



# **FINGERPRINT READER (UART) SN-FPR-UART**



## **USER'S MANUAL**

**V1.1**

**May 2011**

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Cytron Technologies Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Cytron Technologies's products as critical components in life support systems is not authorized except with express written approval by Cytron Technologies. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

## Index

1. Overview and features	2
2. Packing List	6
3. Connection	7
4. Communicate with SN-FPR-UART	12
5. Sample Application: Fingerprint Time Attendance System	27
6. Warranty	34

## **1. Overview and Features**

SN-FPR-UART is a compact fingerprint reader module which is specially designed for project development because of its robustness and the following outstanding features:

- Small in size, clear imaging, fast acquisition, high identification speed, able to sense on dry or wet fingers, supports wide range of applications.
- High stability, convenience (direct access to any MCU through Serial UART operation) and integrated with low current warning.
- Able to retrieve fingerprint raw data and fingerprint identity files.
- Using commercial algorithms with high recognition speed and the collection of fingerprint done by a gentle touch on the sensing area.

### **1.1. Applications**

- Security System
- Authentication
- Access Control
- User Registration
- Attendance System
- Vehicle Control

### **1.2. Technical specifications:**

- Supply voltage: DC 4.5 ~ 7.5V
- Operating current: <140mA
- Communication Interface: UART
  - Baud Rate     19200 bps
  - Start Bit       1
  - Stop Bit        1
  - Data Bit        8
  - Parity Bit      0
- Window area: 14 \* 18mm
- Fingerprint Image resolution: 500 DPI
- Storage capacity: 80 fingerprints
- False Accept Rate: <0.0001%
- False Reject Rate: <0.01%
- Security level: 0~9 adjustable (default 5)
- Fingerprint Image Input Time: < 1 sec
- Matching/Search Time: <1 sec
- Support 1: 1 and 1: N matching
- Support multiple fingerprints for a user

### 1.3. Block Diagram

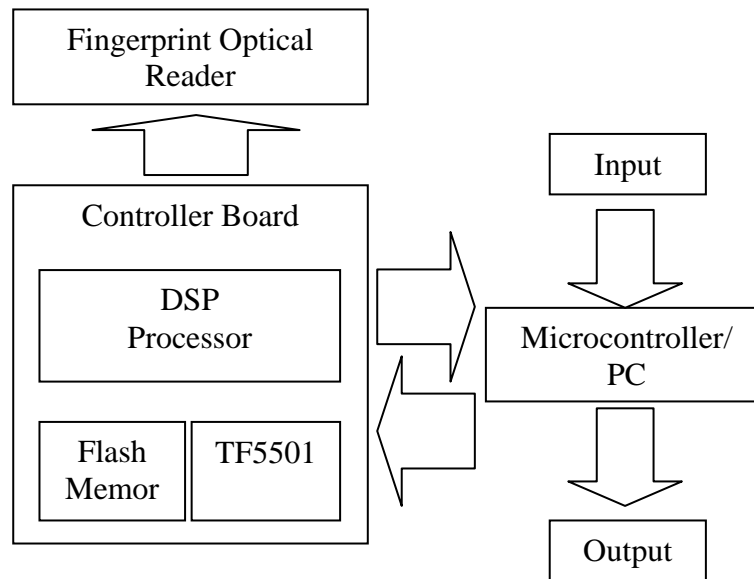


Figure 1.0 Block diagram

### 1.4. Proper way for fingerprint scanning

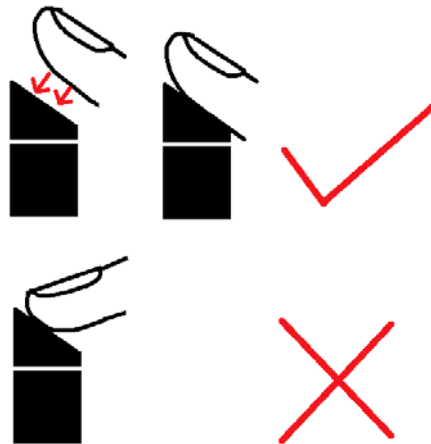


Figure 1.1

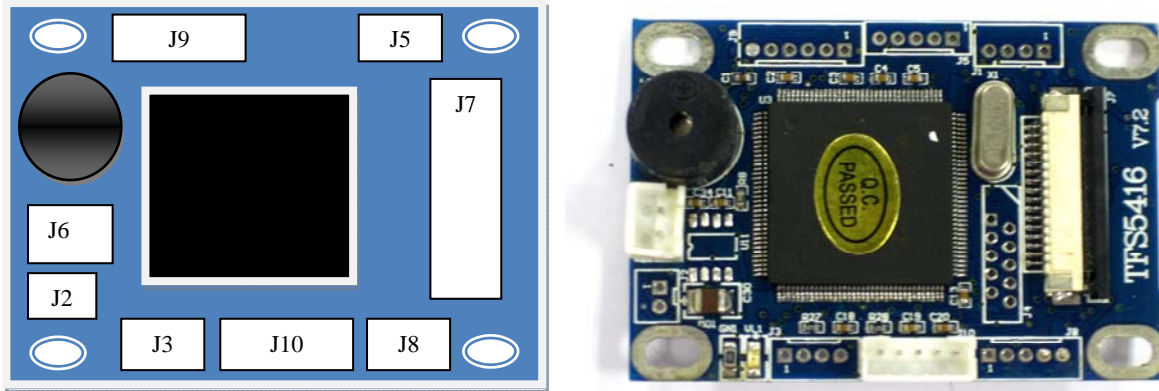


Figure 1.2 Appearance and structure of the board

### 1.5. Port Description

Port	Name	Pin	Specification	Note
J6	Power Supply	Pin1	Ground	5~7.5V same as J10 Pin1
		Pin2	VCC	
J7	Fingerprint collection window			
J10	Microcontroller Interface with power supply	Pin1	VCC	5~7.5V
		Pin2	UART TX	Support both 5V and 3.3V
		Pin3	UART RX	
		Pin4	Ground	
		Pin5	Enable Sleep (Optional)	Active Low

**Note:**

1. Pin with square pad considers as Pin1
2. Power Supply (J6) can be ignored if you have supplied power at UART (J10) line.
3. If system produces “Beep...Beep...Beep” sound and the red LED blinking means insufficient current provided and you are advice to use J6 as power supply.

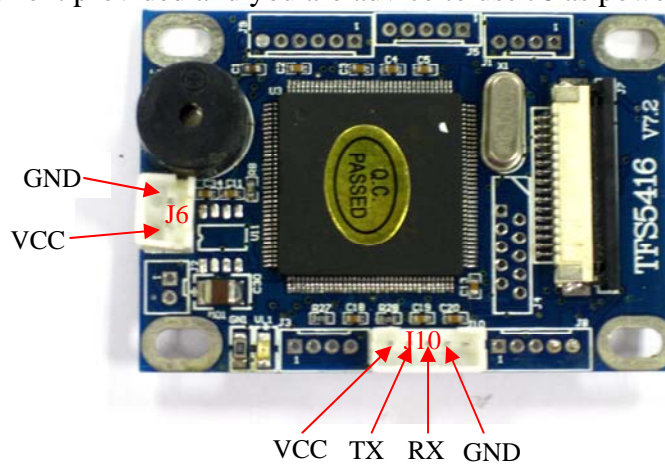


Figure 1.3 Ports and pins

### 3.1. Mechanical Dimensions

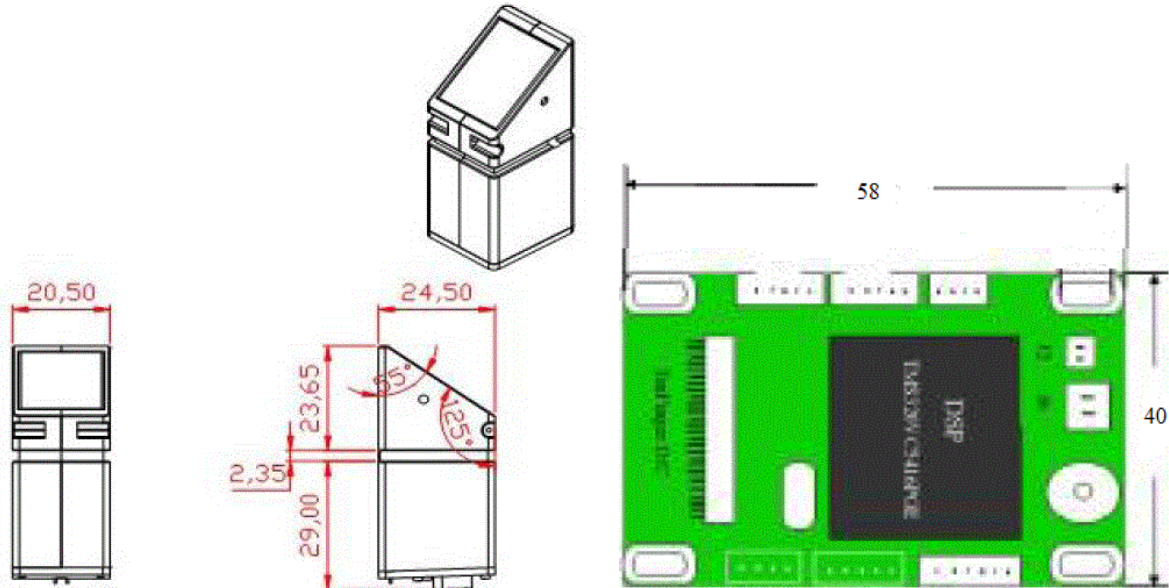


Figure 1.4 Mechanical dimensions in millimeter (mm)

### 3.2. Precaution

If using battery as power source alkaline battery is recommended.

