



## Serial Protocol

---

**NITGEN®**

RS-232C serial protocol for Stand-Alone Fingerprint Recognition Device

---

## Protocol Guide

(Supported device: FIM40 Ver. 1.00,  
FIM50 Ver 1.00)

Version 2.00

## Serial Protocol

---

© Copyright 2010 NITGEN Co., Ltd.

ALL RIGHTS RESERVED

Serial Number:

Specifications can be changed without notice.

# Serial Protocol

---

## Revision Information

Date	Version	Description
2010-05-03	2.0	Release

## Contents

<b>1. GENERAL DESCRIPTION .....</b>	<b>7</b>
<b>2. PACKET DESCRIPTION .....</b>	<b>8</b>
■ PACKET STRUCTURE.....	8
■ ERROR CODE.....	9
■ HOW TO MAKE THE HEADER CHECKSUM & THE DATA CHECKSUM.....	9
<b>3 COMMAND DESCRIPTION .....</b>	<b>10</b>
● INITIALIZATION .....	10
CMD_REQUEST_CONNECTION.....	10
CMD_SET_BAUDRATE .....	11
CMD_GET_FIRMWARE_VERSION2 .....	12
CMD_GET_DEVICE_INFO.....	13
● MATCHING .....	14
CMD_VERIFY_FP.....	14
CMD_IDENTIFY_FP .....	16
CMD_IDENTIFY_RID_FP .....	18
CMD_INSTANT_MATCHING.....	20
CMD_GET_TEMPLATE.....	21
CMD_CANCEL .....	22
CMD_INSTANT_VERIFY .....	23
CMD_INSTANT_IDENTIFY.....	24
CMD_AUTO_IDENTIFY .....	26
CMD_AUTO_IDENTIFY_RESULT.....	27
● DATABASE MANAGEMENT .....	28
CMD_DELETE_FP.....	28
CMD_DELETE_ALL_FP.....	29
CMD_SET_MASTER .....	30

# Serial Protocol

---

CMD_LEAVE_MASTER_MODE .....	31
CMD_SET_MASTER_PASSWORD .....	32
CMD_READ_USER_DATA .....	33
CMD_WRITE_USER_DATA .....	34
CMD_ERASE_USER_DATA_BLOCK .....	35
CMD_DELETE_MASTER_PASSWORD .....	36
CMD_ENTER_MASTER_MODE2 .....	37
CMD_GET_FP_LIST2 .....	39
CMD_GET_MASTER_LIST2 .....	40
CMD_READ_LOG_DATA2 .....	41
CMD_REGISTER_FP .....	42
CMD_CHANGE_FP .....	44
CMD_ADD_FP .....	46
CMD_GET_FP .....	48
CMD_DELETE_ALL_LOG .....	51
CMD_REGISTER_MULTI_FP .....	52
● CONFIGURATION .....	54
CMD_SET_OPP_OPTION .....	54
CMD_GET_OPP_OPTION .....	55
CMD_SET_SECURITY_LEVEL .....	56
CMD_GET_SECURITY_LEVEL .....	57
CMD_SET_CAPTURE_OPTION .....	58
CMD_GET_CAPTURE_OPTION .....	59
CMD_SET_LOG_OPTION .....	60
CMD_GET_LOG_OPTION .....	61
CMD_SET_CAPTURE_PERIOD .....	62
CMD_GET_CAPTURE_PERIOD .....	63
CMD_SET_SYSINFO .....	64
CMD_GET_SYSINFO .....	65
CMD_SAVE_SYSINFO .....	66
CMD_CHG_NUM_OF_TEMP .....	67
CMD_SET_DEFAULT_SYSINFO .....	68
CMD_CHG_EMULMODE .....	69

# Serial Protocol

---

<code>CMD_CHG_LENGTH_OF_USERID</code> .....	70
● SYSTEM MANAGEMENT .....	71
<code>CMD_STATUS_CHECK</code> .....	71
<code>CMD_GET_FP_IMAGE2</code> .....	72
<code>CMD_UPGRADE_FIRMWARE2</code> .....	73
<code>CMD_SET_TIME</code> .....	74
<code>CMD_GET_TIME</code> .....	75
<code>CMD_GET_IMAGE_QUALITY</code> .....	76
<b>APPENDIX A. LOG DATA BLOCK</b> .....	<b>77</b>
■ LOG DATA FORMAT .....	77
<b>APPENDIX B. EMULATION MODE COMMAND TABLE</b> .....	<b>80</b>
■ FIM20 EMULATION COMMAND.....	80
■ FIM30 EMULATION COMMAND.....	82
<b>APPENDIX C. PACKET RESULT LIST</b> .....	<b>84</b>
<b>APPENDIX D. DATA STURCTURE</b> .....	<b>88</b>
<b>APPENDIX E. EXAMPLES</b> .....	<b>99</b>
1. REQUEST CONNECTION .....	99
2. USER ENROLLMENT.....	100
3. USER DELETION .....	110
4. AUTHENTICATION .....	111
<b>APPENDIX F. THE EXAPMLE OF FIRMWARE UPGRADE</b> .....	<b>116</b>
<b>APPENDIX G. SUPPORT INFORMATION</b> .....	<b>117</b>

# Serial Protocol

---

## 1. GENERAL DESCRIPTION

NITGEN FIM modules are independent fingerprint identification devices processing commands through UART interface. They receive command from host, run function and return result of command. Host can control FIM modules and check the result by using command.

Command is sent and received by packet structure. This packet consists of communication start byte, command code, parameters, data size and auxiliary data. In packet, checksum is added for

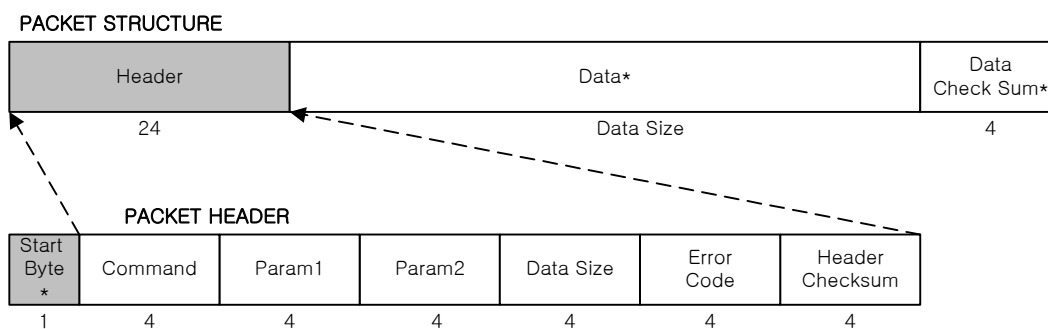
This document describes packet structure and command code for FIM modules.

# Serial Protocol

## 2. PACKET DESCRIPTION

### ■ Packet Structure

The following figure shows the organization of a packet. The packet consists of start byte, header, data (optional), and data checksum (optional). Data and data checksum block is sent only if needed.



\* If data size is zero, then data and data check sum is not used.

\* Start byte: 0x7E

The maximum size of a packet is 64Kbyte.

$$\text{Size (Start Byte)} + \text{Size (Header)} + \text{Size (Data)} + \text{Size (Data Checksum)} \leq 65,536$$

If data size is so large that the host/device cannot carry data in a single packet, the host/device divides data into small data blocks and sends them over several subsequent packets. And the packet index has the value from 0 to 255. The maximum data size that can be sent is calculated as the following.

$$\text{Max Data block} = 256 \times 65,507 = 16,769,792 \text{ [byte]}$$

Packet index is transferred by the parameter of the header. The following explains the format of packet index.

$$(\text{Packet index (0~N)} \ll 8) + (\text{Max Packet Index N})$$

For example, if single packet is sent, packet index is 0x0000.

If two packets are sent, the first packet index is 0x0001, and the second packet index is 0x0101.



# Serial Protocol

If three packets are sent, packet indexes are sequentially 0x0002, 0x0102, and 0x0202.

**Warning:** The total data size of multiple packets is dependent on the target devices.

The multi-packet is executed after last packet is transferred.

## ■ Error Code

If the host sends the command packet, the device returns the acknowledge packet with the packet error code. If Error code is not “ERR\_NONE”, the previously sent command packet is ignored in the device. The host needs to check the returned error code, and then retry or does something.

ERROR CODE LIST		
ERR_NONE	The command packet successfully executed	0x0
ERR_CHECKSUM_ERROR	There exists checksum error in header or data block.	0x2
ERR_INVALID_CMD	The command sent to the device is invalid.	0x5

## ■ How to Make the Header Checksum & the Data Checksum

Checksum data can be calculated by adding all byte data.

For example, in order to create the header checksum, 20 bytes from start byte to error code are added.

## Serial Protocol

### 3 COMMAND DESCRIPTION

---

This capture describes communication commands.

- Initialization

#### CMD\_REQUEST\_CONNECTION

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x01	Command	0x01
Param1	0	Param1	RESULT_SUCCEEDED
Param2	0	Param2	User Count
Data Size	0	Data Size	0
Error Code	0	Error Code	Error code

This command is easy and simple command for testing device operation and get information about the number of user.

# Serial Protocol

## CMD\_SET\_BAUDRATE

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x02	Command	0x02
Param1	0	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_PARAM RESULT_CANCELED
Param2	Baud rate (0 ~ 4) 0 – 115,200 bps 1 – 57,600 bps 2 – 38,400 bps 3 – 19,200 bps 4 – 9,600 bps 5 – 14,400 bps	Param2	Baud rate (0 ~ 4) 0 – 115,200 bps 1 – 57,600 bps 2 – 38,400 bps 3 – 19,200 bps 4 – 9,600 bps 5 – 14,400 bps
Data Size	0	Data Size	0
Error Code	0	Error Code	Error code

This command is only supported in FIM30 emulation mode.

This command changes the baudrate of module UART interface channel and save the speed.

If succeeded, the device returns acknowledge packet and changes baudate to new speed.

New baud rate is applied from the next packet.

## Serial Protocol

### CMD\_GET\_FIRMWARE\_VERSION2

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x04	Command	0x04
Param1	0	Param1	RESULT_SUCCEEDED RESULT_CANCELED
Param2	0	Param2	Version Information
Data Size	0	Data Size	0
Error Code	0	Error Code	Error code

This command requests the version of firmware in FIM modules.

The version information is BCD code and has the following format.

Data: 0x0000aabb

Version: aa.bb

# Serial Protocol

## CMD\_GET\_DEVICE\_INFO

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x05	Command	0x05
Param1	0	Param1	RESULT_SUCCEEDED RESULT_CANCELED
Param2	0	Param2	Device Name Refer to the following <b>Device type</b>
Data Size	0	Data Size	0
Error Code	0	Error Code	Error code

Device Type of acknowledge packet

0x5060 – FIM5060 module

Legacy Device type

0x00 – Reserved for old device

0x01 – Reserved for old device

0x02 – FIM10\_HV (EOL)

0x03 – FIM10\_LV (EOL)

0x04 – FIM01\_HV (EOL)

0x13 – FIM1030 (EOL)

0x33 – FIM2030

0x34 – FIM2040

0x3030 – FIM3030

0x3040 – FIM3040

0x3200 – FIM3200

# Serial Protocol

## • MATCHING

### CMD\_VERIFY\_FP

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x11	Command	0x11
Param1 <sup>1)</sup>	0 – FP verification 1 – Password	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_DATASIZE RESULT_INVALID_DATA RESULT_INVALID_ID RESULT_INVALID_PARAM RESULT_NOT_IN_TIME RESULT_CANCELED RESULT_EXTRACT_FAIL
Param2	(Packet Index (0~N) << 8) + (Max Packet Index N)	Param2	<b>IF</b> (Param1 == Succeeded)  <b>IF</b> (Command Param1 = 0) Template Index Number  <b>ELSE</b> 0  <b>ELSE</b> 0
Data Size	<b>IF</b> FP verification Size (a fraction of FPID)  <b>ELSE IF</b> password Size (a fraction of FPID + password)  <b>ELSE</b> 0	Data Size	0
Error Code	X	Error Code	Error Code
Data	<b>IF</b> (Param1 == 0) A fraction of FPID  <b>ELSE IF</b> (Param1 == 1) A fraction of FPID + password  <b>ELSE</b>	Data	-

## Serial Protocol

---

	-		
--	---	--	--

This command is used to verify user with user's ID.

Verification means finding user with user's ID, what we called, 1:1 matching.

There are two methods for verification – fingerprint and password.

According to methods, data has different structure. For fingerprint verification, user's ID is sent and for password verification, user's ID and password are sent in data.

# Serial Protocol

## CMD\_IDENTIFY\_FP

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x12	Command	0x12
Param1	0x00 – User ID only request 0x01 – User ID and Template index request 0x02 – User ID and user type request	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_PARAM RESULT_NOT_IN_TIME RESULT_IDENTIFY_TIMEOUT (FIM01 & FIM20xx only) RESULT_CANCELED RESULT_EXTRACT_FAIL
Param2	0	Param2	(Packet Index (0~N) << 8) + (Max Packet Index N)
Data Size	0	Data Size	IF (Param1 == Succeeded) IF (Command Param1 = 0x00) Size of FPID (various between devices) ELSE IF (Command Param1 = 0x01) Size of (FPID + Template Index) ELSE IF (Command Param1 = 0x02) Size of (FPID + User Type) ELSE 0 ELSE 0
Error Code	0	Error Code	Error Code
Data	-	Data	IF (Param1 == Succeeded) IF (Command Param1 = 0) FPID ELSE IF (Command Param1 = 1) (FPID + Template Index) ELSE 0 ELSE



## Serial Protocol

---

			0
--	--	--	---

This command is used to identify user.

Identification means finding user with unknown ID, what we called, 1:N matching.

If device find user ID having matched fingerprint, it returns user's ID.

According to the Param1 of command packet, auxiliary information such as template index or user type is also returned.

