

SilkID 指纹模块通信协议

版本号: V2.2

重要申明

首先感谢您选择本产品。在使用前,请您仔细阅读本产品的说明书。以避免设备受到不必要的损害!本公司提醒您正确使用,将得到良好的使用效果和验证速度。

非经本公司书面同意,任何单位和个人不得擅自摘抄、复制本手册内容的部分或全部,并不得以任何形式传播。

本手册中描述的产品中,可能包含我公司及其可能存在的许可人享有版权的软件,除非获得相关权利人的许可,否则,任何人不能以任何形式对前述软件进行复制、分发、修改、摘录、反编译、反汇编、解密、反向工程、出租、转让、分许以及其他侵权软件版权的行为,但是适用法禁止此类限制的除外。



由于产品的不断更新,本公司不能承诺实际产品与该资料一致, 同时也不承担由于实际技术参数与本资料不符所导致的任何争议,任何改动恕不提前通知。

目 录

1.	数据包协议.....	1
1.1	普通数据包传输协议.....	1
1.2	扩展数据包传输协议.....	2
2.	模块的应用方式.....	5
2.1	验证模式.....	5
2.2	读头(图片)模式.....	5
2.3	读头(模板)模式.....	5
3.	系统配置.....	5
3.1	保存验证记录.....	7
3.2	用户容量.....	7
3.3	记录数量.....	7
3.4	记录容量.....	7
3.5	指纹数量.....	8
3.6	指纹容量.....	8
3.7	超时.....	8
3.8	波特率.....	8
3.9	自动应答.....	9
3.10	模块 ID.....	9
3.11	固件版本号.....	9
3.12	创建日期.....	9
3.13	工作模式.....	9
3.14	模板格式.....	10
3.15	闪灯控制.....	10
4.	模块操作命令.....	11
4.1	Flag/Error 表.....	13
4.2	MD_SYS_WP.....	13
4.3	MD_SYS_SP.....	14
4.4	MD_SYS_RP.....	15
4.5	MD_SYS_STATUS.....	16
4.6	MD_ENROLL_SCAN.....	17
4.7	MD_ENROLL_IMAGE_X.....	18
4.8	MD_IDENTIFY_IMAGE_X.....	20
4.9	MD_ENROLL_TMP.....	21
4.10	MD_VERIFY_SCAN.....	22
4.11	MD_IDENTIFY_FREE.....	23
4.12	MD_DEL_ALL_TMP.....	23
4.13	MD_DEL_TMP.....	24
4.14	MD_READ_TMP_X.....	25
4.15	MD_ADD_USER.....	26
4.16	MD_READ_USER.....	27
4.17	MD_DELETE_USER.....	28
4.18	MD_DEL_ALL_USER.....	29

4.19	MD_SCAN_IMAGE_X.....	30
4.20	MD_SET_TIME	31
4.21	MD_GET_TIME.....	32
4.22	MD_DEL ALOG	33
4.23	MD_LOAD_LOG_X.....	34
4.24	MD_LOAD_USER_X.....	36
4.25	MD_LOAD_TMP_X	37
4.26	MD_DEL_DB	39
4.27	MD_SCAN_TEMPLATE.....	40
4.28	MD_DISCONNECT	41
4.29	MD_RESET.....	42
4.30	MD_ENABLEDEVICE	42
4.31	MD_DISABLEDEVICE	43
4.32	MD_UPDATE_FW	44
4.33	MD_WT_FILE_X.....	45
4.34	MD_WU_FILE_X	46
4.35	MD_RT_FILE_X	46
4.36	MD_RU_FILE_X.....	47
4.37	MD_IDENTIFY_FREE.....	48
5	注意事项.....	49
5.2	数据包.....	49
5.3	用户 ID	49
5.4	指纹模板长度.....	49
5.5	验证记录.....	49
5.6	命令耗时.....	49
5.7	删除命令.....	50
6	附件.....	50
6.2	数据传输参照文档.....	50

1. 数据包协议

1.1 普通数据包传输协议

在指纹模块的数据包协议中，一个数据包为 13bytes 的长度，主机和模块之间的通信采用相同的数据包格式。

数据包格式：

Start	Command	Param	Size	Flag/Error	Checksum	End
1byte	1byte	4bytes	4bytes	1byte	1byte	1byte

1.Start code: 1byte，一个数据包的开始标志，是个固定的值（0x70）。

2.Command: 1byte，命令，具体命令请参照指纹模块命令表。

3.Param: 4byte，表示 User ID 或系统参数。

4.Size: 4byte，用于表示数据包之后要传输的数据长度。

5.Flag/Error: 1byte，当 Host 到模块传输时为 Flag，用于区分传输的数据，当模块到 Host 时，为 Error，表示命令执行的结果。

6.Checksum: 1byte，用于检验是否为有效的数据包。Checksum 是从 Start code 到 Flag/Error 每个字节相加之和的低 8 位，即 Checksum 之前的所有字节之和除以 256 的余数。（备注：有些命令在传输时，也会带 Checksum，但是有些是 2Byte，有些是 4Byte，不论 Checjksum 的长度是多少，其计算方法都是一样的。）

7.End code: 1byte，数据包的结束标志。是个固定值（0x0A）；

备注：

1.主机和模块之间的通信采用的是相同的数据包格式。一般情况下都是 Host 发送一个命令，Module 就回复一个应答，主机到模块的数据包我们称之为“请求数据包”，模块到主机的数据包我们称之为“应答数据包”。但是在某种条件下，一个请求数据包会接收到两个应答数据包，例如 MD_ENROLL_SCAN 命令。

2.数据传输时，采用 little endian，最低位的字节最先被传输。例如要传输一个 0x764308 的 4byte 数据时，字节的传输顺序是：0x08 0x43 0x76 0x00。这种传输规则，主机和模块都必须遵从。

3.Checksum 的运算是 Checksum 之前的每个字节相加之后除以 256 的余数。例如：

Start	Command	Param	Size	Flag/Error	Checksum	End
0x70	0x15	0x03	0x200	0x70	0xFA	0x0A

