

SilkID-SDK 接口

版本号：V2.2

目 录

一、接口函数.....	3
1. 建立连接.....	3
2. 断开连接.....	3
3. 登记用户模板.....	3
4. 指纹图像登记指纹.....	3
5. 从模块获取手指的指纹图像.....	4
6. 通过指纹图像进行识别用户.....	4
7. 清空所有数据.....	4
8. 读取验证通过的记录.....	4
9. 设置模块的时间.....	5
10. 获取时间.....	5
11. 获取系统状态.....	5
12. 读取参数.....	6
13. 参数设置.....	6
14. 保存参数.....	7
15. 删除所有用户信息.....	7
16. 删除用户.....	7
17. 下载所有用户信息.....	7

18.	读取单个用户信息.....	8
19.	增加和修改用户信息.....	8
20.	上传指纹模板.....	9
21.	删除用户指纹模板.....	9
22.	删除所有指纹模板.....	9
23.	下载指定的指纹模板.....	9
24.	下载所有指纹模板.....	10
25.	指纹验证.....	10
26.	删除所有验证通过记录.....	10
27.	扫描指纹模板.....	10
28.	模块复位.....	11
29.	禁用模块.....	11
30.	启用模块.....	11
31.	控灯操作.....	11
32.	升级固件.....	11
33.	上传模板存储文件.....	12
34.	上传用户信息存储文件.....	12
35.	下载用户信息存储文件.....	12
36.	下载模板存储文件.....	12
37.	实时指纹模板比对.....	12
二、注意事项.....		13
三、FAQ.....		13

一、接口函数

1. 建立连接

接口名：Connect

功能：与模块建立通信连接

参数：dev：串口设备，当为 NULL 时默认打开 “/dev/ttyS0” 设备文件

type：连接方式，0 为 USB，1 为 232

返回值：0-false，>0-true

```
int Connect(char *dev, int type);
```

2. 断开连接

接口名：Disconnect

功能：与模块断开通信连接

描述：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。

参数：无

返回值：0-false，>0-true

```
int Disconnect(void);
```

3. 登记用户模板

接口名：EnrollUserByScan

功能：登记用户模板

描述：用户需在模块的采集器上按压指纹三次进行登记。

注意：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。

参数：userID：用户 ID

返回值：0-false，>0-true

```
int EnrollUserByScan (int userID);
```

4. 指纹图像登记指纹

接口名：EnrollTemplateByImage

功能：指纹图像登记指纹(扩展通信协议)

描述：Host 发送指纹图像到模块中进行登记指纹。对于指纹图像的传输，使用扩展通信协议格式。

注意：模块中的原始图像数据可能与登记的该枚指纹图像比对不符，原因可能是该枚指纹图像经过旋转等方式后，导致发送数据时其图像数据的组织顺序与原始指纹图像数据不一致的情况。其传输的图像数据为真实的纯图像数据。

参数：int userID：用户 ID， data：图像数据， dataSize：图像数据大小

返回值：0-false，>0-true

int EnrollTemplateByImage (**int** userID, **unsigned char** * data , **int** dataSize);

5. 从模块获取手指的指纹图像

接口名：GetFingerImage

功能：从模块获取手指的指纹图像

描述：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。获取到的数据为真实的纯图像数据。

参数：width：图像宽度， height：图像高度， data：图像数据

返回值：0-false，>0-true

int GetFingerImage (**int** *width, **int** *height, **unsigned char** * data);

6. 通过指纹图像进行识别用户

接口名：IdentifyByImage

功能：通过指纹图像进行识别用户

描述：模块收到图像识别命令后，会返回一个确认信息，紧接着模块会进入接受图像数据状态，这时采用扩展协议进行数据传输。

参数：imageSize：指纹图像大小， userID：用户 ID， index：指纹索引值， data：指纹图像

返回值：0-false，>0-true

int IdentifyByImage (**int** imageSize, **int** *userID, **int** *index, **unsigned char** * data);

7. 清空所有数据

接口名：ClearDB

功能：清空所有数据

描述：当调用此接口时，会删除所有的指纹模板、用户信息和指纹验证通过记录。

参数：无

返回值：0-false，>0-true

int ClearDB (**void**);

8. 读取验证通过的记录

接口名：ReadAllLogs

功能：从模块中读取所有验证通过的记录到 host.

描述：此接口需循环调用从而取出模块中验证通过的记录，当读到记录时返回 1，否则返回 0.