In 1 ID multi-template mode, User can be registered with a number of different finger in 1 ID.

Template index shows which template is matched in ID.

User type is user's privilege such as normal user or master user.

# Serial Protocol

## CMD\_IDENTIFY\_RID\_FP

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x13	Command	0x13
Param1	0x00 - User ID only request 0x01 - User ID and Template index Request 0x02 - User ID and User Type request	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_PARAM RESULT_NOT_IN_TIME RESULT_IDENTIFY_TIMEOUT (FIM01 & FIM20xx only) RESULT_CANCELED
Param2	(Packet Index (0~N) << 8) + (Max Packet Index N)	Param2	(Packet Index (0~N) << 8) + (Max Packet Index N)
Data Size	Size (FPID)	Data Size	IF (Param1 == Succeeded) IF (Command Param1 == 0x00) Size of FPID ELSE IF (Command Param1 == 0x01) Size of (FPID + Template Index) ELSE IF (Command Param1 == 0x02) Size of (FPID + User Type) ELSE 0 ELSE 0
Error Code	0	Error Code	Error Code
Data	A fraction of FPID	Data	IF (Param1 == Succeeded) IF (Command Param1 == 0x00 ) FPID ELSE IF (Command Param1 == 0x01) (FPID + Template Index) ELSE IF (Command Param1 == 0x02) (FPID + User Type) ELSE

## Serial Protocol

			0
			ELSE
			0

This command is only supported in FIM40/50 and FIM20 emulation mode.

This command is used to identify user with limited user's ID. It provides more information about user's ID and narrows the range of search. That results in fast response time.

If device find user ID having matched fingerprint, it returns user's ID.

According to the Param1 of command packet, auxiliary information such as template index or user type is also returned.

In 1 ID multi-template mode, User can be registered with a number of different finger in 1 ID.

Template index shows which template is matched in ID.

User type is user's privilege such as normal user or master user.

When you want to reduced ID matching, you must send ID having '\*' (0x2A) that matching one unknown digit.

For example, If you want to identify with ID starting with "12" and ID requires 4 digits, you must send reduced ID such as "12\*\*".

# Serial Protocol

## CMD\_INSTANT\_MATCHING

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x15	Command	0x15
Param1	<b>(Template Mode &lt;&lt; 8) + Aux Info</b> <b>Template Mode</b> 0 - single template 1 - Multi Template (FIM30 emulation mode Only) <b>Aux Info (FIM30 emulation mode only)</b> In Template Mode == 0 (single-template) 0 – Nitgen template 1 - FDA01 compatible style 2 – ISO 19794-2 template 3 – ANSI 378 template In Template Mode == 1 (Multi-template) 1~10 – the number of templates (NITGEN Format Only)	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_DATASIZE RESULT_INVALID_DATA RESULT_NOT_IN_TIME RESULT_INVALID_PARAM RESULT_TOO_LARGE_DATA RESULT_CANCELED RESULT_EXTRACT_FAIL
Param2	(Packet Index (0~N) << 8) + (Max Packet Index N)	Param2	0
Data Size	Size (A fraction of TEMPLATE_INFO)	Data Size	0
Error Code	0	Error Code	Error Code
Data	A fraction of TEMPLATE_INFO	Data	-

The template of FIM40/50 and FIM20 include format information of template in itself.

By using this command, host can verify fingerprint with template data saved in host.

The template data can be obtained form CMD\_GET\_TEMPLATE command.

# Serial Protocol

## CMD\_GET\_TEMPLATE

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x16	Command	0x16
Param1	0 - Default 2 – ISO 19794-2 Format 3 – ANSI 378 Format	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_NOT_IN_TIME RESULT_CANCELED RESULT_EXTRACT_FAIL
Param2	0	Param2	(Packet Index (0~N) << 8) + (Max Packet Index N)
Data Size	0	Data Size	<b>IF</b> (Param1 == Succeeded) Size (A fraction of TEMPLATE_INFO) <b>ELSE</b> 0
Error Code	0	Error Code	Error Code
Data	-	Data	<b>IF</b> (Param1 == Succeeded) A fraction of TEMPLATE_INFO <b>ELSE</b> -

The structure of TEMPLATE\_INFO is explained in Appendix D.

This command returns template data of fingerprint. According to Param1 of command packet, ISO or ANSI format can be returned.

Template data is used for CMD\_INSTANT\_MATCHING.

## Serial Protocol

### **CMD\_CANCEL**

COMMAND PACKET		ACKNOWLEDGEMENT PACKET <sup>1)</sup>	
Command	0x17	Command	0x17
Param1	0	Param1	RESULT_IDLE_STATUS
Param2	0	Param2	0
Data Size	0	Data Size	0
Error Code	0	Error Code	Error Code

If the device received the CMD\_CANCEL command packet in idle, the device returns the CMD\_CANCEL acknowledge packet with RETURN\_CANCEL in paramter1. Otherwise, it stops the currently executing command and returns acknowledge packet with RESULT\_CANCELED.

# Serial Protocol

## CMD\_INSTANT\_VERIFY

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x18	Command	0x18
Param1	0	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_DATASIZE RESULT_INVALID_DATA RESULT_NOT_IN_TIME RESULT_INVALID_PARAM RESULT_INVALID_ID RESULT_TOO_LARGE_DATA RESULT_CANCELED
Param2	(Packet Index (0~N) << 8) + (Max Packet Index N)	Param2	<b>IF</b> (Param1 == Succeeded) Template Index Number <b>ELSE</b> 0
Data Size	Size (A fraction of FPID + Template)	Data Size	0
Error Code	0	Error Code	Error Code
Data	A fraction of FPID + Template	Data	-

The structure of data is explained in Appendix D.

This command is only supported in FIM40/50 and FIM20 emulation mode.

This command verifies template data using user's ID. It is similar to verification except that template data form host instead of sensor.

# Serial Protocol

## CMD\_INSTANT\_IDENTIFY

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x19	Command	0x19
Param1	0 – User ID only request 1 – User ID and Template index request (FIM40/50 & FIM20 emulation only) 2 –User ID and User Type request (FIM 40/50 only)	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_DATASIZE RESULT_INVALID_DATA RESULT_NOT_IN_TIME RESULT_IDENTIFY_TIMEOUT RESULT_INVALID_PARAM RESULT_TOO_LARGE_DATA RESULT_CANCELED
Param2	(Packet Index (0~N) << 8) + (Max Packet Index N)	Param2	X
Data Size	Size (A fraction of Template)	Data Size	<b>IF</b> (Param1 == Succeeded) <b>IF</b> (Command Param1 = 0) Size of FPID (various between devices) <b>ELSE IF</b> (Command Param1 = 1) Size of (FPID + Template Index) <b>ELSE</b> 0 <b>ELSE</b> 0
Error Code	0	Error Code	Error Code
Data	A fraction of Template	Data	<b>IF</b> (Param1 == Succeeded) <b>IF</b> (Command Param1 = 0) FPID <b>ELSE IF</b> (Command Param1 = 1) (FPID + Template Index) <b>ELSE</b> 0 <b>ELSE</b> 0



## Serial Protocol

---

The structure of data is explained in Appendix D.

This command is only supported in FIM40/50 and FIM20 emulation mode.

This command identifies template data. It is similar to verification except that template data form host instead of sensor.

## Serial Protocol

### **CMD\_AUTO\_IDENTIFY**

COMMAND PACKET		ACKNOWLEDGEMENT PACKET <sup>1)</sup>	
Command	0x1A	Command	0x1A
Param1	Selection of Operation 0x00 – Stop Auto Identification 0x01 – Start Auto Identification	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_PARAM
Param2	0	Param2	0
Data Size	0	Data Size	0
Error Code	0	Error Code	Error Code

By using this command, you can change the mode of module to auto-identify mode.

In auto-identify mode, if user places finger on sensor, module returns result CMD\_AUTO\_IDENTIFY\_RESULT acknowledge packet. And module ignores other command except CMD\_AUTO\_IDENTIFY.

## Serial Protocol

### CMD\_AUTO\_IDENTIFY\_RESULT

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command		Command	0x1B
Param1		Param1	RESULT_SUCCEEDED RESULT_FAILED
Param2		Param2	0
Data Size		Data Size	IF (Param1 == Succeeded) Size of FPID (various between devices) ELSE 0
Error Code		Error Code	Error Code
Data		Data	IF (Param1 == Succeeded) FPID ELSE 0

This command is used only for acknowledge packet. In auto-identify mode, module returns the result of identification using this packet.

# Serial Protocol

## • DATABASE MANAGEMENT

### CMD\_DELETE\_FP

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x22	Command	0x22
Param1	0	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_DATASIZE RESULT_INVALID_DATA REAULT_INVALID_ID RESULT_NOT_MASTER_MODE
Param2	(Packet Index (0~N) << 8) + (Max Packet Index N)	Param2	<b>IF</b> (Param1 == Succeeded) Registered User Count <b>ELSE</b> 0
Data Size	Size (A fraction of FPID)	Data Size	0
Error Code	0	Error Code	Error Code
Data	A fraction of FPID	Data	-

This command is only executed in master mode.

This command deletes user. If succeeded, currently remained user count is returned in Param2.

# Serial Protocol

## CMD\_DELETE\_ALL\_FP

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x23	Command	0x23
Param1	0 – Delete all user 1 – Delete all user (except Master) 2 – Delete all Master	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_PARAM RESULT_NOT_MASTER_MODE RESULT_CANCELED
Param2	0	Param2	0
Data Size	0	Data Size	0
Error Code	0	Error Code	Error Code

This command is only executed in master mode.

This command deletes users.

According to the Param1 of command packet, normal user or master users are deleted selectively.

# Serial Protocol

## CMD\_SET\_MASTER

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x24	Command	0x24
Param1	0 – Clear Master Flag 1 – Set Master Flag	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_DATASIZE RESULT_INVALID_DATA RESULT_INVALID_PARAM RESULT_INVALID_ID RESULT_NOT_MASTER_MODE RESULT_EXCEEDED_MASTER_CNT
Param2	(Packet Index (0~N) << 8) + (Max Packet Index N)	Param2	Master Count
Data Size	Size (A fraction of FPID)	Data Size	0
Error Code	0	Error Code	Error Code
Data	A fraction of FPID	Data	-

This command is only executed in master mode.

This command changes the privilege of user. Host can changes normal user to master or reversely.

## Serial Protocol

---

### CMD\_LEAVE\_MASTER\_MODE

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x26	Command	0x26
Param1	0	Param1	RESULT_SUCCEEDED RESULT_NOT_MASTER_MODE
Param2	0	Param2	0
Data Size	0	Data Size	0
Error Code	0	Error Code	Error code

In FIM module, there are two operating mode – normal mode and master mode.

Host can request authentication such as verification, identification and so on in normal mode. But for changing system option and managing users, host need to enter the master operation mode.

This command is used to leave master operation mode. To enter master operation mode, use CMD\_ENTER\_MASTER\_MODE2 command.

# Serial Protocol

## CMD\_SET\_MASTER\_PASSWORD

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x27	Command	0x27
Param1	0	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_DATASIZE RESULT_INVALID_DATA RESULT_NOT_MASTER_MODE
Param2	(Packet Index (0~N) << 8) + (Max Packet Index N)	Param2	0
Data Size	Size (A fraction of Password)	Data Size	0
Error Code	0	Error Code	Error code
Data	A fraction of Password	Data	-

This command is only executed in master mode.

FIM provides board password. This board password provides another method to enter master operating mode.

In default system, this password is empty.



## Serial Protocol

### CMD\_READ\_USER\_DATA

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x2B	Command	0x2B
Param1	Address	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_PARAM RESULT_NOT_MASTER_MODE
Param2	User data length (byte)	Param2	IF (Param1 == RESULT_SUCCEEDED) User data length (byte)  ELSE 0
Data Size	0	Data Size	IF (Param1 == RESULT_SUCCEEDED) User data length (byte)  ELSE 0
Error Code	0	Error Code	Error code
Data	-	Data	IF (Param1 == RESULT_SUCCEEDED) User data  ELSE -

This command is only executed in master mode.

Host can read data in custom user area.

Custom user area is 64 Kbytes. The range of address is from 0x00000000 to 0x00010000.

## Serial Protocol

### CMD\_WRITE\_USER\_DATA

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x2C	Command	0x2C
Param1	Address	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_PARAM RESULT_NOT_MASTER_MODE
Param2	User data length (byte)	Param2	0
Data Size	User data length	Data Size	0
Error Code	0	Error Code	Error code
Data	User Data	Data	-

This command is only executed in master mode.

Custom user area is 64 Kbytes. The range of address is from 0x00000000 to 0x00010000.

## Serial Protocol

---

### CMD\_ERASE\_USER\_DATA\_BLOCK

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x2D	Command	0x2D
Param1	0	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_NOT_MASTER_MODE
Param2	0	Param2	0
Data Size	0	Data Size	0
Error Code	0	Error Code	Error code

This command is only executed in master mode.

This command erases custom user area. The data in custom area is initialized to 0xFF.

## Serial Protocol

### **CMD\_DELETE\_MASTER\_PASSWORD**

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x2E	Command	0x2E
Param1	0	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_NOT_MASTER_MODE
Param2	0	Param2	0
Data Size	0	Data Size	0
Error Code	0	Error Code	Error code

This command is only executed in master mode.

This command deletes board password set by CMD\_SET\_MASTER\_PASSWORD command.

