

与”5.添加指纹”中第一次命令、应答一样

- b) 第二次
与“5.添加指纹”中第二次命令、应答一样
- c) 第三次

命令数据格式:

字节	1	2	3	4	5	6	7	8
命令	0xF5	0x06	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	0x06	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

数据包当应答数据第 5 字节为 ACK_SUCCESS 是返回

7. 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

8. 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: 删除全部用户 1/2/3: 删除权限为 1/2/3 的全部用户	0	CHK	0xF5
应答	0xF5	0x05	0	0	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5

9. 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: 取用户总数 0xFF: 取指纹容量	0	CHK	0xF5
应答	0xF5	0x09	用户数 /指纹 容量	用户数 /指纹 容量	ACK_SUCCESS ACK_FAIL	0	CHK	0xF5

			(高 8 位)	(低 8 位)	0xFF (如果命令为取容量)			
--	--	--	---------	---------	-----------------	--	--	--

10. 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

11. 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

12. 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

13. 设置/读取对比等级 (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 的比对等级	0	CHK	0xF5
应答	0xF5	0x28	0	当前的比对 等级	ACK_SUCCUSS ACK_FAIL	0	CHK	0xF5

说明:

比对等级为 0-9, 取值越大比对越严格, 默认值为 5

14. 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_SUCCUSS 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 280*280 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 140*140, 并取灰度的高 4 位, 每两个像素合成一个字节传输 (前一像素在高四位, 后一像素在低四位)。

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 140*140/2 个字节的数
据。

图像数据长度 Len 恒为 9800 字节。

15. 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_SUCCUSS 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。

16. 上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
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。

应答数据格式:

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

17. 上传特征值与 DSP 模块数据库指纹比对 1:1 (命令为>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+2
应答	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	0x43	0	0	ACK_SUCCUSS ACK_FAIL	0	CHK	0xF5

18. 上传特征值与 DSP 模块数据库指纹比对 1:N (命令为>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+2
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。

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

19. 下载 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_SUCCUSS 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。

20. 上传特征值并按指定用户号存入 DSP 模块数据库 (命令为>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
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。

应答数据格式:

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

21. 取已登录所有用户的用户号及权限 (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_SUCCUSS ACK_FAIL	0	CHK	0xF5

2) 数据包

字节	1	2	3	4--Len+1	Len+2	Len+3
应答	0xF5	用户号 (高 8 位)	用户号 (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*用户号+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)	用户号 2 权限 (1/2/3)	...

22. 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 Byte5=1: 0	0: 设置新的超 时时间 1: 读取当前的 超时时间	0	CHK	0xF5
应答	0xF5	0x2E	0	当前超时时 间	ACK_SUCCUSS ACK_FAIL	0	CHK	0xF5

