Fingerprint Recorder SDK

Programming Guide

Programming Guide

©2019 Hangzhou Hikvision Digital Technology Co., Ltd.

About this Manual

This Manual is subject to domestic and international copyright protection. Hangzhou Hikvision Digital Technology Co., Ltd. ("Hikvision") reserves all rights to this manual. This manual cannot be reproduced, changed, translated, or distributed, partially or wholly, by any means, without the prior written permission of Hikvision.

Trademarks

HIKVISION and other Hikvision marks are the property of Hikvision and are registered trademarks or the subject of applications for the same by Hikvision and/or its affiliates. Other trademarks mentioned in this manual are the properties of their respective owners. No right of license is given to use such trademarks without express permission.

Disclaimer

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, HIKVISION MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, REGARDING THIS MANUAL. HIKVISION DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE OF THE MANUAL, OR THE CORRECTNESS, ACCURACY, OR RELIABILITY OF INFORMATION CONTAINED HEREIN. YOUR USE OF THIS MANUAL AND ANY RELIANCE ON THIS MANUAL SHALL BE WHOLLY AT YOUR OWN RISK AND RESPONSIBILITY.

REGARDING TO THE PRODUCT WITH INTERNET ACCESS, THE USE OF PRODUCT SHALL BE WHOLLY AT YOUR OWN RISKS. HIKVISION SHALL NOT TAKE ANY RESPONSIBILITIES FOR ABNORMAL OPERATION, PRIVACY LEAKAGE OR OTHER DAMAGES RESULTING FROM CYBER ATTACK, HACKER ATTACK, VIRUS INSPECTION, OR OTHER INTERNET SECURITY RISKS; HOWEVER, HIKVISION WILL PROVIDE TIMELY TECHNICAL SUPPORT IF REQUIRED.

SURVEILLANCE LAWS VARY BY JURISDICTION. PLEASE CHECK ALL RELEVANT LAWS IN YOUR JURISDICTION BEFORE USING THIS PRODUCT IN ORDER TO ENSURE THAT YOUR USE CONFORMS THE APPLICABLE LAW. HIKVISION SHALL NOT BE LIABLE IN THE EVENT THAT THIS PRODUCT IS USED WITH ILLEGITIMATE PURPOSES. IN THE EVENT OF ANY CONFLICTS BETWEEN THIS MANUAL AND THE APPLICABLE LAW, THE LATER PREVAILS.

Contents

Chapter 1 Overview

1.1 Introduction

This manual mainly introduces the fingerprint recorder SDK based on the format of Dynamic Link Library (DLL) file. This SDK provides a series of APIs for the developer to fast develop the fingerprint application software with the fingerprint recorder.

SDK Main Functions:

- Collect Fingerprint Picture
- Add Fingerprint Template

Supported Fingerprint Recorder Models:

- DS-K1F830-F
- DS-K1F820-F
- DS-K1F181-F
- DS-K1F310-F
- DS-K1F320-F

System Requirement:

- Windows System (32-bit or 64-bit)

SDK Files:

```
├──bin
│    ├──x86
│    ├──x64
│    └──VcDemo
├──demo
│    ├──C#Demo
│    ├──JavaDemo
│    └──VcDemo
├──docs
│    Fingerprint Recorder SDK Programming Guide_V2.2.1
├──include : Header file
│    FPModule_SDK.h
└──libs : LIB file/ DLL file
     ├──x64
     │  FPModule_SDK_x64.dll
     │  FPModule_SDK_x64.lib
     └──x86
     │  FPModule_SDK.dll
     │  FPModule_SDK.lib
```

1.2 Update History

Version 2.2.1(20200227)

1. Optimized Lighting logic

Version 2.2.0 (20190225)

2. Optimized the fingerprint collection for limiting the low quality fingerprint.
3. Optimized the algorithm of evaluating fingerprint quality according to actual situation. The fingerprint with lower than 65 score is low quality.