# Serial Protocol

## CMD\_ENTER\_MASTER\_MODE2

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x2F	Command	0x2F
Param1	<u>Master authentication type</u> Master FP verification = 0 Master password verification = 1 Board password verification = 2 Null = 3 Master instant FP verification = 4	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_DATASIZE RESULT_INVALID_DATA RESULT_NOT_IN_TIME RESULT_INVALID_PARAM RESULT_INVALID_ID RESULT_CANCELED RESULT_EXTRACT_FAIL
Param2	(Packet Index (0~N) << 8) + (Max Packet Index N)	Param2	<u>Master authentication type</u> Master FP = 0 Master password = 1 FDA board password = 2 Null = 3 Master FP from host = 4 Master FP from host (FDA01 style) = 5 (FIM10 only)
Data Size	<b>IF</b> Master FP Size (A fraction of FPID) <b>ELSE IF</b> master password Size (A fraction of FPID + Password) <b>ELSE IF</b> device board password Size (A fraction of password) <b>ELSE IF</b> Master FP from host Size (A fraction of FPID + Template) <b>ELSE IF</b> null 0	Data Size	0
Error Code	X	Error Code	Error code
Data	<b>IF</b> Master FP A fraction of FPID	Data	-

## Serial Protocol

	<b>ELSE IF</b> master password A fraction of FPID + Password <b>ELSE IF</b> device board password A fraction of Password <b>ELSE IF</b> Master FP from host FPID + Template <b>ELSE IF</b> null -		
--	--	--	--

This command is used to enter master operation mode.

If one of the followings is satisfied, device requires master authentication.

1. User having master privilege exists
2. Board password is not null string.

Otherwise, host can enter master mode without authentication by setting Param1 of command packet to 3.

To enter master operation mode, host need to be authenticated. There are four methods.

1. Master user Fingerprint verification
2. Master user password verification
3. Board password verification
4. Master user instant verification

# Serial Protocol

## CMD\_GET\_FP\_LIST2

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x30	Command	0x30
Param1	List data selection 0 = User count, ID list 1 = User count	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_NOT_MASTER_MODE RESULT_INVALID_PARAM RESULT_CANCELED
Param2	Packet Index (0~N)	Param2	IF (Param1 == RESULT_SUCCEEDED) (Packet Index (0~N) << 8) + (Max Packet Index N) <b>ELSE</b> -
Data Size	0	Data Size	IF (Param1 == RESULT_SUCCEEDED) Size of (a piece of FP list block) <b>ELSE</b> 0
Error Code	0	Error Code	Error code
Data	-	Data	A piece of FP list block

This command is only executed in master mode.

This command is used to get user list including masters.

The user list block may be different according to devices. Before using CMD\_GET\_FP\_LIST2 packet, check device information using CMD\_GET\_DEVICE\_INFO

User list block consists of the number of user, the size of FPID, and user list.

If **Param1** of command packet is '0', device returns the list of users.

If **Param1** of command packet is '1', device returns only the number of users.

The structure of data is explained in Appendix D.

# Serial Protocol

## CMD\_GET\_MASTER\_LIST2

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x31	Command	0x31
Param1	List data selection 0 = Master count, ID list 1 = Master count	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_NOT_MASTER_MODE RESULT_INVALID_PARAM RESULT_CANCELED
Param2	Packet index (0~N)	Param2	IF (Param1 == RESULT_SUCCEEDED) (Packet Index (0~N) << 8) + (Max Packet Index N) <b>ELSE</b> -
Data Size	0	Data Size	IF (Param1 == RESULT_SUCCEEDED) Size of (a piece of master list block) <b>ELSE</b> 0
Error Code	0	Error Code	Error code
Data	-	Data	A piece of master list block

This command is only executed in master mode.

This command is used to get master list except normal users.

The Master list block may be different according to devices. Before using CMD\_GET\_MASTER\_LIST2 packet, check device information using CMD\_GET\_DEVICE\_INFO

Master list block consists of the number of master, the size of FPID, and master list.

If **Param1** of command packet is '0', device returns the list of masters.

If **Param1** of command packet is '1', device returns only the number of masters.

The structure of data is explained in Appendix D.



# Serial Protocol

## CMD\_READ\_LOG\_DATA2

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x32	Command	0x32
Param1	Log request mode 0 = <b>Param2</b> previous log read 1 = oldest unread log 2 = last written log 3 = All log 4 = from oldest unread to last	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_NOT_MASTER_MODE RESULT_INVALID_PARAM RESULT_CANCELED
Param2	<b>IF Param1 == 0</b> Nth log <b>ELSE IF Param1 == 3</b> Index(0~N) <b>ELSE IF Param1 == 4</b> Index (0~N) <b>ELSE</b> 0	Param2	<b>IF</b> (Param1 == RESULT_SUCCEEDED) (Packet Index (0~N) << 8) + (Max Packet Index N) <b>ELSE</b> -
Data Size	0	Data Size	<b>IF</b> (Param1 == RESULT_SUCCEEDED) Size of a piece of Log data block <b>ELSE</b> 0
Error Code	0	Error Code	Error code
Data	-	Data	<b>IF</b> (Param1 == RESULT_SUCCEEDED) Size of a piece of Log data block <b>ELSE</b> 0

This command is only executed in master mode.

Log data block consists of the number of returned log, the size of log, and log data

The log data block may be different according to emulation modes. Before using CMD\_READ\_LOG2 packet, check device information using CMD\_GET\_SYSINFO

# Serial Protocol

## CMD\_REGISTER\_FP

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x33	Command	0x33
Param1	0 – User 1 – Master Otherwise – Reserved	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_DATASIZE RESULT_INVALID_DATA RESULT_NOT_IN_TIME RESULT_INVALID_PARAM RESULT_USED_ID RESULT_DB_IS_FULL RESULT_NOT_MASTER_MODE RESULT_ANOTHER_FINGER RESULT_CANCELED RESULT_EXTRACT_FAIL RESULT_INVALID_SEQUENCE
Param2	Packet index 0x00 – Extract 1 <sup>st</sup> Template from sensor with ID and password 0x10 – Extract 1 <sup>st</sup> Template from sensor with auto-generated ID 0x01 – Extract 2 <sup>nd</sup> Template from sensor & Save 0x02 – Extract 2 <sup>nd</sup> Template from sensor & Save with different finger (FIM20 emulation mode Only) 0x03 – Extract 3 <sup>rd</sup> Template from sensor 0x04 – Extract 4 <sup>th</sup> Template from sensor & save 0x05 – Extract 4 <sup>th</sup> Template from sensor & save with different finger	Param2	<b>IF</b> (Param1 == RESULT_SUCCEEDED) && ( ((Packet index == 0x01 or 0x02) && (2 templates mode))    ((Packet index == 0x11 or 0x12) && (4 templates mode)) ) Registered User Count (Only valid if succeed) <b>ELSE</b> 0
Data Size	<b>IF</b> (Packet index == 0) Size of (FPID + Password)	Data Size	0

## Serial Protocol

	<b>ELSE</b> 0		
Error Code	0	Error Code	Error Code
Data	<b>IF</b> (Packet index == 0) FPID + password <b>ELSE</b> 0	Data	-

This command is only executed in master mode.

User's privilege is set in **Param1** of command packet.

The DB of user is saved only after executing command packet with Param2 value such as 0x01 or 0x02 in 2 templates mode, or 0x04 or 0x05 in 4 templates mode.

FIM provides auto-generated ID function.

Auto-generated ID is 4 digits from "0000" to "9999". It makes ID from "0000".

FIM provides different finger mode.

FIM requires two fingerprint images for ensuring divergence of image.

In non-different finger mode, device checks whether input fingers are same finger or not.

If not, device returns fail.

In different finger mode, device registers user without checking. Therefore, host can register user with 2 different fingers in 2 templates mode or 4 fingers in 4 templates mode.

For registration, host process two stages in 2 templates mode.

Firstly send CMD\_REGISTER\_FP with 0x00 (send ID) or 0x10 (auto-generated ID) in **Param2**.

Secondly send CMD\_REGISER\_FP with 0x01 (same fingers) or 0x02 (different fingers) in **Param2**.

The 4 templates mode is supported in FIM20 emulation mode.

For using FIM40/50 more efficiently, use CMD\_REGISTER\_MULTI\_FP command instead of this command. This command is provided for compatibility of old modules such as FIM20 and FIM30.

# Serial Protocol

## CMD\_CHANGE\_FP

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x34	Command	0x34
Param1	0x01 – Change Master Privilege & Save 0x02 – Change Password & Save 0x03 – Change User security Level & Save Others – reserved	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_DATASIZE RESULT_INVALID_DATA RESULT_NOT_IN_TIME RESULT_INVALID_PARAM RESULT_INVALID_ID RESULT_NOT_MASTER_MODE RESULT_CANCELED RESULT_EXTRACT_FAIL
Param2	<b>IF</b> (Param1 == 0x01) 0 – set to normal user 1 – set to master <b>ELSE IF</b> (Param1 == 0x03) Security Level <b>ELSE</b> Reserved	Param2	0
Data Size	<b>IF</b> (Param1 == 0x01) Size of (FPID) <b>ELSE IF</b> (Param2 == 0x02) Size of (FPID + Password) <b>ELSE IF</b> (Param3 == 0x03) Size of(FPID) <b>ELSE</b> 0	Data Size	0
Error Code	0	Error Code	Error Code
Data	<b>IF</b> (Param1 == 0x01) FPID <b>ELSE IF</b> (Param1 == 0x02) FPID + password	Data	-

## Serial Protocol

	<b>ELSE IF</b> (Param3 == 0x03)  FPID  <b>ELSE</b>  0		
--	---	--	--

This command is only executed in master mode.

The structure of data is explained in Appendix D.

This command is only supported in FIM40/50 and FIM20 emulation mode.

Host can change user's privilege, password and verification security level. When user is registered, verification security is not set and system verification security level is applied. By changing user's security level, host can verify user with individual security level.

# Serial Protocol

## CMD\_ADD\_FP

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x35	Command	0x35
Param1	DB structure version <b>For FIM30 emulation mode</b> 0x01 - FIM_OLD_DB_FIM30 data structure 0x11 – FIM_DB_FIM30 data structure (NITGEN, ISO 19794-2 and ANSI 378) <b>For FIM40&amp;50 and</b> <b>FIM20 emulation mode</b> 0x01 – FIM_OLD_DB2 data structure 0x02 – FIM_OLD_DB4 data structure 0x11 – FIM_DB2 data structure (NITGEN, ISO 19794-2 and ANSI 378) 0x12 – FIM_DB4 data structure (NITGEN, ISO 19794-2 and ANSI 378) <b>For FIM40&amp;50 Only</b> 0x20 – FIM_MT_DB data structure (NITGEN, ISO 19794-2 and ANSI 378) Others – reserved	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_DATASIZE RESULT_INVALID_DATA RESULT_INVALID_PARAM RESULT_USED_ID RESULT_DB_IS_FULL RESULT_NOT_MASTER_MODE RESULT_CANCELED RESULT_WRONG_TEMP_MODE
Param2	(Packet index (0~N) << 8) + (Max Packet Index N)	Param2	0
Data Size	Size (a piece of DB structure)	Data Size	0
Error Code	0	Error Code	Error Code
Data	A piece of DB structure	Data	-

This command is only executed in master mode.

The structure of data is explained in Appendix D.

## Serial Protocol

---

This command is used for adding new user's DB to device.

Caution is needed because different data structures are used according to emulation mode (FIM30, FIM20 and none emulation mode) and template format (NITGEN, ISO and ANSI).

User's DB can be obtained by using CMD\_GET\_FP command.

# Serial Protocol

## CMD\_GET\_FP

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x36	Command	0x36
Param1	Get operation 0 – FPID DB 1 – First DB 2 – Next DB Others – reserved	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_DATASIZE RESULT_INVALID_DATA RESULT_INVALID_PARAM RESULT_INVALID_ID RESULT_NOT_MASTER_MODE RESULT_CANCELED
Param2	DB structure version <b>For FIM30 Emulation mode</b> 0x01 – FIM_OLD_DB_FIM30 data structure (Template: NITGEN format) 0x11 – FIM_DB_FIM30 data structure (Template: ISO 19794-2 format) 0x13 – FIM_DB_FIM30 data structure (Template: ANSI 378 format) <b>For FIM40/50 and FIM20 Emulation Mode</b> 0x01 – FIM_OLD_DB2 data structure 0x02 – FIM_OLD_DB4 data structure 0x11 – FIM_DB2 data structure (Template:-NITGEN format) 0x12 – FIM_DB4 data structure (Template:-NITGEN format) 0x13 – FIM_DB2 data structure (Template: ISO 19794-2 format) 0x14 – FIM_DB4 data structure (Template: ISO 19794-2 format)	Param2	0



## Serial Protocol

	0x15 – FIM_DB2 data structure (Template: ANSI 378 format) 0x16 – FIM_DB4 data structure (Template: ANSI 378 format) <b>For FIM40&amp;50 Only</b> 0x20 – FIM_MT_DB data structure (Template: NITGEN format) 0x21 – FIM_MT_DB data structure (Template: ISO 19794-2 format) 0x22 – FIM_MT_DB data structure (Template: ANSI 378 format) Others – reserved		
Data Size	<b>IF</b> (Param1 == 0) Size of FPID <b>ELSE</b> 0	Data Size	<b>IF</b> (Param1 == RESULT_SUCCEEDED) Size of DB structure <b>ELSE</b> 0
Error Code	0	Error Code	Error Code
Data	<b>IF</b> (Param1 == 0) FPID <b>ELSE</b> -	Data	<b>IF</b> (Param1 == RESULT_SUCCEEDED) DB structure <b>ELSE</b> 0

This command is only executed in master mode.

The structure of DB is explained in Appendix D.

Caution is needed because different data structures are used according to emulation mode (FIM30, FIM20 and none emulation mode) and template format (NITGEN, ISO and ANSI).

User's DB can be added by using CMD\_ADD\_FP command.

There are two ways to get users' DB.

The first way is the request with user ID (Param1 == 0). Device returns user's DB having matching ID. If not find, return error.

The second way is request with sequential order using Param1 of command packet.

In first request, get first DB (Param1 == 1).

## Serial Protocol

---

And get next DB (Param1 == 2) until error is returned.

