

Fingerprint Identification Module
- YE-UL1701B -
User Manual & Operation Guide

YANEN 研恩科技

www.yanen.com.tw

Table of Contents

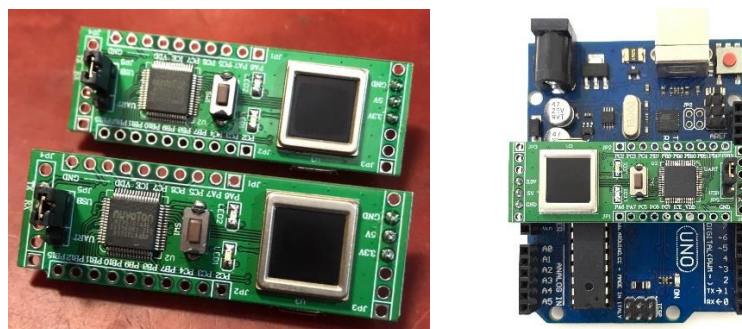
1. INTRODUCTION	3
1.1. OVERVIEW	3
1.2. SPECIFICATION.....	4
1.3. FEATURE	4
1.4. HARDWARE INTERFACE.....	4
2. PROTOCOL	5
3. LED LIGHT DISPLAY INSTRUCTIONS	6
3.1. CONNECTION MODE	6
3.2. STANDALONE MODE	6
4. OPERATION EXAMPLE	7
4.1. CONNECTION TO ARDUINO	7
4.1.1. Hardware Installation	7
4.1.2. Arduino Source Code	8
4.1.3. Arduino Command.....	8
4.2. CONNECTION TO RASPBERRY PI	12
4.2.1. Hardware Installation	12
4.2.2. Raspberry Pi Source Code	12
4.2.3. Raspberry Pi Command.....	12
4.3. CONNECTION TO PC SIDE	14
4.3.1. Hardware installation.....	14
4.3.2. Tool installation.....	15
4.3.3. Tool operation instruction.....	15
4.3.4. Operation of Quickick Tool.....	15

1. Introduction

This document is the operation manual of the "Fingerprint Evaluation Board (YE-UL1701B)". The following instructions are called YE-UL1701B. The contents of the document contain the YE-UL1701B introduction, specifications, communication protocols, operating instructions and usage examples in different environments. The operating examples in this manual also provide test tools and source code. The YE-UL1701B is a high-performance fingerprint identification module that provides product developers with the need to develop fingerprint identification requirements (application areas such as fingerprint locks, safes, industrial controls, licensing, finance, etc.). In addition, the YE-UL1701B can be used in development kits such as Arduino, Raspberry Pi, FPGA, ARM, pcDuino, etc. in microprocessor control boards with UART (Serial Port) interface. This document provides the user with a complete description of the steps to easily complete the operation.

1.1. Overview

- YE-UL1701B is a fingerprint identification sensor. The sensor consists of high-speed processor, high-performance fingerprint algorithm and FLASH chip.
- The sensor has stable performance and simple structure. The functions include fingerprint comparison, image scanning transmission, search, fingerprint storage and system unique internal code protection mechanism.
 - The fingerprint matching display program can register 24 fingerprints for comparison, and the comparison speed is fast and the correct rate is high, which can provide developers to develop various applications.
 - The system-specific XOM mechanism protects the code from "execution" and cannot be "read" to protect important intellectual property code.
- Functional characteristics: Independently complete fingerprint acquisition, fingerprint storage, fingerprint comparison (1:1) and fingerprint search (1: N) functions.
- Small size: The product size is small, and the built-in fingerprint algorithm chip board is convenient for installation.
- Low power consumption: Low power consumption and suitable for most applications.
- Simple application development: Developers can develop fingerprint-related products based on control instructions without the need for professional fingerprint identification knowledge.



YE-UL1701B Fingerprint Sensor

1.2. Specification

• EVB Size : 53.3x17.8x1.6(mm)	• MCU : Nuvoton M2351, ARM-M0 Plus, 32-bit, 64 MHz
• Sensor Module : A172	• FAR : <0.001% (Security Level, 3)
• Active area : 176*176pixels	• FRR : <0.1% (Security Level, 3)
• Resolution : 508 Dpi	• Operating environment : -20 to +60°C / <RH 90%
• Sample rate : 40 frames/s	• Storage environment : -40 to +70°C / <RH 90%
• Matching mode : 1:1, 1:N	• Communication interface : UART
• Matching time : < 0.3s	• Power : UART 3.3V or USB
• Enrollment time : < 0.2s	

1.3. Feature

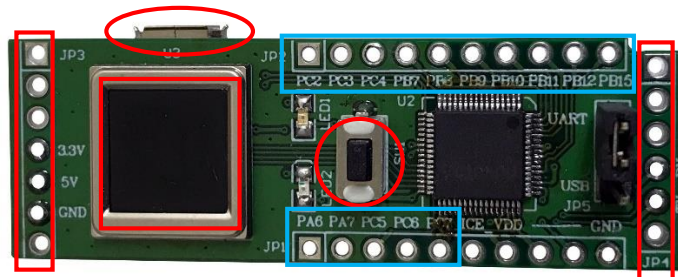
- USB 2.0 full speed device
- LED and push buttons
- Fingerprint sensor
- Crypto acceleraor
- Emulating drivers
 - HID protocol
 - Storage class
 - Biometric class

1.4. Hardware Interface

The communication interface of the YE-UL1701B module is UART. The YE-UL1701B hardware description and wiring diagram are shown below. The oval red frame at the top left of the figure is the USB interface, the square red frame is the fingerprint sensor, the middle circular red frame is the reset button, the left and right rectangular red frame has the UART interface, and the UART pin is VCC (3.3V), GND, RX, and TX are defined as follows. Another 15 GPIOs are available for users (blue frame).

When the YE-UL1701B module communicates with user device, definition of pin is as follows:

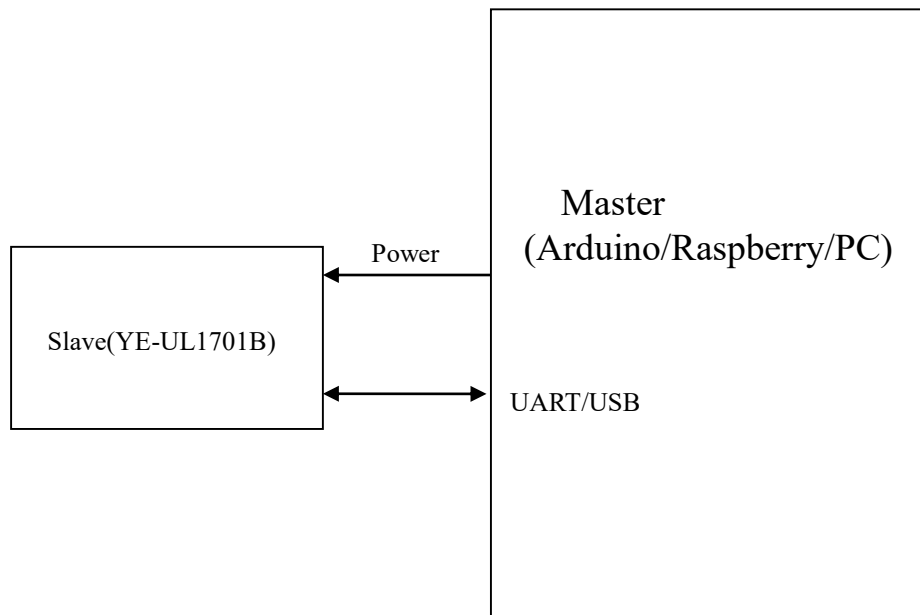
Name	Type	Description
VCC	In	Power input (3.3V)
GND	--	Signal ground. Connected to power ground
RX	In	Data input.
TX	Out	Data output.



YE-UL1701B hardware description and wiring diagram

2. Protocol

The communication protocol between the YE-UL1701B and the connected control board. The control board (such as Arduino...) is the master (Master) and the YE-UL1701B is the slave (Slave). Slave are powered by UART and USB.



The channel used by this communication protocol is UART.

The initial transfer rate of the UART is 9600, N, 8, and 1. This setting can be changed by command.

For the contents of the communication protocol, please refer to the **YE-EVB Communication Protocol** document.

3. Led light display instructions

The LED light of the YE-UL1701B shows the current working status.

After the YE-UL1701B is powered or reset, if the red LED and the green LED flash in turn, it means that the operation is in the connection mode. If the red LED or the green LED is blinking alone, it means that the work is in the standalone mode.

3.1. Connection mode

After power supply or reset, if the red LED and the green LED flash in turn, it means that it is receiving the command status, and the master (Arduino or Raspberry PI) can command it to perform related functions.

The LED light display shows the following.