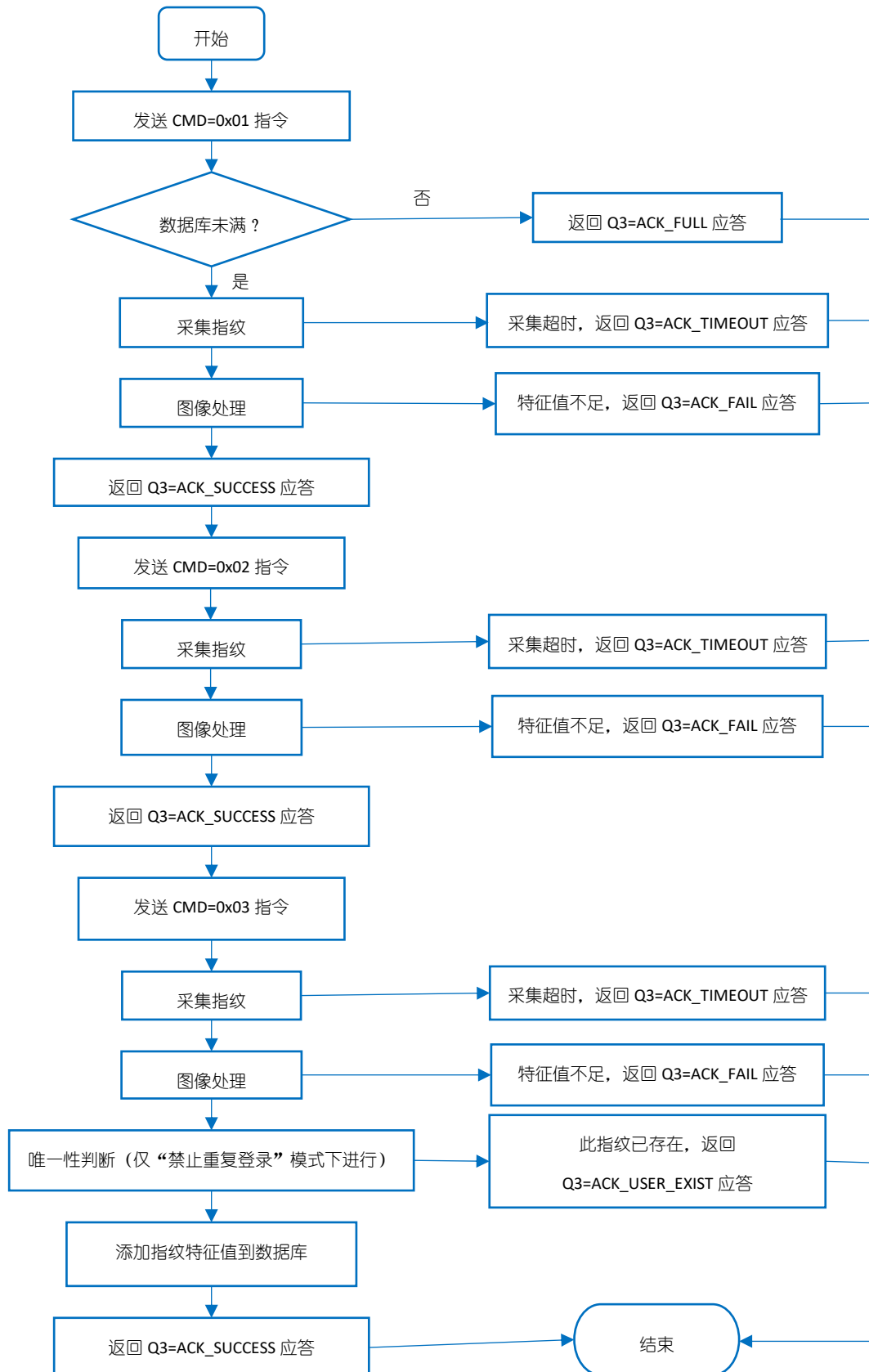
说明:

