



**AFM60指纹模块
使用手册**

**Semiconductor Fingerprint Module
User's Guide**

(Ver 1.7)

深圳市凯迈生物识别技术有限公司

2023年3月30日

地址：深圳市南山区西丽街道西丽社区打石一路深圳国际创新谷1栋B座801

电话：0755-33581519 传真：0755-33581529

网址：www.szcamabio.cn



目录

一、概述	4
二、 模块硬件描述	5
2.1 硬件特性	5
2.2 外形及接口定义	5
2.2.1 外观尺寸图(分为带灯与不带灯两个外观版本)	5
2.2.2 AFM60接口图	6
2.2.3 接口使用注意事项	7
三、 模块的技术特性	8
3.1 技术特性	8
3.2 出厂设置值	8
四、 通讯协议	9
4.1 通讯处理过程	9
4.2 通讯包 PACKET 的分类	9
4.2.1 命令包 Command packet	9
4.2.2 响应包 Response packet	10
4.2.3 指令/响应的数据包 Data Packet	10
4.3 通讯包的帧结构	10
4.3.1 通讯包Packet识别代码	10
4.3.2 命令包 (Command packet) 的帧结构	10
4.3.3 响应包 (Response packet) 的帧结构	11
4.3.4 指令数据包的帧结构	11
4.3.5 响应数据包 (Response data packet) 的帧结构	11
五、 通讯命令COMMAND 的详细说明	12
5.1 指纹特征模板(TEMPLATE RECORD)的数据结构	12
5.2 命令列表 COMMAND LIST	13
5.3 COMMAND的详细说明	13
5.3.1 检查连接状态 (CMD_TEST_CONNECTION)	14
5.3.2 设置参数 (CMD_SET_PARAM)	14
5.3.3 获取参数 (CMD_GET_PARAM)	16
5.3.4 获取设备信息 (CMD_DEVICE_INFO)	17
5.3.5 进入IAP模式 (CMD_ENTER_IAP_MODE)	18
5.3.6 设置本模块的序列号 (CMD_SET_MODULE_SN)	19
5.3.7 获取本模块的序列号 (CMD_GET_MODULE_SN)	20
5.3.8 采集指纹图像 (CMD_GET_IMAGE)	21
5.3.9 检测是否有指纹 (CMD_FINGER_DETECT)	22
5.3.10 上传Image Buffer中的指纹图像至HOST (CMD_UP_IMAGE_CODE)	23
5.3.11 下载指纹图像至模块的Image Buffer (CMD_DOWN_IMAGE)	24



5.3.12 保存Ram Buffer中的指纹模板数据 (CMD_STORE_CHAR)26

5.3.13 从本模块数据库中读取指定编号模板并保存在Ram Buffer (CMD_LOAD_CHAR) 27

5.3.14 上传指定的Ram Buffer中的模板数据到HOST (CMD_UP_CHAR) 28

5.3.15 从HOST下传指纹模板到模块指定的Ram Buffer中 (CMD_DOWN_CHAR) 29

5.3.16 删除指定编号范围内的指纹模板 (CMD_DEL_CHAR) 31

5.3.17 获取指定编号范围内可注册的首个编号 (CMD_GET_EMPTY_ID) 31

5.3.18 获取指定编号的注册状态 (CMD_GET_STATUS) 32

5.3.19 检测指定编号范围内的指纹模板数据是否坏损 (CMD_GET_BROKEN_ID) 33

5.3.20 获取指定编号范围内已注册的指纹总数 (CMD_GET_ENROLL_COUNT)34

5.3.21 获取已注册 ID列表 (CMD_GET_ENROLLED_ID_LIST) 35

5.3.22 从ImageBuffer生成并暂存模板在指定的RamBuffer中 (CMD_GENERATE) 36

5.3.23 将RamBuffer中的2个或3个临时模板融合成1个模板数据 (CMD_MERGE)37

5.3.24 指定2个RamBuffer中的模板之间比对 (CMD_MATCH)38

5.3.25 指定编号范围的1: N识别 (CMD_SEARCH) 38

5.3.26 指定编号的1:1比对指纹 (CMD_VERIFY)39

5.3.27 通讯错误 (Incorrect Command) 40

5.4 注意事项 40

六、 附录 41

6.1 响应 (RESPONSE) 及错误代码表 41

6.2 登记及比对流程图 41

6.2.1 注册流程 ENROLL 41

6.2.2 验证及识别流程 Verify & Identify.....43



一、概述

AFM60为指纹采集和单片指纹处理器二合一设计的指纹识别二次开发模块，具有体积小、功耗低、接口简单、可靠性高、指纹模板小（496字节）、大容量指纹识别快（500枚指纹识别响应时间小于1秒）等优点，可以非常方便将其嵌入用户系统，组成满足客户需求的指纹识别产品。

尤其具备自学习功能，指纹识别过程中，最新提取的指纹特征值识别成功后将该特征值融合到指纹数据库中，使用户在使用过程中越用越好用。

AFM60模块通讯接口为UART，本模块作为从设备，由主设备发送相关命令对其进行控制。

该模块具有可调节的安全等级功能、指纹特征数据的读/写功能和 1:N 识别及 1:1 验证功能等。

AFM60模块还采用了国内最先进的人体触摸激活系统技术，当手指接触到指纹采集器窗口时，系统自动激活并采集比对指纹。让您的产品更智能，您不再为没有空间放置机械按键而苦恼（早期都采用按键做系统激活功能），与此同时还提高了您产品整体的美观及整洁度。

主要功能包括如下：

- 通过 UART 收发指令
- 在PCB上有预留USB2.0接口测试点，可通过USB接口烧录、升级模组程序
- 最小存储条件下实现指纹数据的登记及比对：指纹模板为496字节
- 半导体传感器芯片
- 1:N识别 及 1:1验证功能
- 具有循环采集比对功能（Identify Free）：主机HOST发送该指令后模块一直处于采集指纹比对并将比对结果返回HOST
- 指纹特征数据的读/写功能
- 可适当调节的安全等级
- 采集指纹并提取特征值上传，下载指纹特征值与当前采集的指纹比对功能
- （该功能特别适合于将指纹存储在接触式IC卡或非接触式IC卡中以验证是否持卡人本人）
- 可设置的波特率
- 可设置的设备编号
- 可以设置本模块访问密码（14个数字字符组成的密码），使本模块处于口令保护模式，极大提高了通讯的安全性
- 自动感应上电激活系统

应用领域：

- ◆ 指纹门禁控制
- ◆ 指纹签到、指纹考勤机等
- ◆ 指纹锁、指纹保险柜等
- ◆ 指纹POS终端机等手持设备应用



二、模块硬件描述

2.1 硬件特性

No	项 目	内 容
1	型号	AFM60
2	CPU	168MHz RISC 处理器
3	Sensor	FPC1021半导体指纹传感器，具备超高图像质量
4	传感器尺寸	12*12mm
5	传感器分辨率	160X160 (pixel)
6	输入电压	DC3.3V±10%
7	工作电流	小于50mA
8	录入次数	3次
9	识别角度	支持360度
10	通讯方式	3.3V UART 或 5.0V USB

