



API调用说明

一、函数声明列表.....	1
二、API 详细说明.....	2
三、函数返回码列表.....	错误！未定义书签。
四、API 调用示例.....	错误！未定义书签。
五、注册搜索流程图.....	13

一、函数声明列表

1. 设备是否在线	IsInit
2. 打开设备函数	OpenComm
3. 关闭设备函数	CloseComm
4. 设备通信测试	Run_TestConnection
5. 设置参数	Run_SetParam
6. 获取参数	Run_GetParam
7. 获取设备信息	Run_GetDeviceInfo
8. 设置 ID 号注释	Run_SetIDNote
9. 读出模置 ID 号注释	Run_GetIDNote
10. 设置模块序列号 SN	Run_SetModuleSN
11. 获取模块序列号 SN	Run_GetModuleSN
12. 获取图像	Run_GetImage
13. 探测手指是否存在	Run_FingerDetect
14. 上传图像	Run_UpImage
15. 下传图像	Run_DownImage
16. 控制灯函数	Run_SLEDControl
17. 存储指纹特征	Run_StoreChar
18. 加载指纹特征函数	Run_LoadChar
19. 上传特征函数	Run_UpChar
20. 下传特征函数	Run_DownChar
21. 删除指纹特征函数	Run_DelChar
22. 获取空 ID	Run_GetEmptyID
23. 获取指纹位置注册状态	Run_GetStatus
24. 检查指定编号范围指纹模板是否破损	Run_GetBrokenID
25. 获取指定编号范围指纹注册总数	Run_GetEnrollCount
26. 生成指纹特征函数	Run_Generate
27. 合成模板函数	Run_Merge
28. 特征比对函数	Run_Match
29. 特征搜索	Run_Search
30. 指定位置指纹比对函数	Run_Verify
31. 获取设备描述信息	GetDeviceInformation



深圳市指昂科技有限公司

Shenzhen Zhi Ang Science and Technology Co., Ltd.

二、API 详细说明

1、打开设备函数

public boolean OpenComm()

参数:

无

返回值:

True: 表示打开成功 False: 表示打开失败

2、关闭设备函数

public boolean CloseComm()

参数:

无

返回值:

True: 表示关闭成功 False: 表示关闭失败

3、设备是否在线

public boolean IsInit()

参数:

无

返回值:

True: 表示在线 False: 表示离线

4、设备通信测试

int Run_TestConnection()

参数:

无

返回值:

0 为成功, 其它返回值请参考错误返回码

5、设置参数

int Run_SetParam(**int** p_nParamIndex, **int** p_nParamValue)

参数:

nParamIndex: 参数 (0: Device ID; 1: Security Level, 2:Baudrate, 3:Duplication Check, 4: Auto Learn)

p_nParamValue: 参数值

Device ID : (0-255)

Level : (1-5) 1 最低 5 最高

Baudrate: 可设置值为 1~8

1:9600bps; 2:19200bps; 3:38400bps; 4:57600bps; 5:115200bps;

6:230400bps; 7:460800bps; 8:921600bps

Duplication Check: 指纹重复检查 (Duplication Check) : 可设置 0/1 (禁用/允许)

Auto Learn: 自学习功能 (Auto Learn) : 可设置 0/1 (禁止/允许)

返回值:

0 为成功, 其它返回值请参考错误返回码

6、获取参数

int Run_GetParam(**int** p_nParamIndex, **int** p_nParamValue)

参数:

nParamIndex: 参数 (0: Device ID; 1: Security Level, 2:Baudrate, 3:Duplication Check, 4: Auto Learn)

p_nParamValue[]: 参数值

返回值:

0 为成功, 其它返回值请参考错误返回码

7、获取设备信息

int Run_GetDeviceInfo(**String** p_szDevInfo)

参数:

String p_szDevInfo : 获取 Target 的 Device Information, 本设备固件名及版本号

返回值:

0 为成功, 其它返回值请参考错误返回码

8、设置 ID 号注释



深圳市指昂科技有限公司

Shenzhen Zhi Ang Science and Technology Co., Ltd.

int Run_SetIDNote(**int** p_nTplNo, String p_pstrNote)

参数:

int p_nTplNo : 指纹 ID 号

String p_pstrNote : ID 号注释说明 (64byte)

返回值:

0 为成功, 其它返回值请参考错误返回码

9、读出模置 ID 号注释

int Run_GetIDNote(**int** p_nTplNo, String[] p_pstrNote)

参数:

int p_nTplNo : 指纹 ID 号

String[] p_pstrNote : ID 号注释说明 (64byte)

返回值:

0 为成功, 其它返回值请参考错误返回码

10、设置模块序列号 SN

int Run_SetModuleSN(String p_pstrModuleSN)

参数:

String p_pstrModuleSN : 设备序列号 (16byte)

返回值:

0 为成功, 其它返回值请参考错误返回码

11、获取模块序列号 SN

int Run_GetModuleSN(String[] p_pstrModuleSN)

参数:

String[] p_pstrModuleSN : 设备序列号 (16byte)

返回值:

