ZKFinger Reader SDK Development Guide C#

Version: 1.0

Date: May 2016

ZKFinger Reader SDK Development Guide

Release History

Date	Version	Remarks
May 21, 2016	1.0	Basic version

Contents

1 Overview ... 4
2 Privacy Policy ... 4
3 System Requirements ... 4
4 Installation and Deployment ... 4
5 Description of SDK Interfaces ... 5
  5.1 Referenced Class Library ... 5
  5.2 Description of the Class Library ... 5
  5.3 Member Variables ... 6
  5.4 Interface Description ... 6
    5.4.1 Initialize ... 6
    5.4.2 Finalize ... 6
    5.4.3 GetDeviceCount ... 6
    5.4.4 OpenDevice ... 7
    5.4.5 CloseDevice ... 7
    5.4.6 AcquireFingerprint ... 7
    5.4.7 GenerateRegTemplate ... 8
    5.4.8 AddRegTemplate ... 8
    5.4.9 DelRegTemplate ... 9
    5.4.10 Clear ... 9
    5.4.11 Identify ... 9
    5.4.12 VerifyByID ... 10
    5.4.13 Match ... 10
    5.4.14 Blob2Base64String ... 10
    5.4.15 Base64String2Blob ... 11
    5.4.16 ByteArray2Int ... 11
    5.4.17 Int2ByteArray ... 11
    5.4.18 ExtractFromImage ... 12
    5.4.19 SetParameters ... 12
    5.4.20 GetParameters ... 13
6 Appendixes ... 13
  6.1 Parameter Codes ... 13
  6.2 Error Codes ... 14

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

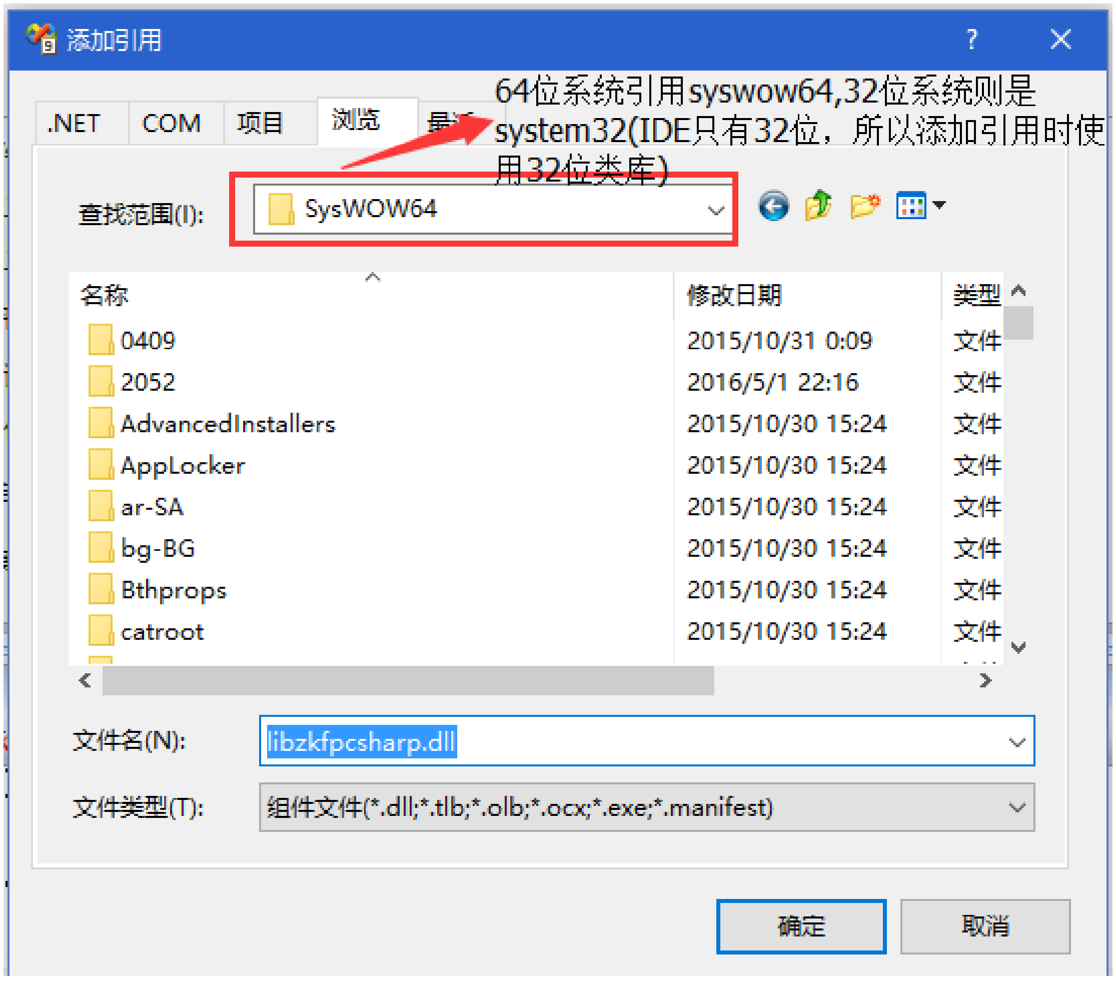
1) Operating system: Windows XP or a later version, .net framework 3.5
2) Applicable development language: C#

