SDA0x Developer's Guide
(DRAFT – SUBJECT TO CHANGE)

SDA04 - FIPS201 OEM Fingerprint Module, 10,000 users, 533MHz CPU
SDA04M – OEM Fingerprint Module, 10,000 users,533MHz CPU
SDA03M – OEM Fingerprint Module, 3,000 users,400MHz CPU

July 2008

SG1-1300A-001 (07/2008)

SecuGen END USER LICENSE AGREEMENT

SecuGen Corporation ("SecuGen") is granting you (an individual or an entity, either of which is referred to herein as "Licensee") a license to use SecuGen's Developer's Kit or its Evaluation Kit, including computer software, hardware, associated media and printed materials (individually "DK" and "EK," and collectively the "Products") only upon the condition that Licensee accepts all of the terms and conditions contained in this User License Agreement (the "Agreement"). If Licensee does not agree to the terms of this Agreement, Licensee should not install or use the Products, but rather, promptly return the Products to the place of purchase.

1. **Grant of License**. This Agreement grants Licensee a personal, limited, non-transferable, non-exclusive right to install and use one copy of the Products on a single computer exclusively for the following purposes:
   a. Licensee may use the DK solely for developing, designing, and testing SecuGen software applications for use with SecuGen products ("Applications").
   b. Licensee may install and use one copy of the EK for the sole purpose of running Applications provided to Licensee by SecuGen.
   c. Licensee may modify the sample source code located in the Products' "samples" directories ("Sample Code") to design, develop and test SecuGen Applications. Licensee may also reproduce and distribute the Sample Code in object code form along with any modifications Licensee makes to the Sample Code, provided that Licensee complies with the distribution requirements described below. For purposes of this section, "modifications" shall mean enhancements to the functionality of the Sample Code.
   d. Licensee may copy and redistribute the Sample Code provided that: (a) it is distributed as part of an Application prepared by Licensee; (b) Licensee's Application adds significant and primary functionality to the Sample Code; (c) the Sample Code only operates in conjunction with the Products; (d) Licensee does not permit further redistribution of the Sample Code; (e) Licensee does not use SecuGen's name, logo or trademarks to mark Licensee's Application; (f) Licensee includes a valid copyright notice on Licensee's Application; and (g) Licensee agrees to indemnify, hold harmless, and defend SecuGen from and against any claims or lawsuits, including attorney's fees, that arise or result from the use or distribution of Licensee's Application.
   e. No other uses and/or distribution of the Products or Sample Code are permitted without SecuGen's prior written consent. SecuGen reserves all rights not expressly granted to Licensee.

2. **Limitations or Restrictions**. Licensee may not: (i) modify or translate the Products; (ii) reverse engineer, decompile, or disassemble the Products, except and only to the extent this restriction is expressly permitted by applicable law; (iii) separate the Products component parts to transfer to a third party; (iv) rent, lease, sell or lend the Products.

3. **Intellectual Property Rights**. Licensee acknowledges that no title to the intellectual property in the Products and any components thereof are transferred to Licensee. All rights, title and copyrights in and to the Products and any copies are owned by SecuGen. The Products are protected by copyright laws and international copyright conventions and treaties.

4. **Warranty**. SecuGen's Product warranty runs to the Licensee for a period of thirty (30) days. If the date of receipt by Licensee of Products is the same as the date of purchase by Licensee of the Products, the thirty day warranty period begins on the date of purchase. If the date of receipt by Licensee of Products is later than the date of purchase of the Products by Licensee, then the thirty day warranty period begins on the date of shipment by a distributor of the Products to Licensee. During the thirty day warranty period, SecuGen warrants that (a) the media on which the Products or documentation covering Products is delivered shall be free from material defects in workmanship and materials, and (b) the Products, when installed, configured, used and maintained in accordance with SecuGen's then-current published installation, configuration, use and maintenance specifications, will, in its unaltered form, conform substantially to SecuGen's then-current published functional specifications for such Products. SecuGen's sole obligation, for a breach of this warranty shall be for SecuGen to replace the media in the case of breach of this Warranty. SecuGen does not warrant that the Products will meet Licensee's requirements, that the Products will operate in combinations selected for use by Licensee or that use of the Products will be uninterrupted or error-free. Because not all errors in software can or need be corrected, SecuGen does not warrant that the Products are error-free or that all software errors will be corrected.

5. **LIMITATIONS OF WARRANTY**. THE PRODUCTS ARE DEEMED ACCEPTED BY LICENSEE. SECUGEN DISCLAIMS ALL WARRANTIES NOT EXPRESSLY PROVIDED IN THIS AGREEMENT INCLUDING, WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. ANY AND ALL RISK ARISING OUT OF THE USE OR PERFORMANCE OF THE PRODUCTS REMAINS WITH LICENSEE. IN NO EVENT SHALL SECUGEN OR ITS SUPPLIERS BE LIABLE FOR ANY CONSEQUENTIAL, INCIDENTAL, DIRECT, INDIRECT, SPECIAL, PUNITIVE, OR OTHER DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR OTHER PECUNIARY LOSS) ARISING OUT OF THIS AGREEMENT OR THE USE OF OR INABILITY TO USE THE PRODUCTS, EVEN IF SECUGEN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES/JURISDICTIONS DO

NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION APPLIES ONLY TO THE EXTENT PERMITTED BY APPLICABLE LAW.

6. **LIMITATION OF LIABILITY**. SECUGEN'S TOTAL LIABILITY AND LICENSEE'S EXCLUSIVE REMEDY UNDER THIS AGREEMENT SHALL NOT EXCEED THE REPLACEMENT COST OF A SINGLE COPY OF THE PRODUCTS.

7. **Indemnification**. Licensee shall hold harmless, indemnify and defend SecuGen, its officers, directors and employees, from and against any claim, suit or proceeding and any losses, damages, fines and expenses (including attorneys' fees and costs) arising out of or relating to any claims that Licensee's use of the Products in conjunction with any Licensee Application infringes the patent, copyright, trademark, trade secret, or other proprietary rights of any third party, or resulting from any breach of this Agreement by Licensee.

8. **Termination**. Without prejudice to any other rights, SecuGen may terminate this Agreement if Licensee fails to comply with any term or condition of this Agreement. Upon termination of this Agreement, Licensee shall immediately discontinue the use of the Products and certify destruction of all full or partial copies of the Products and related materials provided by SecuGen. Licensee may also terminate this Agreement at any time by destroying the Products and all copies thereof.

9. **General**. Licensee acknowledges that he or she has read this Agreement, understands it, and that by using the Products Licensee agrees to be bound by the terms and conditions set forth in this Agreement. Licensee further agrees that the Agreement is the complete and exclusive statement of the understanding between SecuGen and Licensee which supersedes any proposal or prior agreement, oral or written, and any other communication between SecuGen and Licensee relating to the subject matter of this Agreement. This Agreement may not be modified except in writing duly signed by an authorized representative of SecuGen and Licensee. If any provision of this Agreement is held to be unenforceable for any reason, such provision shall be reformed only to the extent necessary to make it enforceable, and such decision shall not affect the enforceability of such provision under other circumstances, or of the remaining provisions hereof under all circumstances.

10. **U.S. Government Restricted Rights**. Use duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (2) of the Commercial Computer Software - Restricted Rights regulation found at 48 CFR 52-227-19, as applicable. Manufacturer is SecuGen Corporation, 2356 Walsh Avenue, Santa Clara California 95051, USA. Licensee acknowledges that the laws and regulations of the United States restrict the export and re-export of the Products. Licensee agrees that the Products in any form will not be exported or re-exported without the appropriate United States and foreign government approval.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY MADE TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED INTO NEW EDITIONS OF THIS PUBLICATION. SECUGEN CORPORATION MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME AND WITHOUT NOTICE.

Contents

Before You Begin

Biometrics Overview

Biometrics is an automated method of recognizing a person based on physical or behavioral characteristics. Biometric information that can be used to accurately identify people includes fingerprint, voice, face, iris, handwriting, and hand geometry.

There are two key functions offered by a biometric system. One method is **identification**, a "one-to-many" matching process in which a biometric sample is compared sequentially to a set of stored samples to determine the closest match. The other is **verification**, a "one-to-one" matching process in which the biometric system checks previously enrolled data for a specific user to verify whether that individual is who he or she claims to be. The verification method provides the best combination of speed and security, especially where multiple users are concerned, and requires a user ID or other identifier for direct matching.

With an increasing reliance on online technology and other shared resources, the information age is quickly revolutionizing the way transactions are initiated and completed. Business transactions of all types are increasingly being handled online and remotely. This unprecedented growth in electronic transactions has underlined the need for a faster, more secure, and more convenient method of user verification than passwords can provide.

Using biometric identifiers offers several advantages over traditional and current methods. This is because only biometric authentication is based on the identification of an intrinsic part of a human being. Tokens such as smart cards, magnetic stripe cards, and physical keys, can be lost, stolen, duplicated, or left behind; passwords can be forgotten, shared, hacked or unintentionally observed by a third party. By eliminating all of these potential trouble spots, only biometric technology can provide the security and convenience needed for today's complex electronic landscape.

Advantages of Using Fingerprints

The advantages of using fingerprints include widespread public acceptance, convenience, and reliability. It takes little time and effort to acquire one's fingerprint with a fingerprint identification device, and so fingerprint recognition is considered among the least intrusive of all biometric verification techniques. Ancient officials used thumbprints to seal documents thousands of years ago, and law enforcement agencies have been using fingerprint identification since the late 1800s. Fingerprints have been used so extensively and for so long, there is a great accumulation of scientific data supporting the idea that no two fingerprints are alike.

About SecuGen

SecuGen provides biometric solutions for physical and network security employing advanced fingerprint recognition technology. The company's comprehensive product line includes quality optical fingerprint sensors and peripherals, software, and development kits used for a variety of innovative applications including Internet, enterprise network and desktop security, physical access control, time and attendance management, and financial and medical records control. SecuGen patented products feature the industry's longest warranty and are renowned for their accuracy, reliability and versatility. Based in Silicon Valley, SecuGen has been serving the biometric community since 1998 and is an active member of the Biometrics Consortium (www.biometrics.org) and the BioAPI Consortium (www.bioapi.org).

Technical Support

Email techsupport@secugen.com
Website www.secugen.com

About SecuGen Products

SecuGen Sensor Qualities

- **Excellent Image Quality:** Clear, distortion-free fingerprint images are generated using advanced, patented optical methods. Quality imaging yields better sampling for minutiae data extraction.
- **Durability:** Mechanical strength tests show resistance to impact, shock and scratches.
- **Powerful Software:** Precise, fast processing algorithm ensures efficiency and reliability.
- **Ruggedness and Versatility:** Solid engineering and superior materials allows for use under extreme conditions.
- **Ergonomic Design:** Compact, modular design for seamless integration into small devices, ease of use, and compatibility make it ideal for a broad range of applications.
- **Low Cost:** Products are developed to deliver high performance, zero maintenance at very affordable prices for general and industrial use.

Advantages of SecuGen Sensors Over Other Optical Sensors

- Unique optical method captures fine details, even from dry skin
- Extremely low image-distortion
- Reinforced materials
- Wear resistance
- Attractively small size
- Ease of integration
- Ready-to-use
- Low cost through longer life and no maintenance requirements

Advantages SecuGen Sensors Over Semiconductor (Capacitive) Sensors

- Non-metal, non-silicon components make it less susceptible to corrosion when exposed to salts, oil and moisture from skin and environment
- Superior surface properties eliminate need for costly coating and processing procedures
- Greater mechanical strength, wear-resistance, and durability
- Broader range of applicability, especially for use in extreme conditions and climates
- Immunity from electrostatic discharge
- Low cost through longer life and no maintenance requirements

Strengths of SecuGen Software and Algorithms

- Unique image processing algorithm extracts fingerprint minutiae very accurately
- High signal-to-noise ratio processing algorithm screens out false features
- Highly efficient matching algorithm
- Fast overall process of extraction, matching and verification
- Encryption function to protect user privacy
- Compatibility with existing desktop, laptop PCs interface computers
- Ease in developing applications for various purposes

Chapter 1. Product Overview

The SecuGen® SDA04 is an advanced stand-alone fingerprint recognition device with onboard CPU, SDRAM and flash memory. It uses the SecuGen FIPS201 certified fingerprint sensor and the NIST MINEX certified ANSI INCITS 378-2004 fingerprint algorithms. The SDA04 consists of a processor boardt [Fig. 1] and an advanced fingerprint optic module, SDOPP04 [Fig.2]. The standard SDA04 includes 128MB DDRAM and 128MB NAND FLASH memory capable of storing fingerprint data for up to 3000 registered system users.

Commenté [d1]: Update when 1:10000 tested

The SecuGen® SDA04M uses the same processor board as the SDA04, but uses the SDOPP03M fingerprint sensor and SecuGen's FDAxx fingerprint algorithms.

Sensors based on CMOS technology are frail in comparison with the SecuGen optic module, which is renowned for its ability to withstand extreme environmental conditions, heavy use as well as external shock, scratches and impact. This makes the SDA04 an ideal solution for all outdoor applications. Moreover, the compact size of its components make the overall device smaller than most other optics-based fingerprint recognition devices, and for that reason SDA04 can be integrated into a wider range of applications than other fingerprint devices.

Features

- *Smart Capture* allows fingerprint recognition under very bright conditions, such as sunlight outdoors.
- *SecuSearch Engine* provides high-speed fingerprint identification for SG400 templates.
- *Standard 128 Mbytes FLASH* provides a large capacity for fingerprint storage - up to 3000 templates i n identificaton mode and 3000 templates in verification mode
- *FIPS201 compliant fingerprint sensor*
- *NIST MINEX Certified ANSI INCITS 378-2004 fingerprint algorithms*

Commenté [d2]: Confirm final numbers

1.1. Features

The SDA04 Developer Kit (DK) runs on Windows 2000, XP and Vista platforms, helping programmers build

reliable SDA04 applications quickly and easily. The optical fingerprint reader is distinguished by SecuGen's patented technologies that allow rapid capture of high-contrast, low-distortion fingerprint images. An advanced fingerprint recognition algorithm extracts random feature points (also called minutiae) then encrypts and stores them as mathematical templates in flash memory. When a comparison fingerprint image is captured, it may be compared with any previously stored template for matching and identification/verification of the user's identity.

- Durability
- High Image Quality
- Small Size
- User-friendly

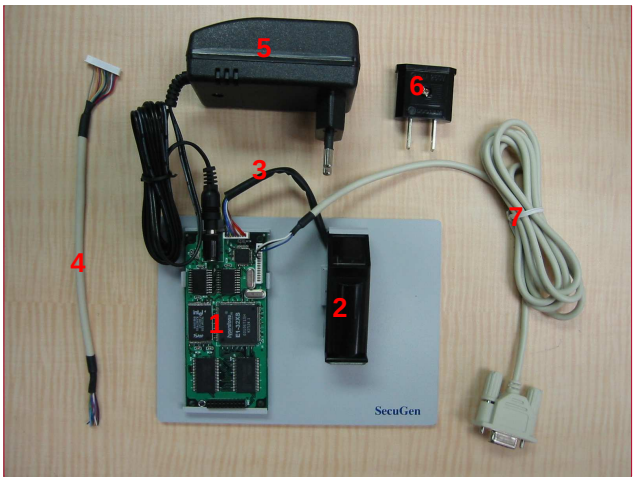
1.2. Applications

The SDA04 is designed to satisfy today's growing need for security applications in a wide variety of fields including:

- Physical access control system (PACS)
- Door-lock system (indoor/outdoor)
- Time & attendance system
- Safety deposit box
- Vehicle access control
- Automatic teller machines (ATM)
- Point-of-sale machines (POS)
- Verification of credit cards/smart cards
- Secure network system

1.3. SDA Developer Kit Contents

The SDA DK consists of hardware and the software needed to integrate the SDA04 into a wide variety of "embedded fingerprint" applications. All programs in the DK run on Microsoft Windows 2000 and XP platforms.



Commenté [d3]: Need updated picture of DK contents

1. SDA04 board with CPU, DDRAM (128MB) and NAND Flash Memory (128MB)
2. SDOPP04 fingerprint optic module
3. 20 PIN ZIF ribbon cable, Molex 52746-2090 (SDA04 board <->SDOPP04)
4. C4 cable, 11-wire, 15-pin connector (SDA04 board <-> host controller)
5. Power Adaptor (110V~220V AC, 5V DC)
6. Wall power converter
7. C3 cable, serial DB9 connector (SDA04 board <-> PC)
8. SDA04 DK CD

Commenté [d4]: Confirm contencts.

- SDATest Demo Program (Windows 2000/XP/Vista)
- Sample database with 2995 fingerprint templates
- Documentation
- C# Sample Source Code (Windows XP/Vista)

Chapter 2. SDA DK PC Demo Programs

2.1. SDATest DK Demo for Windows – Fingerprint Controller

The SDATest Fingerprint Controller is included in the Developer's Kit, and runs on Windows 2000 or XP. It uses a sample program to demonstrate the basic fingerprint controls and functions of SecuGen's SDA03 and SDA04 technology.

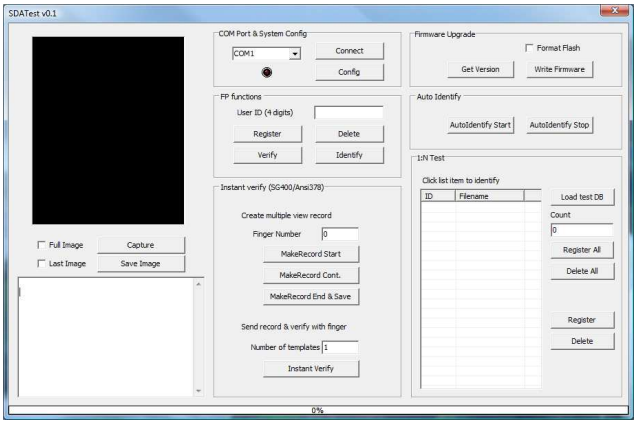
To run the SDATest program, do the following:

- Use a computer with Windows 2000, XP or Vista installed.
- Copy SDATEST.EXE and FP3000.TDB from the DK CD to a directory on your computer
- Double-click SDATEST.EXE.

The SDATest Fingerprint Controller performs the following functions:

- Initializes connection to external systems (e.g. a PC)
- Registers, verifies, deletes and updates users
- Sets system configuration

Fingerprint controller (SDATest.exe) User Interface



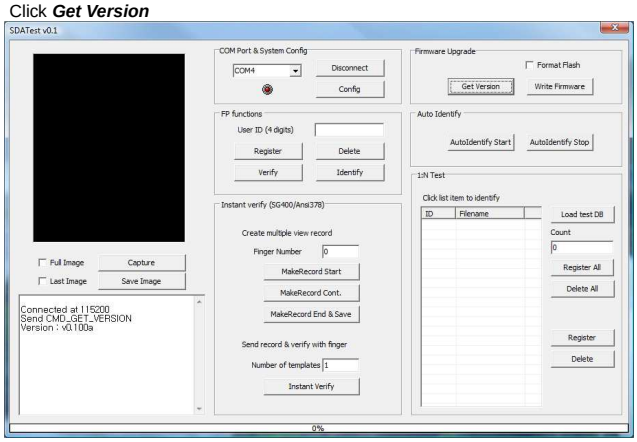
2.1.1. Initializing Connection

The first step performed by the SDA04 Fingerprint Controller is initializing communication between the SDA04 and another computer system. The serial port and communication speed must be selected to start the program. If initialization is successful, the program proceeds to the next step. If initialization is not successful, check the serial port selection and try again. After changing the serial port connection (e.g. from COM1 to COM4), the communication speed will be automatically detected and reset to the appropriate value so there should be no need to select it.

Serial Port Setting – Choose the serial port the SDA04 is connected to (COM1, COM2, COM3 or COM4).

Communication Speed – The SDA04 and SDA04M can communicate at speeds of 1200 to 921600bps. The SDA03M can communicate at speeds of 1200 to 460800bps. The default on initial power on is 115200 bps.

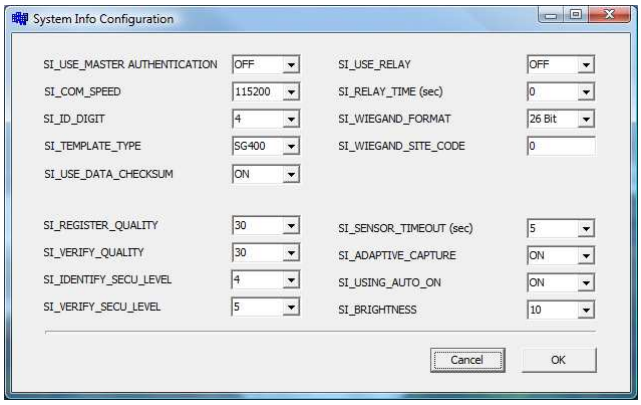
1. Click ***Connect***
   You should receive the message "Connected at 115200
2. Click ***Get Version***



2.1.2. System Configuration

Several system configuration settings can be changed by administrators, including brightness, security level, communication speed, master verification, and smart capture. All of these controls are found in the Configuration Dialog window below.

1. Click ***Config***



Refer to Appendix A for a detailed description of these parameters and how they affect the functionality of SDA04.

