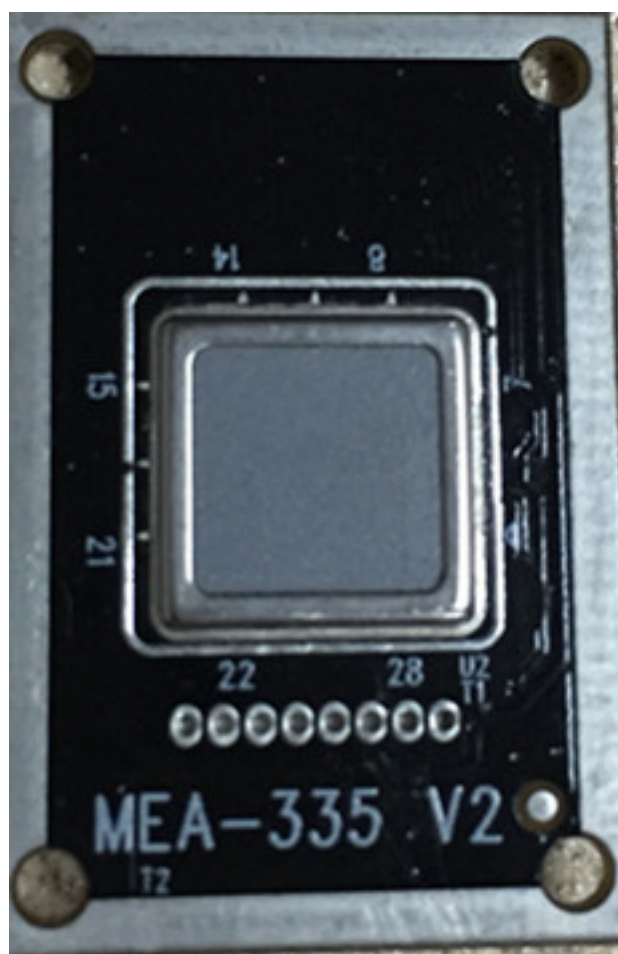


MEA-335 指纹模块用户手册

（通讯方式：USB/UART）

MEA-335 Fingerprint Module User's Guide
(Ver 2.0)



目录

(一)	概述.....	1
(二)	通讯协议.....	2
2.1	通讯处理过程.....	2
2.2	通讯包 PACKET 的分类	3
2.2.1	命令包 <i>Command packet</i>	3
2.2.2	响应包 <i>Response packet</i>	3
2.2.3	指令/响应的数据包 <i>Data Packet</i>	3
2.3	通讯包的帧结构.....	4
2.3.1	通讯包 <i>Parket</i> 识别代码.....	4
2.3.2	命令包 (<i>Command packet</i>) 的帧结构.....	4
2.3.3	响应包 (<i>Response packet</i>) 的帧结构.....	5
2.3.4	指令数据包 (<i>Command Data Packet</i>) 的帧结构.....	5
2.3.5	响应数据包 (<i>Response data packet</i>) 的帧结构.....	6
(三)	通讯命令 (COMMAND) 综述.....	7
3.1	指纹特征模板 (TEMPLATE RECORD) 的数据结构	7
3.2	命令列表 (COMMAND LIST)	7
(四)	各通讯命令 (COMMAND) 详细说明.....	9
4.1	连接测试 (CMD_TEST_CONNECTION)	9
4.2	设置参数 (CMD_SET_PARAM)	10
4.3	读取参数 (CMD_GET_PARAM)	12
4.4	读取设备信息 (CMD_DEVICE_INFO)	13
4.5	使模块进入 IAP 模式 (CMD_ENTER_IAP_MODE)	14
4.5	采集指纹图像 (CMD_GET_IMAGE)	15
4.6	检测手指 (CMD_FINGER_DETECT)	16
4.7	上传指纹图像到主机 (CMD_UP_IMAGE_CODE)	17
4.8	下载指纹图像到模块 (CMD_DOWN_IMAGE)	19
4.9	控制采集器背光灯 (CMD_SLED_CTRL) 开/关.....	21
4.10	保存指纹模板数据到模块指纹库 (CMD_STORE_CHAR)	22
4.11	读取指纹库中的指纹并暂存在指定的 RAMBUFFER 中 (CMD_LOAD_CHAR)	23
4.13	将暂存在 RAMBUFFER 中的指纹模板上传到主机 (CMD_UP_CHAR)	24
4.14	下载指纹模板数据到模块指定的 RAMBUFFER (CMD_DOWN_CHAR)	25
4.15	删除指定编号范围内的指纹 (CMD_DEL_CHAR)	27
4.16	获取指定编号范围内可注册的首个编号 (CMD_GET_EMPTY_ID)	28
4.17	检查指定的编号是否已被注册 (CMD_GET_STATUS)	29

MEA_335 指纹识别模块用户手册

4.17	检查指定编号范围内的指纹库是否有数据损坏 (CMD_GET_BROKEN_ID)	30
4.18	获取指定编号范围内已注册的指纹总数 (CMD_GET_ENROLL_COUNT)	31
4.19	从暂存在 IMAGEBUFFER 中的指纹图像产生模板 (CMD_GENERATE)	32
4.20	合成指纹模板数据用于入库 (CMD_MERGE)	33
4.21	指定 2 个 RAMBUFFER 之间的模板做比对 (CMD_MATCH)	34
4.22	指定编号范围的 1: N 识别 (CMD_SEARCH)	35
4.23	指定 RAMBUFFER 与指纹库中指定编号的模板比对 (CMD_VERIFY)	36
4.25	设置模块序列号 (CMD_SET_MODULE_SN)	37
4.26	读取模块序列号 (CMD_GET_MODULE_SN)	39
4.27	取消采集指纹 (CMD_FP_CANCEL)	40
4.28	获取已注册 ID 列表 (CMD_GET_ENROLLED_ID_LIST)	41
4.29	进入休眠状态 (CMD_ENTER_STANDBY_STATE)	43
4.30	通讯错误返回 (INCORRECT COMMAND)	44
4.31	注意事项.....	44
(五)	响应 (RESPONSE) 及错误代码表 (ERROR CODE)	45
(六)	登记及比对流程图.....	46
6.1	指纹模块的注册流程 (ENROLL PROCESS)	46
6.2	指纹模块的验证及识别流程 (VERIFY & IDENTIFY)	47
6.4	滑动采集器模块指纹验证和识别流程 (VERIFY & IDENTIFY)	49
(七)	硬件描述.....	50
7.1	主处理板尺寸	50
7.2	指纹模块接口信号定义 (接插件为 1.25MM 间距)	52
7.3	MEA-335 指纹识别模块技术参数.....	54

