

# 海凌科半导体指纹模组

# 用户手册

文档版本：V1.2.0

发布日期：2020.09.11

型 号：HLK-FPM383C

## 前言及声明

感谢您购买深圳市海凌科电子有限公司（ShenZhen Hi-Link Electronic Co.,Ltd.）的指纹模块。

本手册针对软、硬件应用开发工程师编写，包含硬件接口、系统资源、指令系统、安装信息等内容。为了确保应用开发顺利进行，在进行模块系统集成前请仔细阅读本用户手册。请妥善保管手册，以便遇到问题时快速查阅。

本文件包含深圳市海凌科电子有限公司的私有信息，在未经本公司书面许可的情况下，第三方不得擅自复制或修改；当然，任何在没有授权、特殊条件、限制或告知的情况下对此信息的复制和擅自修改都是侵权行为。

我们已尽最大努力以保证本手册的准确性。然而，如您有任何疑问或者发现错误，可直接与我公司联系，我们将十分感激。

因我公司奉行不断完善改进产品的宗旨，在任何时间，无需告知任何方的情况下，深圳市海凌科电子有限公司有权对本公司产品和服务进行更改、添加、删除、改进以及其他任何变更。如有需要，请您访问我公司的网站 ([www.hlktech.com](http://www.hlktech.com)) 或电话联系，以获取最新信息。

未经本公司书面许可，任何单位及个人不得以任何方式或理由对本公司产品、服务、信息、材料的任何部分进行使用、复制、修改、抄录、传播或与其它产品捆绑使用、销售。

深圳市海凌科电子有限公司对其发行的或与合作公司共同发行的包括但不限于产品或服务的全部内容及其网站 ([www.hlktech.com](http://www.hlktech.com)) 上的材料拥有版权等知识产权，受法律保护。

在对本公司产品的使用中，深圳市海凌科电子有限公司不背负任何责任或者义务；而第三方在使用中则不得侵害任何专利或者其他知识产权。

所有产品的售出都受制于本公司在订购承认书里的销售条款和条件。本公司利用测试、工具、质量控制等技术手段来支持产品的相关性能符合所需规格的一定程度的保证。除了明确的政府书面要求外，没必要执行每款产品的所有参数测试。

**凡侵犯本公司版权等知识产权的，本公司必依法追究其法律责任。**

本公司法律事务部受本公司指示，特此郑重法律声明！

## 联系我们

---

深圳市海凌科电子有限公司

ShenZhen Hi-Link Electronic Co., Ltd.

---

网址: <http://www.hlktech.com>

地址: 深圳市龙华区民治街道民治社区1970科技园1栋三楼

电话: 0755-23152658

## 目 录

前言及声明 .....	2
联系我们 .....	3
目 录 .....	4
1. 概 述 .....	6
1.1. 技术支持 .....	7
1.2. 应用范围 .....	7
2. 主要技术指标 .....	8
3. 硬件接口 .....	9
3.1. 模组对外接口定义 .....	9
3.2. 串行协议 .....	9
3.3. 上电/复位延时 .....	9
4. 通讯协议 .....	10
4.1. 链路层描述 (UART) .....	10
4.2. 应用层描述 .....	11
5. 命令集汇总 .....	13
5.1. 指纹注册 .....	16
5.2. 查询注册结果 .....	16
5.3. 保存指纹模板 .....	18
5.4. 查询指纹保存结果 .....	18
5.5. 取消指纹注册或匹配 .....	19
5.6. 更新存储特征值 (自学习) .....	20
5.7. 查看特征更新结果 .....	20
5.8. 自动注册 .....	21
5.9. 指纹匹配 .....	21
5.10. 查询匹配结果 .....	22
5.11. 指纹匹配 (同步) .....	23
5.12. 指纹特征清除 .....	23
5.13. 查询指纹特征清除结果 .....	25
5.14. 查询指纹 ID 是否存在 .....	25
5.15. 查询指纹特征存储分布 .....	26
5.16. 查询手指在位状态 .....	26
5.17. 指纹特征清除 (同步) .....	27
5.18. 注册指纹确认 .....	28
5.19. 查询注册指纹确认结果 .....	29
5.20. 指纹特征信息下载 .....	29
5.21. 指纹特征数据下载 .....	30
5.22. 指纹特征信息上传 .....	30
5.23. 指纹特征数据上传 .....	31

5.24.	密码设置命令 .....	31
5.25.	复位指纹模块 .....	32
5.26.	获取指纹模板数量 .....	32
5.27.	获取增益 .....	33
5.28.	获取匹配门限 .....	33
5.29.	设置休眠模式 .....	34
5.30.	设置指纹注册拼接次数 .....	35
5.31.	设置 LED 控制信息 .....	35
5.32.	获取系统策略 .....	36
5.33.	设置系统策略 .....	37
5.34.	获取指纹模块 ID .....	37
5.35.	心跳命令 .....	38
5.36.	设置波特率 .....	38
5.37.	设置通信密码 .....	39
6.	附 录 .....	40
6.1.	错误信息定义表 .....	40
6.2.	上电运行流程 .....	41
6.3.	注册流程 .....	42
6.4.	验证流程 .....	43
6.5.	删除流程 .....	44
6.6.	校验和函数代码示例 .....	45
6.7.	复位功能(单路供电模组) .....	45
6.8.	自学习功能 .....	45

# 1. 概述

HLK-FPM383C系列指纹模块是深圳市海凌科电子有限公司推出的稳定量产产品。

HLK-FPM383C系列指纹模块具备完整的指纹处理能力，可以在无需上位机参与的情况下，完成指纹录入，图像处理，指纹比对，指纹特征储存等功能。和同类指纹产品相比，具有以下特色：

- ❖ 模组集成度高、体积更小、功耗更低。
- ❖ 模组使用自主研发指纹识别算法和芯片，各项性能指标优于同类产品。
- ❖ 模组使用的指纹识别算法提供模板拼接功能，提升用户体验。
- ❖ 模组采用指纹特征储存在算法芯片中的方案，可以防止指纹特征被暴力破解，提高产品安全等级。
- ❖ 产品具备防雾防尘防破坏能力，有效解决了人体静电的影响、提高了产品采像质量、增加了产品耐用性。
- ❖ 模组传感器表面可按用户要求定制颜色。
- ❖ 模组具有触摸唤醒功能，并可有效识别塑胶手指、硅胶手指、橡胶手指、指模、指套等假手指。
- ❖ 产品结构简单，模组化设计，提高了产品的稳定性和一致性、便于大规模批量生产。
- ❖ 公开接口代码和命令集，可实现组装式二次开发，降低客户开发难度。
- ❖ 自主知识产权技术可为客户提供高效、灵活的二次开发支持，充分满足客户需求且无知识产权纠纷。

## 1.1. 技术支持

海凌科拥有完备的技术团队，所有员工均来自指纹行业的专业人才，可以对用户开发提供良好的技术支持和售后服务工作。

## 1.2. 应用范围

HLK-FPM383C系列指纹模块应用广泛，适合各类指纹识别系统，例如：

- ❖ 指纹门锁、保险柜、枪盒、金融等安全领域；
- ❖ 门禁系统、工控机、POS 机、驾培、考勤等身份领域；
- ❖ 私人会所、管理软件、授权许可等管理领域；

客户可以根据本手册提供的技术资料，开发出相应的指纹识别应用系统。

## 2. 主要技术指标

---

请参考各型号模组规格书.



## 3. 硬件接口

### 3.1. 模组对外接口定义

请参考各型号模组规格书。

### 3.2. 串行协议

采用半双工异步串口通讯，默认波特率为57600bps。

传送的帧格式为 10 位，一位 0 电平起始位，8 位数据位（低位在前）和一位停止位，无校验位。

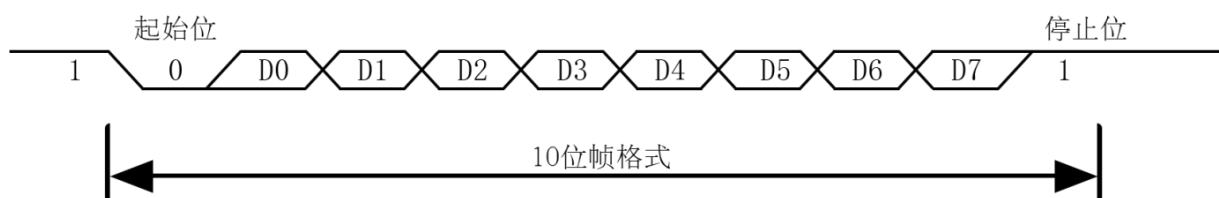


图 1 串行协议视图

### 3.3. 上电/复位延时

模块上电后，约需200ms时间进行初始化工作。在此期间，模块不能响应上位机命令。

## 4. 通讯协议

通讯协议定义了 HLK-FPM383C 系列模板与上位机之间进行数据交换的规则。硬件上采用 UART 接口，软件实现上参考如下通讯协议和指令集。

本协议为2层结构：应用层和链路层。应用层主要负责具体功能定义，链路层用来进行数据传输。在传输协议中定义的2字节或者4字节，采用小字节序（little endian），高位在前低位在后（例如 0x12345678 实际传输方式为 0x12 0x34 0x56 0x78，不是 0x78 0x56 0x34 0x12）。

链路层包括了数据层

应用层
链路层（IIC/UART）

### 4.1. 链路层描述（UART）

UART 链路层为半双工点对点工作方式，所有命令必须由上位机发起，指纹模块进行响应。UART 数据帧格式：

格式	帧头	应用层数据长度	帧头校验和	应用层数据
长度（bytes）	8	2	1	7+N

图 2 UART 数据帧格式表

说明：

帧头：UART 起始头界定，无论是上位机发送数据还是指纹模块发送数据，帧头数据固定为 0xF1 1F E2 2E B6 6B A8 8A；

应用层数据长度：描述应用层实际数据的长度，此数据长度不包含帧头、应用层数据长度、帧头校验和 3 部分；

帧头校验和：为帧头+应用层数据长度的校验和，用来检查数据长度是否有误。

## 4.2. 应用层描述

应用层采用请求/响应的通讯协议，所有命令由上位机发起，指纹模块响应，如下图所示：

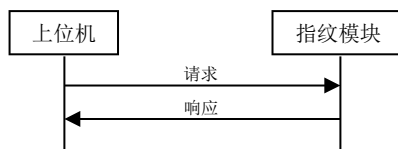


图 3 请求/响应示意图

从上位机开发难易程度考虑，应用层协议使用帧长度固定的方式来进行通讯，使用和校验的方式来保证数据的完整性。在实际应用中，可以通过修改协议的校验密码来确保数据加密，不会被第三方破解。（注意：校验密码设置后会立刻生效，请妥善保管密码，密码遗失后需要返厂初始化。）