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

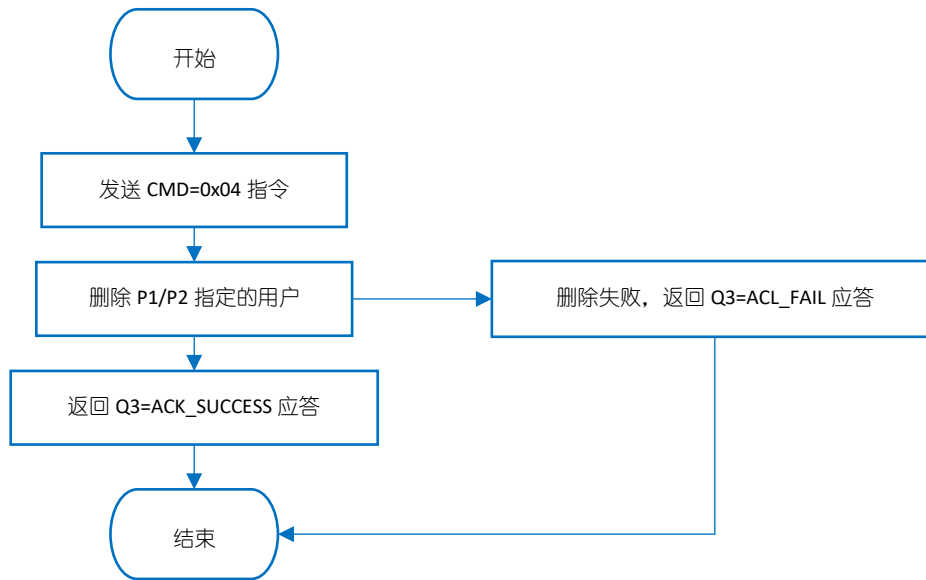
注: T0 为采集/处理一幅图像所需的时间, 一般为 0.2-0.3s

通信协议操作流程示例

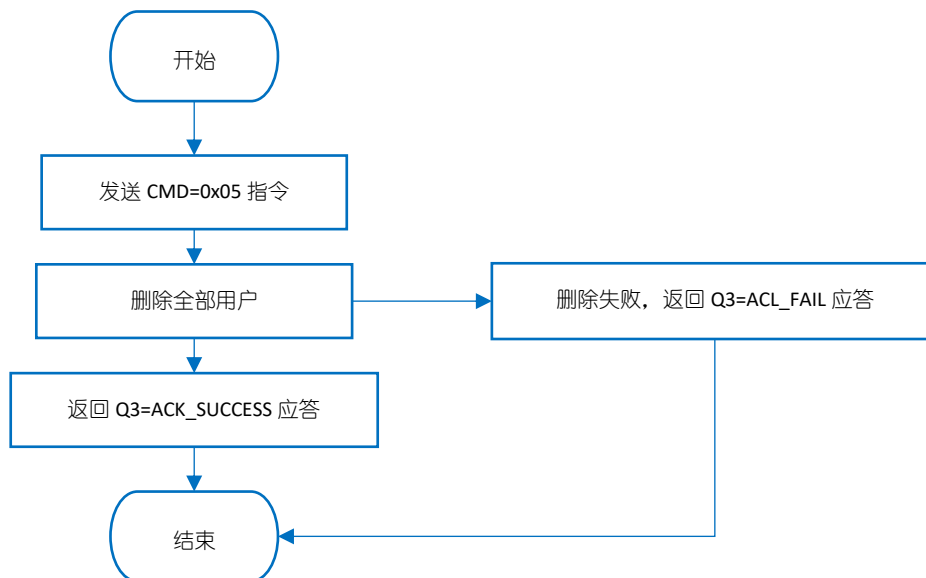
添加指纹



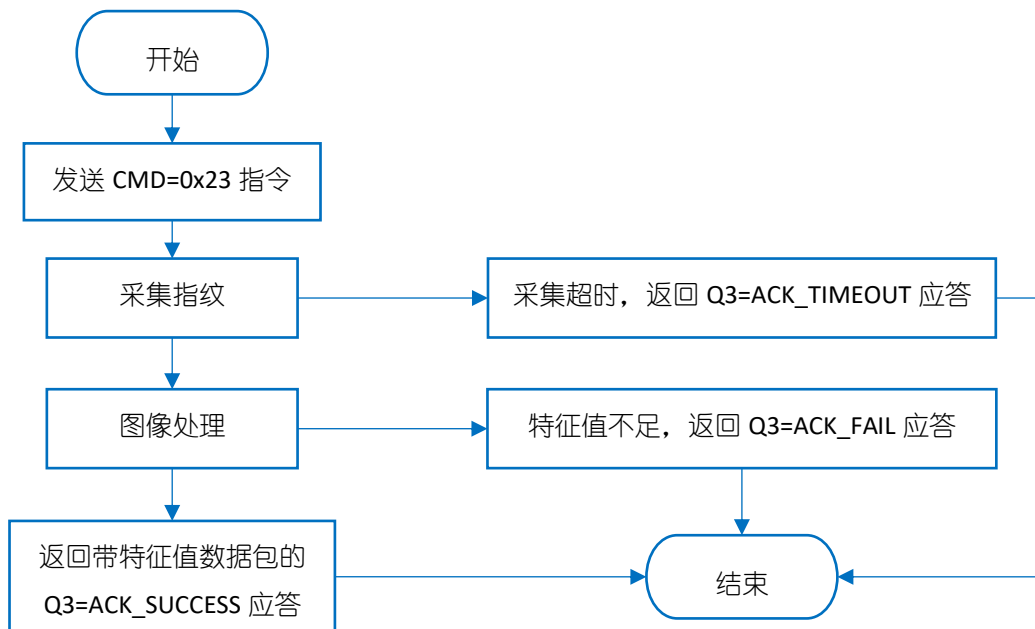
删除指定用户



删除全部用户



采集图像并提取特征值上传



使用说明

如果您是要连接指纹模块到 PC 上使用，您还需要准备一个 UART 转 USB 模块。这里我们推荐微雪电子 [FT232 USB UART Board \(micro\)](#) 模块。