## Serial Protocol

---

### CMD\_DELETE\_ALL\_LOG

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x37	Command	0x37
Param1	0	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_NOT_MASTER_MODE RESULT_CANCELED
Param2	0	Param2	0
Data Size	0	Data Size	0
Error Code	0	Error Code	Error Code

This command is only executed in master mode.

This command deletes all log data. Once deleted, there is no way to recover.

# Serial Protocol

## CMD\_REGISTER\_MULTI\_FP

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x38	Command	0x38
Param1	0 – User 1 – Master Otherwise – Reserved	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_DATASIZE RESULT_INVALID_DATA RESULT_NOT_IN_TIME RESULT_INVALID_PARAM RESULT_USED_ID RESULT_DB_IS_FULL RESULT_NOT_MASTER_MODE RESULT_ANOTHER_FINGER RESULT_CANCELED RESULT_EXTRACT_FAIL RESULT_INVALID_SEQUENCE
Param2	Packet index Refer to the following <b>define (1)</b>	Param2	<b>IF</b> ((Param1 == RESULT_SUCCEEDED) && (Capture Mode == 0x03 or 0x04)) Registered User count <b>ELSE</b> 0
Data Size	<b>IF</b> ( (Finger Index == 0) && (Capture Mode == 0) ) Size of (FPID + Password) <b>ELSE</b> 0	Data Size	0
Error Code	0	Error Code	Error Code
Data	<b>IF</b> ( (Finger Index == 0) && (Capture Mode == 0) ) FPID + password <b>ELSE</b> 0	Data	-

## Serial Protocol

This command is only executed in master mode.

This command is only supported in FIM40/50.

In FIM40/50, 1 ID multi-template function is newly added. For supporting this function, this command is added. When user is added, host can insert up to 10 fingers for one user and save.

### Define (1) – Packet index

Bit								
Order	31 ~ 28	27 ~ 24	23 ~ 20	19 ~ 16	15 ~ 12	11 ~ 8	7 ~ 4	3 ~ 0
							Finger Index	Capture mode

Finger Index: 0x0 ~ 0x9

The index of finger

Capture Mode: 0x0 – First Capture with User ID and Password

0x1 – First Capture with Auto ID

0x2 – Second Capture and Continue

0x3 – Second Capture and Save

0x4 – Save

Auto ID is generated to 4-digits.

### Packet Index Sample

Save 1 template for User with User ID and Password

0x00 -> 0x03

Save 2 templates for User with User ID and Password

0x00 -> 0x02 -> 0x10 -> 0x13

Save 1 template for User with Auto-Generated ID

0x01 -> 0x03

# Serial Protocol

## • CONFIGURATION

### CMD\_SET\_OPP\_OPTION

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x40	Command	0x40
Param1	0	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_PARAM RESULT_NOT_MASTER_MODE
Param2	Option value bit[23..16] = gain bit[15..8] = brightness bit[7..0] = contrast	Param2	Option value bit[23..16] = gain bit[15..8] = brightness bit[7..0] = contrast
Data Size	0	Data Size	0
Error Code	0	Error Code	Error code

This command is only executed in master mode.

This command is only supported in FIM30 emulation mode.

#### Default Value

Gain = 2

Brightness = 45

Contrast = 20

# Serial Protocol

## CMD\_GET\_OPP\_OPTION

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x41	Command	0x41
Param1	0	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_NOT_MASTER_MODE
Param2	0	Param2	<b>IF</b> (Param1 == RESULT_SUCCEEDED) Option value bit[23..16] = gain bit[15..8] = brightness bit[7..0] = contrast <b>ELSE</b> -
Data Size	0	Data Size	0
Error Code	0	Error Code	Error code

This command is only executed in master mode.

This command is only supported in FIM30 emulation mode.

## Serial Protocol

### **CMD\_SET\_SECURITY\_LEVEL**

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x42	Command	0x42
Param1	0	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_PARAM RESULT_NOT_MASTER_MODE
Param2	Verification security level bit[31..16] Identification security level Bit[15..0]	Param2	Verification security level bit[31..16] Identification security level Bit[15..0]
Data Size	0	Data Size	0
Error Code	0	Error Code	Error code

This command is only executed in master mode.

This command is only supported in FIM30 emulation mode.

#### Default Value

Verification security level: 8

Identification security level: 5



# Serial Protocol

## CMD\_GET\_SECURITY\_LEVEL

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x43	Command	0x43
Param1	0	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_NOT_MASTER_MODE
Param2	0	Param2	IF (Param1 == RESULT_SUCCEEDED) Verification security level bit[31..16] Identification security level bit[15..0] <b>ELSE</b> -
Data Size	0	Data Size	0
Error Code	0	Error Code	Error code

This command is only executed in master mode.

This command is only supported in FIM30 emulation mode.

## Serial Protocol

### CMD\_SET\_CAPTURE\_OPTION

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x44	Command	0x44
Param1	0x00 -> Latent & Adaptive 0x01 -> Latent 0x02 -> Adaptive 0x08 -> Max number of Capture in Adaptive	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_FAILED RESULT_INVALID_PARAM RESULT_NOT_MASTER_MODE
Param2	<b>IF</b> (Param1 == 0x0 or 0x01 or 0x02) Turn Off = 0 / Turn On = 1 <b>Else IF</b> (Param1 == 0x08) Max number of capture in Adaptive <b>ELSE</b> -	Param2	<b>IF</b> (Command Param1 == 0x00 or 0x01 or 0x02) Turn Off = 0 / Turn On = 1 <b>ELSE IF</b> (Command Param1 == 0x08) Max number of Capture in Adaptive <b>ELSE</b> -
Data Size	0	Data Size	0
Error Code	0	Error Code	Error code

This command is only executed in master mode.

This command is only supported in FIM30 emulation mode.

The latent option does not return error, but does not used.

The range of Max Number of Capture is from 0 to 255.

## Serial Protocol

### CMD\_GET\_CAPTURE\_OPTION

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x45	Command	0x45
Param1	0x00 -> Reserved 0x01 -> Latent 0x02 -> Adaptive 0x08 -> Max number of Capture in Adaptive	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_NOT_MASTER_MODE
Param2	0	Param2	<b>IF</b> (Param1 == RESULT_SUCCEEDED) <b>IF</b> (Command Param1 == 0x00 or 0x01 or 0x02) Turn Off = 0 / Turn On = 1 <b>ELSE IF</b> (Command Param1==0x08) Max number of capture in adaptive <b>ELSE</b> - <b>ELSE</b> -
Data Size	0	Data Size	0
Error Code	0	Error Code	Error code

This command is only executed in master mode.

This command is only supported in FIM30 emulation mode.

The latent option does not return error, but does not used.

## Serial Protocol

### **CMD\_SET\_LOG\_OPTION**

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x48	Command	0x48
Param1	0	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_PARAM RESULT_NOT_MASTER_MODE
Param2	Disable = 0 / Enable = 1	Param2	Disable = 0 / Enable = 1
Data Size	0	Data Size	0
Error Code	0	Error Code	Error code

This command is only executed in master mode.

This command is only supported in FIM30 emulation mode.

Default Value

Disable

# Serial Protocol

## CMD\_GET\_LOG\_OPTION

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x49	Command	0x49
Param1	0	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_NOT_MASTER_MODE
Param2	0	Param2	IF (Param1 == RESULT_SUCCEEDED) Disable = 0 / Enable = 1  ELSE -
Data Size	0	Data Size	0
Error Code	0	Error Code	Error code

This command is only executed in master mode.

This command is only supported in FIM30 emulation mode.

## Serial Protocol

---

### CMD\_SET\_CAPTURE\_PERIOD

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x4A	Command	0x4A
Param1	0	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_PARAM RESULT_NOT_MASTER_MODE
Param2	Capture period	Param2	Capture period (1 ~ 255)
Data Size	0	Data Size	0
Error Code	0	Error Code	Error code

This command is only executed in master mode.

This command is only supported in FIM30 emulation mode.

Capture period means 100 ms tick count.

When capturing fingerprint, device try to capture until tick count reaches to capture period time.

## Serial Protocol

### CMD\_GET\_CAPTURE\_PERIOD

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x4B	Command	0x4B
Param1	0	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_NOT_MASTER_MODE
Param2	0	Param2	IF (Param1 == RESULT_SUCCEEDED) Capture period (1 ~ 255)  ELSE -
Data Size	0	Data Size	0
Error Code	0	Error Code	Error code

This command is only executed in master mode.

This command is only supported in FIM30 emulation mode.

## Serial Protocol

### **CMD\_SET\_SYSINFO**

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x4C	Command	0x4C
Param1	<b>SI_Type</b>	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_PARAM RESULT_NOT_MASTER_MODE RESULT_DB_ISNOT_EMPTY
Param2	SI_Value	Param2	0
Data Size	0	Data Size	<b>IF</b> (Param1 == Succeeded) Size (SI_INFO) <b>ELSE</b> 0
Error Code	0	Error Code	Error code
	-		<b>IF</b> (Param1 == Succeeded) SI_INFO <b>ELSE</b> -

This command is only executed in master mode.

**SI\_TYPE** and **SI\_INFO** are defined in Appendix D.

Caution: Option value is changed temporary by this command. After power off, this value is changed to the previous value. If you want to keep new option value, send CMD\_SAVE\_SYSINFO command after changing option value.



# Serial Protocol

## CMD\_GET\_SYSINFO

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x4D	Command	0x4D
Param1	<b>SI_Type</b>	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_PARAM RESULT_NOT_MASTER_MODE
Param2	X	Param2	SI_Value
Data Size	0	Data Size	<b>IF</b> (Param1 == Succeeded) Size (SI_INFO) <b>ELSE</b> 0
Error Code	X	Error Code	Error code
Data	-	Data	<b>IF</b> (Param1 == Succeeded) SI_INFO <b>ELSE</b> -

**SI\_TYPE** and **SI\_INFO** are defined in Appendix D.

## Serial Protocol

### **CMD\_SAVE\_SYSINFO**

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x4E	Command	0x4E
Param1	0	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_NOT_MASTER_MODE
Param2	0	Param2	0
Data Size	0	Data Size	0
Error Code	0	Error Code	Error code

This command is only executed in master mode.

This command is used to save current system option to non-volatile memory. After executing this command, system options are restored when power-on.

## Serial Protocol

### CMD\_CHG\_NUM\_OF\_TEMP

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x4F	Command	0x4F
Param1	Number of Template (2 or 4)	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_NOT_MASTER_MODE RESULT_INVALID_PARAM RESULT_DB_ISNOT_EMPTY
Param2	X	Param2	X
Data Size	0	Data Size	0
Error Code	X	Error Code	Error code

This command is only executed in master mode.

This command is supported in FIM20 emulation mode

This command is used to change template mode (2 templates or 4 templates). When executing this command, there must be no user in device. If not, error is returned.

After executing this command, option is saved by force because non-saved option causes problem in next power-on.

## Serial Protocol

### **CMD\_SET\_DEFAULT\_SYSINFO**

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x50	Command	0x50
Param1	0	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_NOT_MASTER_MODE
Param2	0	Param2	0
Data Size	0	Data Size	0
Error Code	0	Error Code	Error code
	-		-

This command is only executed in master mode.

This command changes all options to default value except the following options.

SI\_NUM\_OF\_TEMP  
 SI\_CHANNEL0\_BAUDRATE  
 SI\_CHANNEL1\_BAUDRATE  
 SI\_LENGTH\_OF\_USER\_ID  
 SI\_EMULATION\_MODE

Caution) This command does not save the changed option value. So to save options, use CMD\_SAVE\_SYSINFO.

# Serial Protocol

## CMD\_CHG\_EMULMODE

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x51	Command	0x51
Param1	0x01 – FIM20 emulation mode 0x02 – FIM30 emulation mode 0xFF – None emulation mode Otherwise - Reserved	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_NOT_MASTER_MODE
Param2	0	Param2	0x01 – FIM20 emulation mode 0x02 – FIM30 emulation mode 0xFF – None emulation mode Otherwise - reserved
Data Size	0	Data Size	0
Error Code	0	Error Code	Error code
	-		-

This command is only executed in master mode.

When this command is executed, length of user ID (system option) is also changed because FIM20 or FIM30 mode has fixed length of user ID.

After executing this command, option is saved by force because non-saved option causes problem in next power-on

The default system mode is none-emulation mode.

## Serial Protocol

### **CMD\_CHG\_LENGTH\_OF\_USERID**

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x52	Command	0x52
Param1	Length of User ID (4~15)	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_NOT_MASTER_MODE
Param2	0	Param2	Length of User ID
Data Size	0	Data Size	0
Error Code	0	Error Code	Error code
	-		-

This command is only executed in master mode.

If this system option is changed, other commands using user ID, such as CMD\_VERIFY\_FP, CMD\_ADD\_FP, and so on, is changed to accommodate length of user ID.

After executing this command, option is saved by force because non-saved option causes problem in next power-on

The default system mode is none-emulation mode.

# Serial Protocol

## • SYSTEM MANAGEMENT

### CMD\_STATUS\_CHECK

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x62	Command	0x62
Param1	0	Param1	RESULT_SUCCEEDED RESULT_FAILED
Param2	0	Param2	STATUS = IDLE (0x00) BUSY (0x01) : Current executed command DB_UPLOADING (0x03) : During power-up operation, a device isn't ready to communicate
Data Size	0	Data Size	0
Error Code	0	Error Code	Error code
Data	-	Data	-

This command is used to check the status of device.

If device is busy in processing command, it returns BUSY status.

If not, it returns IDLE status.

DB\_UPLOADING status is returned when device is uploading user's DB in boot-up.

# Serial Protocol

## CMD\_GET\_FP\_IMAGE2

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x63	Command	0x63
Param1	0	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_NOT_IN_TIME RESULT_CANCELED
Param2	Packet index (0~N)	Param2	IF (Param1 == RESULT_SUCCEEDED) (Packet Index (0~N) << 8) + (Max Packet Index N)  ELSE 0
Data Size	0	Data Size	IF (Param1 == RESULT_SUCCEEDED) Size of (a piece of image data block)  ELSE 0
Error Code	0	Error Code	Error code
Data	-	Data	IF (Param1 == RESULT_SUCCEEDED) A piece of image data block  ELSE -

This command is used to get fingerprint image.