(一) 概述

本文描述了 BRK 公司指纹识别模块的串口参数，通讯过程，指令/数据格式。 本指令集适用于按压式半导体指纹传感器。

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。

数据传送:

数据以串行异步方式传送，第一位为起始位，其后是数据位。 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。

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

BRK 指纹识别模块所采用的指纹图像如下:

分辨率：508DPI

灰度：256（8 位）灰度 像素

大小：

按压式半导体采集器 BRK301B：128*128；

滑动式半导体采集器 BRK501：170*166；

主要功能:

不仅仅具有活体指纹注册入库（容量支持:50/1000/2000/3000），指纹验证（Verify）和识别（Identify），删除指定编号范围的指纹等常规功能外，还具有如下功能：

1. 上传指纹特征数据到主机，下载指纹特征数据到模块（入库/验证/识别）
2. 上传指纹图像到主机，下载指纹图像到模块（提取指纹特征入库/与活体指纹验证/识别）
3. 检查指定编号范围内的已注册保存在模块闪存内的指纹模板数据是否有损坏情况
4. 设置/读取指纹模块的序列号

(二) 通讯协议

2.1 通讯处理过程

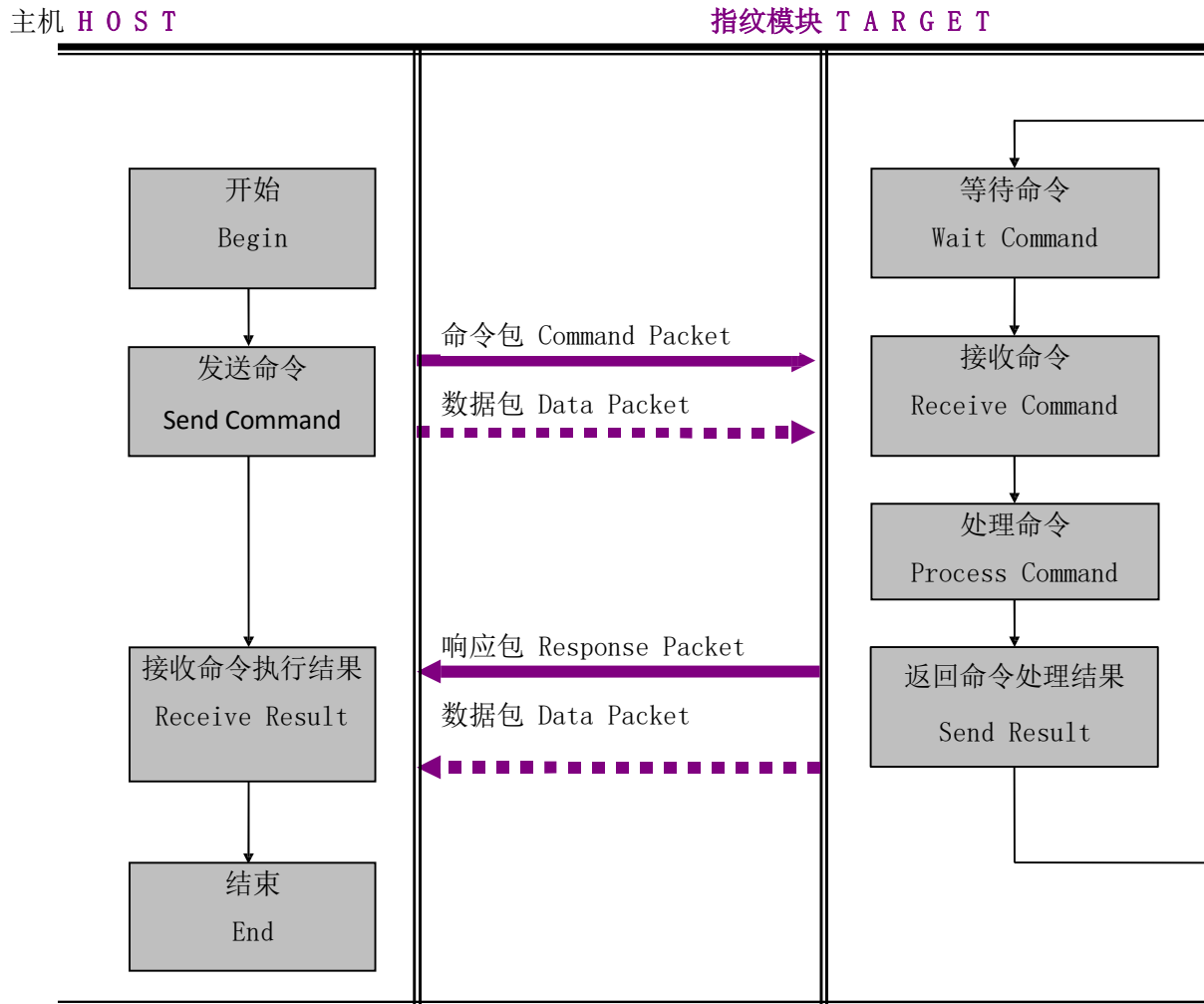


图 2-1 通讯过程

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 在没有收到应答时，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 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 500byte

2.3 通讯包的帧结构

2.3.1 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 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 DESCRIPTION
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 ~ DATA 所有数据的算术和的最低 2 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	...	D15	L	H
0	1	2	3	4	5	6	7	8	9	10	11	...	24	25	26

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 DESCRIPTION
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	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 ~ DATA 所有数据的算术和的最低 2 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 DESCRIPTION
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 ~ DATA 所有数据的算术和的最低 2 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	DESCRIPTION
0	PREFIX	WORD	2byte	包标识 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	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 ~ DATA 所有数据的算术和的最低 2 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）综述

3.1 指纹特征模板(Template Record)的数据结构

Template Data	Checksum
496 byte	2 byte
Template Data	Template Data 的每个字节的算术和的最低 2 字节.