Version 2.1.0 (20181224)

1. Added an API for setting fingerprint recording times: FPModule_SetCollectTimes.

Version 2.0.0 (20180629)

1. Updated the algorithm library.
2. Added 2-byte verification value at the end to the output fingerprint template.
3. Added an API for fingerprint template comparison: FPModule_MatchTemplate.
4. Edited finger pressing times when adding fingerprint. You should press 3 to 4 times.
5. Optimized the strategy for compositing fingerprint template.
6. Edited zooming strategy of FPC1011 fingerprint picture: when the picture resolution is increased to 508 DPI, the picture size will not be enlarged to 256*288, and only the actual size will be adopted.

Version 1.1.1 (20170714)

1. Edited finger pressing times when adding fingerprint. You should press 2 to 4 times.
2. Added light-off prompt for finger pressing when adding fingerprint.
3. Added supported fingerprint recorder model: DS-K1F320-F

Version 1.1 (20170509)

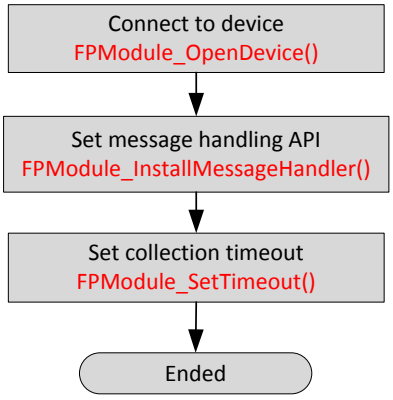
1. Edited API naming rule as *FPModule_XXXXXX ()*, and added name prefix *FPModule_* for APIs.
2. Edited the finger pressing times when adding fingerprint. You should press 3 to 5 times.
3. Added message callback API FPModule_InstaltMessageHandler() for prompt when adding fingerprint.
4. Added APIs FPModule_SetTimeout() and FPModule_GetTimeout() for setting and getting fingerprint collection timeout.
5. Added two output parameters, picture width and picture height, to the collection API.
6. Edited the returned error code and corresponding description.

Version 1.0 (20170509)

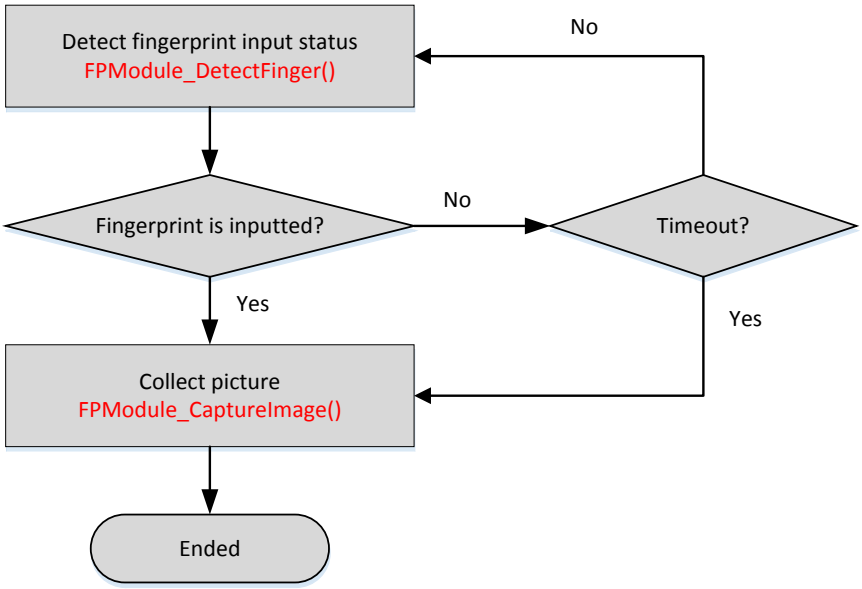
1. Created initial version V1.0.

