**ADH-Tech**

***Data Sheet***

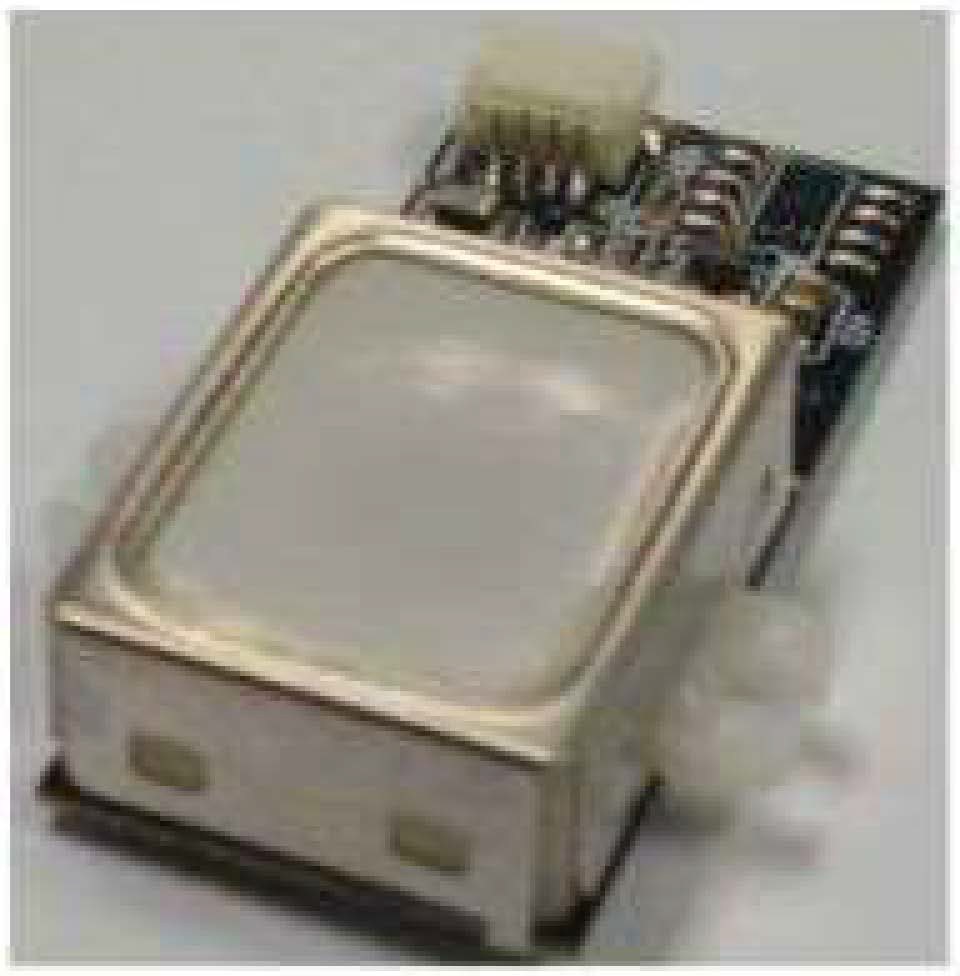
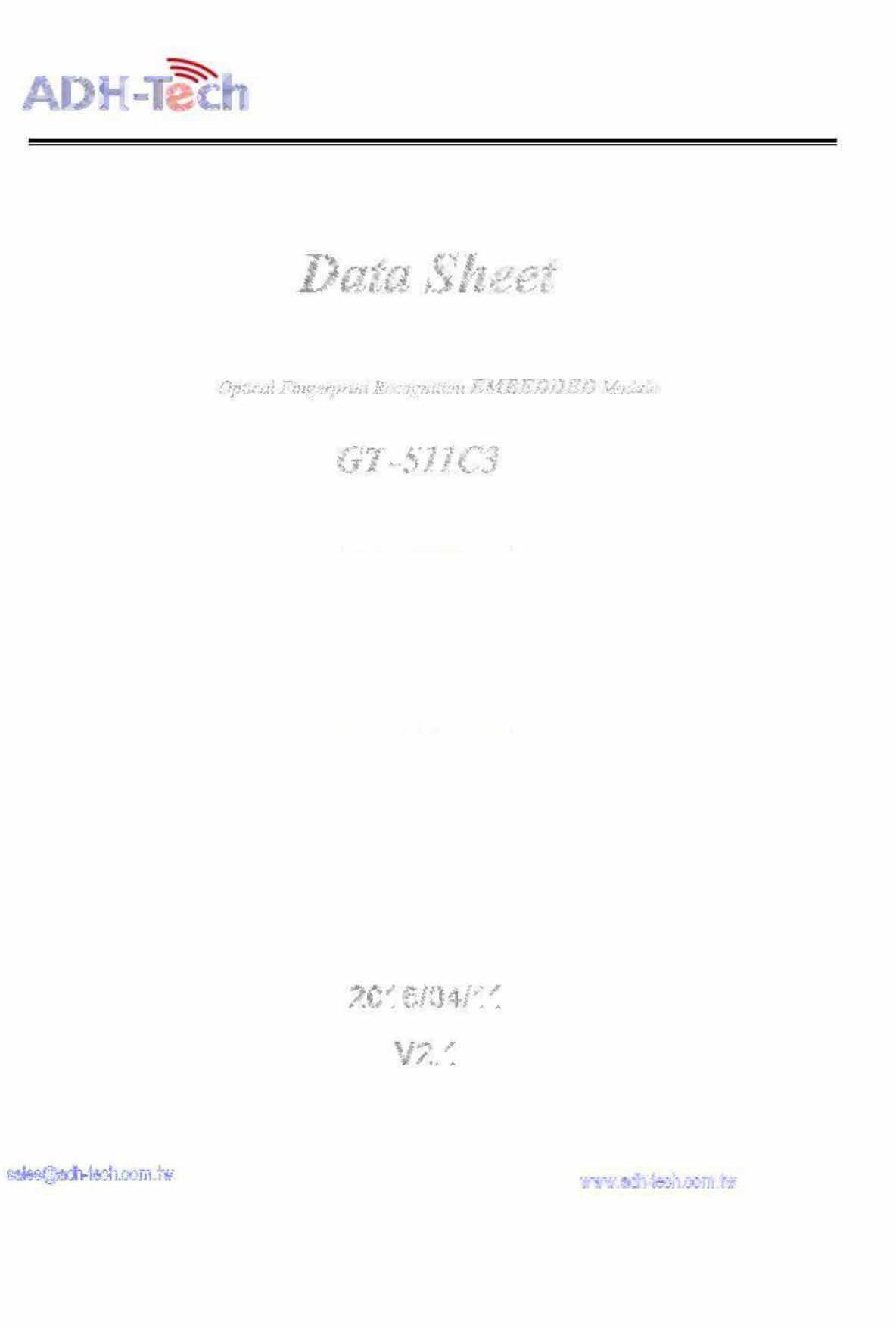
*Optical Fingerprint Recognition EMBEDDED 1\rfodule*