表2-1 硬件特性

2.2 外形及接口定义

2.2.1 外观尺寸图(分为带灯与不带灯两个外观版本)

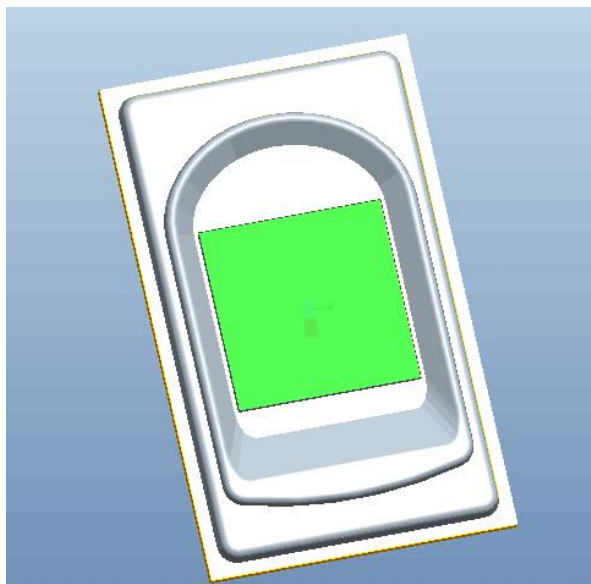


图2-1 AFM60 3D外观图

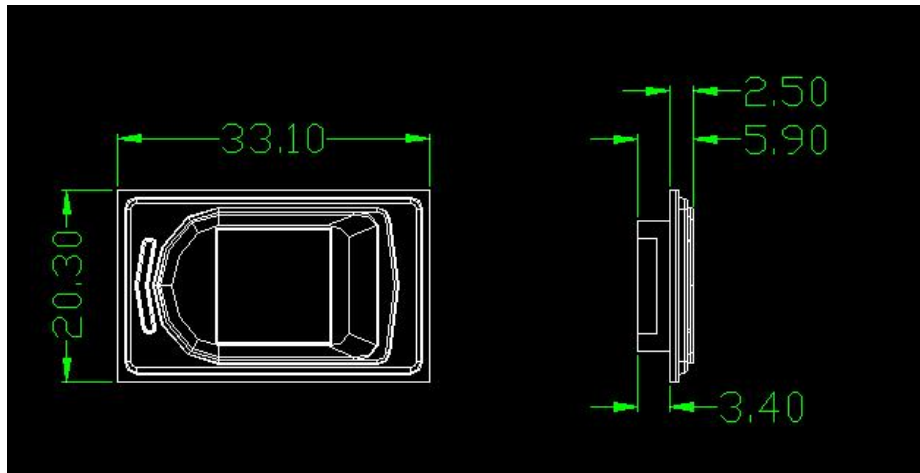
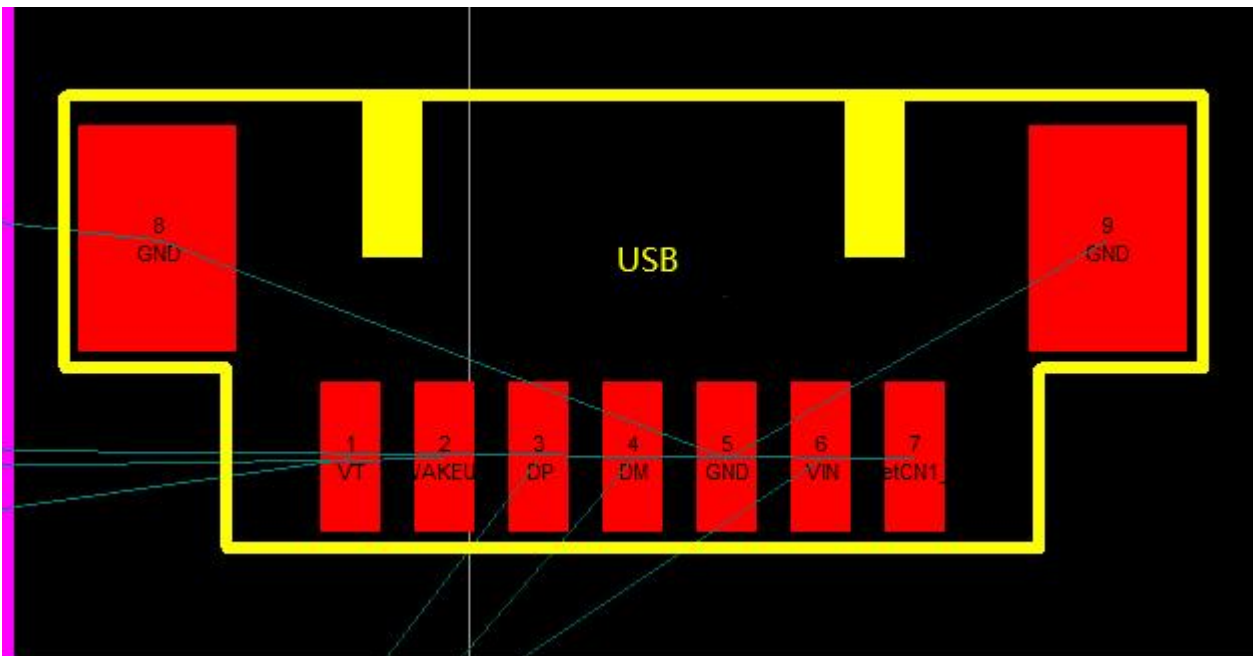
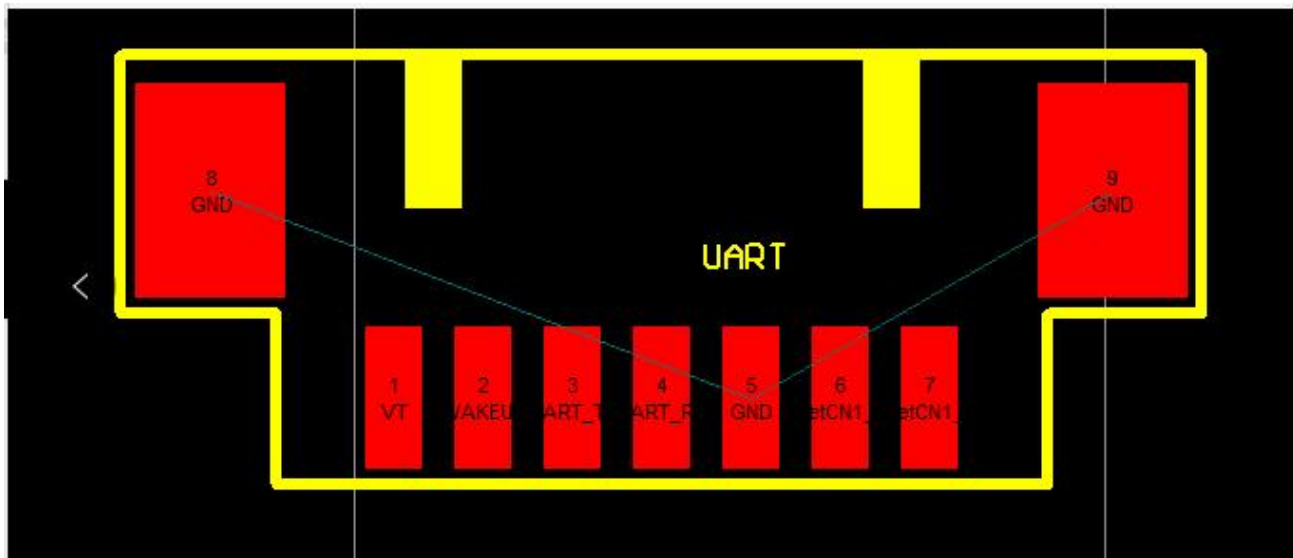


