深圳市三能智控电子科技有限公司

用户手册

Communication Protocol Stack B

(Ver 3.0)

2020 年 05 月第三版

目录

（一）概述

本文描述了 三能公司指纹识别模块的串口参数，通讯过程，通讯指令包/数据包格式。

1.1 串口参数、通讯过程、帧序列中传送顺序

串行通讯所用参数如下：

起始位：1 位（1bit）
数据位：8 位 （8bit）
停止位：1 位（1bit）
校验位：无
波特率：9600/19200/38400/57600/115200/230400/460800/921600 ，默认值：115200BPS

通讯过程:

所有指令的发送、接收必须要遵循一发一收的原则。
主机（Host）在没有收到应答时，不可以向目标模块（TARGET ）发送指令。

通讯帧序列传送顺序：

字节（Byte）遵循最低有效位优先传送的规则
字（Word）遵循低字节优先高字节在后传送的规则。

1.2 主要功能

不仅具有如下常规功能：
指纹录入：根据不同型号，最大可录入的指纹容量为 40/80/100/170/200/400/500/1000/1700/3000 枚。
指纹删除：单个删除/指定编号范围删除。
指纹验证：Verify，1:1 比对
指纹识别：Identify，1:N 搜索（在 1-最大容量数内，可任意指定搜索范围）
还具有如下功能：

1. 上传/下载指纹模板数据：
   上传指纹模板数据到主机/下载指纹模板数据到模块
2. 上传/下载指纹图像：
   上传指纹图像到主机/下载指纹图像到模块（提取指纹特征/录入/与活体指纹验证/识别）
3. 检查指定编号范围内的指纹模板数据是否有坏损情况
4. 读取模块中已注册的用户总数及用户列表
5. 设置/读取指纹模块参数（安全等级、允许/禁止自学习、允许/禁止指纹重复检查、波特率 等）
6. 设置/读取指纹模块的序列号
7. 在板固件更新
8. 读取指纹模块信息（固件类型及版本、算法芯片、指纹传感器）等模块的软硬件信息。

1.3 获取的指纹模块信息说明

利用读取模块信息指令（CMD_DEVICE_INFO）

或 Demo 演示程序中“获取模块信息”按钮得到的信息如下：

模块信息组成规则：

ID_协议栈及算法名称_算法芯片型号_采集器型号_无或模块硬件名称_无或 Inner_(指纹容量)_版本号

ID：表示 三能公司提供

通讯协议栈(由至少 4 个字母组成)及指纹识别算法名称（无或一个数字）：

通讯协议栈支持如下几种：

SEOU：三能通讯协议栈 A（简称：老协议）

SEONU:三能通讯协议栈 B（简称：新协议），本手册描述的通讯协议文档

SEODU:三能通讯协议栈 A+B 双协议都支持（简称：三能双协议）

SYNONU:三能协议栈 B+晟元协议 （简称：三能+晟元双协议）

指纹识别算法名称分类如下：

无：常规通用算法，适用于所有指纹传感器，指纹特征数据大小：496 字节

2： 简称 160 新算法，仅支持 160*160 像数的指纹传感器，指纹特征数据大小：1008 字节

5： 非特征算法，仅支持 FPC1021/1025 指纹传感器，指纹模板数据大小：2024 字节

7： 简称 192 新算法，仅支持 192*192 像数的指纹传感器，指纹特征数据大小：448 字节

算法芯片型号：QS808/ID808/ID809/ID811

ID808：Cotex-M4 内核，144Mhz 主频，96K RAM+1MB FLASH，QFP48 封装(9mm*9mm)

ID809：Cotex-M4 内核，144Mhz 主频，128K RAM+1MB FLASH，QFN36 封装 (6*6)

ID811：Cotex-M4 内核，150/120Mhz 主频，128K RAM+512KB FLASH,内置国密算法，QFN32 封装(4*4)

采集器型号：市场主流的指纹传感器（光学、各种面阵/滑动的电容传感器）几乎都有相应固件支持。

指纹容量：（括号内的数字表示本模块最大指纹容量）

注意：使用模块时这个参数非常重要，不确定时请咨询我司销售，或通过指令或用演示程序读取指纹容量。

1.识别指令（CMD_SEARCH）中指定搜索的 ID 编号范围不能超出指纹容量，否则返回参数错误。

2.使用我司的演示程序时，要选择跟本模块相匹配的指纹容量，否则连续识别时会提示参数错误。

获取的模块信息说明示例：

读取到的模块信息	信息描述
ID_SEONU_ID809_GC0308_DORLO(3000fp)_V1.0	协议栈：三能指令集 B，算法：通用采集器老算法 算法芯片：ID809；采集器：光学（GC0308） 模块型号：DORLO；指纹容量：3000 枚，版本：V1.0
ID_SEODU2_ID809_BF5325_PDI48_Inner(200fp)_V1.1	协议栈：兼容三能指令集 A/B，算法：160*160 新算法 算法芯片：ID809,采集器：BF5325 模块型号：PDI48，指纹容量：200 枚，版本：V1.1
ID_SYNONU5_ID811A1_FPC1021_Inner(40fp)_V1.3	协议栈：兼容三能/晟元协议；算法：FPC1021 专用算法 算法芯片：ID811,采集器：FPC0121；指纹容量：40 枚
ID_SYNONU7_ID808_FPC1020_Inner(500fp)_V1.0	协议栈：兼容三能/晟元协议；算法：192*192 新算法 算法芯片：ID811；采集器：FPC1020；指纹容量：500 枚

1.4 固件中不同算法名称的区别

算法名称 / 比较项目	适用于所有采集器的通用算法 例：SEONU	仅适用于 160*160 采集器的算法 例：SEONU2	仅适用于 192*192 采集器的算法 例：SEONU7	FPC0121/1025 采集器的专用算法 例：SEONU5
指纹模板数据大小	498 字节	1008 字节	448 字节	2024 字节
指纹重复检查功能	支持	不支持	支持	不支持
下载模板与模块中的指纹库搜索比对	支持	不支持	支持	不支持
支持的指纹采集器	任意尺寸	任意 160*160 像数	任意 192*192 像数	仅 FPC1021/1025
推荐的采集器	光学、A288/A360 TCS1/2、FPC1011	BF5325/6632/5838 A176,FPC1021/1025	BF5333/5836 A192、TL192、FPC1020/1024	FPC1021/1025

1.5 应用开发快速入门

应用开发者参考本章节及相关链接的内容可快速完成模块的应用开发。

1. 模块的指纹容量及算法版本未知时：请用指令 CMD_DEVICE_INFO（参见§4.4及§1.3章节内容）
2. 登记指纹：参考§6.1章节的登记流程，内附了各流程步骤用到的每条指令，即执行如下几条指令：
   执行{采集指纹图像（CMD_GET_IMAGE） ->转化成特征模板（CMD_GENERATE）}3 次后-> 融合指纹模板（CMD_MERGE） ->保存指纹模板（CMD_STORE_CHAR）
3. 验证/识别指纹：参考§6.2章节的指纹验证/识别流程，即执行如下几条指令：
   采集指纹图像 CMD_GET_IMAGE-> 生成特征模板 CMD_GENERATE->验证 CMD_VERIFY/或识别 CMD_SEARCH

以上用到的各指令对应的章节列表如下：

指令名称	指令码	参考章节	指令功能简要描述
CMD_GET_IMAGE	0x0020	§4.6	采集指纹图像并暂存在 ImageBuffer 中
CMD_GENERATE	0x0060	§4.20	从 ImageBuffer 图像生成指纹特征并暂存在 RamBuffer0/1/2
CMD_MERGE	0x0061	§4.21	将暂存在 RamBuffer0/1/2 中的三个模板融合生成一个待用的模板，暂存于指定的 Ram Buffer 中
CMD_STORE_CHAR	0x0040	§4.11	把 Ram Buffer 中的模板数据保存到指纹库中指定编号的空间中
CMD_SEARCH	0x0063	§4.23	将指定编号的 Ram Buffer 中的模板在指纹库中检索比对，搜索范围由本指令设定（例如：1-满容量范围内），得到检索结果。
CMD_VERIFY	0x0064	§4.24	将指定编号的 Ram Buffer 中的模板与指纹库中本条指令设定的某个模板进行比对，得到比对结果。

表 1.5：应用开发快速入门必须用到的指令列表

注：指令执行后的应答包中的结果码和错误代码参见第五章的响应及错误代码表

应用开发者也可以用参考我司提供的测试程序源码（NOEMHost_v3.18_20191230.7z）
或直接用串口监控程序（例如：Bus Hound 或 AccessPort 等监控软件），监控指纹模块与测试程序的通讯过程，得到指令收/发结果，对照本手册对应指令的详细描述即可快速理解本手册。