## Serial Protocol

### CMD\_UPGRADE\_FIRMWARE2

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x64	Command	0x64
Param1	0	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_PARAM RESULT_CANCELED
Param2	(Packet Index (0~N) << 8) + (Max Packet Index N-1)	Param2	Command packet param2 value
Data Size	Size of (a fragment of Firmware data block)	Data Size	0
Error Code	0	Error Code	Error code
Data	Firmware data block	Data	-

Each firmware data block consists of the total size of firmware and a portion of firmware data.  
(Refer to Appendix D)

There need a caution .If wrong data is written to device, device can be out of order. This damage cannot be fixed in user.

## Serial Protocol

### CMD\_SET\_TIME

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x65	Command	0x65
Param1	0	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_DATASIZE RESULT_INVALID_DATA RESULT_CANCELED
Param2	0	Param2	0
Data Size	Size of TIME_INFO	Data Size	0
Error Code	0	Error Code	Error code
Data	TIME_INFO	Data	-

TIME\_INFO data structure is defined in Appendix D

This command is used to set device RTC (real time clock).

# Serial Protocol

## CMD\_GET\_TIME

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x66	Command	0x66
Param1	0	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_CANCELED
Param2	0	Param2	0
Data Size	0	Data Size	IF (Param1 == RESULT_SUCCEEDED) Size of TIME_INFO  ELSE 0
Error Code	0	Error Code	Error code
Data	-	Data	IF (Param1 == RESULT_SUCCEEDED) TIME_INFO  ELSE -

TIME\_INFO data structure is defined in Appendix D

This command is used to set device RTC (real time clock).

## Serial Protocol

### **CMD\_GET\_IMAGE\_QUALITY**

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x68	Command	0x68
Param1	0	Param1	RESULT_SUCCEEDED RESULT_FAILED
Param2	0	Param2	Quality Value
Data Size	0	Data Size	0
Error Code	0	Error Code	Error code
	-		-

This command returns image quality after using the following commands.

CMD\_VERIFY\_FP  
 CMD\_IDENTIFY\_FP  
 CMD\_INSTANT\_MATCHING  
 CMD\_GET\_TEMPLATE  
 CMD\_GET\_FP\_IMAGE2  
 CMD\_ENTER\_MASTER\_MODE2  
 CMD\_REGISTER\_FP  
 CMD\_CHANGE\_FP  
 CMD\_FEGISTER\_MULTI\_FP

For other commands, the value of image quality is invalid.

The quality range is from 0 (low quality) to 100 (high quality).

# Serial Protocol

## APPENDIX A. LOG DATA BLOCK

The log data block consists of index, type, event, and information, and the size of a log data block is 28 bytes. The following table shows the organization of a log data block. The max number of log data to be supported is different according to devices.

### ■ Log Data Format

Index (4)	Type (1)	Event (1)	Information (22)		
			ID (10)	Result (2)	Reserved (10)
0 ~ (0xFFFFFFFF - 1)	Command = 0 <sup>1)</sup>	Enroll = 0	FPID (10)	RESULT (2)	X
		Delete = 1	FPID (10)	RESULT (2)	X
		Verify = 2	FPID (10)	RESULT (2)	X
		Identify = 3	FPID (10)	RESULT (2)	X
		Instant Match = 6	X	RESULT (2)	X
		Enter Master Mode = 7	FPID (10)	RESULT (2)	X
		Set Master = 8	X	RESULT (2)	X
		Reset Master = 9	X	RESULT (2)	X
		Delete All = 10	X	RESULT (2)	X
		Change FP = 11	FPID (10)		
		Change Password = 12	FPID (10)		
		Add FP = 13	FPID (10)		
		Instant Verify = 14			
		Instant Identify = 15			
		Change User Security = 16	FPID (10)		
	Error = 1	Error String			
	Type (1)	Event (1)	Information (22)		
			Result (2)	Time (8)	ID (12)
	Command = 2 <sup>2)</sup>	Enroll = 0	RESULT (2)	TIME (8)	FPID (12)
		Delete = 1	RESULT (2)	TIME (8)	FPID (12)
		Verify = 2	RESULT (2)	TIME (8)	FPID (12)
		Identify = 3	RESULT (2)	TIME (8)	FPID (12)

# Serial Protocol

		Instant Match = 6	RESULT (2)	TIME (8)	FPID (12)
		Enter Master Mode = 7	RESULT (2)	TIME (8)	FPID (12)
		Set Master = 8	RESULT (2)	TIME (8)	FPID (12)
		Reset Master = 9	RESULT (2)	TIME (8)	FPID (12)
		Delete All <sup>1)</sup> = 10	RESULT (2)	TIME (8)	FPID (12)
		Change FP = 11	RESULT (2)	TIME (8)	FPID (12)
		Change Password = 12	RESULT (2)	TIME (8)	FPID (12)
		Add FP = 13	RESULT (2)	TIME (8)	FPID (12)
		Instant Verify = 14	RESULT (2)	TIME (8)	FPID (12)
		Instant Identify = 15	RESULT (2)	TIME (8)	FPID (12)
		Change User Security = 16	RESULT (2)	TIME (8)	FPID (12)
	<b>Type (1)</b>	<b>Event (1)</b>	<b>Information (26)</b>		
			<b>Result (2)</b>	<b>Time (8)</b>	<b>ID (16)</b>
	Command = 3 <sup>3)</sup>	Enroll = 0	RESULT (2)	TIME (8)	FPID (16)
		Delete = 1	RESULT (2)	TIME (8)	FPID (16)
		Verify = 2	RESULT (2)	TIME (8)	FPID (16)
		Identify = 3	RESULT (2)	TIME (8)	FPID (16)
		Instant Match = 6	RESULT (2)	TIME (8)	FPID (16)
		Enter Master Mode = 7	RESULT (2)	TIME (8)	FPID (16)
		Set Master = 8	RESULT (2)	TIME (8)	FPID (16)
		Reset Master = 9	RESULT (2)	TIME (8)	FPID (16)
		Delete All <sup>1)</sup> = 10	RESULT (2)	TIME (8)	FPID (16)
		Change FP = 11	RESULT (2)	TIME (8)	FPID (16)
		Change Password = 12	RESULT (2)	TIME (8)	FPID (16)
		Add FP = 13	RESULT (2)	TIME (8)	FPID (16)
		Instant Verify = 14	RESULT (2)	TIME (8)	FPID (16)
		Instant Identify = 15	RESULT (2)	TIME (8)	FPID (16)
		Change User Security = 16	RESULT (2)	TIME (8)	FPID (16)

- 1) This type is supported in FIM30 Emulation Mode.
- 2) This type is supported in FIM20 Emulation Mode.
- 3) This type is supported in FIM40/50 None Emulation Mode.

## Serial Protocol

---

The following table lists log events.

LOG EVENT LIST	
LOGEVT_ENROLL	0x00
LOGEVT_DELETE	0x01
LOGEVT_VERIFY	0x02
LOGEVT_IDENTIFY	0x03
LOGEVT_INSTANT_MATCH	0x06
LOGEVT_ENTER_MASTERMODE	0x07
LOGEVT_SET_MASTER	0x08
LOGEVT_RESET_MASTER	0x09
LOGEVT_DELETE_ALL	0x0A
LOGEVT_CHANGE_FP <sup>1)</sup>	0x0B
LOGEVT_CHANGE_PASSWD <sup>1)</sup>	0x0C
LOGEVT_ADD <sup>1)</sup>	0x0D
LOGEVT_INSTANT_VERIFY <sup>1)</sup>	0x0E
LOGEVT_INSTANT_IDNETIFY <sup>1)</sup>	0x0F
LOGEVT_NONE	0xFF

## APPENDIX B. Emulation mode Command Table

This chapter describes commands used in Emulation mode.

### ■ FIM20 Emulation Command

<b>CONNECTION</b>	CMD_REQUEST_CONNECTION (0x01) CMD_GET_FIRMWARE_VERSION2 (0x04) CMD_GET_DEVICE_INFO (0x05)
<b>MATCHING</b>	CMD_VERIFY_FP (0x11) CMD_IDENTIFY_FP (0x12) CMD_IDENTIFY_RID_FP(0x13) CMD_INSTANT_MATCHING (0x15) CMD_GET_TEMPLATE (0x16) CMD_CANCEL (0x17) CMD_INSTNAT_VERIFY (0x18) CMD_INSTNAT_IDENTIFY (0x19)
<b>DATABASE MANAGEMENT</b>	CMD_DELETE_FP (0x22) CMD_DELETE_ALL_FP (0x23) CMD_SET_MASTER (0x24) CMD_LEAVE_MASTER_MODE (0x26) CMD_SET_MASTER_PASSWORD (0x27) CMD_READ_USER_DATA (0x2B) CMD_WRITE_USER_DATA (0x2C) CMD_ERASE_USER_DATA_BLOCK (0x2D) CMD_DELETE_MASTER_PASSWORD (0x2E) CMD_ENTER_MASTER_MODE2 (0x2F) CMD_GET_FP_LIST2 (0x30) CMD_GET_MASTER_LIST2 (0x31) CMD_READ_LOG_DATA 2(0x32) CMD_REGISTER_FP (0x33) CMD_CHANGE_FP (0x34)



## Serial Protocol

	CMD_ADD_FP (0x35) CMD_GET_FP (0x36) CMD_DELETE_ALL_LOG (0x37)
<b>CONFIGURATION</b>	CMD_SET_SYSINFO (0x4C) CMD_GET_SYSINFO (0x4D) CMD_SAVE_SYSINFO (0x4E) CMD_CHG_NUM_OF_TEMP (0x4F) CMD_SET_DEFAULT_SYSINFO (0x50) CMD_SET_EMULMODE (0x51) CMD_GET_EMULMODE (0x52)
<b>SYSTEM MANAGEMENT</b>	CMD_STATUS_CHECK (0x62) CMD_GET_FP_IMAGE2 (0x63) CMD_UPGRADE_FIRMWARE2 (0x64) CMD_SET_TIME (0x65) CMD_GET_TIME (0x66) CMD_CTL_IO (0x67) CMD_GET_IMAGE_QUALITY (0x68)

# Serial Protocol

## ■ FIM30 Emulation Command

<b>CONNECTION</b>	CMD_REQUEST_CONNECTION (0x01) CMD_SET_BAUDRATE (0x02) CMD_GET_FIRMWARE_VERSION2 (0x04) CMD_GET_DEVICE_INFO (0x05)
<b>MATCHING</b>	CMD_VERIFY_FP (0x11) CMD_IDENTIFY_FP (0x12) CMD_INSTANT_MATCHING (0x15) CMD_GET_TEMPLATE (0x16) CMD_CANCEL (0x17) CMD_AUTO_IDENTIFY (0x1A) CMD_AUTO_IDENTIFY_RESULT (0x1B)
<b>DATABASE MANAGEMENT</b>	CMD_DELETE_FP (0x22) CMD_DELETE_ALL_FP (0x23) CMD_SET_MASTER (0x24) CMD_LEAVE_MASTER_MODE (0x26) CMD_SET_MASTER_PASSWORD (0x27) CMD_READ_USER_DATA (0x2B) CMD_WRITE_USER_DATA (0x2C) CMD_ERASE_USER_DATA_BLOCK (0x2D) CMD_DELETE_MASTER_PASSWORD (0x2E) CMD_ENTER_MASTER_MODE2 (0x2F) CMD_GET_FP_LIST2 (0x30) CMD_GET_MASTER_LIST2 (0x31) CMD_READ_LOG_DATA2 (0x32) CMD_REGISTER_FP (0x33) CMD_DELETE_ALL_LOG (0x37)
<b>CONFIGURATION</b>	CMD_SET_OPP_OPTION (0x40) CMD_GET_OPP_OPTION (0x41) CMD_SET_SECURITY_LEVEL (0x42) CMD_GET_SECURITY_LEVEL (0x43)

## Serial Protocol

	<div>CMD_SET_CAPTURE_OPTION (0x44)</div> <div>CMD_GET_CAPTURE_OPTION (0x45)</div> <div>CMD_SET_LOG_OPTION (0x48)</div> <div>CMD_GET_LOG_OPTION (0x49)</div> <div>CMD_SET_CAPTURE_PERIOD (0x4A)</div> <div>CMD_GET_CAPTURE_PERIOD (0x4B)</div> <div>CMD_SET_EMULMODE (0x51)</div> <div>CMD_GET_EMULMODE (0x52)</div>
<b>SYSTEM MANAGEMENT</b>	<div>CMD_STATUS_CHECK (0x62)</div> <div>CMD_GET_FP_IMAGE2 (0x63)</div> <div>CMD_UPGRADE_FIRMWARE2 (0x64)</div> <div>CMD_CTL_IO (0x67)</div> <div>CMD_GET_IMAGE_QUALITY (0x68)</div>

# Serial Protocol

## APPENDIX C. PACKET RESULT LIST

The following table lists return code in acknowledge packet.

