



Wilson Technology Corporation

11F-2, No. 289, Sec. 2, Guang-Fu Rd., Hsin-Chu 300, Taiwan, R.O.C.

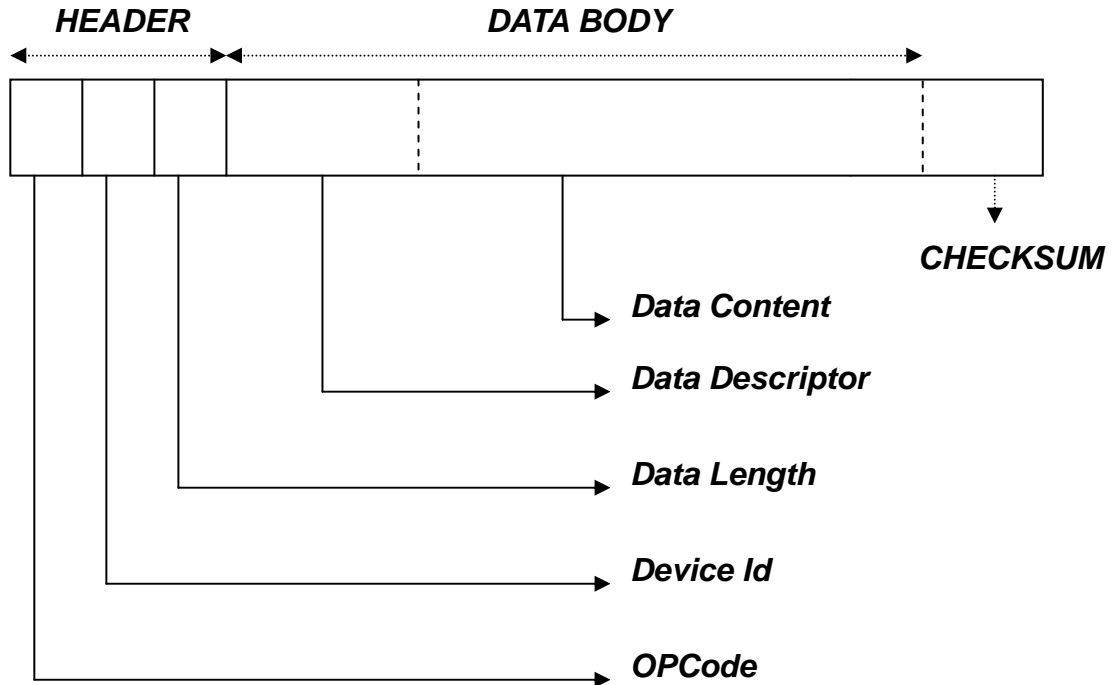
Tel: 886-3-5163339

Fax: 886-3-5163679

WM Series
Software Development Kit
Programmer's Guide

Version Release 3.06

■ Protocol Definition



As shown above, the protocol consists of three parts, **HEADER**, **DATA BODY** and **CHECKSUM**. The largest occupied size of the **DATA BODY** is **568 bytes** while the size of the **HEADER** and **CHECKSUM** are **6 bytes** and **2 bytes** respectively, totally **576 bytes** for a fixed packet.

➤ **HEADER**

The **HEADER** is divided into 3 parts:

1. **OPCode**: 1 byte. This code is to define the function of the protocol. The detailed definition is listed below:

Code	Value
OP_FP_DIAGNOSE	0x30
OP_FP_CHECKBLANK	0x31
OP_FP_ENROLL	0x32
OP_FP_RELEASEENROLL	0x33
OP_FP_VERIFY	0x34
OP_FP_IDENTIFY	0x35
OP_FP_CARDMATCH	0x36

OP_DB_DELETEUSERID	0x40
OP_DB_DELETEALLUSER	0x41
OP_DB_GETUSERINFO	0x42
OP_DB_PUTUSERINFO	0x43
OP_DB_GETIDLIST	0x44
OP_SYS_LOADDEFAULTSETTING	0x50
OP_SYS_UPDATEPROGRAM	0x51
OP_SYS_SNAP	0x52
OP_SYS_GETIMAGE	0x53
OP_SYS_GETDEVICEID	0x54
OP_SYS_SETDEVICEID	0x55
OP_SYS_GETCONFIG	0x56
OP_SYS_SETCONFIG	0x57
OP_SYS_GETGPIO	0x58
OP_SYS_SETGPIO	0x59
OP_SYS_TUNEIMAGE	0x5A
OP_SYS_GETIMAGESIZE	0x5C
OP_SYS_SETPOWERMODE	0x6F

2. **Device Id:** 3 bytes, used for RS232/RS485 to identify the ID of a specified device.
This ID will be compared with the ID previously stored in the system.
3. **Data Length:** 2 bytes. Represent the length of the **DATABODY** currently transmitted.

➤ **DATA BODY**

The DATA BODY is divided into 2 parts:

1. **DATA DESCRIPTOR:** A 8-byte space which allows 8 parameters, each with 1 byte, to be carried. The value of each byte is the size of the type. For example, char -> 1, short -> 2, long-> 4. See details of the function description.
2. **DATA CONTENT:** The actual data of the packet to be sent/received.

➤ **CHECKSUM**

This value is calculated by summing the value of the first byte (Packet[0]) to the value of 573th byte (Packet[573]) and 'mod' by 256*256. This is to confirm the full packet be received correctly.



Wison Technology Corporation

11F-2, No. 289, Sec. 2, Guang-Fu Rd., Hsin-Chu 300, Taiwan, R.O.C.

Tel: 886-3-5163339

Fax: 886-3-5163679

■ **Function List**

Fingerprint Functions	
WIS_TestDevice	WIS_CheckNoFinger
WIS_Enroll	WIS_ReleaseEnroll
WIS_VerifyID	WIS_Identify
WIS_VerifyTemplate	
DB Operation Functions	
WIS_DeleteUserData	WIS_DeleteAllUser
WIS_ReadUserData	WIS_WriteUserData
WIS_GetUserList	
System Functions	
WIS_LoadDefaultSetting	WIS_UpdateProgram
WIS_Snap	WIS_GetImage
WIS_GetDeviceId	WIS_SetDeviceId
WIS_GetConfig	WIS_SetConfig
WIS_GetGPIO	WIS_SetGPIO
WIS_SetParameter	WIS_GetImageSize
WIS_SetPowerMode	

■ Function Description

➤ Fingerprint Functions

1. **short WIS_TestDevice()**

Description

Test if the fingerprint device is OK.