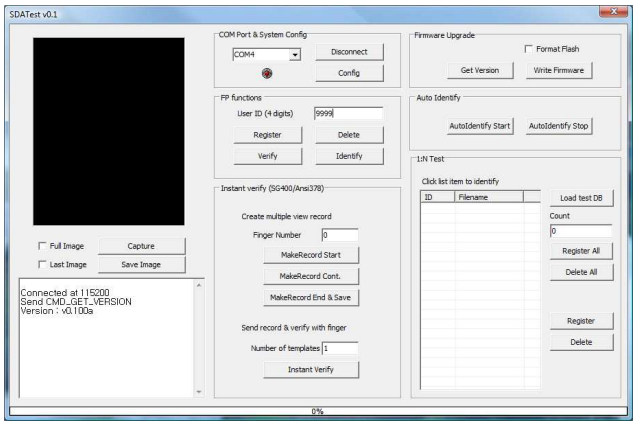
2.1.3. User Registration

User registration, or enrollment, is the process by which new users are added to the SDA04 database. During registration, two fingerprint images from the same finger are captured and stored in the SDA04 Flash for future verification and identification operations. The user is required to release the finger from the sensor momentarily before the second (final) capture is performed.

Tips for Using SecuGen Fingerprint Devices

- Cover the fingerprint sensor completely with the finger to capture the best image.
- Wait for the red LED light inside the fingerprint scanner to turn on before applying fingerprints – this ensures the device is activated.
- Don't press too hard. The amount of pressure required is only as much as it takes to hold a piece of paper between your thumb and finger – not much.
- Don't move the finger during scanning.

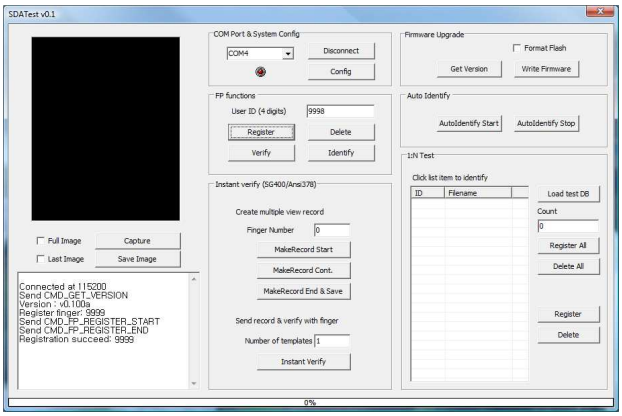
1. Enter User ID – example 9999



2. Click ***Register***

3. Place finger on sensor and then click ***OK*.** Fingerprint is captured.

4. Remove finger and place on sensor again to capture second sample. Click ***OK***.



Note: Two fingerprints from the same finger must be captured to ensure accuracy and reliability during the registration (enrollment) process.

2.1.4. User Verification and Identification

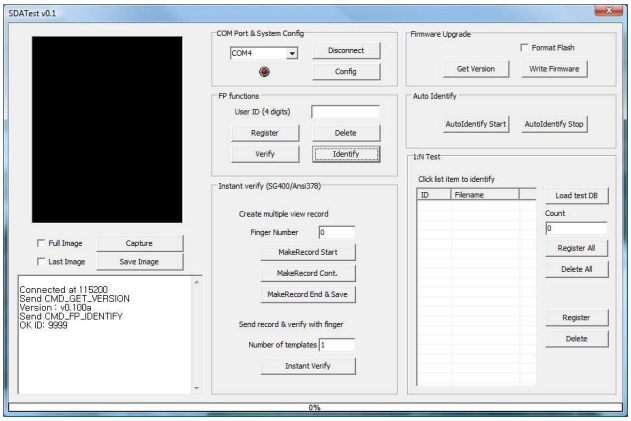
This function compares a captured (input) fingerprint to the fingerprints currently registered in the system database and may perform either of two different operations: identification or verification.

- ***Identification*** is a one-to-many (1:N) matching operation that does not require a User ID as input. This function compares a captured fingerprint to the registered fingerprints stored in the system to determine the eligibility of a user. Identification works well with relatively small databases. The SDA04 uses the SecuGen SecuSearch engine enables extremely fast identification speeds.

- ***Verification*** is a one-to-one (1:1) matching operation that requires a User ID as input to find a specific user's fingerprint.

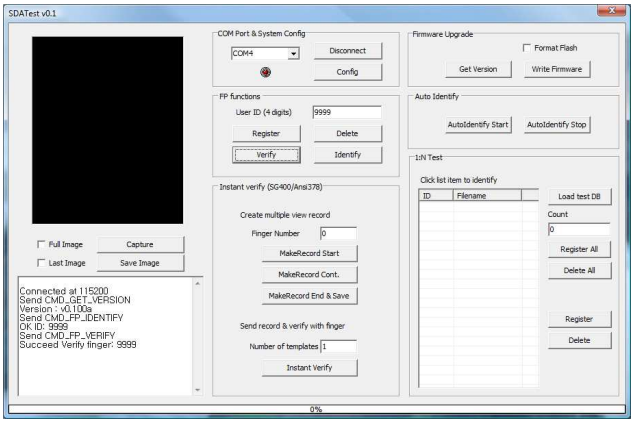
Example A. Identification (1:N matching)

1. Place finger on fingerprint sensor
2. Click ***Identify***
3. The User ID of the user that was identified will be returned



Example B. Verification (1:1 matching)

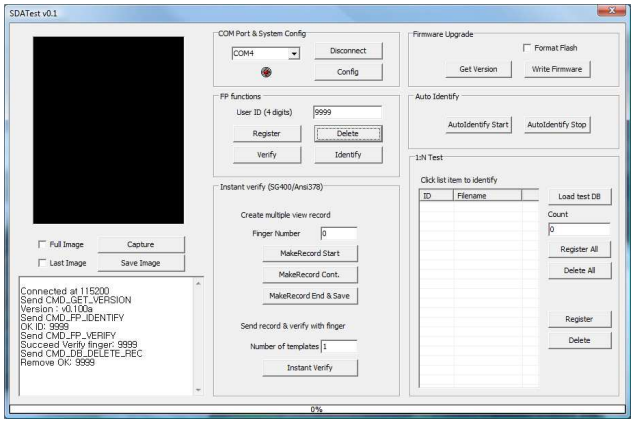
1. Enter the User ID.
2. Place fingerprint on sensor
3. Click ***Verify***



2.1.5. User Deletion

This function is used to delete previously registered users.

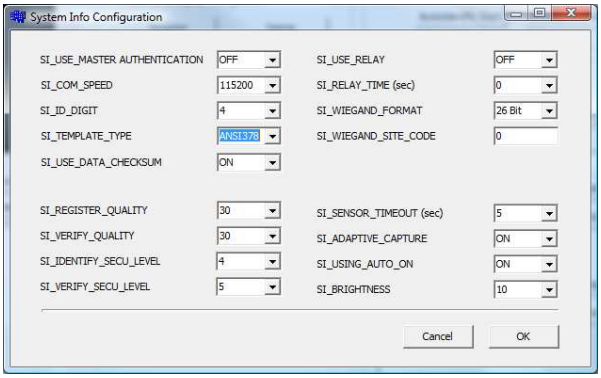
1. Enter the User ID.
2. Click ***Delete***.



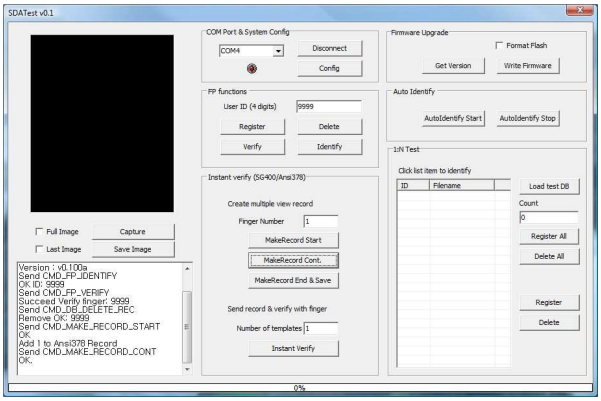
3. To delete all registered users from the user database at once, click ***Delete All***. In this case you do not need to specify a User ID, but a precautionary confirmation dialog will appear before all users are deleted.

2.1.6. Generate Multi-View ANSI378 Template

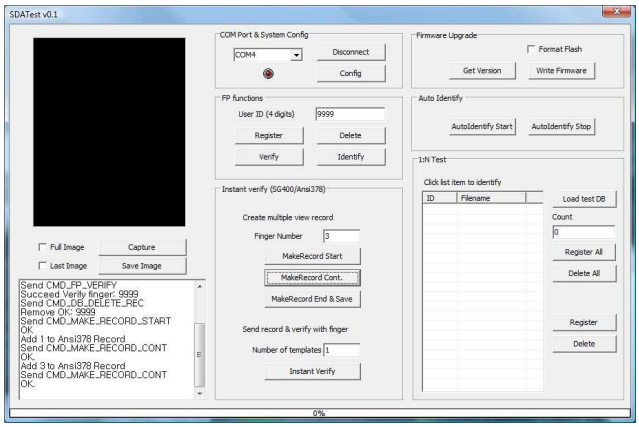
1. Click ***Config***
2. Set SI_TEMPLATE_TYPE to ANSI378



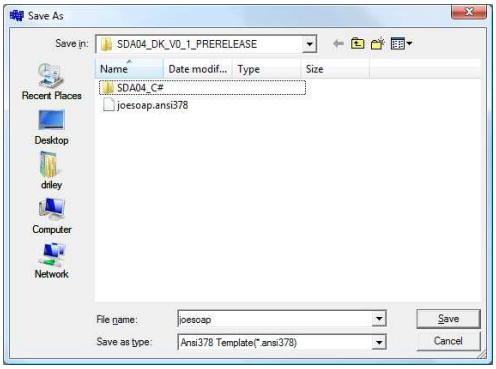
3. Click ***OK***
4. Click ***MakeRecord Start***
5. Enter finger number for finger view 1
6. Place finger1 on sensor. Fingerprint sensor will capture sample of finger 1
7. Click ***MakeRecord Continue***
8.



8. Enter finger number for finger view 2
9. Place finger2 on sensor
10. Click ***MakeRecord Continue***



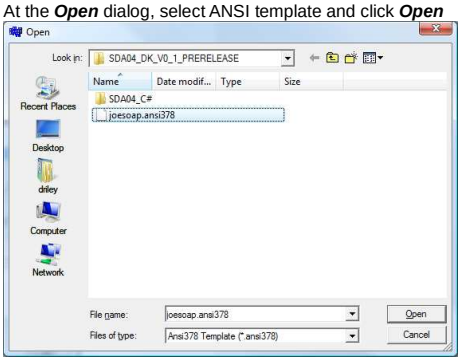
11. Click ***MakeRecord End & Save***



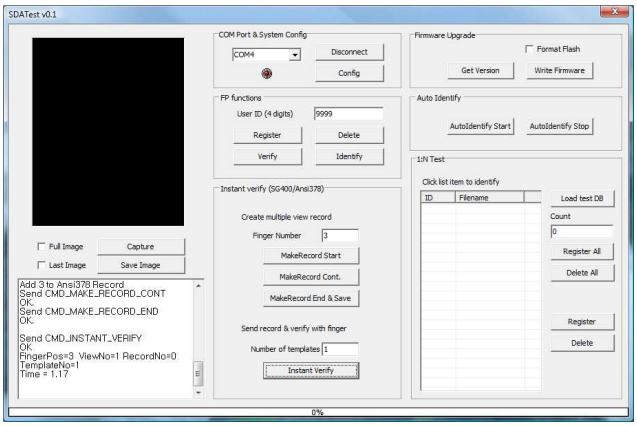
12. Enter username and click ***Save*** to write record to disk

2.1.7. Verify ANSI378 Template

1. Place finger on sensor
2. Click ***Instant Verify***
3. At the ***Open*** dialog, select ANSI template and click ***Open***



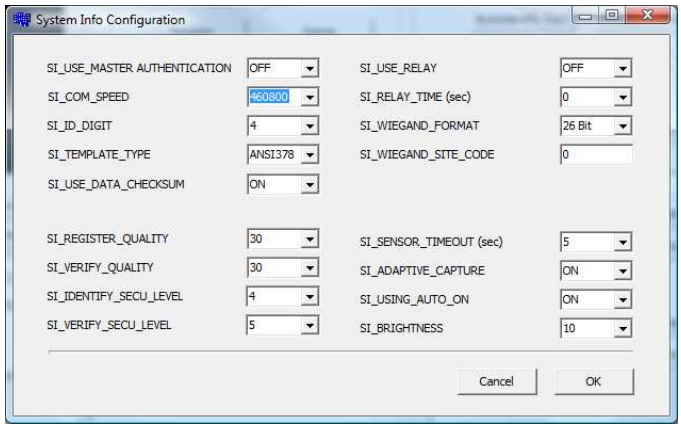
4. Live fingerprint will be extracted and matched against the ANSI record. Finger position and view will be reported in the result.



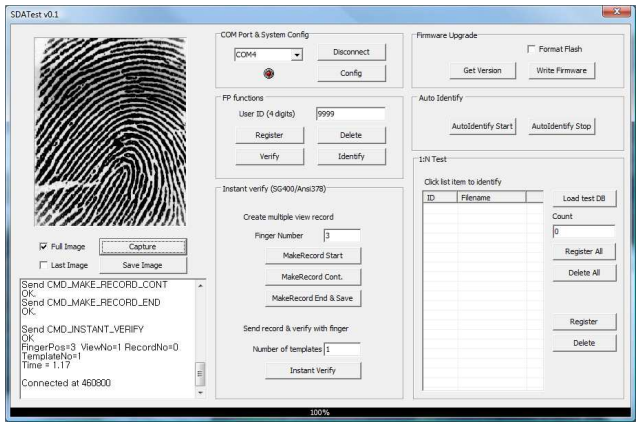
2.1.8. Capturing and Storing Fingerprints

When capturing a full size image, it is preferable to use high speed serial communications whenever possible to ensure fastest possible image transfer.

1. Click Config and set baud rate to 460800bps, then click **OK**



2. Select ***Full Image***
3. Place finger on sensor
4. Click ***Capture***

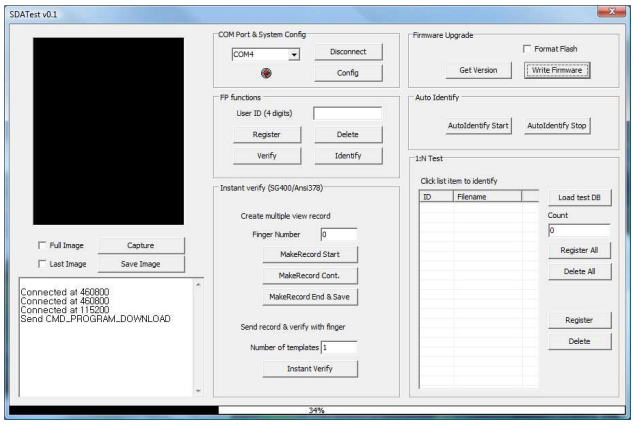


5. If you wish to store the fingerprint image, click ***Save Image*** to save the image as a TIF file.

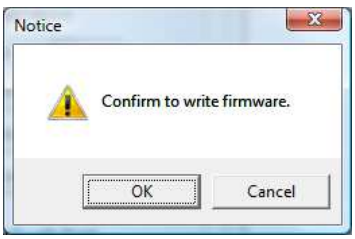
2.2. SDA04 Firmware Upgrade

1. Power on SDA04 and connect at default baud rate of 115200bps isomg SDATest.
2. Click ***Write Firmware***

3. At the Open dialog, select the firmware version you wish to load and click ***Open***
4. Progress bar will indicate firmware transfer status.



5. When the firmware has been downloaded a confirmation dialog will be displayed.



6. Click ***Cancel*** to abort the firmware update.
7. Click ***OK*** to write the firmware to the SDA04 FLASH.

Note: A firmware update will not delete the users that are registered in the SDA04 database.

2.3. SDA04 SecuSearch 1:N Sample Database

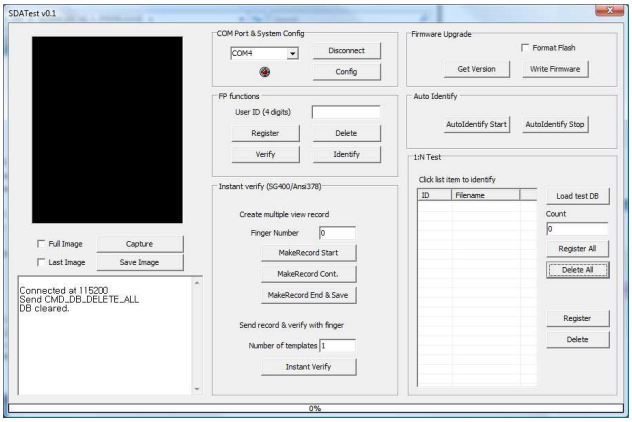
Commenté [d5]: Rework for SDA04

SDA04 DK is shipped with a sample database containing 2995 fingerprint records. This database is provided for the purpose of testing the high speed search engine.

Note: This section is for testing purposes only. All fingerprint data on the SDA04 will be erased during the database load process.

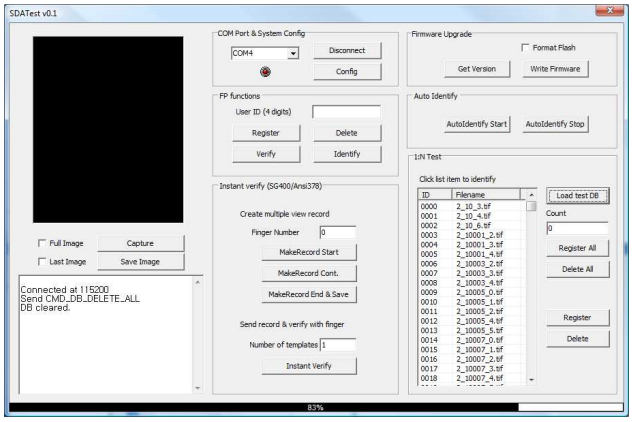
2.3.1. Loading the test database

1. Power on SDA04.
2. Launch SDATest and connect to SDA04 at the default of 115200bps
3. Click ***Delete All*** to erase all fingerprint data on the SDA04

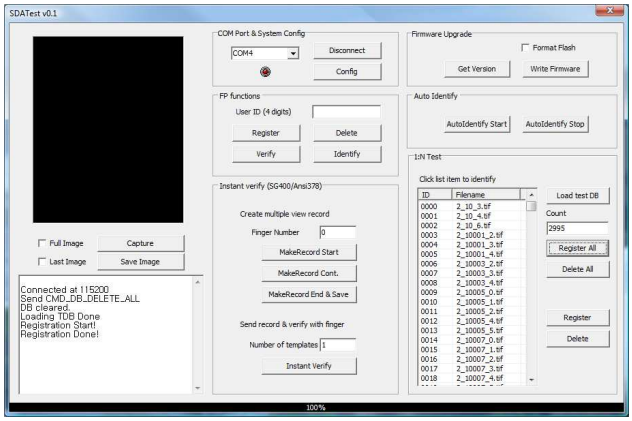


4. Click ***Load Test DB*** and select fp3000.tdb. Click ***Open***.

5. 2995 Fingerprint templates will be loaded into SDATest memory in preparation for writing to the SDA04.

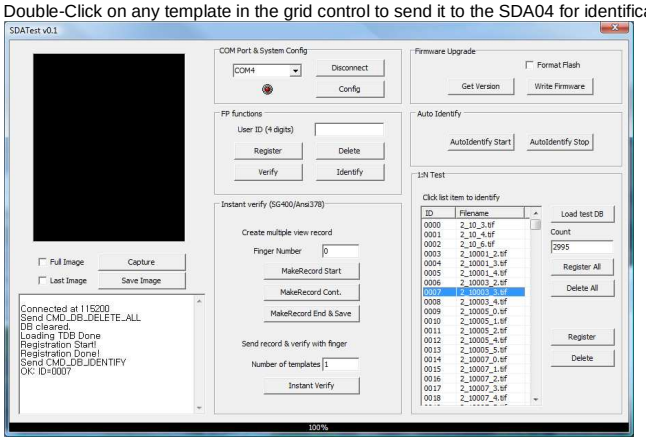


6. Click ***Register All*** to write the 2995 templates to the SDA04.At the speed of 115200bps, it takes just over four minutes to build the 3000 user database.

