RT1011 Fingerprint Development Kit Instructions

Catalog

1. Fingerprint parameters

Name of Technical Indicators	Index value
Real test value of FRR rejection rate	0.005% below
False Recognition Rate FAR Actual Test Value	Less than 0.00008%
Interface standard	USB
Antistatic index	IEC-61000-4-2 > 15KV
Peak power consumption	110 mA
working temperature	- 10 C to 60 C
Working humidity	0%~98%
working voltage	5V

2. Fingerprint image

The size of the fingerprint image is 256*288, or 16384 bytes.8-bit RAW format image.When fingerprint images are captured by functions, the number of declared received image data should not be less than 73728.It is generally defined as 73728.

Fingerprint image resolution is 500 DPI

3. Fingerprint template

Fingerprint template is a set of fingerprint feature points, which is composed of fingerprint feature points.

In SDK, fingerprint template is divided into reference template and matching template.Reference template is generated when fingerprint is registered. It is

generated by capturing fingerprint images many times.The matching template is generated from a fingerprint image collected at one time.Generally, matching templates are used to compare with reference templates to confirm the recognition results.

The reference template size is fixed, 512 BBYTE.

The matching template size is fixed, 256 BYTE.

4. Constant Definition

FPM_DEVICE=1 & Equipment
FPM_PLACE=02 & & Please press your finger
FPM_LIFT=03 & & Please raise your finger
FPM_CAPTURE=4 & & Image Acquisition Completed
FPM_GENCHAR=5 & & Acquisition Matching Template Completion
FPM_ENRFPT=6 & Registration Reference Template Completion
FPM_NEWIMAGE=7 & & New Fingerprint Image
FPM_TIMEOUT=8 & timeout
FPM_IMGVAL=9 & Image Quality

REFTPSIZE = 512 & Reference Template Size
MATTPSIZE = 256 & Matching Template Size
IMGWIDTH= 256 & Image Width
IMGHEIGHT = 288 & Image Height
IMGSIZE = 73728 & Image data size

SREFTPSIZE = 768 & Reference Template Size (String Template)
SMATTPSIZE = 384 & Match Template Size (String Template)

5. Function description:

Turn on the shutdown device
Int WINAPI OpenDevice (int comnum, int nbaud, int style); * Function: Open fingerprint device * Parameters: All three parameters are integers for setting device parameters, where all 0, OpenDevice (0, 0, 0) is used.

* Return value: Return integer, Return 1 indicates that the device was successfully opened, otherwise it means that the device failed to open.

Int WINAPI LinkDevice ();
* Function: Connecting fingerprint devices to use after OpenDevice succeeds is actually to verify communication reliability
* Parameters: No parameters
* Return value: Return integer, Return 1 indicates connection success, otherwise connection failure
Note: In the program, open the fingerprint device, and both OpenDevice and LinkDevice return 1 to indicate that the device was successfully opened.

Int WINAPI CloseDevice ();
* Function: Turn off the open fingerprint device
* Parameters: No parameters
* Return value: Return integer, Return 1 indicates closure success, otherwise closure failure, usually in the program, do not need to determine the value of CloseDevice

Registration reference template and acquisition matching template

Void WINAPI EnrolFpChar ();
* Function: Start to register fingerprint reference template
* Parameters: No parameters
* Return value: No return value.Result returned by message

Void WINAPI GenFpChar ();
* Function: Start collecting fingerprint matching template
* Parameters: No parameters
* Return value: No return value.Result returned by message

Getting fingerprint messages

Int WINAPI GetWorkMsg ();
* Function: Get fingerprint work messages
* Parameters: No parameters
* Return value: Return integer. Return number greater than 1 indicates valid fingerprint work message. The meaning of each value is determined by the above constant.

Int WINAPI GetRetMsg ();
* Function: Get Fingerprint Processing Return Message
* Parameters: No parameters
* Return value: Return integer, return 1, indicating that the function in the current fingerprint mode is successfully executed, otherwise it will fail.

Acquisition of the corresponding data after successful registration reference template and acquisition of matching template

BOOL WINAPI GetFpCharByEnl (BYTE * fpbuf, int * fpsize);
* Function: Get the reference template, which is in binary format.
* Parameter: The first parameter is pre-allocated memory space, and the second parameter is

used to return the actual size of the template. * Return value: Return integer, return 1 indicates success or failure
BOOL WINAPI GetFpStrByEnl (char * fpstr); * Function: Get the reference template, which is in string format. * Parameter: The parameter is pre-allocated memory space * Return value: Return integer, return 1 indicates success or failure
BOOL WINAPI GetFpCharByGen (BYTE * tpbuf, int * tpsize); * Function: Get the matching template, which is in binary format * Parameter: The first parameter is pre-allocated memory space, and the second parameter is used to return the actual size of the template. * Return value: Return integer, return 1 indicates success or failure
BOOL WINAPI GetFpStrByGen (char * tpstr); * Function: Get the matching template to get the reference template. The obtained template is in string format. * Parameter: The parameter is pre-allocated memory space * Return value: Return integer, return 1 indicates success or failure
Note: All of the above functions are executed in the process of getting fingerprint messages

template matching
Int WINAPI Match Template One (BYTE * pSrcData, BYTE * pDstData, int nDstSize); * Function: Compare reference template and matching template in binary format * Parameters: The first parameter is the matching template, the second parameter is the reference template, and the third parameter is the size of the reference template, 512. * Return value: Return the score of fingerprint alignment, more than 100 generally means that the alignment is successful. Modifying this score can provide security level.
Int WINAPI MatchTemplateEx (char * pSrcData, char * pDstData); * Function: Compare reference template and matching template in string format * Parameters: The first parameter is the matching template, and the second parameter is the reference template. * Return value: Return the score of fingerprint alignment, more than 100 generally means that the alignment is successful. Modifying this score can provide security level.
Int WINAPI Match Template OneEx (BYTE * pSrcData, int nSrcSize, BYTE * pDstData, int nDstSize); * Function: Comparing reference templates and reference templates in binary format * Parameters: The first parameter is reference template, the second parameter is reference template size, the third parameter is reference template, and the fourth parameter is reference template size. Both reference templates are 512. * Return value: Return the score of fingerprint alignment, more than 100 generally means that the alignment is successful. Modifying this score can provide security level.
Note: Templates have two formats and require one-to-one correspondence.String formats cannot be mixed with binary formats.Templates in binary format can provide speed for comparison.Templates in string format provide ease of use.

Drawing fingerprint image

Int WINAPI DrawImage (HDC hdc, int left, int top);
* Function: Draw on a specified window
* Parameters: The first parameter is the drawing device handle, which is obtained by GetDC, the second parameter is the X coordinate of the drawing, and the third parameter is the Y coordinate of the drawing.
* Return value: integer, return 1 indicates success of drawing, return 0 indicates failure of drawing.

6. Use functions

The basic flow chart of the above functions is as follows:
In the graph, the template for acquisition and comparison is in string format, and the template for binary format is in the same way.
In practice, the reference template is usually stored in the database with the corresponding user information when registering.When recognizing, we only need to collect the matching template, and then compare the matching template with the reference template in the database to determine that the current finger is the finger of the user in the database.(That is, whether it's 1:1 or 1:N, you just need to collect the matching template once, and then use the MatchTemplateEx function to determine whether the matching is successful.)