Chapter 2API Call Process

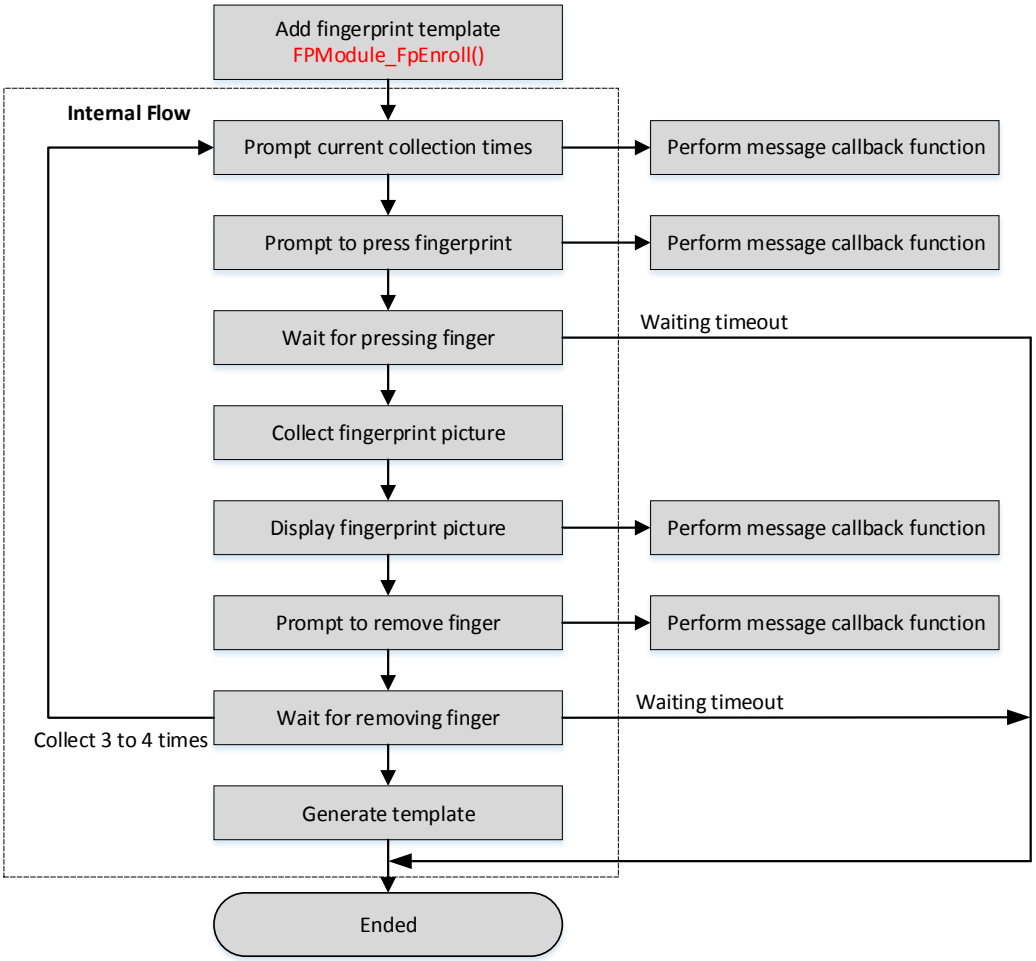
2.1 Initialize Fingerprint Recorder



2.2 Collect Fingerprint Picture



2.3 Add Fingerprint Template



Chapter 3 API Definition

3.1 Login FPModule_OpenDevice

API Definition: int _stdcall FPModule_OpenDevice (void);

Description: Before you can call other APIs of fingerprint recorder, you should log in to the recorder first.

Return Value: Return 0 for login succeeded, and return -1 for communication failed.

3.2 Logout FPModule_CloseDevice

API Definition: int _stdcall FPModule_CloseDevice ();

Description: Call this API after finishing using the fingerprint recorder to release resource.