（二）通讯协议（协议栈 B）-综述

模块上电后固件 BOOT 需要时间（即硬件及算法初始化时间），主机必须等待模块完成初始化后才能给模块发送指令

指纹模组在上电后 MCU 的 GPIO 及 UART 端口初始化成功后，会通过 UART 发送一个字节的 0x55，作为通知主机的握手信号。

主机在控制指纹模块上电后等待模块初始化时，可以通过接收此握手信号，提前进入工作状态。

注： 主机控制模块上电后，可通过如下两种方法开始通讯过程

1. 主机收到模块的握手信号 0x55 后，即可开始给模块发送指令
2. 主机控制模块上电后，延时 280ms 即可开始给模块发送指令

2.1 通讯处理过程

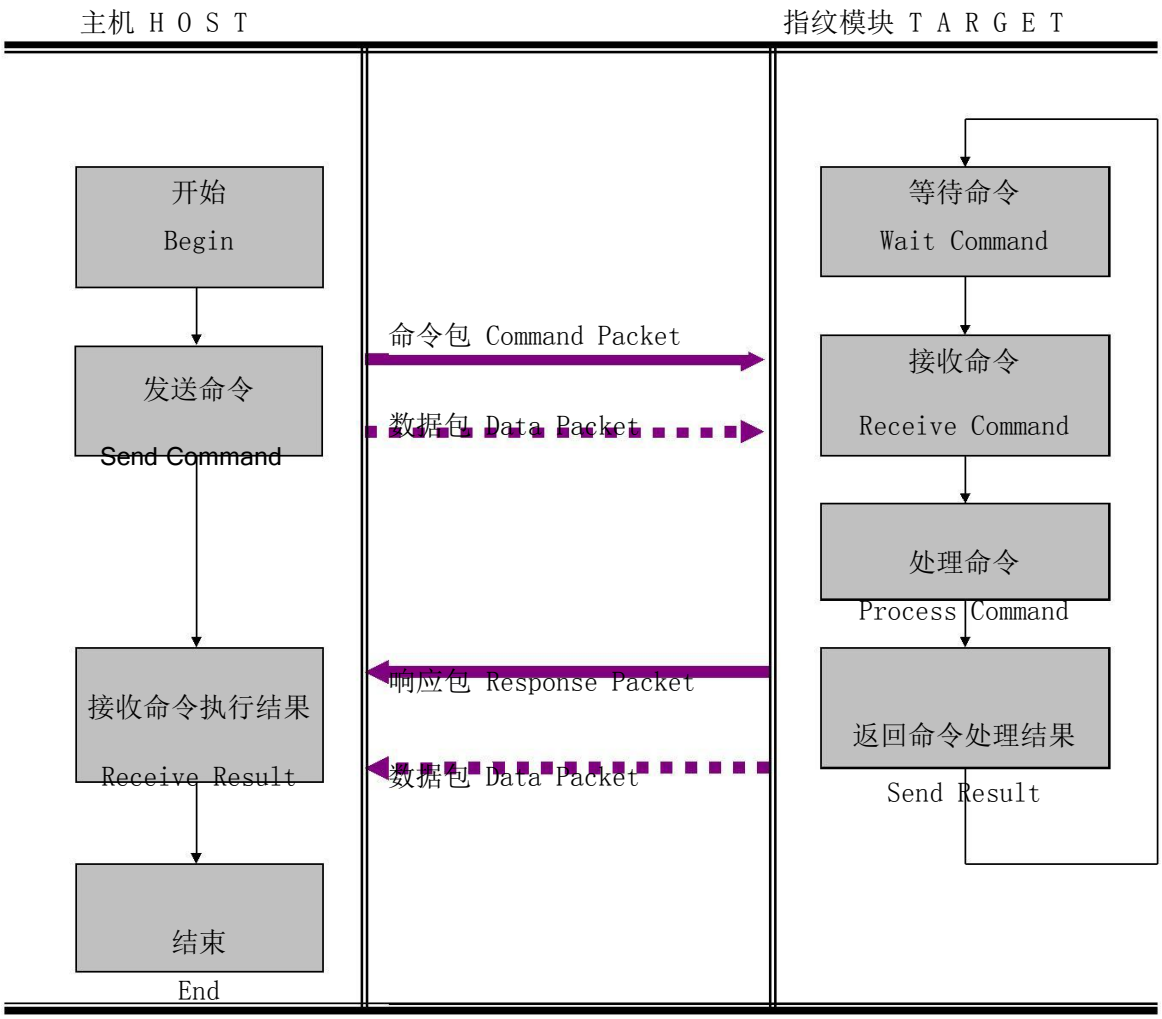

图 2-1 协议栈 B 的通讯过程

注：

通讯过程中，所有指令的发送、接收必须要遵循一发一收的原则。

Host 主机在没有收到模块的应答时，不可以向 TARGET 发送指令。

2.2 通讯包 Packet 的分类

2.2.1 命令包 Command packet

命令包说明从 Host 至 Target 的指令内容。
从 Host 中发出的所有指令，都通过命令包 Command packet 传输。
命令包 Command packet 的帧长度为 26 字节 bytes。

2.2.2 响应包 Response packet

响应包指从 Target 至 Host 的应答内容。
所有指令收到相应处理结果即 Response packet 后终止其使命。
响应包 Response packet 的长度为 26 字节 byte 。

2.2.3 指令/响应的数据包 Data Packet

当指令参数或响应数据的长度大于 16bytes 时，利用指令/响应数据包 Data Packet 传输数据。
Host 须在发送指令数据包之前，利用命令包 Command packet 将数据包的长度告知模块 Target
指令参数或相应数据包的最大长度为 500bytes

2.3 通讯包的帧结构

2.3.1 通讯包 Parket 识别代码

通讯包 Packet 的开始 2byte 为表示通讯包 packet 种类的识别码，其如下表 2-1：

Packet 类别	Code 包类别识别码
命令包 Command packet	0xAA55
响应包 Response packet	0x55AA
指令数据包 Command Data Packet	0xA55A
响应数据包 Response Data Packet	0x5AA5

表 2-1 Packet 识别代码

2.3.2 命令包（Command packet ）的帧结构

PREFIX		SID	DID	CMD		LEN		DATA				CKS	
0x55	0xAA	源 ID	目标 ID	L	H	L	H	D0	D1	…	D15	L	H
0	1	2	3	4	5	6	7	8	9	…	23	24	25

表 2-2 命令包（Command packet）的结构如下：

偏移值 OFFSET	域定义 FIELD	数据类型 TYPE	字节数 SIZE	描述 DESCRYPTION
0	PREFIX	WORD	2byte	包识别码 Packet Identify code
2	SID	BYTE	1byte	源标识 Soruce Device ID
3	DID	BYTE	1byte	目标标识 Destination Device ID
4	CMD	WORD	2byte	命令字 Command Code

6	LEN	WORD	2byte (=n, n < 16)	数据长度 Length of DATA
8	DATA	Byte Array	16byte	命令参数 Command Parameter (实际数据为 n byte)
24	CKS	WORD	2byte	校验和 Check Sum：从 PREFIX 到 DATA 域的所有数据算术和运算后的最低 2 个字节

2.3.3 响应包（Response packet）的帧结构

PREFIX		SID	DID	RCM		LEN		RET		DATA				CKS	
0x55	0xAA	源 ID	目标 ID	L	H	L	H	L	H	D0	D1	…	D13	L	H
0	1	2	3	4	5	6	7	8	9	10	11	…	23	24	25

表 2-3 响应包（Response packet）的结构如下：

偏移值 OFFSET	域定义 FIELD	数据类型 TYPE	字节数 SIZE	描述 DESCRYPTION
0	PREFIX	WORD	2byte	包识别码 Packet Identify code
2	SID	BYTE	1byte	源标识 Soruce Device ID
3	DID	BYTE	1byte	目标标识 Destination Device ID
4	RCM	WORD	2byte	响应码 Response Code
6	LEN	WORD	2byte(=n, n < 16)	长度 Length of RET and DATA
8	RET	WORD	2byte	结果码 Result Code(0 :成功, 1 :失败)
10	DATA	Byte Array	14byte	响应数据 Response Data(实际为 n-2 byte)
24	CKS	WORD	2byte	校验和 Check Sum：从 PREFIX 到 DATA 域 的所有数据算术和运算后的最低 2 个字节

2.3.4 指令数据包（Command Data Packet）的帧结构

PREFIX		SID	DID	CMD		LEN		DATA				CKS	
0x5A	0xA5	源 ID	目标 ID	L	H	L	H	D0	D1	…	Dn-1	L	H
0	1	2	3	4	5	6	7	8	9	…	8+n-1	8+n	8+n+1

表 2-4 指令数据包（Command Data Packet）的结构如下：

偏移值 OFFSET	域定义 FIELD	数据类型 TYPE	字节数 SIZE	描述 DESCRYPTION
0	PREFIX	WORD	2byte	包识别码 Packet Identify code
2	SID	BYTE	1byte	源标识 Source Device ID
3	DID	BYTE	1byte	目标标识 Destination Device ID
4	CMD	WORD	2byte	命令码 Command Code
6	LEN	WORD	2byte(=n , n <500)	数据长度 Length of DATA
8	DATA	Byte Array	nbyte	命令参数 Command parameter
8+n	CKS	WORD	2byte	校验和 Check Sum：从 PREFIX 到 DATA 域 的所有数据算术和运算后的最低 2 个字节

Host 须在发送指令数据包之前先传输命令包（Command packet），使得模块 Target 进入指令数据包（Command Data packet）接收等待状态。

在该命令包（Command packet）的数据域（DATA field）中，须设定待传输的指令数据包的长度。

Host 应在确认 Target 处于指令数据包接收等待状态后传输指令数据包（Command Data Packet）。

2.3.5 响应数据包（Response data packet）的帧结构

PREFIX		SID	DID	RCM		LEN		RET		DATA				CKS	
0xA5	0x5A	源 ID	目标 ID	L	H	L	H	L	H	D0	D1	…	Dn-3	L	H
0	1	2	3	4	5	6	7	8	9	10	11	…	8+n-1	8+n	8+n+1

表 2-5 响应数据包（Response Data Packet）的结构如下：

偏移值 OFFSET	域定义 FIELD	数据类型 TYPE	字节数 SIZE	DESCRYPTION
0	PREFIX	WORD	2byte	包标识 Packet Identify code
2	SID	BYTE	1byte	源标识 Soruce Device ID
3	DID	BYTE	1byte	目标标识 Destination Device ID
4	CMD	WORD	2byte	响应码 Response Code
6	LEN	WORD	2byte(=n, n <500)	结果接数据长度 Length of result data (RET + DATA)
8	RET	WORD	2byte	结果码 Result code(0 : 成功, 1 : 失败)
10	DATA	Byte Array	n-2 byte	响应数据 Response data
8+n	CKS	WORD	2byte	校验和 Check Sum：从 PREFIX 到 DATA 域 的所有数据算术和运算后的最低 2 个字节

注：从模块 Target 至 Host 中传输 14byte 以上数据时，需利用响应数据包（Response data packet）

（三）通讯命令（Command ）简介

3.1 通讯命令中用到的一些概念

指令执行需要用到暂存指纹图像的缓存 ImageBuffer 和暂存指纹模板数据的缓存 Ram Buffer 。

ImageBuffer： 用于暂存指纹图像。

Ram Buffer： 用于暂存指纹特征模板数据。

模块共开放了三个 Ram Buffer ：Ram Buffer0/Ram Buffer1/Ram Buffer2 。

注：断电情况下，暂存在 ImageBuffer 和 Ram Buffer 中的数据会丢失。

指纹库：保存在闪存（FLASH Memory）中指纹模板的集合，

可通过 CMD_STORE_CHAR /CMD_LOAD_CHAR 写入/读出指纹模板数据。

3.2 各版算法生成的指纹模板（Template Record）数据

不同版本的算法生成的指纹模板数据长度是不一样的，在调用上传指纹模板（CMD_UP_CHAR）/下载指纹模板（CMD_DOWN_CHAR）指令时，要注意各版算法生成的指纹模板数据长度的区别（见下表）。

指纹模板数据的构成如下：指纹特征数据(Feature Data)+ 2 字节校验和（CheckSum）

注：CheckSum 为：Feature Data 各字节的算术和运算后的最低 2 字节

算法版本	指纹模板数据长度
常规通用算法，例如固件 SEODU	498 字节
160 新算法,例如固件 SEONU2	1008 字节
192 新算法，例如固件 SYNONU7	448 字节
非标算法，例如固件 SYNONU5	2024 字节

