

DIGENT Fingerprint Module

Function Reference Guide

FC10



Introduction

Thank you for your purchase of our products.

DIGENT Fingerprint Identification OEM Module (bellow simply notes as OEM) is developed for the user to use fingerprint identification engine.

OEM has many functions such as image capture and feature extract functions. You need not consider about fingerprint identification engine and you are able to use these functions only by using OEM. Your HOST machine (computers and other devices) can interface with OEM throughout the RS-232 serial communication.

Using OEM module you can develop your own:

- Fingerprint based Access Control Door Lock
- Fingerprint based Time Attendance machine
- Fingerprint + Smart Card/Proximity/ Barcode/ Swipe card based systems
- Fingerprint based weapon control system
- Fingerprint based machine control
- Fingerprint based car locks
- More

Feature of embedded fingerprint recognition algorithm

DIGENT Fingerprint Recognition Algorithm follows the commonly accepted fingerprint identification scheme, which uses a set of specific fingerprint feature points (minutiae).

However, it contains many powerful algorithmic solutions, which enhance the system performance and reliability. Some of them are listed below:

Image Quality Check Algorithm: This algorithm provides feedback on the quality of the query fingerprint image. It produces error codes such as no fingerprint present; or fingerprint too dry or too wet.

Efficient Image Enhancement Algorithm: This algorithm allows eliminating noises, ridge ruptures and stuck ridges, and extract feature points more reliably even from extremely poor fingerprint images, with processing time less than 0.5 seconds.

Fully Tolerant to Fingerprint Distortion and Rotation Matching Algorithm: In usually, there are many distortions, translation, and rotation in query fingerprint image is caused by various reasons. DIGENT Fingerprint Recognition Algorithm is fully to them.

DIGENT Fingerprint Engine uses a powerful fingerprint matching algorithm instead of other previous methods, which can match 1000 fingerprints per seconds and identify two fingerprints even if they are arbitrary rotated within 360°, translated and

have a few same part of image. And then, this algorithm is not dependent on presence of the fingerprint core or delta points in the image.

Fingerprint Enroll Mode with Feature Collection: DIGENT Fingerprint Recognition Engine has the fingerprint enrollment with feature collection mode. This mode generates the collection of the features from 3 fingerprints of the same finger. Each fingerprint image is processed and its feature is extracted. Then these 3 features are analyzed and collected into a single feature collection, which is stored to the database. This way, the enrolled feature is more reliable and system performance considerably increases using this mode.

Database Pre-Sorting by Global Feature Vector: DIGENT Fingerprint Recognition Engine pre-sorts the database entries using the certain global fingerprint feature vector. Fingerprint matching is performed first with the database entries having global feature most similar to those of the query fingerprint image, then the next entries with most similar is matched, and so on, until the matching result is successful or the end of the database is reached. In most cases there is fairly good chance that the correct match will be found already in the beginning of the search. As a result, the number of matching required to achieve fingerprint identification decreases drastically, and correspondingly, the effective matching speed increases.

Suitable Algorithm to 1:1 and 1:N Mode: DIGENT Fingerprint Recognition Engine supports the both 1:1 and 1:N matching mode.

Specification

Items	Value
Response Time	< 1.5 sec
False Acceptance Rate (FAR)	< 0.0001 %
False Rejection Rate (FRR)	< 0.1 %
Matching Mode	1:N mode
Times of Enrollment	3 times
Number of User	1000
Template Size	480 Byte
Interface	RS-232 one channel, RS-485 one channel Baud Rate - 2400, 4800, 9600, 19200,38400, 57600,115200 Stop Bit – 1,1.5,2Bit Parity – none, even, odd, mask, space
Power	DC 5V Operation Current – 200mA
Operating Environment	Temperature 0°C ~ +60°C Humidity 15% ~ 80 %

1. Usage and Command List

The fingerprint identification procedure using OEM is as follow.

Feature Extract Procedure of Fingerprint

- Capture and check image

First, reads fingerprint image using Fingerprint Reader and saves image data to image region.

Next, checks image quality of data in image region.

If user doesn't press fingerprint on Fingerprint Reader, or if user presses by mistake, OEM reports error to HOST.

- Feature Extract

Extracts inherent feature to user from image data. When you receive extracted feature from OEM, you must save them for matching

User Enrollment and Match procedure

- Enrollment

The user enrollment procedure using OEM is as follow.

First, Inputs fingerprint and extracts feature three times per one finger.

Next, Analyzes three features and collects into a single feature collection.