***G1-511C3***

**2016/04/11**

**V2.1**

[sales@adh-tec-h.oom.lw](mailto:sales@adh-tec-h.oom.lw)



[www.adh-tech.com.tw](http://www.adh-tech.com.tw/)

Contents

1. [Concept 4](#_TOC_250024)
2. [Protocol: Packet Structure 6](#_TOC_250023)

[Command Packet (Command) 6](#_TOC_250022)

[Response Packet (Acknowledge) 6](#_TOC_250021)

[Data Packet (Data) 7](#_TOC_250020)

1. Protocol: Commands Summary 8
2. [Protocol: Error Codes 10](#_TOC_250019)
3. [Protocol: Command Details 12](#_TOC_250018)
   1. [lnitialization(Open) 12](#_TOC_250017)
   2. [Termination(Close) 14](#_TOC_250016)
   3. Fast searching of the device(UsblnternalCheck) 14
   4. [CMOS LED control(Cmosled) 15](#_TOC_250015)

S.S. Changing UART baud rate (ChangeBaudrate) 16

* 1. Get enrolled fingerprint count(GetEnrollCount) 17
  2. Check enrollment status(CheckEnrolled) 17
  3. Start an enrollment(fnrollStart) 18
  4. Make 1st template for an enrollment(fnrolll) 18
  5. [Make 2nd template for an enrollment(fnroll2) 19](#_TOC_250014)
  6. [Make 3rd template for an enrollment, merge three templates(fnroll3) 19](#_TOC_250013)
  7. Check finger pressing status(/sPressFinger) 21
  8. [Delete one fingerprint(Delete/D) 22](#_TOC_250012)
  9. [Delete all fingerprints(DeleteAll) 22](#_TOC_250011)
  10. [1:1 Verification(Verify) 23](#_TOC_250010)
  11. [1:N ldentification(/dentify) 23](#_TOC_250009)
  12. 1:1 Verification of Template(VerifyTemplate) 24
  13. 1:N Identification of Template(ldentifyTemplate) 25
  14. [Capture *fingerprint(CaptureFinger)* 26](#_TOC_250008)
  15. [Make *Template(MakeTemplate)* 27](#_TOC_250007)
  16. Get fingerprint image(Get/mage) 28
  17. Get raw image(GetRawlmage) 29
  18. [Get template(GetTemplate) 29](#_TOC_250006)
  19. [Set *template(SetTemplate)* 30](#_TOC_250005)
  20. Start database download, *obsolete(GetDatabaseStart)* 30
  21. End database download, *obsolete(GetDatabaseEnd)* 31
  22. Upgrade *Firmware(UpgradeFirmware)* 32
  23. Upgrade ISO CD Image(UpgradeISOCDimage) 32
  24. [Set IAP Mode(SetIAPMode) 32](#_TOC_250004)
  25. Set Security Level(SetSecuritylevel) 33
  26. Get Security Level(GetSecuritylevel) 33