参数：userID：用户 ID，

event：存储记录时发生的事件

verified：验证的方式

date：日期

time：时间

reserved：此域可以使用 MD_LC 命令来读取或设置。

返回值：0-false，>0-true

```
int ReadAllLogs(int *userID, char *event, char *verified,
               unsigned long *date, unsigned long *time, char *reserved);
```

备注：其中date为未解析的时间数据，可以通过TimeAnalyse函数进行解析。

接口名：TimeAnalyse

功能：解析数据包中的日期和时间

描述：通过调用“ReadAllLogs”接口取出的数据进行解析。

参数：date：日期数据包，time：时间数据包，

Year：年，Month：月，Day：日，Hour：时，Minute：分，Second：秒

返回值：无

```
void TimeAnalyse(unsigned long date, unsigned long time,
                 int *Year, int *Month, int *Day, int *Hour, int *Minute, int *Second);
```

9. 设置模块的时间

接口名：SetTime

功能：设置模块的时间

参数：Year：年，Month：月，Day：日，Hour：时，Minute：分，Second：秒

返回值：0-false，>0-true

```
int SetTime (int Year, int Month, int Day, int Hour, int Minute, int Second);
```

10. 获取时间

接口名：GetTime

功能：读取模块的当前时间

注意：当以 YYYY/MM/DD 显示日期时，对于年份需要加上 2000.

参数：Year：年，Month：月，Day：日，Hour：时，Minute：分，Second：秒

返回值：0-false，>0-true

```
int GetTime (int *Year, int *Month, int *Day, int *Hour, int *Minute, int *Second);
```

11. 获取系统状态

接口名：GetStatus

功能：读取系统的状态

注意：当前模块一直返回忙状态。

参数：无

返回值：0-false, >0-true

int GetStatus (**void**);

12. 读取参数

接口名：GetParameter

功能：从模块中读取参数

参数：flag：根据参数 ID 读取相关值， value：读取的参数值

参数 ID：

0x36 - Save log

0x82 - Auto response when free scan

0x62 - Time out

0x6E - Firmware version ,

0x71 - Baudrate 115200

0x73 - Current number of Enrolled fp ,

0x74 - The available number of fp that can be enrolled

0x79 - Max user count

0x7C - Log number

0x7B- Maximum Log Count

0x89 - Builed number

0x6D - Module id

返回值：0-false, >0-true

int GetParameter (**int** flag, **int** *value);

13. 参数设置

接口名：SetParameter

功能：设置系统参数

描述：此命令只设置内存中的参数，实际没有保存。如果需要保存，需要调用 SaveParameter 接口

参数：value：实际参数值， flag：参数 ID，详见下表：

参数 ID：

flag	value
0x36 - 保存日志	on-0x31, off-0x30
0x82 - 登记指纹时自动回复	on-0x31, off-0x30
0x62 - 超时	0x31- 1 秒, 0x32-2 秒, ... 最大值:255 秒.
0x50 - 模块工作模式	0x30: 验证模式 0x31: 读头(图片)0x32: 读头(模板)
0x51 - 指纹模板格式	0x30: ZK , 0x32:Ansi 378 , 0x33:ISO 19794-2

返回值：0-false, >0-true

int SetParameter (**int** value, **int** flag);

14. 保存参数

接口名：SaveParameter

功能：保存参数，将参数值保存到文件

参数：无

返回值：0-false，>0-true

int SaveParameter (**void**);

15. 删除所有用户信息

接口名：DeleteAllUsers

功能：在删除所有用户的同时，会删除所有指纹模板信息。

参数：无

返回值：0-false，>0-true

int DeleteAllUsers (**void**);

16. 删除用户

接口名：DeleteUser

功能：根据指定的用户 ID 来删除用户，同时，也会删除该用户的指纹模板

注意：在删除用户时，会先删除该用户的指纹模板。在删除模板时，如果模块突然掉电，这时可能会出现用户没有删除成功或者删除了部分模板情况。因此对于发生此情况，可以再次调用接口来删除。

参数：userID：用户 ID

返回值：0-false，>0-true

int DeleteUser (**int** userID);