图2-3 AFM60尺寸图

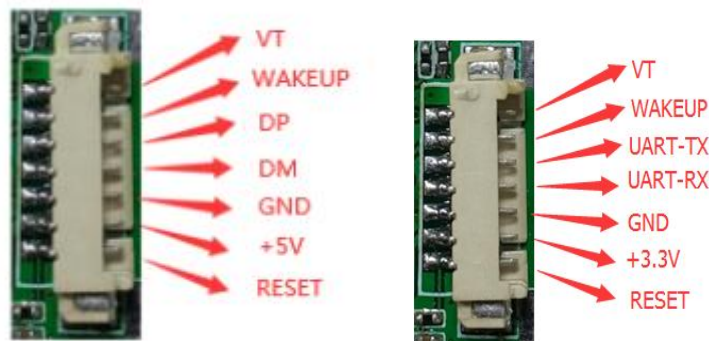
2.2.2 AFM60接口图



USB接口



UART接口



实物接口（左USB/右UART）

2.2.3 接口使用注意事项

1. AFM60模块与外部的控制接口为7P*1.25毫米插座，各管脚定义如上图；
2. 触摸唤醒功能需1脚单独供电（此时6脚需断开），电压为DC3.3±0.3V；
3. 第2脚为触摸唤醒，当模块上有手指触摸时，其输出高电平；手指拿开后，恢复为低电平；
4. 第6脚为模块指纹供电（若使用触摸唤醒功能必须开关控制6脚，以便灵活操控），USB版本的工作电压为5V，UART版本的工作电压为3.3V，工作电流为40mA。
5. 模块复位脚第7脚，用于对模块进行复位控制：低电平复位（复位时间不低于5毫秒），高电平正常工作；
6. 第3脚为模块串口发射脚，与外部串口接收脚连接；第4脚为模块串口接收脚，与外部串口发射脚连接；
7. USB测试点需采用+5V供电，USB-DM为数据线负极，USB-DP为数据线正极，GND接地。



三、模块的技术特性

3.1 技术特性

项目	描述	备注
CPU	168MHz主频、1M Bits Flash Memory、128KB SRAM	
指纹采集器	FPC1021半导体指纹传感器，具备超高图像质量	
指纹登记容量	500 枚	
认假率 (FAR)	< 0.001 % (安全等级Security Level 为 3)	
拒真率 (FRR)	< 0.1 % (安全等级Security Level 为 3)	
比对方式	1:N 及 1:1 比对	
安全等级设置	Level 1 至Level 5可设置，出厂设置值：Level 3	
处理速度	指纹特征提取时间< 0.45 秒	
	1:N 比对时间 (500 枚满注册时) < 1.5 秒	
通讯接口	UART, 3.3V-TTL电平/USB, 5V输入	
串行通讯参数	无奇偶校验Parity = NONE, 1个起始位Start Bit = 1 1个停止位Stop Bit = 1 无流控Flow Control = NONE 波特率为9600、19200、38400、57600、115200 (bps) 出厂设置：115200bps	
复位方式	外部复位	
工作电压	5V/3.3V	
工作电流	小于50mA	
工作环境	工作温度：-10℃ - 60℃ 相对湿度：20% - 80%	

表3-1 AFM60模块技术特性

3.2 出厂设置值

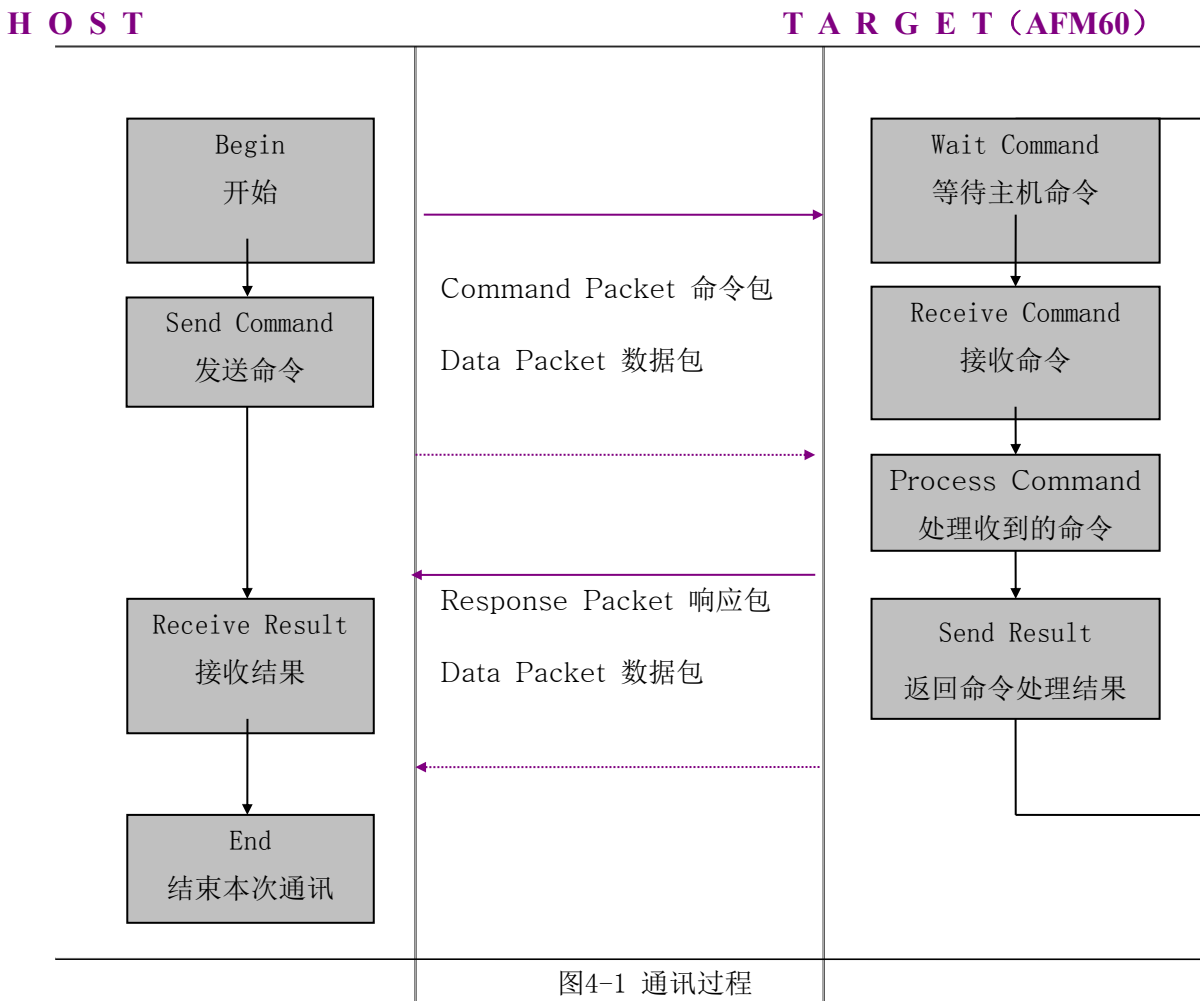
项目	初始值
安全等级 Security Level (1~5)	3级
指纹采集超时时间 Finger Print Time Out (0~60S)	0 秒 (无超时)
波特率 BaudRate (9600, 19200, 38400, 57600, 115200BPS)	115200 bps
指纹重复检查 Duplication Check (ON/OFF)	ON