按照应用层协议描述，这里分请求命令和响应请求 2 种格式来分别描述上位机数据发送和指纹模块信息响应。

数据发送：

格式	校验密码	命令	数据内容	校验和
长度 (Byte)	4	2	N(0~n)	1

图 4 数据发送表

说明：

发送数据包长度合计：(7+N) Bytes，按照不同命令，数据包长度不同，但是每条指令为固定长度。

校验密码：用来保障数据传输安全性，默认密码为 0x00000000，可以通过应用层命令来更改此密码。

命令：

数据内容：实际发送的内容。

校验和：和校验, 按照 BYTE 操作的方式。（看附录具体校验和说明）

响应请求：

格式	校验密码	响应命令	错误码	数据内容	校验和
长度	4	2	4	N(0~n)	1

图 5 响应请求表

说明：

接收数据包长度合计：(11+N) Bytes，按照不同命令，数据包长度不同，但是每条指令为固定长度。

校验密码：用来保障数据传输安全性，默认密码为 0x00000000，可以通过应用层命令来更改此密码。

响应命令：与上位机发送的命令字段对应，表示收到了响应的命令。

错误码：返回此命令执行的结果，0x00000000 表示无错误，其他参考错误信息定义表。

数据内容：返回响应的结果。

校验和：和校验，按照 BYTE 操作的方式。（看附录具体校验和说明）

## 5. 命令集汇总

主机					指纹模块			
命令类型	子命令	CMD1	CMD2	Data (Len)	ACK1	ACK2	Error Code	Data (Len)
指纹类	<a href="#">指纹注册</a>	0x01	0x11	1	0x01	0x11	正常:0x00000000 错误:返回错误码	0
	<a href="#">查询注册结果</a>	0x01	0x12	0	0x01	0x12	正常:0x00000000 错误:返回错误码	3
	<a href="#">保存指纹</a>	0x01	0x13	2	0x01	0x13	正常:0x00000000 错误:返回错误码	0
	<a href="#">查询保存指纹结果</a>	0x01	0x14	0	0x01	0x14	正常:0x00000000 错误:返回错误码	0
	<a href="#">取消指纹注册</a>	0x01	0x15	0	0x01	0x15	正常:0x00000000 错误:返回错误码	0
	<a href="#">更新存储特征值</a>	0x01	0x16	2	0x01	0x16	正常:0x00000000 错误:返回错误码	0
	<a href="#">查看存储特征更新结果</a>	0x01	0x17	0	0x01	0x17	正常:0x00000000 错误:返回错误码	0
	<a href="#">自动注册</a>	0x01	0x18	4	0x01	0x18	正常:0x00000000 错误:返回错误码	4
	<a href="#">指纹匹配</a>	0x01	0x21	0	0x01	0x21	正常:0x00000000 错误:返回错误码	0
	<a href="#">查询匹配结果</a>	0x01	0x22	0	0x01	0x22	正常:0x00000000 错误:返回错误码	6
	<a href="#">指纹匹配(同步)</a>	0x01	0x23	0	0x01	0x23	正常:0x00000000 错误:返回错误码	6
	<a href="#">指纹特征清除</a>	0x01	0x31	N	0x01	0x31	正常:0x00000000 错误:返回错误码	0

	<a href="#">查询指纹特征清除结果</a>	0x01	0x32	0	0x01	0x32	正常:0x00000000 错误:返回错误码	0
	<a href="#">查询指纹ID 是否存在</a>	0x01	0x33	2	0x01	0x33	正常:0x00000000 错误:返回错误码	3
	<a href="#">查询指纹特征存储分布</a>	0x01	0x34	0	0x01	0x34	正常:0x00000000 错误:返回错误码	66
	<a href="#">查询手指在位状态</a>	0x01	0x35	0	0x01	0x35	正常:0x00000000 错误:返回错误码	1
	<a href="#">指纹特征清除(同步)</a>	0x01	0x36	N	0x01	0x36	正常:0x00000000 错误:返回错误码	0
	<a href="#">注册指纹确认</a>	0x01	0x41	0	0x01	0x41	正常:0x00000000 错误:返回错误码	0
	<a href="#">查询注册指纹确认结果</a>	0x01	0x42	0	0x01	0x42	正常:0x00000000 错误:返回错误码	6
	<a href="#">指纹特征信息下载</a>	0x01	0x51	4	0x01	0x51	正常:0x00000000 错误:返回错误码	0
	<a href="#">指纹特征数据下载</a>	0x01	0x52	N	0x01	0x52	正常:0x00000000 错误:返回错误码	0
	<a href="#">指纹特征信息上传</a>	0x01	0x53	2	0x01	0x53	正常:0x00000000 错误:返回错误码	2
	<a href="#">指纹特征数据上传</a>	0x01	0x54	2	0x01	0x53	正常:0x00000000 错误:返回错误码	N
系统类	<a href="#">密码设置</a>	0x02	0x01	4	0x02	0x01	正常:0x00000000 错误:返回错误码	0
	<a href="#">复位模块</a>	0x02	0x02	0	0x02	0x02	正常:0x00000000 错误:返回错误码	0
	<a href="#">获取指纹模板数量</a>	0x02	0x03	0	0x02	0x03	正常:0x00000000 错误:返回错误码	2
	<a href="#">获取增益</a>	0x02	0x09	0	0x02	0x09	正常:0x00000000 错误:返回错误码	3

	<a href="#">获取匹配门限</a>	0x02	0x0B	0	0x02	0x0B	正常:0x00000000 错误:返回错误码	2
	<a href="#">休眠模式</a>	0x02	0x0C	1	0x02	0x0C	正常:0x00000000 错误:返回错误码	0
	<a href="#">设置最大注册次数</a>	0x02	0x0D	1	0x02	0x0D	正常:0x00000000 错误:返回错误码	0
	<a href="#">设置 LED 控制信息</a>	0x02	0x0F	5	0x02	0x0F	正常:0x00000000 错误:返回错误码	0
	<a href="#">获取系统策略</a>	0x02	0xFB	0	0x02	0xFB	正常:0x00000000 错误:返回错误码	4
	<a href="#">设置系统策略</a>	0x02	0xFC	4	0x02	0xFC	正常:0x00000000 错误:返回错误码	0
维护类	<a href="#">读取 ID</a>	0x03	0x01	0	0x03	0x01	正常:0x00000000 错误:返回错误码	16
	<a href="#">心跳</a>	0x03	0x03	0	0x03	0x03	正常:0x00000000 错误:返回错误码	0
	<a href="#">设置波特率</a>	0x03	0x04	4	0x03	0x04	正常:0x00000000 错误:返回错误码	0
	<a href="#">设置通信密码</a>	0x03	0x05	4	0x03	0x05	正常:0x00000000 错误:返回错误码	0

图 6 命令集汇总表

**命令字说明:**

HLK-FPM383C 系列指纹模块指令集按照使用场景进行了优化, 客户在进行二次开发时仅需简单集成, 就可以支持不同场合和应用的指纹识别功能。

**指纹类:**

主要包含客户在实际使用时的一些功能定义: 指纹注册、匹配、指纹特征删除。当上位机有其他指纹算法时, 可以使用指纹图像获取功能来捕获指纹图像。

以下所有示例的校验密码和校验和, 都以默认校验密码 0x00000000 作为参数, 当使用密码设置功能修改校验密码后, 这里需要使用新的校验密码和校验和。

## 5.1. 指纹注册

发送包格式 (Byte) :

REG\_IDX 表示一次指纹注册时按压的次数, 从 0x01 开始发送, 返回结果仅描述此命令已经被指纹模组接收, 并不表示指纹注册成功。指纹注册是否成功需要通过查询注册结果来判断, 当查询的指纹进度 PROC 大于 100 时, 即表示指纹注册完成。此功能为指纹拼接功能, 与手机指纹注册的应用保持一致, 建议用户在使用的时候需要适当移动手指, 用来获得更好的开锁用户体验。详细流程见后面操作流程描述。建议: REG\_IDX 从 0x01 开始, 增加到 0x06 (最多 6 次), 当在 6 次之后通过查询注册结果发现进度 PROC 还未大于 100 时, 可以继续增加 REG\_IDX 值到适当次数。注意: 在 REG\_IDX 达到最大次数而进度 PROC 还未大于 100 时, 也可以进行指纹模板的保存, 不过此指纹模板不完整, 用户体验不好。0x11 命令 (指纹注册) 应当与 0x12 命令 (查询注册结果) 搭配使用。

校验密码				CMD1	CMD2	REG_IDX	校验和
0x00	0x00	0x00	0x00	0x01	0x11	0x01	0xED

发送包示例:

注册 次数	帧头								Data (Len)		帧头 校验	校验密码				命令		REG _IDX	校验 和
第一次	F1	1F	E2	2E	B6	6B	A8	8A	00	08	85	00	00	00	00	01	11	01	ED
第二次	F1	1F	E2	2E	B6	6B	A8	8A	00	08	85	00	00	00	00	01	11	02	EC
第三次	F1	1F	E2	2E	B6	6B	A8	8A	00	08	85	00	00	00	00	01	11	03	EB
第四次	F1	1F	E2	2E	B6	6B	A8	8A	00	08	85	00	00	00	00	01	11	04	EA
第五次	F1	1F	E2	2E	B6	6B	A8	8A	00	08	85	00	00	00	00	01	11	05	E9
第六次	F1	1F	E2	2E	B6	6B	A8	8A	00	08	85	00	00	00	00	01	11	06	E8

图 7 指纹注册发送包格式表

应答包格式 (Byte) :

校验密码				CMD1	CMD2	完成码				校验和
0x00	0x00	0x00	0x00	0x01	0x11	0x00	0x00	0x00	0x00	0xEE

应答包示例:

成功 (6 次相同) :

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	11	00	00	00	00	EE

失败 (系统忙等错误码) :

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	11	00	00	00	04	EA

...

## 5.2. 查询注册结果

发送包格式 (Byte) :

校验密码				CMD1	CMD2	校验和
0x00	0x00	0x00	0x00	0x01	0x12	0xED



发送包示例：

帧头								Data (Len)		帧头校验	校验密码				命令		校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	01	12	ED

图 8 查询注册结果发送包格式表

应答包格式（Byte）：

注册未完成时返回实际 ID 值,注册进度 PROC 值小于 100:

校验密码（4Byte）				CMD1	CMD2	完成码				ID_H	ID_L	PROC	校验和
0x00	0x00	0x00	0x00	0x01	0x12	0x00	0x00	0x00	0x00	0x00	0x01	0x14	0xD8

应答包示例：

成功（ID1 PROC16）：

帧头								Data (Len)		帧头校验	校验密码				命令		错误码				ID_H	ID_L	PR OC	校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0E	7F	00	00	00	00	01	12	00	00	00	00	00	01	10	DC

成功（ID1 PROC32）：

帧头								Data (Len)		帧头校验	校验密码				命令		错误码				ID_H	ID_L	PR OC	校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0E	7F	00	00	00	00	01	12	00	00	00	00	00	01	20	CC

图 9 查询注册结果注册未完成应答包格式表

...

失败（没有检测到手指按压，超时退出等错误）：

帧头								Data (Len)		帧头校验	校验密码				命令		错误码				ID_H	ID_L	PR OC	校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0E	7F	00	00	00	00	01	12	00	00	00	08	00	00	00	E5

...