$0x70+0x15+0x03+0x00+0x00+0x00+0x00+0x02+0x00+0x00+0x70=0$

xFA

Checksum=0xFA&0xFF = 0xFA

此数据包在串口的传输顺序是：0x70 0x15 0x03 0x00 0x00 0x00 0x00 0x02

0x00 0x00 0x70 0xFA 0x0A

1.2 扩展数据包传输协议

当模块和主机之间传输的数据量比较大时，最好采用扩展传输协议，它可以将批量的数据分割成若干个小的数据包，提高数据传输的可靠性，减少数据传输的错误等。

注意：

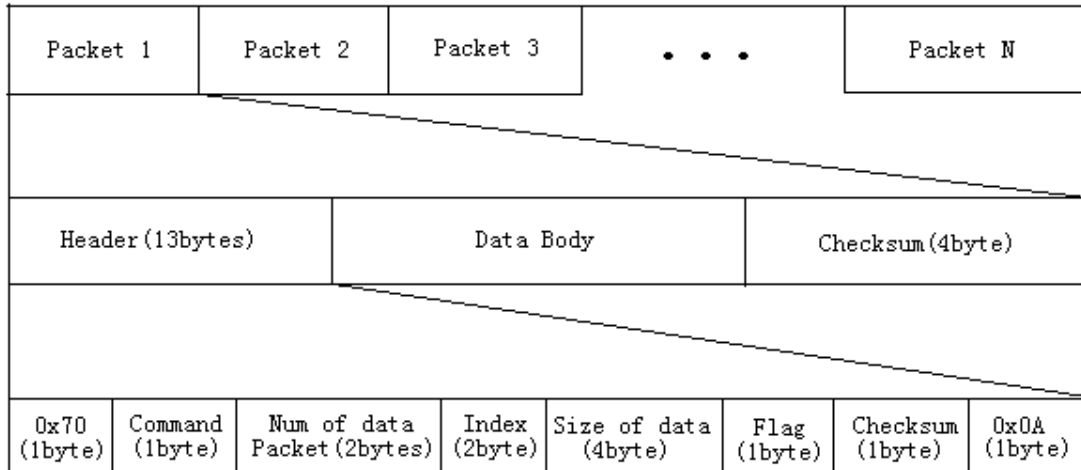
指纹模块可接收或传送的最大数据包为 **4kbyte**，即，主机发送到指纹模块的最大数据包字节长度不得超出 **4kbyte**。

当主机从指纹模块读取批量数据时，在请求命令中，可以声明传输数据包的最大长度（小于 **4Kbyte**），如果没有声明传输数据包的最大长度，指纹模块将以默认的数据包长度（**4Kbyte**）的大小来传输数据。

每个数据包都有 **4byte** 的 checksum，checksum 的值是 **Data Body** 的每个字节的数据之和。如果传输中有任何一个数据包发生错误，指纹模块就会立刻推出数据接收模式，进入普通模式。

数据包格式：

Start	Command	PacketNum	Index	Size	Flag/Error	Checksum	End
1byte	1byte	4bytes		4bytes	1byte	1byte	1byte



例如：使用 MD_ENROLL_IMAGE_X 命令要传输一个 16300 bytes 的数据到模块时，每个数据包的最大长度为 4096byte，那么 16300bytes 将被分割成 4 个小的数据包，4096bytes，4096bytes，4096bytes，4012bytes 被传输。

Packet 1 0x70 0x80 0x04 0x00 0x1000 0x00 0xD4 0x0A

Packet 2 0x70 0x80 0x04 0x01 0x1000 0x00 0xD5 0x0A

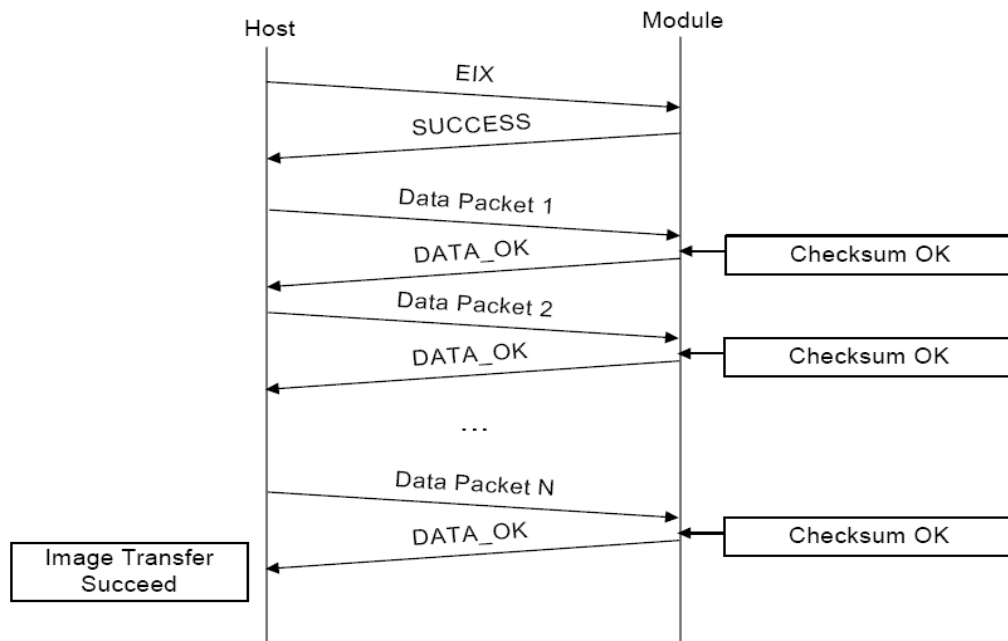
Packet 3 0x70 0x80 0x04 0x02 0x1000 0x00 0xD6 0x0A

Packet 4 0x70 0x80 0x04 0x03 0x0FAC 0x00 0x82 0x0A

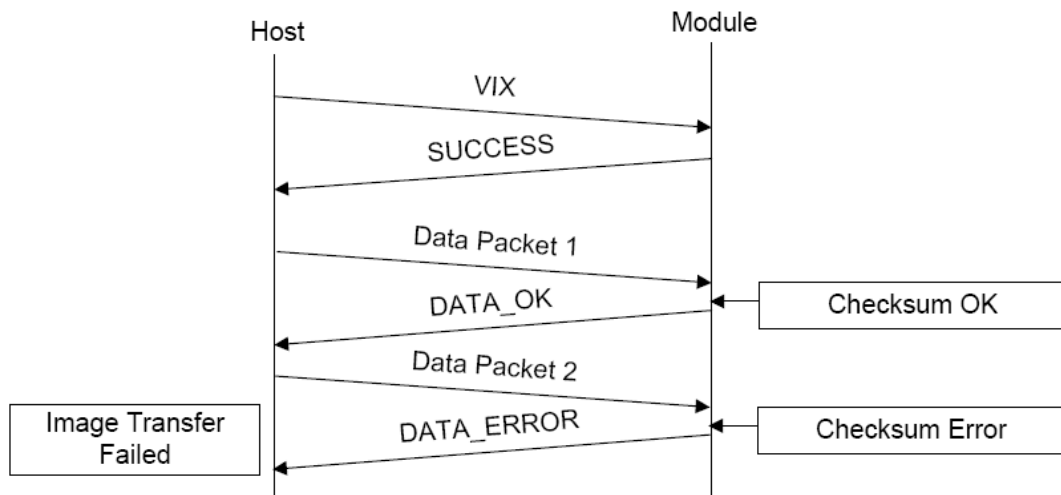
模块成功接收一个数据时，就会返回 DATA_OK(0x83)，否则返回 DATA_ERROR(0x82)。如果发送数据方接收到 DATA_ERROR(0x82)，就要立刻推出数据发送模式。