表 3-2 各版算法生成的指纹模板（Template Record）数据长度

3.3 命令列表（ Command List ）

序号 No.	命令名称 Command Name	命令码 Code	命令功能 Function
1	CMD_TEST_CONNECTION	0x0001	进行与设备的通讯连接测试指令
2	CMD_SET_PARAM	0x0002	设置模块如下参数，注：TimeOut 只适用于滑动采集器 （Device ID, Security Level, Baudrate, Duplication Check, Auto Learn，TimeOut）
3	CMD_GET_PARAM	0x0003	获取模块如下参数： (Device ID, Security Level, Baudrate, Duplication Check, Auto Learn，TimeOut)
4	CMD_GET_DEVICE_INFO	0x0004	获取模块设备信息
5	CMD_ENTER_IAP_MODE	0x0005	将模块设置为 IAP（在应用中编程）状态
6	CMD_GET_IMAGE	0x0020	从采集器采集指纹图像并暂存于 ImageBuffer 中
7	CMD_FINGER_DETECT	0x0021	检测指纹输入状态（有/无手指）
8	CMD_UP_IMAGE	0x0022	将保存于 ImageBuffer 中的指纹图像上传至主机 HOST
9	CMD_DOWN_IMAGE	0x0023	从主机（HOST）下载指纹图像到模块的 ImageBuffer 中
10	CMD_SLED_CTRL	0x0024	控制光学采集器背光灯的开/关，或 LED 灯提示指令
11	CMD_STORE_CHAR	0x0040	保存指定编号 RamBuffer 中的模板数据到指定编号的指纹库中。
12	CMD_LOAD_CHAR	0x0041	读取指定编号的模板数据暂存到指定编号的 Ram Buffer（0/1/2）
13	CMD_UP_CHAR	0x0042	将暂存在指定编号 Ram Buffer 中的指纹模板数据上传至 HOST。
14	CMD_DOWN_CHAR	0x0043	从 HOST 下载一个指纹模板到模块的指定编号 Ram Buffer 中
15	CMD_DEL_CHAR	0x0044	删除指纹库中的指定编号范围内的所有指纹模板 。
16	CMD_GET_EMPTY_ID	0x0045	获取指定范围内可注册的（没有注册过的）第一个模板编号。
17	CMD_GET_STATUS	0x0046	获取指定编号的模板注册状态。
18	CMD_GET_BROKEN_ID	0x0047	检查指定编号范围内的所有指纹模板是否存在坏损的情况
19	CMD_GET_ENROLL_COUNT	0x0048	获取指定编号范围内已注册的模板个数。
20	CMD_GENERATE	0x0060	将暂存在 ImageBuffer 中的指纹图像生成模板数据，并保存于指定 编号的 Ram Buffer 中。
21	CMD_MERGE	0x0061	将暂存于 Ram Buffer 中的两或三个模板数据融合成一个模板数据 注：融合后的模板数据暂存在 RamBuffer 0
22	CMD_MATCH	0x0062	指定 Ram Buffer 中的两个指纹模板之间进行 1:1 比对（验证）
23	CMD_SEARCH	0x0063	指定 Ram Buffer 中的模板与指纹库中指定编号范围内的所有模 板之间进行 1:N 搜索（识别）
24	CMD_VERIFY	0x0064	指定 Ram Buffer 中的指纹模板与指纹库中指定编号的指纹模板 之间进行 1:1 比对（验证）
25	CMD_SET_MODULE_SN	0x0008	在设备中设置模块序列号信息（Module SN）
26	CMD_GET_MODULE_SN	0x0009	获取本设备的模块序列号（ Module SN）
27	CMD_ADJUST_SENSOR	0x0025	调节指纹采集器参数。注：有些型号模块不支持此指令

序号 No.	命令名称 Command Name	命令码 Code	命令功能 Function
28	CMD_GET_ENROLLED_ID_LIST	0x0049	获取已注册 User ID 列表
29	CMD_ENTER_STANDY_STATE	0x000C	使模块进入休眠状态。注：有些型号模块不支持休眠功能

（四）各通讯命令（Command）详细说明

4.1 连接测试（CMD_TEST_CONNECTION 0x0001）

[功能 Function]

检查 Target 和 Host 的连接状态。

Host 需要首先发送此指令检查与 Target 的连接状态。

若不成功，则可认为与 Target 的连接不正常，或 Target 的工作不正常，或串口波特率不正确。

[工作过程 Sequence]

连接正常，则返回 ERR_SUCCESS 。

[命令和响应 Command and Response]

PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID
CMD	0x0001
LEN	0
DATA	无数据
PREFIX	0x55AA
SID	Source Device ID
DID	Destination Device ID
RCM	0x0001
LEN	2
RET	Result Code (ERR_SUCCESS/ ERR_FAIL)
DATA	无数据

表 4-1 CMD_TEST_CONNECTION 指令

实例 4.1-主机发送 CMD_TEST_CONNECTION 指令及模块的响应

主机命令：55 AA 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01

模块响应：AA 55 01 00 01 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 01

4.2 设置参数（CMD_SET_PARAM 0x0002）

[功能 Function]

根据指定的参数类型（Parameter Type），设置设备参数（Device ID, Security Level, Baudrate, Duplication Check, Auto Learn, FP TimeOut）的值并返回其结果。

[工作过程 Sequence]

- 若指定 Parameter Type 无效，则返回 ERR_INVALID_PARAM 。
- 若指定 Parameter Value 无效，则返回 ERR_INVALID_PARAM 。
- 根据 Parameter Type，设置 Parameter Value 并返回其结果。

[命令和响应 Command and Response]

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0002	
LEN	5	
DATA	1bytes	Parameter Type
	4bytes	Parameter Value
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0002	
LEN	2	
RET	Result Code (ERR_SUCCESS/ ERR_FAIL)	
DATA	无数据	

表 4-2 CMD_SET_PARAM 指令

[参数类型 Parameter Type]

Type	Parameter Value Description 参数值描述
0	表示本模块设备编号（Device ID）。可设置范围： 1-255 。
1	表示安全等级（Security Level），默认值为：3

	可设置值：1-5（注：有些特殊需求的固件 ，可设置值为 1-10） 数值越大：误识率（FAR）越低，但拒真率（FRR）越高，通俗讲即如下 设置的安全等级越高：则越不容易出现误识，但也不容易识别通过； 反之，安全等级越低：则容易通过但也容易出现误识。
2	指纹重复检查（Duplication Check）状态开/关。可设置 0 或 1。 若为 1 ，则处理 CMD_STORE_CHAR 指令时，检查是否有已登记的重复指纹。 若为 0 ，则不进行重复检查。 注：指纹重复检查开/关，只对通用算法（498 字节特征算法）的固件有效，其它算法的固件不检测指纹重复（即使置为 1，也不做指纹重复检查）
3	波特率（Baudrate）的索引值。可设置索引值:1-8 。默认：5（即 115200bps） 1:9600bps, 2:19200bps, 3:38400bps, 4:57600bps, 5:115200bps 6:230400bps, 7:460800bps, 8:921600bps
4	表示指纹模板自学习（Auto Learn）状态开/关。可设置 0 或 1 。 若为 1 ：则处理 CMD_SEARCH, CMD_VERIFY 指令时进行智能更新。 若为 0 ：则不进行智能更新。
5	表示采集指纹超时时间（ Fp TimeOut）参数，可设置值：1 秒至 60 秒。 CMD_GET_IMAGE 指令中采用该参数，在 FP TimeOUT 时间内等待指纹的输入。 注：本参数只用于滑动指纹传感器模块，默认值为：5s

实例 4.2-例 1：设置波特率为 57600bps

主机命令： 55 AA 00 00 02 00 05 00 03 04 00 00 00 00 00 00 00 00 00 00 00 00 0d 01

模块响应： AA 55 01 00 02 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 04 01

实例 4.2-例 2：设置安全等级=5

主机命令： 55 AA 00 00 02 00 05 00 01 05 00 00 00 00 00 00 00 00 00 00 00 00 0c 01

模块响应： AA 55 01 00 02 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 04 01

4.3 读取参数（CMD_GET_PARAM 0x0003）

[功能 Function]

根据指定的参数类型（Parameter Type），获取模块 (Device ID, Security Level, Baudrate, Duplication Check, Auto Learn)等的参数值。

有关 Parameter Type，请参考上述 CMD_SET_PARAM 。

[工作过程 Sequence]

① 若指定 Parameter Type 无效，则返回 ERR_INVALID_PARAM 。

② 返回指定 Parameter Type 相应的设备参数。

[命令及响应 Command and Response]

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0003	
LEN	1	
DATA	1byte	Parameter Type
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0003	
LEN	成功 : 6, 失败 : 2	
RET	Result Code (ERR_SUCCESS/ ERR_FAIL)	
DATA	4bytes	成功时：Parameter Value

表 4-3 CMD_GET_PARAM 指令

实例 4.3- 例 1：读取当前安全等级（返回安全等级=3）

主机命令： 55 AA 00 00 03 00 01 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 04 01

模块响应： AA 55 01 00 03 00 06 00 00 00 03 00 00 00 00 00 00 00 00 00 00 00 00 0C 01

实例 4.3- 例 2：读取当前 TimeOut 值（TimeOut=5S）；用于滑动指纹模块

主机命令： 55 AA 00 00 03 00 01 00 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 08 01

模块响应： AA 55 01 00 03 00 06 00 00 00 05 00 00 00 00 00 00 00 00 00 00 00 00 0E 01

4.4 读取设备信息（CMD_DEVICE_INFO 0x0004）

[功能 Function] 获取模块的版本等设备信息（Device Information of Target）。

本设备信息格式如下：参见概述章节中的“模块信息的组成规则”

ID_协议栈及算法名称_算法芯片型号_采集器型号_无或模块硬件名称_无或 Inner_(指纹容量)_版本号

重点关注模块信息中的算法名称及指纹容量的信息。

[工作过程 Sequence]

①首先利用指令应答包，将下次发送的应答数据包的数据长度发送至 HOST。

②利用应答数据包，发送 Device Information 。

[命令及响应 Command and Response]

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0004	
LEN	0	
DATA	无数据	
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0004	
LEN	4	
RET	ERR_SUCCESS	
DATA	2bytes	数据应答包的数据长度
成功时		
PREFIX	0x5AA5	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0004	
LEN	2 + Device Information 长度	
RET	ERR_SUCCESS	
DATA	Device Information	

表 4-4 CMD_DEVICE_INFO 指令

实例 4.4-读取模块信息 CMD_DEVICE_INFO

主机命令： 55 AA 00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 01

模块应答： AA 55 01 00 04 00 04 00 00 00 1A 00 00 00 00 00 00 00 00 00 00 00 00 22 01

响应数据包： A5 5A 01 00 04 00 1F 00 00 00 53 45 4F 4E 5F 47 44 5F 46 50 43 31 30 32 30 28 32 30 30 30 66 70 29 20 56 31 2E 30 00 2B08；

蓝色突出显示的数据块为设备信息“SEON_GD_FPC1020(2000fp) V1.0”的 ASCII 码

4.5 使模块进入 IAP 模式（CMD_ENTER_IAP_MODE 0x0005）

[功能 Function]

将设备设置为 IAP 状态。

[工作过程 Sequence]

收到指令包后，将设备设置为 IAP 状态。

[命令及响应 Command and Response]

PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID
CMD	0x0005
LEN	0
DATA	无数据

PREFIX	0x55AA
SID	Source Device ID
DID	Destination Device ID
RCM	0x0005
LEN	2
RET	Result Code
DATA	无数据

表 4-5 CMD_ENTER_IAP_MODE 指令

注：CMD_ENTER_IAP_MODE 命令将清除固件程序，需要升级固件时才需执行该指令。执行该指令后必须用 USB 重新烧写固件，请慎用该指令！！

4.6 采集指纹图像（CMD_GET_IMAGE 0x0020）

[功能 Function]

从采集器采集指纹图像并保存于 ImageBuffer 中。

[工作过程 Sequence]

从采集器采集指纹图像。若采集图像正确，则返回 ERR_SUCCESS 。否则返回错误码。

对于滑动式半导体指纹传感器：

- 若在 Fp TimeOut 时间内没有检测到指纹，则返回 ERR_TIME_OUT。
- 若在采集过程中或等待指纹输入过程中收到 CMD_FP_CANCEL 指令，则取消此指令的运行并返回 ERR_FP_CANCEL 。

[命令及响应 Command and Response]

PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID
CMD	0x0020
LEN	0
DATA	无数据
PREFIX	0x55AA
SID	Source Device ID
DID	Destination Device ID
RCM	0x0020
LEN	2
RET	Result Code
DATA	0

表 4-6 CMD_IMAGE 指令

实例 4.6- 例 1：发送采集指纹图像后模块检测到手指的命令及响应

主机命令：55 AA 00 00 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 1F 01
模块响应：AA 55 01 00 20 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 22 01

实例 4.6- 例 2：发送采集滑动指纹图像后结果超时（FP TimeOut）的命令及响应

CMD_GET_IMAGE:55 AA 00 00 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 1F 01
ERR_TIME_OUT: AA 55 01 00 20 00 02 00 23 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 45 01

4.7 检测手指（CMD_FINGER_DETECT 0x0021）

[功能 Function]

检查收到指令时刻指纹输入状态并返回其结果。

[工作过程 Sequence]

返回收到该指令的时刻，Sensor 检测到的指纹输入状态。

[命令及响应 Command and Response]

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0021	
LEN	0	
DATA	无数据	
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0021	
LEN	成功 ：3，失败：2	
RET	Result Code	
DATA	1byte	成功时：指纹输入状态 （1：有指纹输入，0：无指纹输入）

表 4-7 CMD_FINGER_DETECT 指令

实例 4.7-例 1：没检测到指纹

Host 命令： 55 AA 00 00 21 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 20 01

Target 响应： AA 55 01 00 21 00 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 24 01

实例 4.7-例 2：检测到有指纹

Host 命令： 55 AA 00 00 21 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 20 01

Target 响应： AA 55 01 00 21 00 03 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 25 01

4.8 上传指纹图像到主机（CMD_UP_IMAGE_CODE 0x0022）

[功能 Function]

根据指定 Image Type ，将保存于 ImageBuffer 中的图像发送至 Host 。

若 Image Type 为 0：则发送全图：

（光学采集器及半导体采集器：242*266(202*258)，滑动采集器 FPC1080：128*436）。

若为 1 ：则发送 1/4 图像（4 个点取 1 个点）。

（光学采集器及半导体采集器：121*133(101*129)，滑动采集器 FPC1080：64*218）。

[工作过程 Sequence]

③ 若指定 Image Type 无效，则返回 ERR_INVALID_PARAM 。

④ 利用指令应答包，将 HOST 待收到图像的大小发送至 HOST。

⑤ 根据 Image Type，利用应答数据包，将图像以 496bytes 单位分成并发送至 HOST。

[命令及响应 Command and Response]

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0022	
LEN	1	
DATA	1byte	Image Type（0：Full，1：Quarter）

PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0022	
LEN	6/2	
RET	Result Code	
DATA	2bytes	成功：图像的宽度 Full 全图像（242/202/128） Quarter 图像（121/101/64）
	2bytes	成功：图像的高度 Full 全图像(266/258/436) Quarter 图像（133/129/218）

成功时：Target 发送应答数据包至 HOST	
PREFIX	0x5AA5
SID	Source Device ID
DID	Destination Device ID
RCM	0x0022
LEN	4+ 图像数据长度
RET	ERR_SUCCESS
DATA	图像数据长度(2bytes) + 图像数据

…

继续发送应答数据包

表 4-8 CMD_UP_IMAGE 指令

注：

1. 调用该指令之前，必须先调用 CMD_GET_IMAGE 将指纹图像保存于 ImageBuffer 中。
2. 高分辨率模式（Full Mode）宽度*高度：242*266/202*258/128*436
3. 低分辨率模式（Quarter Mode）宽度*高度：121*133/101*129/64*218

实例 4.8-例 1：上传全分辨率的 202*258 传感器指纹图像

主机命令： 55 AA 00 00 22 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 22 01

模块响应： AA 55 01 00 22 00 06 00 00 00 CA 00 02 01 00 00 00 00 00 00 00 00 00 00 F5 01

注：全图像宽度=0xCA=202，全图像高度=0x102=258

模块响应数据包：图像数据大小为 202*258=52116 字节，分为 105*496 字节+1*36 字节

A5 5A 01 00 22 00 F4 01 00 00 F0 01 本帧的 496 字节数据 2 字节校验码

。。。共 105 个包含 496 字节图像数据的响应数据包

A5 5A 01 00 22 00 28 00 00 00 24 00 最后一帧 36 字节数据 2 字节校验码

最后 1 个包含 36 字节图像数据的响应数据包

实例 4.8-例 2：上传 1/4 图像分辨率的光学指纹图像

主机命令： 55 AA 00 00 22 00 01 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 23 01

模块响应： AA 55 01 00 22 00 06 00 00 00 65 00 81 00 00 00 00 00 00 00 00 00 00 00 0E 02

注：1/4 图像宽度=0x65=101，全图像高度=0x81=129

模块响应数据包：图像数据大小为 202*258/4=13029 字节，分为 26*496 字节+1*133 字节

A5 5A 01 00 22 00 F4 01 00 00 F0 01 本帧的 496 字节数据 2 字节校验码

。。。共 26 个包含 496 字节图像数据的响应数据包

A5 5A 01 00 22 00 89 00 00 00 85 00 最后一帧 133 字节数据 2 字节校验码

最后 1 个包含 133 字节图像数据的响应数据包

4.9 下载指纹图像到模块（CMD_DOWN_IMAGE 0x0023）

[功能 Function]

将从 Host 收到的图像数据保存于 ImageBuffer 中。

Host 以 496bytes 单位将图像发送至 Target 。这时，同时发送图像数据编号。

注：指纹图像要求：分辨率：500DPI，灰度：8 位灰度

像素大小：光学采集器：242*266；

按压式半导体采集器（如 FPC1011）：202*258；

滑动式半导体采集器（如 FPC1080）：128*436；

[工作过程 Sequence]

- 若图像高度或图像宽度不正确，则返回 ERR_INVALID_PARAM 。
- 利用应答包返回 ERR_SUCCESS 。
- 接收指令数据包将图像保存于 ImageBuffer 中。

[命令及响应 Command and Response]

指令包		
PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0023	
LEN	4	
DATA	2bytes	图像宽度：242/202/128
	2bytes	图像高度：266/258/436

PREFIX	0x55AA
SID	Source Device ID
DID	Destination Device ID
RCM	0x0023
LEN	2
RET	Result Code
DATA	0

指令数据包	
PREFIX	0xA55A
SID	Source Device ID
DID	Destination Device ID
CMD	0x0023
LEN	2 + 图像数据大小
DATA	图像数据编号(2bytes) + 图像数据

响应数据包	
PREFIX	0x5AA5

SID	Source Device ID
DID	Destination Device ID
RCM	0x0023
LEN	2
RET	Result Code
DATA	0

...

指令数据包	
PREFIX	0xA55A
SID	Source Device ID
DID	Destination Device ID
CMD	0x0023
LEN	2 + 图像数据大小
DATA	图像数据编号(2bytes) + 图像数据
响应数据包	
PREFIX	0x5AA5
SID	Source Device ID
DID	Destination Device ID
RCM	0x0023
LEN	2
RET	Result Code
DATA	0

表 4-9CMD_DOWN_IMAGE 指令

4.9 例子：下载指纹图像到 ImageBuffer 中

主机命令： 55 AA 00 00 23 00 04 00 CA 00 02 01 00 00 00 00 00 00 00 00 00 00 00 00 F3 01

模块响应： AA 55 01 00 23 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 25 01

命令数据包： 5A A5 00 00 23 00 F2 01 00 00 本块的 496 字节数据 2 字节校验码

响应数据包： A5 5A 01 00 23 00 02 00 00 00 25 01；数据应答包的长度因为没有数据是 12 个字节

。。。共 105 个包含 496 字节图像数据的命令数据包及响应数据包

命令数据包： 5A A5 00 00 23 00 26 00 69 00 包含 36 字节图像数据的最后数据块 2 字节校验码

响应数据包： A5 5A 01 00 23 00 02 00 00 00 25 01；数据应答包的长度因为没有数据是 12 个字节

4.10 控制采集器背光灯开/关（CMD_SLED_CTRL 0x0024）

[功能 Function]

控制光学采集器背光灯的开/关，或某些型号指纹模块的三色指示灯的显示状态。

[工作过程 Sequence]

不进行任何操作返回 ERR_SUCCESS 。

[命令及响应 Command and Response]

PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID
CMD	0x0024
LEN	2
DATA	LED 状态（1：开，0：关）
PREFIX	0x55AA
SID	Source Device ID
DID	Destination Device ID
RCM	0x0024
LEN	2
RET	ERR_SUCCESS
DATA	0

表 4-10 CMD_SLED_CTRL 指令

实例 4.10-打开背光灯（Backlight LED On）

主机命令： 55 AA 00 00 24 00 02 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 26 01

模块响应： AA 55 01 00 24 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 26 01

4.11 保存指纹模板数据到模块指纹库（CMD_STORE_CHAR 0x0040）

[功能 Function]

将暂存于指定 Ram Buffer 中的指纹模板保存于指定编号的模块指纹库中。

[工作过程 Sequence]

- 若指定 Template 编号无效，则返回错误码 ERR_INVALID_TMPL_NO 。
- 若指定 Ram Buffer 编号无效，则返回错误码 ERR_INVALID_BUFFER_ID 。
- 若 Duplication Check 设置为 OFF，则直接将指定 Ram Buffer 中的指纹模板数据注册于指定编号的指纹库中并返回其结果。
- 若 Duplication Check 设置为 ON，则将指定 Ram Buffer 中的 Template 和已注册的指纹库中的所有 Template 之间进行 1:N 比对。
  若存在比对成功的模板，说明该指纹已注册，则返回（RET）：ERR_DUPLICATION_ID，且 DATA 返回比对成功的 Template 编号。
  否则，将该模板注册于指定 Template 编号的指纹库中并返回其结果。

[命令及响应 Command and Response]

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0040	
LEN	4	
DATA	2bytes	Template 编号
	2bytes	Ram Buffer 编号
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0040	
LEN	LEN 长度跟结果码相关：2/或者 4 当 Result Code 为 ERR_DUPLICATION_ID 时：LEN=4 ； 否则：LEN=2	
RET	Result Code	
DATA	2bytes	数据跟结果码相关：0 或者重复的指纹编号 结果码为 ERR_DUPLICATION_ID 时为：重复的指纹编号； 否则为：0

表 4-11 CMD_STORE_CHAR 指令

实例 4.11-保存 RamBuffer0 中的模板数据到指定编号为 1 的模块数据库中：

主机命令包： 55 AA 00 00 40 00 04 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 44 01

模块响应包： AA 55 01 00 40 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 42 01

4.12 读取模块中的指纹并暂存在 RamBuffer 中（CMD_LOAD_CHAR 0x0041）

[功能 Function]

将指纹库中指定编号中的指纹模板（Template）取出并暂存于指定的 Ram Buffer 中。

[工作过程 Sequence]

- 若指定 Template 编号无效，则返回 ERR_INVALID_TMPL_NO 。
- 若指定 Template 编号中没有注册 Template，则返回错误码 ERR_TMPL_EMPY 。
- 若指定 Ram Buffer 编号无效，则返回错误码 ERR_INVALID_BUFFER_ID 。
- 将指定编号中的 Template 保存于指定 Ram Buffer 中并返回 ERR_SUCCESS 。