注册完成时返回实际 ID 值（ID 为当前未用最小值），注册进度 PROC 值为 100:

校验密码（4Byte）				CMD1	CMD2	完成码				ID_H	ID_L	PROC	校验和
0x00	0x00	0x00	0x00	0x01	0x12	0x00	0x00	0x00	0x00	0x00	0x01	0x64	0x88

应答包示例：

成功（ID1 PROC100）：

帧头								Data (Len)		帧头校验	校验密码				命令		错误码				ID_H	ID_L	PR OC	校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0E	7F	00	00	00	00	01	12	00	00	00	00	00	01	64	88

成功（ID2 PROC100）：

帧头								Data (Len)		帧头校验	校验密码				命令		错误码				ID_H	ID_L	PR OC	校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0E	7F	00	00	00	00	01	12	00	00	00	00	00	02	64	87

图 10 查询注册结果注册完成应答包格式表

获取注册结果错误码			
序号	错误码	说明	建议处理方式
1	0x00000000	正常	本次注册完成，可查看注册进度
2	0x00000003	命令字段非法	检查注册开始命令，重发注册开始。
3	0x00000004	系统繁忙	系统在执行其他命令，请适当延时再查询结果。
4	0x00000005	没有发送该命令的请求，就查询结果	需重发送注册命令，次数重 1 次开始。
5	0x00000006	系统软件上报错误	可能拼接过程中特征数据过大，重新注册本次。
6	0x00000007	硬件错误	可能 SPI 无法通讯，请重试几次，若还是不行请联系我们。
7	0x00000008	没有检测到手指按压，超时退出	目前超时时间为 10S，10s 内没有检测到手指按压，重新注册本次。
8	0x00000009	指纹提取发生错误，可能原因:图像质量不佳等	提示指纹提取发生错误可能图像质量不佳。再重新注册本次
9	0x0000000B	存储空间满	提示存储空间满。删除不需要的指纹，再重新注册
10	0x0000000C	存储写入失败	提示存储写入失败。定位问题，再重新注册
11	0x0000000D	存储读取失败	提示存储读取失败。定位问题，再重新注册

12	0x0000000E	采集的指纹图像质量不佳	提示图像质量不佳。手指可能按压过重、过轻、或者太过潮湿等，再重新注册本次
13	0x0000000F	指纹模板重复	提示指纹模板重复，该枚指纹已录入，再重新注册本次
14	0x00000010	采图面积小	提示采图面积小。手指可能按压过轻或者太过潮湿等，再重新注册本次
15	0x00000011	移动范围过大	提示手指移动范围过大。再重新注册本次
16	0x00000012	移动范围过小	提示手指移动范围过小。再重新注册本次
17	0x00000015	强制退出	请重发开始注册命令
18	0x000000FF	其他错误	通常情况不会出现，若出现可尝试重新注册，多次出现请与我们联系。

图 11 获取注册结果错误码表

### 5.3. 保存指纹模板

发送包格式（Byte）：

保存的 ID 可指定，0x13 命令(保存指纹模板)应当与 0x14 命令（查询指纹保存结果）搭配使用。

校验密码				CMD1	CMD2	ID_H
0x00	0x00	0x00	0x00	0x01	0x13	0x00
ID_L	校验和					
0x01	0xEB					

发送包示例：

帧头								Data (Len)		帧头 校验	校验密码				命令		ID_H	ID_L	校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	09	84	00	00	00	00	01	13	00	01	EB

图 12 保存指纹模板发送包格式表

应答包格式（Byte）：

校验密码				CMD1	CMD2	完成码				校验和
0x00	0x00	0x00	0x00	0x01	0x13	0x00	0x00	0x00	0x00	0xEC

应答包示例：

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	13	00	00	00	00	EC

图 13 保存指纹模板应答包格式表

### 5.4. 查询指纹保存结果

发送包格式（Byte）：

校验密码				CMD1	CMD2	校验和			
0x00	0x00	0x00	0x00	0x01	0x14	0xEB			

发送包示例：

帧头								Data (Len)		帧头 校验	校验密码				命令		校验和			
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	01	14	EB			

图 14 查询指纹保存结果发送包格式表

应答包格式（Byte）：

保存成功返回的ID 为保存后的 ID，返回 0x0F 的错误码 ID 为重复的 ID(重复的 ID 高位在后低位在前)。

校验密码				CMD1	CMD2	完成码				ID_H	ID_L	校验和
0x00	0x00	0x00	0x00	0x01	0x14	0x00	0x00	0x00	0x00	0x00	0x01	0xEA

应答包示例：

成功：

帧头								Data (Len)		帧头校验	校验密码				命令		错误码				ID_H	ID_L	校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0D	80	00	00	00	00	01	14	00	00	00	00	00	01	EA

失败（指纹重复 ID 为重复 ID）：

帧头								Data (Len)		帧头校验	校验密码				命令		错误码				ID_H	ID_L	校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0D	80	00	00	00	00	01	14	00	00	00	0F	00	00	DC

图 15 查询指纹保存结果应答包格式表

获取指纹保存结果错误码			
序号	错误码	说明	建议处理方式
1	0x00000000	正常	保存成功
2	0x00000004	系统繁忙	系统在执行其他命令，请适当延时再查询结果。
3	0x00000005	没有发送该命令的请求，就查询结果	先发送保存指纹命令再查询保存结果，再重新保存，然后查询保存结果
4	0x00000006	系统软件上报错误	提示系统软件上报错误。再重新保存
5	0x00000007	硬件错误	可能保存的ID 大于能存储的最大值。再重新保存
6	0x00000009	指纹提取发生错误，可能原因：图像质量不佳等	提示指纹提取发生错误。重新注册并保存
7	0x0000000C	存储写入失败	提示存储写入失败。定位问题，再重新保存
8	0x0000000D	存储读取失败	提示存储读取失败。定位问题，再重新查询保存结果
9	0x0000000B	存储空间满	提示存储空间满。删除不需要的指纹，再重新保存
10	0x0000000E	采集的指纹图像质量不佳	提示采集的指纹图像质量不佳。重新注册并保存
11	0x0000000F	指纹模板重复	提示指纹模板重复，该枚指纹已录入
12	0x000000FF	其他错误	可能强制停止了注册，重新保存

图 16 获取指纹保存结果错误码表

## 5.5. 取消指纹注册或匹配

发送包格式（Byte）：

校验密码				CMD1	CMD2	校验和
0x00	0x00	0x00	0x00	0x01	0x15	0xEA

发送包示例：

帧头								Data (Len)		帧头校验	校验密码				命令		错误码				校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	01	15	00	00	00	00	EA

图 17 取消指纹注册发送包格式表

应答包格式（Byte）：

校验密码				CMD1	CMD2	完成码				校验和
0x00	0x00	0x00	0x00	0x01	0x15	0x00	0x00	0x00	0x00	0xEA

应答包示例：

帧头								Data (Len)		帧头校验	校验密码				命令		错误码				校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	15	00	00	00	00	EA

图 18 取消指纹注册应答包格式表

## 5.6. 更新存储特征值（自学习）

发送包格式（Byte）：

ID 为需更新的 ID,自学习需配合此命令使用，每次匹配成功后发送此命令更新特征值，0x16 命令(更新存储特征值)应当与 0x17 命令（查查看特征更新结果）搭配使用。

校验密码（4Byte）				CMD1	CMD2	ID_H	ID_L	校验和
0x00	0x00	0x00	0x00	0x01	0x16	0x00	0x00	0xE9

发送包示例：

帧头								Data (Len)		帧头 校验	校验密码				命令		ID _H	ID _L	校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	09	84	00	00	00	00	01	16	00	00	E9

图 19 更新存储特征值发送包格式表

应答包格式（Byte）：

校验密码（4Byte）				CMD1	CMD2	完成码（4Byte）				校验和
0x00	0x00	0x00	0x00	0x01	0x16	0x00	0x00	0x00	0x00	0xE9

应答包示例：

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	16	00	00	00	00	E9

图 20 更新存储特征值应答包格式表

## 5.7. 查看特征更新结果

发送包格式（Byte）：

校验密码				CMD1	CMD2	校验和	
0x00	0x00	0x00	0x00	0x01	0x17	0xE8	

发送包示例：

帧头								Data (Len)		帧头 校验	校验密码				命令		校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	01	17	E8

图 21 查询特征更新结果发送包格式表

应答包格式（Byte）：

校验密码（4Byte）				CMD1	CMD2	完成码（4Byte）				校验和
0x00	0x00	0x00	0x00	0x01	0x17	0x00	0x00	0x00	0x00	0xE8

应答包示例：

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	17	00	00	00	00	E8

图 22 查询特征更新结果应答包格式表

## 5.8. 自动注册

发送包格式 (Byte) :

通过传入按压的总次数和指纹 ID 号就可自动完成采图、提取、拼接、保存等操作；命令返回数据的超时时间为 10s；等待手指标志用于在注册按压后，是否需要坚持等待手指抬起，设置为 1 则表示按压后需要等待手指抬起再次按压才可以进行下一次注册，设置为 0 则不需要检测手指抬起，一直注册到进度 100 为止；按压次数可以设置为 1~6 次；用户可通过传入的 ID 值指定当次注册指纹的 ID 号(可以指定当前系统中已使用的 ID 号并覆盖)，若 ID 为 0xFFFF 则由系统自动分配未使用的指纹 ID。

校验密码				CMD1	CMD2	等待手指	按压次数	ID_H	ID_L	校验和
0x00	0x00	0x00	0x00	0x01	0x18	0x00	0x03	0xFF	0xFF	0xE6

发送包示例:

帧头								Data (Len)		帧头 校验	校验密码				命令		等待 手指	按压 次数	ID_H	ID_L	校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	18	00	03	FF	FF	E6

应答包格式 (Byte) :

注册成功返回指纹 ID。应答包分多次返回，每次采集指纹后返回一次结果，手指检测标志为 1 时会先返回结果再等待手指抬起，保存后返回一次结果。返回包参数中次数指当前采集次数，当次数为 0xFF 时，即保存指纹的应答；当发送包 ID 为 0xFFFF 时，应答包 ID 为最小可以使用的 ID，否则为用户指定的 ID；应答包进度为注册进度，最大值 100。

校验密码				CMD1	CMD2	完成码 (4byte)				次数	ID_H	ID_L	进度	校验和
0x00	0x00	0x00	0x00	0x01	0x18	0x00	0x00	0x00	0x00	0x01	0x00	0x00	0x21	0xC5

应答包示例:

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				次 数	ID_H	ID_L	进 度	校 验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0F	7E	00	00	00	00	01	18	00	00	00	00	01	00	00	21	C5
F1	1F	E2	2E	B6	6B	A8	8A	00	0F	7E	00	00	00	00	01	18	00	00	00	00	02	00	00	42	A3
F1	1F	E2	2E	B6	6B	A8	8A	00	0F	7E	00	00	00	00	01	18	00	00	00	00	03	00	00	64	80
F1	1F	E2	2E	B6	6B	A8	8A	00	0F	7E	00	00	00	00	01	18	00	00	00	00	FF	00	00	64	84

## 5.9. 指纹匹配

发送包格式 (Byte) :

0x21 命令(指纹匹配)应当与 0x22 命令 (查询匹配结果) 搭配使用:

校验密码				CMD1	CMD2	校验和
0x00	0x00	0x00	0x00	0x01	0x21	0xDE

发送包示例:

帧头								Data (Len)		帧头 校验	校验密码				命令		校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	01	21	DE

图 23 指纹匹配发送包格式表

应答包格式 (Byte) :

校验密码				CMD1	CMD2	完成码				校验和
0x00	0x00	0x00	0x00	0x01	0x21	0x00	0x00	0x00	0x00	0xDE

应答包示例：

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	21	00	00	00	00	DE

图 24 指纹匹配应答包格式表

## 5.10. 查询匹配结果

发送包格式 (Byte)：

校验密码				CMD1	CMD2	校验和
0x00	0x00	0x00	0x00	0x01	0x22	0xDD

发送包示例：

帧头								Data (Len)		帧头 校验	校验密码				命令		校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	01	22	DD

图 25 查询匹配结果发送包格式表

应答包格式 (Byte)：

包含匹配结果、匹配分数、匹配的指纹 id 号三项内容：

校验密码 (4Byte)				CMD1	CMD2	完成码 (4Byte)			
0x00	0x00	0x00	0x00	0x01	0x22	0x00	0x00	0x00	0x00
匹配结果		匹配分数		匹配指纹 id 号		校验和			
0x00	0x01	0x27	0x0F	0x00	0x03	0xA3			

应答包示例：

成功：

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				匹配 结果		匹配 分数		匹配 id 号		校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	11	7C	00	00	00	00	01	22	00	00	00	00	00	01	27	0F	00	03	A3

失败 (指纹匹配发生错误, 可能原因: 图像质量不佳等)：

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				匹配 结果		匹配 分数		匹配 id 号		校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	11	7C	00	00	00	00	01	22	00	00	00	0A	00	00	00	00	00	00	D3

失败 (系统忙等错误情况)：

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	22	00	00	00	04	D9

图 26 查询匹配结果应答包格式表

...