17. 下载所有用户信息

接口名：ReadAllUser

功能：下载模块中的所有用户信息到 Host

描述：此接口需循环调用从而取出用户的信息，当有用户数据时返回 1，否则返回 0。

注意：返回的参数中仅 userID、name、fingerprintNum 有效，其它返回结果可能出现在以后版本中。

参数：userID：用户 ID，

name：姓名，

password：密码，

secLevel：用户加密等级 0，1

PIN2：用户第二识别号，

privilege：权限，

Card：卡号码，用于存储对应的 ID 卡的号码

fingerprintNum：用户登记的指纹数

返回值：0-false，>0-true

int ReadAllUser (**int** *userID, **char** *name, **char** *password,

unsigned short *secLevel, **unsigned long** *PIN2, **unsigned char** *privilege,

unsigned char *fingerprintNum, **unsigned char** *Card);

18. 读取单个用户信息

接口名：GetUser

功能：从模块中读取指定用户的信息

注意：返回的参数中仅 userID、name、fingerprintNum 有效，其它返回结果可能出现在以后版本中。

参数：userID：用户 ID，

name：姓名，

password：密码，

secLevel：用户加密等级 0，1

PIN2：用户第二识别号，

privilege：权限，

Card：卡号码，用于存储对应的 ID 卡的号码

fingerprintNum：用户登记的指纹数

返回值：0-false，>0-true

int GetUser (**int** userID, **char** *name, **char** *password,

unsigned short *secLevel, **unsigned long** *PIN2, **unsigned char** *privilege,

unsigned char *fingerprintNum, **unsigned char** *Card);

19. 增加和修改用户信息

接口名：ModifyUser

功能：模块根据传过来的用户信息，来查找该用户是否已存在，如果存在则修改，不存在则增加

注意：参数中仅 userID、name、fingerprintNum 有效，其它返回结果可能出现在以后版本中。

参数：userID：用户 ID，

name：姓名，

password：密码，

secLevel：用户加密等级 0，1

PIN2：用户第二识别号，

privilege：权限，

Card：卡号码，用于存储对应的 ID 卡的号码

fingerprintNum：用户登记的指纹数

返回值：0-false，>0-true

int ModifyUser (**int** userID, **char** *name, **char** *password,

unsigned short secLevel, **unsigned long** PIN2, **unsigned char** privilege,

unsigned char fingerprintNum, **char** *Card);



20. 上传指纹模板

接口名：SetTemplates

功能：从 host 上传指纹模板到模块中

参数：userID：用户 ID， flag：参数 ID， data：指纹数据， dataSize：指纹数据大小

参数 ID : 0 : None

1 : CHECK FINGER(检查指纹是否已存在)

返回值：0-false，>0-true

```
int SetTemplates (int userID, int flag, unsigned char *data, int dataSize);
```

21. 删除用户指纹模板

接口名：DeleteTemplates

功能：根据指定的用户 ID 和指纹索引来删除指纹模板

参数：userID：用户 ID， index：指纹索引值(0-9)， flag：参数 ID

参数 ID： 0 - 表示该用户的所有指纹模板

1 - (DELETE ONLY ONE) 时,只删除指定索引的指纹模板.

返回值：0-false，>0-true

```
int DeleteTemplates (int userID, int index, int flag);
```

22. 删除所有指纹模板

接口名：DeleteAllTemplates

功能：删除所有用户的指纹模板

注意：此命令模块会一直返回成功。

参数：无

返回值：0-false，>0-true

```
int DeleteAllTemplates (void);
```

23. 下载指定的指纹模板

接口名：ReadTemplates

功能：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。

描述：此接口需循环调用从而获取该用户的所有指纹模板，当有该用户的指纹模板数据时返回 1，否则返回 0。

参数：userID：用户 ID， index：指纹索引值， flag：参数 ID

data : 单个模板数据

参数 ID：0-忽略指纹索引 1-根据指纹索引下载

返回值：0-false, >0-true

```
int ReadTemplates(int userID, int index, int flag, unsigned char *data);
```

24. 下载所有指纹模板

接口名：ReadAllTemplates

功能：从模块中下载所有指纹模板数据到 host.

描述：此接口需循环调用从而获取模块中的所有指纹模板，当读取到指纹模板数据时返回 1，否则返回 0.