Next, Matches extracted feature with already enrolled features. If matching feature exists, OEM requires to input fingerprint again.

- Matching

Matches extracted feature with enrolled feature in HOST or OEM.

Transmit/Receive Data Structure

Host interfaces with OEM using RS-232 serial communication.

If you are going to write HOST program, you must know about how HOST sends command to OEM and what behavior OEM does by responding to it.

The structure of communication protocol is as follow.

Structure:

Start Code (7Eh)	Transmit Address	Receive Address	Command Code	Data Sequence	Checksum
---------------------	---------------------	--------------------	-----------------	------------------	----------

Start Code (1byte): Represents the start of Transmit/Receive data sequence.

Transmit Address (2 Bytes): Represents transmit address. (HOST - 0, OEM >=1)

Receive Address (2 Bytes): Represents receive address. (HOST - 0, OEM >= 1)

Command Code (2 Bytes): Represents command's type.

Data Sequence: Represents data sequence according to command code.

Checksum (2 Bytes): Represents the checksum of Transmit/Receive data sequence.

Remark:

Start code is fixed as 7Eh.

When HOST sends command to OEM, OEM receives it, performs action according to this and sends command to HOST replying this command.

HOST should send command to OEM, wait until reply command, check received command if it is acknowledge command and perform next action.

Checksum is the data to confirm the correctness of received data; it represents the sum of all bytes except start code.

Command List

Command List that HOST Sends

No	Command Name	Code	Function
1	ReqSetCommState	00H	Instructs OEM to set communication state
2	ReqEraseRom	02H	Instructs OEM to erase all flash ROM data.
3	ReqImageCapture	10H	Instructs OEM to capture Image.
4	ReqExtractFeature	11H	Instructs OEM to extract feature data.
5	ReqMatchFinger	12H	Instructs OEM to perform feature matching
6	ReqGetSortKey	13H	Instructs OEM to get sort key from feature data.
7	ReqGetUserNum	20H	Instructs OEM to send enrolled user count.
8	ReqAddUser	21H	Instructs OEM to add user data
9	ReqUpdateUser	22H	Instructs OEM to update user data
10	ReqDeleteUser	23H	Instructs OEM to delete user data.
11	ReqGetUser	24H	Instructs OEM to send user data.
12	ReqGetAllUser	25H	Instructs OEM to send all user data.
13	ReqFindUser	26H	Instructs OEM to find user data.
14	ReqGetPassword	50H	Instructs OEM to send password

Command List that OEM Sends

No	Command Name	Code	Function
1	AckSetCommState	00H	Notifies that OEM has changed communication state
2	AckEraseRom	02H	Notifies that OEM has erased all its flash ROM data.
3	AckImageCapture	10H	Sends image quality check result to HOST.
4	AckExtractFeature	11H	Sends feature data to HOST.
5	AckMatchFinger	12H	Sends matching result to HOST.
6	AckGetSortKey	13H	Sends sort key to HOST.
7	AckGetUserNum	20H	Sends user count enrolled in HOST to HOST.
8	AckAddUser	21H	Send added user ID to HOST.
9	AckUpdateUser	22H	Notifies that OEM has updated user data to HOST.
10	AckDeleteUser	23H	Notifies that OEM has deleted user data to HOST.
11	AckGetUser	24H	Sends user data to HOST.
12	AckGetAllUser	25H	Sends all user data to HOST.
13	AckFindUser	26H	Sends matched user ID to HOST.
14	AckGetPassword	50H	Send password to HOST

2. Commands

2-1. Configuration Set Commands

ReqSetCommState

Code: 00h
Data Sequence: Communication State (2 Bytes)
Direction: HOST->OEM
Remark:

HOST sends this command to change communication such as baud rate, parity, stop bits. The data structure of communication state that HOST sends is as follows.

MSB		LSB	
Don, t Care (7bit)	Stop Bits (2bit)	Parity Bits (3bit)	Baud Rate (4 bit)

The meanings of bit fields are as follow.

Stop Bits		Parity		Baud Rate	
Bit Value	Stop Bits	Bit Value	Parity	Bit Value	Baud Rate (BPS)
00	1 Bit	000	None	0000	2400
		001	Odd	0001	4800
01	1.5 Bits	010	Even	0010	9600
		011	Mask	0011	19200
10	2 Bits	100	Space	0100	38400
				0101	57600
				0110	115200

The default value is 9600 bps (baud rate), None (Parity), 1 bit (Stop bits).

AckSetCommState

Code: 00h
Data Sequence: None
Direction: OEM->HOST
Remark:

OEM sends this command to notice that it changed communication state.