If system produces Beep...Beep...Beep sound and the red LED blinking means insufficient current provided.

#### 4. Packing List

Please check the parts and components according to the packing list. If there are any parts missing, please contact us at [sales@cytron.com.my](mailto:sales@cytron.com.my) immediately.

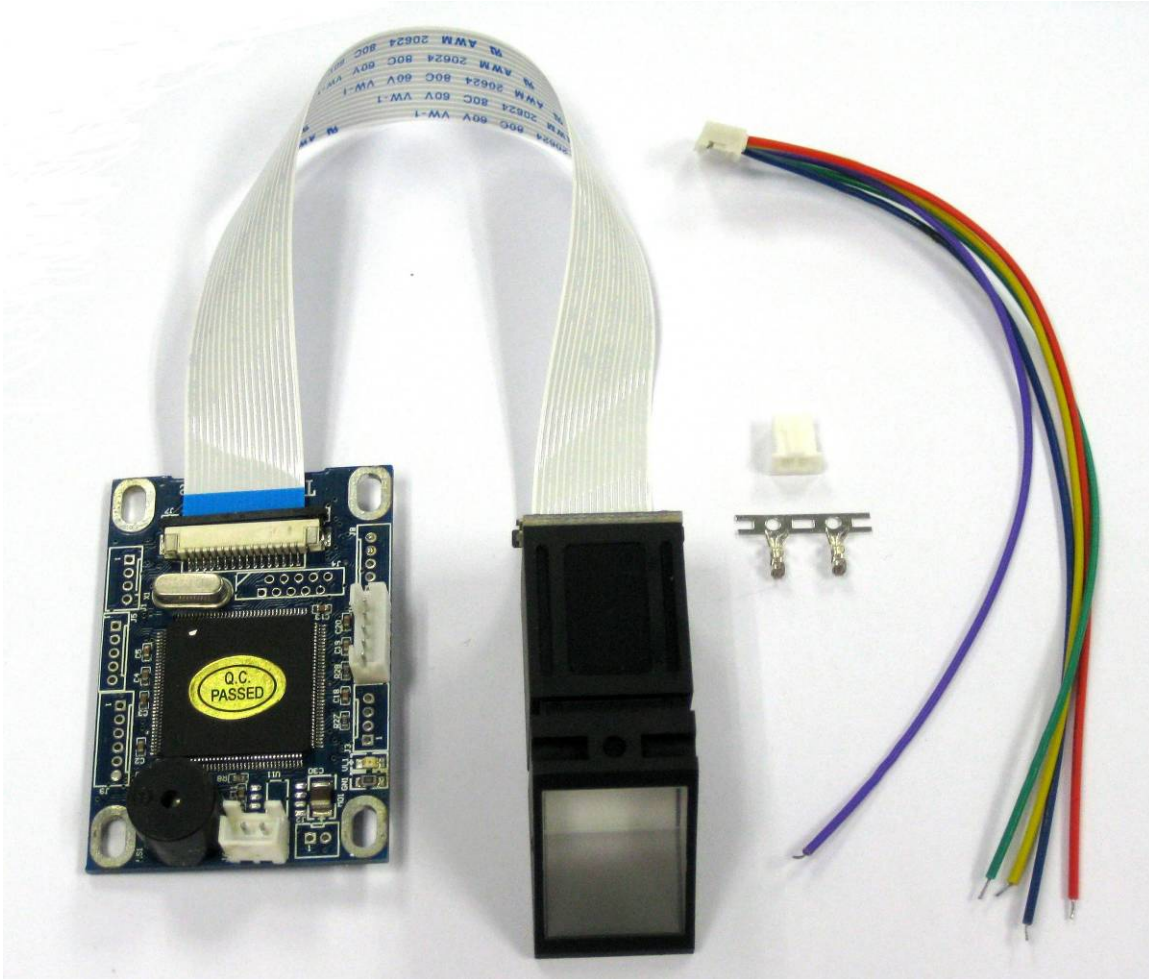


Figure 2.0

- i. 1 x DSP controller board
- ii. 1 x Optical fingerprint reader
- iii. 1 x Cable for J7
- iv. 1 x Cable for J10
- v. 1 x 2530-02 connector for J6



## 5. Connections

Cable connector for JP10 is provided. Connector 2510-04 can be connected to either UC00A (USB-UART) or PR29 (microcontroller board) UART port. This connector is not including in SN-FPR-UART packing list. User need to buy separately. Cable connector provided is for JP10. This cable need to be connected to SN-FPR-UART fingerprint reader. Figure 3.0 below shows the types of the connector which not include in SN-FPR-UART packing list.

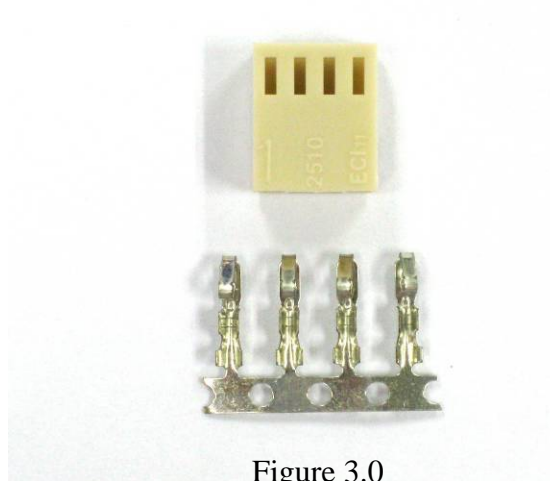


Figure 3.0

### 3.1 Guide for making 5 ways 2510 cable for fingerprint reader.

This section will guide user making 4 ways 2510 cable shown as figure below.

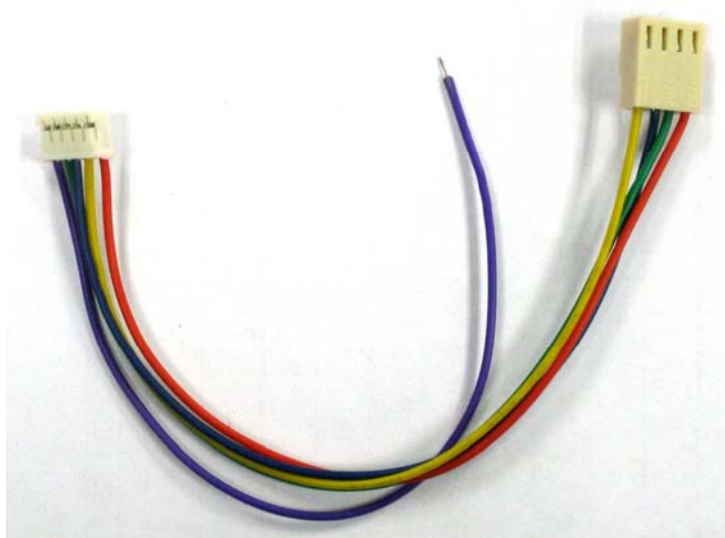


Figure 3.1



At step 5, ensure the connection is right. Rx pin of SN-FPR-UART must connect to Tx pin of microcontroller and Tx pin of SN-FPR-UART must connect to Rx pin of microcontroller.