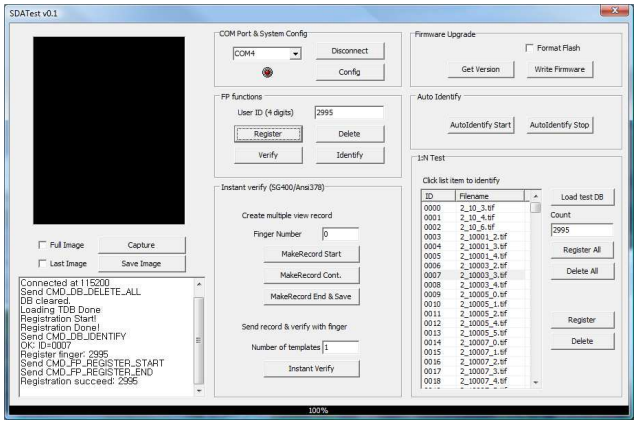


2.3.2. Identifying fingerprints in the test database

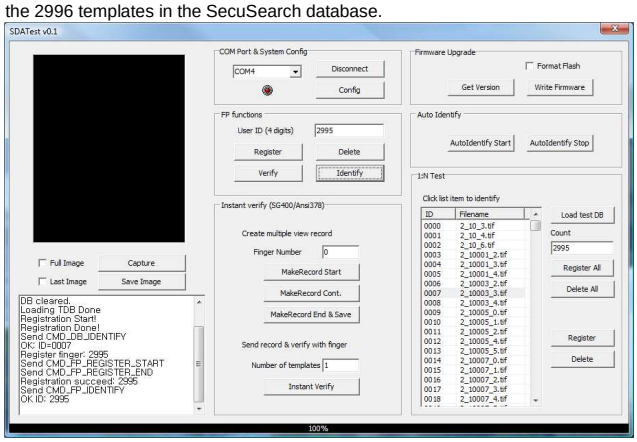
1. If the grid control is not already populated, load the test database into SDATest memory as described in section 2.3.1. Do not write the templates to the SDA04.
2. Double-Click on any template in the grid control to send it to the SDA04 for identification.



3. Since only only 2995 templates are loaded, it is advisable to register your own fingerprint to test identification with the fingerprint capture. Click Register and follow the registration process as described in *Section 2.1.3* above



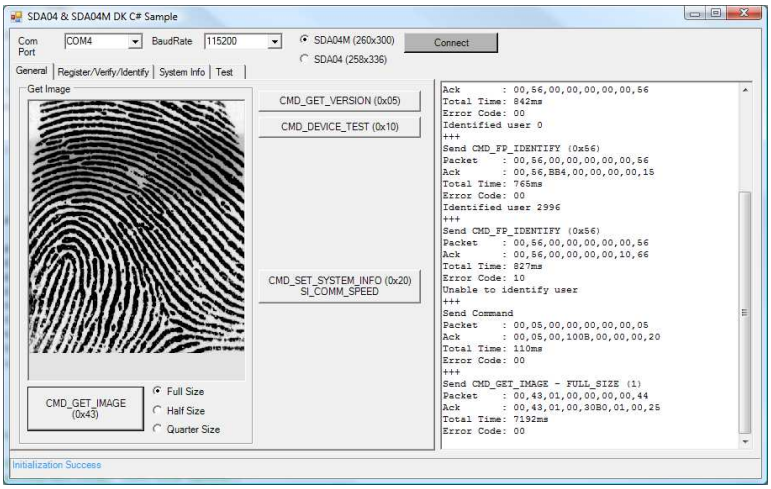
4. Place your finger on the sensor and click ***Identify***. Total time for the identification operation with 3000 templates is less than 3 seconds for fingerprint capture, minutiae extraction and identification against the 2996 templates in the SecuSearch database.

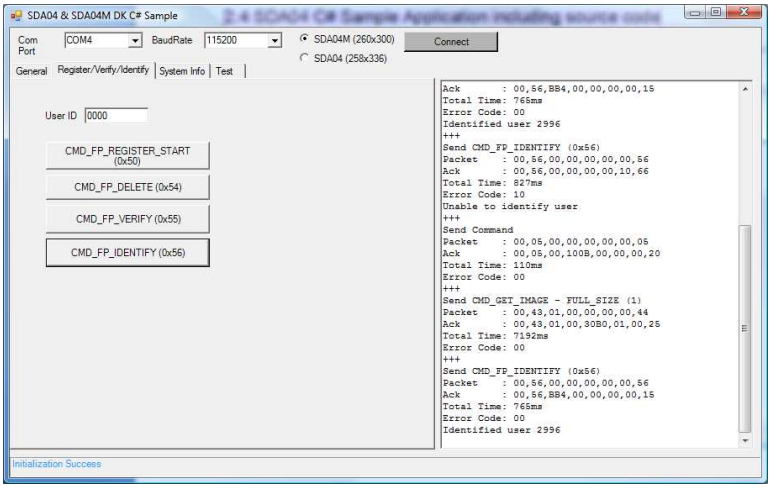


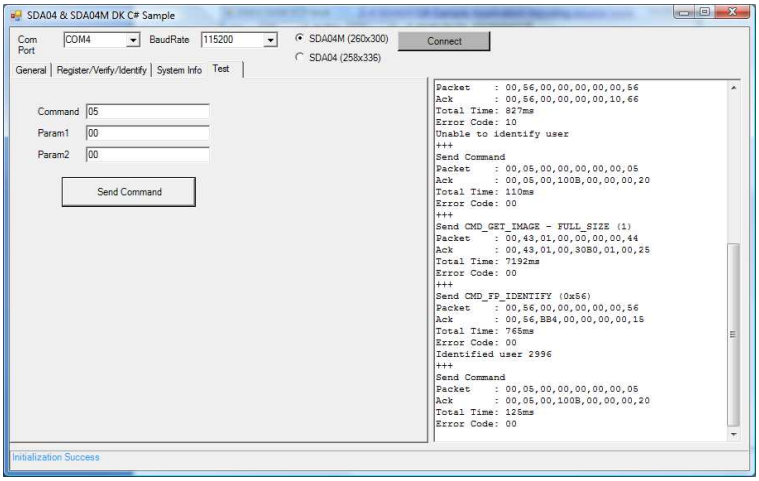
2.4 SDA04 C# Sample Application including source code

Commenté [d6]: Complete this section

The SDA04 DK Contains sample source code for a host controller implemented in C#. This application can be built on both Windows XP and Windows Vista using Microsoft Visual Studio 2005 or later.






SDA04 & SDA04M DK C# Sample
Com Port
COM4
BaudRate
115200
SDA04M (260x300)
SDA04 (258x336)
Connect
General
Register/Verify/Identify
System Info
Test
Command
05
Param1
00
Param2
00
Send Command
Packet : 00,56,00,00,00,00,00,56
Ack : 00,56,00,00,00,00,10,66
Total Time: 827ms
Error Code: 10
Unable to identify user
+++
Send Command
Packet : 00,05,00,00,00,00,00,05
Ack : 00,05,00,100B,00,00,00,20
Total Time: 110ms
Error Code: 00
+++
Send CMD_GET_IMAGE - FULL_SIZE (1)
Packet : 00,43,01,00,00,00,00,44
Ack : 00,43,01,00,30B0,01,00,25
Total Time: 7192ms
Error Code: 00
+++
Send CMD_FP_IDENTIFY (0x56)
Packet : 00,56,00,00,00,00,00,56
Ack : 00,56,BB4,00,00,00,00,15
Total Time: 765ms
Error Code: 00
Identified user 2996
+++
Send Command
Packet : 00,05,00,00,00,00,00,05
Ack : 00,05,00,100B,00,00,00,20
Total Time: 125ms
Error Code: 00
Initialization Success

Chapter 3. Order of Operations

Commenté [d7]: Not sure what this section is for. Perhaps simple guidelines.

3.1. Main Controller

3.2. Main Controller Function

3.2.1. Enrolling User Fingerprints (Register User)

3.2.2. Removing User (Delete)

3.2.3. Changing Fingerprint (FP Change)

3.2.4. Enrolling Master (Register Master)

3.2.5. Deleting Database (DB all clear)

3.3. Main Controller Configuration

3.3.1. Auto Tuning

3.3.2. Setting Exposure

3.3.3. Setting Gain

3.3.4. Setting Security

SDA04 Security Level Basics

Before deploying SecuGen fingerprint recognition devices, administrators should understand the basic concepts behind security levels and their effects on the performance of the fingerprint recognition functions. There will always be a trade-off between user convenience and security.

For example, when you increase the security level, you may experience a higher rate of false rejections (FRR). False rejection occurs when an authorized (registered) user's fingerprint is not successfully matched against the stored sample because more minutiae data (fingerprint characteristics) are required for a match. The increased security level raises the matching threshold, which results in a more discriminating fingerprint recognition system.

On the other hand, when you decrease the security level, you may experience a higher rate of false acceptance (FAR). False acceptance is potentially more serious because this means an unauthorized user is granted access.

The consequence of a "slightly higher security level" could be a slight increase in FRR, which is usually nothing more than an inconvenience to users who should re-attempt to match their fingerprints. But the consequence of a "much higher security level" could be a great increase

in FRR, which could pose a hindrance to the smooth operation of a fingerprint recognition system. In high security environments where the security level is raised, users will need to be more attentive to physical technique when placing their fingers on the sensor, and they may require basic education regarding environmental considerations such as bright light and moisture and their possible effects on fingerprint capturing. In summary, where FRR can be irritating, the net results of FAR can be far more serious if critical resources are at stake.

SecuGen devices use *security levels* to fine-tune the fingerprint matching process, allowing different installations to emphasize security over convenience, or vice versa, depending on the requirements of a site. Security levels range from 1 (Lowest) to 9 (Highest). The default setting of SecuGen devices depends on whether the system is set to use *identification* or *verification* as the mode of authentication. For example, the security level in SecuGen's "Identify" mode may range from 6-9, with a default of 8. The default matching mode in SecuGen devices is "Verify," which uses input fingerprint minutiae plus some other data, such as User ID, PIN, or smart card, to quickly compare live fingerprint data (1:1 match) with the enrolled/stored data corresponding to the User ID, PIN or smart card, effectively proving the user's proclaimed identity before granting access. The default security level for verification is 5, or "Normal." Lower security settings may be appropriate for applications with many users, such as time-and-attendance systems at large-scale sites where high security is not considered as critical as speed and convenience.

Note: FAR and FRR are directly and inversely proportional, so it is important to establish security policies based on a full understanding of this fact.

3.3.5. Getting Configuration

3.3.6. Getting Version Information

3.3.7. Getting Total Number of Enrolled Users and Masters (get reg CNT)

3.4. Special Feature in the Main Controller.

3.4.1. Deleting Database without Master Fingerprint (Shadow)

Chapter 4. Physical Specifications

4.1. SDA04 and SDA04M

Please refer to the SDA04/SDA04M datasheet for physical specifications

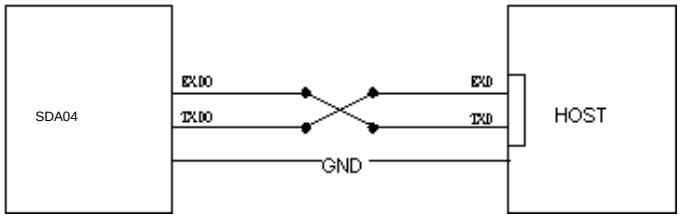
4.2. SDA03M

Please refer to the SDA03M datasheet for physical specifications

Chapter 5. Connecting SDA04 to External Devices

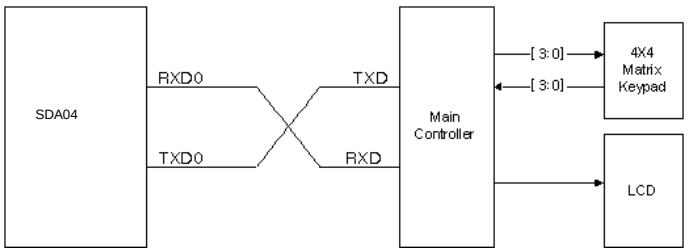
5.1. Connecting to a Host PC

When using a PC to develop or debug applications, the TTL level or RS232 level serial port can be used to connect the SDA04 to the host PC's serial port. The default baud rate is 115200bps. The SDA04 and SDA04M support baud rates from 1200 to 921600 bps. SDA03M supports baud rates from 1200 to 460800 bps.



5.2. Main Controller

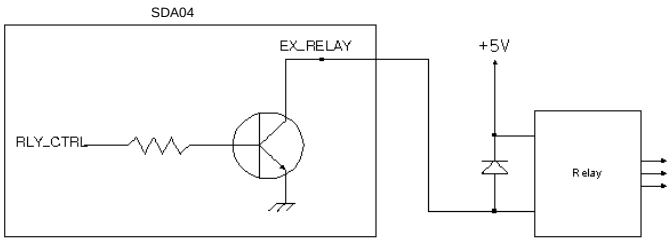
The main controller can use a normal microcontroller with serial ports such as Z80 and 8051. When the main controller sends a command according to a defined protocol, the SDA04 Processing Unit analyzes and performs the command. Serial port 0 is used for communication between the two components.



5.3. Relay Operation

The SDA04 has built-in relay circuits connected by the open collector (see figure below). The electric current required to operate the relay varies according to the resistor of the transistor base. Currently, the resistor is set at about 5V 100mA. The relay is operated by command signals from the host PC or the main controller's serial port.

Commenté [d8]: Check this

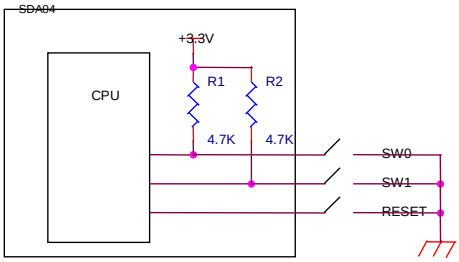


5.4. External Reset Signal

When the power is on, the CPU is automatically initialized. However, when an error occurs during operation, it may be necessary to initialize manually. To perform manually initialize the CPU, press the reset button. For applications requiring initialization from an external program, make sure the EX_RESET# signal is connected to GND.

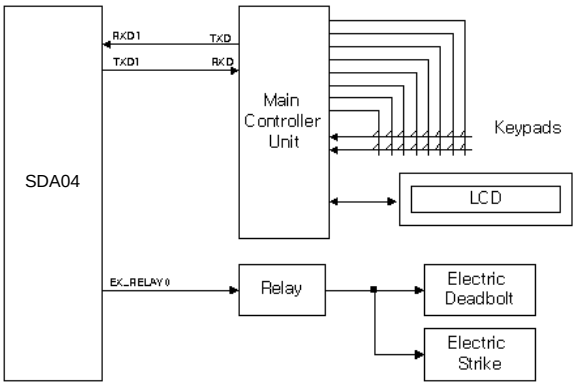
5.5. External Switch Input

The SDA04 receives input using two external switches.



- SW0, SW1: general purpose input
- RESET: external reset input
- Output Port (open collector type): 5V, 100mA

5.6. Example: Key Lock System



This example of a key lock system has a 4x4 matrix keypad and LCD user interface. A main controller supports serial port (Z80, 8051). A normal one-chip microcontroller is sufficient to support the keypad and LCD. The SDA04 communication speed default is 115200 bps with a 12-byte command packet format.

Chapter 6. SDA04 Packet Structure

Communication between the built-in CPU and the main controller is performed through the serial port. The communication speed can be set to 1200, 2400, 9600, 19200, 38400, 57600, 115200, 230400 or 921600 bps. SDA03M supports speeds up to 460800 bps.

All commands are transmitted from the main controller to the SDA04, which then sends acknowledgements for each command back to the main controller using the same command packet format. For the ACK packet, the SDA04 transmits the result of a wrong command as an error code in the ErrorCode field. If the command has been executed properly, the ErrorCode value will be 0 (M2ERROR_NONE).

Note: For a complete list of all available commands and descriptions of each, please refer to Appendix A, *Protocol Reference*.

6.1. Packet Structure

Communication between the SDA04 Processing Unit and the main controller is performed using P1 and P2 type packets. The size of the fields (in bytes) and their brief descriptions are listed below. In order to exchange data over 4 bytes, the lwExtraData, hwExtraData fields should be used. The SDA04 main controller should receive data of the length specified in the ExtraData field since the value in the ExtraData field indicates this. (For more detailed descriptions of each command, refer to Appendix A.)

6.1.1. P1 Packet

1 Byte	1 Byte	2 Bytes	2 Bytes	2 Bytes	2 Bytes	1 Byte	1 Byte
Channel	Command	Param1	Param2	lwExtraData	hwExtraData	ErrorCode	Checksum

6.1.2. P2 Packet (P1 with Extra data)

P1 Packet	data	CS1

Data length is defined by lwExtraData and hwExtraData in the P1 packet area. Normally, Extra checksum is disabled. Only after enabling this option is the extra checksum added and sent to the host controller. The host controller also has to send an additional checksum byte when it enables the extra checksum function in the SDA04. Use the option SI_USE_DATA_CHECKSUM (0x10) in the command CMD_SET_SYSTEM_INFO (0x20).

The values of the first 11 bytes of the packet (excluding the checksum byte) are summed and then divided by 0x0100 (256). This will create a 1-byte shift. The remaining value from this shift is the checksum byte. Checksum can be used to assure correct data transfer for remote registration.

Channel	Currently not in use
Command	Command field
Param1	The first parameter where data is to be transmitted (if any)

Param2	The second parameter where data is to be transmitted (if any)
lwExtraData	When more than 4 bytes of data are transmitted, this is the data length low word value
hwExtraData	When more data is transmitted, this is the data length high word value
ErrorCode	The executed result of the command
Checksum	Used for communication error detection

In the **Command** field, the command that the SDA04 Processing Unit executes is specified. Some commands require a parameter to transmit data, and in this case, this is done through Param1 and Param2. If more data is transmitted in addition to Param1 and Param2 (more than 4 bytes data), the **lwExtraData** and **hwExtraData** fields are used. The low word value of the length of data is stored in **lwExtraData** field, and the high word value in **hwExtraData**. The **Checksum** field is used to detect communication errors. The values of the first 11 bytes (excluding the checksum byte) are summed and then divided by 0x0100 (256). This will create a 1-byte shift. The remaining value from this shift is the checksum byte.

6.2. How Packets Are Used

All commands are transmitted to the SDA04 from the main controller. The SDA04 replies to each command with an acknowledgement using the same format as the command packet. (The command packet is sent to the SDA04 from the main controller, and the ACK packet is sent to the main controller from the SDA04.) The SDA04 sends the result of the command in the ErrorCode field of the ACK. If execution of the command is completed without error, the ErrorCode is '0' (M2ERROR_NONE).

Example
The following representation shows the command packet when using the command **CMD_FP_VERIFY (0x55)** to verify a user whose User ID is 1234. The command is 0x55, and Param1 is 0x1234.

Channel Command Param1

0x00	**0x55**	**0x1234**	0x0000	0x0000	0x0000	0x00	Chksum

As mentioned previously, during word transmission the high word value is transmitted followed by the low word value. Therefore, the byte sequence of actual transmission is as follows:

0x00 → 0x55→ **0x34** → **0x12** → 0x00 → 0x00 → 0x00 → 0x00 → 0x00 → 0x00 → 0x00

In most cases, one command packet executes one function. But in some cases several command packets are needed to complete a function. Commands such as Master Register, User Register, Change of User Fingerprint, User Delete and System Mode Change, among others, require master authentication first and can only be executed independently if master authentication is disabled by CMD_SET_SYSTEM_INFO.

6.3. Commands

Commands are classified into one of the following four categories depending on the parameter.

- Without parameters
- Transmitting values to SDA04
- Obtaining values from SDA04
- Exchanging data over 4 bytes in size

► *Without parameters*

CMD_GET_VERSION

This command is used to check the current version of SDA04.

Command Packet from the Main Controller

0x00	0x05	0x0000	0x0000	0x0000	0x0000	0x00	Chksum

The SDA04 receives and executes the CMD_GET_VERSION command and then sends out CMD_GET_VERSION in the command field and the execution results in the ErrorCode field all in the same packet format. If the command has been properly executed, the ACK packet from the SDA04 to the main controller will be as follows:

ACK Packet of SDA04 (Major, Minor Version Number in Param1 & Param2)

0x00	0x05	0x0000	0x100A	0x0000	0x0000	0x00	Chksum

► *With parameters*

CMD_RELAY_ONOFF

This command controls use of the relay function. The main controller sends out the relay number to Param1 of the command packet, and determines whether or not to send the relay function to Param2, then to the SDA04. The SDA04 executes the command after receiving this command packet, and then returns the execution result in the ErrorCode field of the ACK Packet.

The example below opens relay No. 1 (0x0001).

Command Packet of the Main Controller

0x00	**0x12**	**0x0001**	**0x0001**	0x0000	0x0000	**0x00**	Chksum

ACK Packet of SDA04 (Relay No. and Status in Param1 & Param2)

0x00	**0x12**	0x0001	0x0001	0x0000	0x0000	**0x00**	Chksum

► *Getting parameters*

When getting parameters, the SDA04 sends information requested from the main controller into Param1 or Param2.

CMD_GET_SYSTEM_INFO

This command collects current SDA04 system configuration values. The example below retrieves the security level.

After being given CMD_GET_SYSTEM_INFO (0x30) in the command field of the command packet, and SI_VERIFY_SECU_LEVEL in the Param1 field, the SDA04 shows the current security level in Param2 of the ACK packet. The security level is now set at Normal (0x05).

Command Packet of the Main Controller

0x00	**0x30**	**0x0002**	0x0000	0x0000	0x0000	0x00	Chksum

ACK Packet of SDA04 (Security Level Return in Param2)

0x00	**0x30**	**0x0002**	**0x0005**	0x0000	0x0000	0x00	Chksum

► *Exchanging data over 4 bytes in size*

In order to exchange data over 4 bytes, the lwExtraData, hwExtraData fields should be used. The SDA04 main controller should receive data of the length whose value is specified in the ExtraData field.

