Developer's Guide

SigIDp-R

Contents

Overview

The SigIDp-R Fingerprint SDK for Windows is for use with the Topaz IDLite 1x5 (TF-S463), IDLite LCD 1x5 (TF-LBK463), and IDGem LCD 1x5 (TF-LBK464) combined signature and fingerprint capture devices. SigIDp-R supports Windows 10 or newer desktop operating systems and Windows server 2016 or newer. It also supports virtual desktops and applications using Citrix version 7.15 and newer.

Public Member Functions

int GetFingerprintFile(string sFilePath)

Opens fingerprint device and captures fingerprint, saves the ASCII hex string of fingerprint data to the sFilePath (use TFT extension).

Returns
0 for success; greater than 0 for failure; 1 no fingerprint device detected; 2 invalid file path; 3 operation failure.

int ValidateFingerprintFile(string sFilePath)

Takes the ASCII hex string (TFT format) return from GetFingerprintFile() as basis for comparison.

Returns
0 for success; greater than 0 for failure; 1 no fingerprint device detected; 2 invalid path to TFT file; 3 sFilePath doesn't exist; 4 input file is empty; 5 validation operation failure; all other values – validation failure.

string GetFingerprintString()

Opens fingerprint reader and captures fingerprint as ASCII hex string.

Returns
Fingerprint image as ASCII hex string.

int ValidateFingerprintString(string stringfingerprint)

Takes input stringfingerprint, the ASCII hex string (TFT format) returned from GetFingerprintString() as the basis for comparison.

Returns
0 for validation success; greater than 0 for failure

int SetIdentificationFingerprint()

Opens the fingerprint device and captures a fingerprint to be used for comparison when calling IdentifyUser().

Returns
0 for success; greater than 0 for failure; 1 no fingerprint device detected; 2 could not store fingerprint to internal storage; 3 operation exception.

int IdentifyUser(string stringfingerprint)

Compares input stringfingerprint with the stored fingerprint string captured through GetVerificationFingerPrintString().

Returns
0 for the first user matched, 1 for the second user and continue, up to user 3; -1 for no fingerprint device detected; -2 for no user string setup prior; -3 for input string is empty; -4 for compare failure; -5 for compare with bad result; -6 for operation exception.

int CreateBmp(string sFilePath)

Opens the fingerprint device and captures a fingerprint, then saves the image to sFilePath in bmp format. If sFilePath already existed, it will be overwritten. The image will be 125x125 pixel bitmap with 24-bit true color image.

Returns
0 for success; greater than 0 for failure; 1 no fingerprint device detected; 2 sFilePath is empty; 3 for operation failed.

int CreateJpg(string sFilePath)

Opens the fingerprint device and captures a fingerprint, then saves the image to sFilePath in JPEG format. If sFilePath already existed, it will be overwritten.

Returns
0 for success; greater than 0 for failure; 1 no fingerprint device detected; 2 sFilePath is empty; 3 for operation failed.

int CreateTif(string sFilePath)

Opens the fingerprint device and captures a fingerprint, then saves the image to sFilePath in TIF format. If sFilePath already existed, it will be overwritten.

Returns
0 for success; greater than 0 for failure; 1 no fingerprint device detected; 2 sFilePath is empty; 3 for operation failed.

int CreateLowResBmp(string sFilePath)

Opens the fingerprint device and captures a fingerprint, then saves the image to sFilePath in bitmap format. The image will be saved as 125x125 pixel image in 8-bit grayscale color palette. If sFilePath already existed, it will be overwritten.

Returns
0 for success; greater than 0 for failure; 1 no fingerprint device detected; 2 sFilePath is empty; 3 for operation failed.

int CreateLowResJpg(string sFilePath)

Opens the fingerprint device and captures a fingerprint, then saves the image to sFilePath in low resolution JPEG format. If sFilePath already existed, it will be overwritten.

Returns
0 for success; greater than 0 for failure; 1 no fingerprint device detected; 2 sFilePath is empty; 3 operations failed.

int CreateLowResTif(string sFilePath)

Opens the fingerprint device and captures a fingerprint, then saves the image to sFilePath in TIF format with low resolution. If sFilePath already existed, it will be overwritten.

Returns
0 for success; greater than 0 for failure; 1 no fingerprint device detected; 2 sFilePath is empty; 3 for operation failed.

string GetBmpString()

Opens the fingerprint device and captures a fingerprint. Convert to ASCII hex bitmap string in return. This is useful for web applications to send a fingerprint image on the web.

Returns
ASCII hex string with length greater than one. Returned string with 0 length means operation failure.

int SetBmpString(string stringfingerprint, string sFilePath)

Saves input stringfingerprint as a bitmap in location specified by sFilePath.

Returns
0 for success; greater than 0 for failure; 1 no fingerprint device detected; 2 sFilePath is empty; 3 stringfingerprint is empty or less than 2 bytes long; 4 operations failed.

dynamic GetBmpBufferBytes()

Opens the fingerprint device and captures a fingerprint, then converts to image as an ASCII hex string in byte array format. The byte array will comprise a 125x125 pixel bitmap with a 24-bit true color palette.

Returns
ASCII hex byte array with length greater than one. Returns string with 0 length for operation failure.

dynamic GetBmpBufferBytesLowRes()

Opens the fingerprint device and captures a fingerprint, then converts to image as an ASCII hex string in byte array format. The byte array will comprise a 125x125 pixel bitmap with an 8-bit grayscale color palette.

Returns
ASCII hex byte array with length greater than one. Returns string with 0 length for operation failure.

bool BitmapBufferWrite()

Captures a fingerprint as image once and saves into internal storage. This command should be used with BmpBufferSize() and GetBmpBufferByte() to loop through every byte in the buffer. BmpBufferClose() is called to end this operation sequence.

Returns
true for success, false for failure

bool BitmapBufferWriteLowRes()

Captures a fingerprint as image once and saves into internal storage. This command should be used with BmpBufferSizeLowRes() and GetBmpBufferByteLowRes() to loop through every byte in the buffer. BmpBufferCloseLowRes() is called to end this operation sequence.

Returns
true for success, false for failure.

byte GetBmpBufferByte(int index)

Get bitmap data from the internal buffer by the input index.
This command should be used with BitmapBufferWrite() and BmpBufferSize() to loop through every byte in the buffer.

Returns
Single byte data from the internal buffer.

byte GetBmpBufferByteLowRes(int index)

Get bitmap data from the internal buffer by the input index.
This command should be used with BitmapBufferWriteLowRes() and BmpBufferSizeLowRes() to loop through every byte in the buffer.

Returns
Single byte data from the internal buffer.

int BitmapBufferSize()

Returns size of captured fingerprint image in internal storage. This command should be used with BmpBufferWrite() and GetBmpBufferByte() to loop through every byte in the buffer.

Returns
The length of the internal buffer.

int BitmapBufferSizeLowRes()

Returns size of captured fingerprint low resolution image. This command should be used with BmpBufferWriteLowRes() and GetBmpBufferByteLowRes() to loop through every byte in the buffer.

Returns
The length of the internal buffer.

bool BmpBufferClose()

Releases captured image storage used by BitmapBufferWrite().

Returns
true for success; false for failure.

bool BmpBufferCloseLowRes()

Releases captured image storage used by BitmapBufferWriteLowRes().

Returns
true for success; false for failure.

int CloseIdentificationFingerprint()

Releases the storage allocated by SetIdentificationFingerprint().

Returns
0 for success; greater than 0 for failure.

bool ConnectQuery()

Verifies that the fingerprint device is connected.

Returns
true for device found; false for device not found.

int GetDialogDelay()

Gets the dialog delay time set within the ActiveX.

Returns
Dialog delay time in milliseconds.

void SetWindowPosMode(int mode)

Sets fingerprint dialog window position mode to one of the following values:

Mode

0	Auto, default value, parent windows center
1	Manual - user defined location
2	Center of current screen
3	Top left of current screen
4	Top center of current screen
5	Top right of current screen
6	Center right of current screen
7	Bottom right of current screen
8	Bottom center of current screen
9	Bottom left of current screen
10	Left center of current screen

void SetWindowPos(int winPosX, int winPosY)

Sets fingerprint dialog window position when PosMode is set to manual. winPosX is the number of pixels in x axis from top left corner of current screen, winPosY is the number of pixels in y axis from top left corner of the current screen.