例如：

1. MD_ENROLL_IMAGE_X 的数据传输过程



2. MD_IDENTIFY_IMAGE_X 命令传输失败的过程



2. 模块的应用方式

模块支持三种不同的应用方式。其应用方式为：验证模式、读头（图片）模式、读头（模板）模式。根据不同的应用场景进行选择。

2.1 验证模式

模块默认使用此方式。在此方式下，用户按压指纹后，模块立即对该按压的指纹进行 1:N 识别，与此同时模块会发送结果到 Host。在此种模式下模板格式只支持 ZKTeco 模板格式。

2.2 读头(图片)模式

在此方式下，模块输出指纹图像到 Host，Host 接收到指纹图像后进行相关业务逻辑处理。

2.3 读头(模板)模式

在此方式下，模块输出指纹模板到 Host，Host 接收到指纹模板后进行相关业务逻辑处理。模板数据的格式根据设置的格式所决定。

3. 系统配置

系统参数是用来打开或关闭某一功能，或修改某一命令执行时的有效时长等而设定的。每一个参数对应一个参数 ID，要读取或修改参数值时，只需指定对应的参数 ID，就可以对其操作了。

每一个参数都可以通过命令 MD_SYS_RP，读取其值。但有些参数是可以读写的，有些参数为只读，所有可改写的参数都可以通过命令 MD_SYS_WP 对其改写。要想修改的系统参数在下次重启后有效，还需要使用 MD_SYS_SP 命令对其保存，即当需要修改一些系统参数值时，先把要修改的参数全部修改完成之后，再使用命令 MD_SYS_SP 对其保存，以便重启之后有效。具体的读写过程请参照本文档中的 MD_SYS_WP 和 MD_SYS_RP 命令说明。

Table 1 系统配置参数表(如何配置系统参数详见命令 3.1、3.2、3.3)

配置参数名称	参数 ID	描述	值（*表示默认值）
保存验证记录	0x36	当执行指纹验证或识别时，是否保存记录	*0x30：不保存 0x31：保存
用户容量	0x79	最大可登记的用户数量	4byte 的整数
记录数量	0x7C	当前保存的验证记录数量	4byte 的整数

记录容量	0x7B	最大可保存的验证记录数量	4byte 的整数
指纹数量	0x73	当前已登记的指纹数量	4byte 的整数
指纹容量	0x74	最大可登记的指纹数量	4byte 的整数
超时	0x62	超时时间设定	0x30 : 永不超时 0x31 : 1 second ... *0x3A : 10 seconds ... 0x44 : 20 seconds ...
波特率	0x71	波特率的设定	*0x35 : 115200 bps
自动应答	0x82	自动应答	*0x30 : 无应答 0x31 : 发送应答
模块 ID	0x6D	模块 ID	0 ~ 65535
工作模式	0x50	验证模式: 在模块内部进行指纹比对 读头(图片): 发送按压的指纹图像到 host 进行处理 读头(模板): 发送按压手指的指纹模板到 host 进行处理	*0x30: 验证模式 0x31: 读头(图片) 0x32: 读头(模板)
模板格式	0x51	设置模板格式, 模块内部只能识别 ZKTeco 格式模板	*0x30: ZK 0x32: Ansi 378 0x33: ISO 19794-2
闪灯控制	0x31	控灯操作	高 8 位表示灯类型 0x80: 绿灯, 0x40: 红灯, 0xC0: 黄灯 低 8 位表示灯的显示时长, 范围: 1~255, 单位为秒

固件版本号	0x6E	固件版本号	4 个 byte 数据
创建日期	0x89	创建日期	4byte 的数据

3.1 保存验证记录

当执行指纹验证或识别时，是否保存记录。验证的记录可被用作考勤的依据，当使用模块做考勤机时，此参数是很有用的。当被用于门禁或者实时监控设备时，如果需要保存验证记录，就可以将此参数打开。

注意：

模块本身没有时钟芯片，每次重启后，时间归 0。要确保验证记录的准确性，每次连接模块时，请同步时钟。并且间隔一小时同步一次。

参数 ID	读写类型	有效值	默认值
0x36	Read/Write	0x30: 不保存验证记录 0x31: 保存验证记录	0x30: 不保存验证记录

3.2 用户容量

可支持的最大用户容量。

注：不同的版本，可容纳的最大用户数是不同的。

参数 ID	读写类型	有效值	默认值
0x79	Read		

3.3 记录数量

表示当前已保存的记录数量。

参数 ID	读写类型	有效值	默认值
0x7C	Read		

3.4 记录容量

最大可容纳的记录数量。不同的版本，可容纳的最大记录数量是不同的。

参数 ID	读写类型	有效值	默认值
-------	------	-----	-----

0x7B	Read		
------	------	--	--

3.5 指纹数量

表示当前已登记的指纹数量。

参数 ID	读写类型	有效值	默认值
0x73	Read		

3.6 指纹容量

表示最大可登记的指纹数量。不同的版本，可登记的最大指纹数量是不同的。

参数 ID	读写类型	有效值	默认值
0x74	Read		

3.7 超时

超时时间设定。

登记指纹时，模块发出应答命令之后，在设定时间内没有再接到下一步动作的请求命令时，就会自动退出当前的登记状态，并向主机发送超时应答。

扩展数据传输时，两个数据包或连续的命令之间的等待时间不得超过本设定值，否则，模块自动超时返回，并向主机发送超时应答。

参数 ID	读写类型	有效值	默认值
0x62	Read/Write	0x30 : 永超时 0x31 : 1 second ... *0x3A : 10 seconds ... 0x44 : 20 seconds ...	0x3A:10sec

3.8 波特率

设定 RS232 通讯的波特率。默认采用的是 115200bps。

参数 ID	读写类型	有效值	默认值
0x71	Read	*0x35:115200 bps	*0x35 : 115200 bps

3.9 自动应答

在此模式中，用于设置模块即使在没有接受到任何 host 请求命令的情况下。模块是否需要向主机自动发送应答数据包。

参数 ID	读写类型	有效值	默认值
0x82	Read/Write	*0x30: 无应答 0x31:发送应答	*0x30:无应答

3.10 模块 ID

每个模块都会给默认一个 ID 号，系统同时接多个模块时，此参数非常有用。

参数 ID	读写类型	有效值	默认值
0x6D	Read/Write	0 ~ 65535	1

3.11 固件版本号

固件版本号是使用 4 个字符来表示的。例如 V1.0A，其值为 0x56313041—对应的 ASCII 是 ‘V’，‘1’，‘0’，‘A’。

参数 ID	读写类型	有效值	默认值
0x6E	Read	4 个 byte 字符	

3.12 创建日期

4byte 的字符的数据，表示固件的创建日期。例如 2008—12—30，其值为 0x08123000。

参数 ID	读写类型	有效值	默认值
0x89	Read		

3.13 工作模式

模块支持三种工作模式：验证模式、读头（图片）、读头（模板）

参数 ID	读写类型	有效值	默认值
-------	------	-----	-----

0x50	Read/Write	*0x30: 验证模式 0x31: 读头(图片) 0x32: 读头(模板)	*0x30: 验证模式
------	------------	---	-------------