1. [Protocol: Flowchart, description 34](#_TOC_250003)
   1. [Capture of the fingerprint image 34](#_TOC_250002)
   2. Identifying and Verifying 34
   3. [Enrollment 34](#_TOC_250001)
2. [PC Demo 36](#_TOC_250000)
3. Mechanical Dimensions 39

## Concept

This device is one chip module with;

* fingerprint algorithm
* optical sensor

The major functions are the followings.

* High-accuracy and high-speed fingerprint identification technology
* Ultra-thin optical sensor
* 1:1 verifi cation, 1:N identificat ion
* downloading fingerprint image from the device
* Reading & writing fingerprint template(s) from/to the device
* Simple UART & USB communication protocol Technical Speci fication

|  |  |
| --- | --- |
| **Item** | **Value** |
| CPU | ARM Cortex M3 Core |
| Sensor | optical Sensor |
| Effective area of the Sensor | 14 x 12.5(mm) |
| Image Size | 202 x 258 Pixels |
| Resolution | 450 dpi |
| The maximum number of  fingerprints | 200 fingerprints |
| Matching Mode | 1:1, l:N |
| The size of template | 496 Bytes (template) + 2 Bytes (checksum) |
| Communication interface | UART, default baud rate = 9600bps after power on  USB Verl.1, Full speed |
| False Acceptance Rate (FAR) | < 0.001% |
| False Rejection Rate(FRR) | < 0.1% |
| Enrollment time | < 3 sec (3 fingerprints) |
| Identifi cation time | < 1.0 sec (200 fingerprints) |
| Operating voltage | DC 3.3-6V |
| Operating current | < 130mA |

|  |  |  |
| --- | --- | --- |
| Operating environment | Temperatur e | -20°( ~ +60°( |
| Humidity | 20% ~ 80% |
| Storage environment | Temperatur e | -20°( ~ +60°( |
| Humidity | 10% ~ 80% |

1. **Protocol: Packet Structure**

**(Multi-byte item is represented as Little Endian.)**

Command Packet (Command)



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **OFFSET** |  |  | **ITEM** | **E** |  |  | **DESCRIPTION** |  |
| 0 | | | 0xSS | | **BYTE** | | Command start codel | | |
| 1 | | | **0xAA** | | **BYTE** | | Command start code2 | | |
| 2 | | | *Device ID* | | **WORD** | | Device ID: default is 0x000l, always fixed | | |
| 4 | | | *Parameter* | | **DWORD** | | Input parameter | | |
| 8 | | | *Command* | | **WORD** | | Command code | | |
| **10** | | | *Check Sum* | | **WORD** | | Check Sum (byte addition)  OFFSET[0] + ... +OFFSET[9] = *Check Sum* | | |

Response Packet (Acknowledge)



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **OFFSET** |  |  | **ITEM** | **E** |  |  | **DESCRIPTION** |  |
| 0 | | | 0xSS | | BYTE | | Response start codel | | |
| 1 | | | 0xAA | | BYTE | | Response start code2 | | |
| 2 | | | *Device ID* | | WORD | | Device ID: default is 0x000l, always fixed | | |
| 4 | | | *Parameter* | | DWORD | | **Response** = = **Ox30:** (ACK) Output Parameter  **Response** = = **Ox31:** (NACK) Error code | | |
| 8 | | | *Response* | | WORD | | **Ox30:** Acknowledge (ACK).  **Ox31:** Non-acknowledge (NACK). | | |
| 10 | | | *Check Sum* | | WORD | | Check Sum (byte addition)  OFFSET[0] + ...+OFFSET[9] = *Check Sum* | | |

**Data Packet (Data)**



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **OFFSET** |  |  | **ITEM** | **E** |  |  | **DESCRIPTION** |  |
| 0 | | | 0xSA | | BYTE | | Data start codel | | |
| 1 | | | 0xAS | | BYTE | | Data start code2 | | |
| 2 | | | *Device ID* | | WORD | | Device ID: default is 0x000l, always fixed | | |
| 4 | | | *Data* | | N BYTES | | N bytes Data  The size is pre-defined per protocol stage | | |
| 4+N | | | *Check Sum* | | WORD | | Check Sum (byte addition)  OFFSET[0] +...+OFFSET[4+N-1]*=Check Sum* | | |