[命令及响应 Command and Response]

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0041	
LEN	4	
DATA	2bytes	Template 编号
	2bytes	Ram Buffer 编号
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0041	
LEN	2	
RET	Result Code	
DATA	0	

表 4-12 CMD_LOAD_CHAR 指令

实例 4.12-读取编号为 1 的模板数据暂存在 RamBuffer0 中：

主机命令包： 55 AA 00 00 41 00 04 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 45 01

模块响应包： AA 55 01 00 41 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 43 01

4.13 将暂存在 RamBuffer 中的指纹模板上传到主机（CMD_UP_CHAR 0x0042）

[功能 Function] 将指定 Ram Buffer 中的 Template 发送至 Host 。

[工作过程 Sequence]

①指定 Ram Buffer 编号无效，则返回 ERR_INVALID_BUFFER_ID 。
②利用指令应答包将 HOST 待接收的 Template 数据的大小发送至 HOST。
③利用应答数据包将指定编号中的 Template 数据发送至 HOST。

[命令及响应 Command and Response]

PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID
CMD	0x0042
LEN	2
DATA	Ram Buffer ID
PREFIX	0x55AA
SID	Source Device ID
DID	Destination Device ID
RCM	0x0042
LEN	4
RET	ERR_SUCCESS or ERR_FAIL
DATA	成功：下次数据应答包的数据长度 (Template Record Size + 2) 注：各版算法生成的指纹模板数据长度是不一样的，参见第三章的表 3-2 失败：错误码
成功时模块发送响应数据包	
PREFIX	0x5AA5
SID	Source Device ID
DID	Destination Device ID
RCM	0x0042
LEN	Template Record Size + 2 注：各版算法生成的指纹模板数据长度是不一样的，参见第三章的表 3-2
RET	ERR_SUCCESS
DATA	Template Record Data

表 4-13 CMD_UP_CHAR 指令

注：

1. 调用该指令之前，必须先调用 CMD_GENERATE, CMD_DOWN_CHAR, CMD_LOAD_CHAR 当中的一个指令，将 Template 保存于某个 Ram Buffer 中。
2. 各版算法生成的指纹模板数据长度是不一样的，参见第三章的表 3-2

实例 4.13-上传 RamBuffer0 中的模板数据到 HOST 的示例：

例一、通用算法固件（例如 SEONU，指纹模板数据为 498 字节）

主机命令包： 55 AA 00 00 42 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 43 01

模块响应包： AA 55 01 00 42 00 04 00 00 00 F2 01 00 00 00 00 00 00 00 00 00 00 00 00 39 02

响应数据包： A5 5A 01 00 42 00 F4 01 00 00 498 字节的本指纹模板数据 2 字节校验码

例二、160 新算法固件（例如 SEONU2，指纹模板数据为 1008 字节）

主机命令包： 55 AA 00 00 42 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 43 01

模块响应包： AA 55 01 00 42 00 04 00 00 00 F0 03 00 00 00 00 00 00 00 00 00 00 00 00 39 02

响应数据包： A5 5A 01 00 42 00 F2 03 00 00 1008 字节的本指纹模板数据 2 字节校验码

例三、192 新算法固件（例如 SEONU7，指纹模板数据为 448 字节）

主机命令包： 55 AA 00 00 42 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 43 01

模块响应包： AA 55 01 00 42 00 04 00 00 00 C0 01 00 00 00 00 00 00 00 00 00 00 00 00 07 02

响应数据包： A5 5A 01 00 42 00 C2 01 00 00 448 字节的本指纹模板数据 2 字节校验码

例四、非标算法(FPC1021 专用算法)固件（例如 SEONU5，指纹模板数据为 2024 字节）

主机命令包： 55 AA 00 00 42 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 43 01

模块响应包： AA 55 01 00 42 00 04 00 00 00 E8 07 00 00 00 00 00 00 00 00 00 00 00 00 35 02

2024 字节的模板数据分成五个数据包（四个 496 字节的数据包加一个 40 字节的数据包）如下：

响应数据包 1： A5 5A 01 00 42 00 F2 01 00 00 496 字节的本指纹模板数据 2 字节校验码

响应数据包 2： A5 5A 01 00 42 00 F2 01 00 00 496 字节的本指纹模板数据 2 字节校验码

响应数据包 3： A5 5A 01 00 42 00 F2 01 00 00 496 字节的本指纹模板数据 2 字节校验码

响应数据包 4： A5 5A 01 00 42 00 F2 01 00 00 496 字节的本指纹模板数据 2 字节校验码

响应数据包 5： A5 5A 01 00 42 00 2A 00 00 00 40 字节的本指纹模板数据 2 字节校验码

4.14 下载指纹模板数据到模块指定的 RamBuffer（CMD_DOWN_CHAR 0x0043）

[功能 Function]

从 Host 接收指纹模板数据（Template Data）并保存于指定的 Ram Buffer 中。

[工作过程 Sequence]

○Host 发送指令包，使 Target 进入数据(Ram Buffer + Template)接收等待状态。该指令包的 DATA 域中已设有下次发送的指令数据包的长度。

○ Target 检查接收到的指令包的准确性。

若不正确，则返回错误码并结束处理。

若待接收的数据大小不正确，则返回 ERR_INVALID_PARAM 。

若正确，则向 HOST 发送应答包表示模块已进入数据(Ram Buffer 编号 + Template 数据)接收等待状态，并进入数据(Ram Buffer 编号 + Template 数据)接收等待状态。

○ Host 收到 Target 已进入数据接收等待状态的应答包，则利用指令数据包指定编号的 RamBuffer 和 Template 数据并发送至 Target。

○Target 收到指令数据包后，若 Ram Buffer ID 无效，则返回 ERR_INVALID_BUFFER_ID 。

○检查收到的 Template 的 CheckSum 。若不正确，则返回 ERR_INVALID_TMPL_DATA 。

○将收到的 Template 保存于指定 Ram Buffer 中并返回 ERR_SUCCESS 。

[命令及响应 Command and Response]

指令包	
PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID
CMD	0x0043
LEN	2
DATA	2 + Template Record Size
PREFIX	0x55AA
SID	Source Device ID
DID	Destination Device ID
RCM	0x0043
LEN	4
RET	Result Code
DATA	0
指令数据包	
PREFIX	0xA55A
SID	Source Device ID
DID	Destination Device ID
CMD	0x0043
LEN	2 + Template 大小(498)

DATA	Ram Buffer 编号(2byte) + Template 数据
PREFIX	0x5AA5
SID	Source Device ID
DID	Destination Device ID
RCM	0x0043
LEN	4
RET	Result Code
DATA	0

表 4-14 CMD_DOWN_CHAR 指令

注：保存于 RamBuffer2 中的 Template，若调用（CMD_SEARCH, CMD_VERIFY, CMD_GENERATE, CMD_STORE_CHAR, CMD_DEL_CHAR, CMD_GET_EMPTY_ID, CMD_GET_STATUS, GET_BROKEN_ID, CMD_GETN_ENROLL_COUNT ）等指令后，则会被清掉。尽量避免使用 RamBuffer2 。

实例 4.14-下载指纹模板数据到模块的 RamBuffer0 例一、通用算法固件（例如 SEONU，指纹模板数据为 498 字节）

主机命令：55 AA 00 00 43 00 02 00 F4 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 39 02

命令响应：AA 55 01 00 43 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 45 01

数据块包：5A A5 00 00 43 00 F4 01 00 00 498 字节指纹模板数据 2 字节校验码

响应包： A5 5A 01 00 43 00 02 00 00 00 45 01 ；应答包的长度因无数据，所以 12 个字节。

例二、160 新算法固件（例如 SEONU2，指纹模板数据为 1008 字节）

主机命令：55 AA 00 00 43 00 02 00 F3 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 39 02

命令响应：AA 55 01 00 43 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 45 01

数据块包：5A A5 00 00 43 00 F2 03 00 00 1008 字节指纹模板数据 2 字节校验码

响应包： A5 5A 01 00 43 00 02 00 00 00 45 01 ；应答包的长度因无数据，所以 12 个字节。

例三、192 新算法固件（例如 SEONU7，指纹模板数据为 448 字节）

主机命令：55 AA 00 00 43 00 02 00 C2 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 07 02

命令响应：AA 55 01 00 43 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 45 01

数据块包：5A A5 00 00 43 00 C2 01 00 00 448 字节指纹模板数据 2 字节校验码

响应包： A5 5A 01 00 43 00 02 00 00 00 45 01 ；应答包的长度因无数据，所以 12 个字节。

例四、非标算法固件（例如 SEONU5），2024 字节模板数据分成五个数据包传输，特别说明如下：

主机命令：55 AA 00 00 43 00 02 00 EC 07 00 00 00 00 00 00 00 00 00 00 00 00 00 00 37 02

命令响应：AA 55 01 00 43 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 45 01

2024 字节数据分拆成五个数据包（4*496+40），在模板数据前增加 2 字节的数据块编号，传送如下：

数据块包 1：5A A5 00 00 43 00 F4 01 00 00 00 00 496 字节指纹模板数据 2 字节校验码

响应包 1： A5 5A 01 00 43 00 02 00 00 00 45 01 ；应答包的长度因无数据，所以 12 个字节。

数据块包 2：5A A5 00 00 43 00 F4 01 00 00 01 00 496 字节指纹模板数据 2 字节校验码

响应包 2： A5 5A 01 00 43 00 02 00 00 00 45 01 ；应答包的长度因无数据，所以 12 个字节。

数据块包 3：5A A5 00 00 43 00 F4 01 00 00 02 00 496 字节指纹模板数据 2 字节校验码

响应包 3： A5 5A 01 00 43 00 02 00 00 00 45 01 ；应答包的长度因无数据，所以 12 个字节。

数据块包 4：5A A5 00 00 43 00 F4 01 00 00 03 00 496 字节指纹模板数据 2 字节校验码

响应包 4： A5 5A 01 00 43 00 02 00 00 00 45 01 ；应答包的长度因无数据，所以 12 个字节。

数据块包 5：5A A5 00 00 43 00 F4 01 00 00 04 00 40 字节指纹模板数据 2 字节校验码

响应包 5： A5 5A 01 00 43 00 02 00 00 00 45 01 ；应答包的长度因无数据，所以 12 个字节。

4.15 删除指定编号范围内的指纹 （CMD_DEL_CHAR 0x0044）

[功能 Function]

删除指定编号范围(起始 Template 编号 ~ 结束 Template 编号)内全部已注册的 Template 。

[工作过程 Sequence]

① 若指定范围无效，则返回 ERR_INVALID_PARAM 。
② 若指定范围内没有注册 Template，则返回 ERR_TMPL_EMPTY 。
③ 删除指定范围内已注册的所有 Template 并返回其结果。

[命令及响应 Command and Response]

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0044	
LEN	4	
DATA	2bytes	起始 Template 编号
	2bytes	结束 Template 编号
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0044	
LEN	2	
RET	Result Code	
DATA	0	

表 4-15CMD_DEL_CHAR 指令

实例 4.15- 例 1：删除数据库中编号为 1-3000 的所有指纹

主机命令：55 AA 00 00 44 00 04 00 01 00 BB 08 00 00 00 00 00 00 00 00 00 00 00 00 0B 02
模块响应：AA 55 01 00 44 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 46 01