参数：data：单个模板数据

返回值：0-false，>0-true

int ReadAllTemplates (**unsigned char** *data);

25. 指纹验证

接口名：Verify

功能：等待用户按手指从而进行 1:1 指纹模板验证

描述：调用此接口函数时，需要传输 User ID 到模块，模块收到此命令后，等待用户按手指从而进行 1:1 的验证

参数：userID：用户 ID

返回值：0-false，>0-true

int Verify (**int** userID);

26. 删除所有验证通过记录

接口名：DeleteAllLogs

功能：删除所有验证通过的记录

注意：调用此接口模块会一直返回成功。

返回值：0-false，>0-true

参数：无

int DeleteAllLogs (**void**);

27. 扫描指纹模板

接口名：ScanTemplate

功能：扫描指纹模板

描述：此功能在模块设置为读头(模板)方能正常工作。host 需要一直调用此函数来读取指纹模板。调用此命令时，模块立即检查内部缓冲区是否有模板存在，若存在模板立即读取缓冲区的模板并返回。当内部缓冲区没有模板存在也会立即返回，其模板内容为空。模板最大长度为 2K。

注意：返回的模板格式支持 ZK、Ansi 378、ISO 19794-2 三种格式，其结果会根据所设参数决定。

参数：data：指纹模板数据

返回值：0-false，1-true

void ScanTemplate(**unsigned char** *data);

28. 模块复位

接口名：Reset

功能：模块复位

参数：无

返回值：0-false, 1-true

int Reset(**void**);

29. 禁用模块

接口名：DisableDevice

功能：禁用模块

描述：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。

参数：无

返回值：0-false, 1-true

int DisableDevice(**void**);

30. 启用模块

接口名：EnableDevice

功能：启用模块

描述：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。

参数：无

返回值：0-false, 1-true

int EnableDevice(**void**);

31. 控灯操作

接口名：SetParameter

功能：通过设置参数接口来进行闪灯控制

参数：

flag：其值为 0x31

value：高 8 位为灯类型，0x80:绿灯，0x40:红灯，0xC0:黄灯

低 8 位为灯显示时长，其范围为：1~255，单位为秒。

返回值：0-false, >0-true

int SetParameter (**int** value, **int** flag);

32. 升级固件

接口名：Upgrade

功能：升级固件

参数：fw:固件内容，size:固件大小

返回值：0-false，>0-true

int Upgrade (**unsigned char** *fw, **int** size);

33. 上传模板存储文件

接口名：UploadTemplatesFile

功能：将 host 上的模板存储文件上传至模块中

参数：fw:指纹模板数据，size:模板数据大小

返回值：0-false，>0-true

int UploadTemplatesFile (**unsigned char** *fw, **int** size);

34. 上传用户信息存储文件

接口名：UploadUserFile

功能：将 host 上的用户信息存储文件上传至模块中

参数：fw:用户信息数据，size:用户信息数据大小

返回值：0-false，>0-true

int UploadUserFile (**unsigned char** *fw, **int** size);

35. 下载用户信息存储文件

接口名：DownloadUserFile

功能：从模块中下载用户信息存储文件到 host.

参数：无

返回值：0-false，>0-true

int DownloadUserFile (**void**);

36. 下载模板存储文件

接口名：DownloadTemplatesFile

功能：从模块中下载指纹模板存储文件至 host.

参数：无

返回值：0-false，>0-true

int DownloadTemplatesFile (**void**);

37. 实时指纹模板比对

接口名：FreeScan

功能：实时接收在模块指纹采集器上比对成功与否的数据到 host

参数：无

返回值：0-false, -1-no, >0-UserID

int FreeScan (**void**);

二、注意事项

- 1、 用户 ID 范围：1~65535
- 2、 模块的工作模式设置为验证模式时，当用户在模块的采集器上按压手指进行比对时，模块会发送验证结果信息到 host。

三、FAQ

- 1、 如何从 A 模块中获取所有用户的指纹模板并上传至 B 模块中？

答：方法一：

1)首先调用 DownloadTemplatesFile 函数获取已登记的所有用户的指纹模板到 host。 .

2)然后调用 UploadTemplateFile 函数(上传时模块会先清空模块内的所有指纹模板)将已保存在 host 上的指纹模板上传至模块。

3)重启模块（软复位或重插拔）数据生效。

方法二：

1) 首先调用 ReadTemplates 函数下载指定的用户指纹模块到 host。

2) 然后调用 SetTemplates 函数将下载到 host 的指定用户的指纹模板上传至模块。即可完成操作。

注意：DownloadTemplatesFile 函数和 UploadTemplateFile 函数必须配套使用。

- 2、 如何通过指纹图像登记用户？

答：首先您得准备好该用户的指纹图像数据，然后调用 EnrollTemplateByImage 函数进行登记。

- 3、 怎样登记用户？

答：通过调用 EnrollUserByScan 函数，然后根据模块的提示灯按压三次手指即可完成指纹模板登记用户。