The red and green LEDs flash in turn:

Indicates the status of the received command and waits for the command from the master.

The red LED and the green LED are ON (light) at the same time:

In the reminder state, it will enter the registration state after two or three seconds.

The red LED flashes:

In the registration state, waiting for the fingerprint to press to register, if you register two fingerprints, each fingerprint needs to be pressed 3 times. After each successful pressing of the fingerprint, the green LED will light for two or three seconds.

The green LED flashes:

In the input fingerprint state, wait for the fingerprint to press to scan the fingerprint image to compare with the registered fingerprint or upload the fingerprint image. If the fingerprint is compared, the green LED is on for two or three seconds when the comparison is successful, and the red LED is on for two or three seconds when the comparison fails. If the fingerprint image is uploaded, the red LED and the green LED alternately flash twice during image uploading.

The red LED and the green LED flash twice in turn:

Indicates the fingerprint image upload status.

3.2. Standalone mode

After power or reset, if the red LED or the green LED is flashing alone, it means that it is working in standalone mode. The LED lights are described as follows.

The red LED and the green LED are ON (light) at the same time:

In the reminder state, it will enter the registration state after two or three seconds.

The red LED flashes:

In the registration state, waiting for the fingerprint to press to register, if you register two fingerprints, each fingerprint needs to be pressed 3 times. After each successful pressing of the fingerprint, the green LED will light for two or three seconds.

The green LED flashes:

In the comparison state, waiting for the fingerprint to press and compare with the registered fingerprint. If the comparison is successful, the green LED is on for two or three seconds, and if the comparison fails, the red LED is on for two or three seconds. If the continuous comparison fails 3 times, it will enter the reminder state.

4. Operation Example

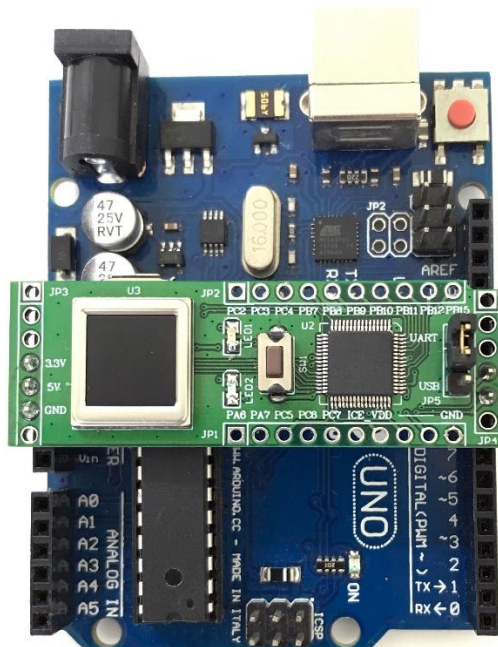
Example of connection test environment in Arduino, Raspberry Pi and PC as follow.

4.1. Connection to Arduino

Example operation based on Arduino UNO.

4.1.1. Hardware Installation

The cable line connection between YE-UL1701B and Arduino UNO is as follows.



Compare	Arduino	YE-UL1701B
	3.3V	3.3V
	GND	GND
	GPIO10	RX
	GPIO9	TX

4.1.2. Arduino Source Code

The source code please refer to Fingerprint_Demo.ino file.

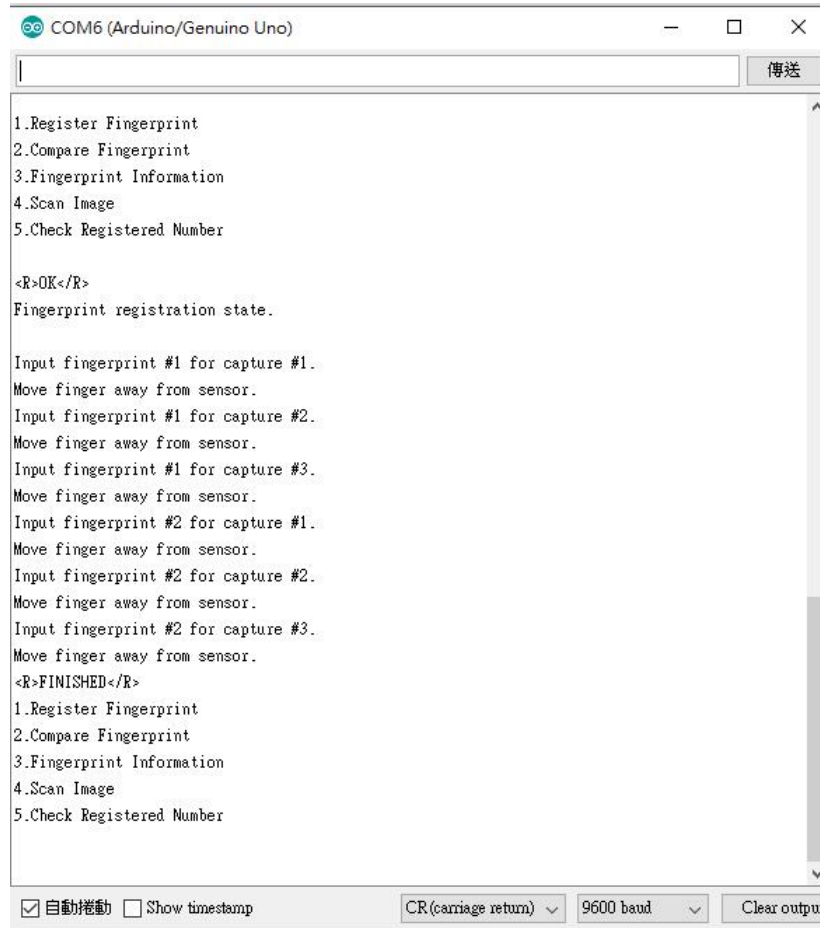
4.1.3. Arduino Command

After executing Fingerprint_Demo.ino in the Arduino IDE, the following items will appear on the screen.

```
=====
1. Register Fingerprint
2. Compare Fingerprint
3. Query Fingerprint Image Information
4. Scan Fingerprint Image
5. Check Registered Fingerprint Number
=====
```

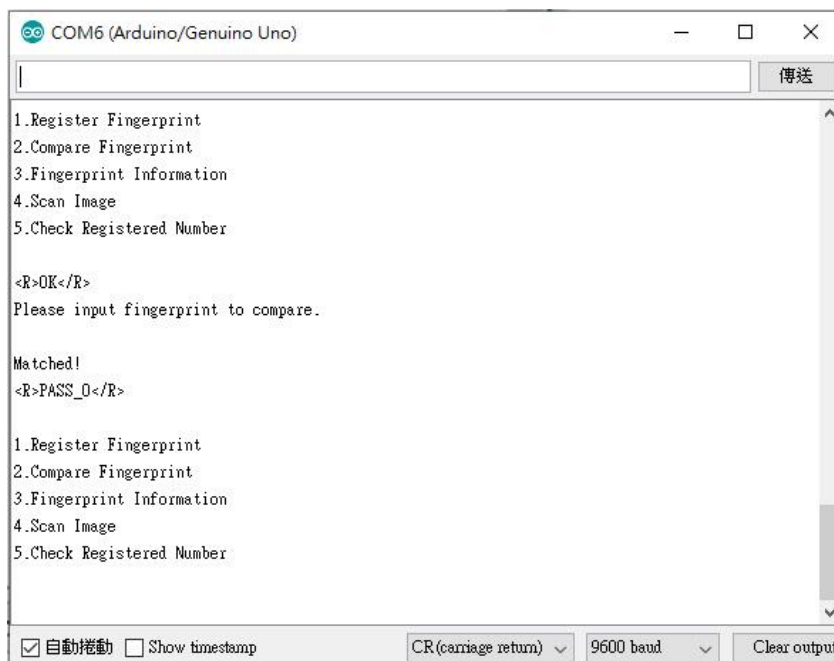
1) Register Fingerprint :

Register fingerprint (enter number 1), the user must register 24 fingerprints in the fingerprint sensing area, and each fingerprint registration must be pressed 3 times. When the fingerprint registration is completed and the number of registered fingerprints is less than 24, it means that the registration is interrupted after registering to N pieces. (When each press is successful, the light will turn green.). After successful execution, the response is as follows :



2) Compare Fingerprint :

Register fingerprint (enter number 2), When the user presses the fingerprint in the fingerprint sensing area, the matching error or the correct message will be displayed. After successful execution, the response is as follows :



Comparison error (error fingerprint)

<R>OK</R>

Please input fingerprint to compare.

Unmatched!