表 3-1 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)

3.2 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	进行与设备的通讯测试
2	CMD_SET_PARAM	0x0002	设置设备参数 (Device ID, Security Level, Baudrate, Duplication Check, Auto Learn)
3	CMD_GET_PARAM	0x0003	获取设备参数 (Device ID, Security Level, Baudrate, Duplication Check, Auto 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_GET_IMAGE	0x0020	从采集器采集指纹图像并保存于 ImageBuffer 中
7	CMD_FINGER_DETECT	0x0021	检测指纹输入状态
8	CMD_UP_IMAGE	0x0022	将保存于 ImageBuffer 中的指纹图像上传至 HOST
9	CMD_DOWN_IMAGE	0x0023	HOST 下载指纹图像到模块的 ImageBuffer 中
10	CMD_SLED_CTRL	0x0024	控制采集器背光灯的开/关（注：半导体传感器不用此功能）
11	CMD_STORE_CHAR	0x0040	将指定编号 Ram Buffer 中的 Template，注册到指定编号的库中
12	CMD_LOAD_CHAR	0x0041	读取库中指定编号中的 Template 到指定编号的 Ram Buffer
13	CMD_UP_CHAR	0x0042	将保存于指定编号的 Ram Buffer 中的 Template 上传至 HOST
14	CMD_DOWN_CHAR	0x0043	从 HOST 下载 Template 到模块指定编号的 Ram Buffer 中
15	CMD_DEL_CHAR	0x0044	删除指定编号范围内的 Template。
16	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。
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 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	将保存于 Ram Buffer 中的两或三个模板数据融合成一个模板数据
22	CMD_MATCH	0x0062	指定 Ram Buffer 中的两个指纹模板之间进行 1:1 比对

序号 No	命令名称 Command Name	命令码 Code	命令功能 Function
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 比对
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
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_FP_CANCEL	0x0025	取消指纹采集操作 (只适用于带 TimeOut 参数的滑动传感器)
28	CMD_GET_ENROLLED_ID_LIST	0x0049	获取已注册 ID 列表
29	CMD_ENTER_STANDY_STATE	0x000C	使模块进入休眠状态。

(四) 各通讯命令（Command）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 用于暂存指纹模板数据，模块共有三个 Ram Buffer：

Ram Buffer0, Ram Buffer1 和 Ram Buffer2 。 注：断电情况下，ImageBuffer 和 Ram Buffer 中的数据会丢失。

4.1 连接测试（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 Target 的连接不正常，或 Target 的工作不正常，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
DATA	无数据

表 4-1 CMD_TEST_CONNECTION 指令

4.1 例子：HOST 发送 CMD_TEST_CONNECTION 指令及模块的响应

HOST 命令： 55 AA 00 00 0100 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01

Target 响应： AA 55 01 00 010002000000 00 00 00 00 00 00 00 00 00 00 00 00 03 01

4.2 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, FP TimeOut) 并返回其结果。

[工作过程 Sequence]

- ①若指定 Parameter Type 无效, 则返回 ERR_INVALID_PARAM 。
- ②若指定 Parameter Value 无效, 则返回 ERR_INVALID_PARAM 。
- ③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	
DATA	无数据	

表 4-2 CMD_SET_PARAM 指令

[参数类型 Parameter Type]

Parameter Type	Parameter ValueDescription
0	表示本设备编号 (Device ID)。可设置 1 ~ 255 。

1	<p>表示安全等级（Security Level）：可设置值：1~5 。默认为：3</p> <p>Security Level 对应的识别率如下表：</p> <table><tr><th>Security Level</th><th colspan="2">识别率</th></tr><tr><td rowspan="2">Level 1</td><td>认假率 FAR（False Acceptance Rate）</td><td>0. 1%</td></tr><tr><td>拒真率 FRR（False Rejection Rate）</td><td>0.005%</td></tr><tr><td rowspan="2">Level 2</td><td>认假率 FAR（False Acceptance Rate）</td><td>0.003%</td></tr><tr><td>拒真率 FRR（False Rejection Rate）</td><td>0.01%</td></tr><tr><td rowspan="2">Level 3</td><td>认假率 FAR（False Acceptance Rate）</td><td>0.001 %</td></tr><tr><td>拒真率 FRR（False Rejection Rate）</td><td>0.1 %</td></tr><tr><td rowspan="2">Level 4</td><td>认假率 FAR（False Acceptance Rate）</td><td>0.003%</td></tr><tr><td>拒真率 FRR（False Rejection Rate）</td><td>0.5%</td></tr><tr><td rowspan="2">Level 5</td><td>认假率 FAR（False Acceptance Rate）</td><td>0.0001%</td></tr><tr><td>拒真率 FRR（False Rejection Rate）</td><td>%</td></tr></table>	Security Level	识别率		Level 1	认假率 FAR（False Acceptance Rate）	0. 1%	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）	0.003%	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）	%
Security Level	识别率																												
Level 1	认假率 FAR（False Acceptance Rate）	0. 1%																											
	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）	0.003%																											
	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）	%																											
2	<p>指纹重复检查（Duplication Check）状态开/关。可设置 0 或 1。</p> <p>若为 1 ，则处理 CMD_STORE_CHAR 指令时进行重复检测。</p> <p>若为 0 ，则不进行重复检测。</p>																												
3	<p>波特率（Baudrate）参数。可设置索引值： 1 ~ 8 。</p> <p>1:9600bps, 2:19200bps, 3:38400bps, 4:57600bps, 5:115200bps</p> <p>6:230400bps, 7:460800bps, 8:921600bps</p>																												
4	<p>表示指纹模板自学习（Auto Learn）状态开/关。可设置 0 或 1 。</p> <p>若为 1 ：则处理 CMD_SEARCH, CMD_VERIFY 指令时进行智能更新。</p> <p>若为 0 ：则不进行智能更新。</p>																												
5	<p>表示采集指纹超时时间（ Fp TimeOut）参数，可设置值：1 秒至 60 秒。</p> <p>CMD_GET_IMAGE 指令中采用该参数，在 FP TimeOUT 时间内等待指纹的输入。</p>																												

4.2 例子: 设置波特率为 921600BPS

HOST 命令: 55 AA 00 00 020005000308000000 00 00 00 00 00 00 00 00 00 00 00 11 01

Target 响应: AA 55 01 00 020002000000 00 00 00 00 00 00 00 00 00 00 00 00 00 04 01

4.3 读取参数 (CMD_GET_PARAM)

[功能 Function]

根据指定 Parameter Type，获取设备参数(Device ID, Security Level, Baudrate, Duplication Check, Auto Learn, FP TimeOut)。

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	
DATA	4bytes	成功时：Parameter Value

表 4-3 CMD_GET_PARAM 指令

4.3 例子 1：读取当前安全等级 (返回安全等级=3)

Host 命令: 55 AA 00 00 0300010001 00 00 00 00 00 00 00 00 00 00 00 00 00 00 04 01
Target 响应: AA 55 01 00 03000600000003000000 00 00 00 00 00 00 00 00 00 00 0C 01

4.3 例子 2：读取当前 TimeOut 值 (TimeOut=5S)；

Host 命令: 55 AA 00 00 0300010005 00 00 00 00 00 00 00 00 00 00 00 00 00 00 08 01
Target 响应: AA 55 01 00 03000600000005000000 00 00 00 00 00 00 00 00 00 00 0E 01

4.4 使模块进入 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 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！！

4.5 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 4-6CMTD IMAGE 指令

4.6 例子 1: 发送采集指纹图像后模块检测到手指的命令及响应

[illegible]

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

[illegible]

4.6 检测手指 (CMD_FINGER_DETECT)

[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]