Parameter

None

Return Value

-1	There is problem with your fingerprint system.
0	The fingerprint device is OK.
-10	FLASH_READ_ERROR
-41	DEVICE_CONNECT_FAIL

Remarks

This function diagnoses your fingerprint device. Before testing, please clean the capture area and *make sure that there is no finger on the reader.*

Example

* Transmitted Data, totally 16 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0030
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x0008
DATA BODY	Descriptor	Packet[6..13]	8	0
	Content	---	0	---
CHECKSUM		Packet[14..15]	2	0x0038

- ✧ The 0x30(OP_FP_DIAGNOSE) value of **OPCode** indicates the function of WIS_TestDevice().
- ✧ The **Device Id** is set to 0, meaning no **Device Id** is required.
- ✧ The 0x0008 value of **Data Length** indicates the total bytes of the DATABODY.
- ✧ All the values of the **Descriptor** are set to 0, meaning no parameters will be sent.
- ✧ The **CHECKSUM** value is equal to the sum of the value mod by 256*256.

* Received Data, totally 18 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0030
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000A
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7..13]	7	0
	Content	Packet[14..15]	2	0 or -1 or -10 or -41
CHECKSUM		Packet[16..17]	2	0x003C

- ✧ The 0x30(OP_FP_DIAGNOSE) value of **OPCode** indicates the function of WIS_TestDevice().
- ✧ The **Device Id** is set to 0, meaning no **Device Id** is required.
- ✧ The 0x000a value of **Data Length** indicates the total bytes of the DATABODY.
 - Only the first byte of the descriptor is set to value of 0x0002, indicating that the function contains only one parameter of 2 bytes (may be in the type of **short**). The value of this parameter will be in the first 2 bytes of the content.
- ✧ The **CHECKSUM** value is equal to the sum of the value mod by 256*256.



Wison Technology Corporation

11F-2, No. 289, Sec. 2, Guang-Fu Rd., Hsin-Chu 300, Taiwan, R.O.C.

Tel: 886-3-5163339

Fax: 886-3-5163679

2. **short WIS_CheckNoFinger()**

Description

To check if there is any fingerprint on the reader.

Parameter

None

Return Value

-1	There is a fingerprint on the reader.
0	There is no fingerprint on the reader.
-10	FLASH_READ_ERROR
-41	DEVICE_CONNECT_FAIL

Remarks

This function is mainly used in the enrollment process. To get the stable and real features of a fingerprint during the enrollment, the user must remove his finger from the reader after **WIS_Enroll()** has successfully been processed for this snapped fingerprint image. You can check if a fingerprint has actually been lifted off the reader by using this function.

Example

* Transmitted Data, totally 16 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0031
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x0008
DATA BODY	Descriptor	Packet[6..13]	8	0
	Content	---	0	---
CHECKSUM		Packet[14..15]	2	0x0039

* Received Data, totally 18 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0031
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000A
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7..13]	7	0
	Content	Packet[14..15]	2	0 or -1 or -10 or -41
CHECKSUM		Packet[16..17]	2	0x003D

3. **short WIS_Enroll(UNSIGNED CHAR* rEnrTemplate)**

Description

To generate a final fingerprint code, *EnrTemplate*.

Parameter

rEnrTemplate: a returned template which is generated after successful enrollment.

The size of rEnrTemplate is equal to 160/320/480 bytes depending on the template type in the configuration.

Return Value

0x41	QUALITY_A, indicates the quality of the enrolled fingerprint and enrollment succeeded.
0x42	QUALITY_B, indicates the quality of the enrolled fingerprint and enrollment succeeded.
0x43	QUALITY_C, indicates the quality of the enrolled fingerprint and enrollment succeeded.
0x44	QUALITY_D, indicates the quality of the enrolled fingerprint and enrollment succeeded.
-7	QUALITY_NOT_YET, enrollment is not completed
-10	FLASH_READ_ERROR
-41	DEVICE_CONNECT_FAIL
-42	CAPTURE_TIMEOUT

Remarks

This function generates the final fingerprint code, *EnrTemplate*, from several input *RawTemplate* by collecting their common features. The purpose of enrollment is to get enough stable characteristics to represent the corresponding fingerprint. You should call **WIS_ReleaseEnroll()** to release the system resource. Basically, the kernel process of enrollment works in a continuous loop as following:

- (a) Use **WIS_Enroll()** to get a good-enough fingerprint.
- (b) If the return value is not one of the qualities defined, repeat step 1 and until the *quality* of the fingerprint is derived.
- (c) If you have tried more than 5 times and still cannot derive the *quality* of the finger, it means that the finger you have chosen to enroll may not be good enough. You should change to another finger and restart the enrollment.
- (d) If you want to improve the enrolled quality, you can continue executing step 1 and 2 to get a better final fingerprint code with better *quality*.
- (e) If you have tried to *enhance* the enrolled quality more than 3 times but the *quality* still remains in a certain quality without any improvement, it seems that



Wison Technology Corporation

11F-2, No. 289, Sec. 2, Guang-Fu Rd., Hsin-Chu 300, Taiwan, R.O.C.

Tel: 886-3-5163339 Fax: 886-3-5163679

the enrolled *quality* has been stable. Any attempt to enhancement may be in vain. You should stop the enrollment with the stable enrolled *quality*. If you are not satisfied with the current enrolled *quality*, choose another finger and restart the enrollment.

(f) call **WIS_ReleaseEnroll()** to free the resource.

Example

Input: None

Output: rEnrlTemplate

* Transmitted Data, totally 16 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0032
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x0008
DATA BODY	Descriptor	Packet[6..13]	8	0
	Content	---	0	---
CHECKSUM		Packet[14..15]	2	0x003A

* Received Data, totally 178 bytes for Template of 160 bytes

* Received Data, totally 178 +160 bytes for Template of 320 bytes

* Received Data, totally 178 +320 bytes for Template of 480 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0032
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x00AA
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7]	1	0x00A0
		Packet[8..13]	6	0
	Content	Packet[14..15]	2	-7 (or 0x41/ 0x42/0x43/0x 44, -10,-32,-4 1,-42)
		Packet[16..175]	160	rEnrlTemplate
CHECKSUM		Packet[176..177]	2	<variable>

* <variable> : This value varies depending on the value of rEnrlTemplate.