ReqEraseRom

Code: 02h
Data Sequence: None
Direction: HOST->OEM
Remark:

HOST sends this command to erase all flash ROM of OEM.

AckEraseRom

Code: 02h
Data Sequence: None
Direction: OEM->HOST
Remark:

OEM sends this command to HOST that it has erased all flash ROM data.

2-2.Fingerprint Recognition Commands

ReqImageCapture

Code: 10h
Data Sequence: None
Direction: HOST->OEM
Remark:

HOST sends this command to OEM to input fingerprint from capture sensor and check image quality. When OEM receives this command, it tries to capture image data, saves to image data region, checks image quality and sends result to HOST by AckImageCapture command.

AckImageCapture

Code: 10h
Data Sequence: *Image Quality Result* (1Byte)
Direction: OEM->HOST
Remark:

OEM Sends this command to HOST to send image quality check result. The meaning of *Image Quality Result* is as follow.

0 – Normal
1 – No Fingerprint
FF – Image Capture Error

ReqExtractFeature

Code: 11h
Data Sequence: *Extract Index* (1Byte) + *Extract Flag* (1 Byte)
Direction: HOST->OEM
Remark:

HOST sends this command to extract feature from image data.

Extract Index represents the extract stage.

If value is 0, it represents extraction for first fingerprint enrollment.

If value is 1, it represents extraction for second fingerprint enrollment.

If value is 2, it represents extraction for third fingerprint enrollment.

If value is 3, it represents extraction for fingerprint matching.

Extract Flag represents whether to receive (1) feature data or not (0).

AckFeatureExtract

Code: 11h
Data Sequence: Result (1Byte) [+ *Feature Data*]
Direction: OEM->HOST
Remark:

OEM sends this command to HOST to send extract result.

The size of *Feature Data* is 480 bytes.

When Result is 1 and extract index is greater than 2 and extract flag is 1, OEM sends feature data to HOST.

ReqMatchFinger

Code: 12h
Data Sequence: *Feature Data*
Direction: HOST->OEM
Remark:

HOST sends this command to perform matching feature data it sends with the feature data OEM extracted.

AckMatchFinger

Code: 12h
Data Sequence: *Matching Result (1 Byte)*
Direction: OEM->HOST
Remark:

If result is 1, it means that matching is OK and, if result is 0 it means that matching is fail.

ReqGetSortKey

Code: 13h
Data Sequence: *Feature Data*
Direction: HOST->OEM
Remark:

HOST sends this command to get sort key from the feature data it sends and the feature data that OEM extracts.

AckGetSortKey

Code: 13h
Data Sequence: *Sort Key (2 Byte).*
Direction: OEM->HOST
Remark:

OEM sends this command to HOST to send sort key.
The sort key is used to reduce matching time.

2-3. User Management Commands

OEM has data flash ROM, so it can save user data. The user data is consist of header data and feature data. When the user data is saved to OEM, it is not necessary for HOST to have additional data ROM, and there is no need to send all user data to OEM for matching.

ReqGetUserNum

Code: 20h
Data Sequence: *None*
Direction: HOST->OEM
Remark:

HOST sends this command to get the number of user data saved in OEM.

AckGetUserNum

Code: 20h
Data Sequence: *Number of User Data(2 Byte).*
Direction: OEM->HOST
Remark:

OEM sends this command to HOST to send *Number of User Data*.

ReqAddUser

Code: 21h
Data Sequence: *Header Data (32 Byte) + Flag (1Byte)[+ Feature Data]*
Direction: HOST->OEM
Remark:

HOST sends this command to add user data.

When HOST sends this command to OEM, OEM saves *header data* and *feature data*.

If the **flag** value is 1, it means that the HOST sends *feature data* and if the *flag* value is 0, it means that the HOST doesn't send *feature data*. When the *flag* value is 0, OEM saves user using feature data area in OEM. So before using this command, the feature data must be extracted.

AckAddUser

Code: 21h
Data Sequence: *Result + User ID (2 Bytes)*
Direction: OEM->HOST
Remark:

OEM sends this command to HOST to send *User ID* enrolled.

When OEM successes in user enrollment, the result is 1 and OEM sends *User ID* and when OEM fails in user enrollment, the result is 0 and OEM sends 0FFFFh.

ReqUpdateUser

Code: 22h
Data Sequence: *User ID + Header Data + Flag [+ Feature Data]*
Direction: HOST->OEM
Remark:

HOST sends this command to update user data.
When HOST sends this command to OEM, OEM updates user data corresponding *User ID* that HOST sends.
The data structure following *User ID* is the same as ReqAddUser.
When the *flag* value is 0, the feature data is not updated.