返回收到该指令时刻 Sensor 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 4-7CMD_FINGER_DETECT 指令

4.7 例子 1：没检测到指纹

Host 命令：55 AA 00 00 2100 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 21000300000000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 24 01

4.7 例子 2：检测到有指纹

Host 命令：55 AA 00 00 2100 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 21000300000001 00 00 00 00 00 00 00 00 00 00 00 00 00 00 25 01

4.7 上传指纹图像到主机 (CMD_UP_IMAGE_CODE)

[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：

(按压式半导体采集器 BRK301B：128*128；按压式半导体采集器 BRK501：170*166；)。若为 1：则发送 1/4 图像（4 个点取 1 个点）。

[工作过程 Sequence]

- ①若指定 Image Type 无效，则返回 ERR_INVALID_PARAM。
- ②利用指令应答包，将 HOST 待收到图像的大小发送至 HOST。
- ③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 (170/128) Quarter 图像 (85/64)
	2bytes	成功：图像的高度 Full 全图像 (166/128) Quarter 图像 (83/64)
	成功时：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 4-8CMD_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: 170*166/128*128
3. 低分辨率模式 (Quarter Mode) 宽度*高度: 85*83/64*64

4.8 例子 1: 上传全分辨率 301b 指纹图像

Host 命令: 55 AA 00 00 2200010000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 22 01

Target 响应: AA 55 01 00 22000600000080008000 00 00 00 00 00 00 00 00 00 00 00 00 22 08

注: 全图像宽度=0x80=128, 全图像高度=0x80=128

Target 响应数据包: 图像数据大小为 128*128=16384 字节, 分为 33*496 字节+1*16 字节

A5 5A 01 00 2200F4010000F001 本帧的 496 字节数据 2 字节校验码

。 。 。

共 33 个包含 496 字节图像数据的响应数据包

A5 5A 01 00 2200140000001000 最后一帧 36 字节数据 2 字节校验码

最后 1 个包含 16 字节图像数据的响应数据包

4.8 例子 2: 上传 1/4 图像分辨率的 301b 指纹图像

Host 命令: 55 AA 00 00 2200010001 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 23 01

Target 响应: AA 55 01 00 22000600000040004000 00 00 00 00 00 00 00 00 00 00 00 00 30 01

注: 1/4 图像宽度=0x40=64, 全图像高度=0x40=64

Target 响应数据包: 图像数据大小为 128*128/4=4096 字节, 分为 8*496 字节+1*128 字节

A5 5A 01 00 2200F4010000F001 本帧的 496 字节数据 2 字节校验码

。 。 。

共 8 个包含 496 字节图像数据的响应数据包

A5 5A 01 00 2200840000008000 最后一帧 128 字节数据 2 字节校验码

最后 1 个包含 128 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)

[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。

[工作过程 Sequence]

- ①若图像高度或图像宽度不正确，则返回 ERR_INVALID_PARAM 。
- ②利用应答包返回 ERR_SUCCESS 。
- ③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

Host 命令： 55 AA 00 00 23000400CA000201 00 00 00 00 00 00 00 00 00 00 00 00 F3 01

Target 响应： AA 55 01 00 230002000000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 25 01

HOST 命令数据包： 5A A5 00 002300F20100 00 本块的 496 字节数据 2 字节校验码

Target 响应数据包： A5 5A 01 00 2300020000 00 25 01；数据应答包的长度因为没有数据是 12 个字节
。。。共 33 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 00230026006900 包含 36 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 230002000000 25 01；数据应答包的长度因为没有数据是 12 个字节

4.10 保存指纹模板数据到模块指纹库 (CMD_STORE_CHAR)

[功能 Function]

将保存于指定 Ram Buffer 中的模板保存于指定编号的模块指纹库中。

[工作过程 Sequence]

- ① 若指定 Template 编号无效，则返回错误码 ERR_INVALID_TMPL_NO。
- ② 若指定 Ram Buffer 编号无效，则返回错误码 ERR_INVALID_BUFFER_ID。
- ③ 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。
- ④ 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	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 4-11 CMD_STORE_CHAR 指令

4.10 例子：保存 RamBuffer0 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 4000040001000000 00 00 00 00 00 00 00 00 00 00 00 00 44 01

Target 响应包： AA 55 01 00 400002000000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 42 01

4.11 读取指纹库中的指纹并暂存在指定的RamBuffer中（CMD_LOAD_CHAR）

[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]

- ①若指定 Template 编号无效，则返回 ERR_INVALID_TMPL_NO 。
- ②若指定 Template 编号中没有注册 Template，则返回错误码 ERR_TMPL_EMPTY 。
- ③若指定 Ram Buffer 编号无效，则返回错误码 ERR_INVALID_BUFFER_ID 。
- ④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

4.11 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 41000400010000 00 00 00 00 00 00 00 00 00 00 00 00 00 45 01
 Target 响应包： AA 55 01 00 410002000000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 43 01

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

[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）， 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 4-13CMD_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 中的一个指令，将 Template 保存于 Ram Buffer 中。

4.12 例子：上传 RamBuffer0 中的模板数据到 HOST

Host 命令包: 55 AA 00 00 420002000000 00 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 420004000000F201 00 00 00 00 00 00 00 00 00 00 00 00 00 00 39 02

Target 响应数据包: A5 5A 01 00 4200F4010000498 字节的本指纹模板数据 2 字节校验码

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

[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]

- ①Host 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。
- ②Target 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。
- ③ 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 RamBuffer 编号和 Template 数据并发送至 Target。④Target 收到指令数据包后, 若 Ram Buffer ID 无效, 则返回 ERR_INVALID_BUFFER_ID。
- ⑤检查收到的 Template 的 CheckSum。若不正确, 则返回 ERR_INVALID_TMPL_DATA。
- ⑥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 4-14CMD_DOWN_CHAR 指令

注：保存于 Ram Buffer2 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。建议，不要使用 Ram Buffer2。

4.13 例子：下载指纹模板数据到模块的 Rambuffer0 中

Host 命令：55 AA 00 00 43000200F401 00 00 00 00 00 00 00 00 00 00 00 00 00 00 39 02

Target 响：AA 55 01 00 430002000000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 45 01

Host 命令数据包：5A A5 00 00 4300F4010000498 字节指纹模板数据 2 字节校验码

Target 响应数据包：A5 5A 01 00 430002000000 45 01；数据应答包的长度因为没有数据是 12 个字节

4.14 删除指定编号范围内的指纹 (CMD_DEL_CHAR)

[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

4.14 例子：删除数据库中编号为 1-2000 的所有指纹

Host 命令: 55 AA 00 00 440004000100D007 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 1F 02
 Target 响应: AA 55 01 00 440002000000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 46 01

4.15 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 4-16 CMD_GET_EMPTY_ID 指令