Wison Technology Corporation

11F-2, No. 289, Sec. 2, Guang-Fu Rd., Hsin-Chu 300, Taiwan, R.O.C.

Tel: 886-3-5163339

Fax: 886-3-5163679

4. short WIS_ReleaseEnroll()

Description

To release all the internal resource which created during the enrollment process.

Parameter

None

Return Value

0	resource is released successfully
-1	Fail

Remarks

This function releases all the internal resource created during the enrollment process. Call this function only if **WIS_Enroll()** is no longer in use.

Example

* Transmitted Data, totally 16 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0033
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x0008
DATA BODY	Descriptor	Packet[6..13]	8	0
	Content	---	0	---
CHECKSUM		Packet[14..15]	2	0x003B

* Received Data, totally 18 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0033
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000A
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7..13]	7	0
	Content	Packet[14..15]	2	0 or -1
CHECKSUM		Packet[16..17]	2	0x00BF

5. **short WIS_VerifyID(unsigned char* ID, short Security, short* rScore, unsigned char* rFid, unsigned char* rFingerNo)**

Description

To do 1-1 fingerprint verification by giving a specify user ID, security level and fingerprint.

Parameter

ID: an input string of less than 8 bytes to specify a user's ID.

Security: an input value for security level where SECURITY_A gives the highest security.

AUTO_SECURITY	0
SECURITY_A	1
SECURITY_B	2
SECURITY_C	3
SECURITY_D	4

rScore: an output value which means the similarity, ranges from 0 to 100.

rFid: an output value which means the matched finger id.

rFingerNo: an output value which indicate the finger order. The order is from 0x31 ~ 0x33.

Return Value

0	Verified, rScore ranges from 55 to 100.
-1	Not verified, rScore <55
-8	INVALID_SECURITY : improper security level
-10	FLASH_READ_ERROR
-41	DEVICE_CONNECT_FAIL
-42	CAPTURE_TIMEOUT

Remarks

Before calling this function, you must prompt user to place his(her) finger as **WIS_VerifyID()** will capture the image from the fingerprint device to generate the feature, *RawTemplate*. This function then verifies this *RawTemplate* with the *input EnrlTemplate* that has already stored in the fingerprint module.



Wilson Technology Corporation

11F-2, No. 289, Sec. 2, Guang-Fu Rd., Hsin-Chu 300, Taiwan, R.O.C.

Tel: 886-3-5163339

Fax: 886-3-5163679

Example

Input: ID="TEST" Security=SECURITY_C

Output: rScore=100 rFid=0x05 rFingerNo=0x01(the first enrolled finger)

* Transmitted Data, totally 22 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0034
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000E
DATA BODY	Descriptor	Packet[6]	1	0x0004
		Packet[7]	1	0x0002
		Packet[8..13]	6	0
	Content	Packet[14..17]	4	0x54 0x45 0x53 0x54 (TEST)
		Packet[18..19]	2	0x0003
CHECKSUM		Packet[20..21]	2	0x018B

* Received Data, totally 22 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0034
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000E
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7]	1	0x0002
		Packet[8]	1	0x0001
		Packet[9]	1	0x0001
		Packet[10..13]	4	0
	Content	Packet[14..15]	2	0(or -1,-10,-4 1,-42)
		Packet[16..17]	2	0x0064 (0x00 ~ 0x64)
		Packet[18]	1	0x0005
	Packet[19]	1	0x0001	
CHECKSUM		Packet[20..21]	2	0x00B2



Wison Technology Corporation

11F-2, No. 289, Sec. 2, Guang-Fu Rd., Hsin-Chu 300, Taiwan, R.O.C.

Tel: 886-3-5163339

Fax: 886-3-5163679

6. **short WIS_Identify(short Security, unsigned char* rID, short rScore, unsigned char* rFid, unsigned char* rFingerNo)**

Description

To do 1-N fingerprint identification with the enrolled templates saved in the system.

Parameter

Security: an input value for security level. See **WIS_VerifyID()** for details.

rID: a returned ID string which give a matched score or highest score.

rScore: an output value which means the similarity, ranges from 0 to 100.

rFid: an output value which means the matched finger id.

rFingerNo: an output value which indicate the finger order. The order is from 0x31 ~ 0x33.

Return Value

0	Verified, rScore ranges from 55 to 100.
-1	Not verified, rScore <55
-8	INVALID_SECURITY : improper security level
-10	FLASH_READ_ERROR
-41	DEVICE_CONNECT_FAIL
-42	CAPTURE_TIMEOUT

Remarks

Before calling this function, you must prompt user to place his(her) finger as **WIS_Identify()** will capture the image from the fingerprint device to generate the feature, *RawTemplate*. The function then matches this *RawTemplate* with every *EnrlTemplate* that is stored in the fingerprint module and returns the highest score.

Example

Input: Security=SECURITY_C

Output: rID="TEST" rScore=100 rFid=0x05 rFingerNo=0x01

* Transmitted Data, totally 18 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0035
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000A
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7..13]	7	0
	Content	Packet[14..15]	2	0x0003
CHECKSUM		Packet[16..17]	2	0x0044

* Received Data, totally 26 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0035
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x0012
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7]	1	0x0004
		Packet[8]	1	0x0002
		Packet[9]	1	0x0001
		Packet[10]	1	0x0001
		Packet[11..13]	3	0
	Content	Packet[14..15]	2	0(or -1,-10,-41,-42)
		Packet[16..19]	4	0x54 0x45 0x53 0x54 (TEST)
		Packet[20..21]	2	0x0064 (0x00 ~ 0x64)
		Packet[22]	1	0x0005
		Packet[23]	1	0x0001
CHECKSUM		Packet[24..25]	2	0x01FB



Wison Technology Corporation

11F-2, No. 289, Sec. 2, Guang-Fu Rd., Hsin-Chu 300, Taiwan, R.O.C.

Tel: 886-3-5163339

Fax: 886-3-5163679

7. **short WIS_VerifyTemplate(unsigned char* EnrlTemplate, short Security, short *rScore)**

Description

To do 1-1 verification by giving a specified EnrlTemplate, Security level and fingerprint.