实例 4.15- 例 2：删除数据库中编号为 1-1700 的所有指纹

主机命令：55 AA 00 00 44 00 04 00 01 00 A4 06 00 00 00 00 00 00 00 00 00 00 00 00 F2 01
模块响应：AA 55 01 00 44 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 46 01

实例 4.15- 例 3：删除数据库中编号为 1-500 的所有指纹

主机命令：55 AA 00 00 44 00 04 00 01 00 F4 01 00 00 00 00 00 00 00 00 00 00 00 00 3D 02
模块响应：AA 55 01 00 44 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 46 01

实例 4.15- 例 4：删除数据库中编号为 1-200 的所有指纹

主机命令：55 AA 00 00 44 00 04 00 01 00 C8 00 00 00 00 00 00 00 00 00 00 00 00 00 10 02
模块响应：AA 55 01 00 44 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 46 01

4.16 获取指定编号范围内可注册的首个编号 （CMD_GET_EMPTY_ID 0x0045）

[功能 Function]

获取指定范围(起始 Template 编号 ~ 结束 Template 编号)内可注册(没有注册 Template 的)的第一个 Template 编号。

[工作过程 Sequence]

① 若指定范围无效，则返回 ERR_INVALID_PARAM 。

② 搜索指定范围内可注册的第一个 ID。
若存在，则返回其值。否则，返回 ERR_EMPTY_ID_NOEXIST 。

[命令及响应 Command and Response]

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0045	
LEN	4	
DATA	2bytes	起始 Template 编号
	2bytes	结束 Template 编号
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0045	
LEN	成功 ：4, 失败 ：2	
RET	Result Code	
DATA	2bytes	成功时：可注册的第一个 Template 编号

表 4-16CMD_GET_EMPTY_ID 指令

实例 4.16- 获取 1-2000 编号范围内（0x1-0x07D0）的首个可注册编号，结果该编号为 11

主机命令：55 AA 00 00 45 00 04 00 01 00 D0 07 00 00 00 00 00 00 00 00 00 00 00 00 20 02

模块响应：AA 55 01 00 45 00 04 00 00 00 0B 00 00 00 00 00 00 00 00 00 00 00 00 00 54 01

4.17 检查指定的编号是否已被注册（CMD_GET_STATUS 0x0046）

[功能 Function]

获取指定编号中的 Template 的注册状态。

[工作过程 Sequence]

若指定 Template 编号无效，则返回 ERR_INVALID_TMPL_NO 。
若指定编号中已有 Template 注册，则返回 1 。否则，返回 0 。

[命令及响应 Command and Response]

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0046	
LEN	2	
DATA	指定的 Template 编号	
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0046	
LEN	成功 ：3，失败 ：2	
RET	Result Code	
DATA	1byte	成功时：注册状态 （1：已注册，0：没有注册）

表 4-17 CMD_GET_STATUS 指令

实例 4.17- 例 1：获取 ID 编号=1 的注册状态，可注册

主机命令：55 AA 00 00 46 00 02 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 48 01
模块响应：AA 55 01 00 46 00 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 49 01

实例 4.17- 例 2：获取 ID 编号=1 的注册状态，已注册

主机命令：55 AA 00 00 46 00 02 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 48 01
模块响应：AA 55 01 00 46 00 03 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 4A 01

4.18 检查指定范围内的指纹库是否有数据损坏（CMD_GET_BROKEN_ID 0x0047）

[功能 Function]

检查指定范围(起始 Template 编号 ~ 结束 Template 编号)内的已注册模板的是否有损坏。

在 Flash 的 Write 操作中，有可能因突然断电等原因导致模板的损坏。

HOST 在任意时刻（例如，Target 的初始启动），利用该指令，检查模板的破损情况。

已破损的模板，需要删除重新注册。

[工作过程 Sequence]

○若指定范围无效，则返回 ERR_INVALID_PARAM 。

○检查指定范围内所有已注册的模板的破损情况。

若存在已破损模板，则返回已破损模板的个数和第一个已破损模板编号。

否则，模板个数和模板编号都为 0 。

[命令及响应 Command and Response]

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0047	
LEN	4	
DATA	2bytes	起始 Template 编号
	2bytes	结束 Template 编号
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0047	
LEN	成功 : 6, 失败 : 2	
RET	Result Code	
DATA	2byte	成功时：破损 Template 的个数
	2byte	成功时：第一个破损 Template 编号

表 4-18 CMD_GET_BROKEN_ID 指令

实例 4.18- 获取 1-200 范围内的指纹坏损的 ID 编号

主机命令：55 AA 00 00 47 00 04 00 01 00 C8 00 00 00 00 00 00 00 00 00 00 00 00 00 13 02

模块响应：AA 55 01 00 47 00 06 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 4D 01

4.19 获取编号范围内已注册的指纹总数（CMD_GET_ENROLL_COUNT 0x0048）

[功能 Function]

获取指定范围(起始 Template 编号 ~ 结束 Template 编号)内已注册的指纹总数。

[工作过程 Sequence]

○若指定范围无效，则返回 ERR_INVALID_PARAM 。

○返回模块中注册的指纹的个数。

[命令及响应 Command and Response]

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0048	
LEN	4	
DATA	2bytes	起始 Template 编号
	2bytes	结束 Template 编号
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0048	
LEN	成功 ：4，失败 ：2	
RET	Result Code	
DATA	2bytes	已注册的 Template 个数

表 4-19 CMD_GET_ENROLL_COUNT 指令

实例 4.19- 获取 1-200（0x0001~0x00C8）范围内的已注册用户总数，总数为 10（0x000A）

主机命令：55 AA 00 00 48 00 04 00 01 00 C8 00 00 00 00 00 00 00 00 00 00 00 00 00 14 02

模块响应：AA 55 01 00 48 00 04 00 00 00 0A 00 00 00 00 00 00 00 00 00 00 00 00 00 56 01

4.20 从暂存在 ImageBuffer 中的指纹图像产生模板（CMD_GENERATE 0x0060）

[功能 Function]

从 ImageBuffer 中的指纹图像产生指纹模板 Template 并保存于指定 Ram Buffer 中。

[工作过程 Sequence]

- 若指定 Ram Buffer 编号无效，则返回错误码 ERR_INVALID_BUFFER_ID 。
- 检查 ImageBuffer 中图像的正确性。若不正确，则返回 ERR_BAD_QUALITY 。
- 将生成的 Template 保存于指定 Ram Buffer 中并返回 ERR_SUCCESS 。

[命令及响应 Command and Response]

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0060	
LEN	2	
DATA	2bytes	Ram Buffer 编号
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0060	
LEN	2	
RET	Result Code	
DATA	0	

表 4-20 CMD_GENERATE 指令

注：

CMD_GENERATE 指令是从 ImageBuffer 生成 Template Data。因此，调用该指令之前，需要事先调用 CMD_GET_IMAGE 指令，将图像保存于 ImageBuffer 中。

实例 4.20- 例 1：从 ImageBuffer 中生成模板数据保存在 RamBuffer0 中

主机命令：55 AA 00 00 60 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 61 01

模块响应：AA 55 01 00 60 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 62 01

实例 4.20- 例 2：从 ImageBuffer 中生成模板数据保存在 RamBuffer1 中

主机命令：55 AA 00 00 60 00 02 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 62 01

模块响应：AA 55 01 00 60 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 62 01

实例 4.20- 例 3：从 ImageBuffer 中生成模板数据保存在 RamBuffer2 中

主机命令：55 AA 00 00 60 00 02 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 63 01

模块响应：AA 55 01 00 60 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 62 01

4.21 合成指纹模板数据用于入库（CMD_MERGE 0x0061）

[功能 Function]

将暂存在 Ram Buffer 中的模板合并生成模板数据并后保存于指定的 Ram Buffer 中。合成个数可为 2 或 3：

若为 2 ：则合成 Ram Buffer0 和 Ram Buffer1 的 Template 。

若为 3 ：则合成 Ram Buffer0、Ram Buffer1 和 Ram Buffer2 的 Template 。

[工作过程 Sequence]

- 若指定 Ram Buffer 编号无效，则返回错误码 ERR_INVALID_BUFFER_ID 。
- 若合成个数无效，则返回 ERR_GEN_COUNT 。
- 根据合成个数，合成 Template 并生成一个 Template 。若合成失败返回错误码。
- 将生成的 Template 保存于指定的 Ram Buffer 中并返回 ERR_SUCCESS 。

[命令及响应 Command and Response]

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0061	
LEN	3	
DATA	2bytes	Ram Buffer 编号
	1byte	合成个数(2/3)
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0061	
LEN	2	
RET	Result Code	
DATA	0	

表 4-21 CMD_MERGE 指令

实例 4.21- 将 RamBuffer 中 3 个暂存的指纹模板融合为 1 个指纹模板数据

主机命令：55 AA 00 00 61 00 03 00 00 00 03 00 00 00 00 00 00 00 00 00 00 00 00 00 66 01

模块响应：AA 55 01 00 61 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 63 01

4.22 指定 2 个 RamBuffer 之间的模板做比对 （CMD_MATCH 0x0062）

[功能 Function]

指定的两个 Ram Buffer 中的 Template 之间进行比对。

[工作过程 Sequence]

- 若指定 Ram Buffer 编号无效，则返回错误码 ERR_INVALID_BUFFER_ID 。
- 指定的 Ram Buffer 中的两个 Template 之间进行比对并返回其结果。
  若比对成功，则 RET 返回 ERR_SUCCESS 且 DATA 返回智能更新结果。
  否则，RET 返回 ERR_VERIFY 。

[命令及响应 Command and Response]

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0062	
LEN	4	
DATA	2bytes	待比对的第一个 Ram Buffer 编号
	2bytes	待比对的第二个 Ram Buffer 编号
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0062	
LEN	2	
RET	Result Code	
DATA	无数据	

表 4-22 CMD_MATCH 指令

实例 4.22- 将 RamBuffer0 与 RamBuffer1 中的指纹模板进行 1:1 比对，结果：匹配成功

主机命令：55 AA 00 00 62 00 04 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 66 01

模块响应：AA 55 01 00 62 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 64 01

4.23 指定编号范围内的 1：N 识别（CMD_SEARCH 0x0063）

[功能 Function]

指定 Ram Buffer 中的模板与指定搜索范围(起始 Template 编号 ~ 结束 Template 编号)内的所有已注册指纹 Template 之间进行 1:N 比对并返回其结果。

[工作过程 Sequence]

- 若指定 Ram Buffer 编号无效，则返回错误码 ERR_INVALID_BUFFER_ID 。
- 若指定搜索范围无效，则返回错误码 ERR_INVALID_BUFFER_ID 。
- 若没有已注册 Template ，则返回错误码 ERR_ALL_TMPL_EMPTY 。
- 指定 Ram Buffer 中的 Template 与已注册的所有模板之间进行比对并返回其结果。
  若搜索成功，则 RET 返回 ERR_SUCCESS 且在 DATA 域返回被搜索出的模板编号和智能更新结果。否则，RET 返回 ERR_IDENTIFY 。

[命令及响应 Command and Response]

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0063	
LEN	6	
2bytes	2bytes	Ram Buffer 编号 DATA
	待搜索的起始 Template 编号	
	2bytes	待搜索的结束 Template 编号

PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0063	
LEN	成功 ： 5, 失败 ： 2	
RET	Result Code	
DATA	3bytes	成功时： Template 编号(2bytes) + 智能更新结果(1byte)

表 4-23 CMD_SEARCH 指令

实例 4.23- 暂存在 RamBuffer0 中的模板在 1-200 编号范围内的指纹比对，比对结果 ID=8

主机命令：55 AA 00 00 63 00 06 00 00 00 01 00 C8 00 00 00 00 00 00 00 00 00 00 00 31 02

模块响应：AA 55 01 00 63 00 05 00 00 00 08 00 01 00 00 00 00 00 00 00 00 00 00 00 71 01

4.24 指定 RamBuffer 与指纹库中指定编号的模板比对（CMD_VERIFY 0x0064）

[功能 Function]

指定 Ram Buffer 中的模板与数据库中指定编号的模板之间进行 1:1 比对并返回其结果。

[工作过程 Sequence]

- 若指定 Template 编号无效，则返回错误码 ERR_INVALID_TMPL_NO 。
- 若指定 Ram Buffer 编号无效，则返回错误码 ERR_INVALID_BUFFER_ID 。
- 若不存在指定编号注册的 Template ，则返回错误码 ERR_TMPL_EMPTY。
- 指定 Ram Buffer 中的模板与指定编号中的模板之间进行比对并返回其结果。
  若比对成功：则 RET 返回 ERR_SUCCESS 且 DATA 返回 Template 编号和智能更新结果。否则：RET 返回 ERR_VERIFY 。

[命令及响应 Command and Response]

PREFIX	0xAA55	
SID	Source Device ID	
DID	Destination Device ID	
CMD	0x0064	
LEN	4	
DATA	2bytes	待比对的 Template 编号
	2bytes	Ram Buffer 编号
PREFIX	0x55AA	
SID	Source Device ID	
DID	Destination Device ID	
RCM	0x0064	
LEN	成功 ：5，失败 ：2	
RET	Result Code	
DATA	3bytes	成功时：Template 编号(2bytes) + 智能更新结果（1 ：已进行智能更新，0：没有更新）

表 4-24 CMD_VERIFY 指令

实例 4.24- 将 RamBuffer0 中的指纹模板与数据库中编号为 8 的指纹 1:1 验证

主机命令：55 AA 00 00 64 00 04 00 08 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 6F 01

模块响应：AA 55 01 00 64 00 05 00 00 00 08 00 01 00 00 00 00 00 00 00 00 00 00 00 72 01

4.25 设置模块序列号（CMD_SET_MODULE_SN 0x0008）

[功能 Function]

模块接收从主机发来的模块序列号（Module SN）并保存于模块中。Module SN 为 16 字节。

[工作过程 Sequence]

①Host 发送指令包，使得 Target 进入数据(Module SN)接收等待状态。
该指令包的 DATA 域中，已设置有下次发送的指令数据包的长度。

②Target 检测接收到的指令包的正确性：
若不正确：则返回错误码并结束处理。
若待接收数据的大小不正确：则返回 ERR_INVALID_PARAM。
若正确：则为了告知已进入数据(Module SN)接收等待状态向 HOST 发送应答包，并进入数据(Module SN)接收等待状态。

③Host 收到应答包后，在指令数据包中设置 Module SN 并发送至 Target。

④Target 收到指令数据包后，将 Module SN 设置于模块并返回其结果。

[命令及响应 Command and Response]

指令包	
PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID
CMD	0x0008
LEN	2
DATA	16(Module SN 数据块长度)
PREFIX	0x55AA
SID	Source Device ID
DID	Destination Device ID
RCM	0x0008
LEN	2
RET	Result Code
DATA	无
指令数据包	
PREFIX	0xA55A
SID	Source Device ID
DID	Destination Device ID
CMD	0x0008
LEN	16(Module SN 的数据块长度)
DATA	Module SN(16bytes)
PREFIX	0x5AA5
SID	Source Device ID

DID	Destination Device ID
RCM	0x0008
LEN	2
RET	Result Code
DATA	无

表 4-25 CMD_SET_MODULE_SN 指令

实例 4.25- 设置模块序列号为：IDWD2011-0123456

主机命令：55 AA 00 00 08 00 02 00 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 19 01

模块响应：AA 55 01 00 08 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0A 01

数据包： 5A A5 00 00 08 00 10 00 49 44 57 44 32 30 31 31 2D 30 31 32 33 34 35 36 95 04

结果响应：A5 5A 01 00 08 00 02 00 00 00 0A 01

4.26 读取模块序列号（CMD_GET_MODULE_SN 0x0009）

[功能 Function]

将保存于模块的 Module SN 发送至 Host。

[工作过程 Sequence]

○以指令应答包的形式，将 HOST 待接收的 Module SN 的大小指定为应答数据并应答。
○将保存于模块的 Module SN，利用应答数据包发送。

[命令及响应 Command and Response]

PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID
CMD	0x0009
LEN	0
DATA	无
PREFIX	0x55AA
SID	Source Device ID
DID	Destination Device ID
RCM	0x0009
LEN	4
RET	Result Code
DATA	成功：下一个数据应答包的数据长度(Module SN Size(16)) 失败：错误码
成功时的应答数据包	
PREFIX	0x5AA5
SID	Source Device ID
DID	Destination Device ID
RCM	0x0009
LEN	Module SN Size(16)
RET	ERR_SUCCESS
DATA	Module SN(16bytes)

表 4-26 CMD_GET_MODULE_SN 指令

实例 4.26- 读取模块序列号，得到的序列号为：IDWD2011-0123456

主机命令：55 AA 00 00 09 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 08 01

模块响应：AA 55 01 00 09 00 04 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00 00 00 1D 01

主机接收到的模块序列号数据包：

A5 5A 01 00 09 00 12 00 00 00 49 44 57 44 32 30 31 31 2D 30 31 32 33 34 35 36 99 04

4.27 获取已注册 ID 列表（CMD_GET_ENROLLED_ID_LIST 0x0049）

[功能 Function] 将注册于模块中的 ID 列表信息发送至 HOST。

其 ID 列表信息结构如下：

每个字节的每个位表示第 x(x = 字节号(从 0 开始) * 8 + 位号(从 0 开始))个编号的指纹注册状态。

若为 0，则表示没有注册。若为 1，则表示已注册。

例如；假设 ID 列表信息的第二个字节为 01000001(2 进制)，每个位的含义如下：

从右开始第 0 位(1)：8*2+0 = 16（第 16 编号中已注册指纹）

从右开始第 1 位(0)：8*2+1 = 17（第 17 编号中没注册指纹）

…

从右开始第 6 位(1)：8*2+6 = 22（第 22 编号中已注册指纹）

从右开始第 7 位(0)：8*2+7 = 23（第 23 编号中没注册指纹）

[工作 Sequence]

○以指令应答包的形式将 HOST 待接收的 ID 列表信息的大小设为应答数据发送应答。

○以应答数据包发送模块中已注册 ID 列表信息。

[命令及响应 Command and Response]

PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID
CMD	0x0049
LEN	0
DATA	无

PREFIX	0x55AA
SID	Source Device ID
DID	Destination Device ID
RCM	0x0049
LEN	4
RET	Result Code
DATA	成功：下一个应答数据包的数据长度(ID List Information Size) 失败：错误码

成功时的应答数据包	
PREFIX	0x5AA5
SID	Source Device ID
DID	Destination Device ID
RCM	0x0049
LEN	ID List Information Size
RET	ERR_SUCCESS
DATA	ID List Information

表 4-27 CMD_GET_ENROLLED_ID_LIST 指令

实例 4.27 - 获取 ID=1~490 已注册情况下的用户列表指令：

主机命令：55 AA 00 00 49 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 48 01

模块响应：AA 55 01 00 49 00 04 00 00 00 41 00 00 00 00 00 00 00 00 00 00 00 00 00 8E 01

命令数据包：

A5 5A 01 00 49 00 43 00 00 00 FE FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF 07 00 00 00 55 3E

User ID List 数据中：

第 0 个字节为：0xFE=1111 1110，则 ID=0 未注册，ID=1~7 已注册：

第 0（0*8+0）位为 0，ID=0 未注册

第 1（0*8+1）位为 1，ID=1 已注册

。。。

第 7（0*8+7）位为 1，ID=7 已注册

第 1 个字节为：0xFF=1111 1111，则 ID=8~15 都已注册：

第 8（1*8+0）位为 1，ID=8 已注册

第 9（1*8+1）位为 1，ID=9 已注册

。。。

第 15（1*8+7）位为 1，ID=15 已注册

。。。

第 61 个字节为：0x07=0000 0111，则：

第 488（61*8+0=488）位为 1，ID=488 已注册

第 489（61*8+1=489）位为 1，ID=489 已注册

第 490（61*8+2=490）位为 1，ID=490 已注册

第 491（61*8+3=491）位为 0，ID=491 未注册

。。。

第 495（61*8+7=495）位为 0，ID=495 未注册

。。。

第 62 个字节为：0x00=0000 0000，则：

第 496（62*8+0=496）位为 0，ID=496 未注册

第 497（62*8+1=497）位为 0，ID=497 未注册

第 498（62*8+2=498）位为 0，ID=498 未注册

第 499（62*8+3=497）位为 0，ID=499 未注册

第 500（62*8+4=500）位为 0，ID=500 未注册

。。。

4.28 进入休眠状态（CMD_ENTER_STANDBY_STATE 0x000C）

[功能 Function]

使模块进入休眠状态。

[工作 Sequence]

模块收到指令包之后，返回 ERR_SUCCESS 并进入休眠状态。

[命令及响应 Command and Response]

PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID
CMD	0x000C
LEN	0
DATA	无数据
PREFIX	0x55AA
SID	Source Device ID
DID	Destination Device ID
RCM	0x000C
LEN	成功: 2
RET	0
DATA	无数据

表 4-28 CMD_ENTER_STANDBY_STATE

实例 4.28- 使模块进入休眠状态

主机命令：55 AA 00 00 0C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0B 01

模块响应：AA 55 01 00 0C 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0E 01

注意：

1. 关断模块电源前建议先发送本指令（CMD_ENTER_STANDBY_STATE），使本模块先进入待机状态再关断模块供电。
2. 某些型号模块不支持休眠功能。

4.29 自动调整指纹传感器（CMD_ADJUST_SENSOR 0x0025）

[功能 Function]

自动调整采集器寄存器参数以便获取较佳效果的指纹图像

[工作 Sequence]

调节 sensor 并返回其结果。

[Command and Response]

PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID
CMD	0x0025
LEN	0
DATA	无数据
PREFIX	0x55AA
SID	Source Device ID
DID	Destination Device ID
RCM	0x0025
LEN	成功: 2
RET	0
DATA	无数据

表 4-29 CMD_ADJUST_SENSOR 指令

注：某些型号模块的传感器不支持此项操作，执行此指令也无效果。

实例 4.29- 自动调整指纹传感器

主机命令：55 AA 00 00 25 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 24 01

模块响应：AA 55 01 00 25 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 27 01

4.30 通讯错误返回（Incorrect Command）

[功能 Function]

