

YN020指纹模块命令使用说明

Linux版

2017年2月

珠海耀阳电子科技有限公司

目 录

1 提供文件	1
2 功能	2
FpCommand.....	2
3 命令说明	3
1 FF_INITIALIZE_CODE	3
2 FF_VERIFY_CODE.....	3
3 FF_IDENTIFY_CODE	3
4 FF_ENROLL_CODE.....	4
5 FF_CLEAR_TEMPLATE_CODE	4
6 FF_CLEAR_ALLTEMPLATE_CODE.....	5
7 FF_GET_EMPTY_ID_CODE	5
8 FF_GET_IMAGE_CODE	5
9 FF_READ_TEMPLATE_CODE	6
10 FF_WRITE_TEMPLATE_CODE	6
11 FF_SET_SECURITYLEVEL_CODE.....	6
12 FF_SET_DUP_CHECK_CODE	7
13 FF_GET_ENROLL_COUNT_CODE.....	7
14 FF_SET_RESIDUAL_CHECK_CODE	7
15 FF_UPDATE_TMPL_CODE.....	8
16 FF_TERMINATE	8

1 提供文件

-yoyon_types.h

定义数据类型

-Symbol.h

定义了算法库所利用的常量和函数。

- xxxxxx.so.

算法的共享库，运行时使用。

2 功能

本算法所用功能都通过执行FpCommand函数来实现。FpCommand函数通过第一参数p_wCmdCode作为命令码来指定操作，并带有三个参数。具体格式见下表。

FpCommand

Prototype	UINT8 FpCommand (UINT16 p_wCmdCode, FULL_UINT p_nParam1, FULL_UINT p_nParam2, FULL_UINT p_nParam3)	
Parameter	p_wCmdCode	指令代码
	p_nParam1	参数1
	p_nParam2	参数2
	p_nParam3	参数3
Return Value		请参考每个指令说明。
Function	利用指令代码进行各种功能处理。比如登记注册及比对等。算法库的所有功能，都是利用该函数执行的。	
Note		

可使用的命令码定义如下：

#define	FF_INITIALIZE_CODE	0x0000
#define	FF_VERIFY_CODE	0x0001
#define	FF_IDENTIFY_CODE	0x0002
#define	FF_ENROLL_CODE	0x0004
#define	FF_CLEAR_TEMPLATE_CODE	0x0006
#define	FF_CLEAR_ALLTEMPLATE_CODE	0x0007
#define	FF_GET_EMPTY_ID_CODE	0x0008
#define	FF_GET_IMAGE_CODE	0x000A
#define	FF_READ_TEMPLATE_CODE	0x000C
#define	FF_WRITE_TEMPLATE_CODE	0x000D
#define	FF_SET_SECURITYLEVEL_CODE	0x000E
#define	FF_SET_DUP_CHECK_CODE	0x0011
#define	FF_GET_ENROLL_COUNT_CODE	0x0025
#define	FF_SET_RESIDUAL_CHECK_CODE	0x0030
#define	FF_UPDATE_TMPL_CODE	0x0032
#define	FF_TERMINATE	0xFFFF

3 命令说明

FpCommand 函数，根据第一参数的指令代码执行各种功能。

以下说明所使用的指令代码、参数及返回值。

1 FF_INITIALIZE_CODE

p_wCmdCode	FF_INITIALIZE_CODE	
p_nParam1	0	
p_nParam2	0	
p_nParam3	0	
Return Value	ERR_SUCCESS	初始化成功
	其他	初始化失败
Function	初始化算法库，并从 sys_data_file 中读取登记数据。	
Note	使用本算法，必须首先调用此指令。	

2 FF_VERIFY_CODE

p_wCmdCode	FF_VERIFY_CODE	
p_nParam1	UINT32 p_nTmp1No	输入：指纹ID号
p_nParam2	0	
p_nParam3	0	
Return Value	FF_SUCCESS	验证成功
	其他	验证失败。
Function	验证当前采集的指纹图像和输入指纹ID号是否同一手指	
Note		

3 FF_IDENTIFY_CODE

p_wCmdCode	FF_IDENTIFY_CODE	
p_nParam1	UINT32* p_IdBuff	输出：识别成功返回的指纹ID号
p_nParam2	0	
p_nParam3	0	
Return Value	FF_SUCCESS	成功

	其他	失败。
Function	在已经登录的指纹中，搜寻识别指纹	
Note		

4 FF_ENROLL_CODE

p_wCmdCode	FF_ENROLL_CODE	
p_nParam1	UINT32 p_nFingerNumber	输入： Enroll Step步骤号 可取值0至4
p_nParam2	UINT32 p_nTmp1No	输入：待登记的模板编号ID
p_nParam3	UINT32* p_pnDup1No	输出：与待登记指纹相同的指纹模板编号
Return Value	FF_SUCCESS	成功
	其他	失败。
Function	<p>指纹登记。</p> <p>根据实际情况，可使用1~3个指纹图像来登记指纹，登记步骤分以下几种情况：</p> <ul style="list-style-type: none"> ➤ 1个指纹登记，指纹合成个数为1，Enroll Step包括0、1、4，其中Enroll Step=1之前，先用FF_GET_IMAGE_CODE获取指纹图像。 ➤ 2个指纹登记，指纹合成个数为2，Enroll Step包括0、1、2、4，其中Enroll Step=1和Enroll Step=2之前，都需要先用FF_GET_IMAGE_CODE获取指纹图像 ➤ 3个指纹登记，指纹合成个数为3，Enroll Step包括0、1、2、3、4，其中Enroll Step=1、Enroll Step=2、Enroll Step=3之前，都需要先用FF_GET_IMAGE_CODE获取指纹图像。 <p>Enroll Step = 0时，检测指定的模板编号是否能够登记模板。</p>	
Note		