Parameter

EnrlTemplate: the enrolled template input for verification with the fingerprint.

Security: an input value for security level where SECURITY_A gives the highest security. See **WIS_VerifyID()** for details.

rScore: an output value which means the similarity, ranges from 0 to 100.

Return Value

0	Verified, rScore ranges from 55 to 100.
-1	Not verified, rScore <55
-3	OUT_OF_MEMORY : insufficient memory
-6	INVALID_TEMPLATE : illegal enrolled template
-8	INVALID_SECURITY : improper security level
-10	FLASH_READ_ERROR
-41	DEVICE_CONNECT_FAIL
-42	CAPTURE_TIMEOUT

Remarks

Before calling this function, you must prompt user to place his(her) finger as **WIS_VerifyTemplate** will capture the image from the fingerprint device to generate the feature, *RawTemplate*. This function then verifies this *RawTemplate* with the *input EnrlTemplate*.

Example

Input: EnrlTemplate=<Template> Security=SECURITY_D

Output: rScore=5

- * Transmitted Data, totally 178 bytes for template of 160 bytes
- * Transmitted Data, totally 178 +160 bytes for template of 320 bytes
- * Transmitted Data, totally 178 + 320 bytes for template of 480 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0036
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x00AA
DATA BODY	Descriptor	Packet[6]	1	0x00A0
		Packet[7]	1	0x0002
		Packet[8..13]	6	0
	Content	Packet[14..173]	160/320/ 480	EnrlTemplate
		Packet[174..175]	2	0x0004
CHECKSUM		Packet[176..177]	2	<Variable>

- * EnrlTemplate : an array of 160/320/480 bytes.
- * <Variable> : This value varies depending on the value of EnrlTemplate.
- * Received Data, totally 20 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0036
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000C
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7]	1	0x0002
		Packet[8..13]	6	0
	Content	Packet[14..15]	2	-1(or 0,-3, -6,-8,-10,-41, -42)
		Packet[16..17]	2	0x0005 (0x00 ~ 0x64)
CHECKSUM		Packet[18..19]	2	0x0249

➤ **DB Operation Functions**

1. **short WIS_DeleteUserData(unsigned char *ID)**

Description

Delete the user's information that stored in the fingerprint module.

Parameter

ID: an input string of less than 8 bytes to specify a user's ID.

Return Value

0	Succeed
-1	Fail
-10	FLASH_READ_ERROR
-11	FLASH_WRITE_ERROR
-32	USER_NOT_FOUND

Example

Input: ID="TEST"

* Transmitted Data, totally 20 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0040
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000C
DATA BODY	Descriptor	Packet[6]	1	0x0004
		Packet[7..13]	7	0
	Content	Packet[14..17]	4	0x54 0x45 0x53 0x54 (TEST)
CHECKSUM		Packet[18..19]	2	0x0190

* Received Data, totally 18 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0040
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000A
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7..13]	7	0
	Content	Packet[14..15]	2	0(or -1,-10, -11,-32)
CHECKSUM		Packet[16..17]	2	0x004C

2. **short WIS_DeleteAllUser()**

Description

Delete all the users information that stored in the fingerprint module.

Return Value

0	Succeed
-1	Fail
-10	FLASH_READ_ERROR
-11	FLASH_WRITE_ERROR

Example

* Transmitted Data, totally 16 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0041
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x0008
DATA BODY	Descriptor	Packet[6..13]	7	0
	Content	---	0	---
CHECKSUM		Packet[14..15]	2	0x0049

* Received Data, totally 18 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0041
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000A
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7..13]	7	0
	Content	Packet[14..15]	2	0(or -1,-10,-11)
CHECKSUM		Packet[16..17]	2	0x004D



Wison Technology Corporation

11F-2, No. 289, Sec. 2, Guang-Fu Rd., Hsin-Chu 300, Taiwan, R.O.C.

Tel: 886-3-5163339

Fax: 886-3-5163679

3. **short WIS_ReadUserData(unsigned char* ID, unsigned char FingerNo, unsigned char *rUserInfo)**

Description

Load a specified user's information that stored in the fingerprint module.

Parameter

ID: an input string of less than 8 bytes to specify a user's ID.

FingerNo: is a char to specify which enrolled finger's information to be loaded.

'1': means first enrolled finger.

'2': means second enrolled finger.

'3': means third enrolled finger.

rUserInfo: an output structure WIS_USER_INFO, which carries the user's information.

```
typedef struct Wis_User_Info
{
    unsigned char id[16];
    unsigned char role;          /* Administrator : 0x00   User : 0x01 */
    unsigned char password[8];
    unsigned char securityLevel;
    unsigned char fingerNo;     /* 1st, 2nd, 3rd, Finger */
    unsigned char fingerId;     /* 0 ~ 9 */
    unsigned char quality;      /* Enrolled quality */
    unsigned char enrTemplate[160/320/480]; /* Check Template type
    unsigned char reserved[3];
} WIS_USER_INFO;
```

Return Value

0	Succeed
-1	Fail
-10	FLASH_READ_ERROR
-21	INVALID_FINGERNO
-23	INVALID_OPERATION
-32	USER_NOT_FOUND



Wisron Technology Corporation

11F-2, No. 289, Sec. 2, Guang-Fu Rd., Hsin-Chu 300, Taiwan, R.O.C.

Tel: 886-3-5163339

Fax: 886-3-5163679

Example

Input: ID = "TEST" FingerNo=0x01

Output: rUserInfo

* Transmitted Data, totally 21 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0042
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000D
DATA BODY	Descriptor	Packet[6]	1	0x0004
		Packet[7]	1	0x0001
		Packet[8..13]	6	0
	Content	Packet[14..17]	4	0x54 0x45 0x53 0x54 (TEST)
		Packet[18]	1	0x01
CHECKSUM		Packet[19..20]	2	0x0195

* Received Data, totally 210 bytes for template type of 160 bytes

* Received Data, totally 210 + 160 bytes for template type of 320 bytes

* Received Data, totally 210 + 320 bytes for template type of 480 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0042
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x00C2
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7]	1	0x00B8
		Packet[8..13]	6	0
	Content	Packet[14..15]	2	0(or -1,-10, -21,-23,-32)
		Packet[16..207]	192/352/ 512	<rUserInfo>
CHECKSUM		Packet[208..209]	2	<Variable>

* <rUserInfo> : output structure WIS_USER_INFO

* <Variable> : This value varies depending on the value of rUserInfo.