查询匹配结果的错误码			
序号	错误码	说明	建议处理方式
1	0x00000000	正常	匹配成功, 可查看匹配 ID 和匹配分数
2	0x00000004	系统繁忙	系统在执行其他命令, 请适当延时再查询结果。
3	0x00000005	没有发送该命令的请求, 就查询结果	先发送匹配命令再查询匹配结果
4	0x00000006	系统软件上报错误	提示系统软件上报错误。再重新匹配
5	0x00000007	硬件错误	提示硬件错误。再重新匹配
6	0x00000008	没有检测到手指按压, 超时退出	提示用户没有按压手指。再重新匹配
7	0x00000009	指纹提取发生错误, 可能原因: 图像质量不佳等	提示指纹提取错误。可能图像质量不佳, 再重新匹配
8	0x0000000A	指纹模板库为空	请注册指纹再进行匹配
9	0x0000000E	采集的指纹图像质量不佳	提示采集的指纹图像质量不佳。可能手指按压过重、过轻、或者太过潮湿等, 重新匹配

10	0x00000010	采图面积小	提示采图面积小。手指按压过轻或按压方式不对，重新匹配
11	0x000000FF	其他错误	重新匹配

图 27 查询匹配结果错误码表

## 5.11. 指纹匹配（同步）

发送包格式（Byte）：

校验密码				CMD1	CMD2	校验和
0x00	0x00	0x00	0x00	0x01	0x23	0xDC

发送包示例：

帧头								Data (Len)		帧头 校验	校验密码				命令		校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	01	23	DC

图 28 查询匹配结果发送包格式表

应答包格式（Byte）：

校验密码（4Byte）				CMD1	CMD2	完成码（4Byte）			
0x00	0x00	0x00	0x00	0x01	0x23	0x00	0x00	0x00	0x00
匹配结果		匹配分数		匹配指纹 id 号		校验和			
0x00	0x01	0x27	0x0F	0x00	0x03	0xA2			

应答包示例：

成功：

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				匹配 结果		匹配 分数		匹配 id 号		校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	11	7C	00	00	00	00	01	23	00	00	00	00	00	01	27	0F	00	03	A2

失败（指纹匹配发生错误，可能原因：图像质量不佳等）：

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				匹配 结果		匹配 分数		匹配 id 号		校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	11	7C	00	00	00	00	01	23	00	00	00	0A	00	00	00	00	00	00	D2

## 5.12. 指纹特征清除

发送包格式（Byte）：

CL\_FLAG 为清除指纹特征的模式标志，0x01 表示清除所有指纹，0x02 表示清除多条指纹，0x03 表示清除一块指纹，默认 0x00 表示清除单个指纹。ID\_H、ID\_L 为将要删除的指纹 ID（ID 值从 0 开始分配），0x31 命令(指纹特征清除)应当与 0x32 命令（查询指纹特征清除结果）搭配使用：

清除全部或单条指纹特征发送包格式：

校验密码（4Byte）				CMD1	CMD2	CL_FLAG	ID_H	ID_L	校验和
0x00	0x00	0x00	0x00	0x01	0x31	0x00	0x00	0x01	0xCD

发送包示例：

指纹特征清除（全部）：

帧头								Data (Len)		帧头 校验	校验密码				命令		CL_ FLAG	ID _H	ID _L	校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0A	83	00	00	00	00	01	31	01	00	01	CC

指纹特征清除（单条）：

帧头								Data (Len)		帧头校验	校验密码				命令		CL_FLAG	ID_H	ID_L	校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0A	83	00	00	00	00	01	31	00	00	01	CD

清除多条指纹特征发送包格式：

COUNT 为待删除指纹数量，IDn 为待删除指纹特征 ID 值（ID 值从 0 开始分配）。

校验密码（4Byte）				CMD1	CMD2	CL_FLAG	COUNT_H	COUNT_L
0x00	0x00	0x00	0x00	0x01	0x31	0x02	0xXX	0xXX
ID1_H	ID1_L	ID2_H	ID2_L	... ..		IDn_H	IDn_L	校验和
0x00	0x01	0x00	0x02	... ..		0x00	0xXX	0xXX

发送包示例：

指纹特征清除（多条）：

帧头								Data (Len)		帧头校验	校验密码				命令		CL_FLAG	COUNT_H	COUNT_L
F1	1F	E2	2E	B6	6B	A8	8A	00	0E	7F	00	00	00	00	01	31	02	00	02
ID1_H	ID1_L	ID2_H	ID2_L	校验和															
00	01	00	02	C7															

块清除命令 CL\_FLAG 为 0x03, 传入要删除的首 ID 和尾 ID 即可。

校验密码（4Byte）				CMD1	CMD2	CL_FLAG	FIRST_ID_H	FIRST_ID_L
0x00	0x00	0x00	0x00	0x01	0x31	0x03	00	02
LAST_ID_H	LAST_ID_L	校验和						
0x00	0x06	0xC3						

发送包示例：

指纹特征清除（块删除）：

帧头								Data (Len)		帧头校验	校验密码				命令		CL_FLAG	FIRST_ID_H	FIRST_ID_L
F1	1F	E2	2E	B6	6B	A8	8A	00	0C	81	00	00	00	00	01	31	03	00	02
LAST_ID_H	LAST_ID_L	校验和																	
00	06	C3																	

图 28 指纹特征清除发送包格式表

应答包格式（Byte）：

校验密码（4Byte）				CMD1	CMD2	完成码（4Byte）				校验和
0x00	0x00	0x00	0x00	0x01	0x31	0x00	0x00	0x00	0x00	0xCE

应答包示例：

帧头								Data (Len)		帧头校验	校验密码				命令		错误码				校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	31	00	00	00	00	CE

图 29 指纹特征清除应答包格式表



### 5.13. 查询指纹特征清除结果

发送包格式 (Byte), 无数据域:

校验密码 (4Byte)				CMD1	CMD2	校验和
0x00	0x00	0x00	0x00	0x01	0x32	0xCD

发送包示例:

帧头								Data (Len)		帧头 校验	校验密码				命令		校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	01	32	CD

图 30 查询指纹特征清除结果发送包格式表

应答包格式 (Byte):

校验密码 (4Byte)				CMD1	CMD2	完成码 (4Byte)				校验和
0x00	0x00	0x00	0x00	0x01	0x32	0x00	0x00	0x00	0x00	0xCD

应答包示例:

帧头								Data (Len)		帧头 校验		校验密码				命令		错误码				校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	32	00	00	00	00	CD	

图 31 查询指纹特征清除结果应答包格式表

获取删除结果错误码			
序号	错误码	说明	建议处理方式
1	0x00000000	正常	删除成功
2	0x00000001	无法识别的命令	提示无法识别的命令。查看命令发送是否正确, 再次重发
3	0x00000002	命令字段长度非法	提示命令字段长度非法。查看命令字段长度是否正确, 再次重发
4	0x00000003	命令字段非法	提示命令字段非法。查看命令字段是否正确, 再次重发
5	0x00000004	系统繁忙	系统在执行其他命令, 请适当延时再查询结果。

图 32 获取删除结果错误码表

### 5.14. 查询指纹 ID 是否存在

发送包格式 (Byte):

ID\_H、ID\_L 为将要查询的指纹 ID:

校验密码 (4Byte)				CMD1	CMD2	ID_H	ID_L	校验和
0x00	0x00	0x00	0x00	0x01	0x33	0x00	0x01	0xCB

发送包示例:

帧头								Data (Len)		帧头 校验	校验密码				命令		ID _H	ID _L	校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	09	84	00	00	00	00	01	33	00	01	CB

图 33 查询指纹 ID 是否存在发送包格式表

应答包格式 (Byte):

获取成功完成码为 0x00000000, 其他为查询失败。如果查询的指纹存在, 则返回 STATE 为 1, 否则 STATE 为 0; 数据中返回 ID\_H、ID\_L 为本次操作查询的指纹 ID:

校验密码 (4Byte)				CMD1	CMD2	完成码 (4Byte)				STATE	ID_H	ID_L	校验
0x00	0x00	0x00	0x00	0x01	0x33	0x00	0x00	0x00	0x00	0x01	0x00	0x01	0xCA

应答包示例:

存在:

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				STA TE	ID H	ID L	校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0E	7F	00	00	00	00	01	33	00	00	00	00	01	00	01	CA

不存在:

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				STA TE	ID_ H	ID_ L	校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0E	7F	00	00	00	00	01	33	00	00	00	00	00	00	01	CB

图 34 查询指纹 ID 是否存在应答包格式表

## 5.15. 查询指纹特征存储分布

发送包格式 (Byte):

校验密码 (4Byte)				CMD1	CMD2	校验和
0x00	0x00	0x00	0x00	0x01	0x34	0xCB

发送包示例:

帧头								Data (Len)		帧头 校验	校验密码				命令		校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	01	34	CB

图 35 查询指纹特征存储分布发送包格式表

应答包格式 (Byte):

获取成功完成码为 0x00000000, 其他为查询失败。NUM\_H、NUM\_L 为指纹储存总数, INFO\_0 ~ INFO\_63 为指纹储存状态。第 0~7 枚指纹状态对应于 INFO\_0 的 0~7bit, 第 8~15 枚指纹状态对应于 INFO\_1 的 0~7bit, 如此类推。指纹存在, 则对应的 bit 置为 1, 否则为 0。

校验密码 (4Byte)				CMD1	CMD2	完成码 (4Byte)				NUM_H	NUM_L
0x00	0x00	0x00	0x00	0x01	0x34	0x00	0x00	0x00	0x00	0x20	0x00
INFO_0	INFO_1	INFO_2	...							INFO_63	校验和
0xFF	0x00	0x00	...							0x00	checksum

应答包示例:

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码			
F1	1F	E2	2E	B6	6B	A8	8A	00	4D	40	00	00	00	00	01	34	00	00	00	00
NUM_H	NUM_L	INFO_0	INFO_1	INFO_2	...												INFO_63	校验和		
20	00	01	00	00	...												00	C8		

图 36 查询指纹特征存储分布应答包格式表

## 5.16. 查询手指在位状态

发送包格式 (Byte):

校验密码 (4Byte)				CMD1	CMD2	校验和
0x00	0x00	0x00	0x00	0x01	0x35	0xCA

发送包示例：

帧头								Data (Len)		帧头 校验	校验密码				命令		校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	01	35	CA

图 37 查询手指在位状态发送包格式表

应答包格式（Byte）：

获取成功完成码为 0x00000000，其他为查询失败。如果手指在位，则返回 STATE 为 1，否则 STATE 为 0：

校验密码（4Byte）				CMD1	CMD2	完成码（4Byte）				STATE	校验
0x00	0x00	0x00	0x00	0x01	0x35	0x00	0x00	0x00	0x00	0x01	0xC9

应答包示例：

在位：

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				STAT E	校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0C	81	00	00	00	00	01	35	00	00	00	00	01	C9

不在位：

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				STA TE	校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0C	81	00	00	00	00	01	35	00	00	00	00	00	CA

图 38 查询手指在位状态应答包格式表

## 5.17. 指纹特征清除（同步）

发送包格式（Byte）：

CL\_FLAG 为清除指纹特征的模式标志，0x01 表示清除所有指纹，0x02 表示清除多条指纹，0x03 表示清除一块指纹，默认 0x00 表示清除单个指纹。ID\_H、ID\_L 为将要删除的指纹 ID（ID 值从 0 开始分配）：

清除全部或单条指纹特征发送包格式：

校验密码（4Byte）				CMD1	CMD2	CL_FLAG	ID_H	ID_L	校验和
0x00	0x00	0x00	0x00	0x01	0x36	0x00	0x00	0x01	0xC8

发送包示例：

指纹特征清除（全部）：

帧头								Data (Len)		帧头 校验	校验密码				命令		CL_ FLAG	ID _H	ID _L	校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0A	83	00	00	00	00	01	36	01	00	01	C7

指纹特征清除（单条）：

帧头								Data (Len)		帧头 校验	校验密码				命令		CL_ FLAG	ID _H	ID _L	校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0A	83	00	00	00	00	01	36	00	00	01	C8

清除多条指纹特征发送包格式：

COUNT 为待删除指纹数量，IDn 为待删除指纹特征 ID 值（ID 值从 0 开始分配）。

校验密码（4Byte）				CMD1	CMD2	CL_FLAG	COUNT_H	COUNT_L
0x00	0x00	0x00	0x00	0x01	0x36	0x02	0xXX	0xXX
ID1_H	ID1_L	ID2_H	ID2_L	... ..		IDn_H	IDn_L	校验和
0x00	0x01	0x00	0x02	... ..		0x00	0xXX	0xXX

发送包示例：  
指纹特征清除（多条）：

帧头								Data (Len)		帧头 校验	校验密码				命令		CL_ FLAG	COUNT _H	COUNT _L
F1	1F	E2	2E	B6	6B	A8	8A	00	0E	7F	00	00	00	00	01	36	02	00	02
ID 1_H	ID 1_L	ID 2_H	ID 2_L	校验 和															
00	01	00	02	C2															

块清除命令 CL\_FLAG 为 0x03, 传入要删除的首 ID 和尾 ID 即可。

校验密码 (4Byte)				CMD1	CMD2	CL_FLAG	FIRST_I D_H	FIRST_I D_L
0x00	0x00	0x00	0x00	0x01	0x36	0x03	0x00	0x02
LAST_I D_H	LAST_I D_L	校验和						
0x00	0x06	0xBE						

发送包示例：  
指纹特征清除（块删除）：

帧头								Data (Len)		帧头 校验	校验密码				命令		CL_ FLAG	FIRST_ ID_H	FIRST_ID _L
F1	1F	E2	2E	B6	6B	A8	8A	00	0C	81	00	00	00	00	01	36	03	00	02
LAST_ID _H	LAST_ID _L	校验 和																	
00	06	BE																	

图 39 指纹特征清除发送包格式表

应答包格式（Byte）：

校验密码 (4Byte)				CMD1	CMD2	完成码 (4Byte)				校验和
0x00	0x00	0x00	0x00	0x01	0x36	0x00	0x00	0x00	0x00	0xC9

应答包示例：

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	36	00	00	00	00	C9

图 40 查询指纹特征清除结果应答包格式表

## 5.18. 注册指纹确认

此命令用于在注册指纹成功之后，要求用户作一次确认录入成功的判断，根据行业应用客户需求使用。

发送包格式（Byte）：

0x41 命令(注册指纹确认)应当与 0x42 命令（查询注册指纹确认结果）搭配使用：

校验密码 (4Byte)				CMD1	CMD2	校验和
0x00	0x00	0x00	0x00	0x01	0x41	0xBE

发送包示例：

帧头								Data (Len)		帧头 校验	校验密码				命令		校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	01	41	BE

图 41 注册指纹确认发送包格式表

应答包格式 (Byte) :

校验密码 (4Byte)				CMD1	CMD2	完成码 (4Byte)				校验和
0x00	0x00	0x00	0x00	0x01	0x41	0x00	0x00	0x00	0x00	0xBE

应答包示例:

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	41	00	00	00	00	BE

图 42 注册指纹确认应答包格式表

## 5.19. 查询注册指纹确认结果

发送包格式 (Byte) :

校验密码				CMD1	CMD2	校验和
0x00	0x00	0x00	0x00	0x01	0x42	0xBD

发送包示例:

帧头								Data (Len)		帧头 校验	校验密码				命令		校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	01	42	BD

图 43 查询注册指纹确认结果发送包格式表

应答包格式 (Byte) :

包含匹配结果, 匹配分数, 匹配 id 号 (预留) 三项内容:

校验密码 (4Byte)				CMD1	CMD2	完成码 (4Byte)			
0x00	0x00	0x00	0x00	0x01	0x42	0x00	0x00	0x00	0x00
匹配结果		匹配分数		id (预留)		校验和			
0x00	0x01	0x27	0x0F	0x00	0x00	0x86			

应答包示例:

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码			
F1	1F	E2	2E	B6	6B	A8	8A	00	11	7C	00	00	00	00	01	42	00	00	00	00
匹配 结果		匹配 分数		ID (预留)		校验和														
00	01	27	0F	00	00	86														

图 44 查询注册指纹确认结果应答包格式表

## 5.20. 指纹特征信息下载

发送包格式 (Byte) :

ID\_H、ID\_L 为将要下载的指纹 ID (ID 值从 0 开始分配), INFO 为要下载的总数据的大小 (最大 8192Byte)。

校验密码 (4Byte)				CMD1	CMD2	ID_H	ID_L	INFO_H	INFO_L	校验和
0x00	0x00	0x00	0x00	0x01	0x51	0x00	0x00	0x07	0xEC	0xBB

发送包示例:

帧头								Data (Len)		帧头 校验	校验密码				命令		ID _H	ID _L	INF O_H	INF O_L	校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	51	00	00	07	EC	BB

图 45 指纹特征信息下载发送包格式表

应答包格式 (Byte) :

校验密码 (4Byte)				CMD1	CMD2	完成码 (4Byte)				校验和
0x00	0x00	0x00	0x00	0x01	0x51	0x00	0x00	0x00	0x00	0xAE

应答包示例:

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	51	00	00	00	00	AE

图 46 查询注册指纹确认结果应答包格式表

## 5.21. 指纹特征数据下载

发送包格式 (Byte) :

FRAME 为当前发送的第几帧(第一帧从 0 开始), DATA 的数据个数不固定, 由用户发下去的数据长度决定 (len-9,最大不超过 128byte)。下载完 INFO 的所有数据。

校验密码				CMD1	CMD2	FRAME_H	FRAME_L	DATA1	...	DATAn	校验和
0x00	0x00	0x00	0x00	0x01	0x52	0xXX	0xXX	0xXX	...	0xXX	0xXX

发送包示例:

帧头								Data (Len)		帧头 校验	校验密码				命令		FRAME _H	FRAME _L	DATA1	...	DATAn	校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	89	04	00	00	00	00	01	52	0xXX	0xXX	0xXX	...	0xXX	0xXX

图 47 指纹特征数据下载发送包格式表

应答包格式 (Byte)

校验密码 (4Byte)				CMD1	CMD2	完成码 (4Byte)				校验和
0x00	0x00	0x00	0x00	0x01	0x52	0x00	0x00	0x00	0x00	0xAD

应答包示例:

帧头								Data (Len)		帧头校验	校验密码				命令		错误码				校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	01	52	00	00	00	00	AD

图 48 指纹特征数据下载应答包格式表

## 5.22. 指纹特征信息上传

发送包格式 (Byte) :

ID\_H、ID\_L 为将要上传的指纹 ID:

校验密码 (4Byte)				CMD1	CMD2	ID_H	ID_L	校验和
0x00	0x00	0x00	0x00	0x01	0x53	0x00	0x00	0xAC

发送包示例:

帧头								Data (Len)		帧头 校验	校验密码				命令		ID _H	ID _L	校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	09	84	00	00	00	00	01	53	00	00	AC

图 49 指纹特征信息上传发送包格式表

应答包格式 (Byte) :

INFO 上传的数据大小:

校验密码 (4Byte)				CMD1	CMD2	完成码 (4Byte)				INFO_H	INFO_L	校验和
0x00	0x00	0x00	0x00	0x01	0x53	0x00	0x00	0x00	0x00	0x07	0xEC	0xB9

应答包示例:

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				INFO _H	INFO _L	校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0D	80	00	00	00	00	01	53	00	00	00	00	07	EC	B9

图 50 指纹特征信息上传应答包格式表

## 5.23. 指纹特征数据上传

发送包格式 (Byte) :

NUM\_H、NUM\_L 为上传次数或帧数(第一帧从 0 开始):

校验密码 (4Byte)				CMD1	CMD2	NUM_H	NUM_L	校验和
0x00	0x00	0x00	0x00	0x01	0x54	0x00	0x00	0xAB

发送包示例:

帧头								Data (Len)		帧头 校验	校验密码				命令		NUM _H	NUM _L	校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	09	84	00	00	00	00	01	54	00	00	AB

图 51 指纹特征数据上传发送包格式表

应答包格式 (Byte) :

上位机应当记录并累加每帧特征数据的长度, 截取 0x53 命令所获取的 INFO 上传长度的数据:

校验密码 (4Byte)				CMD1	CMD2	完成码 (4Byte)				NUM_H	NUM_L	DATA1	...	DATA 128	校验和
0x00	0x00	0x00	0x00	0x01	0x54	0x00	0x00	0x00	0x00	0x00	0x00	0xXX	...	0xXX	0xXX

应答包示例:

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				NUM_H	NUM_L	DATA1	...	DATA 128	校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	8D	00	00	00	00	00	01	54	00	00	00	00	00	00	XX	...	XX	XX

图 52 指纹特征数据上传应答包格式表

## 5.24. 密码设置命令

发送包格式 (Byte) :

如设置密码为 0x12345678, 则 PSW0~PSW3 分别为 0x12,0x34,0x56,0x78。不写 flash, 掉电恢复。

校验密码				CMD1	CMD2	PSW0
0x00	0x00	0x00	0x00	0x02	0x01	0x12
PSW1	PSW2	PSW3	校验和			
0x34	0x56	0x78	0xE9			

发送包示例:

帧头								Data (Len)		帧头校验	校验密码				命令		PSW 0	PS W1	PS W2	PS W3	校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	02	01	12	34	56	78	E9

图 53 密码设置命令发送包格式表

应答包格式 (Byte) :

校验密码				CMD1	CMD2	完成码				校验和
0x12	0x34	0x56	0x78	0x02	0x01	0x00	0x00	0x00	0x00	0xE9

应答包示例:

帧头								Data (Len)		帧头校验	校验密码				命令		错误码				校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	12	34	56	78	02	01	00	00	00	00	E9

图 54 密码设置命令应答包格式表

密码设置成功。如果再发获取指纹模块 ID 命令, 格式如下。

校验密码				CMD1	CMD2	校验和
0x12	0x34	0x56	0x78	0x03	0x01	0xE8

发送包示例:

帧头								Data (Len)		帧头校验	校验密码				命令		校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	12	34	56	78	03	01	E8

图 55 密码设置成功校验包格式表

## 5.25. 复位指纹模块

发送包格式 (Byte) :

校验密码				CMD1	CMD2	校验和
0x00	0x00	0x00	0x00	0x02	0x02	0xFC

发送包示例:

帧头								Data (Len)		帧头校验	校验密码				命令		校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	02	02	FC

图 49 复位指纹模块发送包格式表

应答包格式 (Byte) :

校验密码				CMD1	CMD2	完成码				校验和
0x00	0x00	0x00	0x00	0x02	0x02	0x00	0x00	0x00	0x00	0xFC

应答包示例:

帧头								Data (Len)		帧头校验	校验密码				命令		错误码				校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	02	02	00	00	00	00	FC

图 56 复位指纹模块应答包格式表

## 5.26. 获取指纹模板数量

发送包格式 (Byte) :

