

# UniFinger Evaluation Kit

**User's guide**

Ver. 4.1



## Revision History

Rev No.	Issued date	Description
1.0		Initial Release
2.0		Document contents revised
3.0	Nov. 15. 2004	Covers SFM3000 and SFM3500 EVK
3.0.1	Jul. 21, 2005	In SFM3500 EVK, the pin outs of relay and buzzer are corrected
4.0	Oct. 4, 2005	The description of PC User Interface Program is updated based on UniFinger UI V4.0. Module SDK is replaced by SFM SDK.
4.1	Dec.2, 2005	The description of PC User Interface Program is updated based on UniFinger UI V4.1.

### Important Notice

Information in this document is provided in connection with Suprema products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Suprema's Terms and Conditions of Sale for such products, Suprema assumes no liability whatsoever, and Suprema disclaims any express or implied warranty, relating to sale and/or use of Suprema products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Suprema products are not intended for use in medical, life saving, life sustaining applications, or other applications in which the failure of the Suprema product could create a situation where personal injury or death may occur. Should Buyer purchase or use Suprema products for any such unintended or unauthorized application, Buyer shall indemnify and hold Suprema and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Suprema was negligent regarding the design or manufacture of the part.

Suprema reserves the right to make changes to specifications and product descriptions at any time without notice to improve reliability, function, or design.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Suprema reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

Contact your local Suprema sales office or your distributor to obtain the latest specifications and before placing your product order.

Copyright © by Suprema Inc., 2005

\*Third-party brands and names are the property of their respective owners.

## Contents

1. Overview .....	6
2. Evaluation Kit Contents .....	7
2.1. UniFinger module .....	7
2.1.1. Module board .....	8
2.1.2. Fingerprint sensor (board) .....	8
2.2. Interface board .....	9
2.2.1. Type A .....	9
2.2.2. Type B .....	10
2.3. Serial cable .....	12
2.4. Power adaptor .....	12
2.5. Software CD .....	12
3. Quick Start : SFM1000/2000/3000 EVK .....	14
3.1. Hardware installation .....	14
3.2. Standalone evaluation .....	15
3.2.1. Enroll .....	16
3.2.2. Identify .....	16
3.2.3. Delete .....	16
3.2.4. LED indication conventions .....	16
3.2.5. Buzzer indication .....	17
4. Quick Start : SFM3500 EVK .....	18
4.1. Hardware installation .....	18
4.2. Standalone evaluation .....	19
4.2.1. Enroll .....	20
4.2.2. Identify .....	21
4.2.3. Delete .....	21
4.2.4. Verification by Wiegand input .....	21

---

4.2.5. LED indication conventions .....	21
4.2.6. Buzzer indication.....	21
4.3. Evaluation with external devices.....	22
4.3.1. Terminal pinout assignment .....	22
4.3.2. RS422/485 Interface .....	23
4.3.3. Wiegand Interface .....	24
4.3.4. Using relay signal .....	24
4.3.5. General IO interface .....	25
5. PC User Interface Program .....	26
5.1. Getting started .....	26
5.2. Main window .....	27
5.2.1. Main commands.....	29
5.2.2. Advanced enroll commands .....	31
5.2.3. Template commands.....	32
5.2.4. Admin level commands .....	33
5.2.5. Misc commands .....	33
5.2.6. Smart card commands .....	34
5.3. System parameters window.....	35
5.4. IO window for SFM3000.....	36
5.4.1. Input port .....	37
5.4.2. Output port .....	38
5.4.3. Wiegand .....	39
5.5. IO window for SFM3500.....	39
5.5.1. Input port .....	40
5.5.2. Output/LED port.....	40
5.6. Wiegand window.....	41
5.6.1. Legacy format .....	42
5.6.2. Input/output port.....	42
5.6.3. Signal.....	43

---

5.6.4. Advanced options .....	43
5.6.5. 26 bit standard format .....	43
5.6.6. Pass Through format.....	43
5.6.7. Custom format.....	44
5.6.8. Command card .....	46
5.7. Log window .....	47
5.8. User memory window .....	47
5.9. Upgrade window .....	47
5.9.1. Firmware upgrade .....	48
5.9.2. Save system configuration.....	48
5.9.3. Load system configuration .....	49
6. Command Line Tool .....	50
7. SFM SDK.....	51

## 1. Overview

This document is the user's guide of UniFinger evaluation kit, which is a demonstration system to enable users to evaluate the core functionality of UniFinger standalone module. Suprema provides high-performance standalone fingerprint modules ideal for embedded system application where biometric security is needed.

The evaluation kit helps users to understand the operations of the module easily and provides quick solution to make use of it in developing application systems.

The evaluation kit is composed of interface board, standalone module with fingerprint sensor, and software tools. The interface board provides built-in user interface, such as buttons, LEDs, and buzzer, to support quick evaluation of standalone module's core functionalities, including enrollment, identification, and deleting of user. Also, full functionalities of the standalone module can be easily tested due to the user-friendly interface ports of the interface board and supporting PC user interface tools.

The evaluation kit is available for all the UniFinger standalone modules, including SFM1000, SFM2000, SFM3000, and SFM3500 series. Since SFM1000, SFM2000, and SFM3000 series are compatible, the evaluation kit for these models are also compatible. However, SFM3500 series is high-end standalone module, providing powerful interfaces and additional functionalities, so that its interface board is different from those for the other models.

For further information, please refer to the following related documents:

- *UniFinger SFM1000/SFM2000/SFM3000/SFM3500 Datasheet*
- *UniFinger Packet Protocol Manual*
- *UniFinger Evaluation Kit datasheet*
- *SFM SDK Reference Manual*

## 2. Evaluation Kit Contents

The contents of the evaluation kit package are as follows:

Item	Description
UniFinger module	SFM1000, SFM2000, SFM3000, and SFM3500 series with fingerprint sensors
Interface board	Mother board on which the module is mounted providing diverse interfaces Two types ( A, B ) differentiated by supporting modules
Serial cable	RS232C interface cable to connect the interface board with a standard 9 pin serial port
Power adaptor	DC 5V power adaptor to supply regulated power to the interface board
Software CD	CD containing technical documents and application software tools

### 2.1. UniFinger module

In the evaluation kit, one SFM standalone module with fingerprint sensor is included. Due to the compatibility of SFM1000, SFM2000, and SFM3000 series, users can evaluate various models by replacing only the UniFinger module. The kit for SFM3500 includes a different interface board due to diverse extended functionalities of SFM3500 series.

Module series	Available model	Interface board
SFM1000 series	SFM1000-FL, SFM1000-PR, SFM1000-FT , SFM1000-FC	Type A
SFM2000 series	SFM2000-FL, SFM2000-PR, SFM2020-OP, SFM2000-TC, SFM2000-FT, SFM2000-FC,	Type A
SFM3000 series	SFM3000-FL, SFM3000-FS,	Type A

	SFM3000-PR, SFM3000-TC, SFM3010-FC, SFM3020-OP, SFM3040-OL, SFM3040-OC	
SFM3500 series	SFM3500-FL, SFM3500-FS, SFM3500-PR, SFM3510-FC, SFM3520-OP, SFM3550-TC	Type B

### 2.1.1. Module board



SFM1000



SFM2000



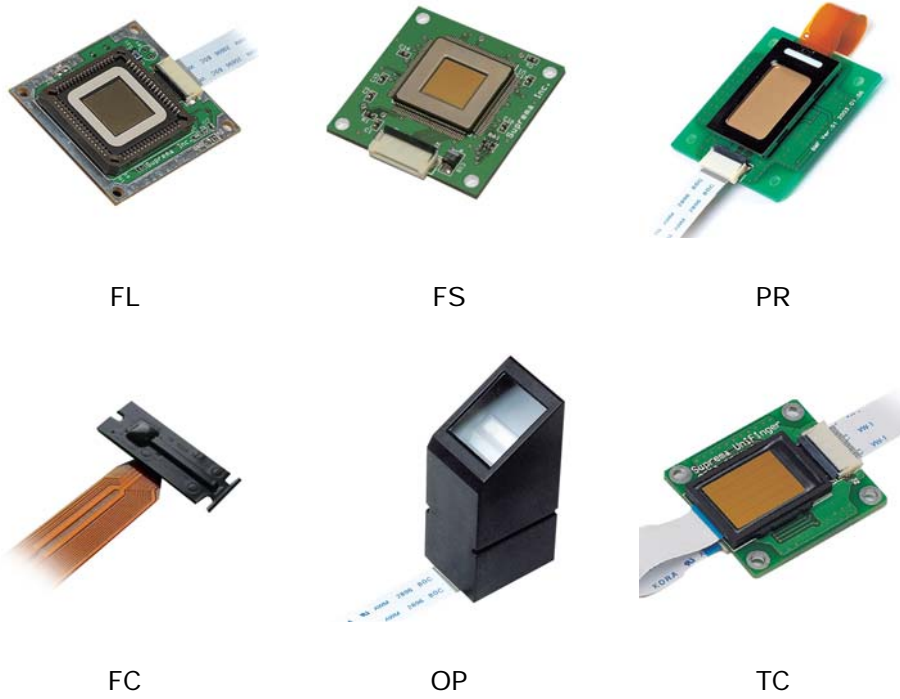
SFM3000



SFM3500

### 2.1.2. Fingerprint sensor (board)





## 2.2. Interface board

### 2.2.1. Type A



<b>Component</b>	<b>Description</b>
JP1	Module interface pin header socket on which the module is mounted
JP2	Module serial interface connector
JP3	Reserved for internal use
JP4	Pin header socket on which the module is mounted. Reserved for internal use.
JP5	Duplicate of JP1 for user's test
SW1, SW2, SW3	Input buttons for enroll/identify/delete all functions, connected to GPIO0 – GPIO2, respectively.
SW4	Power switch
D1, D2, D3	Status output LEDs signaling ready/success/fail, connected to GPIO0 – GPIO2, respectively
D4	Power status LED
LS1	Buzzer connected to GPIO3
J1	Power adaptor jack
J2	Reserved for internal use
P1	RS232C DB9 female connector for PC interface
P2	Reserved for internal use

### 2.2.2. Type B



Component	Description
J1, J2, J3, J4, J5, J6	Connectors correspondent to SFM3500 interface connectors J1-J6, respectively
P1	RS232C DB9 female connector for host serial port
P2	RS232C DB9 female connector for aux serial port
P3	Power adaptor jack
SW1, SW2, SW3	Input buttons for enroll/identify/delete functions, connected to IN0-IN2 port, respectively.
SW4	RS422/485 duplex mode selection switch
SW5	Power switch

D1, D2, D3	Status output LEDs connected to LED0 – LED2 port, respectively
D4	Power status LED
LS1	Buzzer connected to OUT1 port
K1, K2	Relay controlled by OUT0 port
BT1	A CR2025 type lithium cell for time keeping.
CON1, CON2	Terminals for user's interface test
U5	Fixture holes for sensor boards

2.3. Serial cable



2.4. Power adaptor



2.5. Software CD

Item	Description
------	-------------

Manuals	Datasheets of UniFinger modules Datasheet of UniFinger Evaluation Kit Firmware release notes Protocol reference manual Evaluation Kit User guide SFM SDK reference manual Application notes
Brochures	Brochures of UniFinger modules
Utilities	PC UI program Command line tool
SFM SDK	SDK for developing Win32 applications

### 3. Quick Start : SFM1000/2000/3000 EVK

#### 3.1. Hardware installation

The evaluation Kit for SFM1000, SFM2000, and SFM3000 series includes type A interface board. Hardware installation procedure is as follows:

- 1) Mount the standalone module onto the evaluation board by connecting the pin header sockets in the side of the standalone module to the pin headers in the evaluation board.\*
- 2) Connect the fingerprint sensor or sensor board with the module via the sensor cable.
- 3) For PC interface, connect the serial port of PC with the evaluation kit's serial port (P1) using the RS-232 serial cable.
- 4) Plug in the provided DC power adaptor at power adaptor jack and turn on the power switch (SW4) to start the operation.