Return Value: Return 0 for logout succeeded, and return -1 for communication failed.

3.3 Detect Fingerprint Input Status FPModule_DetectFinger

API Definition:

```
int _stdcall FPModule_DetectFinger (
    int     *pdwStatus
);
```

Description: Check whether the fingerprint is inputted in the fingerprint recorder.

Parameters: pdwStatus [OUT]Fingerprint input status. 0- No fingerprint inputted, 1- Fingerprint inputted.

Return Value: Return 0 for executing succeeded, and return 1 for communication failed.

3.4 Collect Fingerprint Picture FPModule_CaptureImage

API Definition:

```
int _stdcall FPModule_CaptureImage (
    unsigned char       *pbyImageData,
    int                 *pdwWidth,
    int                 *pdwHeight
);
```

Description: Collect fingerprint picture, and output original data of fingerprint picture.

Parameters: pbyImageData [OUT] Fingerprint picture data. The data length is the value of (picture height * picture width), and the allowed maximum size is 90KB.

	pdwWidth	[OUT] Fingerprint picture width, unit: pixel.
	pdwHeight	[OUT] Fingerprint picture height, unit: pixel.

Return Value: Return 0 for collection succeeded, return 1 for communication failed, and return 2 for collection timeout.

3.5 Set Collection Timeout FPModule_SetTimeout

API Definition:

```
int _stdcall FPModule_SetTimeoutl (
    int         dwSecond
);
```

Description: Set the waiting timeout of pressing or removing finger when collecting fingerprint.

Parameters:	dwSecond	[IN] Timeout (unit: second), range: from 1s to 60s.

Return Value: Return 0 for setting succeeded, return 1 for communication failed, and return 4 for parameter error.

3.6 Get Collection Timeout FPModule_GetTimeout

API Definition:

```
int _stdcall FPModule_GetTimeoutl (
    int         *pdwSecond
);
```

Description: Get the waiting timeout of pressing or removing finger when collecting fingerprint.

Parameters:	pdwSecond	[OUT] Timeout, unit: second.

Return Value: Return 0 for getting succeeded, return 1 for communication failed.

3.7 Set Collection Times FPModule_SetCollectTimes

API Definition:

```
int __stdcall FPModule_SetCollectTimes(
    int         dwTimes
);
```

Description: Set the fingerprint collection times.

Parameters:	dwTimes	[IN] Collection times, value range: [0,3]. If the value equals to 0, the collection times will be determined automatically, if the value is larger than 0, it is the actual collection times.

Return Value: Return 0 for setting succeeded, return 1 for communication failed, and return 4 for incorrect parameter.

3.8 Get Collection Times FPModule_GetCollectTimes

API Definition:

```
int __stdcall FPModule_GetCollectTimes(
    int         *pdwTimes
```

);

Description: Get the fingerprint collection times.

Parameters: dwTimes [OUT] Collection times.

Return Value: Return 0 for getting succeeded, return other value for failure.

3.9 Register Message Callback

FPModule_InstallMessageHandler

API Definition: int _stdcall FPModule_InstallMessageHandler (

FpMessageHandler msgHandler

);

Description: Register callback function for calling back the collected fingerprint information.

Parameters: msgHandler [IN] Message handling function.

Return Value: Return 0 for success.

Remarks: typedef void (_stdcall *FpMessageHandler)(FP_MSG_TYPE_T enMsgType, void *pMsgData);

FP_MSG_TYPE Message Type	Value	MsgData Message Parameter	Description
FP_MSG_PRESS_FINGER	0	None.	Prompt to press finger.
FP_MSG_RISE_FINGER	1	None.	Prompt to remove finger.
FP_MSG_ENROLL_TIME	2	int dwEnrollTime; *dwEnrollTime*: Current collection times.	Prompt the current collection times.
FP_MSG_CAPTURED_IMAGE	3	struct { int dwWidth; int dwHeight; unsigned char *pbyImage; }FP_IMAGE_DATA; *dwWidth*: Width of fingerprint picture *dwHeight*: Height of fingerprint picture *pbyImage*: Fingerprint picture data	Display the collected fingerprint picture information.