4.15 例子：获取 1-2000 编号范围内 (0x0001-0x07D0) 的首个可注册编号，结果该编号为 11

Host 命令： 55 AA 00 00 45 00 04 00 01 00 D0 07 00 00 00 00 00 00 00 00 00 00 20 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 54 01

4.16 检查指定的编号是否已被注册 (CMD_GET_STATUS)

[功能 Function]

获取指定编号中的 Template 的注册状态。

[工作过程 Sequence]

- ① 若指定 Template 编号无效，则返回 ERR_INVALID_TMPL_NO 。
- ② 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：4，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 4-17CMD_GET_STATUS 指令

4.16 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

4.16 例子二：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 4A 01

4.17 检查指定编号范围内的指纹库是否有数据损坏 (CMD_GET_BROKEN_ID)

[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]

①若指定范围无效,则返回 ERR_INVALID_PARAM。

②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 4-18CMD_GET_BROKEN_ID 指令

4.17 例子: 获取1-2000 范围内的指纹坏损的ID 编号

Host 命令: 55 AA 00 00 47 00 04 00 01 00 D0 07 00 00 00 00 00 00 00 00 00 00 00 00 00 00 22 02

Target 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 00 00 4D 01

4.18 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)内已注册的指纹总数。

[工作过程 Sequence]

① 若指定范围无效，则返回 ERR_INVALID_PARAM 。

② 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 5-19CMD_GET_ENROLL_COUNT 指令

4.18 例子：获取 1-2000 (0x0001~0x07D0) 范围内的已注册用户总数，总数为 10 (0x000A)

Host 命令： 55 AA 00 00 480004000100D007 00 00 00 00 00 00 00 00 00 00 00 00 23 02
 Target 响应： AA 55 01 00 4800040000000A 00 00 00 00 00 00 00 00 00 00 00 00 56 01

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

[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]

- ①若指定 Ram Buffer 编号无效，则返回错误码 ERR_INVALID_BUFFER_ID 。
- ②检查 ImageBuffer 中图像的正确性。若不正确，则返回 ERR_BAD_QUALITY 。
- ③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 4-20CMD_GENERATE 指令

4.19 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 0002 0000 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 0002 0000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 62 01

4.19 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 0002 0001 00 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 0002 0000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 62 01

4.19 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 0002 0002 00 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 0002 0000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 62 01

4.20 合成指纹模板数据用于入库 (CMD_MERGE)

[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]

- ①若指定 Ram Buffer 编号无效, 则返回错误码 ERR_INVALID_BUFFER_ID 。
- ②若合成个数无效, 则返回 ERR_GEN_COUNT 。
- ③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。
- ④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

4.20 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 0003 0000 0003 00 00 00 00 00 00 00 00 00 00 00 00 00 00 66 01

Target 响应包: AA 55 01 00 61 0002 0000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 63 01

4.21 指定 2 个 RamBuffer 之间的模板做比对 (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]

- ①若指定 Ram Buffer 编号无效，则返回错误码 ERR_INVALID_BUFFER_ID。
- ②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 4-22CMD_MATCH 指令

4.21 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 0004 0000 0001 00 00 00 00 00 00 00 00 00 00 00 00 00 00 66 01
 Target 包: AA 55 01 00 62 0002 0000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 64 01

4.22 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) 内的所有已注册指纹 Template 之间进行 1:N 比对并返回其结果。

[工作过程 Sequence]

- ①若指定 Ram Buffer 编号无效，则返回错误码 ERR_INVALID_BUFFER_ID 。
- ②若指定搜索范围无效，则返回错误码 ERR_INVALID_BUFFER_ID 。
- ③若没有已注册 Template ，则返回错误码 ERR_ALL_TMPL_EMPTY 。
- ④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	
DATA	2bytes	Ram Buffer 编号
	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 4-23 CMD_SEARCH 指令

4.22 例子：暂存在 RamBuffer0 中的指纹模板与 1-2000 编号范围内的指纹比对，返回比对结果

Host 命令： 55 AA 00 00 63 0006 0000 0001 00D0 07 00 00 00 00 00 00 00 00 00 00 40 02

Target 响应： AA 55 01 00 63 0005 0000 0008 0001 00 00 00 00 00 00 00 00 00 00 00 71 01

4.23 指定RamBuffer与指纹库中指定编号的模板比对（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。

[工作过程 Sequence]

- ①若指定 Template 编号无效，则返回错误码 ERR_INVALID_TMPL_NO 。
- ②若指定 Ram Buffer 编号无效，则返回错误码 ERR_INVALID_BUFFER_ID 。
- ③若不存在指定编号注册的 Template ，则返回错误码 ERR_TMPL_EMPTY。
- ④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 4-24CMD_VERIFY 指令

4.23 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 0004 0008 0000 00 00 00 00 00 00 00 00 00 00 00 00 6F 01

Target 响应包： AA 55 01 00 64 0005 0000 0008 0001 00 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)

[功能 Function]

从 Host 接收模块序列号 (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
指令数据包	
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 4-27 CMD_SET_MODULE_SN 指令

4.26 读取模块序列号 (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]

- ① 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。
- ② 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 4-28 CMD_GET_MODULE_SN 指令

4.27 取消采集指纹 (CMD_FP_CANCEL)

[功能 Function]

取消指纹采集过程指令。

若在处理 CMD_GET_IMAGE 指令过程中收到 CMD_FP_CANCEL 指令:

则中止 CMD_GET_IMAGE 指令的处理并以 ERR_FP_CANCEL 作为 CMD_GET_IMAGE 指令的返回值返回该错误码。

注：CMD_FP_CANCEL 指令没有应答包。

[工作 Sequence] 设置当前处理中的指令运行取

消标记。 **[命令及响应Command and Response]**

PREFIX	0xAA55
SID	Source Device ID
DID	Destination Device ID
CMD	0x0025
LEN	0
DATA	无

表 4-31 CMD_FP_CANCEL 指令

4.27 例子 1：在采集指纹图像过程中给模块发送取消指令（CMD_FP_CANCE）

[illegible]

4.28 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 $x(x = \text{字节号(从0开始)} * 8 + \text{位号(从0开始)})$ 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。

[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 4-28 CMD_GET_ENROLLED_ID_LIST 指令

4.29 进入休眠状态 (CMD_ENTER_STANDBY_STATE)

[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。

[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 4-29 CMD_ENTER_STANDBY_STATE

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 4-31 Incorrect Command 返回

4.31 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 CMD_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，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: ①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。 ②若没有设置通讯密码, 则可以不经确认密码 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) 登记及比对流程图

6.1 指纹模块的注册流程 (Enroll Process)