PACKET RESULT LIST	
RESULT_SUCCEEDED	0x01
RESULT_FAILED	0x02
RESULT_NOT_MASTER_MODE	0x03
RESULT_USED_ID	0x04
RESULT_INVALID_ID	0x05
RESULT_DB_IS_FULL	0x06
RESULT_NOT_IN_TIME	0x07
RESULT_INVALID_PARAM	0x09
RESULT_OPP_INIT_FAILED	0x0C
RESULT_CANCELED	0x0D
RESULT_ANOTHER_FINGER	0x0E
RESULT_IDLE_STATUS	0x10
RESULT_TOO_LARGE_DATA <sup>1)</sup>	0x11
RESULT_IDENTIFY_TIMEOUT <sup>2)</sup>	0x12
RESULT_DB_ISNOT_EMPTY <sup>3)</sup>	0x13
RESULT_WRONG_TEMP_MODE <sup>3)</sup>	0x14
RESULT_INVALID_DATASIZE <sup>3)</sup>	0x15
RESULT_INVALID_DATA <sup>3)</sup>	0x16
RESULT_EXTRACT_FAIL <sup>4)</sup>	0x17
RESULT_NOT_SUPPORTED	0x18
RESULT_AUTO_IDENTIFY_MODE	0x19
RESULT_INVALID_SEQUENCE	0x20

### RESULT\_SUCCEEDED

This code is returned when the command is executed successfully.

# Serial Protocol

---

## **RESULT\_FAILED**

This code is returned when the command cannot be executed for unknown reason.

## **RESULT\_NOT\_MASTER\_MODE**

This code is returned when the command that requires the master privilege is executed in normal mode.

## **RESULT\_USED\_ID**

This code is returned when ID in command packet already exists.

## **RESULT\_INVALID\_ID**

This code is returned when the ID in command packet is invalid.

## **RESULT\_DB\_IS\_FULL**

This code is returned when there is no space for new user.

## **RESULT\_NOT\_IN\_TIME**

This code is returned when the fingerprint image can not be captured in capture timeout.

## **RESULT\_INVALID\_PARAM**

This code is returned when parameters of the command packet are invalid.  
Normally, value out of range is used.

## **RESULT\_OPP\_INIT\_FAILED**

This code is returned when the initialization of the sensor is failed for hardware problem.

## **RESULT\_CANCELED**

This code is returned when cancel command is transferred during executing a previous command.

# Serial Protocol

---

## **RESULT\_ANOTHER\_FINGER**

This code is returned when the first input finger of a user is not equal to the second one in registration process.

## **RESULT\_IDLE\_STATUS**

This code is returned when there is no executed command for the cancel command.

## **RESULT\_TOO\_LARGE\_DATA**

This code is returned when the size of data is greater than the size of pre-defined data structure.

## **RESULT\_IDENTIFY\_TIMEOUT**

This code is returned when identification process (1:N matching) can't be finished until pre-defined identification timeout.

## **RESULT\_DB\_ISNOT\_EMPTY**

This code is returned when the command requires empty DB, but there exists user or users.

## **RESULT\_WRONG\_TEMP\_MODE**

This code is returned when the template mode that required by command is different from the current template mode.

## **RESULT\_INVALID\_DATASIZE**

This code is returned when the size of data needed is different from the size of data sent.

## **RESULT\_INVALID\_DATA**

This code is returned when the transferred data cannot be comprehended.

## **RESULT\_EXTRACT\_FAIL**

This code is returned when device cannot extract template data from image.

## Serial Protocol

---

### **RESULT\_NOT\_SUPPORTED**

This code is returned when unsupported command is sent.

### **RESULT\_AUTO\_IDENTIFY\_MODE**

The code is returned when any commands are executed in Auto Identification mode.

### **RESULT\_INVALID\_SEQUENCE**

This value is returned if wrong parameter is sent in sequential commands like CMD\_REGISTER\_FP or CMD\_REGISTER\_MULTI\_FP.

# Serial Protocol

## Appendix D. DATA STRUCTURE

---

In this chapter, the structure of data block to be transmitted is explained.

According to the device, the variables are defined as the following.

### **FIM30 emulation mode:**

```
LENGTH_OF_FPID = 10
LENGTH_OF_PASSWD = 16
LENGTH_OF_TEMPLATE_HEADER=0
LENGTH_OF_TEMPLATE_DATA = 400
```

### **FIM20 emulation mode:**

```
LENGTH_OF_FPID = 11
LENGTH_OF_PASSWD = 16
LENGTH_OF_TEMPLATE_HEADER=4
LENGTH_OF_TEMPLATE_DATA = 400
```

### **FIM40/50 none emulation mode:**

```
LENGTH_OF_FPID = 11 (default value, can be changed)
LENGTH_OF_PASSWD = 16
LENGTH_OF_TEMPLATE_HEADER=4
LENGTH_OF_TEMPLATE_DATA = 400
```

Caution) FPID and Password are string. So the last byte is null (0x00). The available size of FPID is (LENGTH\_OF\_FPID - 1), and the available size of password is (LENGTH\_OF\_PASSWD - 1).

#### 1. The structure of a FPID

```
Structure    {
                UINT8 FPID[LENGTH_OF_FPID];
            } ID_INFO
```

#### 2. The structure of a password

```
Structure    {
```



## Serial Protocol

---

```
    UINT8 FPPassword[LENGTH_OF_PASSWD];  
} PASSWORD_INFO
```

### 3. The structure of TEMPLATE\_INFO

TEMPLATE\_INFO consists of TEMPLATE Header and Template Data.

**For NITGEN Format:**

```
Structure    {  
    UINT8 Header[LENGTH_OF_TEMPLATE_HEADER];  
    UINT8 Data[LENGTH_OF_TEMPLATE_DATA];  
} TEMPLATE_INFO
```

**For ISO 19794-2 Format:**

```
Structure    {  
    UINT8 Header[LENGTH_OF_TEMPLATE_HEADER];  
    UINT8 Data[Length of ISO data];  
} TEMPLATE_INFO
```

“Length of ISO data” varies according to the length of ISO template data.

Template Header is defined as followings:

0x00 0x00 0x00 0x03 : NITGEN Data Format

0x00 0x00 0x01 0x00: ISO 17974-2 Format

0x00 0x00 0x02 0x00: ANSI 378 Format

### 4. The structure of a “FPID + Password”

```
Structure    {  
    ID_INFO FPID;  
    PASSWORD_INFO FPPassword;  
}
```

### 5. The structure of a “FPID + Template”

```
Structure    {  
    ID_INFO FPID;  
    TEMPLATE_INFO FPTemplate;
```

## Serial Protocol

---

}

6. The structure of a “FPID + Template Index”

```
Structure  {
    ID_INFO FPID;
    UINT8 Template_Index;
}
```

7. The structure of a “FPID + User type”

```
Structure  {
    ID_INFO FPID;
    UINT8 Right;           // Normal User:0   Master: 1
}
```

8. The structure of a list block used in CMD\_GET\_FP\_LIST2, CMD\_GET\_MASTER\_LIST2

if Param1 == 0

```
Structure  {
    UINT16 User_Number;    // for example, N
    UINT16 ID_Size;        // FIM30 emulation mode: 10
                           // FIM40/50 or FIM20 emulation mode:
                           // 11
    ID_INFO FPID_1;
    ID_INFO FPID_2;
    ...
    ID_INFO FPID_N;
}
```

else if Param1 == 1

```
Structure  {
    UINT16 User_Number;    // for example, N
}
```

User\_Number and ID\_Size are big endian format.

## Serial Protocol

---

9. The structure of a log block used in CMD\_READ\_LOG\_DATA2

```
Structure    {
    UINT16 Log_Number;           // for example, N
    UINT16 Log_Size;
    LOG_DATA Log1;
    LOG_DATA Log2;
    ...
    LOG_DATA LogN;
}
```

Log\_Number and Log\_Size are big endian.

For FIM30 Emulation Mode (28 Bytes)

```
Structure    {
    UINT8 Index[4];             // Big Endian Format
    UINT8 Type;                 // For FIM30 emulation mode, 0x00
    UINT8 Event;
    UINT8 UserID[10];
    UINT8 Result[2];           // Big Endian Format
    UINT8 Reserved[10];       // all 0xFF
} LOG_DATA_TYPE0
```

For FIM20 Emulation Mode (28 Bytes)

```
Structure    {
    UINT8 Index[4];             // Big Endian Format
    UINT8 Type;                 // For FIM20 emulation mode, 0x02
    UINT8 Event;
    UINT8 Result[2];           // Big Endian Format
    UINT* Time[8];
    UINT8 UserID[12];
} LOG_DATA_TYPE2
```

For None Emulation Mode (32 Bytes)

## Serial Protocol

```

Structure {
    UINT8 Index[4];           // Big Endian Format
    UINT8 Type;               // For None emulation mode, 0x03
    UINT8 Event;
    UINT8 Result[2];         // Big Endian Format
    UINT* Time[8];
    UINT8 UserID[16];
} LOG_DATA_TYPE3

```

10. The structure of a firmware block used in CMD\_UPGRADE\_FIRMWARE2

(Refer to Appendix F)

```

Structure {
    UINT32 Firmware_Size; // for example, N = M1+M2+ ... + Mn
    UINT8 Firmware[M*];   // 1 ≤ M* ≤ 32768
}

```

11. The structure of a TIME\_INFO used in CMD\_SET\_TIME and CMD\_GET\_TIME

```

Structure {
    UINT8 HundredthYear; // hundredth Year
    UINT8 Year;           // Remain Year
    UINT8 Month;          // Month: from 1 to 12
    UINT8 Date;           // Date: from 1 to 31
    UINT8 Hour;           // Hour: form 0 to 23
    UINT8 Minute;         // Minute: form 0 to 59
    UINT8 Second;         // Second: form 0 to 59
    UINT8 Reserved;       //
} TIME_INFO

```

All data is BCD code.

12. SI\_TYPE used in CMD\_SET\_SYSINFO and CMD\_GET\_SYSINFO

Code	System Information	Value Range	Default Value	
------	--------------------	-------------	---------------	--

## Serial Protocol

<b>0x01</b>	SI_USING_RELAY	True/False	False	<b>Not Executed</b>
<b>0x02</b>	SI_USING_LOG	True/False	False	
<b>0x03</b>	SI_NUM_OF_TEMP	2, 4	2	<b>Supported in FIM20 Emulation mode only</b>
<b>0x10</b>	SI_WIEGAND_FORMAT	0 – No out 1 – 26 bits 2 – 34 bits	0	<b>No side effect</b>
<b>0x11</b>	SI_WIEGAND_SITECODE		0x0000	<b>No side effect</b>
<b>0x17</b>	SI_IDENTIFY_TIMEOUT	255 or 10 ~ 250	30	100ms tick 255 = unlimited
<b>0x18</b>	SI_RELAY_TIME	0 or 1~100	10	100ms ticks
<b>0x19</b>	SI_CAPTURE_TIMEOUT	More than 10	50	100ms ticks
<b>0x20</b>	SI_IMAGE_BRIGHTNESS	0~100	45	100 - brightest
<b>0x21</b>	SI_IMAGE_GAIN	1,2,4,8	2	
<b>0x22</b>	SI_IMAGE_CONTRAST	0~100	20	
<b>0x28</b>	SI_ADAPTIVE_CAPTURE	True/False	False	
<b>0x30</b>	SI_VERIFY_SECURITY_LEVEL	1~9	5	
<b>0x31</b>	SI_IDENTIFY_SECURITY_LEVEL	6~9	8	
<b>0x32</b>	SI_REGISTER_QUALITY	30~100	40	
<b>0x33</b>	SI_VERIFY_QUALITY	10~100	30	
<b>0x38</b>	SI_USING_LATENT	True/False	False	<b>No side effect</b>
<b>0x40</b>	SI_ENABLE_CHANNEL1	True/False	True	<b>No Side effect</b>
<b>0x48</b>	SI_CHANNEL0_BAUDRATE	0 – 115200 1 – 57600 2 – 38400 3 – 19200 4 – 9600	4	
<b>0x49</b>	SI_CHANNEL1_BAUDRATE	“	4	
<b>0x4A</b>	SI_CURR_CHANNEL_BAUDRATE	“		
<b>0x50</b>	SI_MAX_USER			<b>Supported in FIM20 Emulation</b>

## Serial Protocol

				mode only
<b>0x51</b>	SI_FP_FULL_ROTATION	True/False	False	
<b>0x52</b>	SI_LENGTH_OF_USER_ID	4~15	10	
<b>0x53</b>	SI_NUM_OF_ADAPTIVE_CAP	1~10	5	
<b>0xF0</b>	SI_EMULATION_MODE	1 – FIM20 2 – FIM30 0xFF - None	0xFF	

True means '1' and False means '0'.

SI\_LENGTH\_OF\_USER\_ID and SI\_EMULATION\_MODE are only changed when there is no user.

If SI\_EMULATION\_MODE is changed, SI\_LENGTH\_OF\_USER\_ID can be changed according to the value of SI\_EMULATION\_MODE.

13. DB structure used in CMD\_SET\_SYSINFO and CMD\_GET\_SYSINFO

```

Structure  {
    UINT32  SI_TYPE;
    UINT32  SI_VALUE
} SI_INFO

```

14. DB structure used in CMD\_ADD\_FP and CMD\_GET\_FP

**Old Format for FIM30:**

```

Structure  {
    UINT8  Right;                // Normal User:0  Master: 1
    ID_INFO FIID;                // user ID
    PASSWOR_DINFO  FPPasswd;    // Password
    TEMPLATE_INFO FPTemplate1;  // 1st Template
    TEMPALTE_INFO FPTemplate2;  // 2nd Template
} FIM_OLD_DB_FIM30

```

**New Format for FIM30:**

```

Structure  {
    UINT8 Header[4];            // Data Header

```

## Serial Protocol

---

```

    UINT8  Right;                // Normal User:0  Master: 1
    ID_INFO FIID;                // user ID
    PASSWOR_DINFO FPPasswd;      // Password
    UINT8  SecuLevelInfo;        // 0xFC: Using security Level
                                   // Otherwise: reserved
    UINT8  UserSecuLevel;        // User's verification security
    UINT8  Reserved[6];          // Reserved area for future use
    SIZE_INFO FPSize1;           // size of FPTemplate1
    SIZE_INFO FPSize2;           // size of FPTemplate2
    TEMPLATE_INFO FPTemplate1;   // 1st Template
    TEMPALTE_INFO FPTemplate2;   // 2nd Template
} FIM_DB_FIM30

```

Reserved area must be set full 0xFF.