<R>FAIL</R>

or

Correct (correct fingerprint)

<R>OK</R>

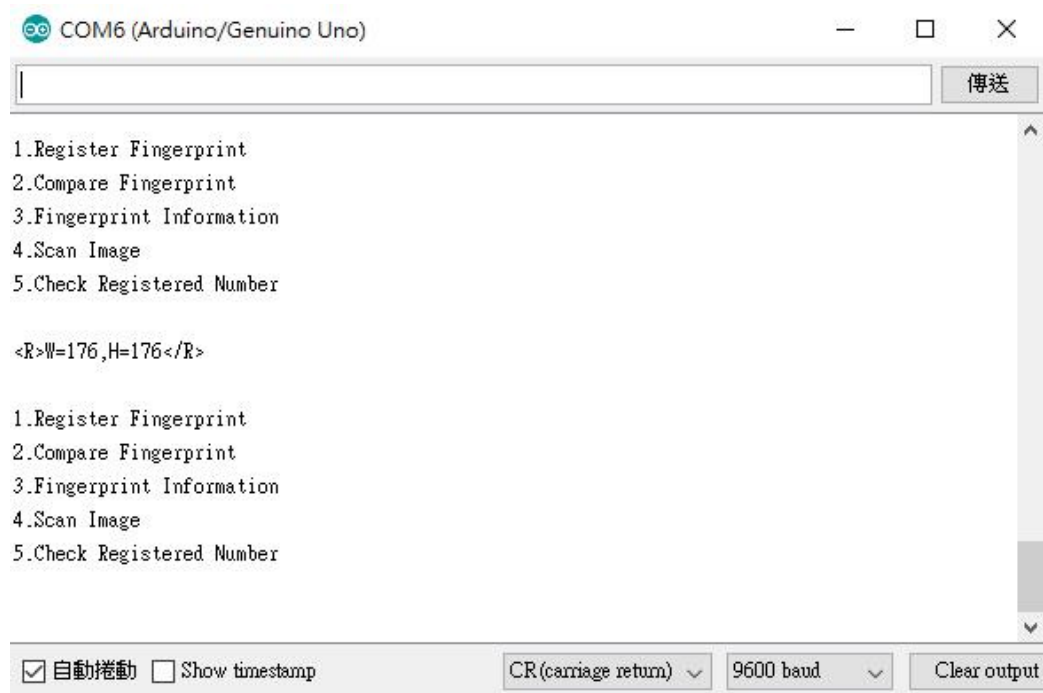
Please input fingerprint to compare.

Matched!

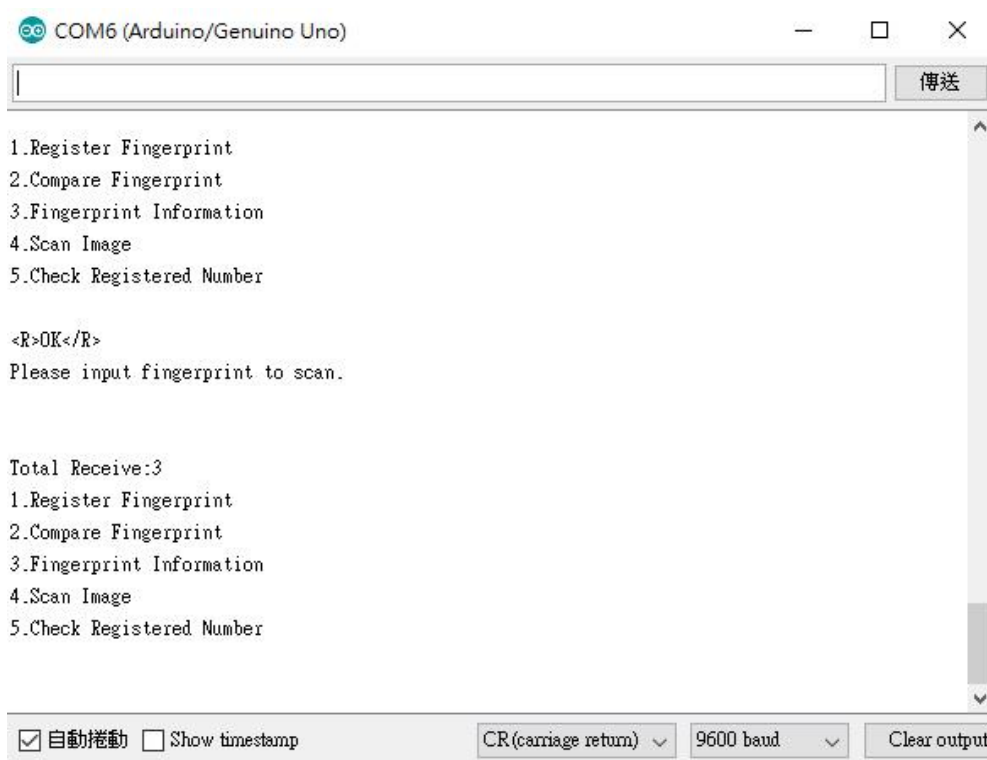
<R>PASS</R>

3) Query Fingerprint Image Information :

Register fingerprint (enter number 3), After successful execution, the response is as follows :

**4) Scan Fingerprint Image :**

Register fingerprint (enter number 4), The user must press the fingerprint in the fingerprint sensing area After successful execution, the response is as follows.



(Note: Most of the built-in SRAM in Arduino is less than 30976 Bits (30KBytes), so all image data cannot be displayed. Users should confirm the hardware specifications used.)

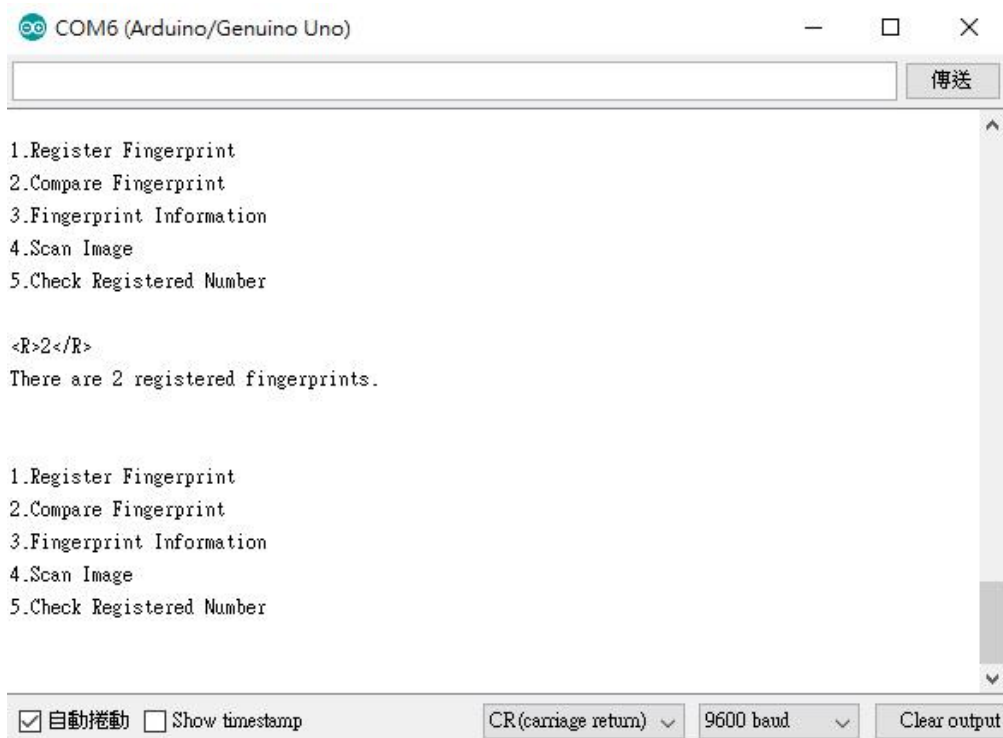
5) Check Registered Fingerprint Number :

Register fingerprint (enter number 5),

Slave responds with <R> and </R> to encapsulate the number, for example:

<R>0<R> Indicates no registration data.

<R>2<R> Indicates that there are 2 registration data.

