ZKFinger Reader SDK Development Guide C API

Version: 2.0

Date: Sep 2016

ZKFinger Reader SDK Development Guide

Release History

Date	Version	Remarks
May 21, 2016	1.0	Basic version
June 1, 2016	1.1	Added external image interfaces.
Sep 18, 2016	2.0	Added 2.0 interface, keep old interface

Contents

1 Overview

Thank you for using ZKFinger Reader SDK. Please read this document carefully before use to fast learn how to use ZKFinger Reader SDK.

2 Privacy Policy

You are authorized to use the software but you must make the following commitment to ZKTeco: You shall not use, copy, modify, lease, or transfer any part of the SDK beyond the clauses of this document.

3 System Requirements

1) Operating system: Windows XP or a later version
2) Applicable development languages: C++, C#, VB, Delphi

4 Installation and Deployment

1) Installation: Install ZKFinger SDK 5.x/ZKOnline SDK 5.x.

5 Description of SDK Interfaces

5.1 Type Definition

See *libzkfptype.h.*
The SDK interfaces uses __stdcall.

```
#ifdef _WIN32
#ifndef APICALL
#define APICALL __stdcall
#endif
```

5.1.1 Constants

1) Maximum length of a template
   [Definition] #define MAX_TEMPLATE_SIZE 2048

2) Fingerprint 1:1 threshold parameter code

[Definition] #define FP_THRESHOLD_CODE 1

3) Fingerprint 1:N threshold parameter code

[Definition] #define FP_MTHRESHOLD_CODE 2

5.2 Interface Description

5.2.1 ZKFPM_Init

[Function]

int APICALL ZKFPM_Init();

[Purpose]

This function is used to initialize resources.

[Parameter Description]

None

[Return Value]

0 Succeeded

Others Failed (See the Appendixes.)

5.2.2 ZKFPM_Terminate

[Function]

int APICALL ZKFPM_Terminate();

[Purpose]

This function is used to release resources.

[Parameter Description]

None

[Return Value]

0 Succeeded

Others Failed (See the Appendixes.)

5.2.3 ZKFPM_GetDeviceCount

[Function]

int APICALL ZKFPM_GetDeviceCount();

[Purpose]

This function is used to acquire the number of devices.

[Parameter Description]

None

[Return Value]

>=0 Device count

<0 The function fails to be called (See the Appendixes.)

5.2.4 ZKFPM_OpenDevice

[Function]

HANDLE APICALL ZKFPM_OpenDevice(int index);

[Purpose]

This function is used to start a device.

[Parameter Description]

index

Device index

[Return Value]

Device operation instance handle

5.2.5 ZKFPM_CloseDevice

[Function]

int APICALL ZKFPM_CloseDevice(HANDLE hDevice);

[Purpose]

This function is used to shut down a device.

[Parameter Description]

hDevice

Device operation instance handle

[Return Value]

0 Succeeded

Others Failed (See the Appendixes.)

5.2.6 ZKFPM_SetParameters

[Function]

int APICALL ZKFPM_SetParameters(HANDLE hDevice, int nParamCode, unsigned char* paramValue, unsigned int cbParamValue);

[Purpose]

This function is used to set fingerprint reader parameters.

[Parameter Description]

hDevice

Device operation instance handle

nParamCode

Parameter code (For details, see the parameter code list.)

paramValue

Parameter value

cbParamValue

Parameter data length

[Return Value]

0 Succeeded

Others Failed (See the Appendixes.)

[Note]

5.2.7 ZKFPM_GetParameters

[Function]

int APICALL ZKFPM_GetParameters(HANDLE hDevice, int nParamCode, unsigned char* paramValue, unsigned int* cbParamValue);

[Purpose]

This function is used to acquire fingerprint reader parameters.

[Parameter Description]

hDevice

Device operation instance handle

nParamCode

Parameter code (For details, see the parameter code list.)

paramValue [out]

Returned parameter value

cbParamValue [in/out]

[in] Memory size allocated based on nParamCode

[out] Data size of the returned parameter value

[Return Value]

0 Succeeded

Others Failed (See the Appendixes.)

[Note]

5.2.8 ZKFPM_AcquireFingerprint

[Function]

int APICALL ZKFPM_AcquireFingerprint(HANDLE hDevice, unsigned char* fpImage, unsigned int cbFPImage, unsigned char* fpTemplate, unsigned int* cbTemplate);

[Purpose]

This function is used to capture a template.