Alternatively, you can use an 8 pin cable (this is not included in the EVK) to connect the module and interface board with JP2( 8pin Molex connector). In this connection, built-in user interfaces, including buttons, LEDs, and buzzer do not work, and only the evaluation by serial communication is possible.

The following figure shows an example of properly assembled evaluation kit(type A).

---

\* The numbers of pin headers of the evaluation board and the sockets of the standalone module may not match. In this case, align bottom lines of the module coincident with the bottom line of the evaluation board.



### 3.2. Standalone evaluation

The core functionality of evaluation kit can be tested using built-in user interfaces on the interface board. There exist three buttons, three LED's and buzzer on the interface board for standalone evaluation without PC interface.

On the interface board, 3 buttons, 3 LEDs, and buzzer are connected to the GPIO ports of the module. By factory default of evaluation kit, GPIO0 ~ GPIO3 of the module are programmed to support the evaluation kit. User can program the function of GPIO's using gpio configuration protocols and the default configuration for evaluation kit is recovered by simple command of GD or 'Default All' button on UniFinger PC UI's gpio configuration menu. For more information please refer to *Appendix A of UniFinger Packet Protocol Manual*.

The three buttons initiates the core functions of enrollment, identification, and template deletion, respectively. The LED's and buzzer indicate the processing status and result.

### 3.2.1. Enroll

By pressing 'Enroll' button, fingerprint enrollment is initiated.

The enroll mode system parameter of the module determines the required number of user's finger touch. On factory default enroll mode, which is 2 times (SFM1000, SFM2000) or 2 templates mode (SFM3000), user should touch finger twice for enrollment. The ready LED indicates that the module is waiting finger touch and it turns off after fingerprint scan is completed with a short beep sound.

After processing of the captured fingerprint image, processing result is indicated by beep and LED. If the processing is successful, a long beep sounds with Pass LED. Otherwise, 2 or 3 successive beeps are heard with Fail LED. For the second scan, this process is repeated. Do not press the 'Enroll' button again between the first and second scans.

Finally, when the enrollment process is successfully completed, enrolled fingerprint is saved to the internal flash memory of the standalone module with an automatically determined ID.

### 3.2.2. Identify

'Identify' button initiates identification process that searches matched fingerprint among the enrolled fingerprints. If matched one is found, the evaluation kit signals Pass by a long beep with Pass LED. Otherwise, it signals Fail by a 2 or 3 successive beeps with Fail LED.

### 3.2.3. Delete

The 'Delete' button deletes all the enrolled fingerprints from the Flash memory of the module immediately. It should be pressed twice to avoid deletion by mistake.

### 3.2.4. LED indication conventions

LED	Indication



---

Ready	Waiting finger input
Pass	Operation success
Fail	Operation fail

## 3.2.5. Buzzer indication

Sound	Indication
1 beep	Finger detected (short) Operation success (long)
2 beeps	Insufficient input of fingerprint
3 beeps	Operation fail / Timeout

## 4. Quick Start : SFM3500 EVK

### 4.1. Hardware installation

The evaluation Kit for SFM3500 series includes type B interface board.

Hardware installation procedure is as follows:

1) SFM3500 series EVK is provided as module with fingerprint sensor is assembled on the interface board. If the module or sensor is re-assembled, following step is required.

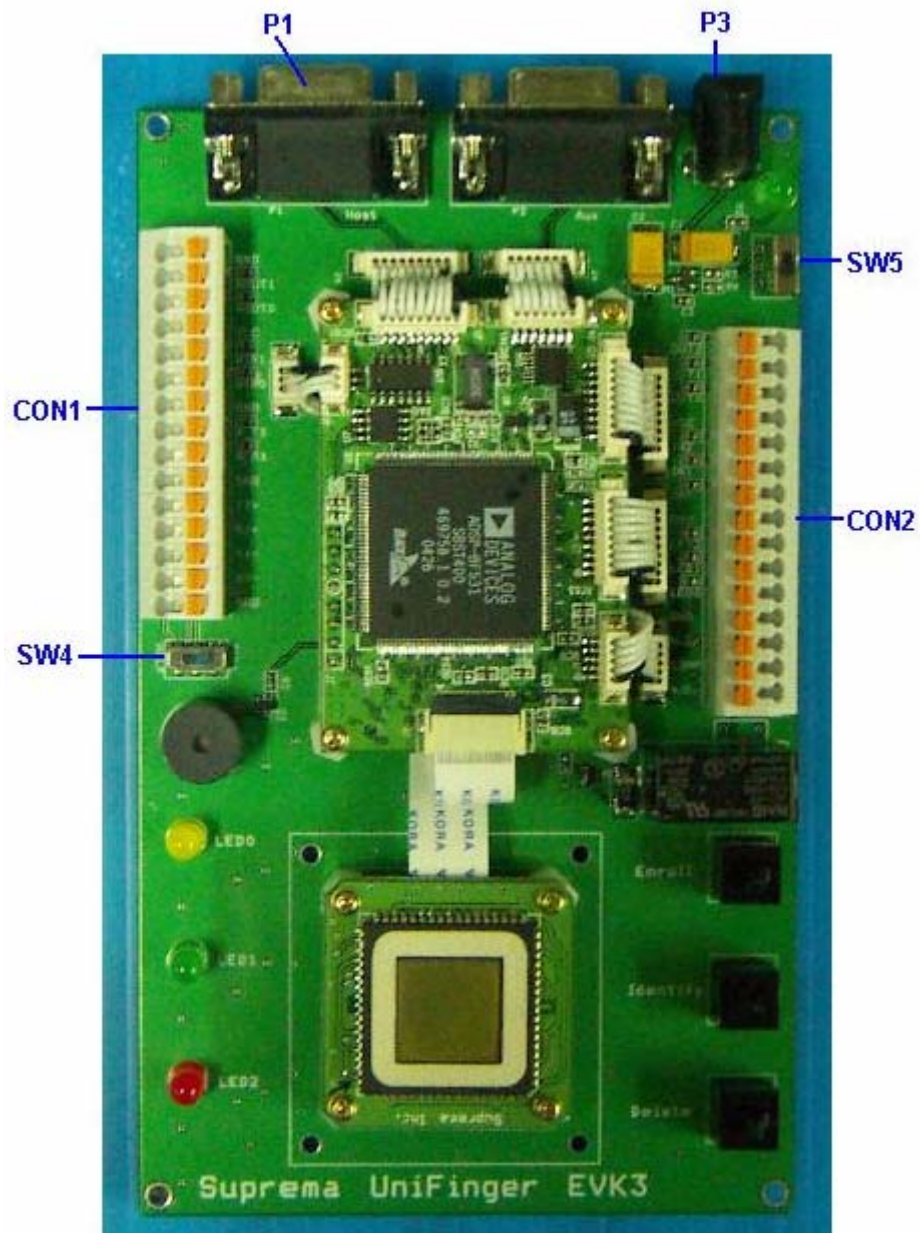
a. Fix the module onto the interface board using screw bolt. Then, connect the interface ports of J1-J6 on the module with the corresponding ports on the interface board using interface cables.

b. Fix the sensor board on the interface board using screw bolt. Then, connect the fingerprint sensor with the module via the sensor cable.

2) For PC interface, connect the serial port of PC with the evaluation kit's serial port (P1) using the RS-232 serial cable.

3) Plug in the provided DC power adaptor at power adaptor jack(P3) and turn on the power switch (SW5) to start the operation.

The following figure shows an example of properly assembled evaluation kit for SFM3500 series.



#### 4.2. Standalone evaluation

The core functionality of evaluation kit can be tested using built-in user interfaces on the interface board. There exist three buttons, three LED's, buzzer, and relay on the interface board for standalone evaluation without PC interface. Also, 2 terminal connectors are present to help evaluation of interfacing with various external devices, such as Wiegand device or relay-controlled device.

On the interface board, 3 buttons are connected to IN0-IN2 ports, 3 LEDs are connected to LED0-LED2 ports, buzzer is connected to OUT1, and relay is connected to OUT0, respectively. By factory default of evaluation kit, each IO is programmed to support the evaluation kit. User can also program the functions of IO's using supported configuration protocols. Also, user can configure the IO's as default configuration for evaluation kit using the command line tool batch program **SetEVKIo.bat** contained in the EVK CD.

The three buttons initiates the core functions of enrollment, identification, and template deletion, respectively. The LED's and buzzer indicate the processing status and result. Also, on identification success, relay is enabled and ID is sent through Wiegand output. If external Wiegand device is connected to Wiegand input ports, Wiegand input can be used to initiate verification (1:1) process.

#### 4.2.1. Enroll

By pressing 'Enroll' button, fingerprint enrollment is initiated.

The enroll mode system parameter of the module determines the required number of user's finger touch. On factory default enroll mode, which is 2 templates mode, user should touch finger twice for enrollment. LED0 indicates that the module is waiting finger touch by blinking and it stops blinking after fingerprint scan is completed with a short beep sound. While the module is in processing, LED0 stays in ON state.

After processing of the captured fingerprint image, LED0 turns off and processing result is indicated by beep and LED. If the processing is successful, a long beep sounds with LED1. Otherwise, 2 or 3 successive beeps are heard with LED2. For the second scan, this process is repeated. Do not press the 'Enroll' button again between the first and second scans.

Finally, when the enrollment process is successfully completed, enrolled fingerprint is saved to the internal flash memory of the standalone module with an automatically determined ID.

## 4.2.2. Identify

'Identify' button initiates identification process that searches matched fingerprint among the enrolled fingerprints. If matched one is found, the evaluation kit signals Pass by a long beep with LED1. Also, relay is enabled to be used for relay control and matched ID is transmitted via Wiegand output port. Otherwise, it signals Fail by a 2 or 3 successive beeps with LED2.

## 4.2.3. Delete

The 'Delete' button deletes all the enrolled fingerprints from the Flash memory of the module immediately. It should be pressed during more than 1 second to initiated the delete function.

## 4.2.4. Verification by Wiegand input

On default IO configuration for evaluation kit, Wiegand input is configured as 'Verify Given ID'. Thus, when Wiegand input is applied from external Wiegand device, verification is initiated. After verification process, output is yielded in the same pattern as identification process.