### Old Format for FIM20:

```

Structure  {
    UINT8  Right;                // Normal User:0  Master: 1
    ID_INFO FIID;                // user ID
    PASSWOR_DINFO FPPasswd;      // Password
    TEMPLATE_INFO FPTemplate1;   // 1st Template
    TEMPALTE_INFO FPTemplate2;   // 2nd Template
    TIME_INFO Time;              // Time Information
} FIM_OLD_DB2

```

```

Structure  {
    UINT8  Right;                // Normal User:0  Master: 1
    ID_INFO FPID;                // user ID
    PASSWORD_INFO FPPasswd;      // Password
    TEMPLATE_INFO FPTemplate1;   // 1st Template
    TEMPLATE_INFO FPTemplate2;   // 2nd Template
    TEMPLATE_INFO FPTemplate3;   // 3rd Template
    TEMPLATE_INFO FPTemplate4;   // 4th Template
    TIME_INFO Time;              // Time Information
}

```

## Serial Protocol

---

} FIM\_OLD\_DB4

New Format for FIM20:

```

Structure  {
    UINT8 Header[4];           // Data Header
    UINT8  Right;              // Normal User:0  Master: 1
    ID_INFO FIID;              // user ID
    PASSWORD_INFO FPPasswd;    // Password
    UINT8 SecuLevelInfo;       // 0xFC: Using security Level
                                // Otherwise: reserved
    UINT8 UserSecuLevel;       // User's verification security
    UINT8 Reserved[6];         // Reserved area for future use
    TIME_INFO Time;            // Time Information
    SIZE_INFO FPSize1;         // size of FPTemplate1
    SIZE_INFO FPSize2;         // size of FPTemplate2
    TEMPLATE_INFO FPTemplate1; // 1st Template
    TEMPALTE_INFO FPTemplate2; // 2nd Template
} FIM_DB2

```

Reserved area must be set full 0xFF.

```

Structure  {
    UINT8 Header[4];           // Data Header
    UINT8  Right;              // Normal User:0  Master: 1
    ID_INFO FPID;              // user ID
    PASSWORD_INFO FPPasswd;    // Password
    UINT8 SecuLevelInfo;       // 0xFC: Using security Level
                                // Otherwise: reserved
    UINT8 UserSecuLevel;       // User's verification security
    UINT8 Reserved[6]          // Reserved area for future use
    TIME_INFO Time;            // Time Information
    SIZE_INFO FPSize1;         // size of FPTemplate1
    SIZE_INFO FPSize2;         // size of FPTemplate2
    SIZE_INFO FPSize3;         // size of FPTemplate3

```



## Serial Protocol

```

        SIZE_INFO FPSize4;           // size of FPTemplate4
        TEMPLATE_INFO FPTemplate1;  // 1st Template
        TEMPLATE_INFO FPTemplate2;  // 2nd Template
        TEMPLATE_INFO FPTemplate3;  // 3rd Template
        TEMPLATE_INFO FPTemplate4;  // 4th Template
    } FIM_DB4

```

Reserved area must be set full 0xFF.

### Format for FIM40/50:

```

    Structure  {
        UINT8 Header[4];           // Data Header
        UINT8 Right;               // Normal User:0 Master: 1
        ID_INFO FPID;              // user ID
        PASSWORD_INFO FPPasswd;    // Password
        UINT8 SecuLevelInfo;        // 0xFC: Using security Level
                                   // Otherwise: reserved
        UINT8 UserSecuLevel;        // User's verification security
        UINT8 Reserved[6];          // Reserved area for future use
        TIME_INFO Time;             // Time Information
        SIZE_INFO FPSize[10];       // size of FPTemplate
        TEMPLATE_INFO FPTemplate1; // 1st Template
        TEMPLATE_INFO FPTemplate2; // 2nd Template
        TEMPLATE_INFO FPTemplate3; // 3rd Template
        .
        .
        TEMPLATE_INFO FPTemplate10; // 10th Template
    } FIM_MT_DB

```

Reserved area must be set full 0xFF.

FPTemplate1 ~ 10 exist only when FPSize[ ] has non-zero value.

For example, When 1 template exists, FPSize[0] has non-zero value and FPSzie[1] ~ FPSzie[9] has zero.

Therefore, FPTemplate2 ~ FPTemplate10 does not exists.

## Serial Protocol

---

```
Structure    {  
                UINT8  Size_H;                // MSB 8 bits  
                UINT8  Size_L;                // LSB 8 bits  
            } SIZE_INFO  
Size = (Size_H x 256) + Size_L
```

**Header** has the following values

“0xC1 0x00 0x00 0x00” – FIM\_DB2 with NITGEN format data  
“0xC1 0x01 0x00 0x00” – FIM\_DB2 with ISO 19794-2 format data  
“0xC1 0x02 0x00 0x00” – FIM\_DB2 with ANSI 378 format data  
“0xC2 0x00 0x00 0x00” – FIM\_DB4 with NITGEN format data  
“0xC2 0x01 0x00 0x00” – FIM\_DB4 with ISO 19794-2 format data  
“0xC2 0x02 0x00 0x00” – FIM\_DB4 with ANSI 378 format data  
“0xC3 0x00 0x00 0x00” – FIM\_MT\_DB with NITGEN format data  
“0xC3 0x01 0x00 0x00” – FIM\_MT\_DB with ISO 19794-2 format data  
“0xC3 0x02 0x00 0x00” – FIM\_MT\_DB with ANSI 378 format data

In ISO 91794-2 format, the maximum number of minutiae supported in FIM is 80.

# Serial Protocol

## Appendix E. EXAMPLES

In this chapter, communication method is explained with examples. These examples are made for the following conditions.

LENGTH\_OF\_FPID = 11  
 LENGTH\_OF\_PASSWD = 16  
 LENGTH\_OF\_TEMPLATE\_HEADER=4  
 LENGTH\_OF\_TEMPLATE\_DATA = 400

### 1. Request Connection

For checking serial connection, use “Request Connection” command. For explanation on real packet data, assume that the device has 10 users in DB. The following figure shows the sequence of packets, and the contents of packets.

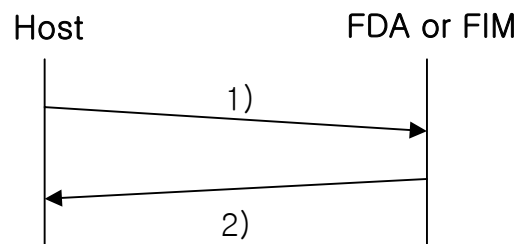


Figure E.1 The sequence of Request Connection

#### 1) The structure of CMD\_REQUEST\_CONNECTION command packet

The following table shows the command packet made in the host.

Command	0x00000001
Param1	0x00000000
Param2	0x00000000
Data Size	0x00000000
Error Code	0x00000000
Header Checksum	0x00000001

The following table shows the sequence of data to be transmitted to the device.

7E	00 00 00 01	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 01
----	-------------	-------------	-------------	-------------	-------------	-------------

## Serial Protocol

---

### 2) Acknowledgement packet

In response to CMD\_REQUEST\_CONNECTION packet from host, the device sends acknowledgement packet meaning a success as the following.

Command	0x00000001
Param1	0x00000001
Param2	0x0000000A
Data Size	0x00000000
Error Code	0x00000000
Header Checksum	0x0000000C

If the host gets the following packet, it means that the communication was successfully done.

7E	00 00 00 01	00 00 00 01	00 00 00 0A	00 00 00 00	00 00 00 00	00 00 00 0C
----	-------------	-------------	-------------	-------------	-------------	-------------

Parameter 2 (0x0A) means 10 users are registered.

## 2. User Enrollment

There are two methods in registering user.

In FIM20 or FIM30 emulation mode, CMD\_REGISTER\_FP is used. But in none emulation mode, CMD\_REGISTER\_MULTI\_FP is recommended.

### 2.1 Using CMD\_REGISTER\_FP (Supported in FIM20 & 30 emulation mode)

This command is used for registering user with fingerprint, password, and master privilege setting in registration. Password is optional.

#### 2.1.1 Enrolling Normal User

Assume that a device has 10 users in DB. The following description explains the sequence of registering normal user with the ID '1234' and the password "5678".

In 2-template mode, device requires two fingerprint image. This sequence is controlled by

1) The structure of CMD\_REGISTER\_FP command packet (First capture)

Command	0x00000033									
Param1	0x00000000									
Param2	0x00000000									
Data Size	0x0000001B									
Error Code	0x00000000									
Header Checksum	0x0000004E									
Data	0x31	0x32	0x33	0x34	0x00	0x00	0x00	0x00	0x00	0x00
	0x00	0x35	0x36	0x37	0x38	0x00	0x00	0x00	0x00	0x00
	0x00	0x00	0x00	0x00	0x00	0x00	0x00			
Data Checksum	0x000001A4									

Param1 is used whether user is master or not. Param2 is used to inform the sequence of registration. This packet request first image capture.

7E	00 00 00 33	00 00 00 00	00 00 00 00	00 00 00 1A	00 00 00 00	00 00 00 4E
----	-------------	-------------	-------------	-------------	-------------	-------------

31 32 33 34 00 00 00 00 00 00 00 00 35 36 37 38 00 00 00 00 00 00 00 00 00 00	00 00 01 A4
---	-------------

In response to CMD\_REGISTER\_FP packet from host, the device sends acknowledgement packet meaning a success as the following.

Command	0x00000033
Param1	0x00000001
Param2	0x00000000

## Serial Protocol

Data Size	0x00000000
Error Code	0x00000000
Header Checksum	0x00000034

If the host gets the following packet, it means that the communication was successfully done.

7E	00 00 00 33	00 00 00 01	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 34
----	-------------	-------------	-------------	-------------	-------------	-------------

### 3) The structure of CMD\_REGISTER\_FP command packet (Second capture)

If the acknowledgement packet to the first CMD\_REGISTER\_FP is returned successfully, the host sends the second CMD\_REGISTER\_FP command packet as the following for requesting second capture.

7E	00 00 00 33	00 00 00 00	00 00 00 01	00 00 00 00	00 00 00 00	00 00 00 34
----	-------------	-------------	-------------	-------------	-------------	-------------

In this packet .Param2 is used to control the sequence of registration.

### 4) Acknowledgement packet

In response to CMD\_REGISTER\_FP packet from host, the device sends acknowledgement packet as the following meaning a success.

7E	00 00 00 33	00 00 00 01	00 00 00 0B	00 00 00 00	00 00 00 00	00 00 00 3F
----	-------------	-------------	-------------	-------------	-------------	-------------

If this acknowledge packet is returned with successful result, user is added to device.

## 2.1.2 Registering Master

Assume that a device has 10 users in DB. The following description explains the sequence of registering master with the ID '1234', the password "5678" and master privilege.

### 1) The structure of CMD\_REGISTER\_FP command packet (First capture)

Command	0x00000033									
Param1	0x00000001									
Param2	0x00000000									
Data Size	0x0000001B									
Error Code	0x00000000									
Header Checksum	0x0000004F									
Data	0x31	0x32	0x33	0x34	0x00	0x00	0x00	0x00	0x00	0x00
	0x00	0x35	0x36	0x37	0x38	0x00	0x00	0x00	0x00	0x00
	0x00	0x00	0x00	0x00	0x00	0x00	0x00			
Data Checksum	0x000001A4									

7E	00 00 00 33	00 00 00 01	00 00 00 00	00 00 00 1A	00 00 00 00	00 00 00 4E
----	-------------	-------------	-------------	-------------	-------------	-------------

7E	00 00 00 33	00 00 00 01	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 34
----	-------------	-------------	-------------	-------------	-------------	-------------

If the acknowledgement packet to the first CMD\_REGISTER\_FP is returned successfully, the host sends the second CMD\_REGISTER\_FP command packet as the following.

7E	00 00 00 33	00 00 00 01	00 00 00 01	00 00 00 00	00 00 00 00	00 00 00 35
----	-------------	-------------	-------------	-------------	-------------	-------------

#### 4) Acknowledgement packet

In response to CMD\_REGISTER\_FP packet from host, the device sends acknowledgement packet as the following meaning a success.

7E	00 00 00 33	00 00 00 01	00 00 00 0B	00 00 00 00	00 00 00 00	00 00 00 3F
----	-------------	-------------	-------------	-------------	-------------	-------------

### 2.1.3 Enrolling Normal User with different finger

In FIM20 & FIM30 emulation mode, device requires two fingerprint image and check the fingerprint by matching first and second image. If two images are matched, device save user with 2 templates what we called 1 ID 1 Fingers 2 Templates.

In some case, host want to save 1 ID with two different fingers.

Here is the method to save 1 ID with two different fingers, what we called 1 ID 2 Fingers 2 Templates.

Assume that a device has 10 users in DB. The following description explains the sequence of registering normal user with the different finger and the ID '1234' and the password "5678".

### 1) The structure of CMD\_REGISTER\_FP command packet (First Capture)

This step is the same process with enrolling normal user

7E	00 00 00 33	00 00 00 00	00 00 00 00	00 00 00 1B	00 00 00 00	00 00 00 4E
----	-------------	-------------	-------------	-------------	-------------	-------------

31 32 33 34 00 00 00 00 00 00 35 36 37 38 00 00 00 00 00 00 00 00 00 00	00 00 01 A4
---	-------------

## 2) Acknowledgement packet



## Serial Protocol

If the host gets the following packet, it means that the communication was successfully done.

7E	00 00 00 33	00 00 00 01	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 34
----	-------------	-------------	-------------	-------------	-------------	-------------

### 3) The structure of CMD\_REGISTER\_FP command packet (Second capture)

If the acknowledgement packet to the first CMD\_REGISTER\_FP is returned successfully, the host sends the second CMD\_REGISTER\_FP command packet as the following.

In order to enroll different finger with same ID, param2 of command packet have to be set in 2.