表3-2 出厂设置值

四、通讯协议

4.1 通讯处理过程



注：

通讯过程除FP Cancel指令之外，其它指令的发送、接收必须要遵循一发一收的原则。
Host在没有收到应答时，请不要向 TARGET 发送指令。

4.2 通讯包 Packet 的分类

4.2.1 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。

4.2.2 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 。

4.2.3 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16byte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将指令数据包Data Packet的长度告知模块Target。
- 指令参数或相应数据包的最大长度为 500byte 。

4.3 通讯包的帧结构

4.3.1 通讯包Packet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4-1:

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Data Packet	0xA55A
响应数据包Response Data Packet	0x5AA5

表4-1 Packet识别代码

4.3.2 命令包 (Command packet) 的帧结构

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

表4-2 Command packet 的结构

OFFSET	FIELD	TYPE	SIZE	DESCRIPTION
0	PREFIX	WORD	2byte	Packet Identify code
2	SID	BYTE	1byte	Soruce Device ID
3	DID	BYTE	1byte	Destination Device ID
4	CMD	WORD	2byte	Command Code
6	LEN	WORD	2byte (=n, n < 16)	Length of DATA
8	DATA	Byte Array	16byte	Command Parameter (实际数据为 n byte)



24	CKS	WORD	2byte	Check Sum(PREFIX ~ DATA 数据算术和的低2byte)
----	-----	------	-------	---------------------------------------

4.3.3 响应包 (Response packet) 的帧结构

PREFIX		SID	DID	RCM		LEN		RET		DATA				CKS	
0x55	0xAA			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4-3 Response packet的结构

OFFSET	FIELD	TYPE	SIZE	DESCRIPTION
0	PREFIX	WORD	2bytes	Packet Identify code
2	SID	BYTE	1byte	Source Device ID
3	DID	BYTE	1byte	Destination Device ID
4	RCM	WORD	2bytes	Response Code
6	LEN	WORD	2bytes(=n, n < 16)	Length of RET and DATA
8	RET	WORD	2bytes	Result Code(0 :成功, 1 :失败)
10	DATA	Byte Array	14bytes	Response Data(实际数据为 n-2 byte)
24	CKS	WORD	2bytes	Check Sum(PREFIX ~ DATA 数据算术和的低2byte)

4.3.4 指令数据包的帧结构

PREFIX		SID	Host ID	CMD		LEN		DATA				CKS	
0x5A	0xA5			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4-4 指令数据packet的结构表

OFFSET	FIELD	TYPE	SIZE	DESCRIPTION
0	PREFIX	WORD	2byte	Packet Identify code
2	SID	BYTE	1byte	Source Device ID
3	DID	BYTE	1byte	Destination Device ID
4	CMD	WORD	2byte	Command Code
6	LEN	WORD	2byte(=n, n < 500)	Length of DATA
8	DATA	Byte Array	nbyte	Command parameter
8+n	CKS	WORD	2byte	Check Sum(PREFIX ~ DATA 数据算术和的下位2byte)

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 应在确认模块Target 的指令数据 packet 接收等待状态后传输指令数据包。

4.3.5 响应数据包 (Response data packet) 的帧结构



PREFIX		SID	DID	RCM		LEN		RET		DATA				CKS	
0xA5	0x5A			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4-5 Response data packet的结构

OFFSET	FIELD	TYPE	SIZE	DESCRIPTION
0	PREFIX	WORD	2byte	Packet Identify code
2	SID	BYTE	1byte	Soruce Device ID
3	DID	BYTE	1byte	Destination Device ID
4	CMD	WORD	2byte	Response Code
6	LEN	WORD	2byte(=n, n <500)	Length of result data (RET + DATA)
8	RET	WORD	2byte	Result code(0 : 成功, 1 : 失败)
10	DATA	Byte Array	n-2 byte	Response data
8+n	CKS	WORD	2byte	Check Sum (PREFIX ~ DATA 数据算术和的下位2byte)

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

五、通讯命令Command 的详细说明

5.1 指纹特征模板(Template Record)的数据结构

Template Data	Checksum
496 byte	2 byte
Template Data	对模板数据Template Data依次按字节进行算数和运算， 取计算结果的最低2 byte

表5-1 Template Record的结构

注：每个指纹特征模板数据为498字节：Template Data(496Bytes)+Checksum(2Bytes)

No	Command Name	Code	Function
1	CMD_TEST_CONNECTION	0x0001	进行与设备的通讯测试
2	CMD_SET_PARAM	0x0002	设置设备参数(Device ID, Security Level, BPS, Duplication Check, Au to Learn)
3	CMD_GET_PARAM	0x0003	获取设备参数(Device ID, Security Level, BPS, Duplication Check, Au to Learn)
4	CMD_GET_DEVICE_INFO	0x0004	获取设备信息
5	CMD_ENTER_IAP_MODE	0x0005	将设备设置为 IAP状态



6	CMD_SET_MODULE_SN	0x0008	从 Host接收模块序列号 (Module SN) 并保存于模块中
7	CMD_GET_MODULE_SN	0x0009	将本模块的序列号 (Module SN) 发送至 Host
8	CMD_ENTER_STDBY	0X000C	使模块进入待机状态
9	CMD_GET_IMAGE	0x0020	从采集器采集指纹图像并保存于 ImageBuffer 中
10	CMD_FINGER_DETECT	0x0021	检测指纹输入状态
11	CMD_UP_IMAGE	0x0022	将保存于 ImageBuffer 中的指纹图像上传至HOST
12	CMD_DOWN_IMAGE	0x0023	在 ImageBuffer 中下载指纹图像
13	CMD_STORE_CHAR	0x0040	将保存于指定 Ram Buffer 中的 Template , 注册于指定编号中
14	CMD_LOAD_CHAR	0x0041	将注册于指定编号中的 Template , 保存于指定 Ram Buffer 中
15	CMD_UP_CHAR	0x0042	将保存于指定 Ram Buffer 中的 Template 上传至HOST
16	CMD_DOWN_CHAR	0x0043	在指定 Ram Buffer 中下载 Template