[Parameter Description]

hDevice

Device operation instance handle

fpImage [out]

Returned fingerprint image

cbFPImage

Memory size of **fpImage**

fpTemplate [out]

Returned fingerprint template

cbfpTemplate [in/out]

[in] Pre-allocated memory size of **fpTemplate**. It is recommended that it be set to **MAX_TEMPLATE_SIZE(2048)**.

[out] Fingerprint template data size that is **actually** returned

[Return Value]

0 Succeeded

Others Failed (See the Appendixes.)

[Note]

5.2.9 ZKFPM_AcquireFingerprintImage

[Function]

int APICALL ZKFPM_AcquireFingerprintImage(HANDLE hDevice, unsigned char* fpImage, unsigned int cbFPImage);

[Purpose]

This function is used to capture a image.

[Parameter Description]

hDevice

Device operation instance handle

fpImage [out]

Returned fingerprint image

cbFPImage

Memory size of **fpImage**

[Return Value]

0 Succeeded

Others Failed (See the Appendixes.)

[Note]

5.2.10 ZKFPM_DBInit

[Function]

HANDLE APICALL ZKFPM_DBInit();

[Purpose]

This function is used to create an algorithm cache.

[Parameter Description]

None

[Return Value]

Cache handle

5.2.11 ZKFPM_DBFree

[Function]

int APICALL ZKFPM_DBFree(HANDLE hDBCache);

[Purpose]

This function is used to release an algorithm cache.

[Parameter Description]

Cache handle

[Return Value]

0 Succeeded

Others Failed (See the Appendixes.)

5.2.12 ZKFPM_DBMerge

[Function]

int APICALL ZKFPM_DBMerge(HANDLE hDBCache, unsigned char* temp1, unsigned char* temp2, unsigned char* temp3, unsigned char* regTemp, unsigned int* cbRegTemp) ;

[Purpose]

This function is used to combine three pre-registered fingerprint templates as one registered fingerprint template.

[Parameter Description]

hDBCache

Cache handle

temp1

Pre-registered fingerprint template 1

temp2

Pre-registered fingerprint template 2

temp3

Pre-registered fingerprint template 3

regTemp[out]

Registered template

cbRegTemp[in/out]

[in] Pre-allocated memory size of **fpTemplate**. It is recommended that it be set to **MAX_TEMPLATE_SIZE(2048)**.

[out] Fingerprint template data size that is actually returned

[Return Value]

0 Succeeded

Others Failed (See the Appendixes.)

5.2.13 ZKFPM_DBAdd

[Function]

int APICALL ZKFPM_DBAdd(HANDLE hDBCache, unsigned int fid, unsigned char* fpTemplate, unsigned int cbTemplate);

[Purpose]

This function is used to add a registered fingerprint template to the cache.

[Parameter Description]

hDBCache

Cache handle

fid

Fingerprint ID (32-bit unsigned integer larger than 0)

fpTemplate

Registered template

cbTemplate

Template length

[Return Value]

0	Succeeded
Others	Failed (See the Appendixes.)

5.2.14 ZKFPM_DBDel

[Function]

int APICALL ZKFPM_DBDel(HANDLE hDBCache, unsigned int fid);

[Purpose]

This function is used to delete the registered template of a specified fingerprint ID from the cache.

[Parameter Description]

hDBCache

Cache handle

fid

Fingerprint ID

[Return Value]

0	Succeeded
Others	Failed (See the Appendixes.)

5.2.15 ZKFPM_DBClear

[Function]

int APICALL ZKFPM_DBClear(HANDLE hDBCache);

[Purpose]

This function is used to clear the cache.

[Parameter Description]

hDBCache

Cache handle

[Return Value]

0 Succeeded

Others Failed (See the Appendixes.)

5.2.16 ZKFPM_DBCount

[Function]

int APICALL ZKFPM_DBCount(HANDLE hDBCache, unsigned int* fpCount);

[Purpose]

This function is used to acquire the number of fingerprint template in the cache.

[Parameter Description]

hDBCache

Cache handle

fpCount [out]

Fingerprint image account

[Return Value]

0 Succeeded

Others Failed (See the Appendixes.)