4 Installation and Deployment

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

5 Description of SDK Interfaces

5.1 Referenced Class Library


添加引用
.NET
COM
项目
浏览
最近
64位系统引用syswow64,32位系统则是system32(IDE只有32位，所以添加引用时使用32位类库)
查找范围(I):
SysWOW64
名称
修改日期
类型
文件名(N):
libzkfpcsharp.dll
文件类型(T):
组件文件(*.dll;*.tlb;*.olb;*.ocx;*.exe;*.manifest)
确定
取消

名称	修改日期	类型
0409	2015/10/31 0:09	文件
2052	2016/5/1 22:16	文件
AdvancedInstallers	2015/10/30 15:24	文件
AppLocker	2015/10/30 15:24	文件
ar-SA	2015/10/30 15:24	文件
bg-BG	2015/10/30 15:24	文件
Bthprops	2015/10/30 15:24	文件
catroot	2015/10/30 15:24	文件

5.2 Description of the Class Library

- **Dynamic library**
  Libzkfpcsharp.dll(system32/syswow64)
- **Namespace**
  libzkfpcsharp
- **Class name**
  Zkfp

5.3 Member Variables

Member variables can be acquired after OpenDevice is executed successfully.

- **imageWidth**
  Width of a fingerprint image
- **imageHeight**
  Height of a fingerprint image
- **devSn**
  Device SN (unique identifier of the device)

5.4 Interface Description

5.4.1 Initialize

[Function]

```
public int Initialize()
```

[Purpose]

This function is used to initialize the device.

[Parameter Description]

[Return Value]

0	Succeeded
Others	Failed (See the error code description.)

5.4.2 Finalize

[Function]

```
public int Finalize()
```

[Purpose]

This function is used to release library resources.

[Parameter Description]

[Return Value]

0	Succeeded
Others	Failed (See the error code description.)

5.4.3 GetDeviceCount

[Function]

```
public int GetDeviceCount()
```

[Purpose]

This function is used to acquire the number of collected devices.

[Parameter Description]

[Return Value]

Device count

5.4.4 OpenDevice

[Function]

public int OpenDevice(int index)

[Purpose]

This function is used to connect to a device.

[Parameter Description]

Index:

Device index (The values ranges from 0 to *n* and *n* indicates the device count minus 1.)

[Return Value]

0	Succeeded
Others	Failed (See the error code description.)

5.4.5 CloseDevice

[Function]

public int CloseDevice()

[Purpose]

This function is used to shut down a device.

[Parameter Description]

[Return Value]

0	Succeeded
Others	Failed (See the error code description.)

5.4.6 AcquireFingerprint

[Function]

public int AcquireFingerprint(byte[] imgBuffer, byte[] template, ref int size)

[Purpose]

This function is used to capture a fingerprint image.