CMD_GET_IMAGE

This command retrieves the captured SDA04 fingerprint image data. The main controller determines the size of the image to be acquired (Param1). The possible values are VIEW_NORMAL (0x01, the original image size), VIEW_HALF (0x02, half size of the original), and VIEW_QUARTER (0x04, one-fourth size of the original). The size of the data depends on the size of the image chosen. It reads the length of data from the ExtraData field of the packet received from SDA04 then receives the correspondingly sized image.

Command Packet of the Main Controller

0x00	0x43	**0x0001**	0x0000	0x0000	0x0000	0x00	Chksum

ACK Packet of SDA04 (Original Image size (0x152a0 = 86688 = 258 x 336)
>> NOTE: SDA04M image size is 0x130B0 = 78000 = 260x300)

0x00	0x43	0x0001	0x0000	**0x52a0**	**0x0001**	0x00	Chksum

After sending the ACK packet above, the SDA04 transmits the image data back to the main controller.

Chapter 7. System Settings

7.1. Checking Firmware Version

In order to determine the SDA04 firmware version, use the CMD_GET_VERSION command. The example below uses CMD_GET_VERSION to show the current firmware version of SDA04.

Command Packet

0x00	0x05	0x0000	0x0000	0x0000	0x0000	0x00	Chksum

ACK Packet

0x00	0x05	version	0x0000	0x0000	0x0000	0x00	Chksum

7.2. Checking Device Status

Check the sensor status or the flash memory status of the SDA04 by using the CMD_DEVICE_TEST command.

Command Packet (0x0001 = Sensor Test)

0x00	0x10	0x0001	0x0000	0x0000	0x0000	0x00	Chksum

ACK Packet

0x00	0x10	0x0001	0x0000	0x0000	0x0000	0x00	Chksum

7.3. System Settings

To change the system settings, use the CMD_SET_SYSTEM_INFO command. Items to be changed are written in Param1, and the new value of each item to be changed is written in Param2.

SI_USE_MASTER_AUTHENTICATION (0x0000)

Setting Master Authentication Function

Some administrative actions require master authentication for security, such as adding or deleting users. Users can choose to implement master authentication for other commands using SI_USE_MASTER_AUTHENTICATION (sub-command of CMD_SET_SYSTEM_INFO). Generally, system defaults do not require master authentication except where system security is an issue. The example below uses the master authentication function, with CMD_SET_SYSTEM_INFO (0x20) in the command field, and SI_USE_MASTER_AUTHENTICATION (0x00) in the Param1 field, and a value of "1" in the Param2 field.

Command Packet

0x00	0x20	**0x0000**	**0x0001**	0x0000	0x0000	0x00	Chksum

ACK Packet

0x00	0x20	0x0000	0x0001	0x0000	0x0000	0x00	Chksum

Caution: When you set SI_USE_MASTER_AUTHENTICATION, the SDA04 should not have any fingerprint registered, otherwise the first fingerprint registered in the SDA04 should have Master Privileges (Root Master). If you lose the Root Master Fingerprint, the SDA04 cannot be managed properly.

SI_VERIFY_SECU_LEVEL (0x0002)

Setting the Security Level

The security level of a system depends on a site's requirements and can be set using the SI_VERIFY_SECU_LEVEL command. The default SDA04 value is set at SLEVEL_NORMAL (0x05) – the "normal" level offers users the best compromise between convenience and security.

Note: Read *SDA04 Security Level Basics* in section 3.3.4 before establishing your security policy .

The example below sets the security level one level higher than the normal level, inputting CMD_SET_SYSTEM_INFO (0x20) in the command field, SI_VERIFY_SECU_ LEVEL (0x02) in Param1 field, and SLEVEL_ABOVE_NORMAL (0x06) into Param2.

Command Packet

0x00	0x20	**0x0002**	**0x0006**	0x0000	0x0000	0x00	Chksum

ACK Packet

0x00	0x20	0x0002	0x0006	0x0000	0x0000	0x00	Chksum

SI_USE_RELAY (0x0003)

Relay Used/Not used

Use SDA04 relay by including SI_USE_RELAY in Param1 field. This command causes the SDA04 to either bypass or use a relay terminal using CMD_RELAY_ONOFF command. *To use Wiegand protocol, this value must be "Off."*

To use Auto Identify, this value must be set to "On", Wiegand to "Off", and the time setting for Auto Identify must be non-zero. The function Auto Identify automatically sets the device in "Identify" mode and sends a relay signal when it meets a signal from the external switch number 1. The output signal from the relay is not a Wiegand signal, but it sets the relay "On" for a specified duration, after which it is reset to "Off". The duration for which the relay stays on is set by SI_RELAY_TIME. The Auto Identify relay time is used in the following cases:

- Environments where a controller is used to only enroll and delete users.
- Hybrid environments where 1-to-few (about 100~200 records in database) identification is used.

In the following example, relay is made available.

Command Packet

0x00	**0x20**	**0x0003**	**0x0001**	0x0000	0x0000	0x00	Chksum

ACK Packet

0x00	0x20	0x0003	0x0001	0x0000	0x0000	0x00	Chksum

SI_COM_SPEED (0x0004)

Changing the Communication Speed

Including the SI_COM_SPEED in the Param1 field of CMD_SET_SYSTEM_INFO command will change the communication speed of the SDA04. Currently the communication speed supports ranges from 1200 bps to 921600 bps (460800 bps in the case of SDA03M). Upon receiving this command, the SDA module sends an ACK packet and then changes the communication speed. If the error code in the ACK packet is M2ERROR_NONE, the main controller initializes its port at the new speed and starts communicating with the SDA04. This value is not saved in flash memory.

The example below changes the communication speed to 57600 bps (0x0003).

Command Packet

0x00	0x20	0x0004	0x0003	0x0000	0x0000	0x00	Chksum

ACK Packet

0x00	0x20	0x0004	0x0003	0x0000	0x0000	0x00	Chksum

SI_IDENTIFY_SECU_LEVEL (0x0008)

Control Security Level for Identify

There are two methods of authentication that can be set, identification ("Identify") and verification ("Verify"). ***Identification*** requires only an input fingerprint, which is compared to a database of stored fingerprint templates for a possible match. ***Verification*** uses a fingerprint and another form of identification such as a user ID or smart card to verify a user's stated identity. The security level for the "Identify" method (also known as 1:N matching) can be set using SI_IDENTIFY_SECU_LEVEL. The default is SLEVEL_ABOVE_NORMAL (0x06). The "Identify" method requires a higher security level than the "Verify" method (also known as 1:1 matching), therefore the value may not be set to less than 0x04.

Note: Read SDA04 Security Level Basics in section 3.3.4 before establishing your security policy.

The example below sets the highest security level and inputs:

- CMD_SET_SYSTEM_INFO (0x20) into the command field
- SI_IDENTIFY_SECU_LEVEL (0x08) into the Param1 field
- SLEVEL_HIGHEST (0x09) into the Param2 field

Command Packet

0x00	0x20	0x0008	0x0009	0x0000	0x0000	0x00	Chksum

ACK Packet

0x00	0x20	0x0008	0x0009	0x0000	0x0000	0x00	Chksum

SI_WIEGAND_FORMAT (0x0009)

Changing the Wiegand Protocol

The SDA04 can be set to use the Wiegand protocol by using SI_WIEGAND_FORMAT in the Param1 field of CMD_SET_SYSTEM_INFO. Currently the SDA04 supports 26-bit Wiegand format and 34-bit Wiegand format. *For more details, refer to Appendix D.*

To use the Wiegand protocol, the relay must be turned off and a Wiegand sitecode must be assigned. First status is Relay = Off, Wiegand Format = 26bit, Wiegand Sitecode = 0x0000, and Wiegand is available. When the status is available, the SDA04 is in "Identify" mode and the automatic external switch is set. If this is successful, the Wiegand signal will use relay for output. Execute "Verify" or "Identify" through the command output and the Wiegand signal through the relay.

The following example changes to 26-bit Wiegand format (0x12c) from Wiegand protocol.

Command Packet

0x00	0x20	0x0009	0x012c	0x0000	0x0000	0x00	Chksum

ACK Packet

0x00	0x20	0x0009	0x012c	0x0000	0x0000	0x00	Chksum

SI_WIEGAND_SITECODE (0x000a)

Changing Wiegand Protocol Sitecode

Using SI_WIEGAND_SITECODE in the Param1 field of the CMD_SET_SYSTEM_INFO command determines whether the 26-bit or 34-bit Wiegand protocol is used. The difference between the two formats is that the 26-bit Wiegand format has a sitecode of 8 bits, whereas 34-bit Wiegand format uses 16 bits for the sitecode.

The following example changes sitecode 0x0001.

Command Packet

0x00	0x20	0x000a	0x0001	0x0000	0x0000	0x00	Chksum

ACK Packet

0x00	0x20	0x000a	0x0000	0x0000	0x0000	0x00	Chksum

7.4. Saving New System Settings

The SDA04 saves the system settings whenever a change is made. The only exception is SI_COM_SPEED. This is reset to the default of 115200BPS when the SDA04 is powered on.

Note: The SDA04 automatically saves all system setting changes to flash memory.

7.5. Checking System Settings

The command CMD_GET_SYSTEM_INFO is used to check system setting values and inputs item(s) to be checked in the Param1 field. After receiving this command, the SDA04 sends out the current system values in the Param2 field.

The following example checks the current system security level. It shows that the current security level is set at SLEVEL_ABOVE_NORMAL.

Command Packet

0x00	**0x30**	**0x0002**	0x0000	0x0000	0x0000	0x00	Chksum

ACK Packet

0x00	**0x30**	**0x0002**	**0x0006**	0x0000	0x0000	0x00	Chksum

7.6. Configuring Image Brightness

Exposure values for image brightness can be adjusted, or "tuned", automatically or manually.

CMD_EXP_AUTOTUNING

Automatic Configuration

Use CMD_EXP_AUTOTUNING for automatic configuration. This requires a fingerprint image capture from a user. Place finger on the fingerprint sensor before the main controller sends this command to the SDA04 and leave it in place until the operation is complete. Once the automatic configuration is done, the SDA04 turns off LED and returns the exposure value in the Param1 field.

The example below configures image brightness by using CMD_EXP_AUTOTUNING (0x16). It shows the optimal brightness value of 312 (0x138).

Command Packet

0x00	**0x16**	0x0000	0x0000	0x0000	0x0000	0x00	Chksum

ACK Packet

0x00	**0x16**	**0x0138**	0x0000	0x0000	0x0000	0x00	Chksum

CMD_SET_SYSTEM_INFO

Manual Configuration

Use CMD_SET_SYSTEM_INFO to manually configure the image brightness by using SI_BRIGHTNESS. Manual configuration is faster than automatic configuration because it does not use CMD_GET_IMAGE to adjust the current image brightness before configuration.

7.7. Acquiring Fingerprint Image

Use CMD_GET_IMAGE to get the fingerprint image from the SDA04 fingerprint input window. When using this command, specify the image size in the Param1 field. Using a smaller image size result in lower image quality, but this will help increase the speed at which images are obtained. The values that can be set are VIEW_NORMAL (0x01, original image size), VIEW_HALF (0x02, 1/2 of original size), and VIEW_QUARTER (0x04, 1/4 of original size). The data size is determined from the image size. It reads the image size in the **lwExtraData** and **hwExtraData** fields from the SDA04 and takes the image specified in bytes in these fields.

The example below gets the original image size. The SDA04 sends image data in bytes of the **ExtraData** field to the main controller right after sending the ACK packet.

Command Packet

0x00	**0x43**	**0x0001**	0x0000	0x0000	0x0000	0x00	Chksum

ACK Packet

0x00	**0x43**	0x0000	0x0000	0x52a0	0x0001	0x00	Chksum

Chapter 8. User Management

A "master" is a registered SDA04 user with the authority to register or delete other users and has complete access to all system information and security settings. The first fingerprint registered in the SDA04 must always be a master fingerprint (also known as the "root master"). This information is then stored in flash memory, allowing additional users to be registered later. If a master fingerprint is not registered, an error code will be returned in M2ERROR_MASTER_NOT_FOUND whenever commands requiring master authentication are issued.

Note: The SDA04 will accept up to five masters per system. If this limit is exceeded, an error message will appear.

- **Commands that require master authentication**

  CMD_SET_SYSTEM_INFO
  CMD_FP_REGISTER_START
  CMD_FP_DELETE
  CMD_DB_GET_RECCOUNT
  CMD_DB_ADD_REC
  CMD_DB_DELETE_REC
  CMD_DB_GET_REC
  CMD_DB_GET_FIRSTREC
  CMD_DB_GET_NEXTREC
  CMD_DB_GET_CURRENTREC
  CMD_DB_DELETE_ALL

- **Root Master**

  When initializing the system for the first time, registration of a master fingerprint may be required. The first fingerprint registered in the system is known as a "root master". When registering a root master, master authentication is not required.

- **Checking for a Root Master**

  To determine whether a root master has been registered, use the command CMD_GET_MASTER_COUNT. After receiving this command, the SDA04 returns the number of masters registered in the system in Param1 of the ACK packet. If it returns '0', a root master needs to be registered.

8.1. User Registration

Users can be divided into two categories, a common user and a master. Master refers to a person who has the authority to add or delete other users. (The root master is also considered a master; the only difference is that root masters do not require master authentication for enrollment in the system because theirs is the first fingerprint registered in the SDA04.)

The commands used for registering the master and users are the same, but the parameter values passed are different.

Master authentication should be done before registering a user. New users should input their fingerprints two times. Below is an abstract of the user registration process.

Note: Registration requires two slightly different samples of minutiae data from the same fingerprint. This requires users to lift the finger off the sensor briefly between fingerprint captures. Unlike the previous the FDA01 model, the SDA04 will accept two fingerprints with identical minutiae. By using SI_USING_DIFF_FP_REGISTER, different fingerprint pairs (e.g. Thumb and Index) can also be registered.

- **Abstract of user registration**

  CMD_FP_VERIFY_MASTER (when MASTER AUTHENTICATION is ON)
  CMD_FP_REGISTER_START
  CMD_FP_REGISTER_END

8.1.1. Master Registration Process

1. CMD_FP_VERIFY_MASTER

This process is needed only when SI_USE_MASTER_AUTHENTICATION option is ON.

This step verifies the master's fingerprint before registering a user. The main controller asks to place the master's finger on the fingerprint sensor first before issuing this command to the SDA04.

- The host displays a message requesting that a master finger be placed on the fingerprint sensor for verification.
- The master places finger on fingerprint sensor, and the main controller sends the CMD_FP_VERIFY_MASTER command to the SDA04.
- When the master's fingerprint has been successfully verified, the SDA04 returns M2ERROR_NONE in the ErrorCode with the master's User ID in Param1 of the ACK packet; if the master's fingerprint verification fails, an error code is returned.
- Proceeds to next step if ACK is M2ERROR_NONE.

2. CMD_FP_REGISTER_START

This step registers a User ID and a user's fingerprint.

- After the user's fingerprint is input, the main controller then sends CMD_FP_REGISTER_START to the SDA04 with User ID in the Param1 field and a value of '1' in the Param2 field of the packet.
- The SDA04 returns M2ERROR_NONE code if the User ID is unique (not one that is already registered) and fingerprint has been successfully capture. If not, it returns an error code.
- Goes to next step if ACK is M2ERROR_NONE.

3. CMD_FP_REGISTER_END

This step recaptures the user's fingerprint image to complete the registration process.

- After the user places a finger on the fingerprint sensor, the main controller sends the

CMD_FP_REGISTER_END command with User ID in Param1 and a value of '1' in Param2 to the SDA04.

- The SDA04 returns M2ERROR_NONE in the ACK packet if the user fingerprint registration is successful; otherwise, an error code is returned.

Master Registration

The example below shows the packets used when registering a master whose ID is '1234' and verifying a master whose ID is '0001'

1. CMD_FP_VERIFY_MASTER Command Packet

0x00	0x57	0x0000	0x0000	0x0000	0x0000	0x00	Chksum

2. CMD_FP_VERIFY_MASTER ACK Packet

0x00	0x57	0x0001	0x0000	0x0000	0x0000	0x00	Chksum

3. CMD_FP_REGISTER_START Command Packet

0x00	0x50	0x1234	1	0x0000	0x0000	0x00	Chksum

4. CMD_FP_REGISTER_START ACK Packet

0x00	0x50	0x0000	0x0000	0x0000	0x0000	0x00	Chksum

5. CMD_FP_REGISTER_END Command Packet

0x00	0x51	0x1234	1	0x0000	0x0000	0x00	Chksum

6. CMD_FP_REGISTER_END ACK Packet

0x00	0x51	0x0000	0x0000	0x0000	0x0000	0x00	Chksum

Note: The values in the Param1 and Param2 fields should be identical in both the CMD_FP_REGISTER_START and CMD_FP_REGISTER_END commands.

- **Root master registration process**

  The registration process of the root master is the same as the master registration process, with the exception that it does not require the master fingerprint verification process. Since the root master is always the first user fingerprint to be enrolled in the system, there is no previously stored master fingerprint to refer to. The root master registration process does not require any previous master verification, and so the command CMD_FP_VERIFY_MASTER is skipped (this was the first step in the master registration process). In this case, the main controller uses the following command sequence when transmitting to the SDA04 for root master registration.

- **Abstract of master registration process**

```
CMD_FP_REGISTER_ST
ART                     // capture fingerprint for registration
CMD_FP_REGISTER_E       // recapture fingerprint for registration
ND
```

8.1.2. User Registration Process

This process registers new users in the system database. Master verification is required to register new users. (The process is the same as for the master except for a value of '0' in the Param2 field in the fingerprint recapture command.)

1. CMD_FP_VERIFY_MASTER

This process is needed only when the SI_USE_MASTER_AUTHENTICATION option is ON.
This step verifies the master's fingerprint before a new user is registered. The main controller requires that a master place a finger on the fingerprint sensor before sending this message to the SDA04.

- After the master places a finger on the fingerprint sensor, the main controller sends the CMD_FP_VERIFY_MASTER command to the SDA04.
- If the master fingerprint successfully matches, the SDA04 returns M2ERROR_NONE in ErrorCode with the master's User ID in Param1 of the ACK packet. Otherwise, an error code is returned.
- Proceeds to next step if the error code in the ACK packet is M2ERROR_NONE.

2. CMD_FP_REGISTER_START

This step is used to register the User ID and the user's fingerprint image.

After the user places a finger on the fingerprint sensor, the main controller sends the CMD_FP_REGISTER_START command to the SDA04 with User ID in Param1 and a value of '1' in Param2.

Input a 4-digit User ID for use by the root master.

- After the user places a finger on the fingerprint sensor, the main controller sends the CMD_FP_REGISTER_START command to the SDA04 with User ID in Param1 and a value of '1' in Param2.
- The SDA04 returns M2ERROR_NONE in the ACK packet if the User ID is unique (not duplicated) and the fingerprint is successfully matched. If the process fails, an error code is returned.

Proceeds to the next step if ACK is M2ERROR_NONE.

3. CMD_FP_REGISTER_END

The registration process requires two fingerprint images to be captured from the same finger. This step re-captures the user's fingerprint image to complete registration.

- After the user places a finger on the fingerprint sensor, the main controller inputs the User ID in Param1 and a value of '0' in Param2, and then sends the CMD_FP_REGISTER_END command to the SDA04.
- If the User ID is unique (not duplicate) and the fingerprint is successfully matched, the User ID and fingerprint information are stored in the SDA04 database, and the main controller returns M2ERROR_NONE. Otherwise, an error code is returned.

User Registration
The example below registers a user with a User ID of '1998.' The master being verified in the process has an ID of '0001.'

1. CMD_FP_VERIFY_MASTER Command Packet

0x00	0x57	0x0000	0x0000	0x0000	0x0000	0x00	Chksum

2. CMD_FP_VERIFY_MASTER ACK Packet

0x00	0x57	**0x0001**	0x0000	0x0000	0x0000	0x00	Chksum

3. CMD_FP_REGISTER_START Command Packet

0x00	0x50	0x1998	0x0000	0x0000	0x0000	0x00	Chksum

4. CMD_FP_REGISTER_START ACK Packet

0x00	0x50	0x0000	0x0000	0x0000	0x0000	0x00	Chksum

5. CMD_FP_REGISTER_END Command Packet

0x00	0x51	**0x1998**	**0x0000**	0x0000	0x0000	0x00	Chksum

6. CMD_FP_REGISTER_END ACK Packet

0x00	0x51	0x0000	0x0000	0x0000	0x0000	0x00	Chksum

8.2. User Verification

This step verifies users already registered in the system database. The command CMD_FP_VERIFY is used whenever verification is required.

The user verification process is outlined below.

- After inputting the User ID, the main controller displays instructions for placing a finger on the fingerprint sensor.
- The main controller sends the User ID and the CMD_FP_VERIFY command to the SDA04.
- The main controller waits for the ACK packet from the SDA04.
- The SDA04 returns M2ERROR_NONE in ErrorCode with User ID in Param1 and master flag in Param2 of the ACK packet if verification is successful. Otherwise, it sends an error code.