## 4.2.5. LED indication conventions

LED	Indication
LED0	Waiting finger input ( blinking ) In processing ( stably on )
LED1	Operation success Waiting Wiegand input ( blinking )
LED2	Operation fail Waiting Wiegand input ( blinking )

## 4.2.6. Buzzer indication

Sound	Indication
1 beep	Finger detected (short)

	Wiegand input detected (2 short) Operation success (long)
2 beeps	Insufficient input of fingerprint
3 beeps	Operation fail / Timeout

### 4.3. Evaluation with external devices

The interface board contains 2 terminal connectors for easy evaluation of integration with external devices. User can test various functionalities supported by SFM3500 series by simply fixing wire to the terminals.

#### 4.3.1. Terminal pinout assignment

##### 1) CON1

Pin number	Name	Description
1	GND	Power ground
2	WOUT1	Wiegand out, DATA1
3	WOUT0	Wiegand out, DATA0
4	VCC	5V DC Power
5	WIN1	Wiegand in, DATA1
6	WIN0	Wiegand in, DATA0
7	GND	Power ground
8	A_TX	Aux CMOS serial Transmit data
9	A_RX	Aux CMOS serial Receive data
10	GND	Power ground
11	TX+	Host RS422 TX+ data ( full duplex ) Host RS485 RTX+ data ( half duplex )
12	RX+	Host RS422 RX+ data ( full duplex )
13	TX-	Host RS422 TX- data ( full duplex ) Host RS485 RTX- data ( half duplex )
14	RX-	Host RS422 RX- data ( full duplex )

15	GND	Power ground
----	-----	--------------

## 2) CON2

Pin number	Name	Description
1	N.O.	Relay, normally open
2	COMMON	Relay, common
3	N.C.	Relay, normally closed with 'COMMON' pin
4	GND	Power ground
5	LED0	LED0 output
6	LED1	LED1 output
7	LED2	LED2 output
8	GND	Power ground
9	IN0	IN0 input
10	IN1	IN1 input
11	IN2	IN2 input
12	VCC	5V DC power
13	OUT0	OUT0 output
14	OUT1	OUT1 output
15	OUT2	OUT2 output

## 4.3.2. RS422/485 Interface

SFM3500 series support built-in RS422/485 serial network interfaces through host serial port. **Notice that RS232C port should not connected while using RS422/485 ports.**

## 1) RS422 (full duplex) interface

When RS422(full duplex) communication is used, 4 signals of TX+, TX-, RX+, and RX- on CON1 should be connected. Also, duplex mode selection switch(SW4) should be set at 'FULL' position and network mode system parameter of the module should be configured as 'full duplex' mode.

## 2) RS485 (half duplex) interface

When RS485(half duplex) communication is used, 2 signals of RTX+(TX+) and RTX-(TX-) on CON1 should be connected. Also, duplex mode selection switch(SW4) should be set at 'HALF' position and network mode system parameter of the module should be configured as 'half duplex' mode.

#### 4.3.3. Wiegand Interface

On evaluation kit, Wiegand input and output devices can be easily integrated with the module

##### 1) Wiegand input

WIN0, WIN1, and GND pins on CON1 should be connected with the external Wiegand device generating Wiegand output. If the external device requires 5V power, 5V power signal (pin 4) can be used. Besides verification using input Wiegand ID supported by default configuration, various functions can be implemented using Wiegand input. For detailed information, refer to the UniFinger Packet Protocol manual.

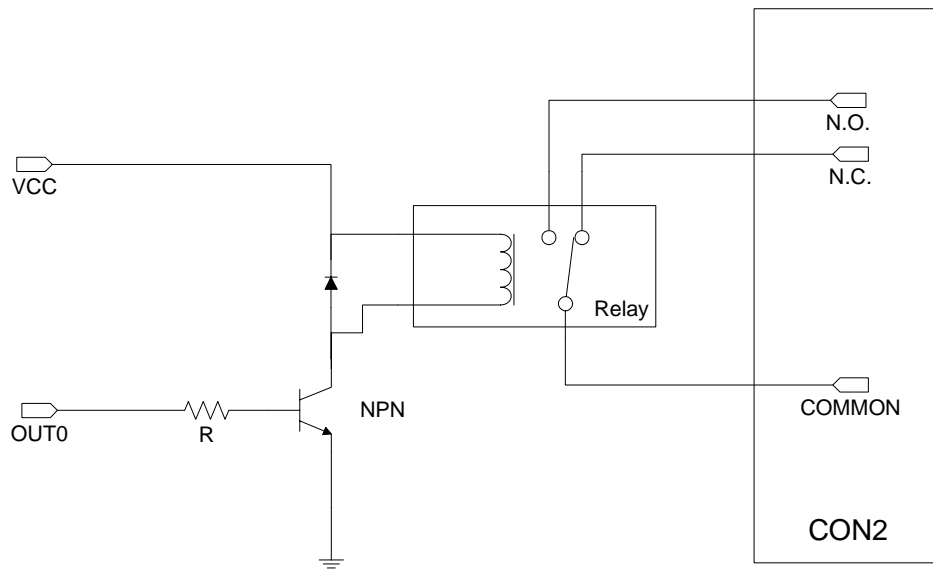
##### 2) Wiegand output

On successful match, used ID can be transmitted via Wiegand output port. WOUT0, WOUT1, and GND signals on CON1 should be connected with the Wiegand input port of external device.

#### 4.3.4. Using relay signal

On interface board, OUT0 signal is used to control relay and related pins exist in CON2 terminal. Circuit diagram for relay signals is as follows. When, OUT0 signal is inactive, which is in normal state, N.C. pin is connected with COMMON pin. When OUT0 signal is activated, N.O. pin is connected with COMMON pin.





#### 4.3.5. General IO interface

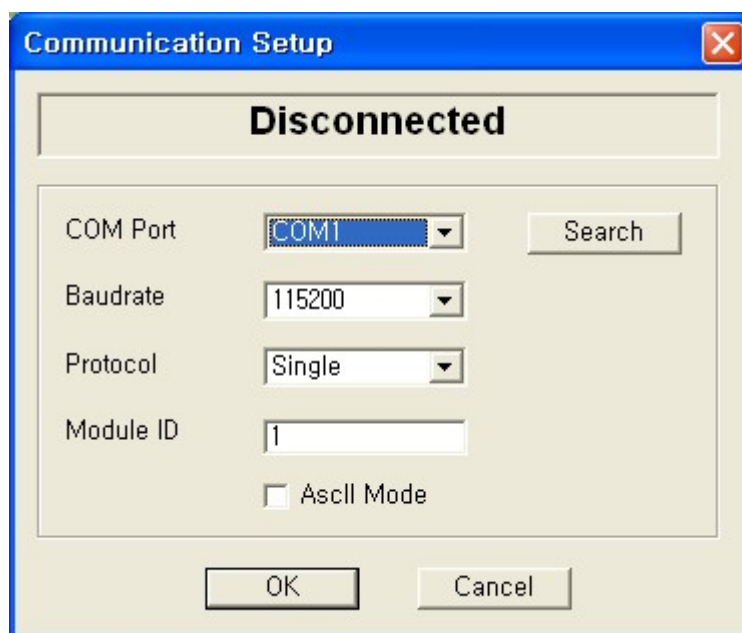
General IO ports of the SFM3500 module, including IN, OUT, and LED ports, are also connected to CON2 connector, which can be utilized for user's application test.

## 5. PC User Interface Program

UniFinger PC user interface program (UniFingerUI\_V\*.exe) is a powerful tool to evaluate full functionalities of UniFinger standalone modules. After the module is connected with PC through a serial interface, it communicates with the module using the Packet Protocol.

### 5.1. Getting started

When the program is executed, it automatically tries to establish serial connection with the evaluation kit. If successful connection is established, the model number and firmware version of the module will be displayed. Otherwise, **Communication Setup** dialog will appear as follows:



If you know the communication parameters such as serial port and baud rate, just change these on this dialog. If you use RS422 or RS485 interface and set the Network Mode parameter to Network(for SFM3000) or Half/Full(for SFM3500), you should choose network protocol and set the module ID correctly. After changing parameters, press the **OK** button to test if these are correct. If the program can connect to the module with these parameters, the communication dialog will disappear. Otherwise, an

error message will be shown.

If you don't know the communication parameters, just press the **Search** button. The program will try all combinations of communication parameters and find the correct one.

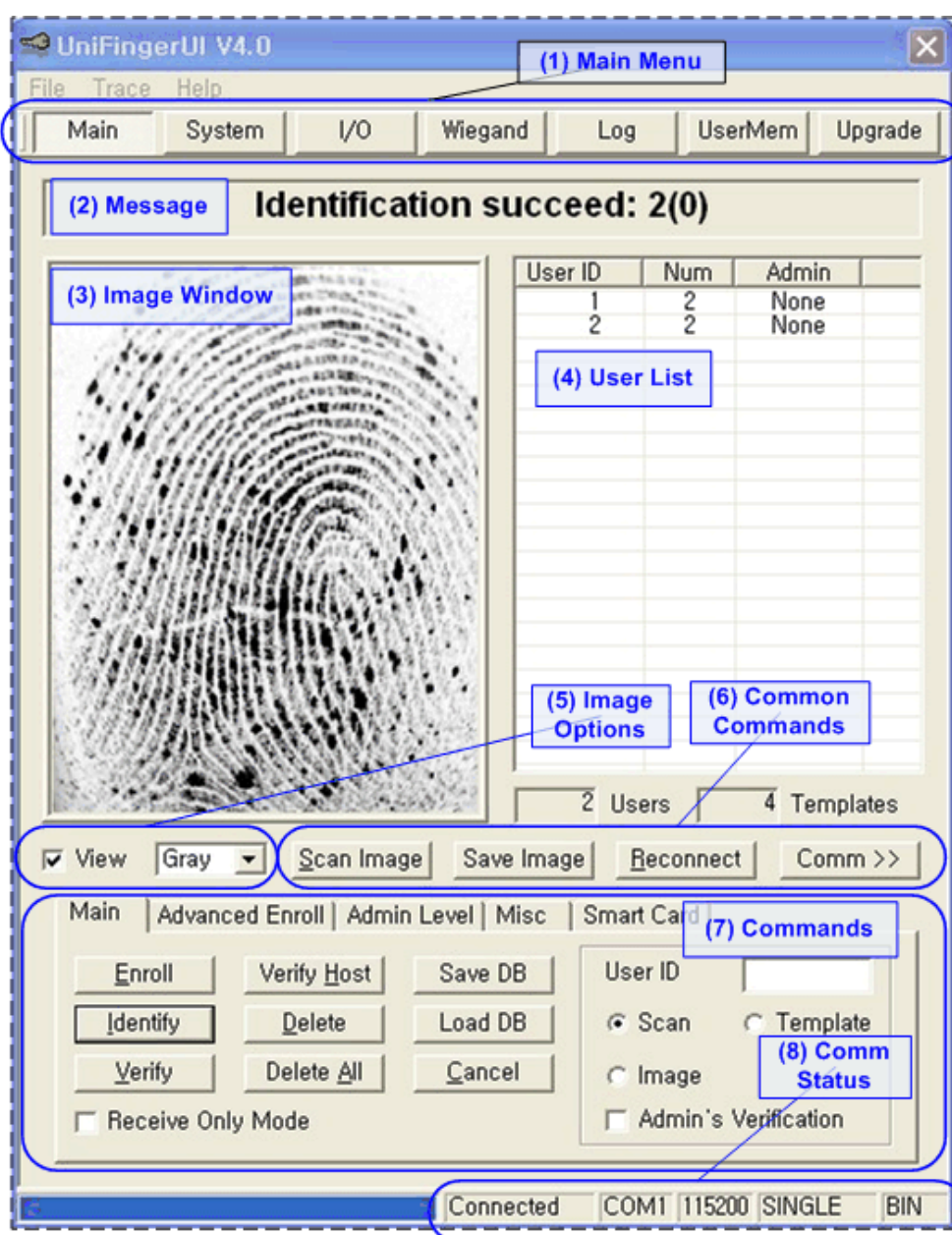
If search fails, please check the followings step by step.