AckUpdateUser

Code: 22h
Data Sequence: None
Direction: OEM->HOST
Remark:

OEM sends this command to HOST that it has updated user data.

ReqDeleteUser

Code: 23h
Data Sequence: *User ID*
Direction: HOST->OEM
Remark:

HOST sends this command to delete user data.
When HOST sends this command to OEM, OEM deletes user data corresponding *User ID* that HOST sends.

AckDeleteUser

Code: 23h
Data Sequence: None
Direction: OEM->HOST
Remark:

OEM sends this command to HOST to notice that it has deleted user.

ReqGetUser

Code: 24h
Data Sequence: *User ID + Flag* (1 Byte)
Direction: HOST->OEM
Remark:

HOST sends this command to get user data.
If the *Flag* value is 1, it means to send header data and feature data and if the *Flag* value is 0, it means to send only header data.

AckGetUser

Code: 24h
Data Sequence: *Result* (1Byte) [+ *User Data*]
Direction: OEM->HOST
Remark:

OEM sends this command to HOST to send user data.
If the *Result* value is 0, it means there is no user data corresponding to *User ID*.

ReqGetAllUser

Code: 25h
Data Sequence: *Flag* (1 Byte)
Direction: HOST->OEM
Remark:

HOST sends this command to get all user data saved in OEM.
If the *Flag* value is 1, it means to send header data and feature data and if the *Flag* value is 0, it means to send only header data.

AckGetAllUser

Code: 25h
Data Sequence: *User Data Sequence*.
Direction: OEM->HOST
Remark:

OEM sends this command to HOST to send *User Data Sequence*.
User Data Sequence is consists of sequence of User ID and User Data.
The length of *User Data Sequence* is as the user count.

ReqFindUser

Code: 26h
Data Sequence: None
Direction: HOST->OEM
Remark:

HOST sends this command to find user matching with extracted feature.

AckFindUser

Code: 26h
Data Sequence: *User ID*
Direction: OEM->HOST
Remark:

OEM sends this command to HOST to send *User ID* matching with extracted feature. If the value of User ID is 0FFFFh, it represents there is no user matching.

ReqGetPassword

Code: 50h
Data Sequence: None
Direction: HOST->OEM
Remark:

HOST sends this command to find matched password with extracted feature.

AckGetPassword

Code: 50h
Data Sequence: Result (1Byte) + Password (6Byte)
Direction: OEM->HOST
Remark:

When OEM receives this command, OEM tries to capture image from image sensor, extracts feature, finds matched user and send matched user's password to HOST
When Result is 0, it means matching is successful and OEM sends 6 byte password of matched user

When Result is 1, it means user doesn't press finger.

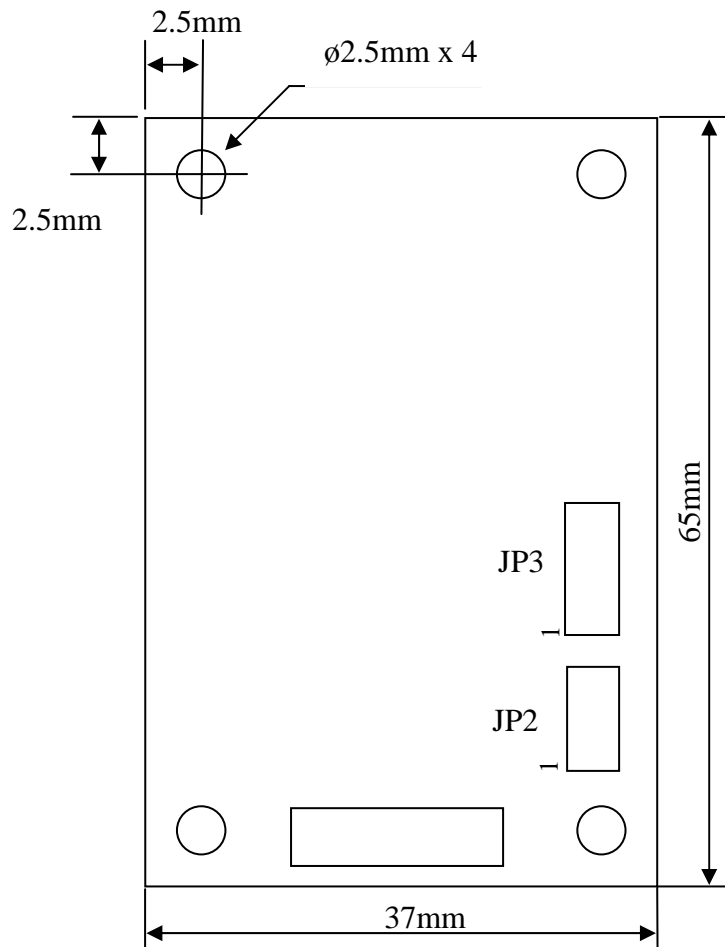
When Result is 2, it means matching is failure.