Wisron Technology Corporation

11F-2, No. 289, Sec. 2, Guang-Fu Rd., Hsin-Chu 300, Taiwan, R.O.C.

Tel: 886-3-5163339

Fax: 886-3-5163679

4. **short WIS_WriteUserData(unsigned char *rUserInfo, unsigned char Flag)**

Description

Create or update a specified user's information in the fingerprint module.

Parameter

rUserInfo: an input structure WIS_USER_INFO, which carries the user's information. See **WIS_ReadUserData** for details.

Flag: an input value

1 (OP_INSERT): Create new user's information

2 (OP_MODIFY): Update existed user's information

Return Value

0	Succeed
-1	Fail
-10	FLASH_READ_ERROR
-11	FLASH_WRITE_ERROR
-21	INVALID_FINGERNO
-23	INVALID_OPERATION
-32	USER_NOT_FOUND

Example

Input: UserInfo.id="TEST"
 UserInfo.role=0x00 /* Administrator=0, User=1 */
 UserInfo.password="0000"
 UserInfo.securityLevel=SECURITY_A
 UserInfo.fingerNo=0x01
 UserInfo.fingerId=0x05
 UserInfo.quality=QUALITY_A
 UserInfo.enrlTemplate=<Template>
 Flag=0x01 /*OP_INSERT=1 or OP_MODIFY=2*/

* Transmitted Data, totally 209/(209+160)/(209+320) bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0043
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x00C1
DATA BODY	Descriptor	Packet[6]	1	0x00B8
		Packet[7]	1	0x0001
		Packet[8..13]	6	0
	Content	Packet[14..205]	192	<rUserInfo>



Wisom Technology Corporation

11F-2, No. 289, Sec. 2, Guang-Fu Rd., Hsin-Chu 300, Taiwan, R.O.C.

Tel: 886-3-5163339

Fax: 886-3-5163679

		Packet[206]	1	0x01
CHECKSUM		Packet[207..208]	2	<Variable>

* <rUserInfo> : input structure WIS_USER_INFO

* <Variable> : This value varies depending on the value of rUserInfo.

* Received Data, totally 18 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0043
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000A
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7..13]	7	0
	Content	Packet[14..15]	2	0(or -1,-10, -11,-21,-23, -41)
CHECKSUM		Packet[16..17]	2	0x004F

5. **short WIS_GetUserList(short NthSeg, short SegNo, unsigned char* rIDList)**

Description

This function is to get the user ID list from the fingerprint module by giving segment id and number of user Ids to get in one segment.

Parameter

NthSeg: an input value, which specifies a segment id where the user Ids stored.

SegNo: an input value, which specifies the number of Ids to get in one segment.

The value must be less than **20**.

rIDList: returned ID list in string format.

Return Value

>0	Succeed. The number of SegNo or the actual number of Ids currently loaded.
-1	Fail
-10	FLASH_READ_ERROR
-22	INVALID_SEGMENT_NUMBER

Remarks

To get the full ID list, it needs to do a while or for loop with the **NthSeg** increments by 1 at each time. The **NthSeg** should start from 0 (the beginning of the database) and the loop should stop when the return value is less than the input **SegNo**, i.e. it has reached the end of the database.

The maximum number of loops = MaxUser/SegNo + (0/1).

For example, if MaxUser=500,

If SegNo=20, then

The maximum number of loops = 500/20 =25.

If SegNo=15, then

The maximum number of loops = 500/15 +1 =34.

The returned user id list is in the format of a series of binary data with each Id spaced by 8 (USERID_LEN). To retrieve the ID, see the sample code below:

```
char UserID[USERID_LEN + 1];
memcpy((unsigned char *)UserID, BinaryData + USERID_LEN*i, USERID_LEN);
UserID[USERID_LEN+1]=0;
```

This will retrieve the ith user id of the currently returned ID list.

Note: *The maximum number of SegNo is defined as 20 in fingerprint module.*

Example

Input: NthSeg=0 SegNo=20

Output: rIDList

* Transmitted Data, totally 20 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0044
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000C
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7]	1	0x0002
		Packet[8..13]	6	0
	Content	Packet[14..15]	2	0
		Packet[16..17]	2	0x0014
CHECKSUM		Packet[18..19]	2	0x0068

* Received Data, totally 130 bytes (7 user's ID)

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0044
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x0042
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7..8]	2	<0x0038>
		Packet[9..13]	5	0
	Content	Packet[14..15]	2	7(or -1,-10, -22) <No of users>
		Packet[16..127]	112	<rIDList>
CHECKSUM		Packet[128..129]	2	<Variable>

* < rIDList > : Sum of the list

* <Variable> : This value varies depending on the value of rUserInfo.

➤ **System Functions**

1. **short WIS_LoadDefaultSetting()**

Description

This function provides an interface for user to reload default system configuration to the fingerprint module. It will take effect immediately without shutting down the system.

Return Value

0	Succeed.
-1	Fail
-11	FLASH_WRITE_ERROR

Example

* Transmitted Data, totally 16 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0050
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x0008
DATA BODY	Descriptor	Packet[6..13]	8	0
	Content	---	---	---
CHECKSUM		Packet[14..15]	2	0x0058

* Received Data, totally 18 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0050
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000A
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7..13]	7	0
	Content	Packet[14..15]	2	0(-1 or -11)
CHECKSUM		Packet[16..17]	2	0x005C



Wison Technology Corporation

11F-2, No. 289, Sec. 2, Guang-Fu Rd., Hsin-Chu 300, Taiwan, R.O.C.

Tel: 886-3-5163339

Fax: 886-3-5163679

2. **short WIS_UpdateProgram(unsigned char* SegBuf, short SegSize, short IsEnd)**

Description

This function provides an interface for user to update user's program through the RS232.

Parameter

SegBuf: an input buffer, which contains the code to be updated per time.