5.2 命令列表 Command List

17	CMD_DEL_CHAR	0x0044	删除指定范围内的 Template
18	CMD_GET_EMPTY_ID	0x0045	获取指定范围内可注册的 (没有注册的) 第一个模板编号
19	CMD_GET_STATUS	0x0046	获取指定编号的模板注册状态
20	CMD_GET_BROKEN_ID	0x0047	检查指定范围内的所有模板的损坏情况
21	CMD_GET_ENROLL_COUNT	0x0048	获取指定范围内已注册的模板个数
22	CMD_GET_ID_LIST	0X0049	获取已注册ID列表
23	CMD_GENERATE	0x0060	从保存于 ImageBuffer 中的指纹图像生产模板并保存于指定Ram Buffer r 中
24	CMD_MERGE	0x0061	将保存于 Ram Buffer 中的两个或三个模板合成并生成一个模板
25	CMD_MATCH	0x0062	指定 Ram Buffer 中的两个模板之间进行 1:1 比对
26	CMD_SEARCH	0x0063	指定 Ram Buffer 中的模板与指定范围内的所有模板之间进行 1:N 比 对
27	CMD_VERIFY	0x0064	指定 Ram Buffer 中的模板与指定编号中的模板之间进行 1:1比对

表5-2 Command列表

5.3 Command的详细说明

模块中含有指令通讯用的 ImageBuffer 和 Ram Buffer 。

ImageBuffer 用于保存图像。

Ram Buffer 用于保存模板。



模块共有三个 Ram Buffer，叫 Ram Buffer0，Ram Buffer1 和 Ram Buffer2 。
断电情况下，ImageBuffer 和 Ram Buffer 中的数据会丢失。

5.3.1 检查连接状态 (CMD_TEST_CONNECTION)

- [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，则可认为与目标模块的连接不正常、或工作不正常、或波特率的设置有误。

- [工作过程 Sequence]

连接正常，则返回 ERR_SUCCESS 。

- [命令及响应 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
DATA	无数据

表 5-1CMD_TEST_CONNECTION指令

HOST命令: 55 AA 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01
Target响应: AA 55 01 00 01 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 03 01

5.3.2 设置参数 (CMD_SET_PARAM)

- [功能 Function]

根据指定 Parameter Type，设置设备参数 (Device ID, Security Level, Baudrate, Duplication Check, Auto Learn) 并返回其结果。

- [工作 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。



● [命令及响应 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	
DATA	无数据	

表5-2 CMD_SET_PARAM指令

● [参数类型 Parameter Type]

Parameter Type	描述 Description		
0	表示设备编号 (Device ID) : 可设置 1 ~ 255		
1	表示安全等级 (Security Level) : 可设置 1 ~ 5 , 出厂默认值为3 Security Level 对应的识别率:		
	Security Level	识别率	
	Level 1	认假率 FAR (False Acceptance Rate)	0.01%
		拒真率 FRR (False Rejection Rate)	0.005%
	Level 2	认假率 FAR (False Acceptance Rate)	0.003%
		拒真率 FRR (False Rejection Rate)	0.01%
	Level 3	认假率 FAR (False Acceptance Rate)	0.001 %
		拒真率 FRR (False Rejection Rate)	0.1 %
	Level 4	认假率 FAR (False Acceptance Rate)	0.0003%
		拒真率 FRR (False Rejection Rate)	0.5%
	Level 5	认假率 FAR (False Acceptance Rate)	0.0001%
		拒真率 FRR (False Rejection Rate)	1%



2	表示设置指纹重复检查 (Duplication Check) : 可设置0/1 (禁用/允许) 若为 1 : 则处理 CMD_STORE_CHAR 指令时进行重复检测。 若为 0 : 则不进行重复检测。
3	表示设置波特率 (Baudrate) : 可设置值为 1 ~ 8 1:9600bps; 2:19200bps; 3:38400bps; 4:57600bps; 5:115200bps; 6:230400bps; 7:460800bps, 8:921600bps
4	表示设置自学习功能 (Auto Learn) : 可设置0/1 (禁止/允许) 若为 1 : 则处理 CMD_SEARCH, CMD_VERIFY 指令时进行智能更新指纹特征数据 若为 0 : 则不进行智能更新

例子: 设置波特率为921600BPS

HOST命令: 55 AA 00 00 02 00 05 00 03 08 00 00 00 00 00 00 00 00 00 00 00 00 11 01
Target响应: AA 55 01 00 02 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 04 01

5.3.3 获取参数 (CMD_GET_PARAM)

- [功能 Function]

根据指定参数类型 (Parameter Type), 获取设备如下参数:

Device ID, Security Level, Baudrate, Duplication Check, Auto Learn

有关参数类型码 (Parameter Type), 请参考上述 CMD_SET_PARAM。

- [工作过程 Sequence]

若指定 Parameter Type 无效, 则返回ERR_INVALID_PARAM。

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	
DATA	4bytes	成功时: Parameter Value

表 5-3CMD_GET_PARAM 指令

例子：读取当前安全等级参数

Host命令： 55 AA 00 00 03 00 01 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 04 01
 Target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

5.3.4 获取设备信息 (CMD_DEVICE_INFO)

● [功能 Function]

获取 Target 的 Device Information ， 本设备固件名及版本号， 固定为 “ID_SEONU_ID809_FPC 1021_Inner_uo_Vx.x ” 。 x.x 表示固件版本号 (F/W Version) 。

● [工作过程 Sequence]

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。

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5-4 CMD_DEVICE_INFO指令

Host命令: 55 AA 00 00 04 00 03 01

Target应答: AA 55 01 00 04 00 04 00 00 00 31 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 39 01

响应数据包: A5 5A 01 00 04 00 33 00 00 00 49 44 5F 53 45 4F 4E 55 5F 49 44 38 30 39 5F 46 50
 43 31 30 3231 43 4D 5F 41 46 4D 36 30 5F 49 6E 6E 65 72 28 35 30 30 66 70 29 20 56 31 2E 30
 00 7B 0E 蓝色数据为设备信息“ID_SEONU_ID809_FPC1021_Inner_uo_Vx.x”的ASCII码

5.3.5 进入IAP模式 (CMD_ENTER_IAP_MODE)

- [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5-5 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。

5.3.6 设置本模块的序列号 (CMD_SET_MODULE_SN)

- [功能 Function]

从 Host接收 Module SN 并保存于模块中。

- [工作 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 Size)
PREFIX	0x55AA
SID	Source Device ID
DID	Destination Device ID
RCM	0x0008
LEN	2
RET	Result Code
DATA	无



数据包	
PREFIX	0xA55A
SID	Source Device ID
DID	Destination Device ID
CMD	0x0008
LEN	16 (Module SN Size)
DATA	Module SN (16bytes)
PREFIX	0x5AA5
SID	Source Device ID
DID	Destination Device ID
RCM	0x0008
LEN	2
RET	Result Code
DATA	无