Verifying a user with a User ID of '1234'

CMD_FP_VERIFY Command Packet

0x00	**0x55**	**0x1234**	0x0000	0x0000	0x0000	0x00	Chksum

CMD_FP_VERIFY ACK Packet

0x00	**0x55**	**0x1234**	**0x0000**	0x0000	0x0000	0x00	Chksum

8.3. User Deletion

This is the command procedure for removing users from the system database. All user deletions require master verification.

- **Abstract for deleting users**

  CMD_FP_VERIFY_MASTER (When MASTER AUTHENTICATION is ON)
  CMD_FP_DELETE

  1. CMD_FP_VERIFY_MASTER
  This process is needed only when SI_USE_MASTER_AUTHENTICATION option is ON.
  The first step requires master fingerprint verification before deleting a user. The main controller sends a message to the SDA04 asking a registered master to place a finger on the fingerprint sensor.

  - The main controller displays a message requesting a master fingerprint.
  - After the master places a finger on the fingerprint sensor, the main controller sends the CMD_FP_VERIFY_MASTER command to the SDA04.
  - The SDA04 returns M2ERROR_NONE in ErrorCode with the master's User ID in Param1 of the ACK packet if master fingerprint verification is successful. Otherwise, it returns an error code.
  - Proceeds to next step if ACK ErrorCode is M2ERROR_NONE.

  2. CMD_FP_DELETE
  If master verification is successful, the main controller issues the command to delete the specified User ID from the SDA04 database.

  - The main controller receives the User ID to be deleted (4 bytes).
  - The main controller sends the User ID and CMD_FP_DELETE command to the SDA04.
  - The SDA04 deletes the User ID.
  - The SDA04 returns M2ERROR_NONE in ErrorCode with User ID in Param1 and master flag in Param2 of the ACK packet if user deletion is successful. Otherwise, an error code is returned.

Deleting a user with a User ID of '1234', verified by master whose ID is '0001'

1. CMD_FP_VERIFY_MASTER Command Packet

0x00	0x57	0x0000	0x0000	0x0000	0x0000	0x00	Chksum

2. CMD_FP_VERIFY_MASTER ACK Packet

0x00	0x57	**0x0001**	0x0000	0x0000	0x0000	0x00	Chksum

3. CMD_FP_DELETE Command Packet

0x00	0x54	**0x1234**	0x0000	0x0000	0x0000	0x00	Chksum

4. CMD_FP_DELETE ACK Packet

0x00	0x54	**0x1234**	**0x0000**	0x0000	0x0000	0x00	Chksum

8.4. Changing Fingerprint Data

When changing fingerprint data, the previously registered fingerprint data will be replaced. If the master fingerprint authorization feature is turned on, a master fingerprint verification will be required before the update is allowed. If master authentication is turned off, only the user's fingerprint will be required to make the change.

- **Abstract of fingerprint data change**

  CMD_FP_VERIFY
  CMD_FP_CHANGE_START
  CMD_FP_CHANGE_END

  1. CMD_FP_VERIFY
  The first step in the process is getting authorization to change (replace) a user's registered fingerprint in the SDA04 database. The user's identity must be verified before the change is made, which requires the user's live fingerprint image capture to compare with the user's registered fingerprint data.

  - The main controller displays a message requesting a fingerprint after inputting the User ID.
  - The main controller sends the CMD_FP_VERIFY command and User ID to the SDA04.
  - The SDA04 returns M2ERROR_NONE in ErrorCode with the User ID in Param1 and master flag in Param2 of the ACK packet if the user's fingerprint successfully verified. Otherwise, it returns an error code.
  - Proceeds to next step if ErrorCode of ACK is M2ERROR_NONE.

  2. CMD_FP_CHANGE_START
  This command captures a new fingerprint image, which is the first of two fingerprints needed for the updated registration. The main controller sends the CMD_FP_ CHANGE_START command to the SDA04 after the user has been verified.

  - The main controller displays a message requesting a fingerprint for new registration.
  - After the user places a finger on the fingerprint sensor, the main controller sends the CMD_FP_CHANGE_START command to the SDA04 with User ID in Param1 of the command packet.
  - The SDA04 returns M2ERROR_NONE if the fingerprint image is successfully captured. Otherwise, an error code is returned.
  - Proceeds to next step if ACK ErrorCode is M2ERROR_NONE.

  3. CMD_FP_CHANGE_END
  This command completes the fingerprint data change by capture a fingerprint image again, which is the second of two fingerprints needed for the updated registration. The main controller issues the CMD_FP_CHANGE_END command to the SDA04 after the user places a finger on the fingerprint sensor a second time.

  - The main controller and displays a message requesting a fingerprint a second time to complete the registration.
  - After the user places a finger on the fingerprint sensor, the main controller sends the CMD_FP_CHANGE_END command to the SDA04 with User ID in Param1 of the command packet.
  - The SDA04 returns M2ERROR_NONE if fingerprint is successfully changed. Otherwise, an error is

returned.

Changing User's Fingerprint Data (User ID is '1234')

1. CMD_FP_VERIFY Command Packet

0x00	**0x55**	**0x1234**	0x0000	0x0000	0x0000	0x00	Chksum

2. CMD_FP_VERIFY Ack Packet

0x00	0x54	**0x1234**	**0x0000**	0x0000	0x0000	0x00	Chksum

3. CMD_FP_CHANGE_START Command Packet

0x00	0x52	**0x1234**	0x0000	0x0000	0x0000	0x00	Chksum

4. CMD_FP_CHANGE_START Ack Packet

0x00	0x52	0x0000	0x0000	0x0000	0x0000	0x00	Chksum

5. CMD_FP_CHANGE_END Command Packet

0x00	0x53	**0x1234**	0x0000	0x0000	0x0000	0x00	Chksum

6. CMD_FP_CHANGE_END Ack Packet

0x00	0x53	**0x1234**	0x0000	0x0000	0x0000	0x00	Chksum

8.5. User Identification

Users can be identified by comparing their fingerprints with all other fingerprints stored in the SDA04 database. The command CMD_FP_IDENTIFY is used for this purpose.

- The host displays a message asking the user to place a finger on the fingerprint sensor.
- The host sends the CMD_FP_IDENTIFY command to the SDA04.
- The SDA04 returns M2ERROR_NONE (0x00) in ErrorCode with User ID in Param1 and master flag in Param2 of the ACK packet if the newly captured fingerprint matches the image already registered in the database. Otherwise, an error code is returned.
- When the identification process is called, it returns an acknowledgement after every 100 false matches with the above error code. This means that the identification process is still running and requires the controller program to continue reading until any error code other than M2ERROR_SEARCHING_FOR_IDENTIFY (0x1f) is returned.

> **Note**: During long processing times for commands such as Identification and Capture, the host controller can terminate the process by sending a 4-byte break command sequence.
> Command Sequence: 0xA8, 0xB8, 0xC8, 0xD8
> Acknowledge: Normal acknowledge format of each command with break error code of 0xFE

- **Auto Identification function**

This function is used to identify automatically using SW0 input. Refer to the option SI_USE_RELAY (0x03) option in the CMD_SET_SYSTEM_INFO (0x20) section.

Commenté [d9]: Confirm

User Identification (User ID is '1234')

1. CMD_FP_IDENTIFY Command Packet

0x00	**0x56**	0x0000	0x0000	0x0000	0x0000	0x00	Chksum

2. CMD_FP_IDENTIFY ACK Packet (Indication ACK that continuously searching)

0x00	**0x56**	**0xffff**	0x0000	0x0000	0x0000	**0x1f**	Chksum

3. CMD_FP_IDENTIFY ACK Packet

0x00	**0x56**	**0x1234**	0x0000	0x0000	0x0000	0x00	Chksum

Appendix A. Protocol Reference

A.1. Features

- *SecuSearch Identification* algorithm: provides faster 1:N identification
- *Smart Capture* function: allows the SDA04 to recognize fingerprints under bright ambient light conditions, such as outdoors under direct sunlight (using the SDOPP04 optic module)

Commenté [d10]: Is smart capture implemented?

A.2. SG400 and ANSI378 modes

The SDA04 supports both the SecuGen proprietary 400 byte template format and ANSI378 template format. Protocol commands that support SecuGen proprietary format have (SG400) appended in the protocol reference. Commands that support ANSI 378 have (ANSI378) appended in the protocol reference. Additionally, the following quick reference images indicate commands that support SG400 and/or ANSI378.

A.2. Available Commands and Descriptions

The main controller uses a 12-byte data packet to communicate with the SDA04 Processing Unit. After receiving and processing commands sent from the main controller, the SDA04 returns results in packet format. The structure of these packets is described below.

Channel	Command	Param1	Param2	lwExtraData	hwExtraData	ErrorCode	Checksum
1 Byte	1 Byte	2 Bytes	2 Bytes	2 Bytes	2 Bytes	1 Byte	1 Byte

Channel	Not used at this time
Command	Specifies the command that SDA04 executes
Param1	Used for data transmissions
Param2	Used if there is data to be transmitted in addition to Param1
lwExtraData. HwExtraData	Used when more than 4 bytes of data (Param1 and Param2) are transmitted. Data length is specified in lwExtraData (the low-word value) and hwExtraData (the high-word value).
ErrorCode	Specifies execution results of the SDA04 for the main controller's commands. If command is successful, its value is always M2ERROR_NONE(0x00). ErrorCode is used only for the ACK packet.
Checksum	The values of the first 11 bytes (excluding the checksum byte) are summed and then divided by 0x0100 (256). This will create a 1-byte shift. The remaining value from this shift is the checksum byte.

CMD_GET_VERSION (0x05)

Gets the current protocol version. The SDA04 sets the major version in Param1 and minor version in Param2 of the ACK packet. ErrorCode is always M2ERROR_NONE.

Command Packet:

Channel	0x00
Command	0x05
Param1	
Param2	
ErrorCode	
LwExtraData	0x0000
HwExtraData	0x0000
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x05
Param1	Version Major Number
Param2	Version Minor Number
ErrorCode	M2ERROR_NONE
LwExtraData	0x0000
HwExtraData	0x0000
CheckSum	

CMD_DEVICE_TEST (0x10)

Tests the hardware operations. The device name to be tested is specified in the Param1 field. The names of each device are listed below.

Param1 Device Name		Description
0x0000	DEVICE_ALL	Test all devices
0x0001	DEVICE_SENSOR	Test image sensor
0x0002	DEVICE_FLASHMEM	Test flash memory
0x0003	DEVICE_CODE_CHECKSUM	Check program code checksum in flash memory

If ErrorCode for DEVICE_ALL is not M2ERROR_NONE, test DEVICE_SENSOR or DEVICE_FLASHMEM to check the current device status.

Command Packet:

Channel	0x00
Command	0x10
Param1	Device name
Param2	
lwExtraData	0x0000
hwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x10
Param1	
Param2	
lwExtraData	0x0000
hwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error M2ERROR_FLASH_OPEN – Flash memory error M2ERROR_SENSOR_OPEN– Image sensor error M2ERROR_CODE_CHECKSUM_ERROR – Checksum error
CheckSum	

CMD_RELAY_ONOFF (0x12)

Turns the SDA04 relay on or off. There are 2 relays. Relay 0 (EX_RELAY0) is PIN4 on J5. Relay 1 (EX_RELAY1) is PIN5 on J5.

Command Packet:

Channel	0x00
Command	0x12
Param1	Relay Number (0 or 1)
Param2	On/Off (On → 1, Off → 0)
lwExtraData	0x0000
hwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x12
Param1	
Param2	
lwExtraData	0x0000
hwExtraData	0x0000
ErrorCode	M2ERROR_NONE
CheckSum	

CMD_OPTICLED_ONOFF (0x14)

Commenté [d11]: SDOPP04 LEDS did not respond

Turns the LED on/off in the optic module.

Command Packet:

Channel	0x00
Command	0x14
Param1	On/Off (On → 1, Off → 0)
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x14
Param1	
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error
CheckSum	

CMD_EXP_AUTOTUNING (0x16)

Commenté [d12]: SDA04 returns M2ERROR_NONE, but did not process.

Automatically sets the exposure level of the optic module. The user must place a finger on the fingerprint sensor for scanning before tuning is complete and the ACK packet is returned.

Command Packet:

Channel	0x00
Command	0x16
Param1	
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x16
Param1	Coarse value of exposure
Param2	Fine value of exposure
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error M2ERROR_SENSOR_OPEN – Image sensor error
CheckSum	

CMD_FP_DIFF_REGISTER (0x19) (SG400)

Commenté [d13]: Is this supported?

When a user registers two different fingerprints to one User ID, this command should be sent to the SDA04 first. This command applies only one time to the next registration. Unlike normal registration, the SDA04 does not compare the two registered fingerprints. If a user wishes to permanently use this option, use SI_USING_DIFF_FP_REGISTER in the CMD_SET_SYSTEM_INFO (0x20) command.

Command Packet:

Channel	0x00
Command	0x19
Param1	
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x19
Param1	
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error
CheckSum	

CMD_SET_SYSTEM_INFO (0x20)

Changes the SDA04 system settings. The specified item ("selector") to be changed is stored in Param1, and its value in Param2. Master authentication is needed to perform this command, so the CMD_FP_VERIFY_MASTER command should be issued before CMD_SET_SYSTEM_INFO.

Note: SDA04 automatically saves all system setting changes to flash memory. The only exception is SI_COMM_SPEED. This parameter is reset to the default of 115200bps when the SDA04 is powered on.

Command Packet:

Channel	0x00
Command	0x20
Param1	Items to be changed (refer to Parameter Summary Table below)
Param2	Value to change, different according to items
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x20
Param1	
Param2	
lwExtraData	0x0000
hwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error
CheckSum	

Parameter Summary Table

	Command Code: SET_SYSTEM_INFO (0x20)		Default
	Parameter1	**Parameter2**	**Value**
0x00	SI_USE_MASTER_AUTHENTICATION	0 or 1	0
0x02	SI_VERIFY_SECU_LEVEL	0 ~ 9	5
0x03	SI_USE_RELAY	0 or 1	0
0x04	SI_COM_SPEED	SDA04/SDA04M 1200~921600 SDA03M 1200~460800	01(115200bps)
0x05	SI_BRIGHTNESS	0~1000	50
0x08	SI_IDENTIFY_SECU_LEVEL	0 ~ 9	6
0x09	SI_WIEGAND_FORMAT	**0***,0x12C,0x12E	0x12C
0x0A	SI_WIEGAND_SITE_CODE	Numeric site code	0
0x0B	SI_REGISTER_QUALITY	30 ~ 100	40
0x0C	SI_VERIFY_QUALITY	10 ~ 100	30
0x0E	SI_RELAY_TIME	**0*** or 1 ~10000	0
0x10	SI_USE_DATA_CHECKSUM	0 or 1	1
0x11	SI_USE_AUTO_ON	0 or 1	1
0x12	SI_ADAPTIVE_CAPTURE	0 or 1	1
0x14	SI_QUALITY_FEEDBACK	0 or 1	1
0x16	SI_ID_DIGIT	RFU	RFU
0x18	SI_TEMPLATETYPE	0x0100, 0x0200	0x0200 (SG400)

* '**0**' in Parameter2 means the function is disabled.

SI_USE_MASTER_AUTHENTICATION (0x00)

Determines if master authentication for commands is being used. This ensures a high level of security when registering or deleting users. To use master authentication, specify '1' in Param2; to disable it, specify '0'. Default is '0'.

SI_VERIFY_SECU_LEVEL (0x02)

Sets SDA04 security levels. Read *SDA04 Security Level Basics* in section 3.3.4 before establishing your security policy.

SDA04 Security Levels		
0*	0x00	Minimum Security
1	0x01	SLEVEL_LOWEST
2	0x02	SLEVEL_LOWER
3	0x03	SLEVEL_LOW
4	0x04	SLEVEL_BELOW_NORMAL
5	0x05	SLEVEL_NORMAL
6	0x06	SLEVEL_ABOVE_NORMAL
7	0x07	SLEVEL_HIGH
8	0x08	SLEVEL_HIGHER
9	0x09	SLEVEL_HIGHEST

* Not Recommended

Authentication of users can be implemented with either of two methods, identification (Identify) or verification (Verify). Only authorized SDA04 users should set security levels for identification using SI_IDENTIFY_SECU_LEVEL.

SI_USE_RELAY (0x03)

Determines whether or not SDA04 relay is used. If value is '1' then relay is set for use; if value is '0,' relay cannot be used. Default is '0'.

Under the following conditions, when a signal from external switch number 1 is received, "identify" automatically returns the result to the relay. The time during which the relay stays on is set by SI_RELAY_TIME. The Auto-Identify relay time is used in the following cases:

1. Environments where a controller is used only to enroll and delete users.
2. Hybrid environments where 1-to-few identification is used.

- SI_USE_RELAY = 1
- SI_WIEGAND_FORMAT = 0
- SI_RELAY_TIME > 0 (non-zero)

SI_COM_SPEED (0x04)

Changes communication speed of current channel. Communication speed can be changed after receiving the ACK packet from the SDA04. The default value is BAUD_115200. Other speeds supported are listed below. This value is not saved in flash memory.

Command	BAUD Rate
0x00F3	BAUD_921600 (SDA04,SDA04M)
0x00F2	BAUD_460800

0x00F1	BAUD_230400
0x0001	BAUD_115200
0x0003	BAUD_57600
0x0005	BAUD_38400
0x000B	BAUD_19200
0x0017	BAUD_9600
0x005F	BAUD_2400
0x00BF	BAUD_1200

SI_BRIGHTNESS (0x05)

Adjusts image brightness by setting the coarse value of the optical unit. Values range from '0' to '1000'.

Note: When 'SI_ADAPTIVE_CAPTURE' is enabled, this value does not affect the fingerprint capture.

SI_EXP_GAIN (0x07)

Adjusts image brightness by setting the gain value of the optical unit. Values range from '0' to '63'. The default value is '10'.

Note: When 'SI_ADAPTIVE_CAPTURE' is enabled, this value does not affect the fingerprint capture.

SI_IDENTIFY_SECU_LEVEL (0x08)

Sets SDA04 security levels. Read *SDA04 Security Level Basics* in section 3.3.4 before establishing your security policy. This is used to set the security level for 1:N identification. The Identify mode should be managed using a higher security level than the Verify mode. The default level for Identify is 8 (SLEVEL_HIGHER).

SDA04 Security Levels		
1	0x01	SLEVEL_LOWEST
2	0x02	SLEVEL_LOWER
3	0x03	SLEVEL_LOW
4	0x04	SLEVEL_BELOW_NORMAL
5	0x05	SLEVEL_NORMAL
6	0x06	SLEVEL_ABOVE_NORMAL
7	0x07	SLEVEL_HIGH
8*	0x08	SLEVEL_HIGHER
9	0x09	SLEVEL_HIGHEST

**Level 8 is the SDA04 default security level for identification (1:n matching) because the identification method uses only biometric data from fingerprints for authentication without other user ID information that iss used by the verification method (1:1 matching) whose default level is 5. Security level is an important consideration when using the identification (fingerprint only) method. Refer to SDA04 Security Level Basics in section 3.3.4 for more information.*

SI_WIEGAND_FORMAT (0x09)

Enables Wiegand protocol or sets the format of Wiegand protocol (26-bit or 34-bit formats are supported). The default is WIEGAND_FMT_26BIT. The 34-bit Wiegand format increases the sitecode field from 8-bits

to 16-bits for larger installations. To use Wiegand, the relay status must be OFF.

1. 0x0000 WIEGAND_NOT_USED
2. 0x012c WIEGAND_FORMAT_26BIT
3. 0x012e WIEGAND_FORMAT_34BIT

Sending the value '0' in Param2, of the following commands temporarily prohibits the transmission of Wiegand output.

- CMD_FP_VERIFY
- CMD_FP_IDENTIFY
- CMD_FP_IDENTIFY_EX
- CMD_DB_VERIFY
- CMD_DB_IDENTIFY
- CMD_DB_IDENTIFY_EX

Commenté [d14]: Per the command references, sending "1" disables the Wiegand output.

Under the following conditions, when a signal from external switch number 1 is received, "identify" automatically returns the results to the relay. The length of time that the relay stays on is set by SI_RELAY_TIME. The Auto-Identify relay time is used in the following cases:

1. Environments where a controller is used only to enroll and delete users.
2. Hybrid environments where 1:N identification is used.

- SI_USE_RELAY = 1
- SI_WIEGAND_FORMAT = 0
- SI_RELAY_TIME > 0 (non-zero)

SI_REGISTER_QUALITY (0x0b)

Configures the value for fingerprint registration image quality. This parameter is only used for fingerprint registration, and determines the quality of the fingerprint images that should be captured. Higher settings will register higher quality fingerprints, but may also cause users with poor quality fingerprints to be rejected during registration. The value range is 30-100 (default is 40).

Note: To reduce the rejection rate in the field, this value may be set higher than 40 (such as 60 or 80). Low quality fingerprints would then be rejected during the registration time.

SI_VERIFY_QUALITY (0x0c)

Configures the value for fingerprint verification image quality. This parameter is only used for fingerprint verification, and determines the quality of the fingerprint images that should be captured. Higher settings will require high quality fingerprints, but may also cause users with poor quality fingerprints to be rejected during verification. The value range is 10-100 (default is 30).