- Check that adequate power is applied to the interface board. Make sure that power LED on the interface board is ON.
- Check that module is properly mounted on the interface board. On type A interface board, bottom line of the module and interface board should be coincident with each other.
- Check that sensor board is properly connected with the module via the sensor cable. Notice that SFM2020 only supports optical sensor via 20 pin flex cable.
- Check that serial cable is connected to the interface board and a PC COM port.

## 5.2. Main window

After connecting to the module, the program will start at the main window. The main window consists of the following components.

- (1) **Main Menu:** You can change to another window by selecting one of the available ones.
- (2) **Message:** Shows the results of commands and error messages.
- (3) **Image Window:** Shows the fingerprint image. Fingerprint images are shown only if the **View** option is checked.
- (4) **User List:** Shows the list of enrolled user IDs. For each user ID, the administration level and the number of enrolled templates are also shown.
- (5) **Image Options:** Determines whether fingerprint images are shown or not. You can also choose the format of fingerprint images.



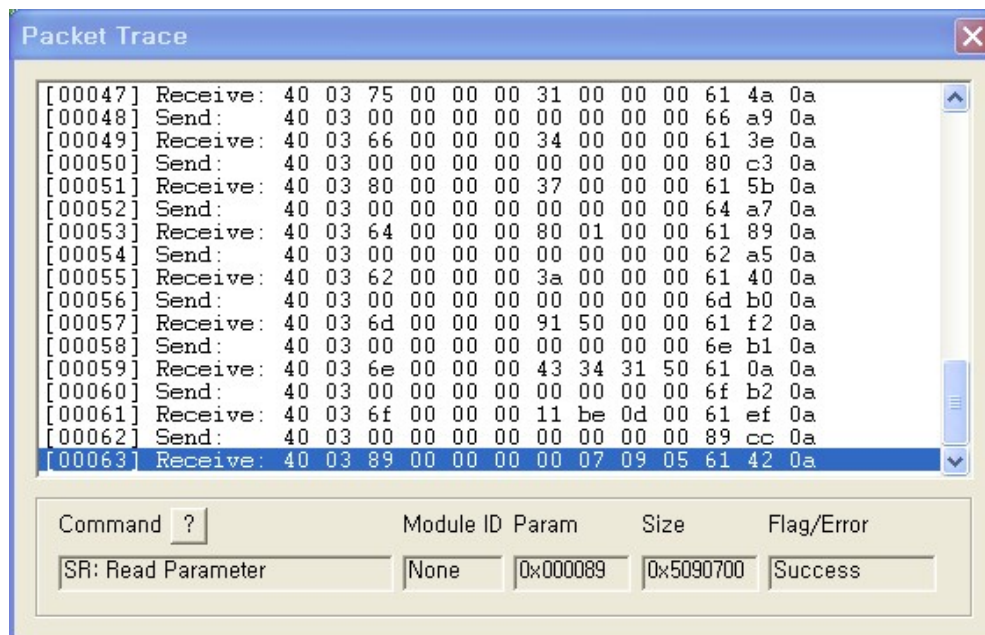
## (6) Common Commands

- A. **Scan Image:** Scans a fingerprint input on the sensor.
- B. **Save Image:** Saves the fingerprint image shown in the **Image Window**.
- C. **Reconnect:** After resetting the system parameters, retrieves the user list again. *You should press **Reconnect** after changing the*

*module connected to the serial port.*

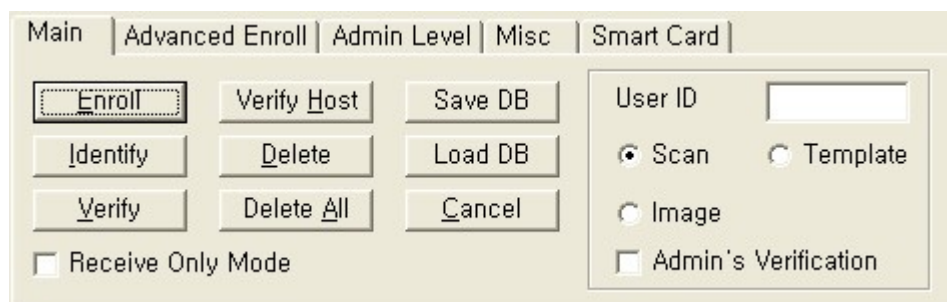
D. **Comm**: Shows the **Communication Setup** dialog.

- (7) **Commands**: The commands are categorized into 5 sub menus. Each menu will be described in the following sections.
- (8) **Comm Status**: Shows the connection status and communication parameters.
- (9) **Packet Trace**: One of the main purposes of the program is helping developers to understand the Packet Protocol. If you select **Trace/Packet Trace** in the pull-down menu, the **Packet Trace** dialog will be popped-up as follows. It shows all the packets exchanged between the program and the module. And if you select a packet in the list, the details of it will be displayed in the bottom of the window. You can also see the online help of the selected command by pressing the ? button.



### 5.2.1. Main commands

You can enroll, identify, verify, and delete templates with these commands.



- **Enroll:** Enrolls templates. If **User ID** field is blank, the enrolled ID will be chosen automatically by the module. You can set the **User ID** field by selecting one in the User List or typing it directly. You can also select one of the following options.
  - A. **Scan:** Scans fingerprint inputs on the sensor and enroll them.
  - B. **Template:** Enrolls a fingerprint template read from a file.
  - C. **Image:** Enrolls a fingerprint image read from a file.

If the **Admin's Verification** option is checked, the program will ask to place an administrator's finger first. If the verification fails or there is no user with Enroll or All administration level, an error message will be shown.

There are other enroll options, too. You can test these advanced options in **Advanced Enroll** window.

- **Identify:** Identifies a fingerprint. You can also select one of the **Scan/Template/Image** options.
- **Verify:** Verifies a fingerprint with the enrolled templates of the specified user ID. You can also select one of the **Scan/Template/Image** options.
- **Verify Host:** Verifies a fingerprint input on the sensor with templates read from a file.
- **Delete:** Deletes the templates of the specified user ID.
- **Delete All:** Deletes all the enrolled templates. If the **Admin's**

**Verification** option is checked, the program will ask to place an administrator's finger first. If the verification fails or there is no user with Delete or All administration level, an error message will be shown.

- **Save DB:** Saves all the templates and information of enrolled user IDs into a file.
- **Load DB:** Loads the templates and user information from a file saved by **Save DB**. All the previously enrolled templates will be erased.
- **Cancel:** When **Scan** option is selected and you press one of the **Enroll/Identify/Verify**, the module will wait for fingerprint inputs. If you press **Cancel**, the module will abort the operation.
- **Receive Only Mode:** If the module processes GPIO inputs or Wiegand inputs, or works in Free Scan mode, response packets are not sent to serial interface by default. By setting Auto Response parameter and checking this option, the program can receive response packets in those cases, too. When this option is checked, all other functions are disabled.

#### 5.2.2. Advanced enroll commands

Main		Advanced Enroll	Admin Level	Misc	Smart Card
User ID	0	Enroll Mode	1 Time	Enroll	
Option	Auto ID	Displacement	No Check	Cancel	
Method	Scan	Image Quality	Weak	Fix Prov.	
<input type="checkbox"/> Admin's Verification		Provisional	Off	Save Option	

There are many options related to enrollment process. You can test all these options in this window.

In addition to **User ID**, **Method**, and **Admin's Verification** which are described in Main commands sections, you can set the following options in the **Option** field

- **None:** Replaces the enrolled templates of the specified user ID with new ones.

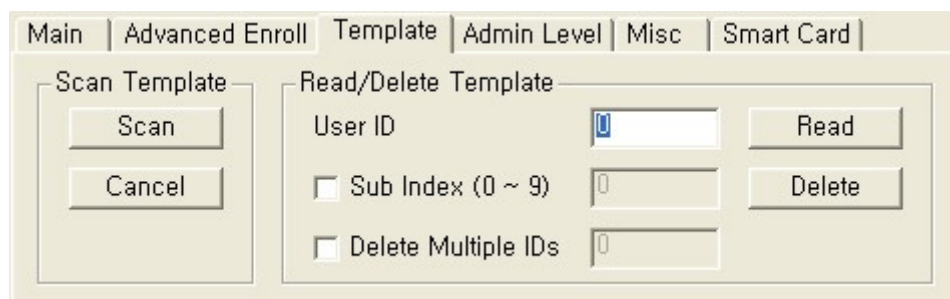
- **Auto ID:** The user id will be assigned automatically by the module.
- **Add New:** Adds fingerprints to the specified user ID. The maximum number of templates per user is 10.
- **Check ID:** Checks if the specified ID has already some templates. If it does, `EXIST_ID` error will be returned.
- **Check Finger:** Checks if the same fingerprint is already enrolled. If it is, `EXIST_FINGER` error will be returned. Otherwise, continue enroll process with **Add New** option.
- **Check Finger Auto ID:** Checks if the same fingerprint is already enrolled. If it is, `EXIST_FINGER` error will be returned. Otherwise, continue enroll process with **Auto ID** option.

You can also change the system parameters related to enrollment process. Refer to the Packet Protocol Manual for details of these parameters.

You can use the following four commands in this window.

- **Enroll:** Enrolls fingerprints with the selected options.
- **Cancel:** Aborts the enrollment process.
- **Fix Prov.:** If Provisional system parameter is on, the enrolled templates are saved into DRAM, not into flash memory. **Fix Prov.** command moves those templates in DRAM into flash memory.
- **Save Option:** Saves the system parameters into flash memory.

### 5.2.3. Template commands



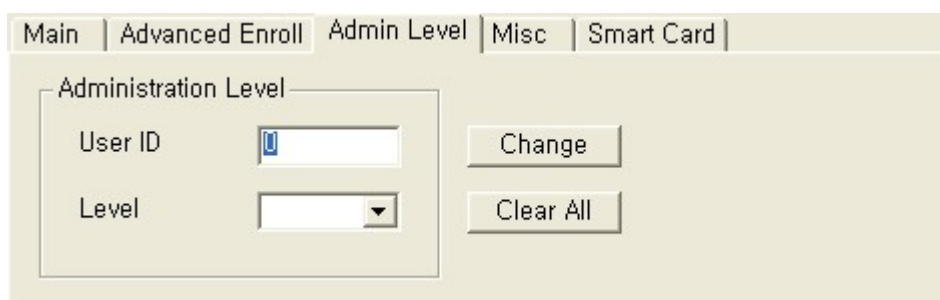
- **Scan:** Scans a fingerprint input on the sensor and saves the template



into a file.