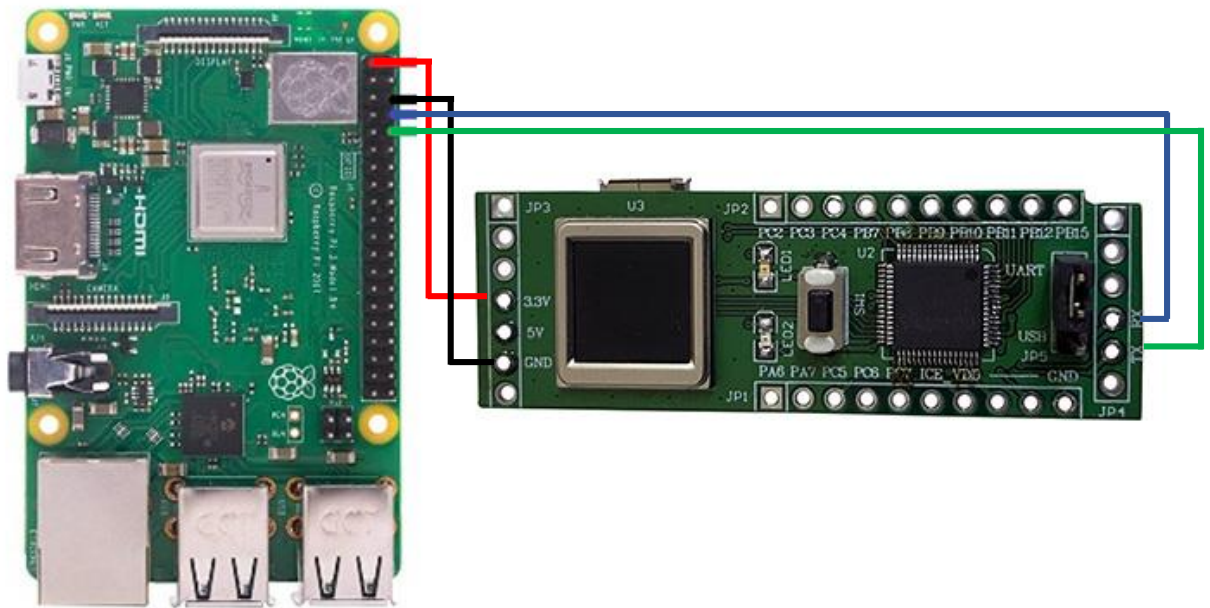


4.2. Connection to Raspberry Pi

Example operation based on Raspberry Pi 3 B+.

4.2.1. Hardware Installation

The cable line connection between YE-UL1701B and Raspberry Pi is as follows.



Compare	Raspberry Pi	YE-UL1701B
Cable color: Red	3.3V	Vcc
Cable color: Black	GND	GND
Cable color: Blue	GPIO14	RX
Cable color: Green	GPIO15	TX

4.2.2. Raspberry Pi Source Code

Test version: Python 3.5.3

The source code please refer to Fingerprint_Demo.py file.

4.2.3. Raspberry Pi Command

After executing Fingerprint_Demo.py in the Python environment, the following items will appear on the screen.

=====

1. Register Fingerprint
2. Compare Fingerprint
3. Query Fingerprint Image Information

-
4. Scan Fingerprint Image
 5. Check Registered Fingerprint Number
 0. End

Please enter the function code:

=====

1) Register Fingerprint :

Register fingerprint (enter number 1), the user must register 24 fingerprints in the fingerprint sensing area, and each fingerprint registration must be pressed 3 times. When the fingerprint registration is completed and the number of registered fingerprints is less than 24, it means that the registration is interrupted after registering to N pieces. (When each press is successful, the light will turn green.). After successful execution, the response is as follows :

<R>OK</R>

Fingerprint registration state.

Input fingerprint #1 for capture #1.

Move finger away from sensor.

Input fingerprint #1 for capture #2.

Move finger away from sensor.

Input fingerprint #1 for capture #3.

Move finger away from sensor.

Input fingerprint #2 for capture #1.

Move finger away from sensor.

Input fingerprint #2 for capture #2.

Move finger away from sensor.

Input fingerprint #2 for capture #3.

Move finger away from sensor.

<R>FINISHED</R>

Registration is complete

2) Compare Fingerprint :

Register fingerprint (enter number 2), When the user presses the fingerprint in the fingerprint sensing area, the matching error or the correct message will be displayed. After successful execution, the response is as follows :

Comparison error (error fingerprint)

<R>OK</R>

Please input fingerprint to compare.

Unmatched!

<R>FAIL</R>

or

Correct (correct fingerprint)

<R>OK</R>

Please input fingerprint to compare.

Matched!

<R>PASS</R>

3) Query Fingerprint Image Information :

Register fingerprint (enter number 3), After successful execution, the response is as follows :

<R>W=176, H=176</R>

4) Scan Fingerprint Image :

Register fingerprint (enter number 4), The user must press the fingerprint in the fingerprint sensing area After successful execution, the response is as follows.

Please input fingerprint to scan.

Receiving data

Total Bytes=30983

Data reception completed

Please enter the storage path and file name, such as /home/pi/Finger/Image1.pgm, and the fingerprint image will be stored in the specified path.

5) Check Registered Fingerprint Number :

Register fingerprint (enter number 5), Slave responds with <R> and </R> to encapsulate the number, for example:

<R>0<R> Indicates no registration data.

<R>2<R> Indicates that there are 2 registration data.

4.3. Connection to PC side

Note: In this test case, the user needs to have a USB to UART (or USB to TTL) line.

4.3.1. Hardware installation

Connect the USB to UART (or USB to TTL) cable to the YE-UL1701B and PC. When the USB to UART cable is connected to the YE-UL1701B, the TX/RX pin of the YE-UL1701B needs to be exchanged.

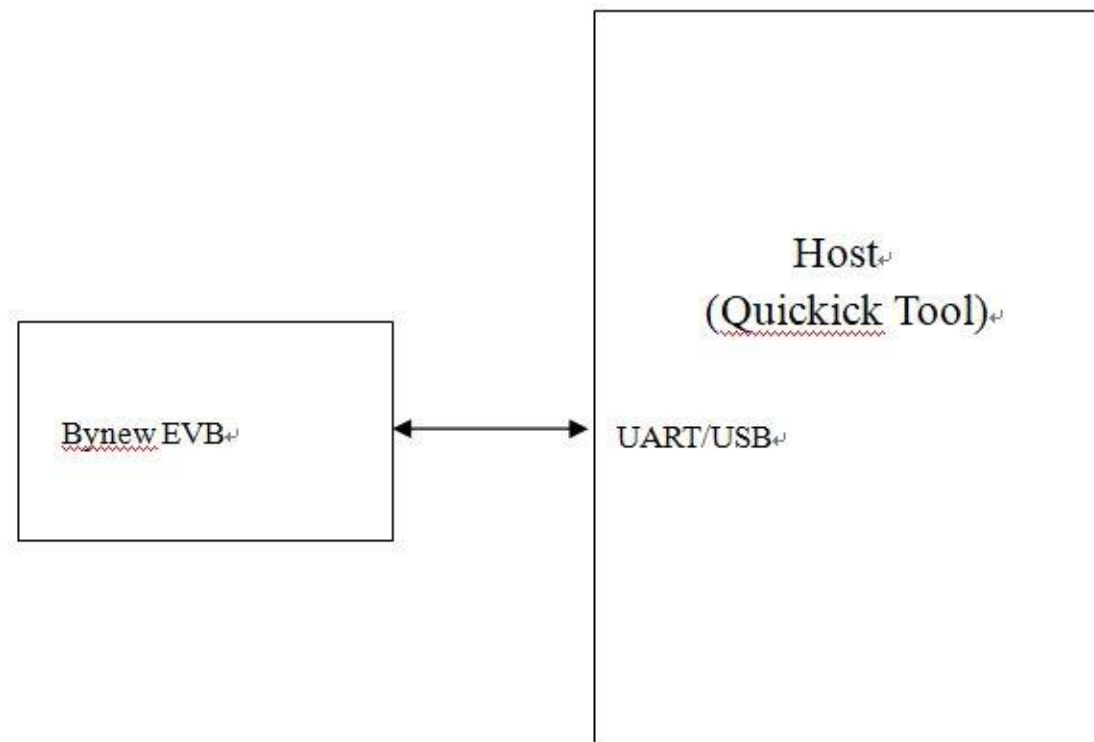
YE-UL1701B	PC side (USB to UART line)	Description
VCC (3.3V)	VCC (3.3V)	Powered by USB.
TX	RX	Exchange docking
RX	TX	Exchange docking
GND	GND	--

After power supply, the YE-UL1701B LED will flash in turn. For the LED display in the connected mode, please refer to the section: Led light display instructions.

4.3.2. Tool installation

The executable file name is BNQuickick.exe.

4.3.3. Tool operation instruction



The communication channel is UART for the present, USB will be included in the future.