1. **Protocol: Commands**

**Summary**

In a command packet *Command* can be one of below.

|  |  |  |
| --- | --- | --- |
| **Number**  **(HEX)** | **Alias** | **Description** |
| 01 | *Open* | Initialization |
| 02 | *Close* | Termination |
| 03 | *UsblnternalCheck* | Check if the connected USB device is valid |
| 04 | *ChangeBaudrate* | Change UART baud rate |
| 05 | *SetlAP/v1ode* | Enter IAP Mode  In this mode, FW Upgrade is available |
| 12 | *CmosLed* | Control CMOS LED |
| 20 | *GetEnrollCount* | Get enrolled fingerprint count |
| 21 | *Check Enrolled* | Check whether the specified ID is already enrolled |
| 22 | *EnrollStart* | Start an enrollment |
| 23 | *Enrolll* | Make 1st template for an enrollment |
| 24 | *Enroll2* | Make 2nd template for an enrollment |
| 25 | *Enroll3* | Make 3rd template for an enrollment, merge three templates into one template, save merged template to the database |
| 26 | *lsPressFinger* | Check if a finger is placed on the sensor |
| 40 | *DeletelD* | Delete the fingerprint with the specified ID |
| 41 | *DeleteAll* | Delete all fingerprints from the database |
| 50 | *Verify* | 1:1 Verification of the capture fingerprint image with the specified ID |
| 51 | *Identify* | 1:N Identification of the capture fingerprint  image with the database |
| 52 | *VerifyTemplate* | 1:1 Verification of a fingerprint template with the specified ID |
| 53 | *IdentifyTemplate* | 1:N Identification of a fingerprint template with the data base |

|  |  |  |
| --- | --- | --- |
| **Number**  **(HEX)** | **Alias** | **Description** |
| 60 | *CaptureFinger* | Capture a fingerprint image(256x256) from  the sensor |

I 61 I *MakeTemplate* I Make template for transmission

|  |  |  |
| --- | --- | --- |
| 62 | *Getlmage* | Download the captured fingerprint image(256x256) |
| 63 | *GetRawlmage* | Capture & Download raw fingerprint  image(320x240) |
| 70 | *GetTemplate* | Download the template of the specified ID |
| 71 | *SetTemplate* | Upload the template of the specified ID |
| 72 | *GetDatabaseStart* | Start database download, **obsolete** |
| 73 | *GetDatabaseEnd* | End database download, **obsolete** |
| 80 | *UpgradeFirmware* | Not supported |
| 81 | *Upgrade!SOCD!mage* | Not supported |
| FO | *SetSecurityLevel* | Set Security Level |
| F1 | *GetSecurityLevel* | Get Security Level |
| 30 | *Ack* | Ackn owledge. |
| 31 | *Nack* | Non-acknowledge. |

### Protocol: Error Codes

When response packet is Non-acknowledge, *Parameter* represents an error code as below.

|  |  |  |
| --- | --- | --- |
| **NACK Parameter** | **Value** | **Description** |
| NACK\_TIMEOUT | 0xl00l | **Obsolete,** capture timeout |
| NACKJNVALID \_BAUDRATE | 0x1002 | **Obsolete,** Invalid serial baud rate |
| NACKJNVALID\_POS | 0x1003 | The specified ID is not between  0~199 |
| NACK\_IS\_NOT\_USED | 0x1004 | The specified ID is not used |
| NACKJS\_ALREADY\_USED | 0xl00S | The specified ID is already used |
| NACK\_COMM\_ERR | 0x1006 | Communication Error |
| NACK\_ VERIFY\_FAILED | 0x1007 | 1:1 Verification Failure |
| NACKJDENTIFY \_FAILED | 0x1008 | 1:N Identification Failure |
| NACK\_DBJS\_FULL | 0x1009 | The database is full |
| NACK\_DBJS\_EM PTY | 0xl00A | The database is empty |
| NACK\_TURN\_ERR | 0xl00B | **Obsolete,** Invalid order of the enrollment  (The order was not as: **EnrollStart** - >  **Enrolll** - > **Enroll2** - > **Enroll3)** |
| NACK\_BAD\_FINGER | 0xl00C | Too bad fingerprint |
| NACK\_EN ROLL\_FAILED | 0xl00D | Enrollment Failure |
| NACKJS\_NOT \_SUPPORTED | 0xl00E | The specified command is not supported |
| NACK\_DEV \_ERR | 0xl00F | Device Error, especially if  Crypto-Chip is trouble |
| NACK\_CAPTURE\_CANCELED | 0x1010 | **Obsolete,** The capturing is canceled |
| NACKJNVALID\_PARAM | 0xl0ll | Invalid parameter |
| NACK\_FINGERJS\_NOT\_PRESSED | 0x1012 | Finger is not pressed |