- **Cancel:** Cancels the **Scan** command.
- **Read:** Reads the templates of the specified user ID. If **Sub Index** is checked, reads only one template of the specified sub index.
- **Delete:** Deletes the templates of the specified user ID. If **Sub Index** is checked, deletes only one template of the specified sub index. If **Delete Multiple IDs** is checked, deletes the templates of the multiple user IDs. For example, if **User ID** field is 2 and **Delete Multiple IDs** field is 10, the module will delete the templates of ID2 ~ ID10.

#### 5.2.4. Admin level commands

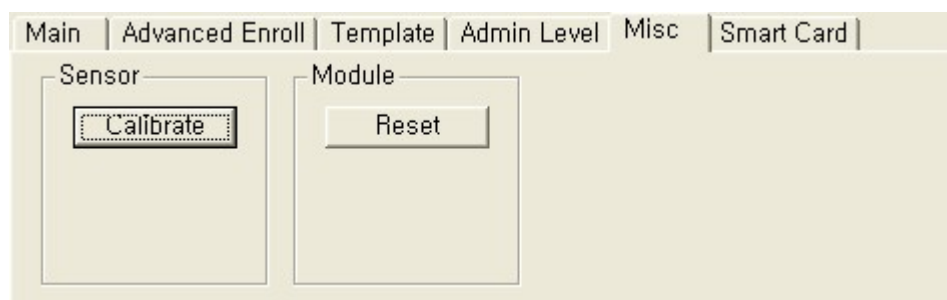


The screenshot shows a software interface with a tabbed menu at the top: 'Main', 'Advanced Enroll', 'Admin Level', 'Misc', and 'Smart Card'. The 'Admin Level' tab is selected. Below the tabs is a window titled 'Administration Level'. This window contains two input fields: 'User ID' (a text box with a blue icon) and 'Level' (a dropdown menu). To the right of these fields are two buttons: 'Change' and 'Clear All'.

Enroll and delete commands can change the fingerprint DB stored in the module. For some applications, it might be necessary to obtain administrator's permission before enrolling or deleting fingerprints. If **Admin's Verification** option is checked, a user with proper administration level should verify first to enroll or delete. You can change the administration level of a user in this window.

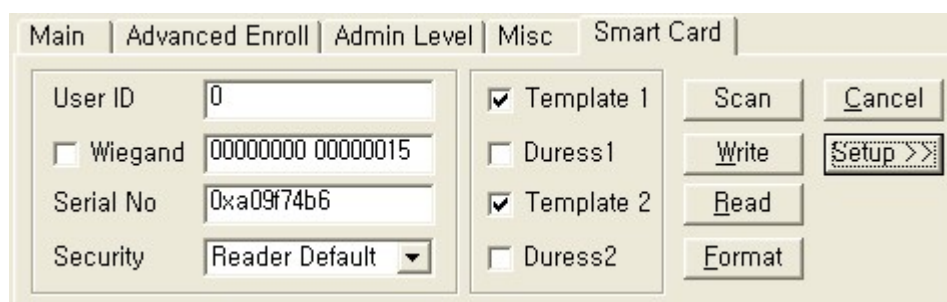
- **Change:** Changes the administration level of a user.
- **Clear All:** Resets the administration levels of all users to None.

#### 5.2.5. Misc commands



- **Calibrate:** Calibrates the sensor. This command is supported for FL and TC models only.
- **Reset:** Resets the module.

#### 5.2.6. Smart card commands



These commands are supported for BioEntry Smart only. You can write/read/format smart cards using these commands.

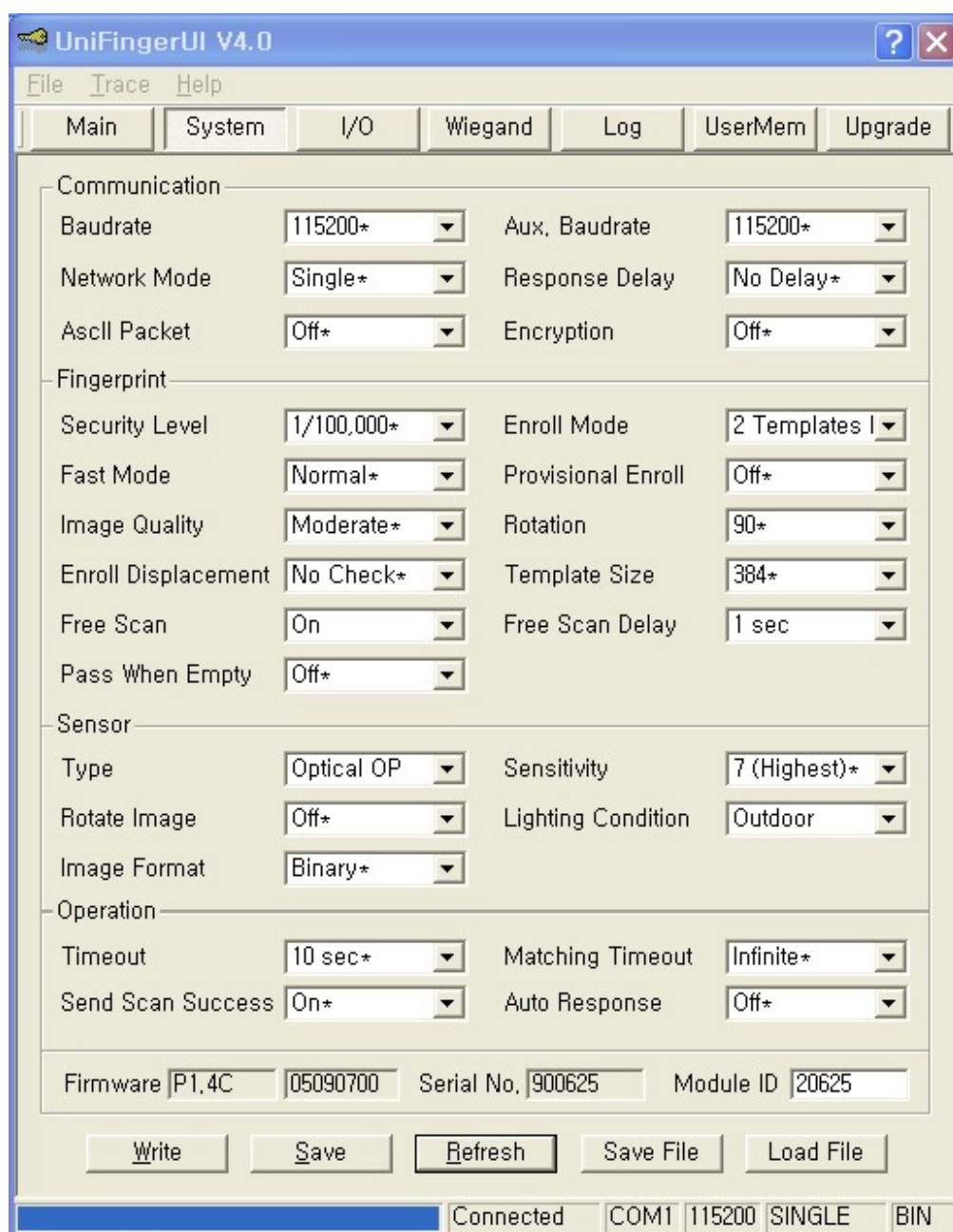
- **Scan:** To write templates into a smart card, you should scan fingerprints first. To enroll two templates into a smart card, select **Template 1** and **Template 2**. To enroll one template, just select **Template 1**. You can also choose if one of the enrolled templates is duress finger.
- **Write:** After scanning fingerprints, you can write these templates into a smart card. You should choose the **User ID** and **Security** level of the templates before writing. Instead of entering the **User ID**, you can also use **Wiegand** string directly.
- **Read:** Reads the contents of a smart card.
- **Format:** Formats a smart card.

- **Cancel:** Cancels the previously issued command.
- **Setup:** Shows the **SmartCard Setup** dialog. You can change site keys, key options, operation mode, and card layout in the dialog.

### 5.3. System parameters window

You can read and write system parameters in this window. If the meaning of a system parameter is not clear, drag the **?** button on the title bar into the parameter field. Then, the online help will be popped up and the description of the parameter will be displayed.

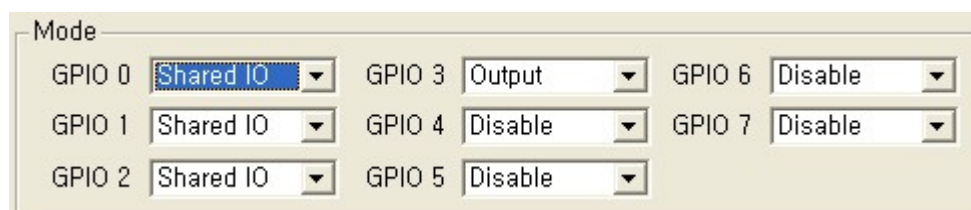
- **Write:** Writes system parameters. Only the modified system parameters will be written.
- **Save:** **Write** command changes the parameters in DRAM, not in flash memory. To make these changes permanent, you should press the **Save** button.
- **Refresh:** Rereads all the system parameters.



- **Save File:** Saves the system parameters into a file.
- **Load File:** Loads system parameters from a file. **Save File** and **Load File** can be very useful when setting same parameters in lots of modules. To make the loaded parameters permanent, press the **Save** button.

#### 5.4. IO window for SFM3000

SFM3000 series modules have 8 configurable GPIO ports. Using these ports, users can implement their own IO functions, such as identify button, LED output, lock control, Wiegand interface, and so on. PC UI provides easy method to program the configurations of GPIO.



To configure a GPIO port, you should select the modes of the GPIO ports in the **Mode** sub window. There are 5 types of GPIO mode; Input, Output, Shared IO, Wiegand Input, and Wiegand Output. Input modes, including general Input and Shared IO, are configurable on GPIO 0 to GPIO 3. Output mode is configurable on any GPIO port. Wiegand modes are restricted on special ports. Wiegand Output mode is only configurable on the couple of GPIO 4 and GPIO 5, and Wiegand Input is configurable on the couple of GPIO 2 and GPIO 3, respectively.

- **Write:** Writes GPIO configurations. To make these changes permanent, you should press the **Save** button.
- **Default:** Resets GPIO configurations to the default values.
- **Save:** Saves the GPIO configurations into flash memory.
- **Refresh:** Rereads the GPIO configurations from the module.
- **Save File:** Saves GPIO configurations into a file.
- **Load File:** Loads GPIO configurations saved by **Save File**.