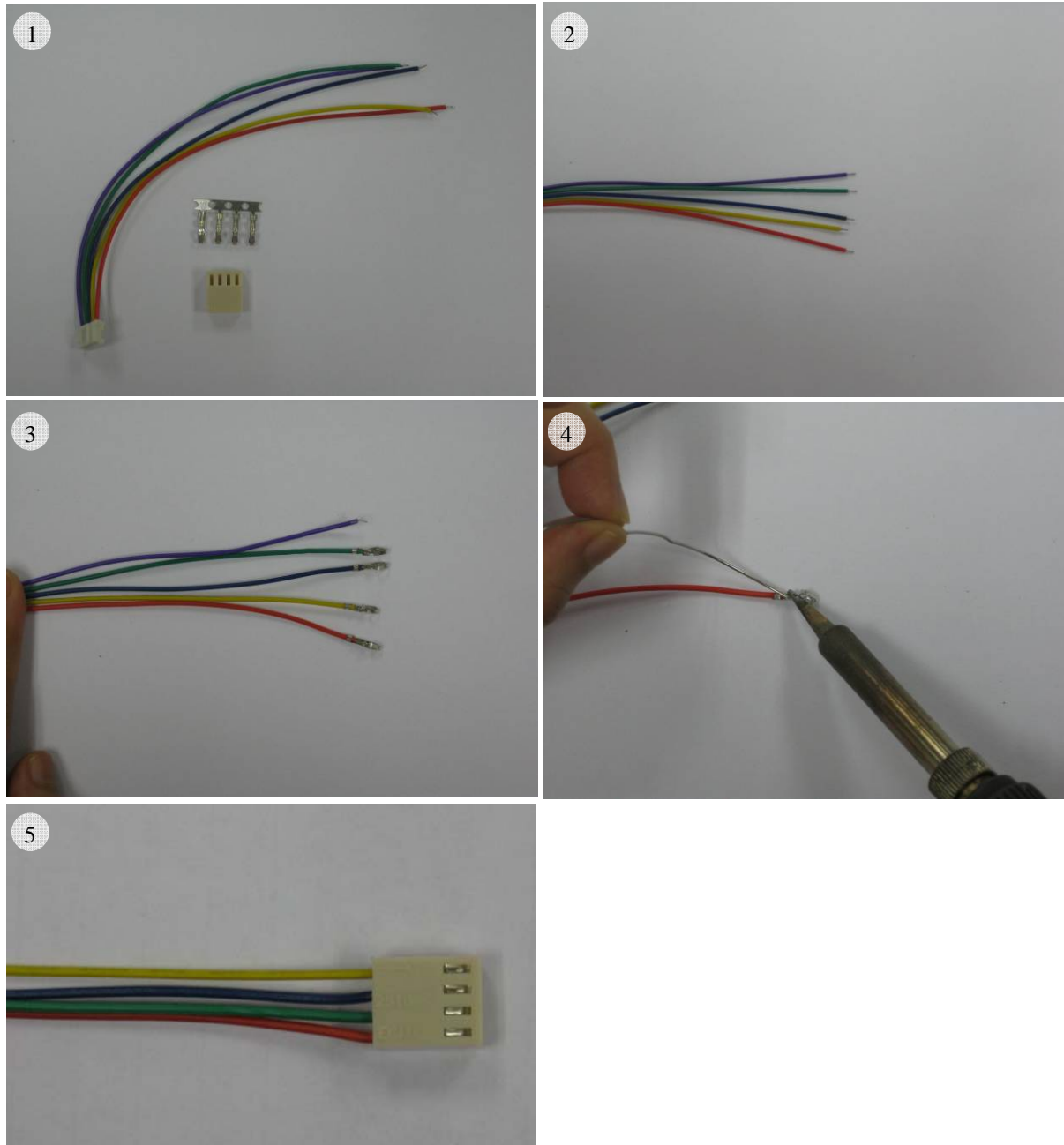


Figure 3.2

### 3.2 Possible connections for SN-FPR-UART fingerprint reader

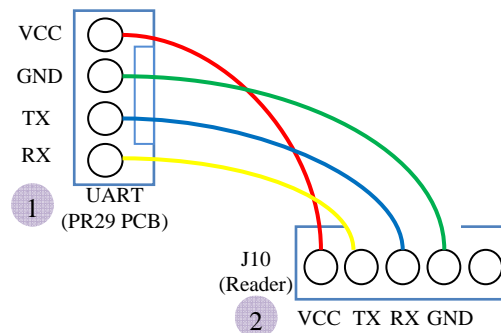


Figure 3.2 Fingerprint reader with UC00A (USB-UART) converter

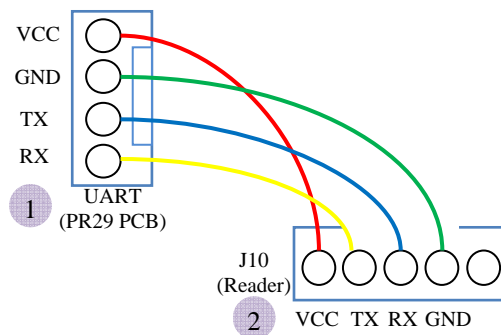
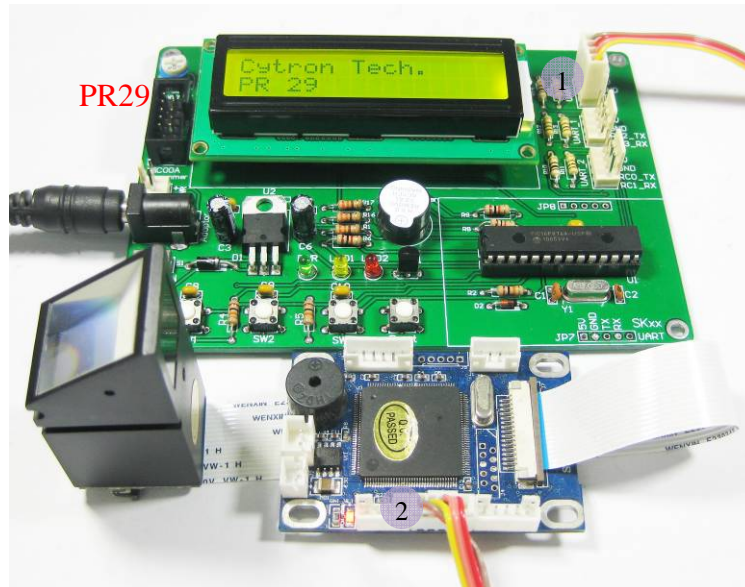


Figure 3.3 Fingerprint reader with PR29 PCB board

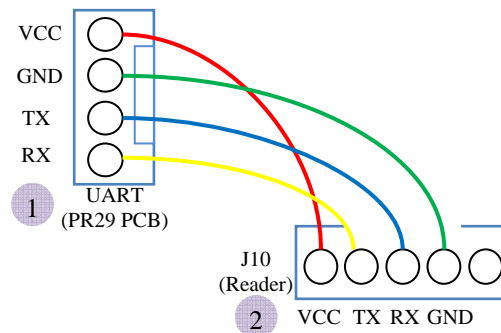
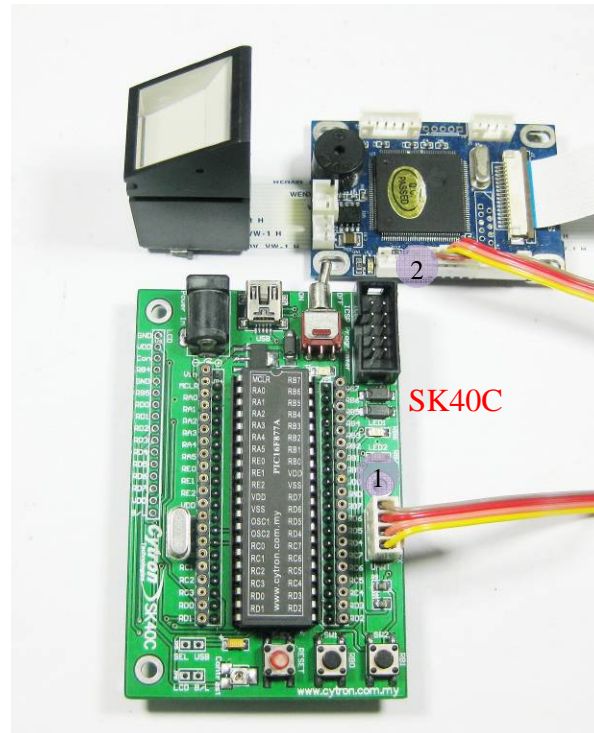


Figure 3.4 Fingerprint reader with SK40C- 40 pins PIC Start-Up kit

## 6. Communicate with SN-FPR-UART

### 6.1. Communication port descriptions

Baud Rate : 19200 bps  
Start bit : 1  
Stop bit : 1  
Data bit : 8  
Parity bit : 0

### 6.2. Format of Commands and Replies

In general, there are 2 types of commands and expected replies

Type 1: 8 bytes

Message format

Byte	1	2	3	4	5	6	7	8
Command	0xF5	CMD	P1	P2	P3	0	CHK	0xF5
Reply	0xF5	CMD	Q1	Q2	Q3	0	CHK	0xF5

Descriptions:

- CMD
    - Types of operations
    - Each operation has its own CMD
  - P1, P2, P3
    - Command arguments
  - Q1, Q2, Q3
    - Response arguments
    - Q3 normally used as an acknowledgement
- ```

#define ACK_SUCCESS          0x00  //operation success
#define ACK_FAIL             0x01  //operation fail
#define ACK_FULL             0x04  //memory full
#define ACK_NOUSER          0x05  //user does not exist
#define ACK_USER_EXISTS     0x07  //user already exists
#define ACK_TIMEOUT         0x08  //time out in fingerprint collection