SegSize: an input value, which represents the data size of the 'SegBuf' for this transmission, i.e. $0 \leq \text{'SegSize'} \leq 496$

IsEnd: an input value, which indicates if this packet is the last 0:YES, otherwise:NO

Return Value

0	Succeed.
-1	Fail
-11	FLASH_WRITE_ERROR

Remarks

This function provides an interface for user to update user program through the RS232. User must separate the hex file of 256KB into several packets of 'SegBuf' to transmit the code packet by packet. The maximum size of the 'SegBuf' must be less than 496 bytes for each packet. There will be more than one packet needed for sending between client and fingerprint module.

The function only updates the user's program, all the user's info and log will be remained unchanged. **Note: DO NOT terminate the process while the program updating is in progress.**

Example

Input: SegBuf SegSize IsEnd

* Transmitted Data, totally xx bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0051
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	<Data Length>
DATA BODY	Descriptor	Packet[6..7]	2	<Buf >
		Packet[8]	1	0x0002
		Packet[9]	1	0x0002
		Packet[10..13]	4	0
	Content	Packet[14..14+Buf-1]	<Buf>	<variable>
		Packet[X+1..X+2]	2	0x00FF
Packet[X+3..X+4]		2	0(otherwise)	
CHECKSUM		Packet[X+5..X+6]	2	<CheckSum>

<Data Length>: The actual length of the DATA BODY.

<Buf>: The value of SegSize.

<Variable>: The sum of the actual value of SegBuf.

<Variable>: The sum of the actual value depending on <Variable> & <Buf>.

* Received Data, totally 18 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0051
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000A
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7..13]	7	0
	Content	Packet[14..15]	2	0(-1 or -11)
CHECKSUM		Packet[16..17]	2	0x005D

3. **short WIS_Snap(short NthSeg, unsigned char ImgFlag, unsigned char* rImgBuf, short* rSize)**

Description

This function provides an interface for user to get the currently captured image buffer from the fingerprint module. The transmission time will depend on the baud rate of RS232.

Parameter

NthSeg: an input value, which specifies a sliced id of image data. This ID starts from 0 and should be incremented by 1 in the while or for loop at each time.

ImgFlag: the value is either LARGE(0) : an image of 256 x 320 while SMALL(1) : 128 x 160

rImgBuf: an image buffer for storing the image captured from the fingerprint reader. The allocated memory must be greater than 256*320 bytes.

rSize: the size of image data received at one time.

Return Value

0	Succeed.
-1	Fail
-10	FLASH_READ_ERROR
-41	DEVICE_CONNECT_FAIL
-42	CAPTURE_TIMEOUT

Remarks

Once the function is called, the fingerprint module captures image from the reader and stores to a frame buffer and send to the client. There are more than one packet needed for sending between client and fingerprint. The loop to receive the image buffer should be stopped when the sum of the rSize received is equal to or greater than 256x 320 for WM100 reader.



Wisom Technology Corporation

11F-2, No. 289, Sec. 2, Guang-Fu Rd., Hsin-Chu 300, Taiwan, R.O.C.

Tel: 886-3-5163339

Fax: 886-3-5163679

Example

Input: NthSeg=<n> ImgFlag=0 /*LARGE=0*/

Output: rImgBuf=<image data> rSize=0x01EE

* Transmitted Data, totally 19 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0052
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000B
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7]	1	0x0001
		Packet[8..13]	7	0
	Content	Packet[14..15]	2	0
		Packet[16]	1	0
CHECKSUM		Packet[17..18]	2	0x0060

* Received Data, totally 512 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0052
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x01F8
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7..8]	2	0x01EE
		Packet[9..13]	5	0
	Content	Packet[14..15]	2	0(-1,-10,-41 or -42)
		Packet[16..509]	2	<Variable>
CHECKSUM		Packet[510..511]	2	<CheckSum>

<Variable>: The sum of the actual value of Packet[16..509].

<CheckSum>: The sum of all the value depending on <Variable>.

4. **short WIS_GetImage(short NthSeg, unsigned char ImgFlag, unsigned char* rImgBuf, short* rSize)**

Description

This function provides an interface for user to get the previously captured image buffer from the fingerprint module. The transmission time will depend on the baud rate of RS232.

Parameter

NthSeg: an input value, which specifies a sliced id of image data. This ID starts from 0 and should be incremented by 1 in the while or for loop at each time.

ImgFlag: the value is either LARGE(0) : an image of 256 x 320 for WM100 while SMALL(1) : 128 x 160

rImgBuf: an image buffer for storing the image captured from the fingerprint reader. The allocated memory must be greater than 256*320 bytes.

rSize: the size of image data received at one time.

Item	LARGE	SMALL
WM100	256 x 320	128 x 160
WM101	192 x 224	96 x 112
WM102	256 x 288	128 x 144

Return Value

0	Succeed.
-1	Fail
-10	FLASH_READ_ERROR
-41	DEVICE_CONNECT_FAIL
-42	CAPTURE_TIMEOUT

Remarks

Once the function is called, the fingerprint module sends the image in the frame buffer to the client. There are more than one packet needed for sending between client and fingerprint. The loop to receive the image buffer should be stopped when the sum of the rSize received is equal to or greater than 256x 320 for WM100 reader.



Wisom Technology Corporation

11F-2, No. 289, Sec. 2, Guang-Fu Rd., Hsin-Chu 300, Taiwan, R.O.C.

Tel: 886-3-5163339

Fax: 886-3-5163679

Example

Input: NthSeg=<n> ImgFlag=0 /*LARGE=0*/

Output: rImgBuf=<image data> rSize=0x01EE

* Transmitted Data, totally 19 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0053
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000B
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7]	1	0x0001
		Packet[8..13]	7	0
	Content	Packet[14..15]	2	0
		Packet[16]	1	0
CHECKSUM		Packet[17..18]	2	0x0061

* Received Data, totally 512 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0053
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x01F8
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7..8]	2	0x01EE
		Packet[9..13]	5	0
	Content	Packet[14..15]	2	0(-1,-10,-41 or -42)
		Packet[16..509]	2	<Variable>
CHECKSUM		Packet[574..575]	2	<CheckSum>

<Variable>: The sum of the actual value of Packet[16..573].

<CheckSum>: The sum of all the value depending on <Variable>.

5. **short WIS_SetDeviceId(unsigned char *deviceId)**

Description

This function provides an interface for writing a device id to this fingerprint module. This ID is mainly used to identify each device in the RS485 network use.