#### 5.4.1. Input port

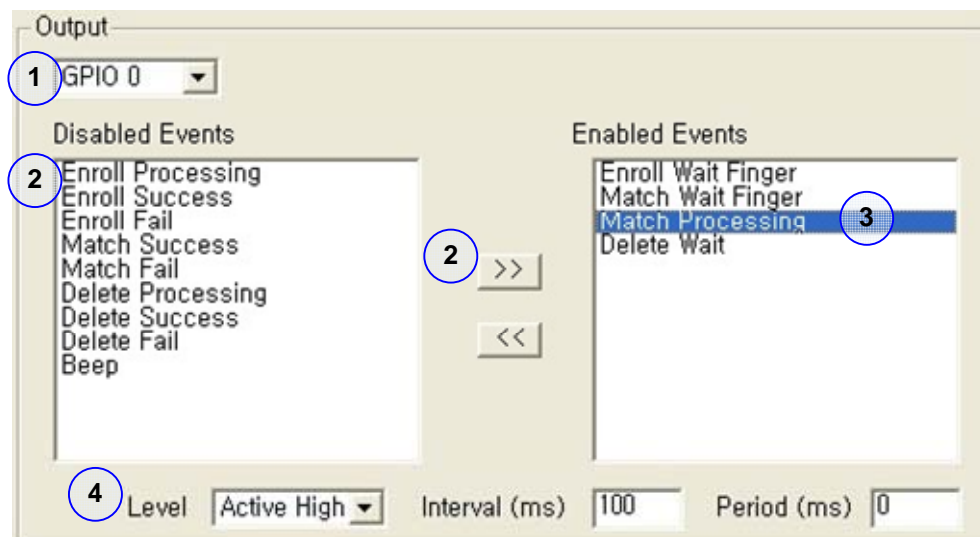
An input function is assignable to the ports whose mode is Input or Shared IO.



- (1) Select the port in the **Input** sub window.
- (2) Select one of the input functions.
- (3) Select the activation level of the input signal. To prevent inadvertent deletion of all templates, 'Delete All by Confirm' function requires two input signals. After sending the first signal, users should send the second one before the timeout period expires. Therefore, if you select 'Delete All by Confirm' as the input function, you should also set the timeout period of it.

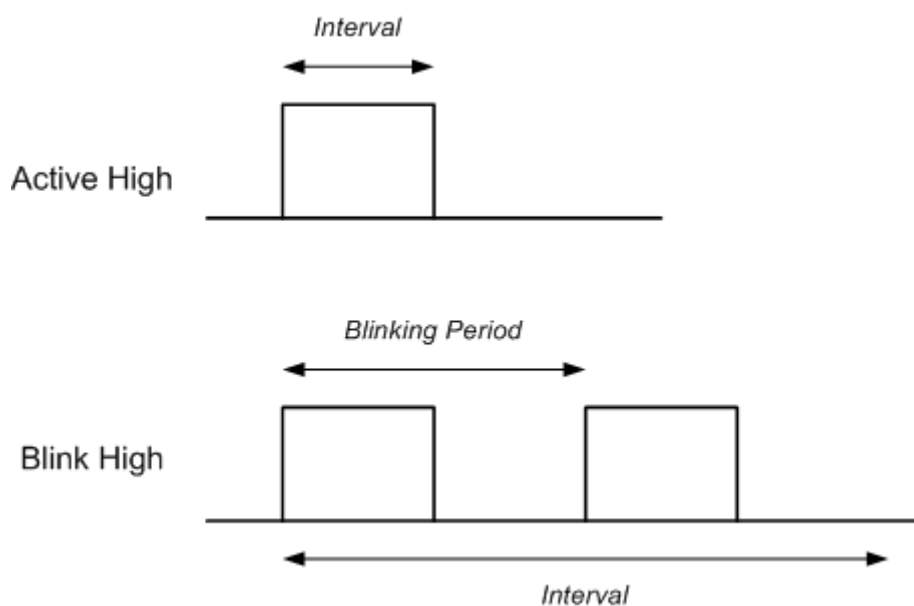
#### 5.4.2. Output port

Output events are assignable to the ports whose mode is Output or Shared IO.



- (1) Select the port in the **Output** sub window.
- (2) Enable an event by pressing >> button or double-clicking on an event in the **Disabled Events** list.
- (3) Select an event in the **Enabled Events** list.

- (4) Configure the output signal of the selected event. When output level is Blink High or Blink Low, you should set Blinking **Period** in addition to **Interval**. The following figure shows the relationship between **Interval** and Blinking **Period**.



#### 5.4.3. Wiegand

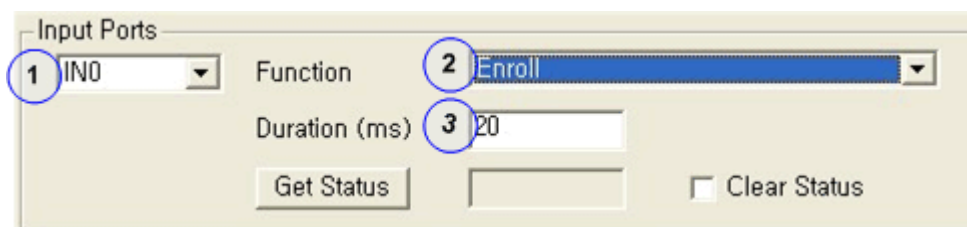
If you assign Wiegand Input or Wiegand Output ports, you should configure the Wiegand format. SFM3000 modules only support 26 bit format and users can set the FC code and the number of FC bits.

#### 5.5. IO window for SFM3500

SFM3500 modules have three input ports, three output ports, and three LED ports which are configurable for specific functions.

- **Write:** Writes IO configurations. To make these changes permanent, you should press the **Save** button.
- **Save:** Saves the IO configurations into flash memory.
- **Refresh:** Rereads the IO configurations from the module.
- **Save File:** Saves IO configurations into a file.
- **Load File:** Loads IO configurations saved by **Save File**.

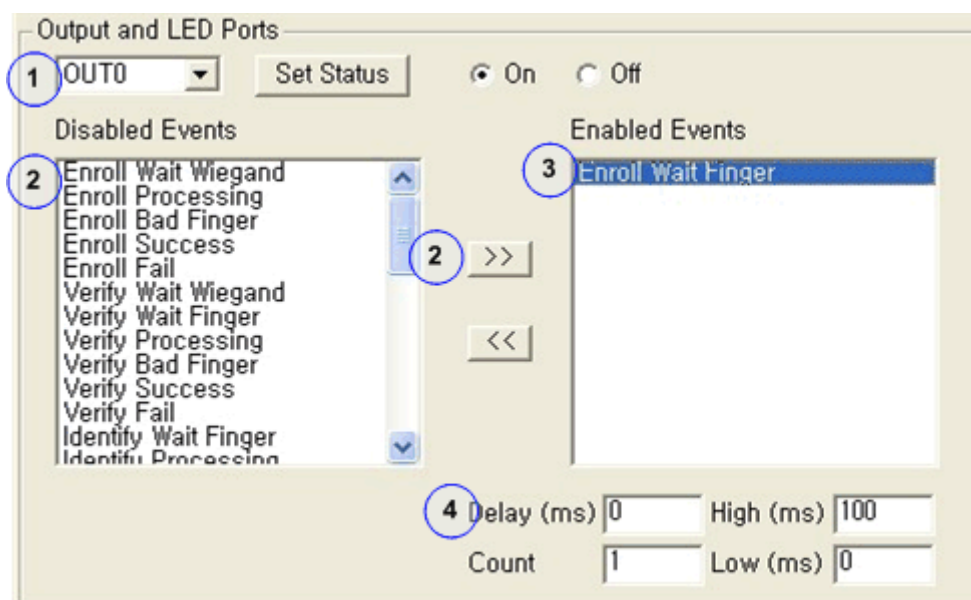
## 5.5.1. Input port



- (1) Select one of the input ports.
- (2) Select one of the input functions.
- (3) Configures the duration of input signal. The minimum duration is 10 ms. If the input signal is shorter than this value, it will be ignored by the module.

You can also read the status of an input port by pressing the **Get Status** button. If **Clear Status** option is checked, the input status will be cleared after reading.

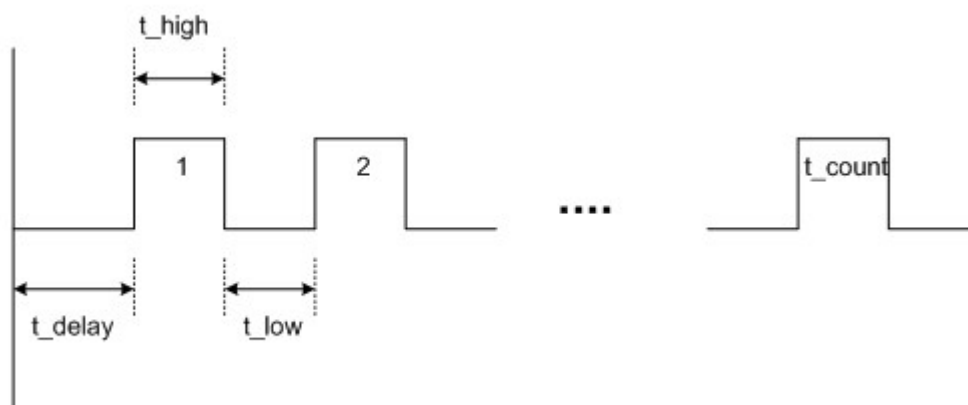
## 5.5.2. Output/LED port



- (1) Select the port in the **Output and LED Ports** sub window.
- (2) Enable an event by pressing >> button or double-clicking on an event in the **Disabled Events** list.



- (3) Select an event in the **Enabled Events** list.
- (4) Configure the output signal of the selected event. The following figure shows examples of an output signal.

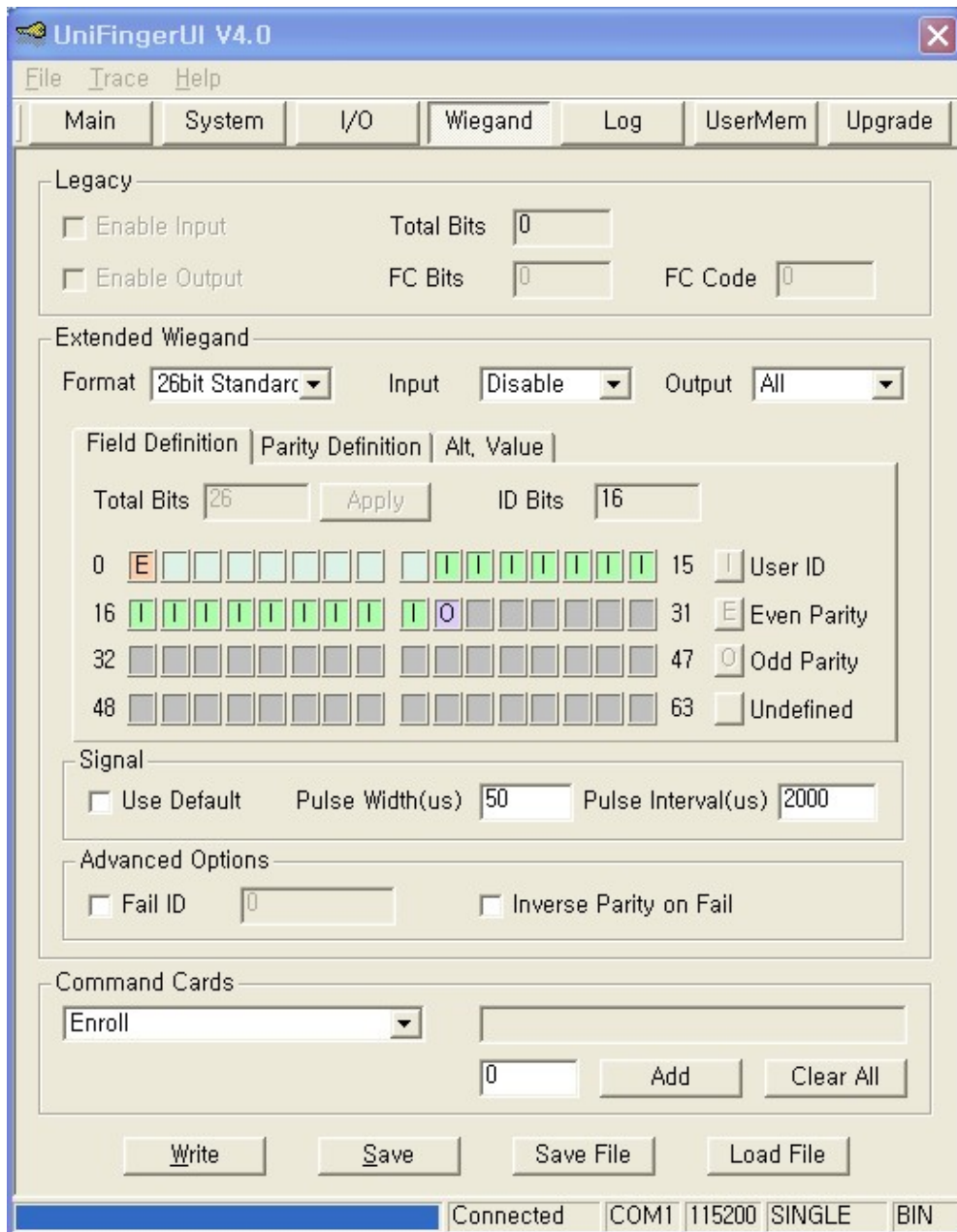


You can also set the status of an output port by pressing the **Set Status** button.