**ADH-Tech**.·

|  |  |  |
| --- | --- | --- |
| Duplicated ID | 0-199 | There is duplicated fingerprint (while enrollment or setting template), This error describes just duplicated ID |



### Protocol: Command Details

#### lnitialization(Open)

Host Device

**COMMAND PACKET**

Command = *Open*

Parameter=

0: not to get extra info Nonzero: to get extra info

**RESPONSE PACKET**

Response = *Ack*

If host requested to get extra

**DATA PACKET**

Data=

typedefstruct\_devinfo

{

DWORDFirmwareVersion; DWORD IsoAreaMaxSize; BYTEDevice SerialNumber[ l ];

} devinfo;

*Open* command is used to initialize the device; especially it gets device's static info.

**Description of *devinfo* structure**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Field** |  |  | **Sample** |  |  | **Description** |  |
| *FirmwareVersion* | | |  | | | Firmware version | | |
| *lsoAreaMaxSize* | | |  | | | Maximum size of ISO CD image | | |
| *OeviceSerialNumber* | | |  | | | Unique serial number  of the device | | |

If the Device's Serial Number is zero, then there is no guarantee for stable operation of the device.

#### Termination(Close)

Host Device

**RESPONSE PACKET**

Response = *Ack*

**COMMAND PACKET**

Command = *Close*

*Close* command does nothing.

* 1. **Fast searching of the *device(UsblnternalCheck)***

Host Device

**RESPONSE PACKET**

Response = *Ack*

Parameter = OxSS

**COMMAND PACKET**

Command = *UsblnternalCheck*

The device operates as removable CD drive. If another removable CD drive exists in the system, connection time maybe will be long. To prevent this, *UsblnternalCheck* command is used for fast searching of the device.

* 1. CMOS LED control(Cmosled)

Host Device

**RESPONSE PACKET**

Response = *Ack*

**COMMAND PACKET**

Command = *CmosLed*

Parameter= 0: Off LED

Nonzero: On LED

Default state of CMOS (Sensor) LED is OFF state.

(But while booting, LED blinks once, this says the LED is OK.) Therefore, please issue LED ON command prior to any capture.

**S.S. Changing UART baud rate *(ChangeBaudrate)***

Host Device

**RESPONSE PACKET**

Response = *Ack:* Response = *Nack:* Error *NACK\_INVALID\_PARAM*

**COMMAND PACKET**

Command = *ChangeBaudrate* Parameter= *baud rate (9600~115200)*

This command changes the UART baud rate at the run-time.

The device initializes its UART baud rate to 9600 bps after power on.

* 1. **Get enrolled fingerprint *count(GetEnrollCount)***

**RESPONSE PACKET**

Response = *Ack*

Parameter = Enrolled Fingerprint Count

**COMMAND PACKET**

Command = *GetEnrollCount*

Host Device

* 1. **Check enrollment *status(CheckEnrolled)***

Host Device

**RESPONSE PACKET**

Response = *Ack:* This ID is enrolled Response = *Nack:* Error *NACK\_INVALID\_POS NACK\_JS\_NOT\_ USED*

**COMMAND PACKET**

Command = *CheckEnrolled*

Parameter= */0(0~199)*



* 1. **Start an *enrollment(EnrollStart)***

Host Device

**RESPONSE PACKET**

Response = *Ack:* OK Response = *Nack:* Error *NACK\_DB\_IS\_FULL NACK\_INVALID\_POS*

*NACK\_IS\_ALREADY\_USED*

**COMMAND PACKET**

Command = *EnrollStart*

Parameter= *10(0~199)*

*(If Parameter's HIWORD is non-zero, fingerprint duplication check will not be) If JO* = = *-1, then "Enrollment without saving" will be stated.*

* 1. **Make 1st template for an *enrollment(Enrolll)***

Host Device

**RESPONSE PACKET**

Response = *Ack:* OK Response = *Nack:* Error *NACK\_ENROLL\_FAILED NACK\_BAD\_FINGER*

*Duplicated 10(0~199)*

**COMMAND PACKET**

Command = *Enrolll*

* 1. Make 2nd template for an enrollment(fnroll2)

Host Device

**RESPONSE PACKET**

Response = *Ack:* OK Response = *Nack:* Error *NACK\_ENROLL\_FAILED NACK\_BAD\_FINGER*

*Duplicated 10(0~ 199)*

**COMMAND PACKET**

Command = *Enroll2*

* 1. Make 3rd template for an enrollment, merge three templates(fnroll3)

Host

Device

**COMMAND PACKET**

Command = *Enroll3*

**RESPONSE PACKET**

Response = *Ack:* OK Response = *Nack:* Error *NACK\_ENROLL\_FAILED NACK\_BAD\_FINGER*

*Duplicated /0(0~199)*

If ***EnrollStart*** was called with */0* = = *-1,* then the enrolled template will not be saved, it will be transferred to the host, as below.