```
- CHK
    - Checksum byte ( XOR 2<sup>nd</sup> Byte to 6<sup>th</sup> Byte )
    - Example:  $CHK = CMD \oplus P1 \oplus P2 \oplus P3 \oplus 0$

Type 2: > 8 bytes (divided into header + packet)

Header message format

| Byte    | 1    | 2   | 3         | 4          | 5  | 6 | 7   | 8    |
|---------|------|-----|-----------|------------|----|---|-----|------|
| Command | 0xF5 | CMD | Hi( Len ) | Low( Len ) | 0  | 0 | CHK | 0xF5 |
| Reply   | 0xF5 | CMD | Hi( Len ) | Low( Len ) | Q3 | 0 | CHK | 0xF5 |

Descriptions:

- CMD
  - Same as above
- Len
  - Length of data (in byte) in the packet ( 16bits - represents by two bytes)
- Hi( Len )
  - Length of the data (in byte) upper 8bits
- Low( Len )
  - Length of the data (in byte) lower 8bits
- CHK
  - Checksum byte ( XOR 2<sup>nd</sup> Byte to 6<sup>th</sup> Byte )

Packet message format

| Byte    | 1    | 2...Len + 1 | Len + 2 | Len + 3 |
|---------|------|-------------|---------|---------|
| Command | 0xF5 | Data        | CHK     | 0xF5    |
| Reply   | 0xF5 | Data        | CHK     | 0xF5    |

Descriptions:

- Len
  - Length of data (in byte)
- CHK
  - Checksum byte ( XOR from 2<sup>nd</sup> Byte to (Len – 2)<sup>th</sup> Byte)

### 6.3. List of Commonly Used Commands and Their Respective Replies

#### 6.3.1. Set / Read Fingerprint Input Mode (command/reply both consist of 8bytes)

There are two modes available for adding fingerprint into module: Allowed Repeat / Not Allowed Repeat for “Not Allowed Repeat” mode, each finger can only create a User and adding the same fingerprint into the module will return an error message.

Command and reply format

| Byte    | 1    | 2    | 3 | 4                                                                                                                               | 5                                           | 6 | 7   | 8    |
|---------|------|------|---|---------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------|---|-----|------|
| Command | 0xF5 | 0x2D | 0 | If 5 <sup>th</sup> Byte=0:<br><br>0: Allowed Repeat<br><br>1: Not Allowed Repeat<br><br><br>If 5 <sup>th</sup> Byte=1:<br><br>0 | 0: Set mode<br><br><br>1: Read current mode | 0 | CHK | 0xF5 |
| Reply   | 0xF5 | 0x2D | 0 | Current Mode                                                                                                                    | ACK_SUCCESS<br>ACK_FAIL                     | 0 | CHK | 0xF5 |

#### 6.3.2. Adding New Fingerprint (command and reply both consist of 8bytes)

For validity, user needs to provide his/her fingerprint for 3 times. Controller should send 3 different commands to the module for the process.

##### i. First Time

| Byte    | 1    | 2    | 3                   | 4                   | 5                                                  | 6 | 7   | 8    |
|---------|------|------|---------------------|---------------------|----------------------------------------------------|---|-----|------|
| Command | 0xF5 | 0x01 | User ID Upper 8 bit | User ID Lower 8 bit | Level of Authority (1/2/3)                         | 0 | CHK | 0xF5 |
| Reply   | 0xF5 | 0x01 | 0                   | 0                   | ACK_SUCCESS<br>ACK_FAIL<br>ACK_FULL<br>ACK_TIMEOUT | 0 | CHK | 0xF5 |

Note: User ID range: 0x001 – 0xFFF



ii. Second Time

| Byte    | 1    | 2    | 3                         | 4                         | 5                                      | 6 | 7   | 8    |
|---------|------|------|---------------------------|---------------------------|----------------------------------------|---|-----|------|
| Command | 0xF5 | 0x02 | User ID<br>Upper 8<br>bit | User ID<br>Lower 8<br>bit | Level of<br>Authority<br>(1/2/3)       | 0 | CHK | 0xF5 |
| Reply   | 0xF5 | 0x02 | 0                         | 0                         | ACK_SUCCESS<br>ACK_FAIL<br>ACK_TIMEOUT | 0 | CHK | 0xF5 |

iii. Third Time

| Byte    | 1    | 2    | 3                         | 4                         | 5                                                            | 6 | 7   | 8    |
|---------|------|------|---------------------------|---------------------------|--------------------------------------------------------------|---|-----|------|
| Command | 0xF5 | 0x03 | User ID<br>Upper 8<br>bit | User ID<br>Lower 8<br>bit | Level of<br>Authority<br>(1/2/3)                             | 0 | CHK | 0xF5 |
| Reply   | 0xF5 | 0x03 | 0                         | 0                         | ACK_SUCCESS<br>ACK_FAIL<br>ACK_USER_EXI<br>ST<br>ACK_TIMEOUT | 0 | CHK | 0xF5 |

Note: User ID must be the same for all the 3 commands

6.3.3. Delete Specific User (command and reply both consist of 8bytes)

| Byte    | 1    | 2    | 3                         | 4                         | 5                       | 6 | 7   | 8    |
|---------|------|------|---------------------------|---------------------------|-------------------------|---|-----|------|
| Command | 0xF5 | 0x04 | User ID<br>Upper 8<br>bit | User ID<br>Lower 8<br>bit | 0                       | 0 | CHK | 0xF5 |
| Reply   | 0xF5 | 0x04 | 0                         | 0                         | ACK_SUCCESS<br>ACK_FAIL | 0 | CHK | 0xF5 |

6.3.4. Delete All (command and reply both consist of 8bytes)

| Byte    | 1    | 2    | 3 | 4 | 5                       | 6 | 7   | 8    |
|---------|------|------|---|---|-------------------------|---|-----|------|
| Command | 0xF5 | 0x05 | 0 | 0 | 0                       | 0 | CHK | 0xF5 |
| Reply   | 0xF5 | 0x05 | 0 | 0 | ACK_SUCCESS<br>ACK_FAIL | 0 | CHK | 0xF5 |

6.3.5. Request for Total Number of User Registered in the Module (command and reply both consist of 8bytes)

| Byte    | 1    | 2    | 3                          | 4                          | 5                       | 6 | 7   | 8    |
|---------|------|------|----------------------------|----------------------------|-------------------------|---|-----|------|
| Command | 0xF5 | 0x09 | 0                          | 0                          | 0                       | 0 | CHK | 0xF5 |
| Reply   | 0xF5 | 0x09 | No. of user<br>Upper 8 bit | No. of user<br>Lower 8 bit | ACK_SUCCESS<br>ACK_FAIL | 0 | CHK | 0xF5 |

6.3.6. Matching 1: 1 (command and reply both consist of 8bytes)

| Byte    | 1    | 2    | 3                      | 4                      | 5                                      | 6 | 7   | 8    |
|---------|------|------|------------------------|------------------------|----------------------------------------|---|-----|------|
| Command | 0xF5 | 0x0B | User ID<br>Upper 8 bit | User ID<br>Lower 8 bit | 0                                      | 0 | CHK | 0xF5 |
| Reply   | 0xF5 | 0x0B | 0                      | 0                      | ACK_SUCCESS<br>ACK_FAIL<br>ACK_TIMEOUT | 0 | CHK | 0xF5 |

6.3.7. Matching 1: N (command and reply both consist of 8bytes)

| Byte    | 1    | 2    | 3                      | 4                      | 5                                                      | 6 | 7   | 8    |
|---------|------|------|------------------------|------------------------|--------------------------------------------------------|---|-----|------|
| Command | 0xF5 | 0x0C | 0                      | 0                      | 0                                                      | 0 | CHK | 0xF5 |
| Reply   | 0xF5 | 0x0C | User ID<br>Upper 8 bit | User ID<br>Lower 8 bit | Level of Authority(1/2/3)<br>ACK_NOUSER<br>ACK_TIMEOUT | 0 | CHK | 0xF5 |

6.3.8. Set / Read Fingerprint Acquisition Timeout Limit (command and reply both consist of 8bytes)

Command and Reply format

| Byte    | 1    | 2    | 3 | 4                                                                                            | 5                                                                 | 6 | 7   | 8    |
|---------|------|------|---|----------------------------------------------------------------------------------------------|-------------------------------------------------------------------|---|-----|------|
| Command | 0xF5 | 0x2E | 0 | If 5 <sup>th</sup> Byte=0:<br><br>New Timeout<br><br><br>If 5 <sup>th</sup> Byte=1:<br><br>0 | 0: Set New Timeout Limit<br><br><br>1: Read current Timeout Limit | 0 | CHK | 0xF5 |
| Reply   | 0xF5 | 0x2E | 0 | Current Timeout Limit                                                                        | ACK_SUCCESS<br>ACK_FAIL                                           | 0 | CHK | 0xF5 |