4.3.4. Operation of Quickick Tool

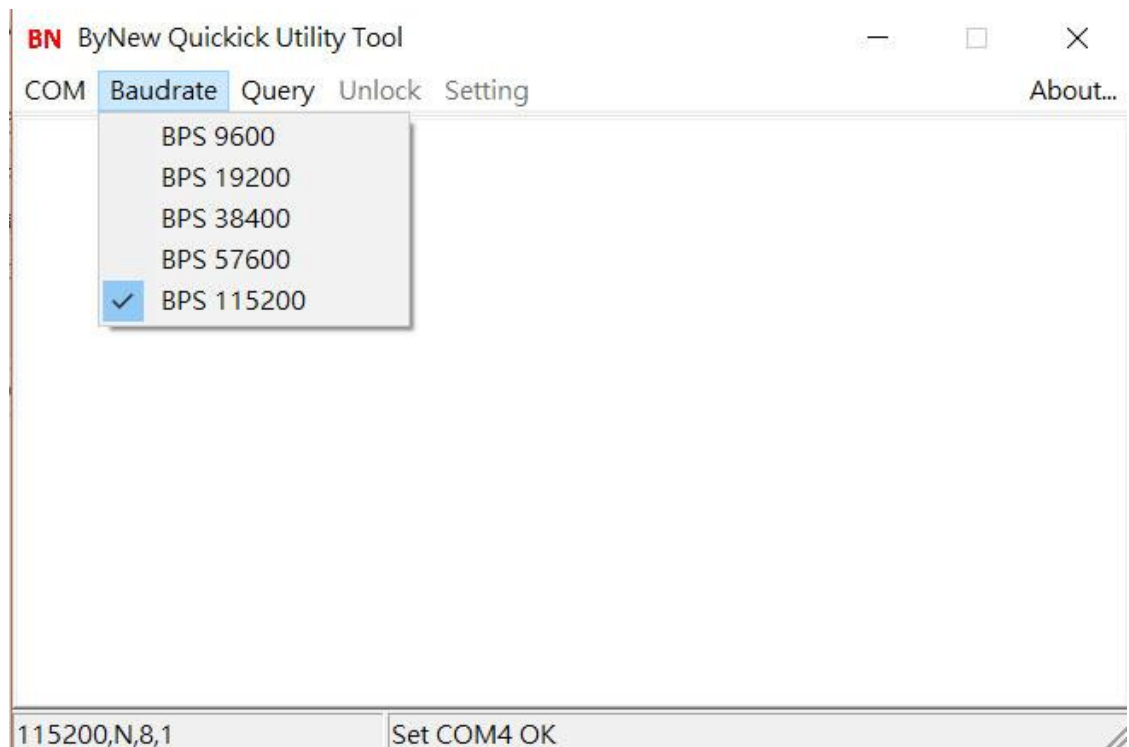
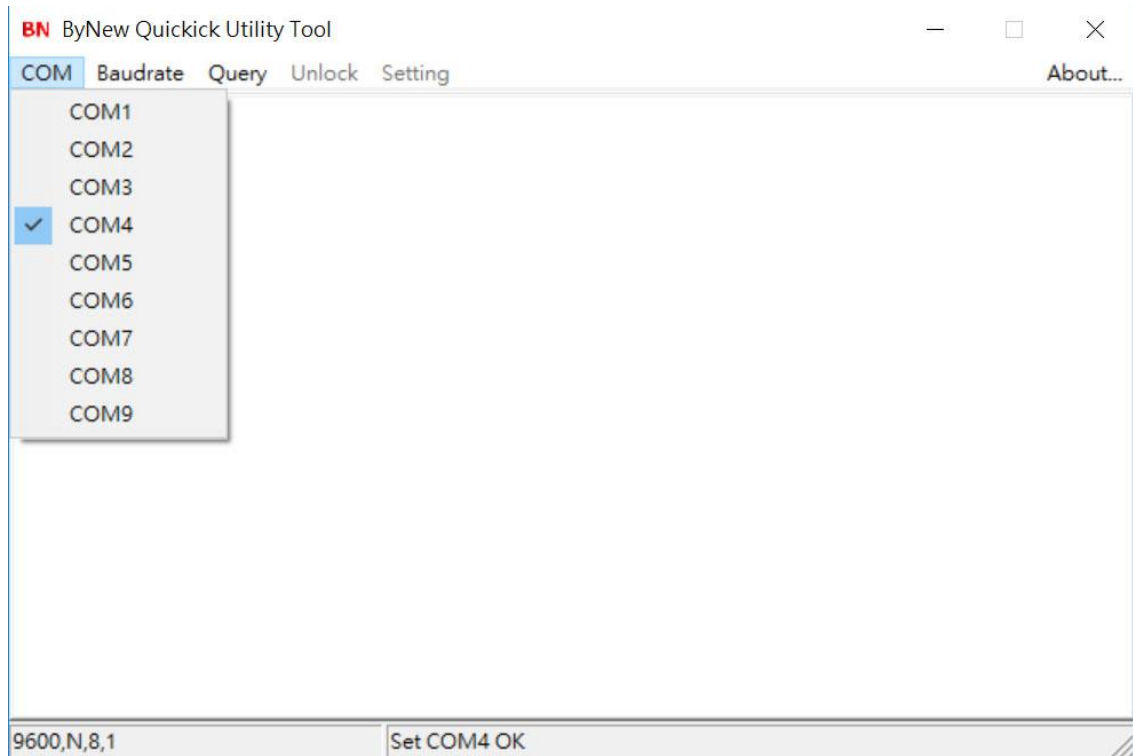
[Startup]

The Quickick tool runs on Windows, just put the BNQuickick.exe, CxImage.exe and FreeImage.dll together at the same directory.

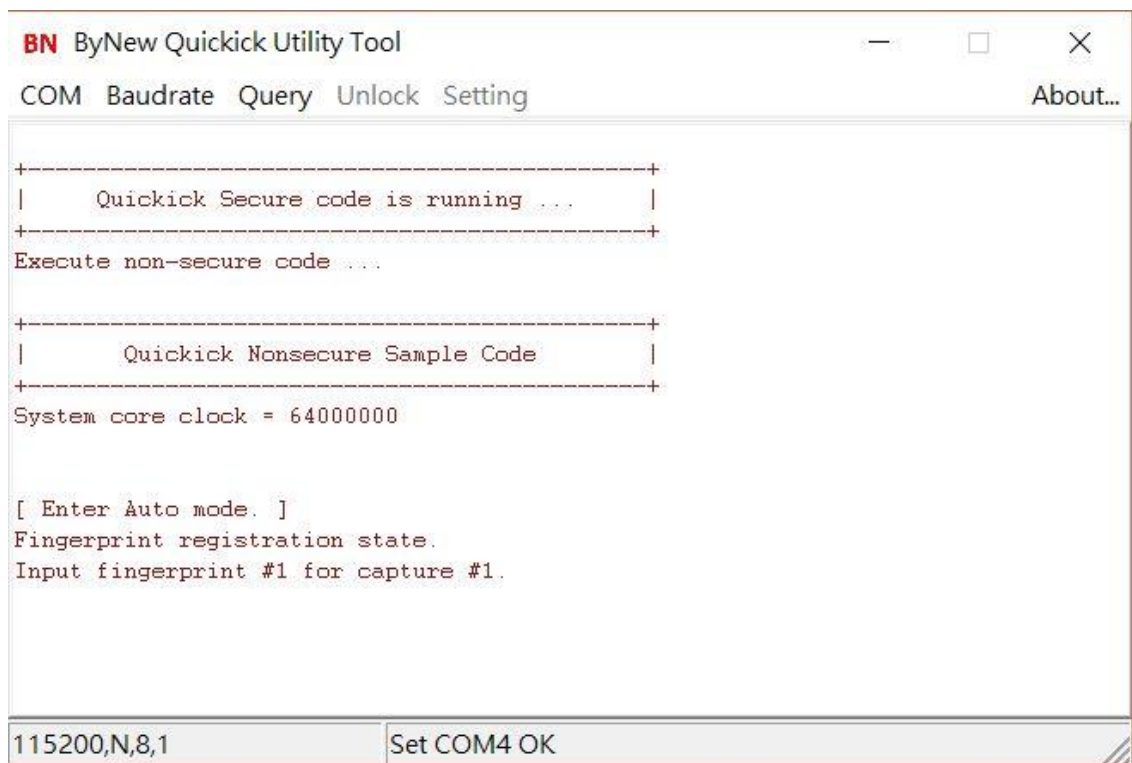
Please connect EVB UART to PC by a USB to TTL converter first, be careful that the VCC of UART must be 3.3V, the run BNQuickick.exe.