**DATA PACKET**

Data = *enrolled template (498bytes)*

To enroll a fingerprint, the host must issue above 4 commands, later chapter describes how to organize these co mman ds.

**ADH-Tech**.·

* 1. **Check finger pressing *status(lsPressFinger)***

Host Device

.

-

**RESPONSE PACKET**

Response = *Ack:*

Parameter = 0: finger is pressed

Parameter = nonzero : finger is not pressed

**COMMAND PACKET**

Command = *JsPressFinqer*

This command is used while enrollment, the host waits to take off the finger per enrollment stag e.

#### Delete one fingerprint(Delete/D)

Host Device

**RESPONSE PACKET**

Response = *Ack:* OK Response = *Nack:* Error *NACK\_INVALID\_POS*

**COMMAND PACKET**

Command = *Oelete/0*

Parameter= *10(0~199)*

#### Delete all fingerprints(DeleteAll)

Host Device

**RESPONSE PACKET**

Response = *Ack:* OK Response = *Nack:* Error *NACK\_DB\_IS\_EMPTY*

**COMMAND PACKET**

Command = *OeleteAll*

#### 1:1 Verification(Verify)

Host Device

**RESPONSE PACKET**

Response = *Ack:* OK Response = *Nack:* Error *NACK\_INVALID\_POS NACK\_JS\_NOT\_ USED NACK\_ VERIFY\_FAILED*

**COMMAND PACKET**

Command = *Verify*

Parameter= *10(0~199)*

#### 1:N ldentification(/dentify)

Host Device

**RESPONSE PACKET**

Response = *Ack:* OK

Parameter = O~199 : identified ID

Response = *Nack:* Error *NACK\_DB\_IS\_EMPTY NACK\_/DENTIFY\_FAILED*

**COMMAND PACKET**

Command = *Identify*

* 1. **1:1 Verification of Tern *plate(VerifyTemplate)***

Host Device

**RESPONSE PACKET**

Response = *Ack:* OK Response = *Nack:* Error *NACK\_ COMM\_ERR NACK\_ VERIFY\_FAILED*

**DATA PACKET**

Data = *a template (498 bytes)*

**RESPONSE PACKET**

Response = *Ack:* OK Response = *Nack:* Error *NACK\_INVALID\_POS NACK\_JS\_NOT\_ USED*

**COMMAND PACKET**

Command = *Verify*

Parameter= *10(0~199)*

* 1. **1:N Identification of *Template(ldentifyTemplate)***

Host Device

**RESPONSE PACKET**

Response = *Ack:* OK

Parameter = O~199: identified ID

Response = *Nack:* Error

*NACK\_ COMM\_ERR NACK\_IOENTIFY\_FAILED*

**DATA PACKET**

Data = *a template (498 bytes)*

**RESPONSE PACKET**

Response = *Ack:* OK Response = *Nack:* Error

*NACK\_OB\_IS\_EMPTY*

**COMMAND PACKET**

Command = *JdentifyTemplate*

#### Capture fingerprint(CaptureFinger)

Host Device

**RESPONSE PACKET**

Response = *Ack:* OK Response = *Nack:* Error

*NACK\_FINGER\_IS\_NOT\_PRESSED*

**COMMAND PACKET**

Command = *CaptureFinger*

Parameter=

0: not best image, but fast Nonzero: best image, but slow

The fingerprint algorithm uses 450dpi 256x256 image for its input.

This command captures raw image from the sensor and converts it to 256x256 image for the fingerprint algo rit hm. If the finger is not pressed, this command returns with non-acknowledge.

Please use best image for enrollment to get best enrollment data. Please use not best image for identification (verification) to get fast user sensibilit y.

* 1. Make Template(MakeTemplate)

Host Device

**DATA PACKET**

Data = *made template (498 bytes)*

**RESPONSE PACKET**

Response = *Ack:* OK Response = *Nack:* Error

*NACK\_BAD\_FINGER*

**COMMAND PACKET**

Command = *MakeTemplate*

This function makes template for transmission. *CaptureFinger* command should be previously issued . Do not use the template for regist ration .

# A D H-T h

* 1. **Get fingerprint *image(Getlmage)***

Host Device

.