Note: Timeout limit (*Tout*) ranges from 0-255. Zero means no timeout while any value other than zero will causes the program to escape from fingerprint acquisition process after approximately  $Tout * 0.3\text{sec}$

## 6.4. List of Other Commands and Their Respective Replies

### 6.4.1. Request for User's Level of Authority

| Byte    | 1    | 2    | 3                         | 4                         | 5                                              | 6 | 7   | 8    |
|---------|------|------|---------------------------|---------------------------|------------------------------------------------|---|-----|------|
| Command | 0xF5 | 0x0A | User ID<br>Upper 8<br>bit | User ID<br>Lower 8<br>bit | 0                                              | 0 | CHK | 0xF5 |
| Reply   | 0xF5 | 0x0A | 0                         | 0                         | Level of<br>Authority(1/2/3)<br><br>ACK_NOUSER | 0 | CHK | 0xF5 |

### 6.4.2. Set / Read Matching Security Level (command and reply both consist of 8bytes)

| Byte    | 1    | 2    | 3 | 4                                                                                           | 5                                            | 6 | 7   | 8    |
|---------|------|------|---|---------------------------------------------------------------------------------------------|----------------------------------------------|---|-----|------|
| Command | 0xF5 | 0x28 | 0 | If 5 <sup>th</sup> Byte=0:<br><br>New value(0~9)<br><br>If 5 <sup>th</sup> Byte=1:<br><br>0 | 0: Set Level<br><br>1: Read current<br>Level | 0 | CHK | 0xF5 |
| Reply   | 0xF5 | 0x28 | 0 | Current<br>Level                                                                            | ACK_SUCCESS<br>ACK_FAIL                      | 0 | CHK | 0xF5 |

### 6.4.3. Acquire Image and Upload Image data (command 8bytes/ reply >8bytes)

#### Command format

| Byte    | 1    | 2    | 3 | 4 | 5 | 6 | 7   | 8    |
|---------|------|------|---|---|---|---|-----|------|
| Command | 0xF5 | 0x24 | 0 | 0 | 0 | 0 | CHK | 0xF5 |

#### Reply Format

##### i. Header

| Byte  | 1    | 2    | 3        | 4        | 5                                      | 6 | 7   | 8    |
|-------|------|------|----------|----------|----------------------------------------|---|-----|------|
| Reply | 0xF5 | 0x24 | Hi (Len) | Low(Len) | ACK_SUCCESS<br>ACK_FAIL<br>ACK_TIMEOUT | 0 | CHK | 0xF5 |

ii. Packet

|       |      |             |         |         |
|-------|------|-------------|---------|---------|
| Byte  | 1    | 2...Len + 1 | Len + 2 | Len + 3 |
| Reply | 0xF5 | Image Data  | CHK     | 0xF5    |

Note: In DSP processor, fingerprint images are 280\* 280 pixels, and each pixel represents by 8bits. For upload purpose, images are resized into 140\* 140 pixels to reduce the total amount of data to be sent. The image only considers upper 4bits of grayscale value for every pixel. Every byte of data represents 2 pixels (lower 4bits pixel1 and upper 4bits pixel2). Transmissions of data start from the first pixel of the first row. Total data of the image (in byte) =  $140 \text{column} * 140 \text{row} / 2$  (where 2pixels share a byte). Len = Total data of the image (in byte) = 9800 bytes

6.4.4. Acquire Image and Upload the Eigenvalue (command 8bytes/ reply >8bytes)

Command format

|         |      |      |   |   |   |   |     |      |
|---------|------|------|---|---|---|---|-----|------|
| Byte    | 1    | 2    | 3 | 4 | 5 | 6 | 7   | 8    |
| Command | 0xF5 | 0x23 | 0 | 0 | 0 | 0 | CHK | 0xF5 |

Reply Format

i. Header

|       |      |      |          |          |                                        |   |     |      |
|-------|------|------|----------|----------|----------------------------------------|---|-----|------|
| Byte  | 1    | 2    | 3        | 4        | 5                                      | 6 | 7   | 8    |
| Reply | 0xF5 | 0x23 | Hi (Len) | Low(Len) | ACK_SUCCESS<br>ACK_FAIL<br>ACK_TIMEOUT | 0 | CHK | 0xF5 |

ii. Packet

|       |      |   |   |   |                 |         |         |
|-------|------|---|---|---|-----------------|---------|---------|
| Byte  | 1    | 2 | 3 | 4 | 5... Len + 1    | Len + 2 | Len + 3 |
| Reply | 0xF5 | 0 | 0 | 0 | Eigenvalue data | CHK     | 0xF5    |

Note: Eigenvalue data length = 193 bytes = Len + 1 – 4

6.4.5. Perform Fingerprint Matching (Scanning) with Specific Eigenvalue (command >8bytes/ reply 8bytes)

Command format

i. Header

|         |      |      |          |          |   |   |     |      |
|---------|------|------|----------|----------|---|---|-----|------|
| Byte    | 1    | 2    | 3        | 4        | 5 | 6 | 7   | 8    |
| Command | 0xF5 | 0x44 | Hi (Len) | Low(Len) | 0 | 0 | CHK | 0xF5 |

ii. Packet

|         |      |   |   |   |                 |         |         |
|---------|------|---|---|---|-----------------|---------|---------|
| Byte    | 1    | 2 | 3 | 4 | 5... Len + 1    | Len + 2 | Len + 3 |
| Command | 0xF5 | 0 | 0 | 0 | Eigenvalue data | CHK     | 0xF5    |

Reply format

|       |      |      |   |   |                                                |   |     |      |
|-------|------|------|---|---|------------------------------------------------|---|-----|------|
| Byte  | 1    | 2    | 3 | 4 | 5                                              | 6 | 7   | 8    |
| Reply | 0xF5 | 0x44 | 0 | 0 | ACK_SUCCESS<br><br>ACK_FAIL<br><br>ACK_TIMEOUT | 0 | CHK | 0xF5 |

#### 6.4.6. Matching Eigenvalue with Database in DSP Processor 1: 1 (command >8bytes/ reply 8bytes)

Command format

i. Header

|         |      |      |          |          |   |   |     |      |
|---------|------|------|----------|----------|---|---|-----|------|
| Byte    | 1    | 2    | 3        | 4        | 5 | 6 | 7   | 8    |
| Command | 0xF5 | 0x42 | Hi (Len) | Low(Len) | 0 | 0 | CHK | 0xF5 |

ii. Packet

|         |      |                           |                           |   |                 |         |         |
|---------|------|---------------------------|---------------------------|---|-----------------|---------|---------|
| Byte    | 1    | 2                         | 3                         | 4 | 5... Len + 1    | Len + 2 | Len + 3 |
| Command | 0xF5 | User ID<br>Upper<br>8bits | User ID<br>Lower<br>8bits | 0 | Eigenvalue data | CHK     | 0xF5    |

Note: Eigenvalue data length = 193 bytes = Len + 1 – 4

Reply format

|       |      |      |   |   |                         |   |     |      |
|-------|------|------|---|---|-------------------------|---|-----|------|
| Byte  | 1    | 2    | 3 | 4 | 5                       | 6 | 7   | 8    |
| Reply | 0xF5 | 0x42 | 0 | 0 | ACK_SUCCESS<br>ACK_FAIL | 0 | CHK | 0xF5 |

#### 6.4.7. Matching Eigenvalue with Database in DSP Processor 1: N (command >8bytes/ reply 8bytes)

Command format

i. Header

|         |      |      |          |          |   |   |     |      |
|---------|------|------|----------|----------|---|---|-----|------|
| Byte    | 1    | 2    | 3        | 4        | 5 | 6 | 7   | 8    |
| Command | 0xF5 | 0x43 | Hi (Len) | Low(Len) | 0 | 0 | CHK | 0xF5 |

ii. Packet

|      |   |   |   |   |              |       |       |
|------|---|---|---|---|--------------|-------|-------|
| Byte | 1 | 2 | 3 | 4 | 5... Len + 1 | Len + | Len + |
|------|---|---|---|---|--------------|-------|-------|

|         |      |   |   |   |                 |     |      |
|---------|------|---|---|---|-----------------|-----|------|
|         |      |   |   |   |                 | 2   | 3    |
| Command | 0xF5 | 0 | 0 | 0 | Eigenvalue data | CHK | 0xF5 |

Note: Eigenvalue data length = 193 bytes = Len + 1 – 4

Reply format

|       |      |      |                           |                           |                                                 |   |     |      |
|-------|------|------|---------------------------|---------------------------|-------------------------------------------------|---|-----|------|
| Byte  | 1    | 2    | 3                         | 4                         | 5                                               | 6 | 7   | 8    |
| Reply | 0xF5 | 0x43 | User ID<br>Upper<br>8bits | User ID<br>Lower<br>8bits | Level of<br>Authority (1/2/3)<br><br>ACK_NOUSER | 0 | CHK | 0xF5 |