[COM and Baudrate]

After Quickkick is run, the first action is to find and set the baudrate of EVB, otherwise the communication of Quickkick and EVB cannot be setup. It has COM and Baudare menu on Quickkick as below:



Please find the com port from the Device Manager and try the baudrate from 9600 to 115200, reset EVB to see which baudrate can have startup message of EVB below:

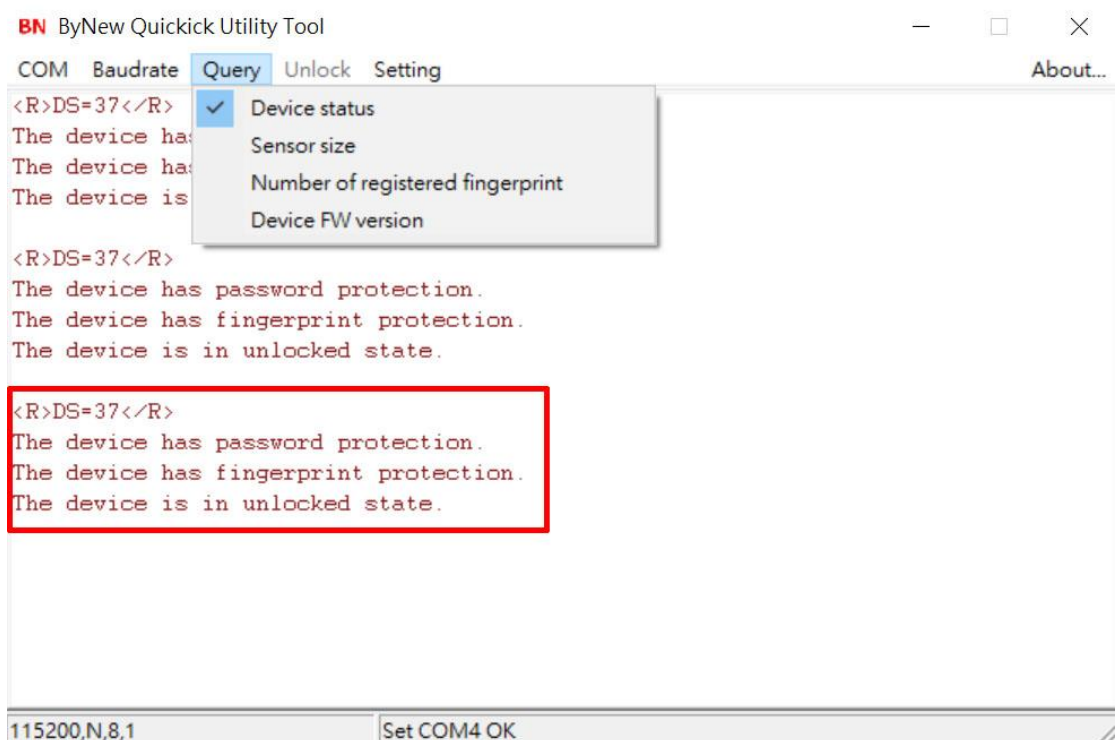


When user see the above message, the communication between Quickkick and EVB has been setup successfully, the left-bottom field displays the current baudrate and the middle-bottom field displays which com port it runs. After the communication has been setup successfully, now user can change the baudrate by click the baudrate item from Baudrate menu.

[Query]

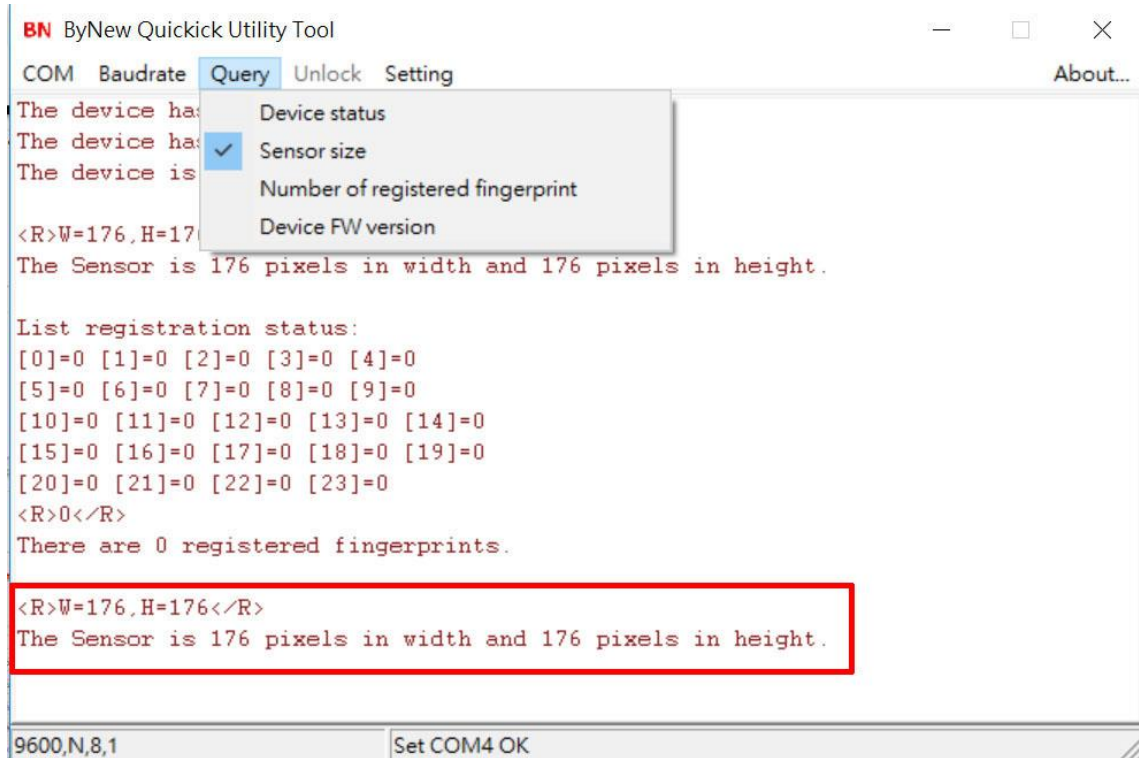
Device status

This is the first command to EVB, EVB will return it has password/fingerprint protection or not, and it is in locked or unlocked state.



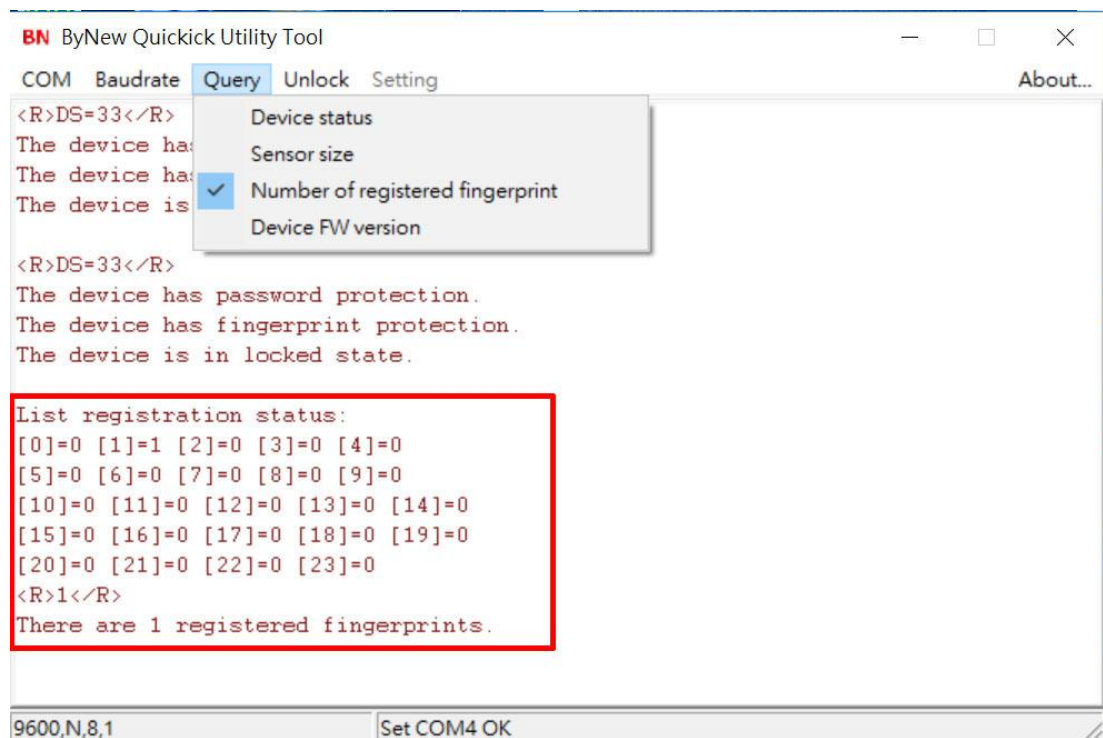
The above example shows that there is password and fingerprint protection in device (EVB), and device is in unlocked state, so Unlock menu is grayed and Setting menu is enabled for user to do setting operation.

Sensor Size



To query the image dimension of fingerprint sensor. It will show how many pixels in width and height for the fingerprint sensor.