如果您是要连接指纹模块到开发板使用，如果您的开发板是 3.3V 工作环境，直接连接指纹模块到开发板的 UART 接口上和 GPIO 即可。如果您的开发板是 5V 工作环境，建议添加电平转换模块。

连接 PC 使用

硬件准备和连接

- UART Fingerprint Sensor (C) *1
- FT232 USB UART Board *1
- micro USB 线（传说中的安卓手机线）*1

将指纹模块按照一下连接方式连接到 FT232 USB UART Board 上，然后连接到电脑。

UART Fingerprint Sensor (C)	FT232 USB UART Board
Vcc	Vcc
GND	GND
RX	TX
TX	RX
RST	不接
WAKE	不接

硬件测试

- 从产品百科下载测试软件
- 打开软件，如果第一次打开软件无法正常使用，需要运行一下注册控件文件（以管理员身份）注册控件，然后再打开
- 设置栏，选择正确的串口号。注意这里串口号只支持 COM1 到 COM8，如果 UART 转 USB 模块在你的电脑中识别到的串口不在这个范围，需要修改一下串口号。

● 测试栏



测试栏中提供了多种功能。

1. 取用户总数：
选择取用户总数，然后点击发送命令。在返回数据显示中即可显示用户总数
2. 用户录入：
选择用户录入，勾选采集两次以及 ID 号自增 1。然后在下方的参数选择里面填入参数。这里参数 1 和参数 2 是用户号，参数 3 是用户权限。然后点击发送命令。之后触摸指纹采集器进行录入即可。
3. 删除指定用户：
选择删除指定用户，然后再下方的参数选择里面填入要删除的用户号，然后点击发送命令
4. 删除全部用户：
选择删除全部用户，然后点击发送命令。
5. 1:1 比对：
选择 1:1 比对，然后在下方的参数选择里面填入要比对的用户号，然后点击发送命令
6. 1:N 比对：
选择 1: N 比对，然后点击发送命令

....

更多功能，请用户自行探索。（注意，软件上提供的有些是不适用本模块的，具体以上面的通信协议为准）

连接 XNUCLEO-F103RB 使用

我们提供有 XNUCLEO-F103RB 例程。用户可以在产品百科界面下载。

UART Fingerprint Sensor (C)	XNUCLEO-F103RB
Vcc	3.3V
GND	GND
RX	PA9
TX	PA10
RST	PB5
WAKE	PB3

备注：指纹模块的引脚，请参照上方[接口说明](#)

1. 将指纹模块连接到 XNUCLEO-F103RB 开发板，并连接好下载器
2. 用 keil5 打开示例程序工程
3. 确定好下载器连接正常，开发板正常识别。
4. 编译并下载程序
5. PC 端打开串口助手软件，设置串口为 115200，数据位 8，停止位 1

根据提示输入指令操作指纹模块，注意不要勾选“发送新行”

连接树莓派使用

我们提供有树莓派 Python 例程。用户可以在产品百科界面下载。

使用程序前，需要先开启树莓派硬件串口：

输入指令：`sudo raspi-config`

选择 Interfacing Options -> Serial -> No -> Yes

然后重启生效。

UART Fingerprint Sensor (C)	Raspberry Pi
Vcc	3.3V
GND	GND
RX	14 (BCM) – PIN 8 (Board)
TX	15 (BCM) – PIN 10 (Board)
RST	24 (BCM) – PIN 18 (Board)
WAKE	23 (BCM) – PIN 16 (Board)

1. 将指纹模块连接到树莓派
2. 下载程序到树莓派系统：
`wget http://www.waveshare.net/w/upload/9/9d/UART-Fingerprint-RaspberryPi.tar.gz`

3. 解压例程：
`tar zxvf UART-Fignerprint-RaspberryPi.tar.gz`
4. 运行程序：
`cd UART-Fignerprint-RaspberryPi/`
`sudo python main.py`
5. 运行程序之后，请按提示操作指纹模块。