5 FF_CLEAR_TEMPLATE_CODE

p_wCmdCode	FF_CLEAR_TEMPLATE_CODE	
p_nParam1	UINT32 p_nTmp1No	要删除的指纹ID号
p_nParam2	UINT32 Tmp1_Count	要删除的指纹数目
p_nParam3	0	
Return Value	FF_SUCCESS	成功
	其他	失败
Function	删除指定ID号开始、指定数目的已注册指纹数据。	

Note	
------	--

6 FF_CLEAR_ALLTEMPLATE_CODE

p_wCmdCode	FF_CLEAR_ALLTEMPLATE_CODE	
p_nParam1	0	
p_nParam2	0	
p_nParam3	0	
Return Value	FF_SUCCESS	成功
	其他	失败
Function	删除所有已注册指纹数据。	
Note		

7 FF_GET_EMPTY_ID_CODE

p_wCmdCode	FF_GET_EMPTY_ID_CODE	
p_nParam1	UINT32* p_pnIDBff	输出：保存空 ID号的内存。
p_nParam2	0	
p_nParam3	0	
Return Value	FF_SUCCESS	成功
	其他	失败
Function	检索没被利用的可使用 ID号。	
Note		

8 FF_GET_IMAGE_CODE

p_wCmdCode	FF_GET_IMAGE_CODE	
p_nParam1	UINT8* img_data	输出：保存指纹图像的外部buffer。调用方必须确保该buffer大于242*266字节。
p_nParam2	0	
p_nParam3	0	
Return Value	FF_SUCCESS	成功
	其他	失败
Function	获取指纹图像。	
Note	算法内部已经保留了该图像的副本，img_data中的图像，调用方可以自行处理	

9 FF_READ_TEMPLATE_CODE

p_wCmdCode	FF_READ_TEMPLATE_CODE	
p_nParam1	UINT32 p_nTplNo	输入：指定的指纹ID号。
p_nParam2	UINT8** p_ppBff	输出：一个指针值，该指针值指向保存指纹模板数据的地址。
p_nParam3	0	
Return Value	FF_SUCCESS	成功
	其他	失败
Function	读出指定 I D 的指纹模板数据所在的地址。	
Note		

10 FF_WRITE_TEMPLATE_CODE

p_wCmdCode	FF_WRITE_TEMPLATE_CODE	
p_nParam1	UINT32 p_nTplNo	输入：需要写入注册的指纹ID号。
p_nParam2	UINT8* p_pBff	输入：该指针指向需要写入注册的指纹模板数据的地址。
p_nParam3	UINT16* p_pnDuplNo	写入指纹模板已经存在在时，返回其指纹ID
Return Value	FF_SUCCESS	成功
	其他	失败
Function	写入指定 I D 的指纹模板数据。	
Note		

11 FF_SET_SECURITYLEVEL_CODE

p_wCmdCode	FF_SET_SECURITYLEVEL_CODE	
p_nParam1	UINT32 p_nSecLevel	待设定的 SecurityLevel 值（1 - 5）
p_nParam2	0	
p_nParam3	0	
Return Value	FF_SUCCESS	成功
	其他	失败

Function	设定新的 SecurityLevel 。被设定的 SecurityLevel 适用于 1:1、1:N 比对。	
Note	建议：一般将 SecurityLevel 值设为3 。	

12 FF_SET_DUP_CHECK_CODE

p_wCmdCode	FF_SET_DUP_CHECK_CODE	
p_nParam1	UINT32 p_nDupOption	1 为进行重复检查，0为不检查
p_nParam2	0	
p_nParam3	0	
Return Value	FF_SUCCESS	成功
	其他	失败
Function	设置指纹登记时是否进行重复检查。	
Note		

13 FF_GET_ENROLL_COUNT_CODE

p_wCmdCode	FF_GET_ENROLL_COUNT_CODE	
p_nParam1	UINT32* p_nEnrollCount	输出：已注册的指纹个数
p_nParam2	0	
p_nParam3	0	
Return Value	FF_SUCCESS	成功
	其他	失败
Function	获取已注册指纹数据总数。	
Note		

14 FF_SET_RESIDUAL_CHECK_CODE

p_wCmdCode	FF_SET_RESIDUAL_CHECK_CODE	
p_nParam1	UINT32 p_pResidual	输入：1为检查残留指纹，0不检查残留指纹
p_nParam2	0	
p_nParam3	0	
Return Value	FF_SUCCESS	成功
Function	设定是够进行残留指纹检查。	
Note		

15 FF_UPDATE_TMPL_CODE

p_wCmdCode	FF_UPDATE_TMPL_CODE	
p_nParam1	UINT32 *p_nAutoLearn	输出：1为更新成功，0未更新
p_nParam2	0	
p_nParam3	0	
Return Value	FF_SUCCESS	成功
Function	在比对、认证命令成功后立即执行此命令，可以自动更新已注册指纹。	
Note		

16 FF_TERMINATE

p_wCmdCode	FF_TERMINATE	
p_nParam1	0	
p_nParam2	0	
p_nParam3	0	
Return Value	FF_SUCCESS	成功
Function	退出算法库，退出时保存数据至系统文件 sys_data_file。	
Note	一般使用方法是在执行算法初始化时，应从保存的地方取得系统数据，作为初始化参数，这样算法就回去上次结束时的状态了。	

注： 调用FpCommand() 函数时，若发送没有以 p_wCmdCode 参数定义的指令代码，则会返回FF_ERR_INCORRECT_COMMAND 。