校验密码				CMD1	CMD2	校验和
------	--	--	--	------	------	-----



0x00	0x00	0x00	0x00	0x02	0x03	0xFB
------	------	------	------	------	------	------

发送包示例：

帧头								Data (Len)		帧头校验	校验密码				命令		校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	02	03	FB

图 57 获取指纹模板数量发送包格式表

应答包格式 (Byte) :

返回的是已注册模组数量：

校验密码 (4Byte)				CMD1	CMD2	完成码 (4Byte)				NUM_H	NUM_L	校验和
0x00	0x00	0x00	0x00	0x02	0x03	0x00	0x00	0x00	0x00	0x00	0x04	0xF7

应答包示例：

帧头								Data (Len)		帧头校验	校验密码				命令		错误码				NUM_H	NUM_L	校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0D	80	00	00	00	00	02	03	00	00	00	00	00	04	F7

图 58 获取指纹模板数量应答包格式表

## 5.27. 获取增益

发送包格式 (Byte) :

校验密码				CMD1	CMD2	校验和
0x00	0x00	0x00	0x00	0x02	0x09	0xF5

发送包示例：

帧头								Data (Len)		帧头校验	校验密码				命令		校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	02	09	F5

图 59 获取增益发送包格式表

应答包格式 (Byte) :

校验密码				CMD1	CMD2	完成码			
0x00	0x00	0x00	0x00	0x02	0x09	0x00	0x00	0x00	0x00
Shift	Gain	PxlCtrl	校验和						
0x1F	0x00	0x04	0xD2						

应答包示例：

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码			
F1	1F	E2	2E	B6	6B	A8	8A	00	0E	7F	00	00	00	00	02	09	00	00	00	00
Shift		Gain		Px1 Ctrl		校验 和														
1F		00		04		D2														

图 60 获取增益应答包格式表

## 5.28. 获取匹配门限

发送包格式 (Byte) :

校验密码				CMD1	CMD2	校验和
0x00	0x00	0x00	0x00	0x02	0x0B	0xF3

发送包示例:

帧头								Data (Len)		帧头 校验	校验密码				命令		校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	02	0B	F3

图 61 获取匹配门限发送包格式表

应答包格式 (Byte) :

校验密码				CMD1	CMD2	完成码			
0x00	0x00	0x00	0x00	0x02	0x0B	0x00	0x00	0x00	0x00
Threshold_H		Threshold_L		校验和					
0x21		0x34		0x9E					

应答包示例:

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				Thresh old_H	Thresh old_L	校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0D	80	00	00	00	00	02	0B	00	00	00	00	21	34	9E

图 62 获取匹配门限应答包格式表

## 5.29. 设置休眠模式

发送包格式 (Byte) :

MODE 字段为 1Byte,00 表示进入普通休眠模式,01 表示进入深度休眠模式

校验密码				CMD1	CMD2	MODE
0x00	0x00	0x00	0x00	0x02	0x0C	0x00
校验和						
0xF2						

发送包示例:

普通休眠:

帧头								Data (Len)		帧头 校验	校验密码				命令		MODE	校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	08	85	00	00	00	00	02	0C	00	F2

深度休眠:

帧头								Data (Len)		帧头 校验	校验密码				命令		MODE	校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	08	85	00	00	00	00	02	0C	01	F1

图 63 设置休眠模式发送包格式表

应答包格式 (Byte) :

校验密码				CMD1	CMD2	完成码				校验和
0x00	0x00	0x00	0x00	0x02	0x0C	0x00	0x00	0x00	0x00	0xF2

应答包示例:

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	02	0C	00	00	00	00	F2

图 64 设置休眠模式应答包格式表

注：当模块进入普通休眠模式时，可以通过按压指纹 sensor 唤醒模块，串口无响应；唤醒后串口可以访问。

当模块进入深度休眠模式时，需要外部输入复位信号来唤醒模块，串口无响应。

### 5.30. 设置指纹注册拼接次数

发送包格式（Byte）：

ENROLL\_NUM 设置范围（1~6）：

校验密码				CMD1	CMD2	ENROLL_NUM
0x00	0x00	0x00	0x00	0x02	0x0D	0x06
校验和						
0xEB						

发送包示例：

帧头								Data (Len)		帧头 校验	校验密码				命令		ENROLL _NUM	校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	08	85	00	00	00	00	02	0D	06	EB

图 65 设置指纹注册拼接发送包格式表

应答包格式（Byte）：

校验密码				CMD1	CMD2	完成码				校验和
0x00	0x00	0x00	0x00	0x02	0x0D	0x00	0x00	0x00	0x00	0xF1

应答包示例：

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	02	0D	00	00	00	00	F1

图 66 设置指纹注册拼接应答包格式表

### 5.31. 设置 LED 控制信息

发送包格式（Byte）： ML-FPM026 无蓝色灯

校验密码				CMD1	CMD2	控制模式
0x00	0x00	0x00	0x00	0x02	0x0F	0x03
灯光颜色	参数 1	参数 2	参数 3	校验和		
0x03	0x64	0x00	0x32	0x53		

参数说明：

参数项	控制模式	灯光颜色		参数 1	参数 2	参数 3
	0: 关闭LED 灯 1: 开启LED 灯	0: 无颜色控制 1: 绿色 2: 红色	PWM 模式	最大占空比 (范围: 0~100)	最小占空比 (范围: 0~100)	占空比每秒变化速率 (单位: 1%/s, 范围: 0~100)
参数说明	2: 当手指触碰时自动点亮LED 灯	3: 红色+绿色				
	3: PWM 控制 LED 灯(呼吸灯) 4: 闪烁LED 灯	4: 蓝色 5: 红色+蓝色 6: 绿色+蓝色	闪烁 模式	LED 点亮时长 (单位: 10ms)	LED 熄灭时长 (单位: 10ms)	闪烁周期数量

		7:红色+蓝色+绿色				
--	--	------------	--	--	--	--

发送包示例:

关闭 LED 灯:

帧头								Data (Len)		帧头校验	校验密码				命令		控制模式	灯光颜色	参数1	参数2	参数3	校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0C	81	00	00	00	00	02	0F	00	00	00	00	00	EF

设置成呼吸灯模式(红绿灯同时点亮, 最大占空比 100, 最小占空比 0, 占空比每秒变化 50%):

帧头								Data (Len)		帧头校验	校验密码				命令		控制模式	灯光颜色	参数1	参数2	参数3	校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0C	81	00	00	00	00	02	0F	03	03	64	00	32	53

设置成闪烁模式(绿灯闪烁, 点亮 200ms, 熄灭 200ms, 循环 5 次):

帧头								Data (Len)		帧头校验	校验密码				命令		控制模式	灯光颜色	参数1	参数2	参数3	校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0C	81	00	00	00	00	02	0F	04	01	14	14	05	BD

图 67 设置 LED 灯控制信息发送包格式表

应答包格式 (Byte):

校验密码				CMD1	CMD2	完成码				校验和
0x00	0x00	0x00	0x00	0x02	0x0F	0x00	0x00	0x00	0x00	0xEF

应答包示例:

帧头								Data (Len)		帧头 校验	校验密 码				命令						错误 码			校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	02	0F		00	00	00	00	00	EF	

图 68 设置 LED 控制信息应答包格式表

## 5.32. 获取系统策略

发送包格式 (4Byte):

校验密码				CMD1	CMD2	完成码				校验和
0x00	0x00	0x00	0x00	0x02	0xFB	0x00	0x00	0x00	0x00	0x03

发送包示例:

帧头								Data (Len)		帧头校验	校验密码				命令		校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	02	FB	03

图 69 获取系统策略发送格式表格

应答包格式 (Byte):

Bit0:内部使用。Bit1:重复指纹检查。Bit2: 自学习功能。Bit3:内部使用。Bit4: 360 度识别。Bit5: 内部使用。其他预留。

校验密码				CMD1	CMD2	Policy0
0x00	0x00	0x00	0x00	0x02	0xFB	0x00
Policy1	Policy2	Policy3	校验和			
0x00	0x00	0x16	0xEC			

应答包示例: 策略 0x00000016 为重复指纹检测、自学习、360 度

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				Poli cy0	Poli cy1	Poli cy2	Poli cy3	校验 和								
K-FHIV303C								F1	1F	E2	2E	B6	6B	A8	8A	00	0F	7E	00	00	00	00	02	FB	00	00	00	00	00	00	00	16	ED

图 70 获取系统策略应答格式表格

### 5.33. 设置系统策略

发送包格式（4Byte）：

Bit0:内部使用。Bit1:重复指纹检查。Bit2: 自学习功能。Bit3:内部使用。Bit4: 360 度识别。  
Bit5: 内部使用。其他预留。

校验密码				CMD1	CMD2	Policy0
0x00	0x00	0x00	0x00	0x02	0xFC	0x00
Policy1	Policy2	Policy3	校验和			
0x00	0x00	0x16	0xEC			

发送包示例：

策略 0x00000016 为重复指纹检测、自学、360 度

帧头								Data (Len)		帧头 校验		校验密码				命令		Poli cy0	Poli cy1	Polic y2	Poli cy3	校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	02	FC	00	00	00	16	EC	

图 71 设置系统策略发送格式表格

应答包格式（Byte）：

校验密码				CMD1	CMD2	完成码				校验和
0x00	0x00	0x00	0x00	0x02	0xFC	0x00	0x00	0x00	0x00	0x02

应答包示例：

帧头								Data (Len)		帧头 校验	校验密码				命令		错误码				校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	02	FC	00	00	00	00	02

图 72 设置系统策略应答格式表格

### 5.34. 获取指纹模块 ID

发送包格式（Byte）：

校验密码				CMD1	CMD2	校验和	
0x00	0x00	0x00	0x00	0x03	0x01	0xFC	

发送包示例：

帧头								Data (Len)		帧头 校验	校验密码				命令		校验 和
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	03	01	FC

图 73 获取指纹模块 ID 发送包格式表

应答包格式（Byte）：

返回字段长度为 16，如下以返回“ML-FPM001-01-101”为例：

校验密码				CMD1	CMD2	完成码				ID0
0x00	0x00	0x00	0x00	0x03	0x01	0x00	0x00	0x00	0x00	0x4D
ID1	ID2	ID3	ID4	ID5	ID6	ID7	ID8	ID9	ID10	ID11
0x4C	0x2D	0x46	0x50	0x4D	0x30	0x30	0x31	0x2D	0x30	0x31
ID12	ID13	ID14	ID15	校验和						
0x2D	0x31	0x30	0x31	0x75						

应答包示例：

HLK-FPM383C

密级：公开

帧头								Data (Len)		帧头校验	校验密码				命令		错误码				ID0
F1	1F	E2	2E	B6	6B	A8	8A	00	1B	72	00	00	00	00	03	01	00	00	00	00	4D
ID1	ID2	ID3	ID4	ID5	ID6	ID7	ID8	ID9	ID10	ID11	ID12	ID13	ID14	ID15	校验和						
4C	2D	46	50	4D	30	30	31	2D	30	31	2D	31	30	31	75						

图 74 获取指纹模块 ID 应答包格式表

## 5.35. 心跳命令

发送包格式 (Byte) :

一般用于检测与模组的通讯。

校验密码				CMD1	CMD2	校验和
0x00	0x00	0x00	0x00	0x03	0x03	0xFA

发送包示例:

帧头								Data (Len)		帧头校验	校验密码				命令		校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	00	00	00	00	03	03	FA

图 75 心跳命令发送包格式表

应答包格式 (Byte) :

校验密码				CMD1	CMD2	完成码				校验和
0x00	0x00	0x00	0x00	0x03	0x03	0x00	0x00	0x00	0x00	0xFA

应答包示例:

帧头								Data (Len)		帧头校验	校验密码				命令		错误码				校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	03	03	00	00	00	00	FA

图 76 心跳命令应答包格式表

## 5.36. 设置波特率

发送包格式 (4Byte) :

如设置波特率为 9600, 则 Baud0~Baud3 分别为 0x00002580.

校验密码				CMD1	CMD2	Baud0
0x00	0x00	0x00	0x00	0x03	0x04	0x00
Baud1	Baud2	Baud3	校验和			
0x00	0x25	0x80	0x54			

发送包示例:

帧头								Data (Len)		帧头校验	校验密码				命令		Baud0	Baud1	Baud2	Baud3	校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	03	04	00	00	25	80	54

波特率为 115200:

帧头								Data (Len)		帧头校验	校验密码				命令		Baud0	Baud1	Baud2	Baud3	校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	03	04	00	01	C2	00	36

图 77 设置波特率发送包格式表

应答包格式 (Byte) :

HLK-FPM383C

密级: 公开

校验密码				CMD1	CMD2	完成码				校验和
0x00	0x00	0x00	0x00	0x03	0x03	0x00	0x00	0x00	0x00	0xFA

应答包示例：

帧头								Data (Len)		帧头校验	校验密码				命令		错误码				校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	03	04	00	00	00	00	F9

图 78 设置波特率应答包格式表

## 5.37. 设置通信密码

发送包格式 (Byte) :

如设置密码为 0x12345678, 则 PSW0~PSW3 分别为 0x12,0x34,0x56,0x78 (写 flash)

校验密码				CMD1	CMD2	PSW0
0x00	0x00	0x00	0x00	0x03	0x05	0x12
PSW1	PSW2	PSW3	校验和			
0x34	0x56	0x78	0xE4			

发送包示例：

设置密码为：0x12345678

帧头								Data (Len)		帧头校验	校验密码				命令		PSW0	PSW1	PSW2	PSW3	校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	03	05	12	34	56	78	E4

密码恢复默认 0x00000000:

帧头								Data (Len)		帧头校验	校验密码				命令		PSW0	PSW1	PSW2	PSW3	校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	12	34	56	78	03	05	00	00	00	00	E4

图 79 设置通信密码发送包格式表

应答包格式 (Byte) :

校验密码				CMD1	CMD2	完成码				校验和
0x12	0x34	0x56	0x78	0x03	0x05	0x00	0x00	0x00	0x00	0xE4

应答包示例：

密码为：0x12345678

帧头								Data (Len)		帧头校验	校验密码				命令		错误码				校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	12	34	56	78	03	05	00	00	00	00	E4

密码为：0x00000000

帧头								Data (Len)		帧头校验	校验密码				命令		错误码				校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	0B	82	00	00	00	00	03	05	00	00	00	00	F8

图 80 设置通信密码应答包格式表

如再发获取指纹模块 ID 命令，格式如下。

校验密码				CMD1	CMD2	校验和
0x78	0x56	0x34	0x12	0x03	0x01	0xE8

发送包示例：

帧头								Data (Len)		帧头校验	校验密码				命令		错误码				校验和
F1	1F	E2	2E	B6	6B	A8	8A	00	07	86	12	34	56	78	03	01	00	00	00	00	E8

图 81 密码设置成功校验包格式表

## 6. 附录

### 6.1. 错误信息定义表

序号	错误码	说明
1	0x00000000	正常完成
2	0x00000001	无法识别的命令
3	0x00000002	命令数据长度非法
4	0x00000003	命令字段数据非法
5	0x00000004	系统忙，无法执行当前命令
6	0x00000005	没有发送该命令的请求，就查询结果
7	0x00000006	系统软件上报错误
8	0x00000007	硬件错误
9	0x00000008	没有检测到手指按压，超时退出
10	0x00000009	指纹提取发生错误
11	0x0000000A	指纹匹配发生错误(指纹模板库为空)
12	0x0000000B	指纹数据存储空间满
13	0x0000000C	存储写入失败
14	0x0000000D	存储读取失败
15	0x0000000E	采集的指纹图像质量不佳
16	0x0000000F	指纹重复
17	0x00000010	采图面积太小，手指与 sensor 接触面太小
18	0x00000011	采图时手指移动范围过大
19	0x00000012	采图时手指移动范围过小
20	0x00000013	指纹 ID 被占用
21	0x00000014	模组采图失败
22	0x00000015	命令强制中断
23	0x00000016	指纹特征数据不需要更新
24	0x00000017	无效指纹 ID
25	0x00000018	增益调整失败
26	0x00000019	数据缓冲区溢出
27	0x0000001A	sensor 休眠状态下收到采图相关消息返回错误



28	0x0000001C	校验和错误
29	0x00000022	注册存储时写 Flash 失败
30	0x000000FF	其他错误

图 82 错误信息定义表

## 6.2. 上电运行流程

如下图所示，当指纹模块上电后，执行以下流程，上位机在上电或者复位后，需要等待大约 100ms 时间，等待指纹模块完成初始化工作。

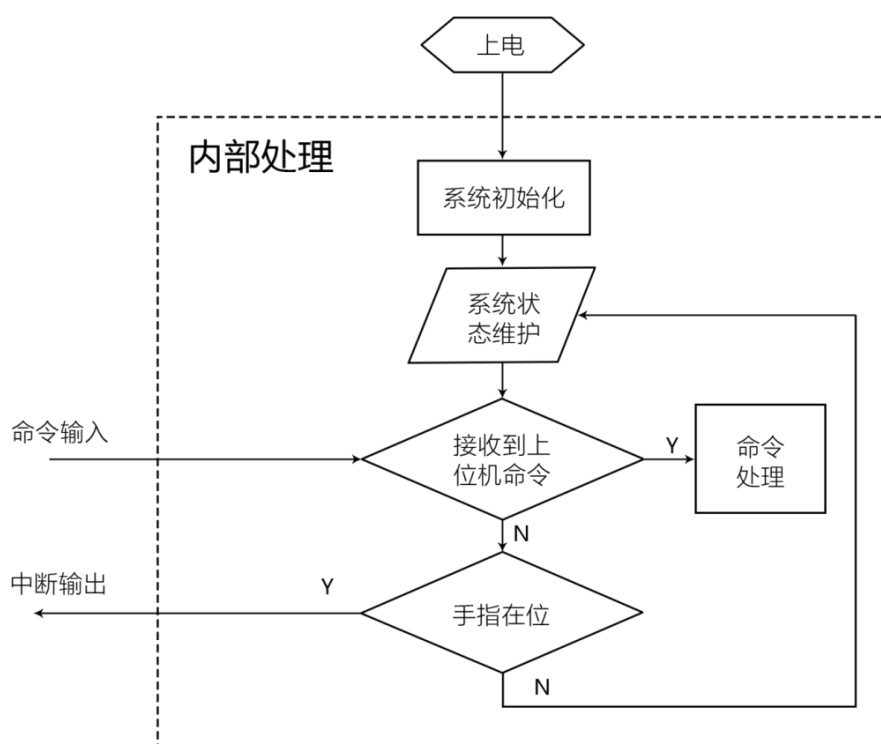


图 83 上电运行流程图

### 说明：

当手指按压在指纹 sensor 上时，不影响指纹模块的初始化，并且初始化完成后，会立刻输出指纹在位的中断信号。

### 6.3. 注册流程

指纹注册流程主要用来增加用户的指纹特征信息，用来进行后续的指纹验证。指纹注册过程如下图所示：

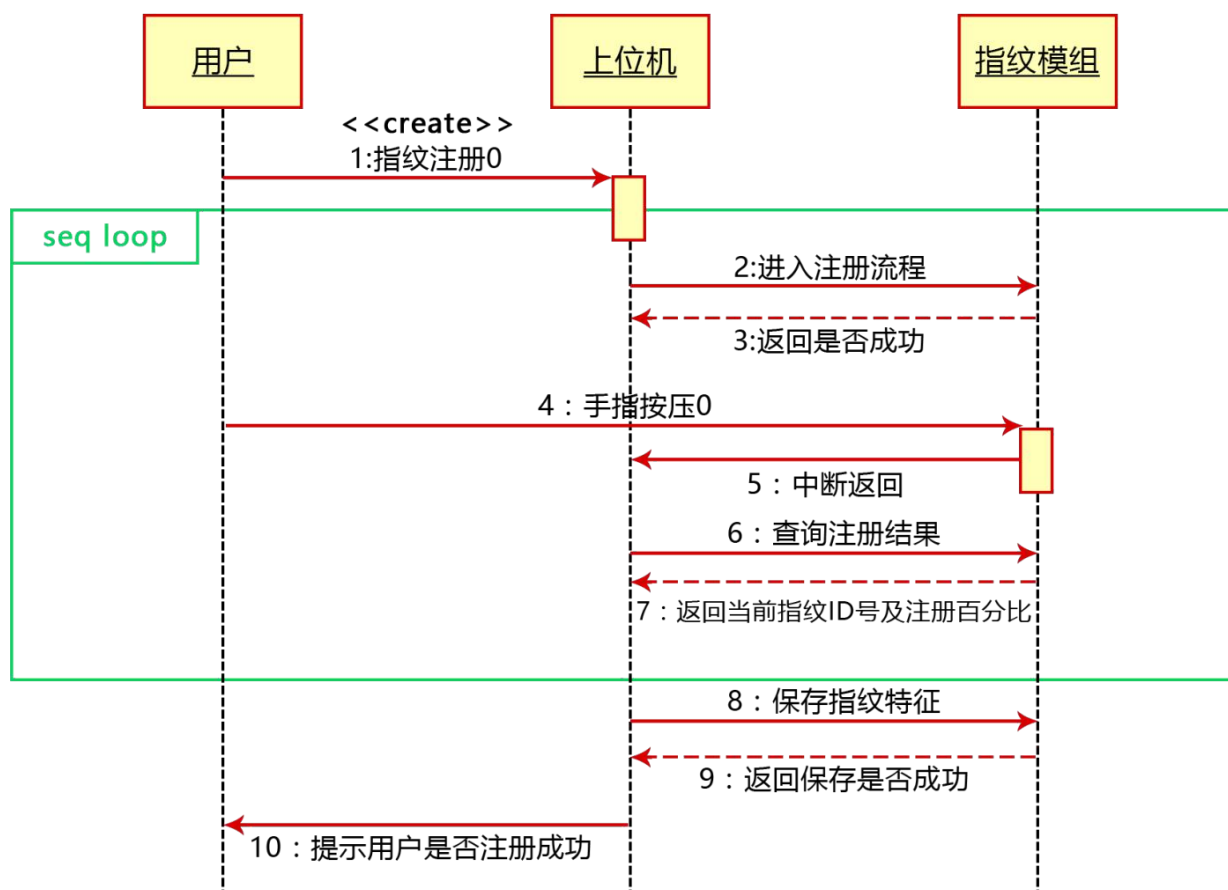


图 84 注册流程图

用户首先在上位机上进行指纹注册操作。上位机在用户操作后会向指纹模块发送进入注册流程命令，指纹模块收到此消息会回复响应，并进入注册流程。在进入注册流程后，超过 10S 没有发送查询注册结果命令，则需要重新发送进入注册流程命令。每次发送查询注册结果的间隔也不能超过 10S，否则也需要重新发送进入注册流程。