## 5.6. Wiegand window

Since firmware V1.4, SFM3500 modules provide Extended Wiegand Protocol, which supports up to 64 bit Wiegand formats. The only constraint is that the size of a field is limited to 32 bits. It also supports advanced options such as Fail ID and Inverse Parity on Fail.

- **Write:** Writes Wiegand configurations. To make these changes permanent, you should press the **Save** button.
- **Save:** Saves Wiegand configurations into flash memory.
- **Save File:** Saves Wiegand configurations into a file.
- **Load File:** Loads Wiegand configurations saved by **Save File**.



#### 5.6.1. Legacy format

In firmware versions older than V1.4, only the 26 bit Wiegand format is supported. You can configure the FC code and the number of FC bits. You can also enable/disable Wiegand ports.

#### 5.6.2. Input/output port

- **Input:** Enable/disable Wiegand input port.

- **Output:** Selects when the Wiegand signal is to be output. If it is set to Wiegand Only, the module outputs Wiegand signal after successful matching only if the verification is initiated by Wiegand input. If it is set to All, the module outputs Wiegand signal whenever matching succeeds.

#### 5.6.3. Signal

- **User Default:** Uses default values for Wiegand signal.
- **Pulse Width:** The width of pulse. The default is 50 us.
- **Pulse Interval:** The interval of pulse. The default is 2000 us.

#### 5.6.4. Advanced options

- **Fail ID:** Normally the module outputs Wiegand signal only if matching succeeds. If this option is checked, the module outputs the fail ID when matching fails.
- **Inverse Parity on Fail:** If this option is checked, the module outputs Wiegand signals with inverted parities when matching fails.

#### 5.6.5. 26 bit standard format

The 26 bit standard format is most widely used and consists of 8 bit FC code and 16 bit ID. Users can set an alternative FC code and enable advanced options.

#### 5.6.6. Pass Through format

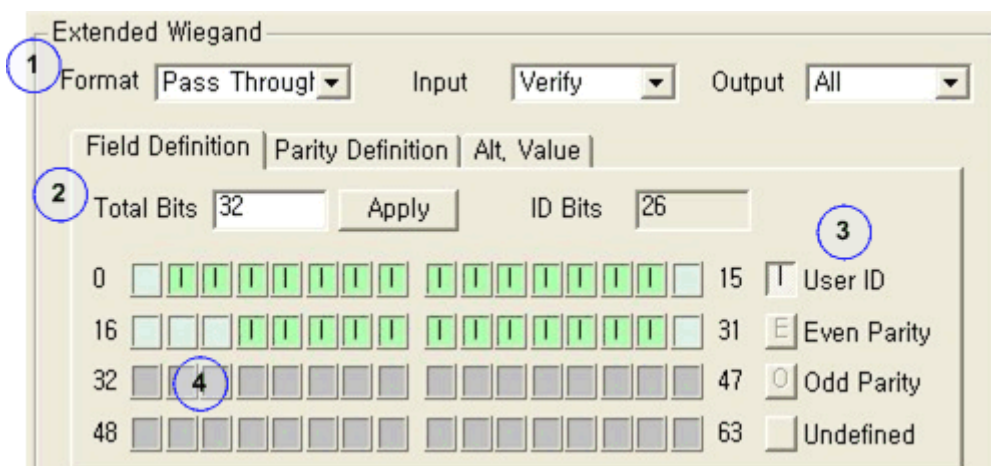
Pass Through format is used when only the format of ID field is known. When Wiegand input string is detected, the module extracts ID bits and starts verification with the ID. If the verification succeeds, the module outputs the Wiegand input string as unchanged. Parity check and advanced options are ignored in this format. By definition, Pass Through format is only useful when the matching is initiated by Wiegand input. If the matching is initiated by Packet Protocol or GPIO input, the bits other than ID field are set to 0.

For example, assume that 32 bit Pass Through format is composed as follows:

XIIIIIIII IIIIIIIIX XXXIIIIII IIIIIIIIX (left most bit is 0<sup>th</sup> bit, BIT0)

I: Id field, X: Unknown field

You can configure this format in the following sequences.



- (1) Select Pass Through in the **Format** field.
- (2) Enter 32 in the **Total Bits** field and press the **Apply** button.
- (3) Select **User ID**.
- (4) Press the ID bits according to the definition.
- (5) Press **Write** or **Save** button.

#### 5.6.7. Custom format

When users know all the information of a Wiegand format, Custom format can be defined. When Wiegand input string is detected, the module checks the parity bits first. If all the parity bits are correct, the module extracts ID bits and starts verification with the ID. Users can also set alternative values of each field and enable advanced options such as Fail ID. If the verification succeeds, the module outputs a Wiegand string. The output string may be different from the input string according to the alternative values and advanced options.

For example, assume that 44 bit Custom format is composed as follows:

```
EAAAAAAAA IIIIIIII IIIIIIII BBBBBI IIIIIIII IIO
```

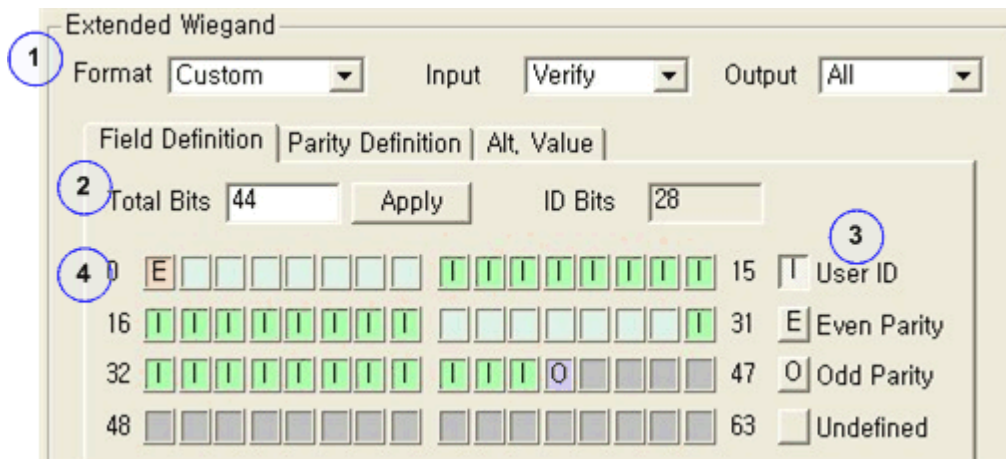
(left most bit is 0<sup>th</sup> bit, BIT0)

E: Even parity for BIT1 ~ BIT22

O: Odd parity for BIT23 ~ BIT42

I: ID bits(Field1 and Field 3), A: Field 0, B: Field 2

You can configure this format in the following sequences.



(1) Select Custom in the **Format** field.

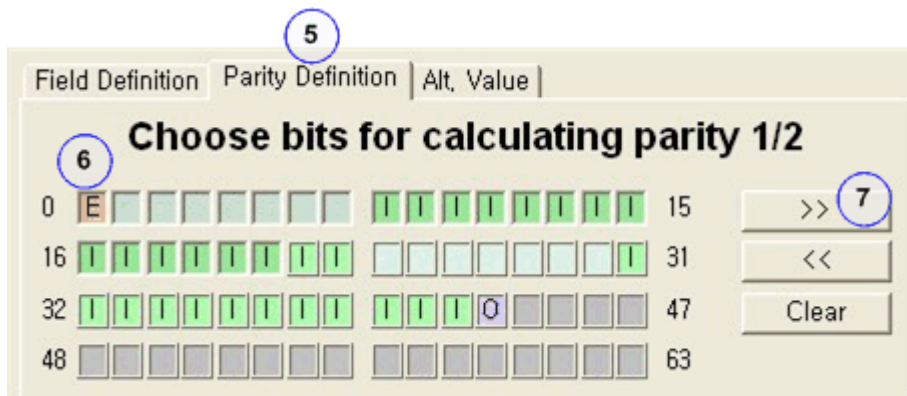
(2) Enter 44 in the **Total Bits** field and press the **Apply** button.

(3) Select **Even Parity**.

(4) Press the even parity bit. In this example, it is BIT0.

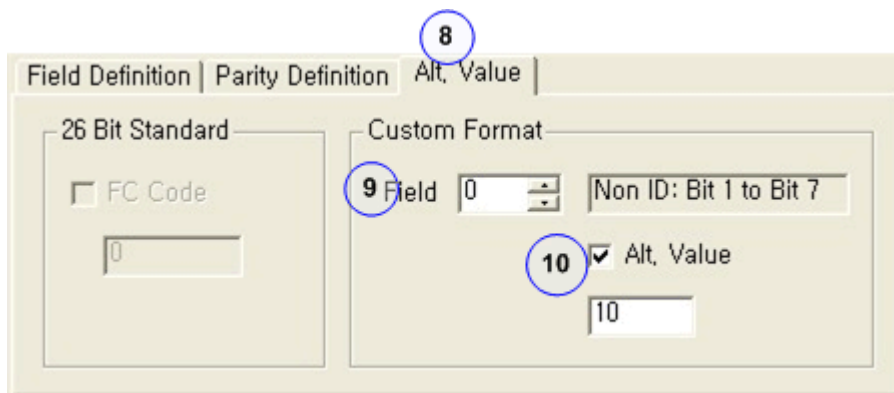
Repeat (3) and (4) for **Odd Parity** and **User ID** according to the definition.

After configuring the bit format, you should configure parity information.



- (5) Select **Parity Definition** sub menu.
- (6) Press the bits which are used in calculating the first parity bit. In this example, they are BIT1 ~ BIT22.
- (7) Press the >> button and repeat (6) for the second parity bit. In this example, they are BIT23 ~ BIT42.