Note: To reduce the rejection rate in the field, this value may be set lower than 30 (such as 20 or 35). Low quality fingerprints would then be rejected during the registration time.

SI_RELAY_TIME (0x0e)

Configures the value for duration time in milliseconds, which corresponds to the length of time the relay stays "on" for the command "Auto Identify" (0-10000 = 0-10 seconds). The value specified must be greater than '0' otherwise "Auto Identify" will not work. This command is not the same as CMD_FP_AUTO_IDENTIFY. The duration the relay stays on is set by SI_RELAY_TIME. Auto identify relay time is used in the following cases:

1. Environments where a controller is used to only enroll and delete users.
2. Hybrid environments where 1:N identification is used.

For more information, refer to SI_USE_RELAY (0x03) or SI_WIEGAND_FORMAT (0x09).

SI_WIEGAND_SITECODE (0x0a)

Sets the sitecode of Wiegand Protocol. The sitecode field is 8 bits when WIEGAND_FORMAT_26BIT is used, and 16 bits when WIEGAND_FORMAT_34BIT is used.

SI_USE_DATA_CHECKSUM (0x10)

Enables the ExtraData checksum function. The values of the first 11 bytes (excluding the checksum byte) are summed and then divided by 0x0100 (256). This will create a 1byte shift. The remaining value from this shift is the checksum byte. The checksum can be used to assure correct data transfer for remote registration.

SI_USE_AUTO_ON (0x11)

SDA04 uses a rapidly blinking light source (LED) in the optic module during user enrollment or verification to prevent false acceptance of latent fingerprints. Some large-scale sites using the SDA04 for relatively low-security applications (e.g. time & attendance systems) may consider turning this feature off for improved performance using the high security option.

Command Packet:	
Channel	0x00
Command	0x20
Param1	0x11: SI_USE_AUTO_ON
Param2	1: Blinking process on (for high security) 0: Not used (low security, faster performance)
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:	
Channel	0x00
Command	0x20
Param1	
Param2	
lwExtraData	0x0000
hwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error
CheckSum	

SI_ADAPTIVE_CAPTURE (0x12)

Enables the *Smart Capture* Function, which is newly implemented in SDA04 to capture a stable and optimized image from the sensor by adjusting the imaging parameters. If this option is "on", the brightness setting cannot be customized.

SI_TEMPLATETYPE (0x18)

Set the template type. SDA04 supports both ANSI INCITS 378-2004 and SecuGen proprietary 400 byte

fingerprint templates. Param1 should be set as follows:

ANSI378 – 0x0100
SG400 – 0x0200

Command to use ANSI378 Format:

Channel	0x00
Command	0x18
Param1	0x0100 (ANSI378)
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x18
Param1	0x0100
Param2	
lwExtraData	0x0000
hwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error
CheckSum	

Command to use SG400 Format:

Channel	0x00
Command	0x18
Param1	0x0200 (SG400)
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x18
Param1	0x0200
Param2	
lwExtraData	0x0000
hwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error
CheckSum	

Recommended System Parameters for Time & Attendance Applications

Strategy

1. Fast user verification by turning off high-security mode
2. Indoor usage only. Latent fingerprints might be accepted if there is any light source directed towards the sensor window.
3. Target lower rejection rate
4. Decrease security level (but it may increase false acceptance rate)
5. Increase registration quality level (may increase failure to enroll rate) if alternative methods such as

tokens or PINs are available.
6. Decrease verification quality level (lower false rejection rate)

	Parameter	Value	Result
0x02	SI_VERIFY_SECU_LEVEL	2 ~ 3	Lower FRR, higher FAR
0x0B	SI_REGISTER_QUALITY	40~80	Lower FRR, higher FTE
0x0C	SI_VERIFY_QUALITY	20	Increased acceptance
0x11	SI_USE_AUTO_ON	0	Avoids latent fingerprint problem
0x12	SI_ADAPTIVE_CAPTURE	1	Dynamic Auto Tuning

See also: CMD_FP_VERIFY (if PIN is used), CMD_FP_IDENTIFY (for a small # of users in database)

Recommended System Parameters for Door Lock Applications

Strategy
1. Target secure authentication
2. Target robustness against environmental factors (outdoors)
3. Target maximum convenience
4. Tips for low false rejection rate
5. Registering more fingerprints per finger with different IDs would decrease FRR by maintaining a stead y FAR

	Parameter	Value	Result
0x02	SI_VERIFY_SECU_LEVEL	3~5	Lower FAR
0x08	SI_IDENTIFY_SECU_LEVEL	5~6	5
0x0B	SI_REGISTER_QUALITY	40	Default
0x0C	SI_VERIFY_QUALITY	30	Default
0x11	SI_USE_AUTO_ON	1	Avoids latent fingerprint problem
0x12	SI_ADAPTIVE_CAPTURE	1	Dynamic Auto Tuning

See also: CMD_FP_IDENTIFY, CMD_FP_AUTO_IDENTIFY, CMD_FP_AUTO_IDENTIFY_STOP, SI_IDENTIFY_RANGE (For SDA04 V1.2x only)

Recommended System Parameters for Smart Card (Token) Applications

Strategy
1. Target secure 1:1 verification
2. Target compatible authentication with a PC application
3. Tips for low false rejection rate
4. Registering more fingerprints per finger would decrease FRR by maintaining a steady FAR

	Parameter	Value	Result
0x02	SI_VERIFY_SECU_LEVEL	3~5	Lower FAR
0x0B	SI_REGISTER_QUALITY	40	Default
0x0C	SI_VERIFY_QUALITY	30	Default
0x11	SI_USE_AUTO_ON	1	Avoids latent fingerprint problem
0x12	SI_ADAPTIVE_CAPTURE	1	Dynamic Auto Tuning

See also: CMD_INSTANT_VERIFY (SI_ADAPTIVE_CAPTURE = 0)

Recommended System Parameters for High Temperature Environments

Strategy
1. Use Smart Capture to automatically compensate for temperature effects on fingerprint capture

	Parameter	Value	Result
0x02	SI_VERIFY_SECU_LEVEL	3~5	Lower FAR

0x05	SI_BRIGHTNESS	100	Default
0x07	SI_EXP_GAIN	20	Default
0x0B	SI_REGISTER_QUALITY	40	Default
0x0C	SI_VERIFY_QUALITY	30	Default
0x11	SI_USE_AUTO_ON	1	Avoids latent fingerprint problem
0x12	SI_ADAPTIVE_CAPTURE	1	Dynamic Auto Tuning
0x13	SI_USING_DIFF_FP_REGISTER	1	Registers more fingerprints

Recommended System Parameters for Dry Finger Conditions

Strategy

1. Longer exposure (brightness) for higher image quality
2. Higher registration quality levels may reject very low quality, dry fingerprints

	Parameter	**Value**	**Result**
0x02	SI_VERIFY_SECU_LEVEL	2~3	Lower FAR
0x05	SI_BRIGHTNESS	500	Increases exposure
0x07	SI_EXP_GAIN	15	Decreases gain noise
0x0B	SI_REGISTER_QUALITY	70	Rejects low-quality fingerprint images
0x0C	SI_VERIFY_QUALITY	30	Default
0x11	SI_USE_AUTO_ON	0 or 1*	Avoids latent fingerprint problem
0x12	SI_ADAPTIVE_CAPTURE	0	Applies custom setting
0x13	SI_USING_DIFF_FP_REGISTER	1	Registers more fingerprints

* Use a value of '0' for indoor usage only

CMD_SET_COMM_SPEED (0x21)

Commenté [d15]: Errors at some baud rates

Changes the speed of the current channel. The specified speed value is passed to Param2. Available speeds for the serial port are listed below. Default is BAUD_115200 This value is not saved in the SDA04 flash memory. When the SDA04 is rebooted, the baud rate will be reset to default.

Param2 Communication Speeds	
0x00F3	BAUD_921600 (SDA04, SDA04M)
0x00F2	BAUD_460800
0x00F1	BAUD_230400
0x0001	BAUD_115200
0x0003	BAUD_57600
0x0005	BAUD_38400
0x000B	BAUD_19200
0x0017	BAUD_9600
0x005F	BAUD_2400
0x00BF	BAUD_1200

Command Packet:

Channel	0x00
Command	0x21
Param1	
Param2	Communication speed of serial port
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x21
Param1	
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error
CheckSum	

CMD_GET_SYSTEM_INFO (0x30)

Gets the current system setting values from the SDA04. The information to be retrieved is stored in Param1. The SDA04 returns the current system setting values in Param2.

SI_USE_MASTER_AUTHENTICATION (0x00)

Returns whether or not master authentication is used. If master authentication is turned off, any user can change a fingerprint in the SDA04 database. If master authentication is turned on, the fingerprint of a master is required before changes to the database are allowed.

1 = master authentication is used.
0 = master authentication is not used

SI_VERIFY_SECU_LEVEL (0x02)

returns SDA04 security level. Read *SDA04 Security Level Basics* in section 3.3.4 before establishing your security policy.

SI_USE_RELAY (0x03)

Determines if the SDA04 relay is being used

SI_COM_SPEED (0x04)

Returns communication speed of the current channel

SI_BRIGHTNESS (0x05)

Returns coarse value of exposure in the optic module

SI_IDENTIFY_SECU_LEVEL (0x08)

Returns security level for the Identify setting of the current channel (default is 8)

SI_WIEGAND_FORMAT (0x09)

Returns Wiegand Protocol format of current channel

1. 0x0000 WIEGAND_NOT_USED
2. 0x012c WIEGAND_FORMAT_26BIT
3. 0x012e WIEGAND_FORMAT_34BIT

SI_WIEGAND_SITECODE (0x0a)

Returns Wiegand Protocol Sitecode of current channel

SI_REGISTER_QUALITY (0x0b)

Returns value for fingerprint registration image quality (default is 40)

SI_VERIFY_QUALITY (0x0c)

Returns value for fingerprint verification image quality (default is 30)

SI_RELAY_TIME (0x0e)

Returns value for duration time in milliseconds, which determines how long the relay stays "on" for Auto-Identify. The Auto-Identify relay time is used in the following cases:

1. Environments where a controller is used only to enroll and delete users.
2. Hybrid environments where 1-to-few identification is used.

Note: SI_RELAY_TIME is different from the command CMD_FP_AUTO_IDENTIFY.

****See CMD_SET_SYSTEMINFO (x020) for details of the following parameters:***

SI_USE_DATA_CHECKSUM (0x10)
SI_USE_AUTO_ON (0x11)
SI_ADAPTIVE_CAPTURE (0x12)
SI_TEMPLATETYPE (0x18)

CMD_GET_MINUTIAE (0x40) (ANSI378, SG400)

ANSI378 SG400

Use this command to make a single-view ANSI378 or SG400 fingerprint record by capturing a live fingerprint image and sending it to the host. The SDA04 sends out an ACK packet followed by the SG400 or ANSI378 record immediately after fingerprint capture. To select SG400 or ANSI378 mode, set the SI_TEMPLATETYPE system parameter using CMD_SET_SYSTEM_INFO. The lwExtraData of the ACK packet contains the ANSI378 or SG400 record size.

ANSI378 Mode

ANSI378

Command Packet

Channel	0x00
Command	0x40
Param1	Finger position number (see constant list)
Param2	0x0000
lwExtraData	0x0000
hwExtraData	0x0000
ErrorCode	0x00
Checksum	Checksum

Example:

0x00	**0x40**	**0x0001**	0x0000	0x0000	0x0000	0x00	Chksum

ACK Packet

Channel	0x00
Command	0x40
Param1	Finger position number
Param2	0x0000
lwExtraData	ANSI378 FP Record size
hwExtraData	0x0000
ErrorCode	0x00
Checksum	Checksum
ExtraData	ANSI378 FP Record (size = lwExtraData)

Example:

0x00	**0x40**	**0x0001**	0x0000	**0x0100**	0x0000	0x00	Chksum	FMR …

SG400 Mode

SG400

Gets minutiae data from the fingerprint on the fingerprint sensor. The fingerprint should be input continuously for high quality images. The fingerprint quality value in Param1 corresponds to the quality of the image. If the value is '0', processing will end after only one capture. In general, the value should be set to at least 30 for matching and at least 40 for fingerprint registration.

Command Packet:

Channel	0x00
Command	0x40
Param1	Quality (0-100)
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

Commenté [d16]: ?? Confirm Param1 is still used as referenced for SG400 mode

Example:

0x00	**0x40**	**0x0001**	0x0000	0x0000	0x0000	0x00	Chksum

ACK Packet:

Channel	0x00
Command	0x40
Param1	
Param2	
LwExtraData	0x190 (= 400 bytes)
HwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error M2ERROR_SENSOR_OPEN – Error on image sensor M2ERROR_TIMEOUT – No fingerprint on sensor, or input error
CheckSum	

The sequence of data:

400 byte minutiae data…

Example:

0x00	**0x40**	0x0000	0x0000	**0x0190**	0x0000	0x00	Chksum

CMD_GET_IMAGE (0x43)

Captures fingerprint image of finger placed on the sensor and sends the image to the main controller. The main controller sets the viewing value of the image in Param1, which can be set to the following:

Param1
0x01: VIEW_NORMAL (0x01) – Original image size
0x02: VIEW_HALF (0x02) – Half of the original image size
0x04: VIEW_QUARTER (0x04) – One-fourth of the original image size

The length of the data to be transferred depends on the image size. If there is no error, the SDA04 sends the ACK packet and then sends the image and ExtraData to the main controller.

Command Packet:

Channel	0x00
Command	0x43
Param1	Size of fingerprint image (0x01, 0x02, 0x04)
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x43
Param1	
Param2	
LwExtraData	Low-word of data length to be transmitted
HwExtraData	High-word of the data length to be transmitted
ErrorCode	M2ERROR_NONE – No error M2ERROR_SENSOR_OPEN – Image sensor error
CheckSum	

The sequence of data:

The length of ExtraData (Image Size)
Image Data…

CMD_GET_REC_COUNT (0x46) (SG400)

SG400

Gets the total number of registered SDA04 users. To find the number of masters registered in the system, use the command CMD_GET_MASTER_COUNT.

Command Packet:

Channel	0x00
Command	0x46
Param1	
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x46
Param1	The number of registered users in the system
Param2	The number of remaining available records
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error
CheckSum	

CMD_GET_MASTER_COUNT (0x47) (SG400)

SG400

Gets the number of registered masters in the system. The maximum number of masters that can be registered is 5.

Command Packet:

Channel	0x00
Command	0x47
Param1	
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x47
Param1	Number of registered masters(0-5)
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error
CheckSum	

CMD_FP_REGISTER_START (0x50) (SG400) SG400

Initiates fingerprint registration. This is the first step in the user enrollment process. The same fingerprint must be input twice to complete registration. The first fingerprint is captured by the "register start" command, which sends the User ID in Param1 and master flag (if master verification is required) in Param2. Following the ACK, the first fingerprint is captured. Users should note that the red LED will blink rapidly, signaling that the user should now place a finger on the fingerprint sensor for image capture. When the red LED stops blinking, the user must remove the finger from the fingerprint sensor.

The second fingerprint is captured by the "register end" command. The "register start" and "register end" commands cannot be executed independently and must be performed in the following order:

1. CMD_FP_VERIFY_MASTER (optional – see the command description for information on using this command)
2. CMD_FP_REGISTER_START
3. CMD_FP_REGISTER_END

The CMD_FP_VERIFY_MASTER command is only valid if master authentication is enabled on the system wit h either CMD_SET_SYSTEM_INFO or CMD_FP_VERIFY_MASTER with *multiple authentications*.

Other commands of interest

CMD_FP_VERIFY_MASTER
CMD_DB_VERIFY_MASTER
CMD_SET_SYSTEM_INFO
CMD_FP_VERIFY_MASTER_END
CMD_DB_VERIFY_MASTER_END

Command Packet:

Channel	0x00
Command	0x50
Param1	User ID
Param2	1 – admin, 0 – user
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x50
Param1	
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	M2ERROR_NONE – Performed successfully M2ERROR_TIMEOUT – No fingerprint on sensor, or fingerprint input failed M2ERROR_ALREADY_REGISTERED_USER – User has already been enrolled M2ERROR_DB_FULL – No more space for user registration in database M2ERROR_MASTER_COUNT_EXCEED – Exceeded maximum number of masters (applicable only when registering masters)
CheckSum	

CMD_FP_REGISTER_END (0x51) (SG400) SG400

This command is issued after the SDA04 returns M2ERROR_NONE (command successful) following the CMD_FP_REGISTER_START command. Whereas the CMD_FP_REGISTER_START command captured the first image, the second image is captured by the CMD_FP_REGISTER_END command. The SDA04 returns M2ERROR_NONE in the ACK packet if registration is successful; if not, an error code is returned. This command cannot be executed independently and must be performed in the following order:

1. CMD_FP_VERIFY_MASTER (optional)
2. CMD_FP_REGISTER_START
3. CMD_FP_REGISTER_END

Note: The values in the Param1, Param2 fields of CMD_FP_REGISTER_START and CMD_FP_REGISTER_END should be identical.

Command Packet:

Channel	0x00
Command	0x51
Param1	User ID
Param2	1 – admin, 0 – user
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x51
Param1	
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error M2ERROR_TIMEOUT – No fingerprint on sensor, or fingerprint input failed M2ERROR_REGISTER_FAILED – Registration failed M2ERROR_MASTERCOUNT_EXCEED – Master count exceeded M2ERROR_FLASH_WRITE_ERROR – Writing in flash memory failed
CheckSum	

CMD_FP_CHANGE_START (0x52) (SG400)

SG400

Used to change previously registered fingerprint data. This is the first step in the user fingerprint change process. The same fingerprint must be input twice to complete the change. The first fingerprint is captured by the "change start" command, which sends the User ID in Param1. Following the ACK, the first fingerprint is captured. Users should note that the red LED will blink rapidly, signaling that the user should now place a finger on the fingerprint sensor for image capture. When the red LED stops blinking, the user must remove the finger from the fingerprint sensor.

The second fingerprint is captured by the "change end" command. The "change start" and "change end" commands cannot be executed independently and must be performed in the following order:

1. CMD_FP_VERIFY
2. CMD_FP_CHANGE_START
3. CMD_FP_CHANGE_END

Command Packet:

Channel	0x00
Command	0x52
Param1	User ID
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x52
Param1	
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error M2ERROR_TIMEOUT – No fingerprint on sensor, or fingerprint input failed M2ERROR_USER_NOT_FOUND – User ID to change is not found
CheckSum	

CMD_FP_CHANGE_END (0x53) (SG400) SG400

Completes fingerprint changes of registered users. This command is issued after the SDA04 returns M2ERROR_NONE (command successful) following the CMD_FP_CHANGE_START command. Whereas the CMD_FP_CHANGE_START command captured the first image, the second image is captured by the CMD_FP_CHANGE_END command. The SDA04 returns M2ERROR_NONE in the ACK packet if update is successful; if not, an error code is returned. This command cannot be executed independently and must be performed in the following order

1. CMD_FP_VERIFY
2. CMD_FP_CHANGE_START
3. CMD_FP_CHANGE_END

Command Packet:

Channel	0x00
Command	0x53
Param1	User ID
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x53
Param1	
Param2	
lwExtraData	0x0000
hwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error M2ERROR_TIMEOUT – No fingerprint on sensor, or fingerprint input failed M2ERROR_FPCHANGE_FAILED – Changing fingerprint failed M2ERROR_FLASH_WRITE_ERROR – Flash memory failed
CheckSum	

CMD_FP_DELETE (0x54) (SG400)

SG400

Deletes a registered user. To delete s user, a User ID must be specified in Param1. Master authentication is also required before users can be deleted. The following operations must be performed to delete users:

1. CMD_FP_VERIFY_MASTER (Optional)
2. CMD_FP_DELETE

The SDA04 returns M2ERROR_NONE in the ErrorCode with User ID in of the ACK packet if deletion is successful; otherwise, an error code is returned.

The CMD_FP_VERIFY_MASTER command cannot be used if master authentication is disabled by the commands CMD_SET_SYSTEM_INFO or the CMD_FP_VERIFY_MASTER with multiple authentications.

Command Packet:

Channel	0x00
Command	0x54
Param1	UserID
Param2	
lwExtraData	0x0000
hwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x54
Param1	User ID
Param2	Master Flag
lwExtraData	0x0000
hwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error
CheckSum	

CMD_FP_VERIFY (0x55) (SG400)

SG400

Verifies users already registered in the system database. When a finger is placed on the fingerprint sensor, the command is issued with the User ID in Param1. The SDA04 returns M2ERROR_NONE in the ErrorCode with User ID in Param1 and master flag in Param2 of the ACK packet if verification is successful; otherwise, an error code is returned. Using the Wiegand protocol, this command only works when status is available for use (send Wiegand signal by relay).

Command Packet:

Channel	0x00
Command	0x55
Param1	User ID
Param2	Wiegand Output Flag ('0' generates Wiegand signal when Wiegand is available for use; '1' Does not generate Wiegand signal)
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x55
Param1	User ID
Param2	Master Flag
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error M2ERROR_TIMEOUT – No fingerprint on sensor, or fingerprint input failed M2ERROR_VERIFICATION_FAIL – Verification of captured fingerprint and registered fingerprint failed M2ERROR_DB_WRONG_USERID – User has not been registered
CheckSum	

Commenté [d17]: Confirm 1 or 0

CMD_FP_IDENTIFY (0x56) (SG400)

SG400

Identifies a user by searching the database of registered users for a match using only the fingerprint image with no accompanying User ID as input. If the user is identified, the User ID is then stored in Param1 and the master flag in Param2 of the ACK packet. Using the Wiegand protocol, this can be done only when the status is available (send Wiegand signal by relay).