0 为成功, 其它返回值请参考错误返回码

12、获取图像

int Run_GetImage()

参数:

无

返回值:

0 为成功, 其它返回值请参考错误返回码

13、探测手指是否存在

int Run_FingerDetect(**int**[] p_pnDetectResult)

参数:

int[] p_pnDetectResult: : (1: 有指纹输入, 0: 无指纹输入)

返回值:

0 为成功, 其它返回值请参考错误返回码

14、上传图像

int Run_UpImage(**int** p_nType, **byte**[] p_pFpData, **int**[] p_pnImgWidth, **int**[] p_pnImgHeight)

参数:

int p_nType : 指纹类型

①若 Image Type 为 0 则为 Full Mode, 发送全图 (光学: 242*266, FPC1011: 202*258)

②若 Image Type 为 1 则为 Quarter Mode, 发送 1/4 图像 (4 个点取 1 个点)。

byte[] p_pFpData : 指纹数据

int[] p_pnImgWidth : 指纹宽度

int[] p_pnImgHeight: 指纹高度

返回值:

0 为成功, 其它返回值请参考错误返回码

15、下传图像



深圳市指昂科技有限公司

Shenzhen Zhi Ang Science and Technology Co., Ltd.

int Run_DownImage(**byte**[] p_pData, **int** p_nWidth, **int** p_nHeight)参数:

参数:

byte[] p_pData: 指纹数据

int[] p_pnImgWidth: 指纹宽度

int[] p_pnImgHeight: 指纹高度

返回值:

0 为成功, 其它返回值请参考错误返回码

16、控制灯函数

int Run_SLEDControl(**int** p_nState)

参数:

int p_nState: 指纹类型 LED 状态 (1: 开, 0: 关)

返回值:

0 为成功, 其它返回值请参考错误返回码

17、存储指纹特征

int Run_StoreChar(**int** p_nTplNo, **int** p_nRamBufferID, **int**[] p_pnDupTplNo)

参数:

int p_nTplNo: 指纹位置 ID

int p_nRamBufferID: 指纹缓存区 (1、2)

int p_p_pnDupTplNo: 指纹存储位置

返回值:

0 为成功, 其它返回值请参考错误返回码

18、加载指纹特征函数

int Run_LoadChar(**int** p_nTplNo, **int** p_nRamBufferID)

参数:

int p_nTplNo: 指纹位置

int p_nRamBufferID: 特征缓存区 (1/2/3)

返回值:

0 为成功, 其它返回值请参考错误返回码

19、上传特征函数

int Run_UpChar(**int** p_nRamBufferID, **byte**[] p_pbyTemplate)

参数:

int p_nRamBufferID: 特征缓存区 (1/2/3)

byte[] p_pbyTemplate: 特征数据

返回值:

0 为成功, 其它返回值请参考错误返回码

20、下传特征函数

int Run_DownChar(**int** p_nRamBufferID, **byte**[] p_pbyTemplate)参数:

参数:

int p_nRamBufferID: 特征缓存区 (1/2/3)

byte[] p_pbyTemplate: 特征数据

返回值:

0 为成功, 其它返回值请参考错误返回码

21、删除指纹特征函数

int Run_DelChar(**int** p_nSTmplNo, **int** p_nETmplNo)

参数:

int p_nSTmplNo: 特征删除开始位置

int p_nETmplNo: 特征删除结束位置

返回值:

0 为成功, 其它返回值请参考错误返回码

22、获取空 ID



深圳市指昂科技有限公司

Shenzhen Zhi Ang Science and Technology Co., Ltd.

int Run_GetEmptyID(**int** p_nSTmplNo, **int** p_nETmplNo, **int**[] p_pnEmptyID)

参数:

int p_nSTmplNo : 开始位置
int p_nETmplNo : 结束位置
int[] p_pnEmptyID : 空 ID 位置

返回值:

0 为成功, 其它返回值请参考错误返回码

23、获取指纹位置注册状态

int Run_GetStatus(**int** p_nTmplNo, **int**[] p_pnStatus)

参数:

int p_nSTmplNo : 指纹 ID
int[] p_pnStatus: (1: 已注册, 0: 没有注册)

返回值:

0 为成功, 其它返回值请参考错误返回码

24、检查指定编号范围指纹模板是否破损

int Run_GetBrokenID(**int** p_nSTmplNo, **int** p_nETmplNo, **int**[] p_pnCount, **int**[] p_pnFirstID)

参数:

int p_nSTmplNo : 开始位置
int p_nETmplNo : 结束位置
int[] p_pnCount: 破损 Template 的个数
int[] p_pnFirstID : 第一个破损 Template 编号

返回值:

0 为成功, 其它返回值请参考错误返回码

25、获取指定编号范围指纹注册总数

int Run_GetEnrollCount(**int** p_nSTmplNo, **int** p_nETmplNo, **int**[] p_pnEnrollCount)

参数:

int p_nSTmplNo : 开始位置
int p_nETmplNo : 结束位置
int[] p_pnEnrollCount : 指纹数量

返回值:

0 为成功, 其它返回值请参考错误返回码

26、生成指纹特征函数

int Run_Generate(**int** p_nRamBufferID)

参数:

int p_nRamBufferID : 特征缓存区 (1/2/3)

返回值:

0 为成功, 其它返回值请参考错误返回码

27、合成模板函数

int Run_Merge(**int** p_nRamBufferID, **int** p_nMergeCount)

参数:

int p_nRamBufferID : 特征缓存区 (1/2/3)
int p_nMergeCount: 特征个数 (2/3)

返回值:

0 为成功, 其它返回值请参考错误返回码

28、特征比对函数

int Run_Match(**int** p_nRamBufferID0, **int** p_nRamBufferID1)

参数:

int p_nRamBufferID0: 特征缓存区 (1/2/3)
int p_nRamBufferID1: 特征缓存区 (1/2/3)

返回值:

0 为成功, 其它返回值请参考错误返回码

29、特征搜索



深圳市指昂科技有限公司

Shenzhen Zhi Ang Science and Technology Co., Ltd.

```
int Run_Search(int p_nRamBufferID, int p_nStartID, int p_nSearchCount, int[] p_pnTplNo, int[] p_pnLearnResult)
```

参数:

int p_nRamBufferID0: 特征缓存区 (1/2/3)
int p_nStartID: 开始搜索位置
int p_nSearchCount: 结束搜索位置
int[] p_pnTplNo: 搜索到返回匹配位置 ID
int[] p_pnLearnResult: 搜索结果 (0: 表示成功)

返回值:

0 为成功, 其它返回值请参考错误返回码

30、指定位置指纹比对函数

```
int Run_Verify(int p_nTplNo, int p_nRamBufferID, int[] p_pnLearnResult)
```

参数:

int p_nSTmplNo: 指纹位置
int p_nRamBufferID0: 特征缓存区 (1/2/3)
int[] p_pnLearnResult: 比对结果 (0: 表示成功)

返回值:

0 为成功, 其它返回值请参考错误返回码

31、获取设备描述信息

```
public boolean GetDeviceInformation(String[] deviceInfo)
```

参数:

String[] deviceInfo: 设备描述信息

返回值:

0 为成功, 其它返回值请参考错误返回码

三、函数返回码列表

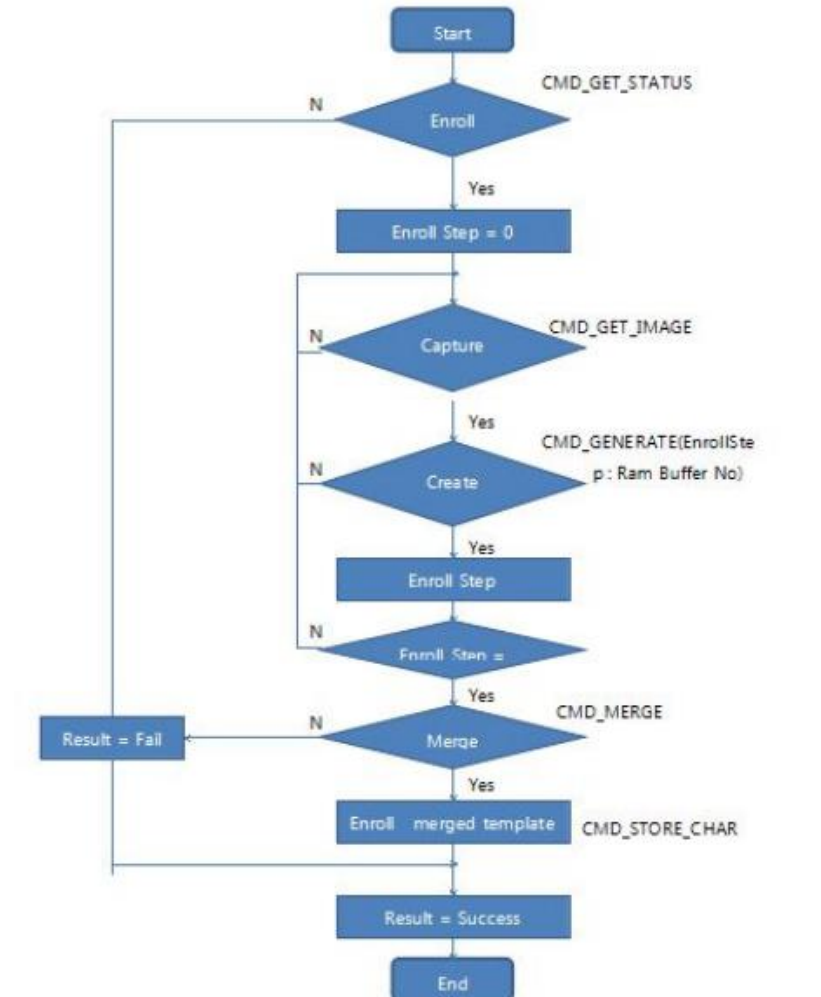
响应 (Response) 及错误代码表

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。

四、注册搜索指纹流程图

注册流程：



搜索指纹流程图

验证及识别流程:

