

UniFinger Engine Image SDK Reference Manual

Version 2.0



Contents

SF_Initialize	2
SF_Uninitialize.....	3
SF_Extraction	4
SF_GetEnrollQuality	5
SF_Matching.....	6
SF_MatchingDB	7
SF_AbortMatching.....	8
SF_RotateTemplate	9
Constant List.....	10
Revision Notes.....	11
Contact	11

SF_Initialize

The **SF_Initialize** function initializes the fingerprint sensor.

int SF_Initialize(void)

Parameters

This function has no parameters.

Return Values

Nonzero if the initialization succeeds; otherwise 0.

SF_Uninitialize

The **SF_Uninitialize** function uninitialize the fingerprint sensor.

void SF_Uninitialize(void)

Parameters

This function has no parameters.

Return Values

This function has no return values.

SF_Extraction

The **SF_Extraction** function yields the template of fingerprint from the fingerprint image.

int SF_Extraction(unsigned char* Image, int Width, int Height, int Resolution, int LimitSize, unsigned char* Template, int* TemplateSize)

Parameters

Image

A pointer to the buffer that contains the fingerprint image.

Width

Width of fingerprint image in pixel. It cannot exceed 640 in case of 500 DPI image.

Height

Height of fingerprint image in pixel. It cannot exceed 640 in case of 500 DPI image.

Resolution

Resolution of fingerprint image in DPI. Many fingerprint images have 500 DPI resolution typically.

LimitSize

Size of template. The larger value can improves authentication performance. It can be from SF_TEMPLATESIZE_MIN to SF_TEMPLATESIZE_MAX. The value 384 is suitable in general case.

Template

A pointer to the buffer that stores the template of fingerprint. *LimitSize* bytes should be allocated to this buffer at least.

TemplateSize

A pointer to the integer variable that contains the actual size of template. If the exact size of template is not needed, *TemplateSize* can be NULL.

Return Values

Nonzero if the generation of template succeeds; otherwise 0.

SF_GetEnrollQuality

The **SF_GetEnrollQuality** function returns quality of processed image.

int SF_GetEnrollQuality(void)

Parameters

This function has no parameters.

Return Values

Quality value which is from 1 to 100. Typically this value should be above 30 for further processing such as enroll and matching. Especially in case of enrollment, the use of good quality image (above 50) is highly recommended.

Remarks

This function returns valid value, only after the **SF_Extraction** function is called.

SF_Matching

The **SF_Matching** function matches two fingerprint templates extracted using the **SF_Extract** function.

int SF_Matching(unsigned char* Template1, unsigned char* Template2, int SecurityLevel)

Parameters

Template1, Template2

A pointer to the buffer that stores the template of fingerprint to be compared.

SecurityLevel

Specifies the level of security. This is in the range 1 to 7. If *SecurityLevel* is 7, the most secure verification is guaranteed. The value 4 is suitable in general case. Physical meaning of *SecurityLevel* is as follows.

<i>SecurityLevel</i>	False Accept Rate (FAR)
1	Below 1 % (1e-2)
2	Below 0.1 % (1e-3)
3	Below 0.01 % (1e-4)
4	Below 0.001 % (1e-5)
5	Below 0.0001 % (1e-6)
6	Below 0.00001 % (1e-7)
7	Below 0.000001 % (1e-8)

Return Values

SF_SUCCESS

The matching succeeds.

SF_MATCHFAIL

The matching fails.

SF_MatchingDB

The **SF_MatchingDB** function matches one fingerprint template with the DB of fingerprint templates.

int SF_MatchingDB(unsigned char TemplateDB, int Count, int* Match, unsigned char* Template, int SecurityLevel, int Timeout)**

Parameters

TemplateDB

A 2-d pointer to the buffer that stores the fingerprint templates enrolled extracted.

Count

Specifies a number of the fingerprint templates enrolled formerly.

Match

A pointer to the value that stores the order of templates matched.

Template

A pointer to the buffer that stores the template of fingerprint to be compared.

SecurityLevel

Specifies the level of security. This is same with the **SF_Matching** function.

Timeout

Specifies maximum time for matching in milliseconds. If elapsed time for matching exceeds timeout, function stops further matching and returns error code. This time is only for matching, not for extraction. If *Timeout* is 0, no timeout occurs.

Return Values

SF_SUCCESS

The matching succeeds.

SF_MATCHFAIL

The matching fails.

SF_MATCHTIMEOUT

The elapsed time for matching exceeds specified timeout.

Remarks

This function can be used for fast 1:N matching.

SF_AbortMatching

The **SF_AbortMatching** function aborts matching process of The **SF_MatchingDB** function.

void SF_AbortMatching(void)

Parameters

This function has no parameters.

Return Values

This function has no return values.

SF_RotateTemplate

The **SF_RotateTemplate** function rotates the template of fingerprint enrolled formerly to the amount of 180 degrees.

void SF_RotateTemplate(unsigned char* Template)

Parameters

Template

A pointer to the buffer that stores the template of fingerprint extracted formerly.

Return Values

This function has no return values.

Remarks

The **SF_RotateTemplate** function can be used to support 360 degree rotational invariant matching. The **SF_Matching** function permits 180 degree (-90 ~ +90 degree) rotation.

Example

```
unsigned char m_EnrollTemplate[ SF_TEMPLATESIZE_MAX ]; // original enrolled template
unsigned char m_RotatedTemplate[ SF_TEMPLATESIZE_MAX ];
memcpy( m_RotatedTemplate, m_EnrollTemplate, SF_TEMPLATESIZE_MAX );
RotateTemplate( m_RotatedTemplate );
// For the SF_Matching function call, both m_EnrollTemplate and m_RotatedTemplate should be used.
```

Constant List

Constant Name	Value
SF_TEMPLATESIZE_MIN	256
SF_TEMPLATESIZE_MAX	1024
SF_VERIFYFAIL	0
SF_MATCHFAIL	0
SF_SUCCESS	1
SF_MATCHTIMEOUT	128
SF_MATCHABORT	129

Revision Notes

V2.0 2005-2-4 Created.

Contact

Suprema Inc.

Dongcheon Bldg., 13-21 Yangjae-dong, Seocho-gu, Seoul, 137-130, Korea

TEL: +82-2-571-9305

FAX: +82-2-571-9306

E-mail: support@supremainc.com, sales@supremainc.com

Web: www.supremainc.com