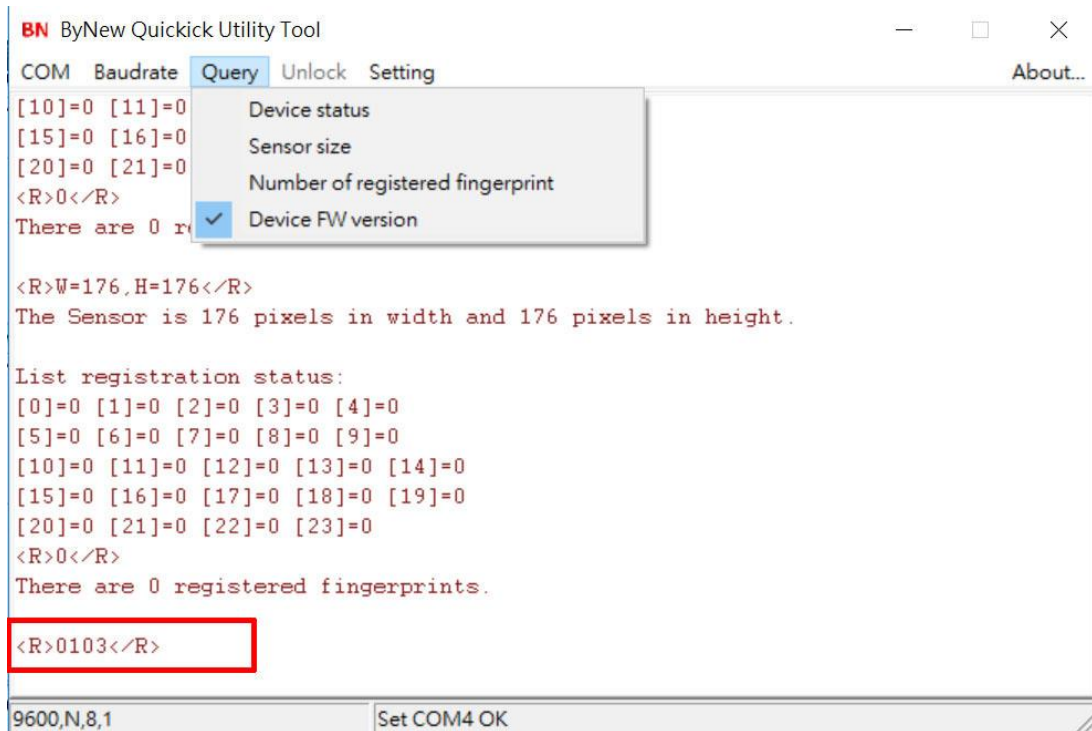
Number of registered fingerprint



It will list all registered fingerprints as above. As the above example, it shows that there is only 1

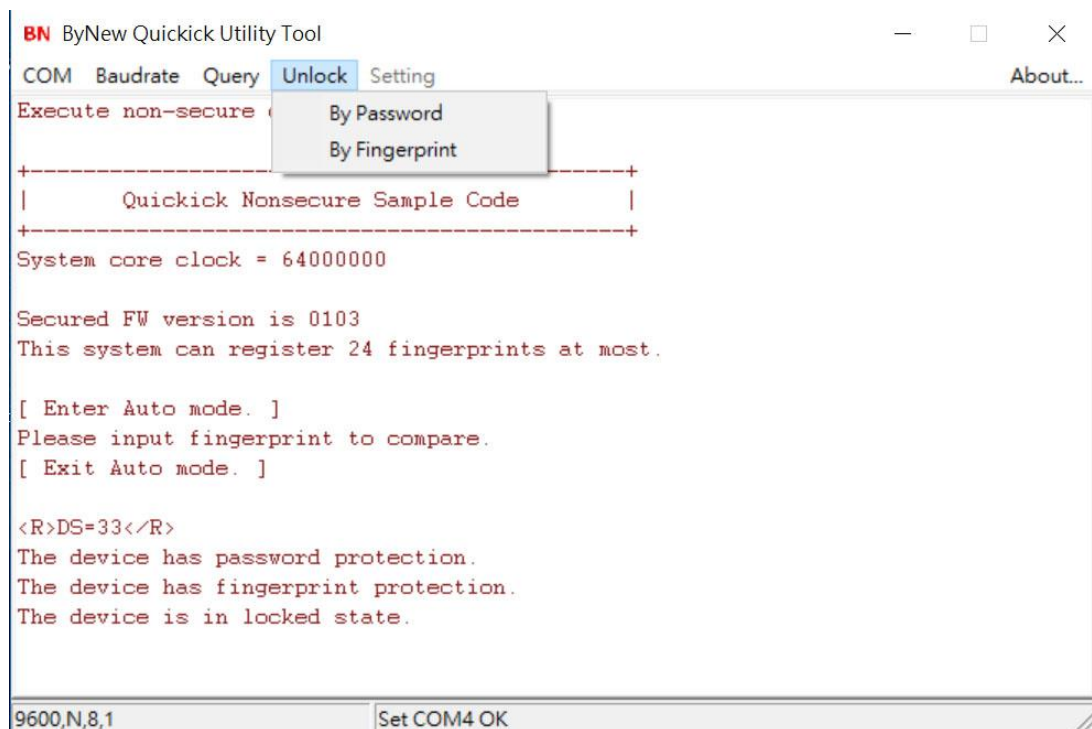
registered fingerprint, and it is registered at index 1, which is the second position (it counts from index 0). The device can have 24 registered fingerprints at max.

Device FW version



It shows the device FW version. As the above example, the FW version is 0103, which means version 1.03

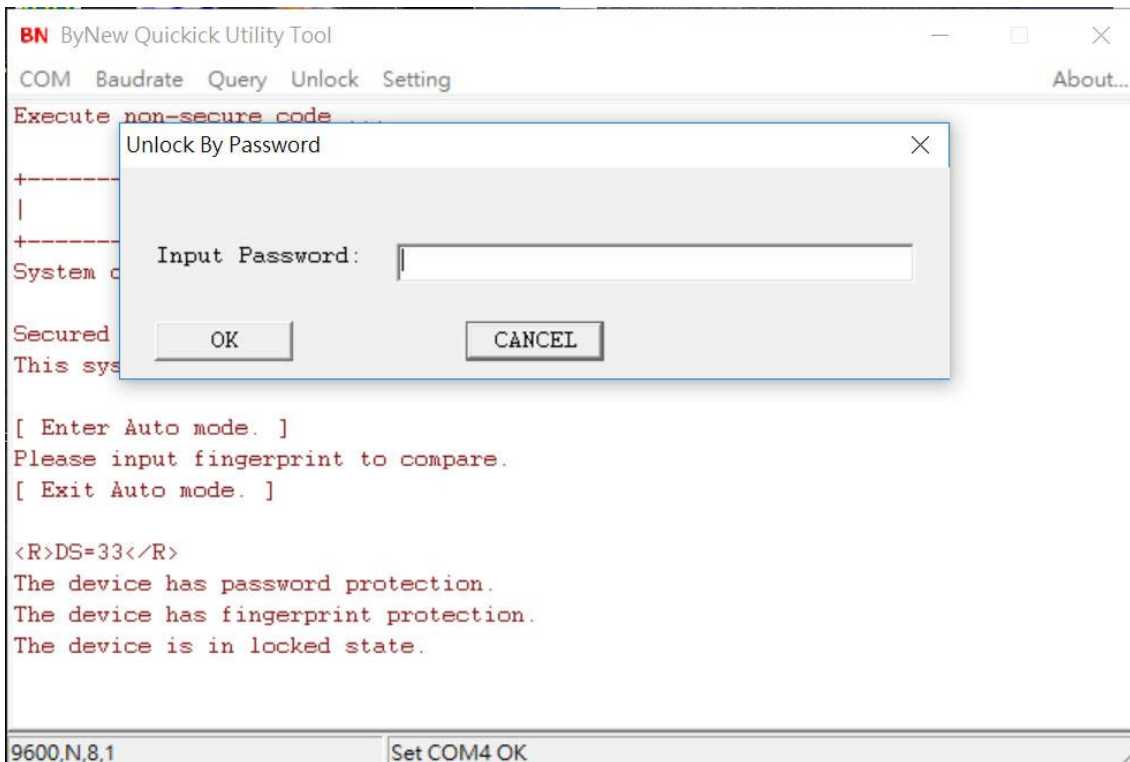
[Unlock]



When the device is in locked state, it needs to unlock it by password or fingerprint in order to do some setting to the device. There can be only 1 password be kept in device, but it can have 24