[Parameter Description]

imgBuffer

Returned image (The array size is imageWidth*imageHeight.)

template

Returned fingerprint template (It is recommended that 2048 bytes be pre-allocated.)

size[in/out]

[in]	Template array length
[out]	Fingerprint template length that is actually returned

[Return Value]

0	Succeeded
Others	Failed (See the error code description.)

5.4.7 GenerateRegTemplate

[Function]

public int GenerateRegTemplate(byte[] temp1, byte[] temp2, byte[] temp3, byte[] regTemp, ref int regTempLen)

[Purpose]

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

[Parameter Description]

temp1

Pre-registered fingerprint template 1

temp2

Pre-registered fingerprint template 2

temp3

Pre-registered fingerprint template 3

regTemp

Returned registered template

regTempLen[in/out]

[in]	regTemp array length
[out]	Fingerprint template length that is actually returned

[Return Value]

0	Succeeded
Others	Failed (See the error code description.)

5.4.8 AddRegTemplate

[Function]

public int AddRegTemplate(int fid, byte[] regTemp)

[Purpose]

This function is used to add a registered template to the memory.

[Parameter Description]

fid

Fingerprint ID (The fingerprint ID is returned after 1:N comparison is successfully conducted.)

regTemp

Registered template

[Return Value]

0	Succeeded
Others	Failed (See the error code description.)

5.4.9 DelRegTemplate

[Function]

public int DelRegTemplate (int fid)

[Purpose]

This function is used to delete a registered fingerprint template from the memory.

[Parameter Description]

fid

Fingerprint ID (The fingerprint ID is returned after 1:N comparison is successfully conducted.)

[Return Value]

0	Succeeded
Others	Failed (See the error code description.)

5.4.10 Clear

[Function]

public int Clear()

[Purpose]

This function is used to clear all fingerprint templates in the memory.

[Parameter Description]

[Return Value]

0	Succeeded
Others	Failed (See the error code description.)

5.4.11 Identify

[Function]

public int Identify(byte[] temp, ref int fid, ref int score)

[Purpose]

This function is used to conduct 1:N comparison.

[Parameter Description]

temp

Template used for comparison

fid

Returned fingerprint ID

score

Returned comparison score

[Return Value]

0	Succeeded
Others	Failed (See the error code description.)

5.4.12 VerifyByID

[Function]

public int VerifyByID(int fid, byte[] temp)

[Purpose]

This function is used to conduct 1:1 comparison based on the fingerprint ID.

[Parameter Description]

fid

Returned fingerprint ID

temp

Template used for comparison

[Return Value]

>=0	Comparison score
Others	Failed (See the error code description.)

5.4.13 Match

[Function]

public int Match(byte[] temp1, byte[] temp2)

[Purpose]

This function is used to conduct 1:1 comparison on two fingerprint templates.

[Parameter Description]

temp1

Template 1 used for comparison

temp2

Template 2 used for comparison

[Return Value]

>=0	Comparison score
Others	Failed (See the error code description.)

5.4.14 Blob2Base64String

[Function]

static public int Blob2Base64String(byte[] buf, int len, ref String strBase64)

[Purpose]

This function is used to convert a byte[] array into a Base64 string.

[Parameter Description]

buf

BLOB data

len

Length

strBase64

Returned Base64 string

[Return Value]

String length

5.4.15 Base64String2Blob

[Function]

`static public byte[] Base64String2Blob(String strBase64)`

[Purpose]

This function is used to convert a Base64 string into a byte[] array.

[Parameter Description]

strBase64

Base64 string

[Return Value]

Byte[] array

5.4.16 ByteArray2Int

[Function]

`static public boolean ByteArray2Int(byte[] buf, ref int value)`

[Purpose]

This function is used to convert a 4-byte array into an integer.