表5-6 CMD_SET_MODULE_SN指令

5.3.7 获取本模块的序列号 (CMD_GET_MODULE_SN)

- [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]

指令包	
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	
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5-7 CMD_GET_MODULE_SN指令

5.3.8 采集指纹图像 (CMD_GET_IMAGE)

- [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。

- [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5-8 CMD_IMAGE指令

例子1：发送采集指纹图像后模块未检测到手指的命令及响应响应



Host命令: 55 AA 00 00 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 1F 01
 Target响应: AA 55 01 00 20 00 02 00 28 00 00 00 00 00 00 00 00 00 00 00 4A 01

例子2: 发送采集指纹图像后模块检测到手指的命令及响应响应

Host命令: 55 AA 00 00 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 1F 01
 Target响应: AA 55 01 00 20 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 22 01

5.3.9 检测是否有指纹 (CMD_FINGER_DETECT)

● [功能 Function]

检查收到本指的时刻指纹输入状态并返回其结果。

● [工作过程 Sequence]

返回收到该指令时刻传感器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, 设备 : 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5-9 CMD_FINGER_DETECT指令

例子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 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 24 01

例子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 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 25 01



5.3.10 上传Image Buffer中的指纹图像至HOST (CMD_UP_IMAGE_CODE)

● [功能 Function]

根据指定的Image Type (图像类型规定了是全分辨率指纹图像还是低分辨率指纹图像) , 将保存于 ImageBuffer 中的图像发送至 Host 。

若 Image Type 为 0 , 则为Full Mode, 发送全图 (242*266)

若为 1 , 则为Quarter Mode, 发送1/4图像 (4个点取1个点) 。

● [工作过程 Sequence]

若指定 Image Type无效, 则返回ERR_INVALID_PARAM 。

利用指令应答包, 将HOST待收到图像的大小发送至HOST。

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
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:242;Quarter:121)
	2bytes	成功: 图像的高度 (Full:266;Quarter:133)
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	

表5-10 CMD_UP_IMAGE指令

Target持续发送应答数据包至HOST, 直至所有应答数据包发送完毕

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. 指纹图像高分辨率模式 (Full Mode) 宽度*高度：242*266，

3. 指纹图像低分辨率模式 (Quarter Mode) 宽度*高度：121*133

例子1：上传全分辨率指纹图像

Host命令： 55 AA 00 00 22 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 22 01

Target响应： AA 55 01 00 22 00 06 00 00 00 F2 00 0A 01 00 00 00 00 00 00 00 00 00 00 25 02

注：全图像宽度=242，全图像高度=266

Target响应数据包：图像数据大小为242*266=64372字节，分为129*496字节+1*388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129个包含496字节图像数据的响应数据包

A5 5A 01 00 22 00 88 01 00 00 84 01 最后一帧388字节数据 2字节校验码

最后1个包含388字节图像数据的响应数据包

例子2：上传1/4图像的低分辨率指纹图像

Host命令： 55 AA 00 00 22 00 01 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 23 01

Target响应： AA 55 01 00 22 00 06 00 00 00 79 00 85 00 00 00 00 00 00 00 00 00 00 26 02

注：1/4图像宽度=121，全图像高度=133

Target响应数据包：图像数据大小为242*266/4=16093字节，分为32*496字节+1*221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32个包含496字节图像数据的响应数据包

A5 5A 01 00 22 00 E1 00 00 00 DD 00 最后一帧221字节数据 2字节校验码

最后1个包含221字节图像数据的响应数据包

5.3.11 下载指纹图像至模块的Image Buffer (CMD_DOWN_IMAGE)

● [功能 Function]

将从 Host 收到的图像数据保存于 ImageBuffer 中。

Host 以 496bytes 为单位将图像发送至 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：242*266

● [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像素)
	2bytes	图像高度 (266像素)
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	图像数据块编号(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5-11 CMD_DOWN_IMAGE指令

例子：下载指纹图像到ImageBuffer中

Host命令： 55 AA 00 00 23 00 04 00 F2 00 0A 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 23 02

Target响应： AA 55 01 00 23 00 02 00 25 01

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

Target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129个包含496字节图像数据的命令数据包及响应数据包

Host命令数据包： 5A A5 00 00 23 00 26 00 69 00 包含388字节图像数据的最后数据块 2字节校验码

Target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

5.3.12 保存Ram Buffer中的指纹模板数据 (CMD_STORE_CHAR)

● [功能 Function]

将指定的Ram Buffer中的模板保存到本模块的指定编号指纹库中。

● [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，则将指定 Ram Buffer中的Template注册于指定 Template编号中并返回其结果。

若 Duplication Check 设置为ON，则指定 Ram Buffer中的 Template 和已注册的所有 Template之间进行 1:N比对：若存在比对成功的模板，则 RET返回 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	Result Code 为 ERR_DUPLICATION_ID 时, 为 4, 否则, 为 2	
RET	Result Code	
DATA	2bytes	Result Code 为 ERR_DUPLICATION_ID 时, 为Template编 号, 否则为 0

表 5-12 CMD_STORE_CHAR指令

例子：保存RamBuffer中的模板数据到指定编号为1的模块数据库中

Host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 00 00 00 00 44 01

Target响应包: AA 55 01 00 40 00 02 00 42 01

5.3.13 从本模块数据库中读取指定编号模板并保存在Ram Buffer (CMD_LOAD_CHAR)

- [功能 Function]

将指定编号中的指纹Template 保存于指定的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_EMPTY 。

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5-13 CMD_LOAD_CHAR指令

例子：读取编号为1的模板数据暂存在RamBuffer0中

Host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 45 01

Target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 43 01

5.3.14 上传指定的Ram Buffer中的模板数据到HOST (CMD_UP_CHAR)

- [功能 Function]

将指定 Ram Buffer中的 Template 发送至 Host 。

- [工作过程 Sequence]

指定 Ram Buffer编号无效，则返回ERR_INVALID_BUFFER_ID 。

利用指令应答包将HOST待收到的 Template数据的大小发送至HOST。

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) 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
RET	ERR_SUCCESS
DATA	Template Record Data

表5-14 CMD_UP_CHAR指令

注：调用该指令之前，必须先调用CMD_GENERATE，CMD_DOWN_CHAR，CMD_LOAD_CHAR 其中的一个指令，将 Template 保存于 Ram Buffer 中。

例子：上传RamBuffer0中的模板数据到HOST

Host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 43 01

Target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 39 02

Target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

5.3.15 从HOST下传指纹模板到模块指定的Ram Buffer中 (CMD_DOWN_CHAR)

● [功能 Function]

从 Host 接收 Template 不呢个保存于指定 Ram Buffer 中。

● [工作过程 Sequence]

Host 发送指令包，使得 Target 进入数据(Ram Buffer编号 + Template书)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，则利用指令数据包设置Ram Buffer编号和 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5-15 CMD_DOWN_CHAR指令