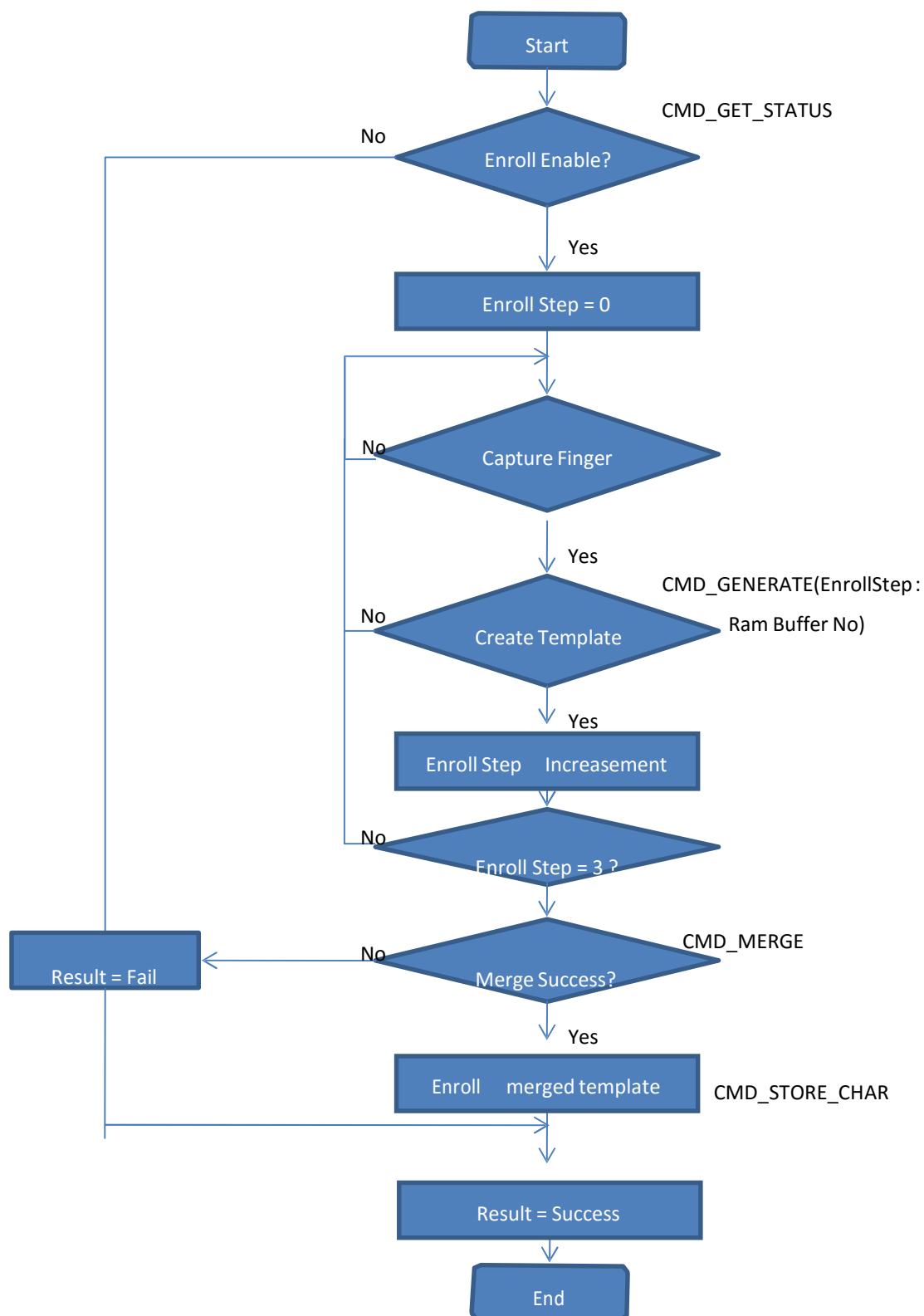


图 6-1 指纹采集器模块的注

6.2 指纹模块的验证及识别流程 (Verify & Identify)