3.14 模板格式

模块支持三种模板格式：ZK、Ansi、ISO。

参数 ID	读写类型	有效值	默认值
0x51	Read/Write	*0x30: ZK 0x32: Ansi 378 0x33: ISO 19794-2	*0x30: ZK

3.15 闪灯控制

控制红、黄、绿灯的显示。

参数 ID	读写类型	有效值	默认值
0x31	Write	高 8 位表示灯类型 0x80: 绿灯, 0x40: 红灯, 0xC0: 黄灯 低 8 位表示灯的显示时长, 范围: 1~255, 单位为秒	

4. 模块操作命令

Tab2 命令表

种类	名称	代码	描述
系统配置	MD_SYS_WP	0x11	写系统参数
	MD_SYS_SP	0x12	保存系统参数
	MD_SYS_RP	0x13	读系统参数
	MD_SYS_STATUS	0x14	查询系统状态
登记	MD_ENROLL_SCAN	0x15	通过三次扫描指纹登记用户指纹模板
	MD_ENROLL_IMAGE_X	0x80	通过指纹图像登记指纹模板
	MD_ENROLL_TMP	0x17	通过指纹模板登记用户
验证	MD_VERIFY_SCAN	0x18	1: 1 扫描验证用户模板是否存在
识别	MD_IDENTIFY_IMAGE_X	0x81	通过指纹图像进行识别用户
	MD_IDENTIFY_FREE	0x2F	FreeScan 模式下的扫描识别
删除	MD_DEL_ALL_TMP	0x27	删除所有指纹模版
	MD_DEL_TMP	0x26	删除指定用户的指纹模板
指纹模版	MD_READ_TMP_X	0x89	读取指定用户的指纹模板
	MD_SCAN_TEMPLATE	0xFC	扫描当前手指的指纹模板
用户	MD_ADD_USER	0xF1	添加一个用户
	MD_READ_USER	0xF2	读一个用户
	MD_DELETE_USER	0xF3	删除用户
	MD_DEL_ALL_USER	0xF5	删除所有用户
指纹图像	MD_SCAN_IMAGE_X	0x83	从模块中获取指纹图像
时间和记录	MD_SET_TIME	0x4A	设置当前时间
	MD_GET_TIME	0x4B	获取模块时间
	MD_DEL ALOG	0x9E	删除所有记录
	MD_LOAD_LOG_X	0xA4	读所有的记录数据
数据库	MD_LOAD_USER_X	0xA0	读所有用户数据
	MD_LOAD_TMP_X	0xA2	读所有指纹模版数据
	MD_DEL_DB	0xF8	删除所有数据
	MD_WT_FILE_X	0xAA	上传指纹模板存储文件
	MD_WU_FILE_X	0x42	上传用户信息存储文件
	MD_RT_FILE_X	0xAB	下载指纹模板存储文件
	MD_RU_FILE_X	0x43	下载用户信息存储文件
升级固件	MD_UPDATE_FW	0x72	升级固件版本
模块复位	MD_RESET	0xD0	进行模块的软复位
启用模块	MD_ENABLEDEVICE	0xFB	启用模块
禁用模块	MD_DISABLEDEVICE	0xFA	禁用模块

备注:

在介绍命令时, # 表述一个变化的值, NULL 表示 0, N/A 表示任意值。

4.1 Flag/Error 表

名称	代码	描述
BUSY	0x34	系统正在忙
SUCCESS	0x61	成功
FAIL	0x63	失败
TIME_OUT	0x64	超时
PARAM_ERROR	0x68	参数错误
NOT_FOUND	0x69	找不到或不支持
MEM_FULL	0x6D	存储容量超出
FP_LIMIT	0x72	超出允许登记的最大指纹数量（10）
INVALID_ID	0x76	无效的 ID 号
CANCELED	0x81	执行的命令被取消
DATA_ERROR	0x82	传输数据错误
EXIST_FINGER	0x86	已经存在的指纹

4.2 MD_SYS_WP

参数设置。改变当前模块系统参数的值，但是没有保存修改的参数，下次上电重启时，参数将恢复原来的值。 如果要想保存系统参数，在重启之后已让有效，需要执行完 MD_SYS_WP 命令之后，还需执行 MD_SYS_SP 命令。

请求命令

数据域	数值	描述
Start code	0x70	
Command	0x11	
Param	NULL	
Size	NULL	
Flag	0x31	SID_GPIO_LEVEL
	0x36	SID_SAVE_LOG
	0x82	SID_AUTO_ACK
	0x62	SID_TIMEOUT
Checksum	#	
End code	0x0A	

应答命令

数据域	数值	描述
Start code	0x70	
Command	0x11	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
	0x69	NOT_FOUND
	0x68	PARAM_ERROR
	0x34	BUSY
Checksum	#	
End code	0x0A	

Error

Error code	描述
SUCCESS	写参数成功
NOT_FOUND	没有找到此参数，或不支持此参数
BUSY	模块正忙于处理其他的命令

4.3 MD_SYS_SP

保存系统参数。将系统参数保存到文件中，重启时，被命令 MD_SYS_WP 改变的参数保持有效。

请求命令

数据域	数值	描述
Start code	0x70	
Command	0x12	
Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	

End code	0x0A	
----------	------	--

应答命令

数据域	数值	描述
Start code	0x70	
Command	0x12	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
	0x34	BUSY
Checksum	#	
End code	0x0A	

4.4 MD_SYS_RP

读系统参数，根据参数 ID 读对应的参数值。

请求命令

数据域	数值	描述
Start code	0x70	
Command	0x13	
Param	NULL	
Size	NULL	
Flag	0x36	SID_SAVE_LOG,
	0x82	SID_AUTO_ACK,
	0x62	SID_TIMEOUT,
	0x6E	SID_FW_VER,
	0x71	SID_BAUDRATE,
	0x73	SID_ENROLL_FP,
	0x74	SID_FP_COUNT,
	0x79	SID_USER_COUNT,
	0x7C	SID_LOG_NUM,
	0x7B	SID_LOG_COUNT
	0x89	SID_BUILD_NUM
	0x6D	SID_MODULE_ID

Checksum	#	
End code	0x0A	

应答命令

数据域	数值	描述
Start code	0x70	
Command	0x13	
Param	Value NULL	Value-成功返回参数值 NULL-读取失败
Size	NULL	
Error	0x61 0x69 0x34	SUCCESS NOT_FOUND BUSY
Checksum	#	
End code	0x0A	

4.5 MD_SYS_STATUS

查询系统状态。查询系统当前的状态，是空闲还是忙于处理其他的命令。

请求命令

数据域	数值	描述
Start code	0x70	
Command	0x14	
Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

应答命令

数据域	数值	描述
-----	----	----

Start code	0x70	
Command	0x14	
Param	0x34	BUSY
Size	NULL	
Error	0x61	SUCCESS
Checksum	#	
End code	0x0A	