注：保存于 Ram Buffer2 中的 Template，若调用如下指令：

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则会清掉。

建议，不要使用 Ram Buffer2 。

例子：下载指纹模板数据到模块的Rambuffer0中

Host命令：55 AA 00 00 43 00 02 00 F4 01 00 39 02

Target响：AA 55 01 00 43 00 02 00 45 01

Host命令数据包：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

Target响应数据包：A5 5A 01 00 43 00 02 00 00 00 45 01 ；数据应答包的长度因为没有数据是12个字节



5.3.16 删除指定编号范围内的指纹模板 (CMD_DEL_CHAR)

- [功能 Function]

删除指定删除范围(起始Template编号 ~ 结束Template编号)内注册的 Template 。

- [工作过程 Sequence]

若指定范围无效, 则返回ERR_INVALID_PARAM 。

若指定范围内没有注册 Template, 则返回ERR_TMPL_EMPTY 。

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5-16 CMD_DEL_CHAR指令

例子: 删除数据库中所有指纹 (即编号为1-500的所有指纹)

Host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 00 00 00 00 00 00 00 3D 02

Target响应: AA 55 01 00 44 00 02 00 46 01

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

- [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5-17 CMD_GET_EMPTY_ID指令

例子：获取1-500编号范围内（0x0001-0x01F4）的首个可注册的编号，该编号为11（0x000B）

Host命令： 55 AA 00 00 45 00 04 00 01 00 F4 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 3E 02

Target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 00 00 00 00 00 00 54 01

5.3.18 获取指定编号的注册状态（CMD_GET_STATUS）

● [功能 Function]

获取指定编号中的 Template 的注册状态。

● [工作 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	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5-18 CMD_GET_STATUS指令

例子1: 获取ID编号=1的注册状态, 可注册

Host命令包: 55 AA 00 00 46 00 02 00 01 00 48 01
 Target响应包: AA 55 01 00 46 00 03 00 49 01

例子2: 获取ID编号=1的注册状态, 已注册

Host命令包: 55 AA 00 00 46 00 02 00 01 00 48 01
 Target响应包: AA 55 01 00 46 00 03 00 00 00 01 00 4A 01

5.3.19 检测指定编号范围内的指纹模板数据是否坏损 (CMD_GET_BROKEN_ID)

● [功能 Function]

检查指定范围(起始 Template编号 ~ 结束 Template编号)内所有已注册模板的坏损情况。
 在 Flash 的 Write 操作中, 有可能因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 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5-19CMD_GET_BROKEN_ID指令

例子: 获取1-500范围内的指纹损坏的ID编号

Host命令: 55 AA 00 00 47 00 04 00 01 00 F4 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 40 02

Target响应: AA 55 01 00 47 00 06 00 4D 01

5.3.20 获取指定编号范围内已注册的指纹总数 (CMD_GET_ENROLL_COUNT)

- [功能 Function]

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

- [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5-20 CMD_GET_ENROLL_COUNT指令



例子：获取1-500（0x0001~0x01F4）编号范围内的已注册用户总数，总数为10（0x000A）

Host命令： 55 AA 00 00 48 00 03 00 01 00 F4 01 00 00 00 00 00 00 00 00 00 00 00 00 00 40 02

Target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 00 00 56 01

5.3.21 获取已注册 ID列表（CMD_GET_ENROLLED_ID_LIST）

● [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]

①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。

②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5-21 CMD_GET_ENROLLED_ID_LIST指令

5.3.22 从ImageBuffer生成并暂存模板在指定的RamBuffer中（CMD_GENERATE）

● [功能 Function]

从 ImageBuffer 中的指纹图像生产 Template 并保存于指定的 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5-22 CMD_GENERATE指令

例子1：从ImageBuffer中生成模板数据保存在RamBuffer0中

Host命令包: 55 AA 00 00 60 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 61 01

Target响应包: AA 55 01 00 60 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 62 01

例子2：从ImageBuffer中生成模板数据保存在RamBuffer1中



Host命令包: 55 AA 00 00 60 00 02 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 62 01

Target响应包: AA 55 01 00 60 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 62 01

例子3: 从ImageBuffer中生成模板数据保存在RamBuffer2中

Host命令包: 55 AA 00 00 60 00 02 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 63 01

Target响应包: AA 55 01 00 60 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 62 01

5.3.23 将RamBuffer中的2个或3个临时模板融合成1个模板数据 (CMD_MERGE)

● [功能 Function]

合成 Ram Buffer 中的 Template 并生成一个 Template 后保存于指定的 Ram Buffer 中。

合成个数可为 2，也可为 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 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5-23 CMD_MERGE指令

例子: 将RamBuffer中3个暂存的指纹模板融合为1个指纹模板数据

Host命令包: 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

Target响应包: 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



5.3.24 指定2个RamBuffer中的模板之间比对 (CMD_MATCH)

● [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5-24 CMD_MATCH指令

例子：将RamBuffer0与RamBuffer1中的指纹模板进行1:1比对

Host命令包: 55 AA 00 00 62 00 04 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 66 01
 Target包: AA 55 01 00 62 00 03 00 65 01

5.3.25 指定编号范围的1: N识别 (CMD_SEARCH)

● [功能 Function]

指定 Ram Buffer 中的 Template 与指定搜索范围(起始 Template编号 ~ 结束 Template编号内)的所有已注册在模块数据库中的指纹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 DATA 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	
DATA	2bytes	Ram Buffer
	2bytes	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5-25 CMD_SEARCH指令

例子：暂存在RamBuffer0中的指纹模板与1-500编号范围内的指纹比对，返回比对结果

Host命令: 55 AA 00 00 63 00 06 00 00 00 01 00 F4 01 00 00 00 00 00 00 00 00 00 00 00 00 5E 02
 Target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 00 00 71 01

5.3.26 指定编号的1:1比对指纹 (CMD_VERIFY)

● [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 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Template 与指定编号中的 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 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	待对比的 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(1byte) (1 : 已进行智能更新, 0: 没有进行智能更新)

表5-26 CMD_VERIFY指令

例子: RamBuffer0中的指纹模板与数据库中编号为8的指纹1:1验证

Host命令包: 55 AA 00 00 64 00 04 00 08 00 6F 01
 Target响应包: AA 55 01 00 64 00 05 00 00 00 08 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 72 01

5.3.27 通讯错误 (Incorrect Command)

● [功能 Function]

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

● [命令及响应 Command and Response]

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

表5-27 Incorrect Command指令

5.4 注意事项

1. 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 C



MD_GET_IMAGE 指令，将图像保存于 ImageBuffer 中。

2. 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。

3. 保存于 Ram Buffer2 中的 Template ，在调用CMD_SEARCH, CMD_VERIFY, CMD_GENERATE, CMD_STORE_CHARACTER, 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 。

六、附录

6.1 响应（Response）及错误代码表

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	不存在已破损 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_MEMORY	0x1C	外部Flash 烧写出错。
15	ERR_INVALID_TMPL_NO	0x1D	指定 Template 编号无效。
16	ERR_INVALID_PARAM	0x22	使用了不正确的参数。
17	ERR_GEN_COUNT	0x25	指纹合成个数无效。
18	ERR_INVALID_BUFFER_ID	0x26	Buffer ID 值不正确。
19	ERR_FP_NOT_DETECTED	0x28	采集器上没有指纹输入。