7E	00 00 00 33	00 00 00 00	<u>00 00 00 02</u>	00 00 00 00	00 00 00 00	00 00 00 35
----	-------------	-------------	--------------------	-------------	-------------	-------------

### 4) Acknowledgement packet

In response to CMD\_REGISTER\_FP packet from host, the device sends acknowledgement packet as the following meaning a success.

7E	00 00 00 33	00 00 00 01	00 00 00 0B	00 00 00 00	00 00 00 00	00 00 00 3F
----	-------------	-------------	-------------	-------------	-------------	-------------

## 2.2 Using CMD\_REGISTER\_MULTI\_FP

This command is newly added in FIM40/50 for supporting 1 ID multi-template mode.

Host can add 1 ID up to 10 fingers.

### 2.2.1 1 ID 1 finger Enrollment

Assume that a device has 10 users in DB. The following description explains the sequence of registering normal user with the ID '1234' and the password "5678'.

#### 1) The structure of CMD\_REGISTER\_MULTI\_FP command packet (First capture)

The following table shows the command packet made in the host.

Command	0x00000038
Param1	0x00000000



In this example, FPID is 11 bytes null-terminated string and Password is 16 bytes null-terminated string. Though password is not used, 16 bytes null data must be sent.

The following table shows the sequence of data to be transmitted to the device.

31 32 33 34 00 00 00 00 00 00 00 00 35 36 37 38 00 00 00 00 00 00 00 00 00 00	00 00 01 A4
---	-------------

In response to CMD\_REGISTER\_FP packet from host, the device sends acknowledgement packet meaning a success as the following.

If the host gets the following packet, it means that the communication was successfully done.

### 3) The structure of CMD REGISTER MULTI FP command packet (Second capture)

## Serial Protocol

---

If the acknowledgement packet to the first CMD\_REGISTER\_MULTI\_FP is returned successfully, the host sends the second CMD\_REGISTER\_MULTI\_FP command packet as the following for requesting second capture.

7E	00 00 00 38	00 00 00 00	00 00 00 03	00 00 00 00	00 00 00 00	00 00 00 39
----	-------------	-------------	-------------	-------------	-------------	-------------

In this packet .Capture mode of param2 is used to control the sequence of registration. By using capture mode 0x03, device capture fingerprint and save user.

#### 4) Acknowledgement packet

In response to CMD\_REGISTER\_FP packet from host, the device sends acknowledgement packet as the following meaning a success.

7E	00 00 00 38	00 00 00 01	00 00 00 0B	00 00 00 00	00 00 00 00	00 00 00 44
----	-------------	-------------	-------------	-------------	-------------	-------------

If this acknowledge packet is returned with successful result, user is added to device.

#### 2.2.2 1 ID multi-fingers Enrollment

Assume that a device has 10 users in DB. The following description explains the sequence of registering normal user with the ID '1234' and the password "5678'.

##### 1) The structure of CMD\_REGISTER\_MULTI\_FP command packet (First capture)

The following table shows the command packet made in the host.

Command	0x00000038
Param1	0x00000000
Param2	0x00000000
Data Size	0x0000001B
Error Code	0x00000000

# Serial Protocol

Header Checksum	0x00000053									
Data	0x31	0x32	0x33	0x34	0x00	0x00	0x00	0x00	0x00	0x00
	0x00	0x35	0x36	0x37	0x38	0x00	0x00	0x00	0x00	0x00
	0x00	0x00	0x00	0x00	0x00	0x00	0x00			
Data Checksum	0x000001A4									

In this example, FPID is 11 bytes null-terminated string and Password is 16 bytes null-terminated string. Though password is not used, 16 bytes null data must be sent.

Param1 is used whether user is master or not. Param2 is used to inform the sequence of registration and finger index. This packet request first image capture.

The following table shows the sequence of data to be transmitted to the device.

7E	00 00 00 38	00 00 00 00	00 00 00 00	00 00 00 1A	00 00 00 00	00 00 00 53
----	-------------	-------------	-------------	-------------	-------------	-------------

31 32 33 34 00 00 00 00 00 00 00 35 36 37 38 00 00 00 00 00 00 00 00 00	00 00 01 A4
---	-------------

## 2) Acknowledgement packet

In response to CMD\_REGISTER\_FP packet from host, the device sends acknowledgement packet meaning a success as the following.

Command	0x00000038
Param1	0x00000001
Param2	0x00000000
Data Size	0x00000000
Error Code	0x00000000
Header Checksum	0x00000039

If the host gets the following packet, it means that the communication was successfully done.

7E	00 00 00 38	00 00 00 01	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 39
----	-------------	-------------	-------------	-------------	-------------	-------------

### 3) The structure of CMD\_REGISTER\_MULTI\_FP command packet (Second capture)

If the acknowledgement packet to the first CMD\_REGISTER\_MULTI\_FP is returned successfully, the host sends the second CMD\_REGISTER\_MULTI\_FP command packet as the

## Serial Protocol

---

following for requesting second capture.

7E	00 00 00 38	00 00 00 00	00 00 00 02	00 00 00 00	00 00 00 00	00 00 00 39
----	-------------	-------------	-------------	-------------	-------------	-------------

In this packet .Capture mode of param2 is used to control the sequence of registration. By using capture mode 0x02, device capture fingerprint but not save yet.

### 4) Acknowledgement packet

In response to CMD\_REGISTER\_FP packet from host, the device sends acknowledgement packet as the following meaning a success.

7E	00 00 00 38	00 00 00 01	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 39
----	-------------	-------------	-------------	-------------	-------------	-------------

5) Repeat from 1) to 4) until host want to finish adding finger to user

Finger index must be changed.

### 6) Save user

To finish and save, set capture mode 0x04

7E	00 00 00 38	00 00 00 00	00 00 00 04	00 00 00 00	00 00 00 00	00 00 00 3B
----	-------------	-------------	-------------	-------------	-------------	-------------

### 7) Acknowledgement packet

In response to CMD\_REGISTER\_FP packet from host, the device sends acknowledgement packet as the following meaning a success.

7E	00 00 00 38	00 00 00 01	00 00 00 0B	00 00 00 00	00 00 00 00	00 00 00 44
----	-------------	-------------	-------------	-------------	-------------	-------------

If acknowledge packet is returned with successful result, user is added to device.

# Serial Protocol

## 3. User Deletion

Two deletion commands such as CMD\_DELETE\_FP for deleting a single user and CMD\_DELETE\_ALL\_FP for deleting all users are supported. For example, assume that a device has 10 users in DB. The following description shows the sequence of deleting user that has the ID '1234'.

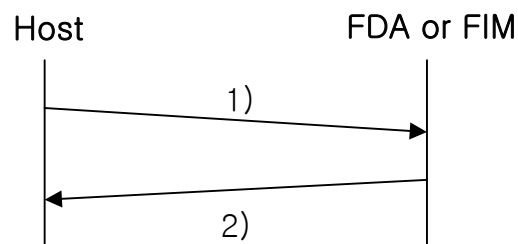


Figure E.3 The sequence of deletion

### 1) The structure of CMD\_DELETE\_FP command packet

The following table shows the command packet made in the host.

Command	0x00000022									
Param1	0x00000000									
Param2	0x00000000									
Data Size	0x0000000B									
Error Code	0x00000000									
Header Checksum	0x0000002D									
Data	0x31	0x32	0x33	0x34	0x00	0x00	0x00	0x00	0x00	0x00
	0x00									
Data Checksum	0x000000CA									

The following table shows the sequence of data to be transmitted to the device.

7E	00 00 00 22	00 00 00 00	00 00 00 00	00 00 00 0A	00 00 00 00	00 00 00 2C
----	-------------	-------------	-------------	-------------	-------------	-------------

31 32 33 34 00 00 00 00 00 00 00	00 00 00 CA
----------------------------------	-------------

# Serial Protocol

---

## 2) Acknowledgement packet

In response to CMD\_DELETE\_FP packet from host, the device sends acknowledgement packet meaning a success as the following.

Command	0x00000022
Param1	0x00000001
Param2	0x00000009
Data Size	0x00000000
Error Code	0x00000000
Header Checksum	0x0000002C

If the host gets the following packet, it means that the communication was successfully done.

7E	00 00 00 22	00 00 00 01	00 00 00 09	00 00 00 00	00 00 00 00	00 00 00 2C
----	-------------	-------------	-------------	-------------	-------------	-------------

## 4. Authentication

There are two methods for authentication.

Verification for 1:1 authentication

Identification for 1:N authentication.

### 4.1 Verification (1:1 Authentication)

The device supports user verification with fingerprint or password.

#### 4.1.1 Verification with fingerprint

The following description shows the sequence of verifying user that has the ID '1234'.

## Serial Protocol

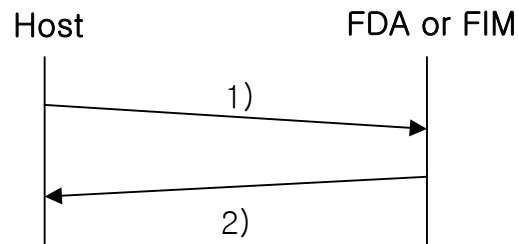


Figure E.4 sequence of verification

### 1) The structure of CMD\_VERIFY\_FP command packet

The following table shows the command packet made in the host.

Command	0x00000011									
Param1	0x00000000									
Param2	0x00000000									
Data Size	0x0000000B									
Error Code	0x00000000									
Header Checksum	0x0000001C									
Data	0x31	0x32	0x33	0x34	0x00	0x00	0x00	0x00	0x00	0x00
	0x00									
Data Checksum	0x000000CA									

The following table shows the sequence of data to be transmitted to the device.

7E	00 00 00 11	00 00 00 00	00 00 00 00	00 00 00 0A	00 00 00 00	00 00 00 1B
31 32 33 34 00 00 00 00 00 00 00						00 00 00 CA

### 2) Acknowledgement packet

In response to CMD\_VERIFY\_FP packet from host, the device sends acknowledgement packet meaning a success as the following.

Command	0x00000011
Param1	0x00000001
Param2	0x00000000



If the host gets the following packet, it means that the communication was successfully done.

#### 4.1.2 Verification with password

### 1) The structure of CMD\_VERIFY\_FP command packet

The following table shows the sequence of data to be transmitted to the device.

## 2) Acknowledgement packet

In response to CMD\_VERIFY\_FP packet from host, the device sends acknowledgement packet

## Serial Protocol

meaning a success as the following.

Command	0x00000011
Param1	0x00000001
Param2	0x00000000
Data Size	0x00000000
Error Code	0x00000000
Header Checksum	0x00000012

If the host gets the following packet, it means that the communication was successfully done.

7E	00 00 00 11	00 00 00 01	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 12
----	-------------	-------------	-------------	-------------	-------------	-------------

### 4.2 Identification (1:N Authentication)

The device supports user verification only with fingerprint.

#### 4.2.1 Identification with fingerprint

The following description shows the sequence of Identification.

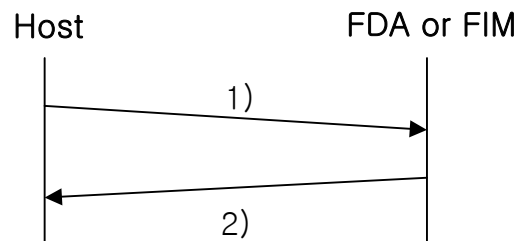


Figure E.5 The sequence of Identification

#### 1) The structure of CMD\_IDENTIFY\_FP command packet

The following table shows the command packet made in the host.

Command	0x00000012
Param1	0x00000000
Param2	0x00000000

## Serial Protocol

Data Size	0x00000000
Error Code	0x00000000
Header Checksum	0x00000012

The following table shows the sequence of data to be transmitted to the device.

7E	00 00 00 12	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 12
----	-------------	-------------	-------------	-------------	-------------	-------------

### 2) Acknowledgement packet

In response to CMD\_IDENTIFY\_FP packet from host, the device sends acknowledgement packet meaning a success as the following.

Command	0x00000012									
Param1	0x00000001									
Param2	0x00000000									
Data Size	0x0000000A									
Error Code	0x00000000									
Header Checksum	0x0000001D									
Data	0x31	0x32	0x33	0x34	0x00	0x00	0x00	0x00	0x00	0x00
Data Checksum	0x000000CA									

If the host gets the following packet, it means that the communication was successfully done.

If succeeded, User ID is also returned.

7E	00 00 00 12	00 00 00 01	00 00 00 00	00 00 00 0B	00 00 00 00	00 00 00 1E
----	-------------	-------------	-------------	-------------	-------------	-------------

31 32 33 34 00 00 00 00 00 00 00	00 00 00 CA
----------------------------------	-------------

# Serial Protocol

## Appendix F. THE EXAPMLE OF FIRMWARE UPGRADE

This chapter explains packet sequence for upgrading firmware with 'CMD\_UPGRADE\_FIRMWARE2' command.

In upgrading firmware, the data block of packet consists of size information and a portion of firmware. For example, assume that firmware size is "S", and firmware data consists of 10 blocks - B0, B1, ... B9 as the following table.

B0	B1	B2	B3	B4	B5	B6	B7	B8	B9
----	----	----	----	----	----	----	----	----	----

Using CMD\_UPGRADE\_FIRMWARE2 command, 10 packets are needed for carrying firmware. The following table shows all 10 packets.

Packet1 Data

Header	S	B0	Checksum
--------	---	----	----------

Packet2 Data

Header	S	B1	Checksum
--------	---	----	----------

...

Packet10 Data

Header	S	B9	Checksum
--------	---	----	----------

---

## Appendix G. Support Information

### NITGEN & COMPANY Co., Ltd.

Ace Techno-Tower II, #105., 197-7,  
Guro-Dong, Guro-Gu,  
Seoul,, Korea (152-766)

Tel: +82-80-060-1600

Fax: +82-2-867-3801

E-mail. [Customer@nitgen.com](mailto:Customer@nitgen.com)