4.6 MD_ENROLL_SCAN

登记用户指纹，通过扫描的方式登记，即发送登记命令之后，等待用户将登记的手指放在指纹采集器上，系统扫描到用户按下手指，就会开始为用户登记指纹。

用户需要按三次指纹，才能登记一个指纹模版。每次按下指纹成功扫描到时，模块都会闪一下黄灯，并会向 **Host** 发送一条通知命令。成功登记后会闪一下绿灯.登记失败时，会闪一下红灯。

请求命令格式如下：

数据域	数值	描述
Start code	0x70	
Command	0x15	
Param	User ID	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

应答命令

数据域	数值	描述
Start code	0x70	
Command	0x15	
Param	User ID	

Size	Finger ID	(0~9)
Flag/Error	0x62	SCAN_SUCCESS
	0x63	FAIL
	0x61	SUCCESS
	0x6D	MEM_FULL
	0x72	FINGER_LIMIT
	0x76	INVALID_ID
	0x86	EXIST_FINGER
Checksum	#	
End code	0x0A	

Error code

Error code	描述
SCAN_SUCCESS	成功扫描到了指纹
FAIL	失败
SUCCESS	登记成功
MEM_FULL	模块内存储器已满
INVALID_ID	无效的用户 ID
EXIST_FINGER	指纹已经存在
FP_LIMIT	登记的指纹超出范围（10）

例如：当为 ID 号为'0x023,'的用户登记一个指纹时，请求数据包如下：

Start	Command	Param	Size	Flag	Checksum	End
0x70	0x15	0x0023	0x00	0x00	0xA8	0x0A

4.7 MD_ENROLL_IMAGE_X

通过指纹图像登记。通过主机传输的指纹图像来为指定的用户登记一枚指纹。对于指纹图像的传输，使用扩展通信协议格式。

注意：

1. 在登记的过程中，可能会出现，指纹模板没有登记成功，但指定的用户被创建的情况，这属于正常情况。

2. 指纹图像必须是 BMP 格式 8Bit 的灰度图像，最好使用本模块读取的图像来登记，否则会影响识别率。并且，此处传输的只是图像真实的数据，并不是整幅 BMP 图像，即，不包括 BMP 的头（BMP headers）和调色板数据（Palette Data），仅仅是图像数据（Bitmap Data）。

请求命令格式如下：

数据域	数值	描述
Start code	0x70	
Command	0x80	
Param	User ID	
Size	Image size	指纹图像大小
Flag	NULL	
Checksum	#	
End code	0x0A	

应答命令

数据域	数值	描述
Start code	0x70	
Command	0x80	
Param	User ID	
Size	NULL	
Error	0x61	SUCCESS
	0x63	FAIL
	0x6C	TIME_OUT
	0x6D	MEM_FULL
	0x72	FP_LIMIT
	0x76	INVALID_ID

Checksum	#	
End code	0x0A	

4.8 MD_IDENTIFY_IMAGE_X

通过指纹图像进行识别用户。识别方式为 1: N。数据包传输过程中使用扩展传输方式。

请求命令格式如下：

数据域	数值	描述
Start code	0x70	
Command	0x81	
Param	Image size	图像数据大小
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

应答命令

数据域	数值	描述
Start code	0x70	
Command	0x81	
Param	Param NULL	Param-识别成功时： 用户 ID 为 0xFFFF & Param, 指纹索引为(Param >> 16) & 0x0F NULL-识别失败
Size	NULL	
Error	0x61 0x63 0x6C 0x69	SUCCESS FAIL TIME_OUT NOT_FOUND

Checksum	#	
End code	0x0A	

4.9 MD_ENROLL_TMP

通过指纹模板登记，即上传指纹模板到模块中。通过从主机传输来的指纹模板给用户登记一枚指纹，此处传输的是真实的指纹模板数据，并不是指纹模板的整个结构体数据。

注意，传输的指纹模板数据长度不得大于模块支持的最大指纹模板长度，否则，返回PARAM_ERROR。

请求命令格式如下：

数据域	数值	描述
Start code	0x70	
Command	0x17	
Param	User ID	
Size	Template size	
Flag	0 0x84	0-None 0x84-CHECK_FINGER
Checksum	#	
End code	0x0A	
Data	Template Data	模板数据
Checksum of data	#	模板数据的校验值

应答命令

数据域	数值	描述
Start code	0x70	
Command	0x17	
Param	User ID	
Size	NULL	
Error	0x61	SUCCESS

	0x6D	MEM_FULL
	0x72	FINGER_LIMIT
	0x76	INVALID_ID
	0x63	FAIL
	0x86	EXIST_FINGER
Checksum	#	
End code	0x0A	

4.10 MD_VERIFY_SCAN

扫描指纹模板验证用户是否存在。通过 1:1 的扫描方式来验证一个用户。

如果保存记录参数打开，模块就会自动保存记录，并返回当前记录的状态。

请求命令格式如下：

数据域	数值	描述
Start code	0x70	
Command	0x18	
Param	User ID	
Size	NULL	
Flag	0x61	
Checksum	#	
End code	0x0A	

应答命令

数据域	数值	描述
Start code	0x70	
Command	0x18	
Param	User ID	
Size	NULL	
Error	0x63	FAIL
	0x6C	TIME_OUT

	0x61	SUCCESS
Checksum	#	
End code	0x0A	

4.11 MD_IDENTIFY_FREE

1:N 用户识别。当用户在指纹采集器上按压手指时，模块会自动识别该手指，并将比对结果发送到 host。

模块发送命令：

数据域	数值	描述
Start code	0x70	
Command	0x2F	
Param	NULL	用户编号：识别成功 错误代码：识别失败
Size	NULL	
Error	0x61 0x63	0x61-SUCCESS 0x63-FAIL
Checksum	#	
End code	0x0A	

4.12 MD_DEL_ALL_TMP

删除所有指纹模板，同时删除所有用户数据。

请求命令格式如下：

数据域	数值	描述
Start code	0x70	
Command	0x27	
Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

应答命令

数据域	数值	描述
Start code	0x70	
Command	0x27	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
Checksum	#	
End code	0x0A	

4.13 MD_DEL_TMP

只删除用户所属的指纹模板。

请求命令格式如下：

数据域	数值	描述
Start code	0x70	
Command	0x26	
Param	UserID、Finger index	Param 域数据为:(用户 ID & 0xFFFF) ((指纹索引 & 0x0F) << 16)
Size	NULL	
Flag	NULL(默认) DELETE_ONLY_ONE (0x77)	删除用户和用户指纹 删除用户的一个指纹
Checksum	#	
End code	0x0A	