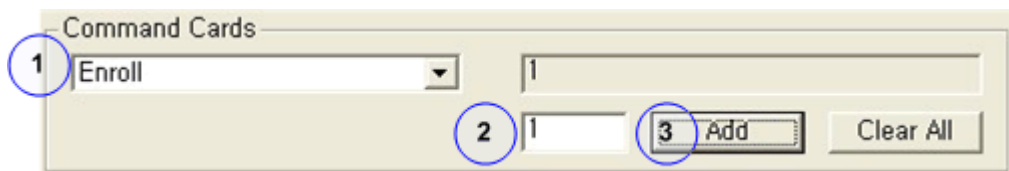
Optionally, alternative values for non-ID fields can be set.



- (8) Select **Alt. Value** sub menu.
- (9) Select a non-ID field.
- (10) Check **Alt. Value** and enter the alternative value.

#### 5.6.8. Command card

Users can map an input function to a Wiegand ID. When the module detects the mapped IDs in Wiegand input port, it will execute the corresponding input function.



For example, you can map ID1 to enroll function as follows:

- (1) Select Enroll in the **Command Cards** list.
- (2) Enter 1 in the text field.

(3) Press the **Add** button.

## 5.7. Log window

SFM3500 modules provide logging facility for recording important events. Users can receive logs from the module and delete unwanted ones.

In the **Time** sub menu, you can do the followings:

- **Get**: Gets the time of RTC embedded in the module.
- **Set**: Sets the time of RTC embedded in the module.

In the **Upload** sub menu, you can do the followings:

- **Selected**: Reads log records from the start index to the end index.
- **Latest**: Reads latest log records. If the **Append** option is checked, the log records are appended to the previously read ones.
- **All**: Reads all the log records.

In the **Delete** sub menu, you can do the followings:

- **Oldest**: Deletes oldest log records. The number of logs to be deleted should be a multiple of 256.
- **All**: Deletes all the log records.

## 5.8. User memory window

SFM modules reserve 256 bytes for user data. This area can be read and written in this window.

- **Write**: Writes the user data to the module.
- **Save**: Saves the user data into flash memory.
- **Refresh**: Rereads the user data.

## 5.9. Upgrade window

You can upgrade firmware, save system configurations, and load configurations in this window.

## 5.9.1. Firmware upgrade



(1) Choose the firmware file to upgrade.

(2) Select one of the two upgrade methods.

- **Upgrade Command:** Upgrades firmware using UG command defined in the Packet Protocol. UG command has been supported since firmware V1.4.
- **Bootloader:** Upgrade is handled by the boot loader not by Packet Protocol. This option is available in all versions of firmware.

(3) Press the **Start** button. The upgrade process varies according to the selected method.

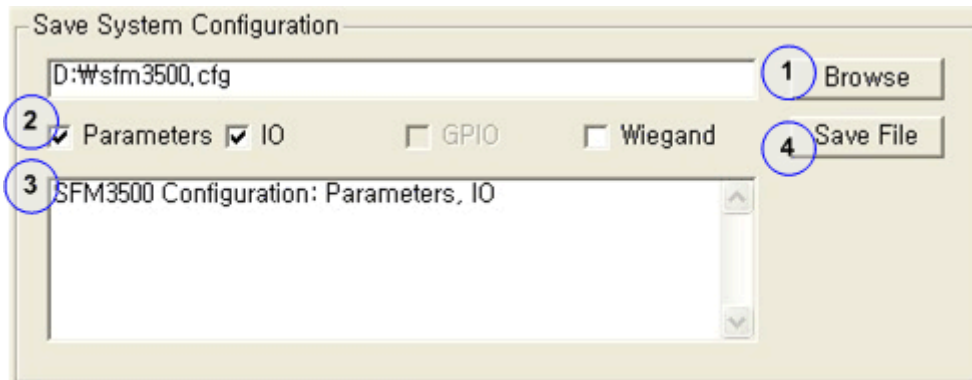
- **Upgrade Command:** Upgrade starts automatically. After upgrade is finished, you don't have to reset the module.
- **Bootloader:** After pressing the **Start** button, you should reset the module. After upgrade is finished, you should reset the module again.

Firmware upgrade in UniFingerUI V4.0 is only applicable to SFM3000/3500 series modules. For SFM1000/2000 modules, older versions of UniFingerUI should be used.

## 5.9.2. Save system configuration

As described earlier, you can save configurations of system parameters, IO, and Wiegand format into a file in respective windows. You can also save all or a combination of these configurations in **Save System Configuration** sub menu.

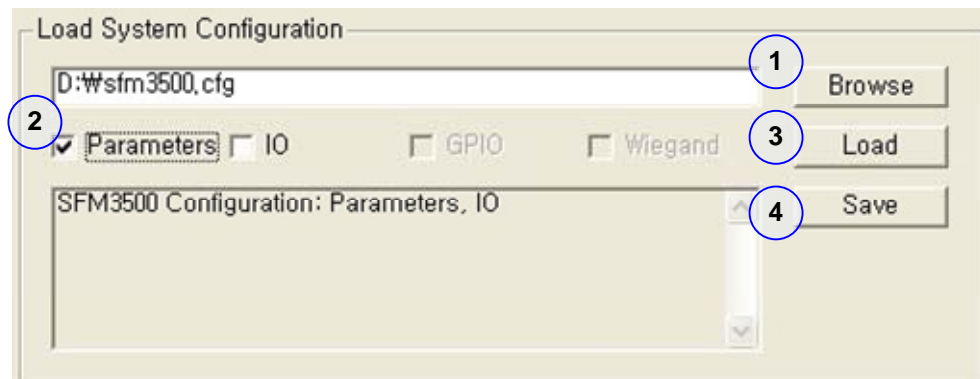




- (1) Choose a file to which the configurations will be saved.
- (2) Choose the configurations to be saved.
- (3) Optionally, you can add a short description of the configuration file.
- (4) Save the selected configurations into the specified file.

#### 5.9.3. Load system configuration

You can load a configuration file saved in **Save System Configuration** sub menu or respective windows.



- (1) Choose the configuration file to be loaded. If it is a valid configuration file, the description and configurations contained in it will be shown.
- (2) Choose the configurations to be loaded.
- (3) Load the selected configurations into the module.
- (4) Save the loaded configurations into flash memory.

## 6. Command Line Tool

UniFinger Command Line Tool is an application program helping evaluation and test of UniFinger modules by command line. User can try all commands defined in the *Packet Protocol Manual* and related command and response packets between host and the module can be traced. This application is implemented on top of the UniFinger module SDK.

In the evaluation kit package, application program is included running on PC Win32 and Linux environment named by **ufcmd.exe** and **ufcmd** each other. Also, the source code of command line tool is included, helping users to learn how to develop applications using the SDK.

For example, user can check the packets between host and the module for VT command as follows:

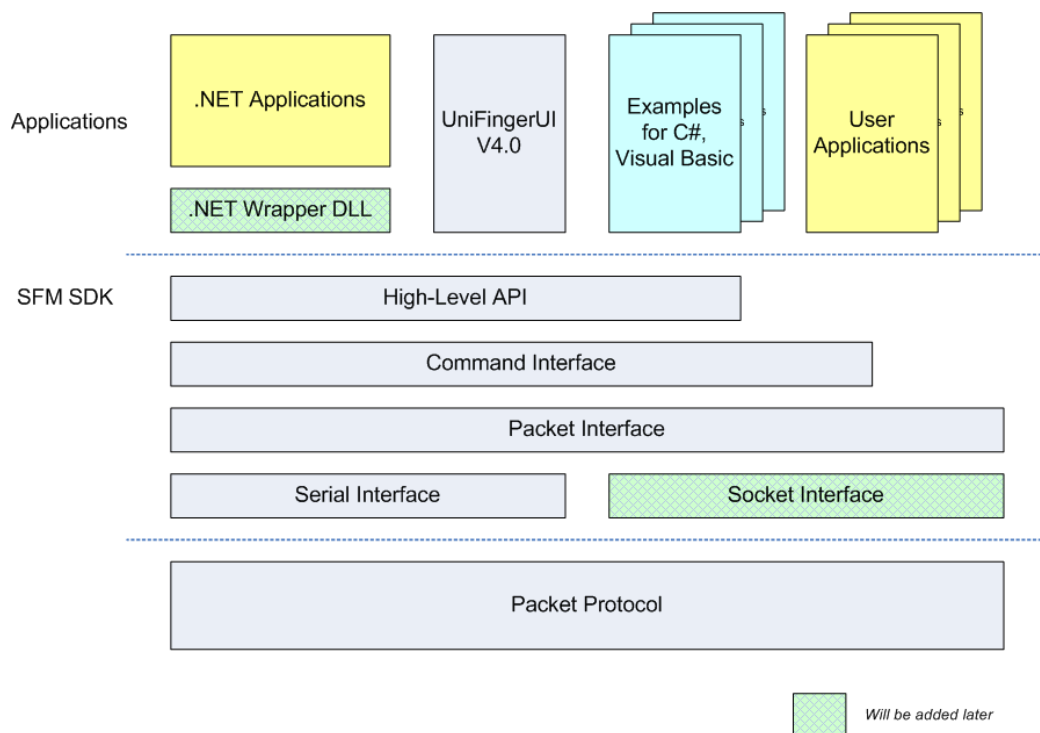
```
>ufcmd -x3 -c vt -i template\template1
[API] Verify by Template(VT): 0 0
      [COM] Send: 40 09 00 00 00 00 00 00 00 00 00 49 0a
      [COM] Receive: 40 09 00 00 00 00 00 00 00 00 62 ab 0a
      [COM] Receive: 40 09 00 00 00 00 00 00 00 00 69 b2 0a
[ERR] Verify: Not found
[API] Finalize UniFinger module
```

Command line tool is also useful in processing sequence of multiple commands, such as system parameter or IO configuration setting. Provided **SetEVKIo.bat** is an example for setting IO configuration.

For further information, please refer to *UniFinger Command Line Tool Manual*.

## 7. SFM SDK

The SFM SDK is a collection of APIs for interfacing with SFM modules and BioEntry readers. In addition to simple wrapper functions for Packet Protocol, it also provides high level APIs such as template DB management, image manipulation, etc. By using the SDK, developers could write Win32 applications quickly without knowing the minute details of Packet Protocol.



As Shown in the above figure, the SDK is composed of several layers and developers could choose whichever layer suited for their applications. Another strong point of the SDK is its extensibility. Many of core APIs provide callback mechanism, with which developers can add customized functions. UniFingerUI V4.0 is a good example of this feature. Completely rewritten from scratch, UniFingerUI V4.0 covers all the core functionalities of SFM modules and shows how to use the SDK in real applications. The source codes of it are also included in the SDK.

For further information, refer to the *SFM SDK Reference Manual*.

## Contact Info

- **Headquarters**

Suprema, Inc. (<http://www.supremainc.com>)

16F Parkview Office Tower,

Jeongja-dong, Bundang-gu,

Seongnam, Gyeonggi, 463-863 Korea

Tel: +82-31-783-4505

Fax: +82-31-783-4506

Email: [sales@supremainc.com](mailto:sales@supremainc.com), [support@supremainc.com](mailto:support@supremainc.com)