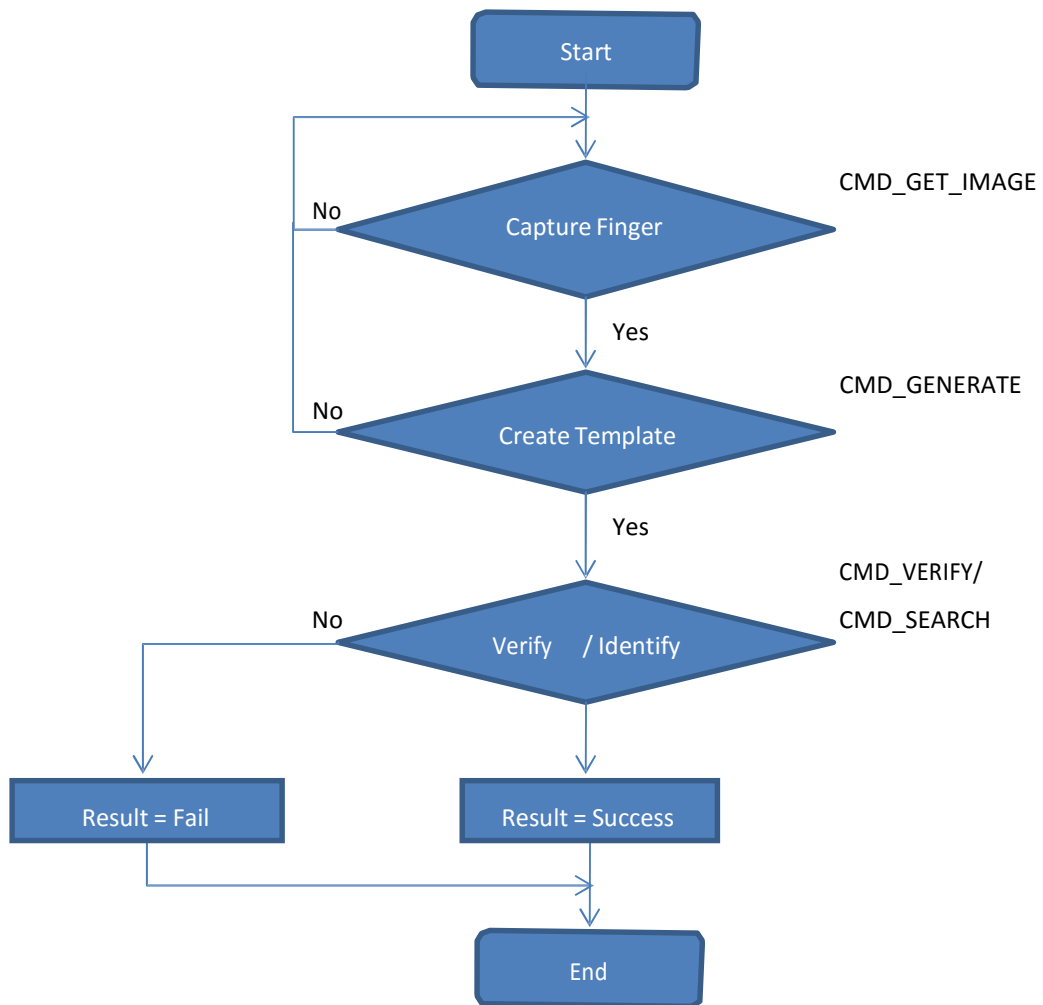
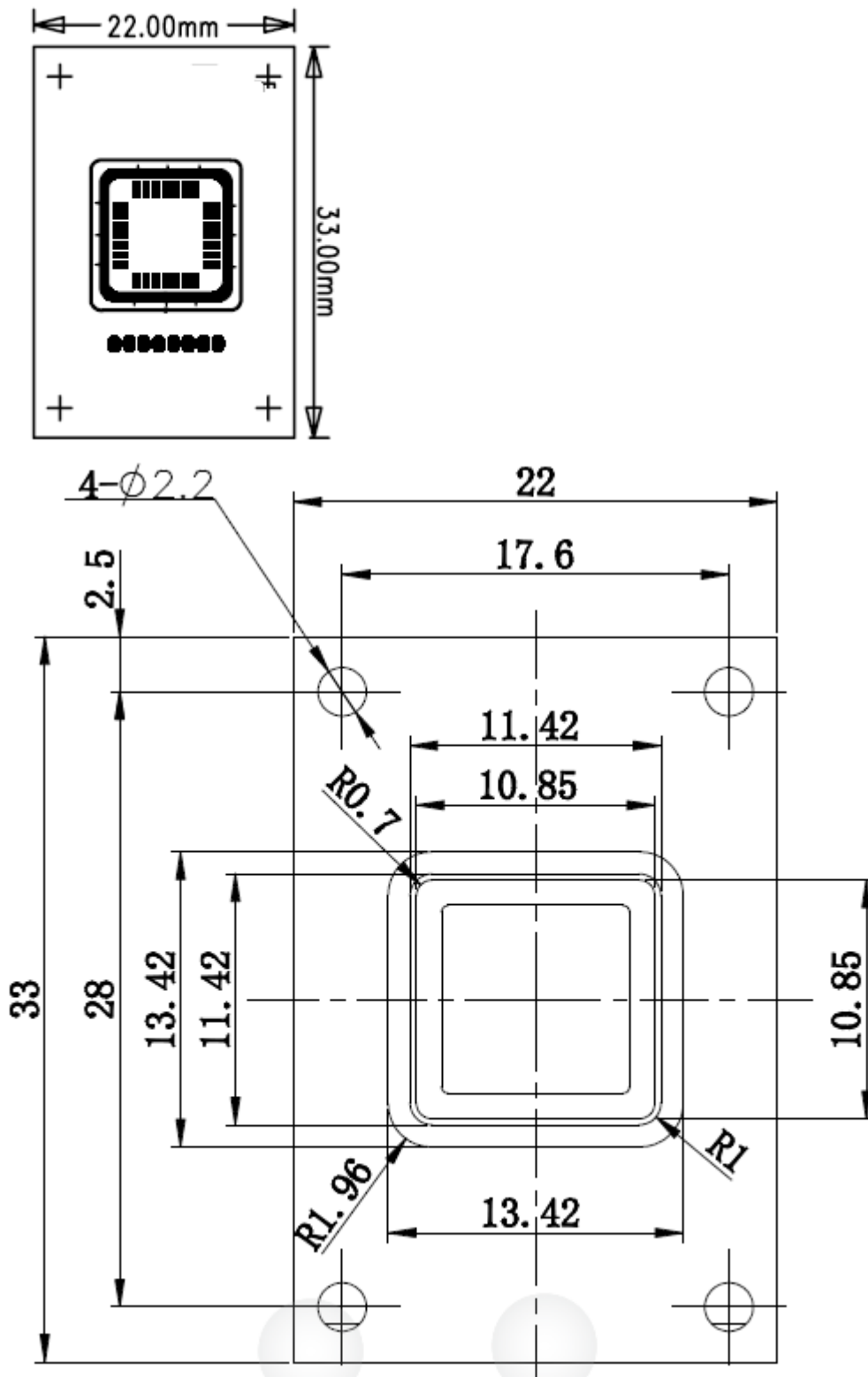


图 6-2 指纹模块的验证及识别流

(七) 硬件描述

7.1 主处理板尺寸: 22*33*4.85 毫米(W * L*H)