Parameter

deviceId: a 3-byte string.

Return Value

0	Succeed.
-10	FLASH_READ_ERROR
-11	FLASH_WRITE_ERROR

Remarks

This device id must be contained in the protocols so as to communicate with the right device in the RS485 network. This ID will be compared with the ID previously stored in the system and the protocols will only be executed only when the ID is matched. The default value of the Device ID is set to 0. *If you any problem in your first time setting, please contact with you system provider.*

Example

Input: deviceId="000"

* Transmitted Data, totally 19 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0055
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000B
DATA BODY	Descriptor	Packet[6]	1	0x0003
		Packet[7..13]	7	0
	Content	Packet[14..16]	3	0x30 0x30 0x30
CHECKSUM		Packet[17..18]	2	0x00F3

* Received Data, totally 18 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0055
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000A
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7..13]	7	0
	Content	Packet[14..15]	2	0(-10 or -11)
CHECKSUM		Packet[16..17]	2	0x0061

6. **short WIS_GetDeviceId(unsigned char *rDeviceId)**

Description

This function provides an interface for reading a device id to this fingerprint module. This ID is mainly used to identify each device in the RS485 network use.

Parameter

deviceId: a 3-byte string.

Return Value

0	Succeed.
-10	FLASH_READ_ERROR

Remarks

See details in function *WIS_SetDeviceId()*.

Example

Output: rDeviceId="000"

* Transmitted Data, totally 16 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0054
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x0008
DATA BODY	Descriptor	Packet[6..13]	8	0
	Content	---	0	---
CHECKSUM		Packet[17..18]	2	0x005C

* Received Data, totally 21 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0054
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000D
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7]	1	0x0003
		Packet[8..13]	6	0
	Content	Packet[14..15]	2	0(-10)
		Packet[16..18]	3	0x30 0x30 0x30
CHECKSUM		Packet[19..20]	2	0x00F6

7. **short WIS_GetConfig(unsigned char *rWis_config)**

Description

This function provides an interface for getting the system configuration of fingerprint module.

Parameter

rWis_config: an output structure of WIS_CONFIG format , which carries the system information.

```
typedef struct Wis_Config
```

```
{
```

```
    unsigned char version[7];    /*Version of the firmware, Read Only*/
```

```
    unsigned char contrast;
```

```
    unsigned char baudrate;    /*for RS232  0: 9600  1: 19200 */
```

```
                                /* 2 : 38400  3 : 57600  4 : 115200 */
```

```
    unsigned char deviceId[3];
```

```
    unsigned char autoAdjustImage; // 1: Adjust image automatically
```

```
    unsigned char SensorType; //reserve
```

```
    unsigned char TemplateType; //2 : 480 Bytes  3: 160 Bytes  4: 320 Bytes
```

```
    unsigned char reserve;
```

```
}WIS_CONFIG;
```

Return Value

0	Succeed.
-1	FAIL
-10	FLASH_READ_ERROR



Wisom Technology Corporation

11F-2, No. 289, Sec. 2, Guang-Fu Rd., Hsin-Chu 300, Taiwan, R.O.C.

Tel: 886-3-5163339

Fax: 886-3-5163679

Example

Input: None

Output: rWis_config

* Transmitted Data, totally 16 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0056
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x0008
DATA BODY	Descriptor	Packet[6..13]	8	0
	Content	---	0	---
CHECKSUM		Packet[17..18]	2	0x005E

* Received Data, totally 34 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0056
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x001A
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7]	1	0x000C
		Packet[8..13]	6	0
	Content	Packet[14..15]	2	0(-1 or -10)
		Packet[16..31]	16	<rConfigInfo>
CHECKSUM		Packet[32..33]	2	<Checksum >

<rConfigInfo>: The sum of the value of Packet[16..31] in WIS_CONFIG format.

<Checksum>: The sum of all the value depending on <rConfigInfo>.



Wislon Technology Corporation

11F-2, No. 289, Sec. 2, Guang-Fu Rd., Hsin-Chu 300, Taiwan, R.O.C.

Tel: 886-3-5163339

Fax: 886-3-5163679

8. **short WIS_SetConfig(unsigned char *WisConfig)**

Description

This function provides an interface for setting the system configuration of fingerprint module. The system needs to be reset to use the configuration.

Parameter

WisConfig: an input structure of WIS_CONFIG format, which carries the system information. See **WIS_GetConfig()** for details.

Return Value

0	Succeed.
-1	FAIL
-10	FLASH_READ_ERROR
-11	FLASH_WRITE_ERROR

Remarks

The user can set all the parameters in the WIS_CONFIG structure except the attribute '**version**'. 'version' can be read only through the **WIS_GetConfig()**.



Wison Technology Corporation

11F-2, No. 289, Sec. 2, Guang-Fu Rd., Hsin-Chu 300, Taiwan, R.O.C.

Tel: 886-3-5163339

Fax: 886-3-5163679

Example

Input: Config.version="WIS305"

Config.baudrate=0x00

Config.contrast=10

Config.deviceId="000"

Config.autoAdjustImage=0x01

Config.TemplateType=2 //480 Bytes

Output: None

* Transmitted Data, totally 32 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0057
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x0018
DATA BODY	Descriptor	Packet[6]	1	0x000C
		Packet[7..13]	7	0
	Content	Packet[14..29]	16	<rConfigInfo>
CHECKSUM		Packet[30..31]	2	<Checksum >

<rConfigInfo>:

[14]=0x57 [15]=0x49 [16]=0x53 [17]=0x33 [18]=0x30 [19]=0x35 [20]=0x00
 [21]=0x00 [22]=0x0A [23]=0x30 [24]=0x30 [25]=0x30 [26]=0x01
 [27]=0x00 [28]=0x02 [29]=0x00

* Received Data, totally 18 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0057
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000A
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7..13]	7	0
	Content	Packet[14..15]	2	0(-1, -10 or -11)
CHECKSUM		Packet[16..17]	2	0x0063

9. **short WIS_GetGPIO(unsigned long *rStatus)**

Description

Note: only support in WM10x series

This function provides an interface for reading the status of 3 input-GPIOs (**GPIO1, GPIO2, GPIO7 -- please refer to Products.pdf**) of the fingerprint module.