[Note]

5.2.17 ZKFPM_DBIdentify

[Function]

int APICALL ZKFPM_DBIdentify(HANDLE hDBCache, unsigned char* fpTemplate, unsigned int cbTemplate, unsigned int* FID, unsigned int* score);

[Purpose]

This function is used to conduct 1:N comparison.

[Parameter Description]

hDBCache

Cache handle

fpTemplate

Fingerprint template

cbfpTemplate

Data length of the fingerprint template

FID [out]

Returned fingerprint ID

Score [out]

Returned comparison score

[Return Value]

0	Succeeded
Others	Failed (See the Appendixes.)

5.2.18 ZKFPM_DBMatch

[Function]

int APICALL ZKFPM_DBMatch (HANDLE hDBCache, unsigned char* fpTemplate1, unsigned int cbfpTemplate1, unsigned char* fpTemplate2, unsigned int cbfpTemplate2);

[Purpose]

This function is used compare whether two fingerprint templates match.

[Parameter Description]

hDBCache

Cache handle

fpTemplate1

Fingerprint template 1

cbfpTemplate1

Data length of fingerprint template 1

fpTemplate2

Fingerprint template 2

cbfpTemplate2

Data length of fingerprint template 2

[Return Value]

>=0	Comparison score
<0	Error (See the Appendixes.)

5.2.19 ZKFPM_ExtractFromImage

[Function]

ZKINTERFACE int APICALL ZKFPM_ExtractFromImage(HANDLE hDBCache, const char* lpFilePathName, unsigned int DPI, unsigned char* fpTemplate, unsigned int *cbTemplate);

[Purpose]

This function is used to extract a fingerprint template from a BMP or JPG file.

[Parameter Description]

hDBCache

Cache handle

lpFilePathName

Full path of a file

DPI

Image DPI

fpTemplate

Fingerprint template

cbfpTemplate

Data length of fingerprint template 1

[Return Value]

0 Succeeded

Others Failed (See the Appendixes.)

[Note]

Only the SDK of the standard version supports this function.

6 Appendixes

6.1 Appendix 1

List of Common Parameter Codes

Parameter Code	Property	Data Type	Description
1	Read-only	Int	Image width
2	Read-only	Int	Image height
3	Read-write (supported only by the LIVEID20R currently)	Int	Image DPI (750/1000 is recommended for children.)
106	Read-only	Int	Image data size
1015	Read-only	4-byte array	VID&PID (The former two bytes indicate VID and the latter two bytes indicate PID.)
2002	Read-write (supported only by the LIVEID20R currently)	Int	Anti-fake function (1: enable; 0: disable)
2004	Read-only	Int	A fingerprint image is true if the lower five bits are all 1's (value&31==31).
1101	Read-only	String	Vendor information
1102	Read-only	String	Product name
1103	Read-only	String	Device SN
101	Write-only (Devices except the LIVE20R need to call a function to disable the parameter.)	Int	1 indicates that the white light blinks; 0 indicates that the parameter is disabled.
102	Write-only (Devices	Int	1 indicates that the

Parameter Code	Property	Data Type	Description
	except the LIVE20R need to call a function to disable the parameter.)		green light blinks; 0 indicates that the parameter is disabled.
103	Write-only (Devices except the LIVE20R need to call a function to disable the parameter.)	Int	1 indicates that the red light blinks; 0 indicates that the parameter is disabled.
104	Write-only (not supported by the LIVE20R)	Int	1 indicates that buzzing is started; 0 indicates that the parameter is disabled.
10001	Write-only(only supported by ISO/ANSI Version)	Int	0 ANSI378; 1 ISO 19794-2

6.2 Appendix 2

Descriptions of Returned Error Values

0	Operation succeeded
1	Initialized
-1	Failed to initialize the algorithm library
-2	Failed to initialize the capture library
-3	No device connected
-4	Not supported by the interface
-5	Invalid parameter
-6	Failed to start the device
-7	Invalid handle
-8	Failed to capture the image
-9	Failed to extract the fingerprint template
-10	Suspension operation
-11	Insufficient memory
-12	The fingerprint is being captured (the device is busy)
-13	Failed to add the fingerprint template to the memory
-14	Failed to delete the fingerprint template
-17	Operation failed (other error)
-18	Capture cancelled
-20	Fingerprint comparison failed (Great differences are incurred when different fingers are pressed or fingers are pressed improperly during registration.)
-22	Failed to combine registered fingerprint templates
-23	Opening the file failed

-24	Image processing failed