7.2 指纹模块接口信号定义（接插件为1.0mm 间距）

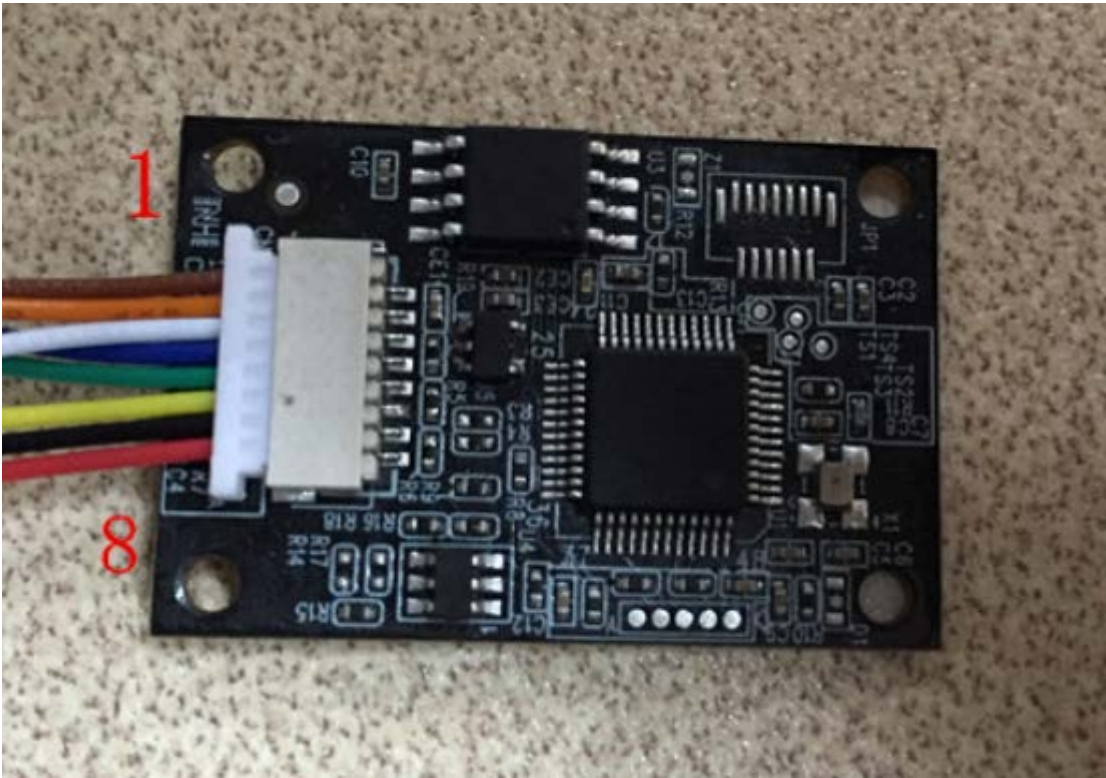
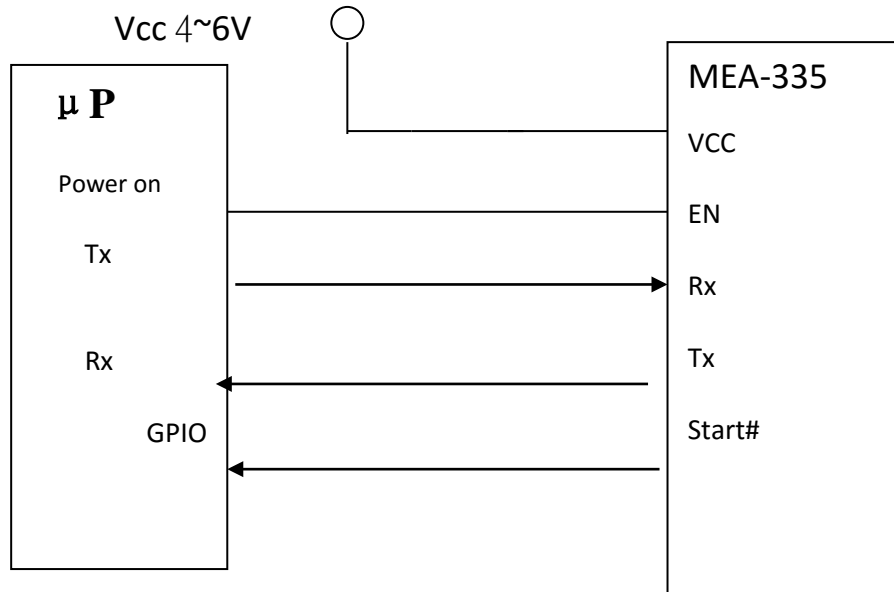


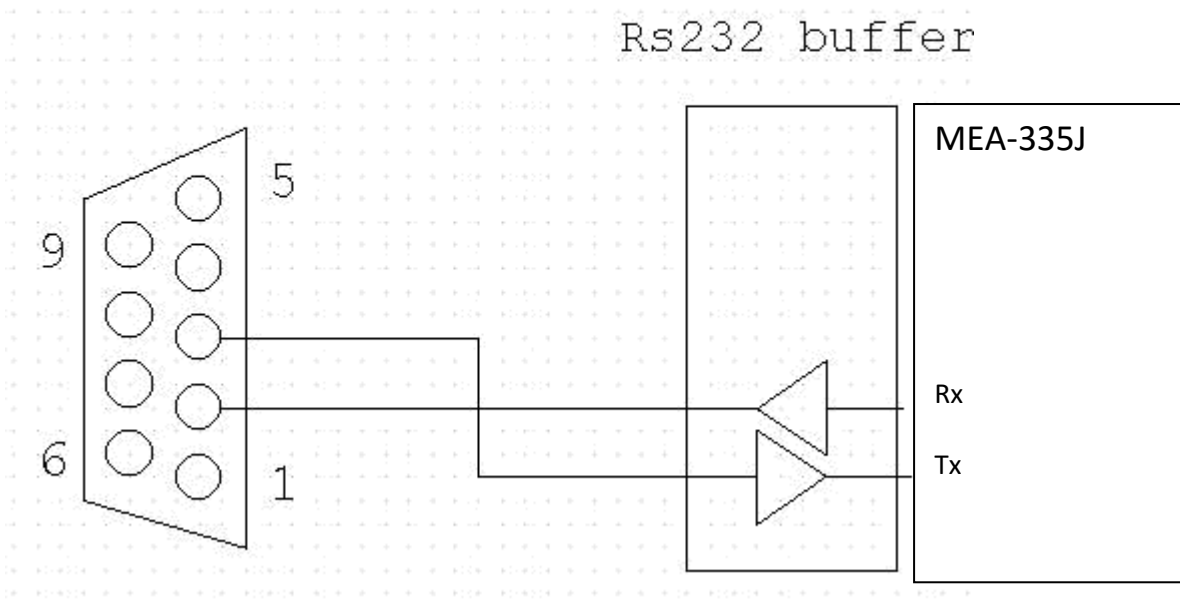
表 7.2-1: MEA-335 模块 USB/UART 通讯接口信号定义表

引脚号	接脚名字	型态	定义
1	通用引脚	IO	GPIO主处理器控制
2	GND	GND	电源地
3	模组电源	电源	电源, 4~6 V
4	Touch 输出	O	触控开关输出数据, High:触摸模组
5	模组致能	I	模组致能; High:开启,Low:关闭
6	RXD/USB D-	收	异步的通信接口传输数据, TTL 3.3V或者 USB D-
7	TXD/USB D+	传	异步的通信接口接收数据, TTL 3.3V或者 USB D+
8	GND	GND	电源地

图： 微处理器连续的通讯简图



串行通讯简图:
PC RS232
connector



*MEA-335J pin1 的电源是+4~6 V。

*MEA-335J 在 CN1 里的通讯使用 RX, TX 和 GND。如果连接一个 PC, 这些信号必须被电位提升转变。

7.3 MEA-335 指纹识别模块技术参数

处理器	32位RISC
闪烁记忆体	1024 K字节
SRAM 记忆体	128 K字节
指纹模板存储量	1: 1 / 1: N 方式-100个以上的指纹模板
传感器	按压式半导体传感器
硬件接口	RS-232 (TTL)/USB
手印传感器决定	~508 DPI
传感器芯片像素数目	128 像素* 128行
传感器尺寸	9*9*0.6毫米
灰度值	256 水平(8 位/像素)
特征档大小	512个字节
错误拒绝率	<1/100
误判率	<1/100,000
可允许的旋转	360度
ESD 静电防护	15KV 空气接触方式
扫描速度	<1秒
比对响应时间1: N	<2秒
重量	少于60 克
传感器温度	-40 ~70 °C
可允许的电压	4~6 V DC
功耗(瓦数)	撷取影像: 20 mA; 全速: 30 mA; 休眠: 0.1 mA
MMI	觸控開關