When Result is not 0, OEM sends 6 bytes zero character.

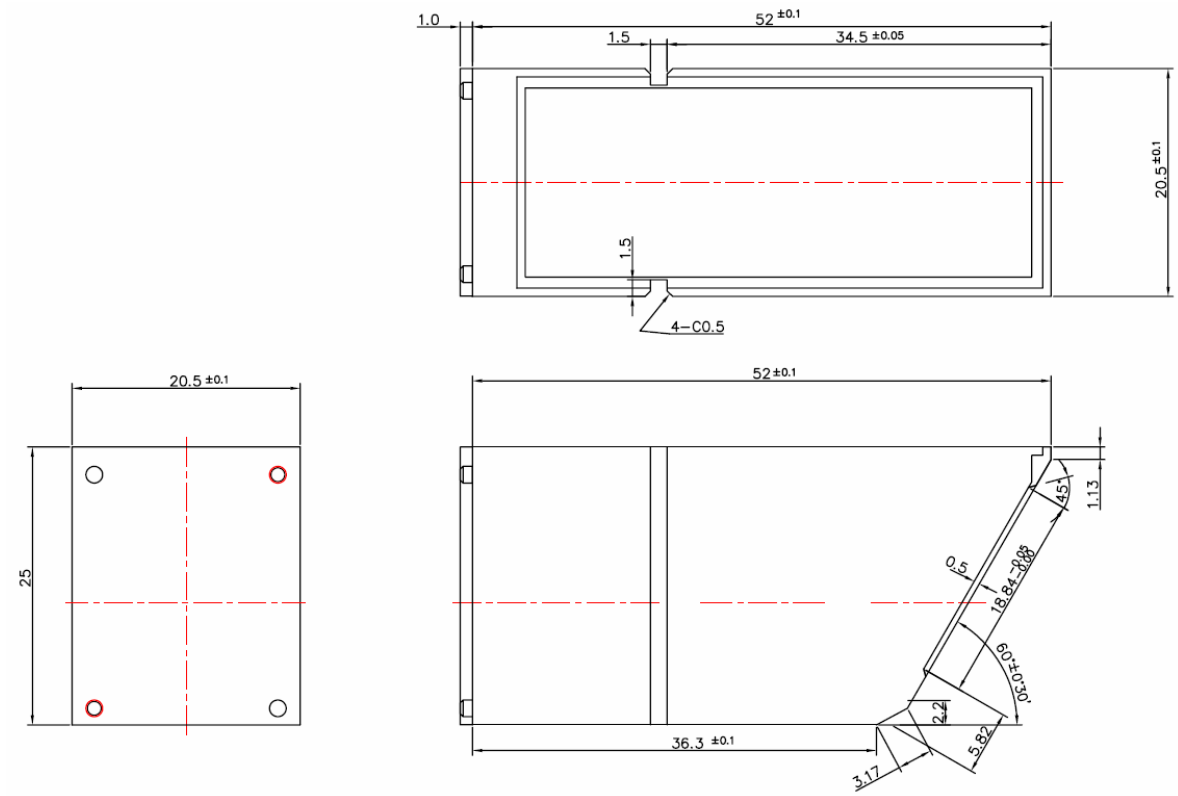
3. Hardware Specification

3-1. Size

PCB Size



Optical Sensor Size



3-2. Connectors

Table : JP2

Pin	Name	Description
1	LED (-)	It shows a communication status. Blinking whenever communicating with HOST.
2	VCC (+)	Power of LEDs & Buzzer. Connecting LEDs & Buzzer (+)
3	Buzzer (-)	Ring the Buzzer shortly two times whenever authentication is completed.
4	LED (-)	Continually power-on after one time blinking once the current flows in. It turns off during communicating with HOST.

Table : JP3

Pin	Name	Description
1	RxD	RS-232, RX
2	GND	RS-232, GND
3	TxD	RS-232, TX
4	GND	Power Ground
5	VCC	Power +5Vdc, ≥ 500 mA