6.4.8. Upload Eigenvalue of the Specified User from DSP Processor Database  
(command 8bytes/ reply >8bytes)

Command format

|         |      |      |                           |                           |   |   |     |      |
|---------|------|------|---------------------------|---------------------------|---|---|-----|------|
| Byte    | 1    | 2    | 3                         | 4                         | 5 | 6 | 7   | 8    |
| Command | 0xF5 | 0x31 | User ID<br>Upper<br>8bits | User ID<br>Lower<br>8bits | 0 | 0 | CHK | 0xF5 |

Reply Format

i. Header

|       |      |      |          |          |                                      |   |     |      |
|-------|------|------|----------|----------|--------------------------------------|---|-----|------|
| Byte  | 1    | 2    | 3        | 4        | 5                                    | 6 | 7   | 8    |
| Reply | 0xF5 | 0x31 | Hi (Len) | Low(Len) | ACK_SUCCES<br>ACK_FAIL<br>ACK_NOUSER | 0 | CHK | 0xF5 |

ii. Packet

|       |      |                           |                           |                                  |                 |            |            |
|-------|------|---------------------------|---------------------------|----------------------------------|-----------------|------------|------------|
| Byte  | 1    | 2                         | 3                         | 4                                | 5... Len + 1    | Len +<br>2 | Len +<br>3 |
| Reply | 0xF5 | User ID<br>Upper<br>8bits | User ID<br>Lower<br>8bits | Level of<br>Authority<br>(1/2/3) | Eigenvalue data | CHK        | 0xF5       |

Note: Eigenvalue data length = 193 bytes = Len + 1 – 4

6.4.9. Assigns Eigenvalue to a Specified User in DSP Processor Database (command  
>8bytes/ reply 8bytes)

Command format



i. Header

| Byte    | 1    | 2    | 3        | 4        | 5 | 6 | 7   | 8    |
|---------|------|------|----------|----------|---|---|-----|------|
| Command | 0xF5 | 0x41 | Hi (Len) | Low(Len) | 0 | 0 | CHK | 0xF5 |

ii. Packet

| Byte    | 1    | 2                   | 3                   | 4                          | 5... Len + 1    | Len + 2 | Len + 3 |
|---------|------|---------------------|---------------------|----------------------------|-----------------|---------|---------|
| Command | 0xF5 | User ID Upper 8bits | User ID Lower 8bits | Level of Authority (1/2/3) | Eigenvalue data | CHK     | 0xF5    |

Note: Eigenvalue data length = 193 bytes = Len + 1 – 4

Reply format

| Byte  | 1    | 2    | 3                   | 4                   | 5                       | 6 | 7   | 8    |
|-------|------|------|---------------------|---------------------|-------------------------|---|-----|------|
| Reply | 0xF5 | 0x41 | User ID Upper 8bits | User ID Lower 8bits | ACK_SUCCESS<br>ACK_FAIL | 0 | CHK | 0xF5 |

#### 6.4.10. Retrieve Level of Authority of All Registered Users (command 8bytes/ reply >8bytes)

Command format

| Byte    | 1    | 2    | 3 | 4 | 5 | 6 | 7   | 8    |
|---------|------|------|---|---|---|---|-----|------|
| Command | 0xF5 | 0x2B | 0 | 0 | 0 | 0 | CHK | 0xF5 |

Reply Format

i. Header

| Byte  | 1    | 2    | 3        | 4        | 5                     | 6 | 7   | 8    |
|-------|------|------|----------|----------|-----------------------|---|-----|------|
| Reply | 0xF5 | 0x2B | Hi (Len) | Low(Len) | ACK_SUCCE<br>ACK_FAIL | 0 | CHK | 0xF5 |

ii. Packet

| Byte  | 1    | 2                       | 3                       | 4... Len + 1                   | Len + 2 | Len + 2 |
|-------|------|-------------------------|-------------------------|--------------------------------|---------|---------|
| Reply | 0xF5 | No. of user Upper 8 bit | No. of user Upper 8 bit | User ID and Level of Authority | CHK     | 0xF5    |

Note: Format for 4<sup>th</sup> Byte until Len + 1 are as following:

User ID and Level of Authority

| Byte | 4                       | 5                       | 6                                         | 7                       | 8                       | 9                                         | ... |
|------|-------------------------|-------------------------|-------------------------------------------|-------------------------|-------------------------|-------------------------------------------|-----|
| Data | User ID1<br>Upper 8bits | User ID1<br>Lower 8bits | Level of Authority<br>of User1<br>(1/2/3) | User ID2<br>Upper 8bits | User ID2<br>Lower 8bits | Level of Authority<br>of User2<br>(1/2/3) | ... |

6.4.11. Sleep the device (command and reply both consist of 8bytes)

Command format

| Byte    | 1    | 2    | 3 | 4 | 5 | 6 | 7   | 8    |
|---------|------|------|---|---|---|---|-----|------|
| Command | 0xF5 | 0x2C | 0 | 0 | 0 | 0 | CHK | 0xF5 |

Reply format

| Byte  | 1    | 2    | 3 | 4 | 5 | 6 | 7   | 8    |
|-------|------|------|---|---|---|---|-----|------|
| Reply | 0xF5 | 0x2C | 0 | 0 | 0 | 0 | CHK | 0xF5 |

## 4.5 Sample Commands/ Replies

### 4.5.1 Sample 1 : Adding New Fingerprint

Let's assume we are going to add a new user into the memory at location 0x001 with Level of Authority 3.

First command to be sent

| Byte    | 1    | 2    | 3    | 4    | 5    | 6 | 7    | 8    |
|---------|------|------|------|------|------|---|------|------|
| Command | 0xF5 | 0x01 | 0x00 | 0x01 | 0x03 | 0 | 0x03 | 0xF5 |

$$*CHK = 0x01 \oplus 0x00 \oplus 0x01 \oplus 0x03 \oplus 0x00 = 0x03$$

Now, controller board will produce a “Beep” sound and the sensing area light up. User has to gently press his/her finger on the sensing area until occurrence of another “Beep” sound.

Expected reply is as below

| Byte  | 1    | 2    | 3 | 4 | 5                                                        | 6 | 7   | 8    |
|-------|------|------|---|---|----------------------------------------------------------|---|-----|------|
| Reply | 0xF5 | 0x01 | 0 | 0 | ACK_SUCCESS /<br>ACK_FAIL /<br>ACK_FULL /<br>ACK_TIMEOUT | 0 | CHK | 0xF5 |

CHK depends on the acknowledgement.

If fifth byte returns ACK\_SUCCESS, we can continue to send second command which is shown below.

| Byte    | 1    | 2    | 3    | 4    | 5    | 6 | 7    | 8    |
|---------|------|------|------|------|------|---|------|------|
| Command | 0xF5 | 0x02 | 0x00 | 0x01 | 0x03 | 0 | 0x00 | 0xF5 |

Again, the same user has to scan his/her finger again on the sensing area.  
 Reply is either one of the acknowledgements show below

| Byte  | 1    | 2    | 3 | 4 | 5                                          | 6 | 7   | 8    |
|-------|------|------|---|---|--------------------------------------------|---|-----|------|
| Reply | 0xF5 | 0x02 | 0 | 0 | ACK_SUCCESS /<br>ACK_FAIL /<br>ACK_TIMEOUT | 0 | CHK | 0xF5 |

CHK is always depends on 2<sup>nd</sup> to 6<sup>th</sup> byte for every 8 bytes command.