注册流程说明如下：

- ① 发送指纹注册命令 (0x01 0x11)。（建议：REG\_IDX 从 0x01 开始，增加到 0x06）
- ② 查询注册结果 (0x01 0x12)。若完成码是 0x00000000，说明本次注册完成，此时可以通过查看 PROC 的值是否增加，如果增加，说明本次注册成功。再者，需要查看 PROC 的值，如果  $PROC < 100$  执行第③步，否则  $PROC \geq 100$ ，说明整个注册流程成功结束，开始执行第④步。若完成码是 0x00000004，意味着 sensor 还没有处理完指纹数据，适当延时 (建议 200ms)，再次获取结果，若是其他原因（如：图像质量不好，手指可能按压过重、过轻、或者太过潮湿等，继续注册即可）执行第①步

③ 查询手指在位状态(0x01 0x35)。如果完成码不是 0x00000000 或者 STATE 是 1 的时候，意味着查询出错或用户手指在上，提示用户手指离开 sensor 再次按压，(建议延时 200ms)再次查询手指在位状态，当完成码是 0x00000000 和 STATE 是 0 的时候，执行第①步。

④ 保存指纹(0x01 0x13)。

⑤ 查询保存指纹结果(0x01 0x14)。若完成码是 0x00000000,说明保存完成。若完成码是 0x00000004,说明未保存完成，延时适当时间(建议 200ms)，再次获取结果。若完成码是 0x0000000B,可能是 flash 存储已满。具体原因用户可根据完成码对照错误信息定义表。

**建议：**因客户需求不同注册的次数有所区别，为方便统一出货，所以建议用户每次注册前发送设置注册次数命令(0x02 0x0D)，设置自己所需的注册次数。

#### 6.4. 验证流程

指纹验证流程主要是对用户进行权限验证，确认指纹是否为已经注册过的指纹。进入指纹匹配流程与查询匹配结果之间不能超过 10S 延时，否则需要重发。

验证流程如下：

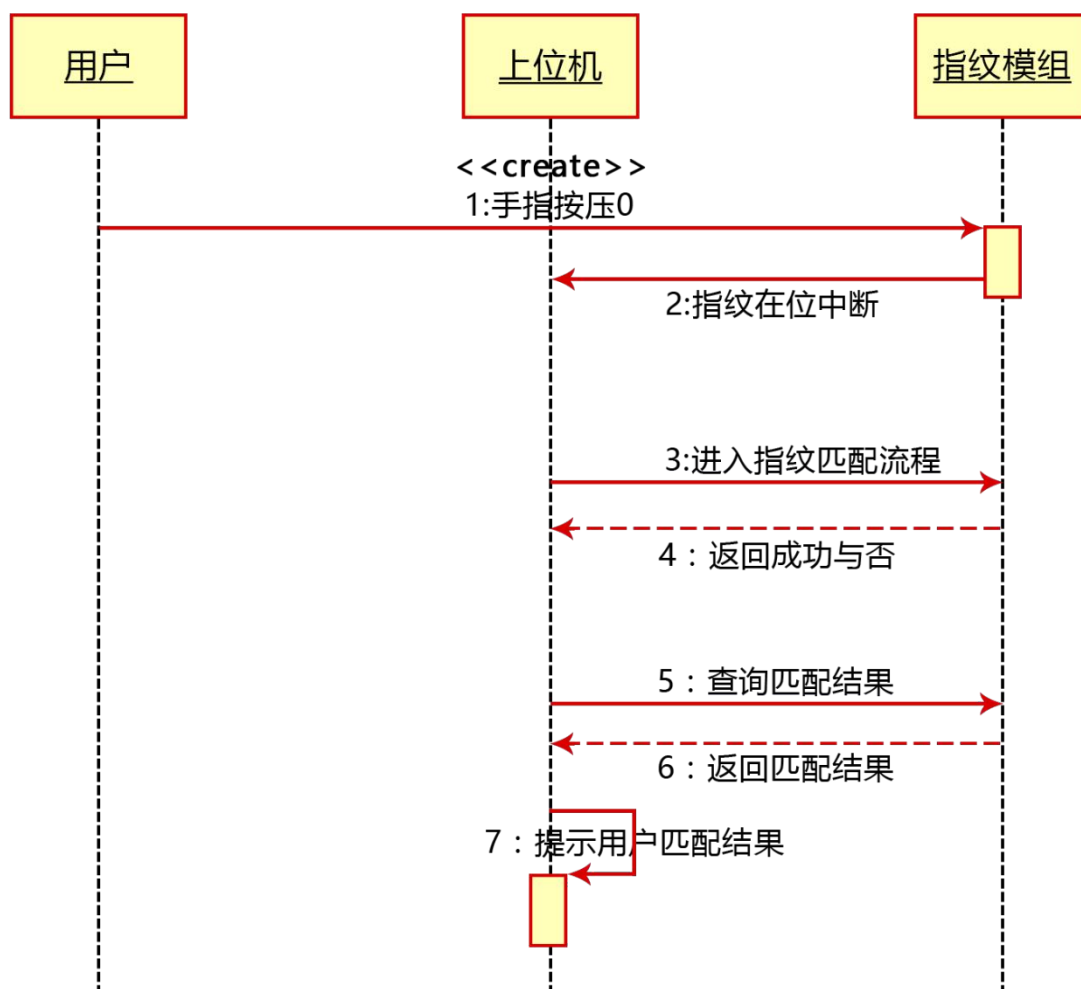


图 85 验证流程图

验证流程说明如下：

- ① 发送指纹匹配命令（0x01 0x21）。发送指纹匹配命令前提示手指按压。
- ② 查询匹配结果（0x01 0x22）。若完成码是 0x00000000, 说明匹配动作完成。此时可查看匹配

结果、匹配分数、匹配指纹 id 号。匹配结果是 1 的话，代表已经匹配到了指纹，否则未匹配到指纹。如果完成码是 0x00000004, 还未匹配完成，延时适当时间(建议延时 200ms)，再次获取结果。如果是其他原因（如：图像质量不好 0x0000000A，手指可能按压过重、过轻、或者太过潮湿等），请重新匹配指纹，执行步骤①。

## 6.5. 删除流程

删除流程主要是为用户删除不必要的手指的指纹。

删除流程如下：

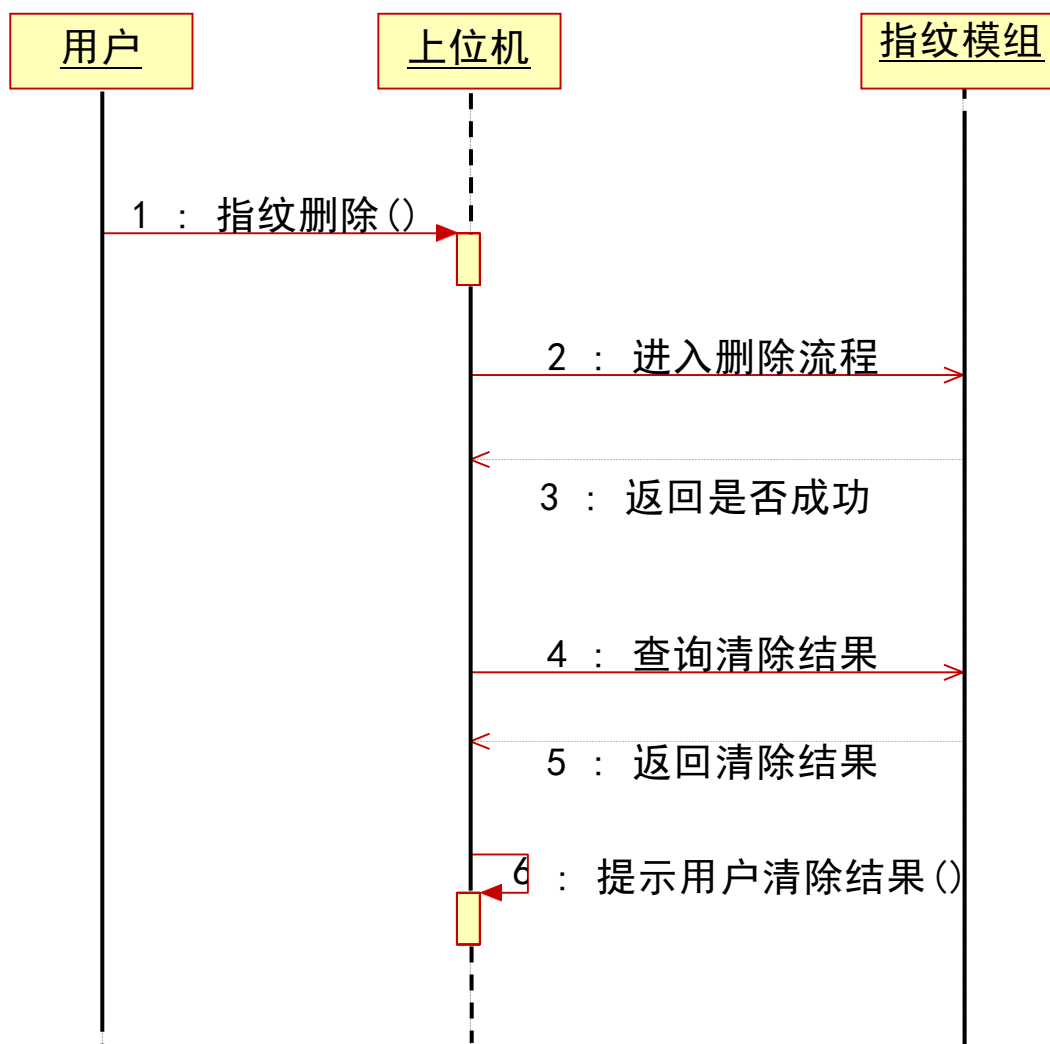


图 86 删除流程图

删除流程说明如下：

① 发送指纹特征清除命令（0x01 0x31）。CL\_ALL 为是否清除所有指纹特征的标志，0x01 表示清除所有指纹，默认 0x00 表示清除单个指纹。ID\_H、ID\_L 为将要删除的指纹 ID。用户可根据自己的需要清除所有、单个或多个（具体发送格式可看指纹特征清除发送示例包）。

② 查询指纹特征清除结果（0x01 0x32）。若完成码为 0x00000000，说明删除成功。若完成码为 0x00000004，说明还未删除完成，延时适当时间（建议 200ms），再次获取结果。其他错误（如：指纹 ID 不存在、flash 硬件错误等）用户可根据完成码查看错误信息对照表，重新发送正确的纹特征清除命令

## 6.6. 校验和函数代码示例

帧头校验和 =  $\sim$ （帧头+应用层数据长度的校验） + 1

发送校验和 =  $\sim$ （校验密码+命令+数据内容） + 1

接收校验和 =  $\sim$ （校验密码+响应命令+错误码+数据内容） + 1

```
static U8Bit FP_protocol_get_checksum(U8Bit * data, U32Bit length)
{
    U32Bit i = 0;
    S8Bit sum = 0;

    for(i = 0; i < length;
        i++){sum += data[i];
    }

    return (U8Bit)((~sum)+1);
}
```

## 6.7. 复位功能(单路供电模组)

用途：唤醒功能（不触摸指纹头的情况下可用复位唤醒）

操作方式：将模组的 RESET 管脚拉低 10ms 再拉高即可。

## 6.8. 自学习功能

自学习是在匹配成功之后将没有注册进去的小部分特征值更新进去，从而增加匹配的成成功率。

流程说明如下：

① 发送匹配命令（0x01 0x21）。按压手指。

- ② 查询指纹匹配结果 (0x01 0x22)。若完成码为 0x00000000 且匹配结果为 0x0001，则匹配成功，执行步骤③。否则失败，无需更新特征值。
- ③ 更新存储特征值 (0x01 0x16)。传入匹配成功后匹配的 ID。
- ④ 查询更新特征值结果 (0x01 0x17)。查询是否更新成功或是否无须更新。