Note: During long processing time commands such as Identification and Capture, host controller can terminate the process by sending 4 bytes of break command sequence.
Command Sequence: 0xA8, 0xB8, 0xC8, 0xD8
Acknowledge: Normal Acknowledge format of each command with Break Error Code of 0xFE

Commenté [d18]: Does this still apply?

Command Packet:

Channel	0x00
Command	0x56
Param1	
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x56
Param1	User ID
Param2	Master Flag
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error M2ERROR_TIMEOUT – No fingerprint on sensor, or fingerprint input failed M2ERROR_IDENTIFY_FAILED – No fingerprint in database, does not identify fingerprint on fingerprint reader at all
CheckSum	

Commenté [d19]: Currently returning M2ERROR_FUNCTION_FAILED(0x10). Should be M2ERROR_IDENTIFY_FAILED

CMD_FP_VERIFY_MASTER (0x57) (SG400)

SG400

Verifies the master before executing a command that requires master authentication, such as deleting a user from the database. After a master finger is placed on the fingerprint sensor, the main controller sends the CMD_FP_VERIFY_MASTER command to the SDA04. The Param1 field specifies the commands (command count) to be authenticated by the master verification. All commands following the **0** and ***n*** commands will be authenticated by the master fingerprint verification. Commands can be repeated by setting 0 or greater than 1 in Param1 field. This can be used for getting data from the SDA04 to a server sequentially. The SDA04 returns M2ERROR_NONE in ErrorCode with the master's User ID in Param1 of the ACK packet if verification is successful; otherwise, an error code is returned.

Command Packet:

Channel	0x00
Command	0x57
Param1	0: All commands after this command will be authenticated by master verification. 1: Only one command after this command will be authenticated. n: Specifies the number of commands to be authenticated.
Param2	
lwExtraData	0x0000
hwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x57
Param1	Master User ID
Param2	
lwExtraData	0x0000
hwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error M2ERROR_TIMEOUT – No fingerprint on sensor, or fingerprint input failed M2ERROR_MASTERFP_NOT_FOUND – No registered master in system M2ERROR_VERIFY_FAIL – Master's fingerprint not matched
CheckSum	

Param1

0	All commands after this command will be authenticated by master verification.
1	Only one command after this command will be authenticated.
n	Specify the number of commands to be authenticated (up to 65,535).

The authenticated command count can be deleted by CMD_FP_VERIFY_MASTER END or CMD_DB_VERIFY_MASTER END and can be reset by CMD_FP_VERIFY_MASTER or CMD_DB_VERIFY_MASTER. For tighter security, set '1' in Param1 for commands requiring authentication.

CMD_FP_VERIFY_MASTER_END (0x58) (SG400) SG400

Invalidates the number of commands left during authentication by the command CMD_FP_VERIFY_MASTER. For example, if three authenticated commands remain, they will be reset by this command. (A master verification fingerprint reader is needed by the CMD_FP_VERIFY_MASTER command to perform commands requiring authentication.)

Command Packet:

Channel	0x00
Command	0x58
Param1	
Param2	
lwExtraData	0x0000
hwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x58
Param1	
Param2	
lwExtraData	0x0000
hwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error
CheckSum	

CMD_FP_IDENTIFY_EX (0x59) (SG400)

SG400

Used in the same way as CMD_FP_IDENTIFY. It is also possible to set the period for the Identify operation, increasing the speed it takes to get a User ID. If the input fingerprint is matched with a registered users' fingerprint data in the system database, M2ERROR-NONE is returned, with the User ID stored in Param1 and the master flag in Param2 of the ACK packet. Using the Wiegand protocol, this can be done only when the status is available (send Wiegand signal by relay).

Note: This command searches specified blocks. If a fingerprint is not found in the block specified, ERROR Code will show the non-match result.

Commenté [d20]: I don't think we search in blocks

Command Packet:

Channel	0x00
Command	0x59
Param1	First ID number of search
Param2	Last ID number of search
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x59
Param1	User ID
Param2	Master Flag
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error M2ERROR_TIMEOUT – No input fingerprint or error occurred M2ERROR_IDENTIFY_FAILED – No fingerprint in database matches the input fingerprint
CheckSum	

CMD_IS_REGISTERED_USER (0x60) (SG400) SG400

Checks whether a specified user is registered. The master flag can be checked in Param2.

Command Packet:

Channel	0x00
Command	0x60
Param1	User
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x60
Param1	User ID
Param2	Master Flag
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	M2ERROR_NONE – Registered user M2ERROR_USER_NOT_FOUND – Not a registered user
CheckSum	

CMD_DB_GET_RECCOUNT (0x70) (SG400)

SG400

Gets the number of user records stored in the system. It is equal to CMD_GET_RECCOUNT.

Command Packet:

Channel	0x00
Command	0x70
Param1	
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x70
Param1	Number of records in Database.
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error
CheckSum	

CMD_DB_ADD_REC (0x71) (SG400)

SG400

This is the protocol used for remote registration and modification of SDA04 user records. Registering large numbers of users may require declaring a double word size of TuserRecord. The low-word value of the double word is stored in LwExtraData, and the high-word value of the double word is stored in HwExtraData. If the SDA04 receives data of any size other than the declared size, the error message for M2ERROR_INSUFFICIENT_DATA will be returned by the SDA04. When transferring large amounts of data, it is recommended to use Checksum.

Note: This command requires writing to the flash memory on the SDA04. Intel specifies a flash memory capability of 100,000 write cycles.

Commenté [d21]: Does this apply to SDA04?

It is possible to set the overwrite option in Param1 if there is a pre-registered ID. If the value is not '0', be sure to remove the pre-registered ID. If a duplicate ID exists, the error message, M2ERROR_AREADY_REGISTERED_USER will be returned by the SDA04.

Note: Processing may fail when adding a user if Minutiae1 and Minutiae2 contain identical data when SI_USING_DIFF_FP_REGISTER is off. This is why users must lift their finger off the fingerprint sensor momentarily before placing it back again for the second (final) fingerprint capture.

Command Packet:

Channel	0x00
Command	0x71
Param1	Overwrite Flag 0: Do not overwrite if there is a pre-registered ID 1: Overwrites the new ID on a pre-registered ID
Param2	
LwExtraData	Low-word of Sizeof (TuserRecord) → 0x0330 (=816 byte = 1 user record)
HwExtraData	High-word of Sizeof (TuserRecord) → 0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x71
Param1	
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error M2ERROR_REGESTER_FAILED – Registration failed M2ERROR_FLASH_OPEN – Error on flash memory M2ERROR_INSUFFICIENT_DATA – Data size is not equal to ExtraData M2ERROR_ALREADY_REGISTERED_USER – User has already been registered M2ERROR_DB_FULL – No room for additional users in database M2ERROR_MASTERCOUNT_EXCEED – Exceeded maximum number of masters
CheckSum	

The sequence of data:

2 Bytes	2 Bytes	400 Bytes	400 Bytes	12 Bytes
UserID	MasterFlag	Mintuiae1	Minutiae2	TimeInfo

CMD_DB_DELETE_REC (0x72) (SG400)

SG400

Deletes a specified user from the database. The SDA04 returns M2ERROR_NONE in ErrorCode with User ID in Param1 and master flag in Param2 of the ACK packet if deletion is successful; otherwise, an error code is returned.

Command Packet:	
Channel	0x00
Command	0x72
Param1	User ID
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	
ACK PACKET:	
Channel	0x00
Command	0x72
Param1	User ID
Param2	Master Flag
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error
	M2ERROR_DB_NO_DATA – No registered user in database
	M2ERROR _USER_NOT_FOUND – Not a registered user
CheckSum	

CMD_DB_GET_REC (0x73) (SG400)

SG400

Gets user records from the SDA04 database by User ID. The SDA04 transmits user records sequentially.

Command Packet:

Channel	0x00
Command	0x73
Param1	User ID
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x73
Param1	
Param2	
LwExtraData	Low-word of Sizeof (TUserRecord)
HwExtraData	High-word of Sizeof (TUserRecord)
ErrorCode	M2ERROR_NONE – No error
	M2ERROR_ USER_NOT_FOUND – Not a registered user
CheckSum	

The sequence of data:

2 Bytes	2 Bytes	400 Bytes	400 Bytes	12 Bytes
UserID	MasterFlag	Minutiae1	Minutiae2	TimeInfo

Note: ExtraData is valid only when ErrorCode is M2ERROR_NONE.

CMD_DB_GET_FIRSTREC (0x74) (SG400) SG400

Gets the first user record from the database, after which the SDA04 sends user records sequentially following the ACK packet. Refer to the commands CMD_DB_GET_NEXTREC and CMD_DB_GET_CURRENTREC.

Command Packet:

Channel	0x00
Command	0x74
Param1	
Param2	
LwExtraData	0
HwExtraData	0
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x74
Param1	
Param2	
LwExtraData	Low-word of Sizeof (TUserRecord)
HwExtraData	High-word of Sizeof (TUserRecord)
ErrorCode	M2ERROR_NONE – No error
	M2ERROR_DB_NO_DATA – No registered user in database
CheckSum	

The sequence of data:

2 Bytes	2 Bytes	400 Bytes	400 Bytes	12 Bytes
UserID	MasterFlag	Minutiae1	Minutiae2	TimeInfo

CMD_DB_GET_NEXTREC (0x75) (SG400)

SG400

Gets user records after the first record (CMD_DB_GET_FIRSTREC or CMD_DB_GET_NEXTREC). The SDA04 sends user records sequentially following the ACK packet. Refer to the CMD_DB_GET_FIRSTREC and CMD_DB_GET_CURRENTREC commands.

Command Packet:

Channel	0x00
Command	0x75
Param1	
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x75
Param1	
Param2	
LwExtraData	Low-word of Sizeof (TUserRecord)
HwExtraData	High-word of Sizeof (TuserRecord)
ErrorCode	M2ERROR_NONE – No error M2ERROR_DB_NO_DATA – No registered user in database
CheckSum	

The sequence of data:

2 Bytes	2 Bytes	400 Bytes	400 Bytes	12 Bytes
UserID	MasterFlag	Minutiae1	Minutiae2	TimeInfo

CMD_DB_DELETE_ALL (0x76) (SG400) SG400

Clears the database of all user data. It is strongly recommended for regular use to erase individual IDs using the CMD_DB_DELETE_REC (0x72) command with a specific ID instead of the CMD_DB_DELETE_ALL command.

Command Packet:

Channel	0x00
Command	0x76
Param1	
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
Checksum	

ACK PACKET:

Channel	0x00
Command	0x76
Param1	
Param2	
lwExtraData	0x0000
hwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error
Checksum	

CMD_DB_GET_CURRENTREC (0x77) (SG400)

SG400

Gets the last record obtained by the commands CMD_DB_GET_FIRSTREC or CMD_ DB_GET_NEXTREC. The SDA04 continually sends user records following the ACK packet. Refer to CMD_DB_GET_FIRSTREC and CMD_DB_GET_NEXTREC.

Command Packet:

Channel	0x00
Command	0x77
Param1	
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
Checksum	

ACK PACKET:

Channel	0x00
Command	0x77
Param1	
Param2	
LwExtraData	Low-word of Sizeof (TUserRecord)
HwExtraData	High-word of Sizeof (TuserRecord)
ErrorCode	M2ERROR_NONE – No error
	M2ERROR_DB_NO_DATA – No user registered in database
Checksum	

The sequence of data:

2 Bytes	2 Bytes	400 Bytes	400 Bytes	12 Bytes
UserID	MasterFlag	Minutiae1	Minutiae2	TimeInfo

CMD_DB_VERIFY (0x78) (SG400) SG400

Verifies users by fingerprint data. Fingerprints from the host to the SDA04 are compared with user fingerprint data saved in the database, and the results are returned. It is similar to the CMD_FP_VERIFY, but CMD_DB_VERIFY command, which gets the fingerprint input from the host instead of the fingerprint sensor. When using the Wiegand protocol, the relay must be turned off first. Please refer to SI_USE_RELAY for more information.

Command Packet:

Channel	0x00
Command	0x78
Param1	User ID
Param2	Wiegand Output Flag ('0' generates Wiegand signal when Wiegand is available for use; '1' does not generate Wiegand signal)
LwExtraData	0x0190(=400Byte)
HwExtraData	0x0000
ErrorCode	
Checksum	

Commenté [d22]: 1 or 0?

Sequence of transmitting data:
Fingerprint data 400 Bytes

ACK PACKET:

Channel	0x00
Command	0x78
Param1	
Param2	
lwExtraData	0x0000
hwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error M2ERROR_VERIFICATION_FAIL – Input fingerprint does not match registered fingerprint M2ERROR_DB_WRONG_USERID – User not registered. M2ERROR_INSUFFICIENT_DATA – Error on transmission of fingerprint data
Checksum	

CMD_DB_IDENTIFY (0x79) (SG400)

SG400

Fingerprint data is sent without a User ID and compared to all registered fingerprint data in the database, returning the results. It is similar to the CMD_FP_IDENTIFY but CMD_DB_IDENTIFY command, which gets fingerprint input from the host instead of the fingerprint sensor. After successful identification of a registered user, the ACK packet returns M2ERROR_NONE with User ID in Param1 and master check value in Param2. When using the Wiegand protocol, the relay must be turned off first. Please refer to SI_USE_RELAY for more information.

Command Packet:

Channel	0x00
Command	0x79
Param1	
Param2	Wiegand Output Flag ('0' generates Wiegand signal when Wiegand is available for use; '1' does not generate Wiegand signal)
LwExtraData	0x0190(=400Byte)
HwExtraData	0x0000
ErrorCode	
Checksum	

Commenté [d23]: 1 or 0?

Sequence of transmitting data:

Fingerprint data 400 Bytes

ACK PACKET:

Channel	0x00
Command	0x79
Param1	User ID
Param2	master Flag
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error M2ERROR_IDENTIFY_FAIL – No registered fingerprint matches input fingerprint M2ERROR_INSUFFICIENT_DATA – Error in fingerprint transmission
Checksum	

CMD_DB_IDENTIFY_EX (0x7a) (SG400)

SG400

Used in the same way as the CMD_DB_IDENTIFY command, but increases the speed and efficiency of the Identify operation by targeting "zones" of ID ranges. When a fingerprint is identified with a registered ID, the error code of the ACK packet will be M2ERROR_NONE with User ID in Param1 and master flag in Param2. When using the Wiegand protocol, the relay must be turned off first. Please refer to SI_USE_RELAY for more information. (When '1' is set in Param2, a Wiegand signal is not generated.)

Command Packet:

Channel	0x00
Command	0x7a
Param1	First ID number of searching area
Param2	Last ID number of searching area
LwExtraData	0x0190(=400Byte)
HwExtraData	0x0000
ErrorCode	
Checksum	

Sequence of transmitting data:

Fingerprint data 400 Bytes

ACK PACKET:

Channel	0x00
Command	0x7a
Param1	User ID
Param2	Master Flag
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error
	M2ERROR_IDENTIFY_FAIL – No registered user in database
	M2ERROR_INSUFFICIENT_DATA – Fingerprint was not transmitted
Checksum	

CMD_DB_VERIFY_MASTER (0x7b) (SG400) SG400

Verifies the master by fingerprint. This command has the same function as the CMD_FP_VERIFY_MASTER command, but CMD_DB_VERIFY_MASTER gets the fingerprint input from the host instead of the fingerprint sensor. Specify the commands (command count) that will be validated by the master fingerprint verification in Param1 (example: CMD_FP_VERIFY_MASTER).

Command Packet:

Channel	0x00
Command	0x7b
Param1	0: All commands after this command will be authenticated by master verification. 1: Only one command after this command will be authenticated. n: Specifies the number of commands to be authenticated.
Param2	
LwExtraData	0x0190(=400Byte)
HwExtraData	0x0000
ErrorCode	
Checksum	

Sequence of transmitting data:

Fingerprint data 400 Bytes…

ACK PACKET:

Channel	0x00
Command	0x7b
Param1	Master User ID
Param2	
lwExtraData	0x0000
hwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error M2ERROR_AUTHENTICATION_FAILED – No match for master's fingerprint M2ERROR_INSUFFICIENT_DATA – Incorrect transmission of fingerprint data
Checksum	

Param1

0	All commands after this command will be authenticated by master verification.
1	Only one command after this command will be authenticated.
n	Specifies the number of commands to be authenticated (up to 65,535).

The authenticated command count can be reset using the following commands: CMD_DB_VERIFY_MASTER_END or CMD_FP_VERIFY_MASTER. Refer to CMD_FP_VERIFY_MASTER for details.

CMD_DB_VERIFY_MASTER_END (0x7c) (SG400) SG400

Invalidates the number of commands left during authentication by the CMD_DB_VERIFY_MASTER command. For example, if three authenticated commands remain, they will be reset by this command. (A master verification fingerprint is needed by the CMD_DB_VERIFY_MASTER command to perform commands requiring authentication, in this case from the host instead of the fingerprint sensor.)

Command Packet:

Channel	0x00
Command	0x7c
Param1	
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
Checksum	

ACK PACKET:

Channel	0x00
Command	0x7c
Param1	
Param2	
lwExtraData	0x0000
hwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error
Checksum	

CMD_DB_GET_ID_LIST (0x7d) (SG400)

SG400

Gets a User ID list from the database. The SDA04 sends User IDs continuously after sending the ACK packet. The length of the User ID list = Total Number of Users * Sizeof (User ID).

Command Packet:

Channel	0x00
Command	0x7d
Param1	
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x7d
Param1	Total number of users
Param2	
LwExtraData	Low-word of (Total number of User * Sizeof (UserID))
HwExtraData	High-word of (Total number of User * Sizeof (UserID))
ErrorCode	M2ERROR_NONE – No error M2ERROR_DB_NO_DATA – No user registered
CheckSum	

The sequence of data:

2 Bytes	2 Bytes	…..	2 Bytes	2 Bytes
UserID	UserID	…..	UserID	UserID

CMD_DB_GET_MASTER_LIST (0x7e) (SG400) SG400

Gets the Master ID list from the database. The SDA04 sends the master ID list continuously after sending the ACK packet. The length of the Master ID list = Total Number of Masters * Sizeof (User ID).

Command Packet:

Channel	0x00
Command	0x7e
Param1	
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0x7e
Param1	Total number of Masters
Param2	
LwExtraData	Low-word of (Total number of Masters * Sizeof (UserID))
HwExtraData	High-word of (Total number of Masters * Sizeof (UserID))
ErrorCode	M2ERROR_NONE – No error
	M2ERROR_DB_NO_DATA – No master registered
CheckSum	

The sequence of data:

2 Bytes	.....	2 Bytes
UserID	.....	UserID

CMD_FP_AUTO_IDENTIFY (0xa1) (SG400)

SG400

Sets "Auto-Identify" mode. Calling this command starts the identify mode and runs until identification ends or the command CMD_FP_AUTO_IDENTIFY_STOP (0xa2) is called. No other commands can be used during this operation.

Unlike the Auto-Identify started by external switches, the CMD_FP_AUTO_IDENTIFY command runs by host command. The external switch Auto-Identify is not dependent on the host for commands.

Command Packet:

Channel	0x00
Command	0xa1
Param1	
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0xa1
Param1	
Param2	
LwExtraData	
HwExtraData	
ErrorCode	M2ERROR_NONE – No error
CheckSum	

CMD_FP_AUTO_IDENTIFY_STOP (0xa2) (SG400) SG400

Stops the Auto-Identify mode. It is only available if the CMD_FP_AUTO_IDENTIFY (0xa1) command is called in advance.

Command Packet:

Channel	0x00
Command	0xa2
Param1	
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	
CheckSum	

ACK PACKET:

Channel	0x00
Command	0xa2
Param1	
Param2	
LwExtraData	
HwExtraData	
ErrorCode	M2ERROR_NONE – No error M2ERROR_NOT_AUTO_ON_MODE – Not auto identify mode
CheckSum	

CMD_INSTANT_VERIFY (0xd0)(ANSI378,SG400) ANSI378 SG400

Use this command to match an ANSI378 or SG400 record against a live captured fingerprint.

In ANSI378 mode, multiple ANSI378 records having multiple fingerprint views inside can be sent. The SDA04 performs 1:1 matching with the live fingerprint and each of the fingerprint views in multiple ANSI378 records until it finds a match. If there is no match, the SDA04 sends out a "Verification Fail Error" message. If there is a match, it sends out the position of the matched fingerprint view.

In SG400 mode, the SDA04 performs 1:1 matching with the live fingerprint against all templates received until it finds a match. If there is no match, the SDA04 sends out a "Verification Fail Error" message. If there is a match, it sends out the position of the matched fingerprint template.

ANSI 378 Mode ANSI378

To send multiple ANSI378 records, simply place each ANSI378 record into the ExtraData buffer in serial order.