6.2 登记及比对流程图

6.2.1 注册流程 Enroll

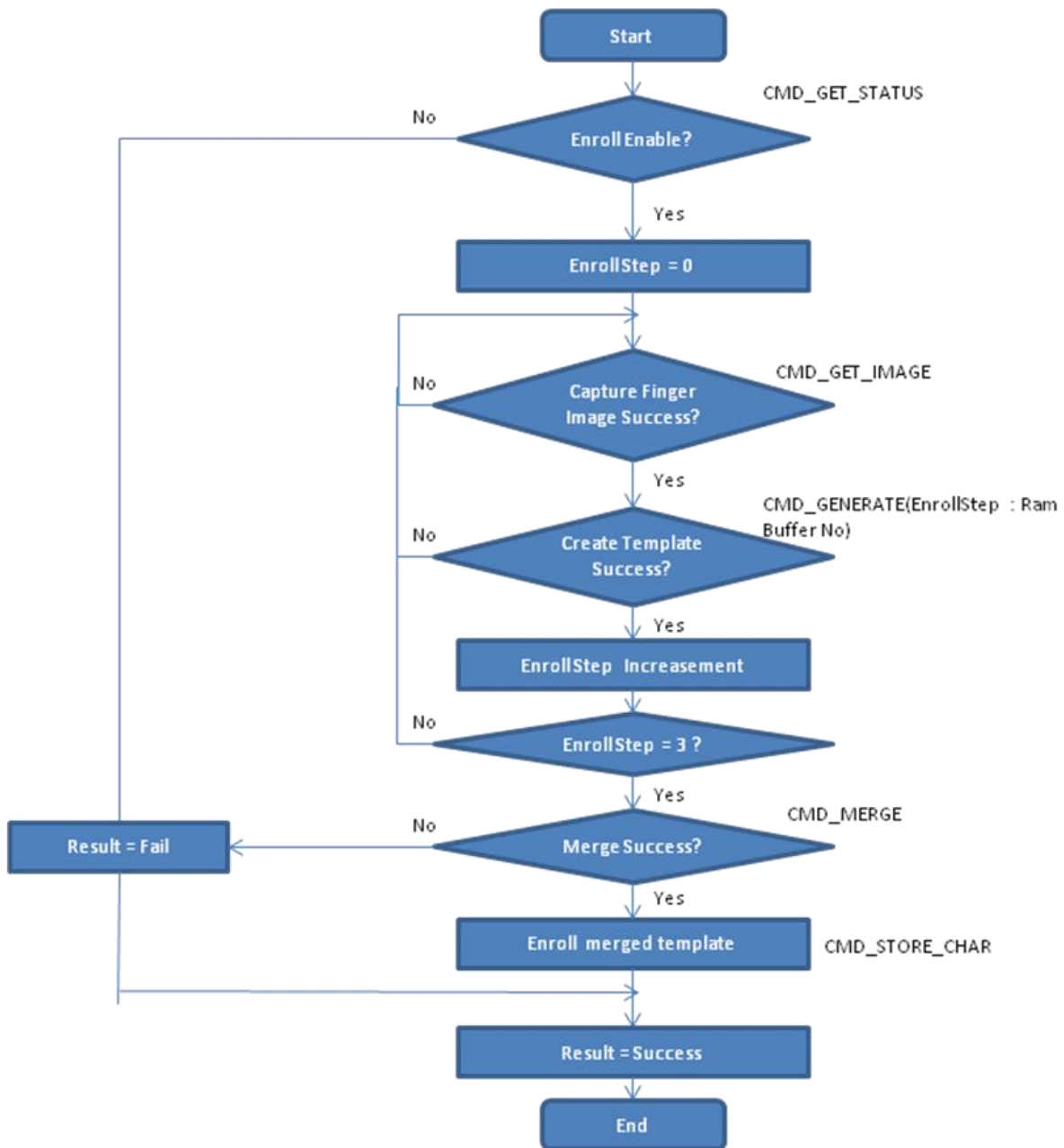


图6-1 注册流程



6.2.2 验证及识别流程 Verify & Identify

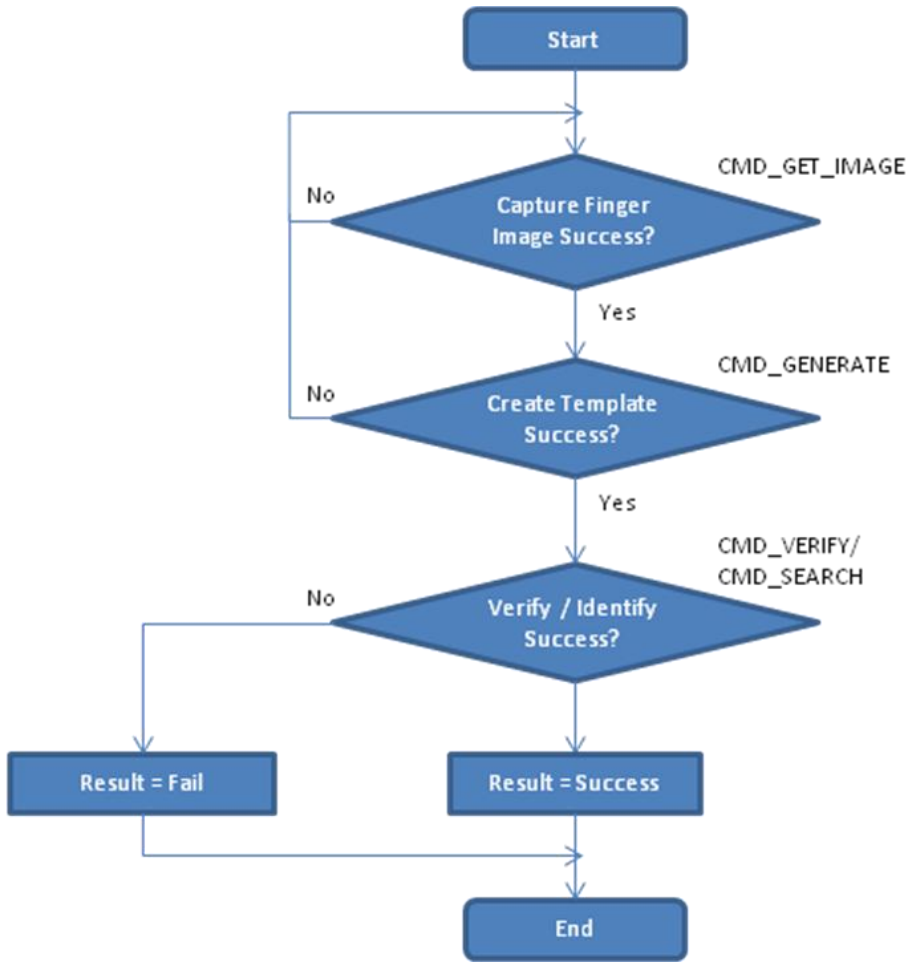


图6-2 验证及识别流程