Parameter

rStatus: the values will range from 0 to 7

Return Value

0: Succeed

Remarks

It supports 3 input-GPIOs with one GPIO pin represents one bit value.

GPIO7	GPIO2	GPIO1	Value (rStatus)	Value Get shown in Demo program
0	0	0	0	8
0	0	1	1	1
0	1	0	2	2
0	1	1	3	3
1	0	0	4	4
1	0	1	5	5
1	1	0	6	6
1	1	1	7	7

Example

Input: None

Output: **rStatus=0**

* Transmitted Data, totally 16 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0058
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x0008
DATA BODY	Descriptor	Packet[6..13]	8	0
	Content	---	0	---
CHECKSUM		Packet[17..18]	2	0x0060

* Received Data, totally 22 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0058
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000E
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7]	1	0x0004
		Packet[8..13]	6	0
	Content	Packet[14..15]	2	0 → 0:true → -1:false
		Packet[16..19]	4	0 (rStatus)
CHECKSUM		Packet[20..21]	2	0x006C

10. **short WIS_SetGPIO(unsigned long status)**

Description

Note: only support in WM10x series

This function provides an interface for setting the status of 3 output-GPIOs (**GPIO4, GPIO5, GPIO6 -- please refer to Products.pdf**) of the fingerprint module for external system use.

Parameter

status: the values range from 0 to 7

Return Value

0: Succeed

Remarks

It supports 3 output-GPIOs with one GPIO pin represents one bit value.

GPIO6	GPIO5	GPIO4	Value (status)	Value Set shown in Demo program
0	0	0	0	8
0	0	1	1	1
0	1	0	2	2
0	1	1	3	3
1	0	0	4	4
1	0	1	5	5
1	1	0	6	6
1	1	1	7	7



Wisom Technology Corporation

11F-2, No. 289, Sec. 2, Guang-Fu Rd., Hsin-Chu 300, Taiwan, R.O.C.

Tel: 886-3-5163339

Fax: 886-3-5163679

Example

Input: **status=0x00000005** /*0 ~ 7*/

Output: None

* Transmitted Data, totally 20 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0059
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000C
DATA BODY	Descriptor	Packet[6]	1	0x0004
		Packet[7..13]	7	0
	Content	Packet[14..17]	4	0x00000005 (status)
CHECKSUM		Packet[18..19]	2	0x006E

* Received Data, totally 18 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x0059
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000A
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7..13]	7	0
	Content			0
		Packet[14..15]	2	→ 0:true → -1:false
CHECKSUM		Packet[16..17]	2	0x0065

11. **short WIS_SetParameter(unsigned char Contrast)**

Description

This function provides an interface for dynamically adjusting the image of WM100 reader by setting the value of contrast.

Parameter

Contrast: the valid values are ranged from 0 to 31. Default=10

Return Value

0	Succeed.
-1	FAIL
-10	FLASH_READ_ERROR
-11	FLASH_WRITE_ERROR

Remarks

This setting will take effect after 60 ms of calling this function.

Example

Input: Contrast=10

Output:None

* Transmitted Data, totally 17 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x005A
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x0009
DATA BODY	Descriptor	Packet[6]	1	0x0001
		Packet[7..13]	7	0
	Content	Packet[14]	1	0x000A
CHECKSUM		Packet[15..16]	2	0x006E

* Received Data, totally 18 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x005A
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000A
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7..13]	7	0
	Content	Packet[14..15]	2	0(-1,-10,-11)
CHECKSUM		Packet[16..17]	2	0x0066

12. **short WIS_GetImageSize(unsigned char ImgFlag, short* Width, short *Height)**

Description

This function provides an interface for user to get the size of the currently captured image.

Parameter

ImgFlag: the value is either LARGE(0) : an image of 256 x 320 for WM100 while SMALL(1) : 128 x 160

Width, Height: the size of the image.

Item	LARGE	SMALL
WM100	256 x 320	128 x 160
WM101	192 x 224	96 x 112
WM102	256 x 288	128 x 144

Return Value

0	Succeed.
-1	Fail
-10	FLASH_READ_ERROR
-41	DEVICE_CONNECT_FAIL

Remarks

This function sends back the size of currently captured image for the programmer to further allocate the buffer needed or for displaying the right ratio of the image.

Example

Input: ImgFlag=0 /*LARGE=0*/

Output: Width, Height

* Transmitted Data, totally 17 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x005C
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x0009
DATA BODY	Descriptor	Packet[6]	1	0x0001
		Packet[7..13]	7	0
	Content	Packet[14.]	1	0
CHECKSUM		Packet[15..16]	2	0x0066

* Received Data, totally 22 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x005C
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000E
DATA BODY	Descriptor	Packet[6..7]	2	0x0002
		Packet[8..9]	2	0x0002
		Packet[10..13]	5	0
	Content	Packet[14..15]	2	0(-1,-10,-41)
		Packet[16..17]	2	<Width>
		Packet[18..19]	2	<Height>
CHECKSUM		Packet[20..21]	2	<CheckSum>

<Width>, <Height>: The value of Width and Height returned.

<CheckSum>: The sum of all the value depending on <Variable>.

13. **short WIS_SetPowerMode(unsigned char mode)**

Description

To set the DSP to the normal or sleep mode.

Parameter

mode: 1: Normal mode 3: Sleep Mode

Return Value

0	Succeed.
-1	Fail

Remarks

This function sets the DSP to the normal or sleep mode. The sleep mode will be wake up automatically with any subsequent protocols.

Example

Input: mode=3(Sleep Mode) Output:None

* Transmitted Data, totally 17 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x006F
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x0009
DATA BODY	Descriptor	Packet[6]	1	0x0001
		Packet[7..13]	7	0
	Content	Packet[14]	1	0x0003
CHECKSUM		Packet[15..16]	2	0x007C

* Received Data, totally 18 bytes

Packet		Data Sequence	Byte	Value
HEADER	OPCode	Packet[0]	1	0x006F
	Device Id	Packet[1..3]	3	0x0000
	Data Length	Packet[4..5]	2	0x000A
DATA BODY	Descriptor	Packet[6]	1	0x0002
		Packet[7..13]	7	0
	Content	Packet[14..15]	2	0(-1,-10,-11)
CHECKSUM		Packet[16..17]	2	0x007B