ExtraData

Record1 HEADER
View1
View2
Record2 HEADER
View1
View2
View3
View4
Record3 HEADER
View1

Command Packet

Channel	0x00
Command	0xD0
Param1	Number of ANSI378 records (not number of fingerprint views)
Param2	0x0000
lwExtraData	Size of total ANSI378 records
hwExtraData	0x0000
ErrorCode	0x00
Checksum	Checksum
ExtraData	ANSI378 records

Example:

0x00	**0xD0**	**0x0003**	0x0000	**0x0300**	0x0000	0x00	Chksum	**ExtraData**

ACK Packet

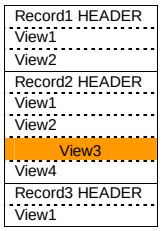
Channel	0x00
Command	0xD0
Param1	**Matched finger information1**
Param2	**Matched finger information2**
lwExtraData	0x0000
hwExtraData	0x0000
ErrorCode	verified 0x00 / fail 0x04
Checksum	Checksum

Example:

0x00	**0x37**	0x0000	0x0000	0x0000	0x0000	0x00	Chksum

Matched finger information1: The Finger Position Number (upper 8 bits) and the Finger View Number of the matched fingerprint view (lower 8bits). These numbers are directly extracted from the ANSI378 record. This information is useful when every fingerprint view has a finger position number and finger view number that are not UNKNOWN.

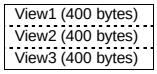
Matched finger information2: The View Number (upper 8 bits) and the ANSI378 Record Number (lower 8 bits).
For example, in the following figure, the third view of the second record is matched among the input ANSI378 records (View Number 3 and Record Number 2). This result is useful when every fingerprint view has an UNKNOWN finger position number and finger view number.



SG400 Mode

SG400

To send multiple SG400 templates, send the 400 byte template data in sequential order and specify the length of the template data in LwExtraData.



Command Packet:

Channel	0x00
Command	0xd0: CMD_INSTANT_VERIFY
Param1	
Param2	
LwExtraData	Size of total SG400 templates
HwExtraData	0x0000
ErrorCode	
CheckSum	

The following data must be sent after the Command Packet:

400byte templates totaling size specified in LwExtraData

ACK PACKET:

Channel	0x00
Command	0xd0
Param1	Matched template view information
Param2	
LwExtraData	0x0000
HwExtraData	0x0000
ErrorCode	M2ERROR_NONE – No error

	M2ERROR_VERIFY_FAILED – Verification failed M2ERROR_KEY_TIME_OVER – Timeout for the encryption key M2ERROR_INVALID_PARAM – Invalid ExtraData size (not 0x190 nor 0x320) M2ERROR_INSUFFICIENT_DATA – Exact data size not received (specified by ExtraData)
CheckSum	

Matched finger information: The position number of the matched fingerprint template will be returned in Param1.

CMD_MAKE_RECORD_START (0x35) (ANSI378,SG400) ANSI378 SG400

Use this command initiate internal variables and make the SDA04 ready to generate multiple-view ANSI378 or SG400 fingerprint records. The ACK of this command is the just reflection of the command packet. To select SG400 or ANSI378 mode, set the SI_TEMPLATETYPE system parameter using CMD_SET_SYSTEM_INFO.

Command Packet

Channel	0x00
Command	0x35
Param1	0x0000
Param2	0x0000
lwExtraData	0x0000
hwExtraData	0x0000
ErrorCode	0x00
Checksum	Checksum

Example:

0x00	**0x45**	0x0000	0x0000	0x0000	0x0000	0x00	Chksum

ACK Packet: The same as the Command Packet.

CMD_MAKE_RECORD_CONT (0x36) (ANSI378,SG400) ANSI378 SG400

Use this command to capture a live fingerprint image from the sensor and extract minutiae (template). The SDA04 does not send out the ANSI378 or SG400 record. Instead, it accumulates the intermediate template information in RAM until it receives the CMD_MAKE_RECORD_END command from the host. This command can be sent multiple times to pack multiple fingerprint views into one ANSI378 or multi-view SG400 record. To select SG400 or ANSI378 mode, set the SI_TEMPLATETYPE system parameter using CMD_SET_SYSTEM_INFO.

In ANSI378 mode, each time this command is sent, the finger position number should be sent together with it. The SDA04 increases the Fingerprint View number automatically, if there are fingerprint templates in RAM with a finger position number.

Command Packet

Channel	0x00
Command	0x35
Param1	Finger position number
Param2	0x0000
lwExtraData	0x0000
hwExtraData	0x0000
ErrorCode	0x00
Checksum	Checksum

Example:

0x00	**0x35**	**0x0001**	0x0000	0x0000	0x0000	0x00	Chksum

ACK Packet: The same as the Command Packet.

CMD_MAKE_RECORD_END (0x37) (ANSI378,SG400) ANSI378 SG400

Use this command to finalize and download the ANSI378 or SG400 record from the SDA04. To select SG400 or ANSI378 mode, set the SI_TEMPLATETYPE system parameter using CMD_SET_SYSTEM_INFO.

Command Packet

Channel	0x00
Command	0x37
Param1	0x0000
Param2	0x0000
lwExtraData	0x0000
hwExtraData	0x0000
ErrorCode	0x00
Checksum	Checksum

Example:

0x00	**0x37**	0x0000	0x0000	0x0000	0x0000	0x00	Chksum

ACK Packet

Channel	0x00
Command	0x37
Param1	0x0000
Param2	0x0000
lwExtraData	ANSI378 FP Record size
hwExtraData	0x0000
ErrorCode	0x00
Checksum	Checksum
ExtraData	ANSI378 FP Record (size = lwExtraData)

Example:

0x00	**0x37**	0x0000	0x0000	**0x0200**	0x0000	0x00	Chksum	Data …

Appendix B. ANSI378 Usage

B.1. Introduction

The SDA04 firmware supports the ANSI-INCITS 378-2004 standard finger minutiae format ("ANSI378"). For more information about this fingerprint template standard, refer to the standard document titled "Information technology - Finger Minutiae Format for Data Interchange", document # ANSI-INCITS 378-2004, available at the ANSI website http://webstore.ansi.org.

Refer to Appendix A for detailed description of SDA04 commands referenced below.

B.2. making ANSI378 Fingerprint Records

B.2.1. Getting ANSI378 fingerprint record with single view

CMD_GET_MINUTIAE (0x40)

B.2.2. Getting ANSI378 fingerprint record with multiple views

CMD_MAKE_RECORD_START (0x35)
CMD_MAKE_RECORD_CONT (0x36)
CMD_MAKE_RECORD_END (0x37)

The SDA04 makes multiple-view ANSI378 fingerprint records by capturing a live fingerprint image and sending it to the host.

To add additional finger views, repeat the CMD_MAKE_RECORD_CONT commands

CMD_MAKE_RECORD_START (0x35)
CMD_MAKE_RECORD_CONT (0x36)
:
:
CMD_MAKE_RECORD_CONT (0x36)
CMD_MAKE_RECORD_END (0x37)

B.3. Verifying ANSI378 Fingerprint Records

CMD_INSTANT_VERIFY (0xD0)

Appendix C. Command Summary

C.1. Basic Commands

Command	Code	Param1	Param2	ExtraData
Basic Control				
CMD_GET_VERSION	0x05	(major version)	(minor version)	
CMD_DEVICE_TEST	0x10	Device # to be tested		
CMD_RELAY_ONOFF	0x12	Relay num (0,1)	On – 1, Off –0	
CMD_LED_ONOFF	0x13	Led num (0,1)	On – 1, Off –0	
CMD_OPTICLED_ONOFF	0x14	On-1, Off –0		
CMD_EXP_AUTOTUNING	0x16			
CMD_FP_DIFF_REGISTER	0x19			
CMD_SET_SYSTEM_INFO	0x20	Selector	Value for selector	
CMD_SET_COMM_SPEED	0x21		Serial port speed	
CMD_GET_SYSTEM_INFO	0x30	Selector		
CMD_GET_IMAGE	0x43	Select size		(Image size)
Template & Record				
CMD_GET_MINUTIAE	0x40	Quality Threshold		(Size of minutiae)
CMD_GET_REC_COUNT	0x46			
CMD_GET_MASTER_COUNT	0x47			
ANSI 378 Template Generation				
CMD_MAKE_RECORD_START	0x35			
CMD_MAKE_RECORD_CONT	0x36	Finger position		
CMD_MAKE_RECORD_END	0x37			
CMD_GET_TEMPLATE	0x40	Finger position		(Size of minutiae)
Fingerprint Register & Verify				
CMD_FP_REGISTER_START	0x50	User ID		
CMD_FP_REGISTER_END	0x51	User ID		
CMD_FP_CHANGE_START	0x52	User ID		
CMD_FP_CHANGE_END	0x53	User ID		
CMD_FP_DELETE	0x54	User ID		
CMD_FP_VERIFY	0x55	User ID		
CMD_FP_IDENTIFY	0x56			
CMD_FP_VERIFY_MASTER	0x57	User ID		
CMD_FP_VERIFY_MASTER_END	0x58			
CMD_FP_IDENTIFY_EX	0x59	Starting User ID of search area	Ending User ID of search area	
CMD_IS_REGISTERED_USER	0x60	User ID		
Database control & verify (SG400 Mode Only- DB Commands do not support ANSI 378)				
CMD_DB_GET_RECCOUNT	0x70			
CMD_DB_ADD_REC	0x71	Overwrite flag		Size of Record

CMD_DB_DELETE_REC	0x72	User ID		
CMD_DB_GET_REC	0x73	User ID		
CMD_DB_GET_FIRSTREC	0x74			
CMD_DB_GET_NEXTREC	0x75			
CMD_DB_DELETE_ALL	0x76			
CMD_DB_GET_CURRENTREC	0x77			
CMD_DB_VERIFY	0x78	User ID	Wiegand Output Flag	Size of Template
CMD_DB_IDENTIFY	0x79			Size of Template
CMD_DB_IDENTIFY_EX	0x7A	Starting User ID of search area	Ending User ID of search area	
CMD_DB_VERIFY_MASTER	0x7B	Number of Authentication		
CMD_DB_VERIFY_MASTER_END	0x7C			
CMD_DB_GET_ID_LIST	0x7D	(Total number of users)		(Number of users * 2 bytes)
CMD_DB_GET_MASTER_LIST	0x7E	(Total number of masters)		(Number of masters * 2 bytes)
Instant verify				
CMD_INSTANT_VERIFY	0xD0	User ID		Size of template
Auto Identify				
CMD_FP_AUTO_IDENTIFY	0xA1			
CMD_FP_AUTO_IDENTIFY_STOP	0xA2			

Note: In the SDA04, all command-related settings are replaced in the CMD_SET_SYSTEM_INFO or CMD_GET_SYSTEM_INFO commands.

C.2. Database Access Commands (SG400 Template Mode Only)

Command	Code	Param1	Param2	ExtraData
CMD_DB_GET_RECCOUNT	0x70	Record count		
CMD_DB_ADD_REC	0x71	Overwrite flag		Size of User Record
CMD_DB_DELETE_REC	0x72			
CMD_DB_GET_REC	0x73	User ID		Size of User Record
CMD_DB_GET_FIRSTREC	0x74			Size of User Record
CMD_DB_GET_NEXTREC	0x75			Size of User Record
CMD_DB_DELETE_ALL	0x76			
CMD_DB_GET_CURRENTREC	0x77			Size of User Record
CMD_DB_VERIFY	0x78	User ID	Wiegand output flag MasterFlag	400
CMD_DB_IDENTIFY	0x79	(User ID)	MasterFlag	400
CMD_DB_IDENTIFY_EX	0x7a	(User ID)	MasterFlag	400
CMD_DB_VERIFY_MASTER	0x7b	# of actions		400
CMD_DB_VERIFY_MASTER_END	0x7c			
CMD_DB_GET_ID_LIST	0x7d	# of users		(# of users * 2 bytes)
CMD_DB_GET_MASTER_LIST	0x7e	# of masters		(# of masters * 2 bytes)

C.3. Auto Identify Commands (SG400 Template Mode Only)

Command	Code	Param1	Param2	ExtraData
CMD_FP_AUTO_IDENTIFY	0xa1			
CMD_FP_AUTO_IDENTIFY_STOP	0xa2			

C.4. Instant Verify Commands

Command	Code	Param1	Param2	ExtraData
CMD_INSTANT_VERIFY	0xd0			Size of template

C.5. ANSI 378 Commands

Command	Code	Param1	Param2	ExtraData
CMD_MAKE_RECORD_START	0x35			
CMD_MAKE_RECORD_CONT	0x36	Finger position		
CMD_MAKE_RECORD_END	0x37			
CMD_GET_TEMPLATE	0x40	Finger position		Size of minutiae
CMD_INSTANT_VERIFY	0xD0	User ID		Size of template

C.6. Commands Requiring Master Authentication

- CMD_SET_SYSTEM_INFO
- CMD_FP_REGISTER_START
- CMD_FP_REGISTER_END
- CMD_FP_REGISTER_START
- CMD_FP_REGISTER_END
- CMD_FP_DELETE
- CMD_DB_ADD_REC
- CMD_DB_DELETE_REC
- CMD_DB_GET_REC
- CMD_DB_GET_FIRSTREC
- CMD_DB_GET_NEXTREC
- CMD_DB_GET_CURRENTREC
- CMD_DB_DELETE_ALL

Appendix D. Data Types

```
// Type define
#ifndef UINT32
#define UINT32 unsigned int
#endif

#ifndef UINT16
#define UINT16 unsigned short
#endif

#ifndef UINT8
#define UINT8 unsigned char
#endif


// Command packet
typedef struct_tagCmdPacket
{
  UINT8   Channel;
  UINT16  Command;
  UINT16  Param1;
  UINT16  Param2;
  UINT16  lwExtraData;
  UINT16  hwExtraData;
  UINT8   ErrorCode;
  UINT8   Checksum;
} TCmdPacket;


// Time structure
typedef struct_tagTimeInfo
{
  UINT16 Year;      // 1999~
  UINT16 Month;     // 1 ~ 12
  UINT16 Day;       // 1 ~ 31
  UINT16 Hour;      // 00 - 23
  UINT16 Min;       // 0 - 59
  UINT16 Sec;       // 0 - 59
} TimeInfo;


// User Record
typedef struct  tagTUserRecord
{
  UINT16  UserId;
  UINT16  MasterFlag;
  UINT8   Minutiae1[400];
  UINT8   Minutiae2[400];
  TimeInfo Time;
```

```
} TUserRecord;


// Log Record
typedef struct tagLogRecord
{
  TimeInfo        Time;
  UINT16          UserId;
  UINT16          EventNum;
} TLogRecord;
```

Appendix E. Constants

• Security level

#define SLEVEL_LOWEST	0x01
#define SLEVEL_LOWER	0x02
#define SLEVEL_LOW	0x03
#define SLEVEL_BELOW_NORMAL	0x04
#define SLEVEL_NORMAL	0x05
#define SLEVEL_ABOVE_NORMAL	0x06
#define SLEVEL_HIGH	0x07
#define SLEVEL_HIGHER	0x08
#define SLEVEL_HIGHEST	0x09

• Image view

#define VIEW_NORMAL	0x01
#define VIEW_HALF	0x02
#define VIEW_QUARTER	0x04

• Template Format

#define ANSI378	0x0100
#define SG400	0x0200

• Device number

#define DEVICE_ALL	0x00
#define DEVICE_SENSOR	0x01
#define DEVICE_FLASHMEN	0x02

• Wiegand protocol format

#define WIEGAND_FMT_NOT_USED	0x00
#define WIEGAND_FMT_26BIT	0x12c (300)
#define WIEGAND_FMT_34BIT	0x12e (302)

• System setup parameter

#define SI_USE_MASTER_AUTHENTICATION	0x00
#define SI_VERIFY_SECU_LEVEL	0x02
#define SI_USE_RELAY	0x03
#define SI_COM_SPEED	0x04
#define SI_BRIGHTNESS	0x05
#define SI_IDENTIFY_SECU_LEVEL	0x08
#define SI_WIEGAND_FORMAT	0x09
#define SI_WIEGAND_SITECODE	0x0A
#define SI_REGISTER_QUALITY	0x0B

```
#define SI_VERIFY_QUALITY          0x0C
#define SI_EXP_CONTRAST            0x0D
#define SI_RELAY_TIME              0x0E
#define SI_USE_DATA_CHECKSUM       0x10
#define SI_USE_AUTO_ON             0x11
#define SI_ADAPTIVE_CAPTURE        0x12
#define SI_ID_DIGIT                0x16
#define SI_TEMPLATETYPE            0x18
```

ANSI378 Finger Position Number

```
#define UNKNOWN_FINGER             0x00
#define RIGHT_THUMB                0x01
#define RIGHT_INDEX                0x02
#define RIGHT_MIDDLE               0x03
#define RIGHT_RING                 0x04
#define RIGHT_LITTLE               0x05
#define LEFT_THUMB                 0x06
#define LEFT_INDEX                 0x07
#define LEFT_MIDDLE                0x08
#define LEFT_RING                  0x09
#define LEFT_LITTLE                0x0a
#define PLAIN_RIGHT_THUMB          0x0b
#define PLAIN_LEFT_THUMB           0x0c
#define PLAIN_RIGHT_FOUR_FP        0x0d
#define PLAIN_LEFT_FOUR_FP         0x0e
```

Appendix F. Error Codes (M2ERROR)

All error codes referenced in this document are summarized in the table below. M2ERROR prefix should be ins erted
before the error name below.

Code	Error Name	Description
0x00	_NONE	Performs the command received from main controller or host (no error)
0x01	_FLASH_OPEN	Command from main controller or host to access flash memory failed due to problem(s) in flash memory
0x02	_SENSOR_OPEN	Failure caused by optic module
0x03	_REGISTER_FAILED	Fingerprint registration failed
0x04	_VERIFY_FAILED	Fingerprint verification failed
0x05	_ALREADY_REGISTERED_USER	UserID already exists
0x06	_USER_NOT_FOUND	UserID is not found in the SDA04 database
0x08	_TIMEOUT	Failed to capture fingerprint in preset time
0x09	_DB_FULL	SDA04 database has insufficient space to enroll a new user
0x0A	_DB_WRONG_USERID	Failure in removing or verifying an unregistered user
0x0B	_DB_NO_DATA	Database has no data
0x0C	_EXTRACT_FAIL	Failed to capture feature points of fingerprint
0x10	_FUNCTION_FAIL	Function call failed
0x11	_INSUFFICIENT_DATA	Received data size doesn't match the size defined in ExtraData
0x12	_FLASH_WRITE_ERROR	Writing in flash memory failed
0x14	_INVALID_PARAM	Parameter of packet is invalid
0x15	_MASTERFP_NOT_FOUND	Fingerprint of master cannot be found (occurs when trying to proceed without master registration)
0x16	_MASTERCOUNT_EXCEED	Number of masters exceeds five (no more than five masters can be registered)
0x17	_AUTHENTICATION_FAIL	Verification of master failed
0x1B	_IDENTIFY_FAILED	No fingerprint in database matches fingerprint in input window
0x20	_INVALID_USERDATA_SIZE	While recording values from SDA04 to host, size of data exceeded the user portion
0x21	_INVALID_USERDATA_ADDRESS	User portion of data was exceeded while recording values from SDA04 to host
0x23	_AUTO_ON_MODE	The size of host user portion is not set
0x24	_NOT_AUTO_ON_MODE	CMD_FP_AUTO_IDENTIFY is now processing; no other operation can be performed.
0x28	_CHECKSUM_ERROR	Protocol checksum error
0x30	_INVALID_FPRECORD	Record format is invalid
0xFF	_UNKNOWN_COMMAND	Commands unknown

Appendix G. Interfacing SDA04 with SecuGen USB Peripherals

SecuGen's PC fingerprint peripherals include FDU02, FDU03, SDU03 and FDU04 based USB port devices. Regardless of which peripheral you have, the SDA04 SG400 format templates are compatible with all SDKs that support the SecuGen USB peripherals. ANSI378 templates generated by SDA04 can be matched with any ANSI INCITS 378-2004 compliant matching algorithm.

Note: Since SecuGen sensors have their own characteristics, fingerprint images captured on these devices may differ slightly from one sensor to another. This slight difference may increase recognition error rates when fingerprint matching is done between different products such as FDU02 ←→ SDA04, FDU03←→SDA04, FDU04←→SDA04 and SDU03←→SDA04.

Ordering Information

Product	Part-number	Description
SDA04	EA4-0101D	SDA04 (Full set includes optic module, board, and cable)
SDA04 PCB	EA1-0005G	SDA04 processing board only
OPP03AC2	EA2-0010D	OPP03AC2 optic module
FDA C1 Cable	EA3-0005A	9pin-9pin, 1.25mm, 9 wire cable SDA04 ←→OPP03AC2
FDA C3 Cable	EA3-0007A	15pin, 1.25mm, 3 wire to DB9 cable SDA04 ←→ PC
FDA C4 Cable	EA3-0022A	15pin, 1.25mm, 11 wire cable SDA04←→ host

SecuGen Sales Contact
Sales@secugen.com
Sales Tel: +1 (408) 727-7717

Purchase Orders
Purchase orders may be sent by email or by fax:
Orders@secugen.com
Fax: +1 (408) 608-6363