**DATA PACKET**

Data = 258x202 image (52116 bytes)

**RESPONSE PACKET**

Response = *Ack:* OK

**COMMAND PACKET**

Command = *Getlmage*

* 1. **Get raw *image(GetRawlmage)***

Host

Device

**DATA PACKET**

Data = 160x120 QVGA image (19200 bytes)

**RESPONSE PACKET**

Response = *Ack:* OK

Capture raw image

**COMMAND PACKET**

Command= *GetRawlmage*

##### Get template(GetTemplate)

Host Device

**DATA PACKET**

Data = *a template (498 bytes)*

**RESPONSE PACKET**

Response = *Ack:* OK Response = *Nack:* Error *NACK\_INVALID\_POS NACK\_IS\_NOT\_ USED*

**COMMAND PACKET**

Command = *GetTemplate*

Parameter= *lO(0~199)*

#### Set template(SetTemplate)

**RESPONSE PACKET**

Response = *Ack:* OK Response = *Nack:* Error *NACK\_ CO/vlfvl\_ERR NACK\_DEV\_ERR*

*Duplicated 10(0~199)*

**DATA PACKET**

Data = *a template (498 bytes)*

**RESPONSE PACKET**

Response = *Ack:* OK Response = *Nack:* Error *NACK\_INVALID\_POS*

**COMMAND PACKET**

Command = *SetTemplate*

Parameter= *10(0~199)*

*(If Parameter's HIWORD is*

*non -zero, fingerprint duplication check will not be)*

Host Device

* 1. **Start database download, *obsolete(GetDatabaseStart)***

**RESPONSE PACKET**

Response = *Ack*

**COMMAND PACKET**

Command = *GetDatabaseStart*

Host Device

*GetDatabaseStart* command does nothing. It exists for historical reaso n; it was used for RS232 communication.

# ADH-T h

* 1. **End database download, *obsolete(GetDatabaseEnd)***

Host Device

-

**RESPONSE PACKET**

Response = *Ack*

**COMMAND PACKET**

Command = *GetDatabaseEnd*

*GetDatabaseEnd* command does nothing. It exists for historical reason; it was used for RS232 communication.

* 1. **Upgrade *Firmware(UpgradeFirmware)***

Not supported

###### Upgrade ISO CD lmage(UpgradeISOCDimage)

Not supported

###### Set IAP Mode(SetIAPMode)

Host Device

**RESPONSE PACKET**

Response = *Ack*

**COMMAND PACKET**

Command= *SetlAPMode*

The Device enter in IAP Mode,

In this mode, FW upgrade is available .

* 1. **Set Securitylevel(0xF0)**

**RESPONSE PACKET**

Response = *Ack*

**COMMAND PACKET**

Command= *SetSecurityLevel(OxFO)* Parameter= 1~ 5

Parameter is equal 5. The level is the highest security level Default value is 3.

Host Device

##### GetSecuritylevel(0xfl)

**RESPONSE PACKET**

Response = *Ack*

Parameter =Security Level

**COMMAND PACKET**

Command=

*GetSecurityLevel(OxFl)*

Host Device

1. **Protocol: Flowchart, description**

###### Capture of the fingerprint image

*/sPressFinger* checks whether a finger placed on the sensor. This function is used especially while enrollment.

*CaptureFinger* captures a fingerprint image (256x256), if a finger isn't placed on the sensor, it returns with error.

If this function returns with success, the device's internal RAM keeps valid

fingerprint image for the subsequent commands. If the host issues other command, the fingerprint image will be used and destroyed.

*GetRawlmage* captures a raw live image (320x240), it doesn't check whether a finger placed on the sensor, this function is used for debug or calibrat ion .

###### ldentifying and Verifying

*Identify* and *ldentifyTemplate* perform 1: N matching operation.

*Verify* and *VerifyTemplate* perform 1: 1 matching operation.

Just before calling of image-related matching functions *(Identify, Verify),* the host must call *CaptureFinger.*

###### Enrollment

An enrollment flowchart is as below.