If again the second reply gives ACK\_SUCCESS, we are one step closer to the completion of the adding operation. Now, issue the last command which shown below

| Byte    | 1    | 2    | 3    | 4    | 5    | 6 | 7    | 8    |
|---------|------|------|------|------|------|---|------|------|
| Command | 0xF5 | 0x03 | 0x00 | 0x01 | 0x03 | 0 | 0x00 | 0xF5 |

User may receive anyone of the reply below. ACK\_SUCCESS indicates the fingerprint has been successfully added to the memory location 0x001 with Level of Authority 3. Note that fingerprint reader not storing the image of the fingerprint but a template which contains eigenvalues of the fingerprint.

| Byte  | 1    | 2    | 3 | 4 | 5                                                                  | 6 | 7   | 8    |
|-------|------|------|---|---|--------------------------------------------------------------------|---|-----|------|
| Reply | 0xF5 | 0x03 | 0 | 0 | ACK_SUCCESS /<br>ACK_FAIL /<br>ACK_USER_EXI<br>ST /<br>ACK_TIMEOUT | 0 | CHK | 0xF5 |

#### 4.5.2 Sample 2 : Delete a Fingerprint

Users are able to delete selected fingerprint template and also delete all template by a single command message. For Sample 2, let's assume that we are now going to delete the fingerprint which is just added in Sample 1. By referring to the format above, we can easily construct the command to delete the fingerprint added in Sample 1 by simply replace both User ID upper and lower bytes with the ID in sample 1. Please note that CHK value also needed to be calculate again. Figure below shows the command that is going to be issued.

| Byte    | 1    | 2    | 3    | 4    | 5 | 6 | 7    | 8    |
|---------|------|------|------|------|---|---|------|------|
| Command | 0xF5 | 0x04 | 0x00 | 0x01 | 0 | 0 | 0x05 | 0xF5 |

There are only two possible acknowledgements available in the reply. ACK\_SUCCESS simply means that the fingerprint is deleted successfully while ACK\_FAIL means unsuccessful deletion.

| Byte  | 1    | 2    | 3 | 4 | 5                         | 6 | 7   | 8    |
|-------|------|------|---|---|---------------------------|---|-----|------|
| Reply | 0xF5 | 0x04 | 0 | 0 | ACK_SUCCESS /<br>ACK_FAIL | 0 | CHK | 0xF5 |

#### 4.5.3 Sample 3: Matching 1: N

In this example, we are going to discuss on the command to perform 1: N matching. 1: N matching is actually comparing the incoming fingerprint eigenvalue with all the available templates in the memory. The command for this operation is straight forward but there are few possible replies to take care of.

| Byte    | 1    | 2    | 3 | 4 | 5 | 6 | 7    | 8    |
|---------|------|------|---|---|---|---|------|------|
| Command | 0xF5 | 0x0C | 0 | 0 | 0 | 0 | 0x0C | 0xF5 |

The command will be the same for all the times as shown in the table above and remember to replace CHK with appropriate value. Once the command is sent, the optical sensing area will light up with a “Beep” sound and user has to scan his/her finger on it. After the scanning process reader will send back an 8 bytes message to the user. There are three possible acknowledgements which may appear in the message. Let's discuss them one by one.

##### Case 1: Matched

| Byte  | 1    | 2    | 3    | 4    | 5    | 6 | 7    | 8    |
|-------|------|------|------|------|------|---|------|------|
| Reply | 0xF5 | 0x0C | 0x00 | 0x02 | 0x02 | 0 | 0x0C | 0xF5 |

Message above shows the fingerprint scanned match with User ID 0x002 which is assigned with Level 2 authority. Every template stored is assigned with respective Level of Authority during adding process.

##### Case 2: Does Not Match

| Byte  | 1    | 2    | 3    | 4    | 5          | 6 | 7    | 8    |
|-------|------|------|------|------|------------|---|------|------|
| Reply | 0xF5 | 0x0C | 0x00 | 0x00 | ACK_NOUSER | 0 | 0x0B | 0xF5 |

Above message will be returned from the DSP processor where there is zero in both 3<sup>rd</sup> and 4<sup>th</sup> byte and with ACK\_NOUSER in the 5<sup>th</sup> byte showing user does not exist in the memory. ACK\_NOUSER is actually 0x07 which is already stated above.

### Case 3: Acquisition Process Time Out

|       |      |      |      |      |             |   |      |      |
|-------|------|------|------|------|-------------|---|------|------|
| Byte  | 1    | 2    | 3    | 4    | 5           | 6 | 7    | 8    |
| Reply | 0xF5 | 0x0C | 0x00 | 0x00 | ACK_TIMEOUT | 0 | 0x04 | 0xF5 |

Default setting for SN-FPR-UART fingerprint reader is set to be no time out. So by default, optical scanner will be waiting for user's finger forever if there is no input. User has to set the time out option by another command first in order to encounter this case in matching process. We are not going to discuss on time out setting command here.

## 7. Sample Application: Fingerprint Time Attendance System

### Cytron Technologies Fingerprint Time Attendance Application

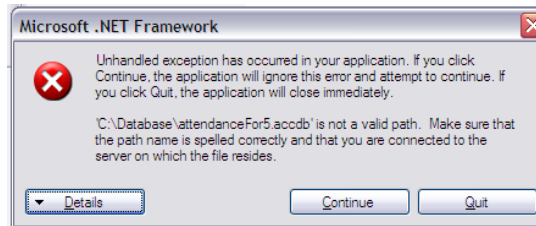
#### Introduction and Description

Cytron Technologies Fingerprint Time Attendance Application is developed by using Microsoft Visual Basic 2008 Express Edition. This application is a sample time attendance system application which is expected to replace punch card system and attendance list signing in neither workplace nor school. However this application is using fingerprints recognition instead of punch card. The application is specially designed to support Cytron SN-FPR-UART fingerprints reader with support of UC00A USB-UART converter which both of them can be easily obtained from [www.cytron.com.my](http://www.cytron.com.my)

Cytron Technologies Fingerprint Time Attendance Application is designed to monitor up to 5 users for 5 days (Monday – Friday). *The application will ignore any marking/authentication on Saturday and Sunday.* As this is only a sample application with Visual Basic project files provided, you can always modify the program to make it better suited for your own preference. This section will show user the way to set up and run the application. Please read the “**Read Me First**” text file before start the program.

#### Steps before starting of Cytron Technologies Fingerprint Time Attendance Application

1. Download the application zip file from Cytron Technologies website
2. Extract the zip file, it contains an executable application and two folders “Database” and “Cytron Technologies Fingerprint Time Attendance Application”
3. Before you open the exe file, make sure you have moved “**Database**” folder to **Drive C:\** of your computer. Else, the error below will be prompt out.



4. Now, you can double click on executable application named “Cytron Technologies Fingerprint Time Attendance Application.exe” to run the application.

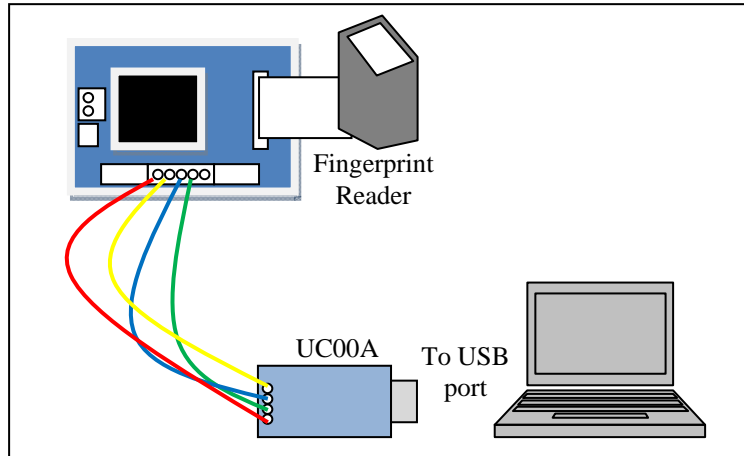


Figure 5.0 Hardware connection

#### Steps to Open Serial Port

1. Make sure all connections are correctly connected and fingerprint reader is switched on.
2. Firstly, choose the correct COM port, if you not sure how to choose the correct COM port please refer to UC00A USB-UART converter user's manual.
3. Select 19200 bps for the Baud Rate and click "Open Port". Figure below shows the port opening setting.



4. Bottom of the window will shows "Connected" if the application establishes a connection successfully.
5. Now, the application is ready to be used.
6. For First-time user, please "Reset" the system and assign an Administrator.

#### Steps to Reset the System

1. I assumed the application is already connected to the COM port.
2. Click on "Reset" button at the right hand side of the window.
3. Another window will show up and ask for Username and Password.



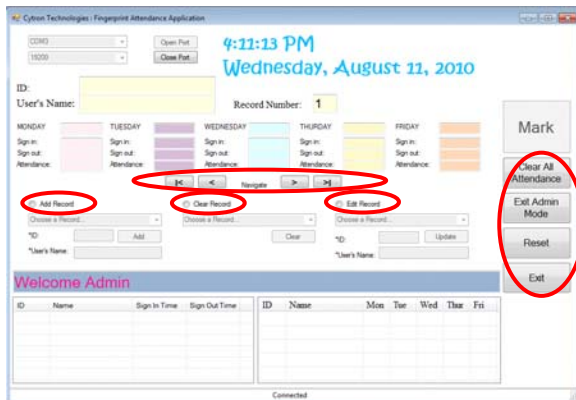
4. Default Username = **cytron**
- Default Password = **cytron**
5. Administrator must be assigned. Thus, instructions will be prompted to complete the operations.

#### Steps to Add, Edit, Clear and Navigate through the database (Admin only)

1. In order to perform all these operations, admin has to click on "Mark" button and scan his/her



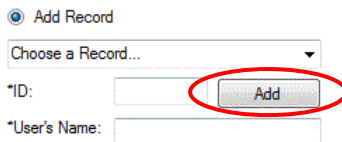
finger. System welcome message will be shown.