[Parameter Description]

buf

Byte array

value

Returned data

[Return Value]

true	Succeeded
false	Failed

5.4.17 Int2ByteArray

[Function]

`static public boolean Int2ByteArray(int value, byte[] buf)`

[Purpose]

This function is used to convert an integer into a 4-byte array.

[Parameter Description]

value

Data

buf

Byte array

[Return Value]

true	Succeeded
false	Failed

5.4.18 ExtractFromImage

[Function]

public int ExtractFromImage(String FileName, int DPI, byte[] template, ref int size)

[Purpose]

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

[Parameter Description]

FileName

Full path of a file

DPI

Image DPI

template

Returned fingerprint template (It is recommended that 2048 bytes be pre-allocated.)

size[in/out]

[in]	Template array length
[out]	Fingerprint template length that is actually returned

[Return Value]

0	Succeeded
Others	Failed (See the error code description.)

[Note]

Only the SDK of the standard version supports this function.

5.4.19 SetParameters

[Function]

public int SetParameters(int code, byte[] pramValue, int size

[Purpose]

This function is used to set a parameter.

[Parameter Description]

code

Parameter code (See the Appendixes.)

pramValue

Parameter value

size

Parameter data length

[Return Value]

0	Succeeded

Others Failed (See the error code description.)

5.4.20 GetParameters

[Function]

)public int GetParameters(int code, byte[] paramValue, ref int size)

[Purpose]

This function is used to acquire a parameter.

[Parameter Description]

code

Parameter code (See the Appendixes.)

pramValue

Parameter value

size

Returned parameter data length

[Return Value]

0	Succeeded
Others	Failed (See the error code description.)

6 Appendixes

6.1 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).

Parameter Code	Property	Data Type	Description
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 except the LIVE20R need to call a function to disable the parameter.)	Int	1 indicates that the 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.

6.2 Error Codes

classname:zkfp

```
public static int ZKFP_ERR_ALREADY_INIT = 1;   /**< Initialized */
public static int ZKFP_ERR_OK = 0;        /**< Operation succeeded */
public static int ZKFP_ERR_INITLIB = -1;      /**< Failed to initialize the algorithm library */
public static int ZKFP_ERR_INIT = -2;    /**< Failed to initialize the capture library */
public static int ZKFP_ERR_NO_DEVICE = -3;       /**< No device connected */
public static int ZKFP_ERR_NOT_SUPPORT = -4;   /**< Not supported by the interface */
public static int ZKFP_ERR_INVALID_PARAM = -5;      /**< Invalid parameter */
public static int ZKFP_ERR_OPEN = -6;  /**< Failed to start the device */
public static int ZKFP_ERR_INVALID_HANDLE = -7;    /**< Invalid handle */
public static int ZKFP_ERR_CAPTURE = -8;   /**< Failed to capture the image */
public static int ZKFP_ERR_EXTRACT_FP = -9;      /**< Failed to extract the fingerprint template */
public static int ZKFP_ERR_ABSORT = -10;   /**< Suspension */
public static int ZKFP_ERR_MEMORY_NOT_ENOUGH = -11; /**< Insufficient memory */
public static int ZKFP_ERR_BUSY = -12;        /**< The fingerprint is being captured */
public static int ZKFP_ERR_ADD_FINGER = -13;    /**< Failed to add the fingerprint template */
public static int ZKFP_ERR_DEL_FINGER = -14;    /**< Failed to delete the fingerprint template */
public static int ZKFP_ERR_FAIL = -17;  /**< Operation failed */
public static int ZKFP_ERR_CANCEL = -18;   /**< Capture cancelled */
public static int ZKFP_ERR_VERIFY_FP = -20; /**<       Fingerprint comparison failed */
```

```
    public static int ZKFP_ERR_MERGE =  -22; /**<      Failed to combine registered fingerprint templates
*/
    public static int ZKFP_ERR_NOT_OPENED = -23; /**<    Device not started        */
    public static int ZKFP_ERR_NOT_INIT = -24; /**<  Not initialized      */
    public static int ZKFP_ERR_ALREADY_OPENED = -25; /**<  Device started     */
```