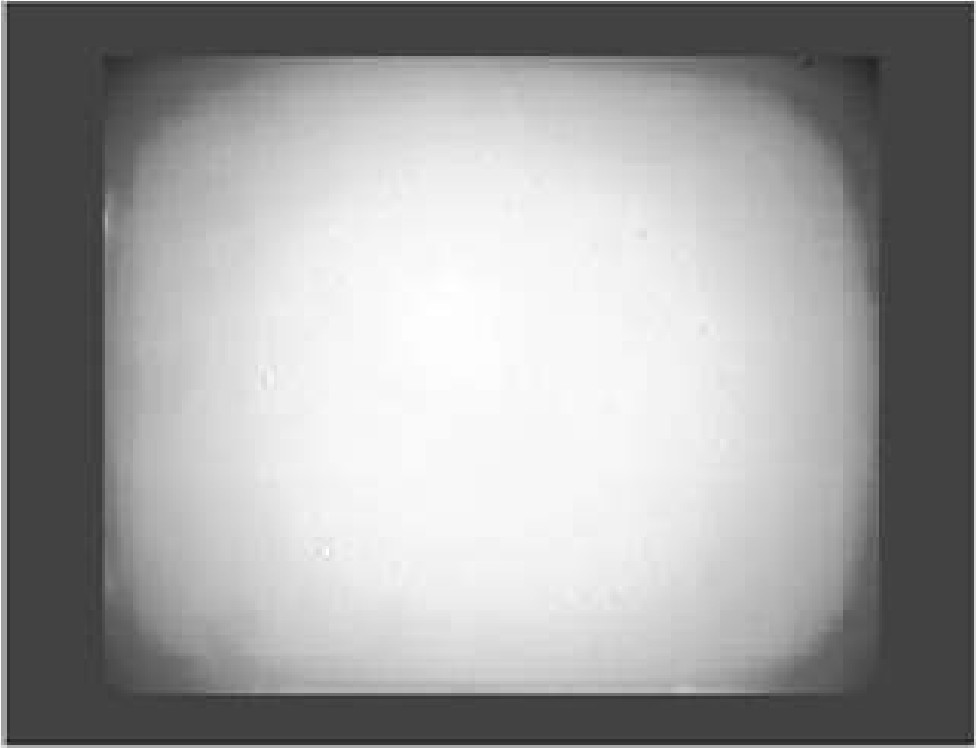
1. *EnrollStart* with a (not used) ID
2. *CaptureFinger*
3. *Enrolll*
4. Wait to take off the finger using */sPressFinger*

**ADH-Te ch**

1. *CaptureFinger*
2. *Enroll2*
3. Wait to take off the finger using *lsPressFinger*
4. *CaptureFinger*
5. *Enroll3*

# ADH-T h

**7. PC Demo**

PC demo program describes how to use the device with its source code.



I **SDKJ ) EMO**

erial Port Numb er u;=[::::=s=B===== ..:.:.'.:l

Bau d rate:: \_[ 1\_15J®O ]

Open

TD: 1 1

Enrnll

l

l

j

]

[

Olose

Save Image To Rfe

[

[

[

[

[

[

[

[

Get User Gourrt.

Verify(1:1)

Delete ID

ldentify-(1:N)

D l E;J!ill

Verify Template

]

Ge- t Template

)dentify Te plate, l

Set Te plate,

l l l l l

Is Press, Flr;iger

]

l

G et Database

J

fa:tlmag

Set D,atabase

l

'Get Raw Jmage

l

Firmware Upgrade l

Cancel

l [

]

ISO Upgrade

Result : Firmware Version: 20120502 JsoAreaMaxSize:- **500** KB

DevTceSN; 1485E:2B4&A5D1010-0DODl:lOOOIJO

l

Get Live Im.age

|  |  |  |  |
| --- | --- | --- | --- |
| **Command Alias** | **UI item to test it** | | |
| *Open, UsblnternalCheck, ChangeBaudrate* | [ Open l | | |
| *Close* |  | | l |
| [ | O ose |
|  |
| *GetEnrollCount* |  | | l |
| [ | Get Us r Coui;rt |
|  |



( )

Gingy Technology Inc.

1 3 C6

Tel: (03) 563-2500, Fax:(03) 563-2508

[http://www.gingytech.com.tw](http://www.gingytech.com.tw/)



|  |  |
| --- | --- |
| **Command Alias I** | **UI item to test it** |
| *CheckEnrolled, EnrollStart, Enrolll, Enroll2, Enroll3, lsPressFinger* | , |
| *Oelete/0* |  |
| *DeleteAll* |  |
| *Verify* |  |
| *Identify* |  |
| *VerifyTemplate* |  |
| *IdentifyTemplate* |  |
| *CaptureFinger, Getlmage* |  |
| *GetRawlmage* |  |
| *GetTemplate, GetDatabaseStart, GetDatabaseEnd* | , |
| *SetTemplate* |  |
| *UpgradeFirmware* |  |
| *Upgrade!SOCD!mage* |  |

Demo program is supported with its source code.

**ADH-Tech**.·

The project is Microsoft Visual C+ + 6.0 project.

We selected VC6.0 to minimize the size of the executable.

The demo program checks whether it is running on removable CD drive, if it is the case, it copies itself to *"My Document"* folder and executes copied version. This is for direct access to the device's removable CD drive.