2. Notice that more operation are available in Admin mode

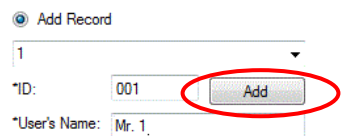
3. Steps for *Add* operation

i. Check on “Add Record”.



ii. Choose a record from the list. Please note that only empty records will be shown in the list.

iii. Key in ID and Name for the new user and press “Add” button.

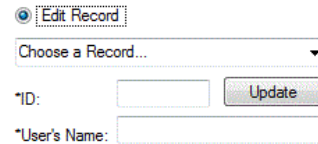


iv. New user has to scan his/her finger to complete the adding process.

v. Newly added record will be shown in the attendance list.

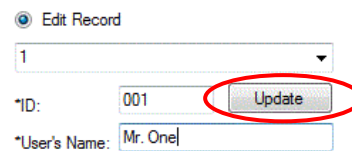
4. Steps for *Edit* operation

i. Check on “Edit Record”



ii. Choose a record from the list. Please note that only registered record will be shown in the list.

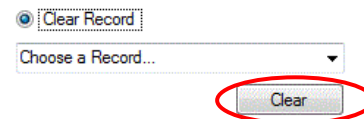
iii. Edit the ID and Name and click on “Update” button.



iv. Updated ID and/or Name will be shown.

5. Steps for *Clear* operation

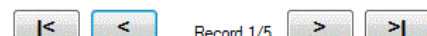
i. Check on “Clear Record”



ii. Choose a record from the list. Please note that only registered record will be shown in the list.

iii. Press on “Clear” button and the operation is done.

6. For record *navigation*, simply click on the arrow sign buttons to navigate next, previous, first or last record.



Function of “Clear All Attendance” button is to clear all the marked attendance and its respective signed in and signed out time for

the whole week. Please make sure the database is exported before performing this operation.

Click on “Exit Admin Mode” to turn the application back into a typical attendance marking system to serve all the registered users.

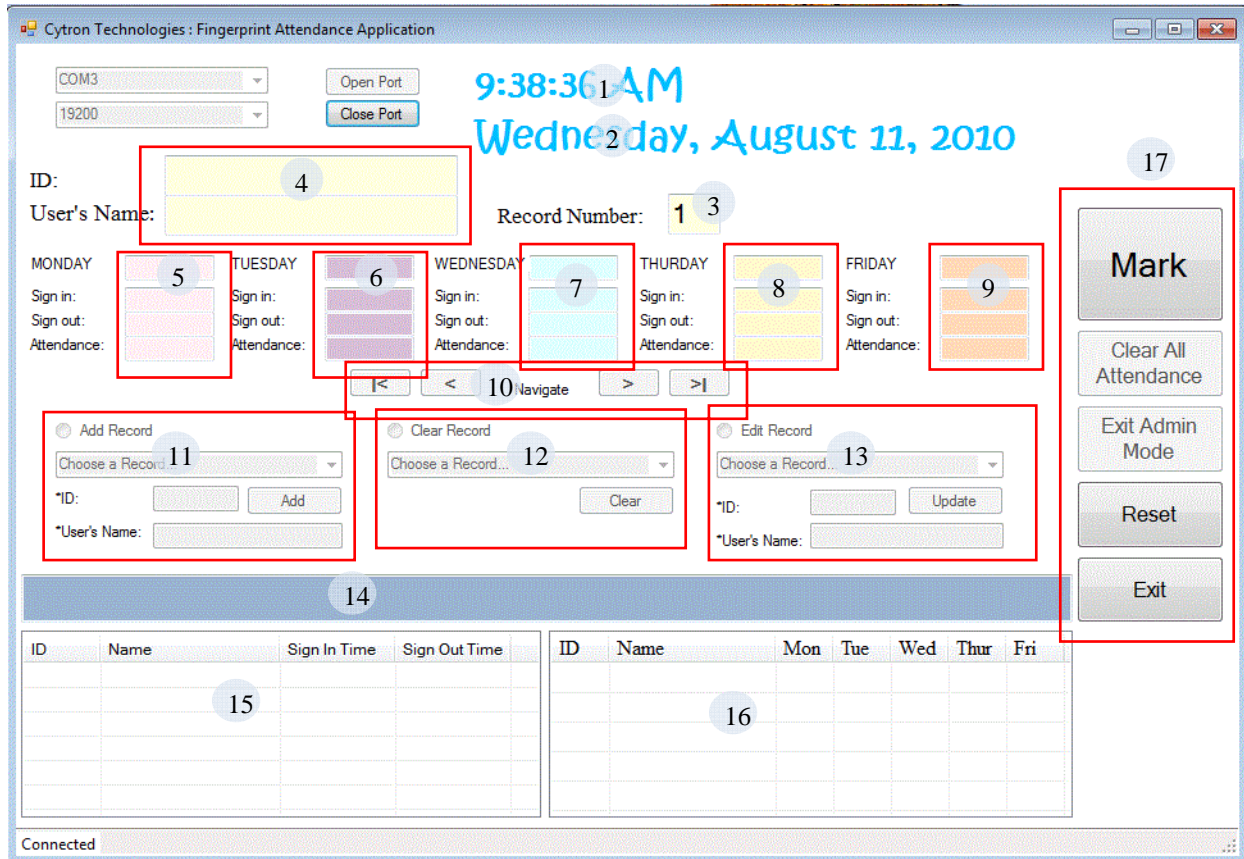


Figure 5.1 Layout

- |                                           |                                                   |
|-------------------------------------------|---------------------------------------------------|
| 1. Current time                           | Attendance system only work from Monday to Friday |
| 2. Today's date and day of week           |                                                   |
| 3. Record number                          |                                                   |
| 4. User ID and Name                       | Show the record directly from database            |
| 5. Monday's details for specified user    |                                                   |
| 6. Tuesday's details for specified user   |                                                   |
| 7. Wednesday's details for specified user |                                                   |
| 8. Thursday's details for specified user  |                                                   |
| 9. Friday's details for specified user    | Editable only in Admin mode                       |
| 10. Record navigation control             |                                                   |
| 11. Add record section                    |                                                   |
| 12. Clear record section                  |                                                   |
| 13. Edit record section                   |                                                   |
| 14. Status/ Message display               |                                                   |
| 15. Sign in/out log                       |                                                   |
| 16. Attendance table                      |                                                   |
| 17. General control buttons               |                                                   |

Cytron Technologies : Fingerprint Attendance Application

COM3  
19200

Open Port  
Close Port

**5:31:34 PM**  
**Wednesday, August 11, 2010**

ID: **001**  
User's Name: **Mr. 1** Record Number: **1**

MONDAY Sign in: Sign out: Attendance:  
TUESDAY Sign in: Sign out: Attendance:  
WEDNESDAY 11/08/2010 Sign in: 8:50:18 AM Sign out: 5:31:00 PM Attendance: 1  
THURSDAY Sign in: Sign out: Attendance:  
FRIDAY Sign in: Sign out: Attendance:

Mark  
Clear All Attendance  
Exit Admin Mode  
Reset  
Exit

Add Record Clear Record Edit Record  
5 Choose a Record... Choose a Record...  
\*ID: Add Clear \*ID: Update  
\*User's Name: \*User's Name:

**Good Bye Mr. 1 ( ID = 001 ) your attendance is marked**

| ID  | Name  | Sign In Time | Sign Out Time |
|-----|-------|--------------|---------------|
| 004 | Mr. 4 | 8:55:54 AM   |               |
| 003 | Mr. 3 | 8:59:18 AM   |               |
| 002 | Mr. 2 |              | 5:30:43 PM    |
| 003 | Mr. 3 |              | 5:30:49 PM    |
| 005 | Mr. 5 |              | 5:30:54 PM    |
| 001 | Mr. 1 |              | 5:31:00 PM    |

| ID  | Name  | Mon | Tue | Wed | Thur | Fri |
|-----|-------|-----|-----|-----|------|-----|
| 001 | Mr. 1 |     |     | 1   |      |     |
| 002 | Mr. 2 |     |     | 1   |      |     |
| 003 | Mr. 3 |     |     | 1   |      |     |
| 004 | Mr. 4 |     |     |     |      |     |
| 005 | Mr. 5 |     |     | 1   |      |     |

Connected

Figure 5.2 Sample Screenshot

## **6. WARRANTY**

- Product warranty is valid for 6 months
- Warranty only applies to manufacturing defect.
- Damage caused by misuse is not covered under warranty.
- Warranty does not cover freight cost for both ways.

*Prepared by*  
***Cytron Technologies Sdn. Bhd.***  
19, Jalan Kebudayaan 1A,  
Taman Universiti,  
81300 Skudai,  
Johor, Malaysia.

*Tel:* +607-521 3178

*Fax:* +607-521 1861

*URL:* [www.cytron.com.my](http://www.cytron.com.my)

*Email:* [support@cytron.com.my](mailto:support@cytron.com.my)

[sales@cytron.com.my](mailto:sales@cytron.com.my)