应答命令

数据域	数值	描述
Start code	0x70	
Command	0x26	

Param	Param	
Size	NULL	
Error	0x61 0x63	SUCCESS FAIL
Checksum	#	
End code	0x0A	

4.14 MD_READ_TMP_X

根据指定的用户 ID 和指纹索引，从模块中下载其指纹模板到 host。

请求命令：

数据域	数值	描述
Start code	0x70	
Command	0x89	
Param	User ID+Index	
Size	NULL	
Flag	0 1	0-忽略 Finger ID。 1-根据指纹索引下载
Checksum	#	
End code	0x0A	

应答命令：

数据域	数值	描述
Start code	0x70	
Command	0x89	
Param	NULL	
Size	NULL	
Error	0x61 0x63 0x69	SUCCESS FAIL NOT_FIND

	0x6C	TIME_OUT
	0x82	DATA_ERROR
Checksum	#	
End code	0x0A	

注意：

每次传输的指纹模板长度，是不定长的，即，每个指纹的模板长度都不一定相等。但是，一次传输多枚指纹模板的长度是相等的。

例如 1：读用户的一个指纹，User ID=1，Finger ID=1，数据包长度为 size=2048Byte；请求数据包如下：

Start	Command	Param	Size	Flag	Checksum	End
0x70	0x89	0x01	0x10800	0x01	0x04	0x0A

例如 2：读用户的所有指纹，User ID=1，数据包长度为 size=2048Byte；请求数据包如下：

Start	Command	Param	Size	Flag	Checksum	End
0x70	0x89	0x01	0x800	0x00	0x02	0x0A

4.15 MD_ADD_USER

在模块文件中添加一个用户，如果用户存在，就修改该用户的信息。

用户的数据结构如下：（使用一个字节内存对齐的方式）

```
typedef struct _User_
{
    U16 ID;           /* 用户 ID */
    U8  Privilege;     /* 用户权限 */
    U8  FpNum;         /* 用户登记的指纹数(自维护) */
    U16 SecLevel;      /* 用户加密等级 (本版本不支持) */
    U32 PIN2;          /* 用户编号(本版本不支持) */
    Char Name[8];      /* 用户姓名 */
}
```

```

Char Password[5]; /* 密码 (本版本不支持) */
U8 Card[5]; /* 卡 ID(本版本不支持) */

}

```

请求命令格式如下：

数据域	数值	描述
Start code	0x70	
Command	0xF1	
Param	NULL	
Size	传输数据的长度	用户结构长度+校验数据大小
Flag	0x61	
Checksum	#	
End code	0x0A	
Data	User Data	用户数据
Checksum of data	#	用户数据的校验值

应答命令：

数据域	数值	描述
Start code	0x70	
Command	0xF1	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
	0x63	FAIL
	0x6D	MEM_FULL
	0x68	PARAM_ERROR
Checksum	#	
End code	0x0A	

4.16 MD_READ_USER

读一个存储在模块文件中的用户。

请求命令格式如下：

数据域	数值	描述
Start code	0x70	
Command	0xF2	
Param	User ID	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

应答命令：

数据域	数值	描述
Start code	0x70	
Command	0xF2	
Param	NULL	
Size	传输的数据长度	用户结构长度+校验数据大小
Error	0x61	SUCCESS
	0x69	NOT_FIND
Checksum	#	
End code	0x0A	
Data	User Data	用户数据
Checksum of data	#	用户数据的校验值

4.17 MD_DELETE_USER

删除用户和用户的指纹，使用此命令可以删除一个用户。

请求命令格式如下：

数据域	数值	描述
-----	----	----

Start code	0x70	
Command	0xF3	
Param	User ID	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

应答命令：

数据域	数值	描述
Start code	0x70	
Command	0xF3	
Param	UserID	
Size	NULL	
Error	0x61	SUCCESS
	0x63	FAIL
Checksum	#	
End code	0x0A	

例如 1：删除 User ID =1 的用户

请求数据包：

Start	Command	Param	Size	Flag	Checksum	End
0x70	0XF3	0x01	0x00	0x00	0x64	0x0A

应答数据包：

Start	Command	Param	Size	Flag	Checksum	End
0x70	0XF3	0x01	0x00	0x61	0XC5	0x0A

4.18 MD_DEL_ALL_USER

删除存储在模块文件中所有用户和指纹数据。在删除所有用户的同时，会删除所有指纹模板信息。

请求命令格式如下：

数据域	数值	描述
Start code	0x70	
Command	0xF5	
Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

应答命令：

数据域	数值	描述
Start code	0x70	
Command	0xF5	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
Checksum	#	
End code	0x0A	

4.19 MD_SCAN_IMAGE_X

从模块获取手指的指纹图像。发送此命令后，需要用户在模块上按手指，当成功按下手指后，模块会以扩展协议发送指纹图像到 Host。

图像的数据结构体如下：

```
typedef struct _Image_
{
    int width;           /* 图像的宽度 */
    int height;          /* 图像高度 */
    int compressed;      /* 当前此值没有使用 */
    int encrypted;       /* 数据加密的图像，当前没有使用 */
    int binary;          /* 图像的数据格式，默认0当前没有使用 */
    int img_len;         /* 图像的长度 */
    int template_len;    /* 此图像生成的指纹模版长度，当前没有使用 */
}
```

```
};
```

请求命令格式如下：

数据域	数值	描述
Start code	0x70	
Command	0x83	
Param	Data Size	数据包大小，默认为 4K
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

应答命令：

数据域	数值	描述
Start code	0x70	
Command	0x83	
Param	NULL	
Size	Image size	
Error	0x61	SUCCESS
	0x6C	TIME_OUT
	0x6D	MEM_FULL
	0x63	MD_FAIL
	0x81	CANCELED
Checksum	#	
End code	0x0A	

4.20 MD_SET_TIME

向模块写入当前的时间。指纹模块没有时钟芯片，模块的时钟必须由主机来设定，可以间隔一小时设置一次模块的时钟。

Data 参数中的年份，为当前的年份加上 2000，例如 2008 年，这个 **Data** 中的年份值为 8。

请求命令格式如下：

数据域	数值	描述
Start code	0x70	
Command	0x4A	
Param	(DD<<16) (MM<<8) YY	Date
Size	8	数据大小+校验数据大小
Flag	NULL	
Checksum	#	
End code	0x0A	
Data	(ss<<16) (mm<<8) hh	Time
Checksum of Data	#	数据的校验值

应答命令

数据域	数值	描述
Start code	0x70	
Command	0x4A	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
	0x63	FAIL
Checksum	#	
End code	0x0A	

4.21 MD_GET_TIME

读模块当前的时间。

请求命令格式如下：

数据域	数值	描述
Start code	0x70	

Command	0x4B	
Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

应答命令

数据域	数值	描述
Start code	0x70	
Command	0x4B	
Param	(DD<<16) (MM<<8) YY	Date
Size	8 NULL	8-读取成功的数据大小 NULL-读取失败时的值
Error	0x61 0x63	SUCCESS FAIL
Checksum	#	
End code	0x0A	
Data	ss<<16 mm<<8 hh	Time
Checksum of data	#	

注意:当以 YYYY-MM-DD 显示日期时, 对于年份需要加上 2000.