因通讯错误、干扰造成的误码等原因，模块收到不正确指令的情况，向 HOST 发送该应答。

[响应 Command and Response]

PREFIX	0x55AA
SID	Source Device ID
DID	Destination Device ID
RCM	0x00FF
LEN	2
RET	ERR_SUCCESS
DATA	-

表 4-30 Incorrect Command 返回

4.31 注意事项

- CMD_GENERATE 指令是从 ImageBuffer 生成 Template Data。
  因此，调用该指令之前，需要事先调用 CMD_GET_IMAGE 指令，将图像保存于 ImageBuffer 中。
- 调用 CMD_VERIFY ， CMD_SEARCH, CMD_GENERATE, CMD_MERGE, CMD_MATCH 指令，则保存于 ImageBuffer 中的图像会被清掉。
- 保存于 Ram Buffer2 中的 Template ，调用 CMD_SEARCH, CMD_VERIFY, CMD_GENERATE, CMD_STORE_CHAR, CMD_DEL_CHAR, CMD_GET_EMPTY_ID, CMD_GET_STATUS, GET_BROKEN_ID, CMD_GETN_ENROLL_COUNT 指令，会被清掉。

  因此，除了注册之外，不要使用 Ram Buffer2 。

（五）响应（Response）及错误代码表（Error Code）

No	Response 及错误代码	值	说明
1	ERR_SUCCESS	0x00	指令处理成功。
2	ERR_FAIL	0x01	指令处理失败。
3	ERR_VERIFY	0x10	与指定编号中 Template 的 1:1 比对失败。
4	ERR_IDENTIFY	0x11	已进行 1:N 比对，但相同 Template 不存在。
5	ERR_TMPL_EMPTY	0x12	在指定编号中不存在已注册的 Template 。
6	ERR_TMPL_NOT_EMPTY	0x13	在指定编号中已存在 Template 。
7	ERR_ALL_TMPL_EMPTY	0x14	不存在已注册的 Template 。
8	ERR_EMPTY_ID_NOEXIST	0x15	不存在可注册的 Template ID 。
9	ERR_BROKEN_ID_NOEXIST	0x16	不存在已损坏的 Template 。
10	ERR_INVALID_TMPL_DATA	0x17	指定的 Template Data 无效。
11	ERR_DUPLICATION_ID	0x18	该指纹已注册。
12	ERR_BAD_QUALITY	0x19	指纹图像质量不好。
13	ERR_MERGE_FAIL	0x1A	Template 合成失败。
14	ERR_NOT_AUTHORIZED	0x1B	没有进行通讯密码确认。 注： ○若已设有通讯密码但没有调用 CMD_VERIFY_DEVPASS 进行确认，则除了 CMD_TEST_CONNECTION, CMD_VERIFY_DEVPASS 之外 的所有指令都返回该错误码。 ○若没有设置通讯密码，则可以不经过确认密码就可以使用所有指令。
15	ERR_MEMORY	0x1C	外部 Flash 烧写出错。
16	ERR_INVALID_TMPL_NO	0x1D	指定 Template 编号无效。
17	ERR_INVALID_PARAM	0x22	使用了不正确的参数。
18	ERR_GEN_COUNT	0x25	指纹合成个数无效。
19	ERR_TIME_OUT	0x23	在 TimeOut 时间内没有输入指纹。
20	ERR_INVALID_BUFFER_ID	0x26	Buffer ID 值不正确。
21	ERR_FP_NOT_DETECTED	0x28	采集器上没有指纹输入。
22	ERR_FP_CANCEL	0x41	指令被取消。

（六）指纹登记及识别流程

6.1 登记指纹流程（Enroll Process）

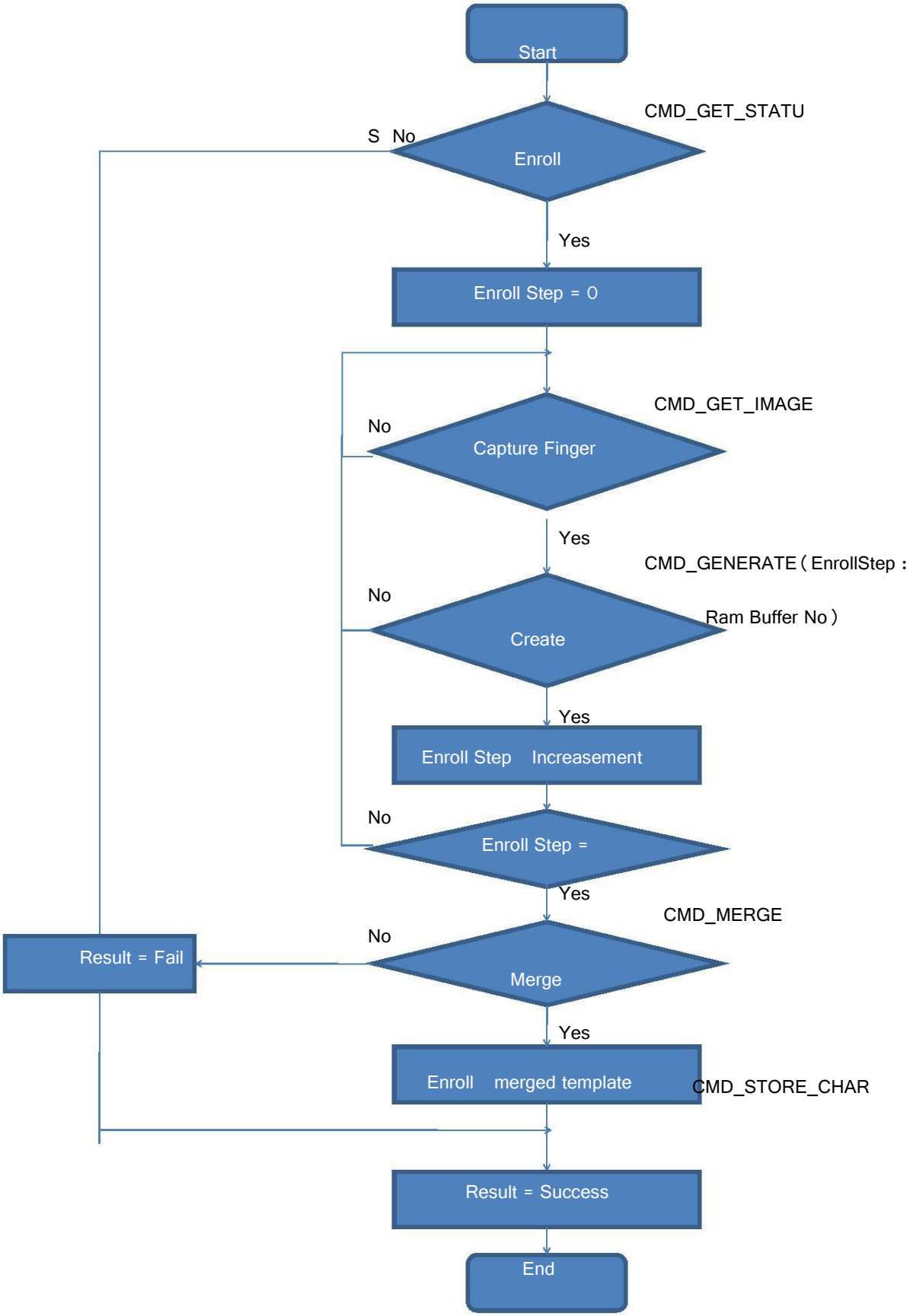

图 6-1：登记（注册）指纹的流程图

6.2 指纹验证及识别（Verify & Identify）流程

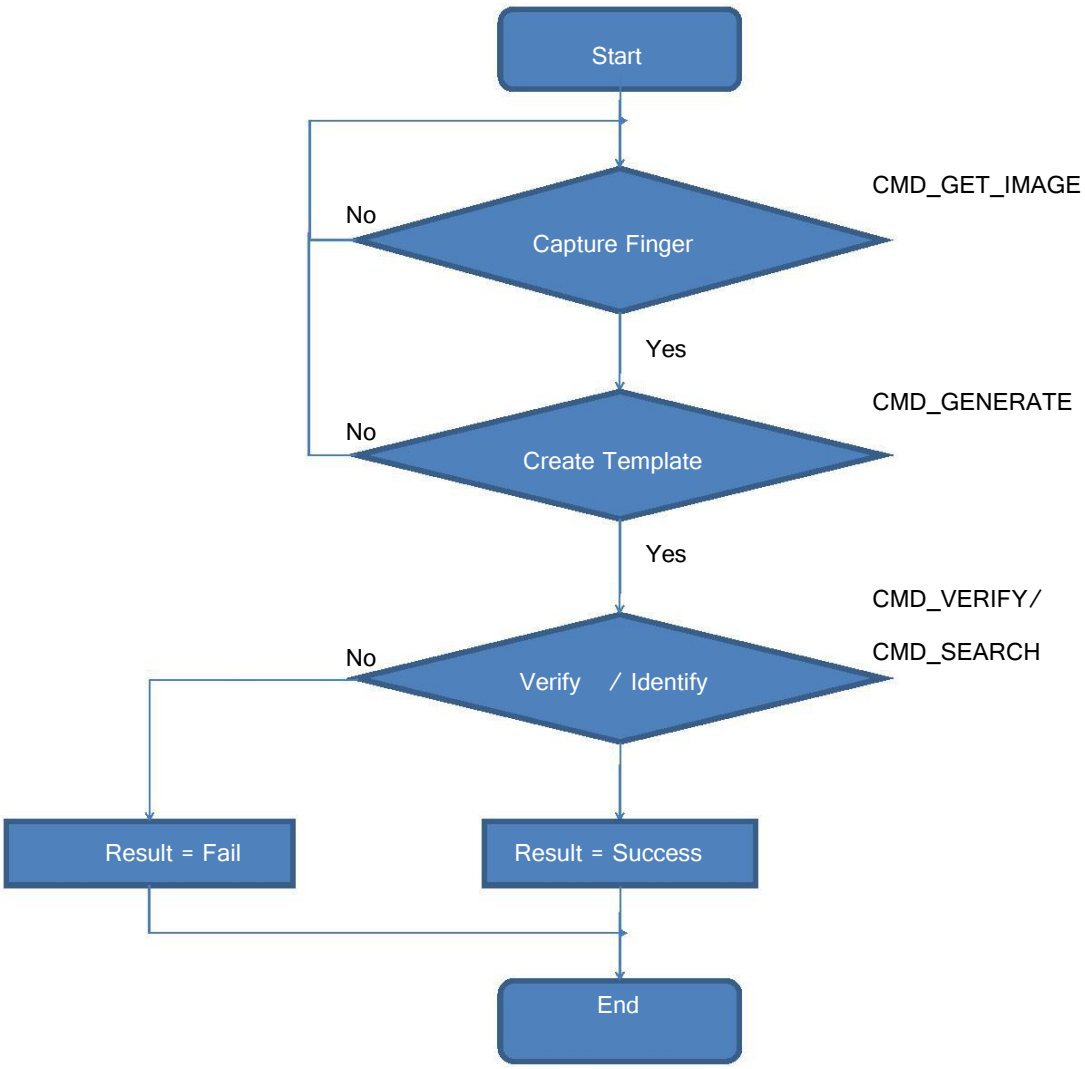

图 6-2：指纹验证及识别流程图