3.10Add Fingerprint Template FPModule_FPEnroll

API Definition:
```
int _stdcall FPModule_FPEnroll (
    unsigned char          *pbyFpTemplate
);
```

Description: Collect fingerprint and get fingerprint template.

Parameters: pbyFpTemplate [OUT] Fingerprint template data, the data length is 512 bytes.

Return Value: Return 0 for adding succeeded, return 1 for communication failed, return 2 for collection timeout, and return 3 for adding failed.

Remarks:

- When collecting fingerprint, you should press the same finger for 2 to 4 times.
- During collection process, call FPModule_InstallMessageHandler() to set callback function and get corresponding prompt information.
- For the waiting timeout of pressing finger, call FPModule_SetTimeout() to set.

3.11Get Fingerprint Template Quality FPModule_GetQuality

API Definition:
```
int _stdcall FPModule_GetQuality (
    unsigned char          * pbyFpTemplate
);
```

Description: Get fingerprint template quality.

Parameters: pbyFpTemplate [IN] Fingerprint template data, the data length is 512 bytes.

Return Value: Return value which is ranging from 0 to 100 for fingerprint template score. Higher score means higher template quality. Value 0: The template score is invalid.

3.12Get Device Version Information FPModule_GetDeviceInfo

API Definition:
```
int _stdcall FPModule_GetDeviceInfo (
    char          *pbyDeviceInfo
);
```

Description: Get the version information of fingerprint recorder.

Example: DS-K1F820-F_CN_STD_V1.0.0_Build170101

Parameters: pbyDeviceInfo [OUT] Version information of fingerprint recorder, the data length is 64 bytes.

Return Value: Return 0 for getting succeeded, and return 1 for communication failed.

3.13Compare Fingerprint Template

FPModule_MatchTemplate

API Definition:
```
int _stdcall FPModule_MatchTemplate (
    unsigned char       *pbyFPTemplate1,
    unsigned char       *pbyFPTemplate2,
    int                 dwSecurityLevel
);
```

Parameters:

pbyFPTemplate1	[IN] Fingerprint template 1, the data size is 512 bytes.
PbyFPTemplate2	[IN] Fingerprint template 2, the data size is 512 bytes.
dwSecurityLevel	[IN] Security level for comparison results, which is between 1 and 5.

Return Value: Return 0 for comparison succeeded, return 6 for comparison failed, and return 4 for input parameter error.

3.14Get SDK Version Information

FPModule_GetSDKVersion

API Definition:
```
int _stdcall FPModule_GetSDKVersion (
    char        *pbySDKVersion
);
```

Description: Get the version information of fingerprint recorder SDK.

Example: FPModuleSDK_Win_x86_V1.0.0_Build170101

Parameters:

pbySDKVersion	[OUT] Version information of fingerprint recorder SDK, the data length is 64 bytes.

Return Value: Return 0 for getting succeeded, and return 1 for communication failed.

Chapter 4 Error Code

Error Name	Error Code	Description	Handling Method
FP_SUCCESS	0	Succeeded.	None.
FP_CONNECTION_ERR	1	Communication failed.	Check the connection of hardware, and reconnect to the device.
FP_TIMEOUT	2	Collection timeout.	Press or remove the finger before the collection timeout. Or set a longer timeout.
FP_ENROLL_FAIL	3	Adding fingerprint template failed.	Try again.
FP_PARAM_ERR	4	Parameter error.	Check the inputted parameter.
FP_EXTRACT_FAIL	5	Extracting features failed.	Record the fingerprint again.
FP_MATCH_FAIL	6	Fingerprint comparison failed	None.


See Far, Go Further