4.22 MD_DEL ALOG

删除所有存储在模块中的所有验证记录数据。

请求命令格式如下:

数据域	数值	描述
Start code	0x70	
Command	0x9E	

Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

应答命令

数据域	数值	描述
Start code	0x70	
Command	0x9E	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
Checksum	#	
End code	0x0A	

4.23 MD_LOAD_LOG_X

发送此命令后，使用扩展传输的方式，读取模块中的验证记录全部数据。读验证记录数据，验证记录的存储是以一定的数据结构体来存储的。并以文件的方式存储在模块中。

```

Typedef struct _Log_{
    BYTE  Event;
    BYTE  verified;
    U32   Date;
    U32   Time;
    U16   userID;
    Char  Reserved[4];
};

```

数据域	字节数	描述
Event	1 byte	存储记录时发生的事件

		参看下面的事件表。
Verified	1 byte	验证的方式，指纹还是其他方式，此域可以忽略不管，之后的模块版本中，会使用到。
Date	4 byte	YY MM DD
Time	4 byte	hh mm ss
User ID	2 byte	User ID
Reserved	4 byte	此域可以使用 MD_LC 命令来读取或设置。

注意：

年份：YY+2000，例如 2008 年，YY=8；但是如果模块时间没有被设置，YY 的值将是永远为 0。 Time 也将会是不准确的错误的值。

记录的事件代码表：

事件	代码	描述
LOG_FREE	0	使用 Free Scan 的方式验证的记录
LOG_SV	1	使用 Scan Verify 命令保存的验证记录
LOG_TV	2	使用 template verify 命令保存的验证记录
LOG_IV	3	使用 image verify 命令保存的验证记录
LOG_SI	4	使用 scan identify 命令保存的验证记录
LOG_TI	5	使用 template identify 命令保存的验证记录
LOG_II	6	使用 image identify 命令保存的验证记录

请求命令格式如下：

数据域	数值	描述
Start code	0x70	
Command	0xA4	
Param	Data size	数据大小，最大为 4k
Size	NULL	
Flag	NULL	

Checksum	#	
End code	0x0A	

应答命令

数据域	数值	描述
Start code	0x70	
Command	0xA4	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
	0x63	FAIL
	0x6C	TIME_OUT
	0x69	NOT_FIND
	0x82	DATA_ERROR
Checksum	#	
End code	0x0A	

4.24 MD_LOAD_USER_X

读所有用户数据。每个用户的数据都是以用户结构体的格式存储在指纹模块中，MD_LOAD_USER_X 就是将每个用户的用户结构体通过扩展传输的方式读取到主机。

请求命令格式如下：

数据域	数值	描述
Start code	0x70	
Command	0xA0	
Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

应答命令

数据域	数值	描述
Start code	0x70	
Command	0xA0	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
	0x6C	TIME_OUT
	0x63	FAIL
	0x69	NOT_FOUND
	0x82	DATA_ERROR
Checksum	#	
End code	0x0A	

4.25 MD_LOAD_TMP_X

读取所有指纹模版数据从模块。每个用户的指纹模版都是以指纹模版结构体的格式存储在指纹模块中，MD_LOAD_TMP_X 就是将每个指纹模版结构体中相关数据按照一定的顺序传输到主机。

```
typedef struct _Template_
{
    U16   Size;           /* 指纹模版的实际长度          */
    U16   ID              /* User ID, 即用户 PIN          */
    BYTE  FingerID;       /* 指纹 ID (Sub_ID or Finger ID) */
    BYTE  Valid;          /* 指纹模版是否有效  1: 有效; 0: 无效 */
    BYTE  Template[1664]; /* 指纹模版数据（最大长度为 1664B） */
};
```

注意：

以上结构都采用 1Byte 的内存对齐格式；

不同手指登记的指纹模板长度不一定相等，即，指纹模板的实际长度不是一个定值。

传输指纹模板时，为了加快传输速度，传输的指纹数据不是指纹模板结构体的长度，而是 Size+ID+FingerID+Valid+Template(实际长度)；

例如，传输如下两枚指纹：

指纹1： Size=1024, ID=1, FingerID=0, Valid=1, Template= (1024Byte 模板数据)；

指纹2： Size=1660, ID=2, FingerID=1, Valid=1, Template= (1660Byte 模板数据)；

那么使用 ReadAllTemplates 命令传输指纹数据时，传输的 16 进制数据包为：

00 04 01 00 00 01 （指纹 1 的 1024Byte 模板数据） 7C 06 02 00 01 01 （指纹 2 的 1660Byte 模板数据）；

即：传输的指纹数据总长度为：（6Byte+1024Byte）+（6Byte+1660Byte）=2696Byte；

需要注意的是：

传输的指纹数越多，耗费的时间越长，请参照本文档下面的注意事项部分。

请求命令格式如下：

数据域	数值	描述
Start code	0x70	
Command	0xA2	
Param	Pakcet size	传输数据包的大小
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

应答命令

数据域	数值	描述
-----	----	----

Start code	0x70	
Command	0xA2	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
	0x6C	TIME_OUT
	0x63	FAIL
	0x69	NOT_FOUND
	0x82	DATA_ERROR
Checksum	#	
End code	0x0A	

4.26 MD_DEL_DB

删除指纹模块中所有的数据，包括用户数据，指纹模版数据，验证记录数据。

请求命令格式如下：

数据域	数值	描述
Start code	0x70	
Command	0xF8	
Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

应答命令

数据域	数值	描述
Start code	0x70	
Command	0xF8	

Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
Checksum	#	
End code	0x0A	

4.27 MD_SCAN_TEMPLATE

扫描指纹模板。在指纹采集器上按手指后 host 从模块中读取该指纹模板，模板最大长度为 2K。执行此命令前，需将系统命令 0x50 的参数值设置为 0x30，然后模块等待用户按压手指。host 需要一直调用此函数来读取指纹模板。

注意：模块的指纹识别有两种方式。第一种是在模块中识别，第二种是模块生成指纹模板通过 host 进行识别。

请求命令格式如下：

数据域	数值	描述
Start code	0x70	
Command	0xFC	
Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

应答命令

数据域	数值	描述
Start code	0x70	
Command	0xFC	
Param	NULL	
Size	size	size-指纹模板长度，最大为

	NULL	2K NULL-无指纹模板
Error	0x61	SUCCESS
Checksum	#	
End code	0x0A	
Data	template	模板数据
Checksum of data	#	

注：此命令一直返回成功标志。

4.28 MD_DISCONNECT

断开连接。模块支持 232 和 usb 通信。模块同一时刻只能通过其中一种方式与 host 进行通信，因此模块在进行切换通信方式时必须断开当前的通信方式，若不断开，下种方式将无法连接。

请求命令

数据域	数值	描述
Start code	0x70	
Command	0xCC	
Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

应答命令

数据域	数值	描述
Start code	0x70	
Command	0xCC	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS

Checksum	#	
End code	0x0A	

4.29 MD_RESET

模块软复位。

请求命令

数据域	数值	描述
Start code	0x70	
Command	0xD0	
Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

应答命令

数据域	数值	描述
Start code	0x70	
Command	0xD0	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
Checksum	#	
End code	0x0A	

4.30 MD_ENABLEDEVICE

启用模块。当使用此命令后，在验证模式下模块将向 host 发送指纹比对结果信息。

请求命令

数据域	数值	描述
Start code	0x70	

Command	0xFB	
Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

应答命令

数据域	数值	描述
Start code	0x70	
Command	0xFB	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
Checksum	#	
End code	0x0A	

4.31 MD_DISABLEDEVICE

禁用模块。当使用此命令后，在验证模式下模块将不向 host 发送指纹比对结果信息。

请求命令

数据域	数值	描述
Start code	0x70	
Command	0xFA	
Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

应答命令

数据域	数值	描述
-----	----	----

Start code	0x70	
Command	0xFA	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
Checksum	#	
End code	0x0A	

4.32 MD_UPDATE_FW

升级固件。在升级处理中先发送固件大小到模块，当模块回复确认后，Host 紧接着以扩展数据格式进行数据传输。

请求命令格式如下：

数据域	数值	描述
Start code	0x70	
Command	0x72	
Param	firmware size	固件数据大小
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

应答命令

数据域	数值	描述
Start code	0x70	
Command	0x72	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
	0x63	FAIL
	0x6C	TIME_OUT

Checksum	#	
End code	0x0A	

4.33 MD_WT_FILE_X

上传指纹模板存储文件。在写的过程中先发送指纹模板存储文件的大小到模块，当模块回复确认后，Host 紧接着以扩展数据格式进行数据传输。

注意：如果指纹模板中的用户 ID 和用户数据中的用户 ID 不是一一对应关系，则用户数据将和指纹模版数据是不匹配的，进行指纹验证时会出错。即，要求传输的指纹模版和用户数据必须是一一对应的关系。

请求命令格式如下：

数据域	数值	描述
Start code	0x70	
Command	0xAA	
Param	templates size	指纹模板存储文件大小
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

应答命令

数据域	数值	描述
Start code	0x70	
Command	0xAA	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
	0x63	FAIL
	0x6C	TIME_OUT
Checksum	#	
End code	0x0A	

4.34 MD_WU_FILE_X

上传用户信息存储文件。在写的过程中先发送用户信息存储文件的大小到模块，当模块回复确认后，Host 紧接着以扩展数据格式进行数据传输。

注意：如果指纹模板中的用户 ID 和用户数据中的用户 ID 不是一一对应关系，则用户数据将和指纹模版数据是不匹配的，进行指纹验证时会出错。即，要求传输的指纹模版和用户数据必须是一一对应的关系。

请求命令格式如下：

数据域	数值	描述
Start code	0x70	
Command	0x42	
Param	Data size	用户信息存储文件大小
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

应答命令

数据域	数值	描述
Start code	0x70	
Command	0x42	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
	0x63	FAIL
	0x6C	TIME_OUT
Checksum	#	
End code	0x0A	

4.35 MD_RT_FILE_X

下载模块中的模板存储文件到 Host。在下载过程中以扩展协议传输。

请求命令格式如下：

数据域	数值	描述
Start code	0x70	
Command	0xAB	
Param	NULL	
Size	NULL	
Flag	NULL	
Checksum	#	
End code	0x0A	

应答命令

数据域	数值	描述
Start code	0x70	
Command	0xAB	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
	0x63	FAIL
	0x6C	TIME_OUT
Checksum	#	
End code	0x0A	

4.36 MD_RU_FILE_X

下载模块中的用户信息存储文件到 Host。在下载过程中以扩展协议传输。

请求命令格式如下：

数据域	数值	描述
Start code	0x70	
Command	0x43	
Param	NULL	
Size	NULL	

Flag	NULL	
Checksum	#	
End code	0x0A	

应答命令

数据域	数值	描述
Start code	0x70	
Command	0x43	
Param	NULL	
Size	NULL	
Error	0x61	SUCCESS
	0x63	FAIL
	0x6C	TIME_OUT
Checksum	#	
End code	0x0A	

4.37 MD_IDENTIFY_FREE

空闲指纹模板扫描识别。模块将实时比对的数据发送给 host。

注：host 端只负责接收数据，不用传送命令。

应答命令

数据域	数值	描述
Start code	0x70	
Command	0x2F	
Param	UserID	比对成功的用户 ID 号
Size	NULL	
Error	0x61	SUCCESS
	0x63	FAIL
	0x6C	TIME_OUT
Checksum	#	
End code	0x0A	

5 注意事项

5.2 数据包

模块默认采用的数据包格式是 16 进制的数据包。

串口传输的数据包最大长度不得超过 4KB;

5.3 用户 ID

用户 ID(即 PIN),取值范围: 1 ~ 65534, 数据类型为 unsigned short ,2Byte 的长度。即, 模块支持的最大用户容量不能超过 65534.

5.4 指纹模板长度

指纹模板长度的最大值为 2048Byte。所有的指纹模板的实际长度都不会大于这个最大值。

不同手指登记的指纹模板的长度不一定相等,即使是同一个手指登记的两个指纹模板的长度也不一定相等,但, 指纹模板的实际长度一定小于指纹模板的最大长度。

指纹模板结构体中的 Size 的值,就是本结构体中的 Template 的实际长度。

5.5 验证记录

验证记录,模块系统出厂默认是不保存的,如果要保存验证记录,可以通过系统参数“保存验证记录(ID: 0x36)”将其打开。

但是需要注意的是,模块本身不带时钟芯片,其内部的 RTC 时钟准确度不高,并且每次系统重启,时间都会归零,因此在开打“保存验证记录”参数后,一定要在模块启动之后或在模块正常工作之前需要先同步时钟,这样保存的验证记录才有意义,否则,保存的验证记录没有实际的参考价值。并且在工作过程中,每过一个小时,需要同步一次时钟。

5.6 命令耗时

大部分的命令在处理时,耗时很短,但是如果需要操作到指纹模板文件或用户数据文件时,耗时时间就会随着用户数和指纹模板数的多少有所变化,将如上电启动时长类似。

耗时比较长的命令有:

- **MD_ENROLL_IMAGE_X**
- **MD_SCAN_IMAGE_X**
- **MD_READ_TMP_X**
- **MD_LOAD_TMP_X**
- **MD_LOAD_USER_X**
- **MD_IDENTIFY_IMAGE_X**

5.7 删除命令

无论是在什么情况下，只要删除用户时，都会将用户的指纹删除，但是删除指纹时，不一定会删除用户，如果用户有多枚指纹时，只删除其中一枚或几枚，但是用户依然有指纹时，是不会将用户也删除的。

6 附件

6.2 数据传输参照文档

SilkID-命令详解.doc