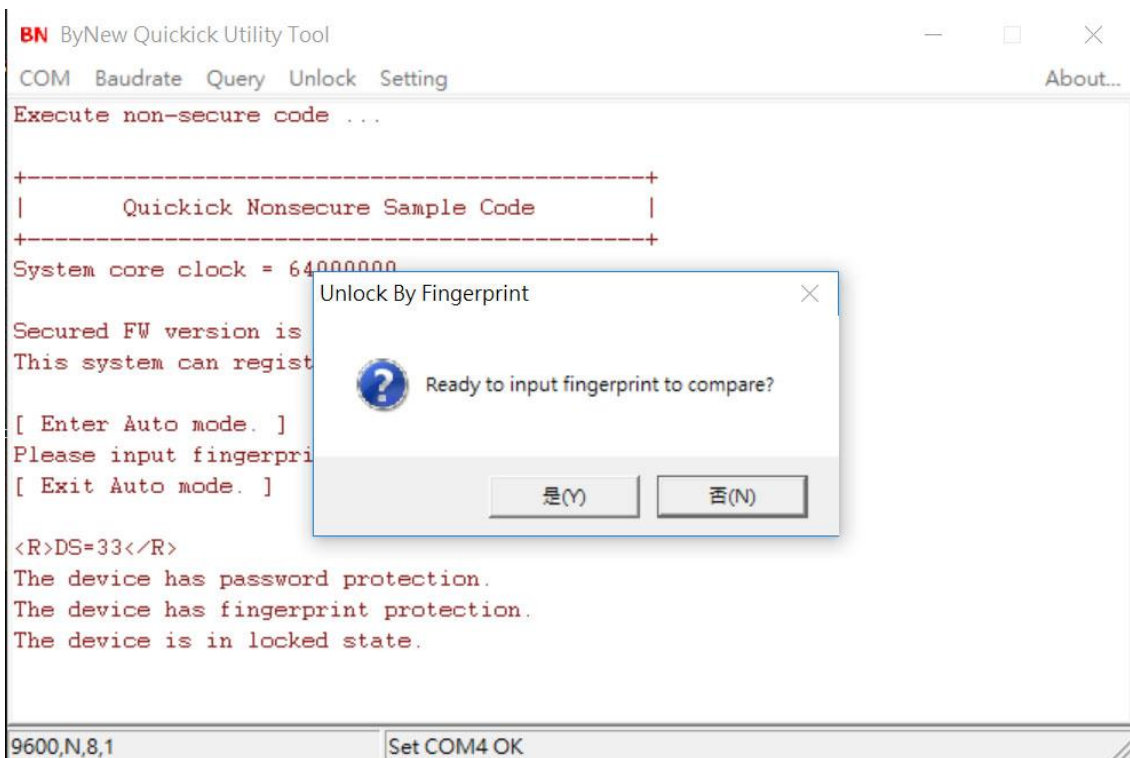
registered fingerprints at max. If user select fingerprint to unlock, it must be any of the registered fingerprint to unlock device.

#By Password

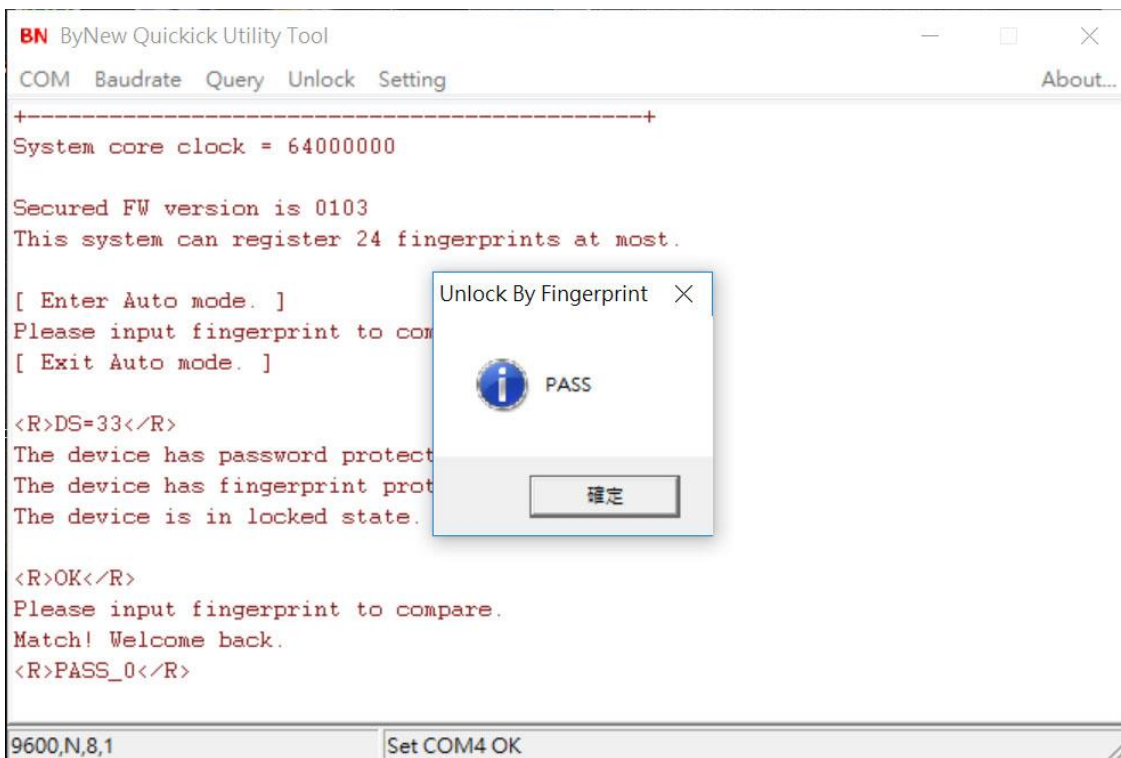


It pop up a message box for user to input password, after comparison, it will respond “Fail” or “Pass”.

#By Fingerprint



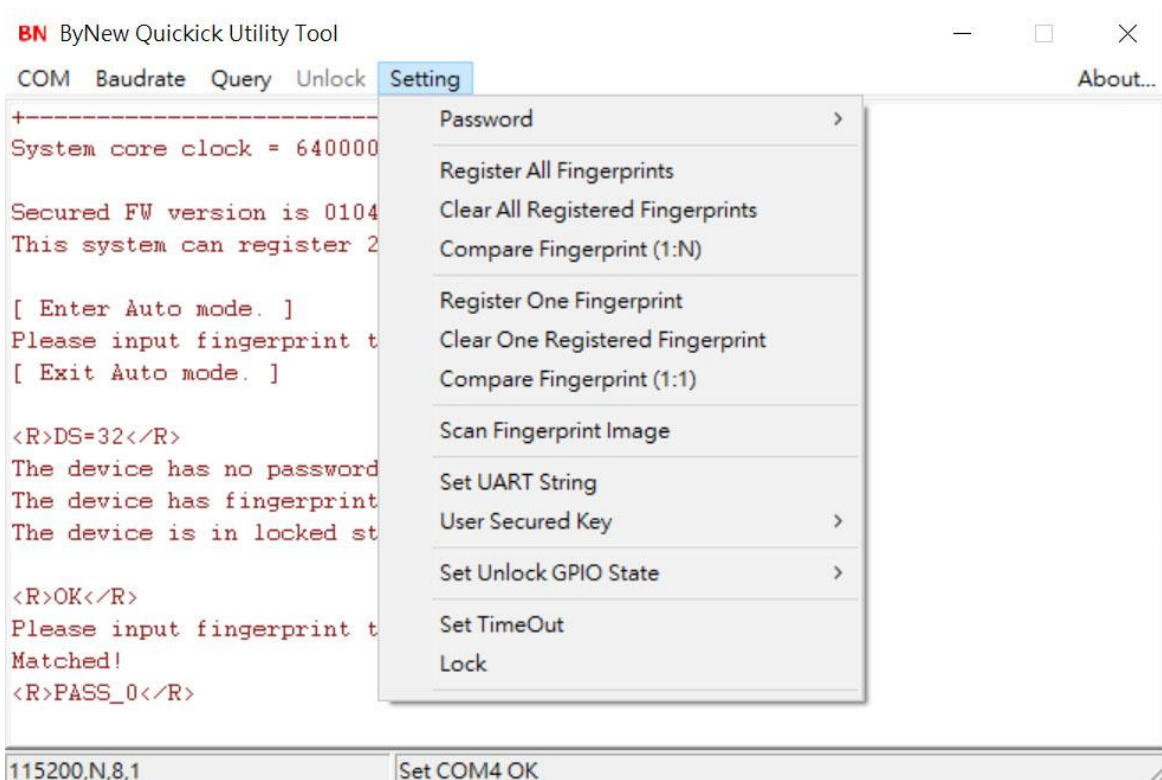
It will pop up a message box to notify user be ready to input his fingerprint, after press “Y”, it will instruct user to enroll his fingerprint to compare.



If the fingerprint is matched with any registered fingerprint, it will respond “PASS” as above, and on the screen, it will also show which registered fingerprint is matched. For the above example, it shows “PASS_0”, so it matches with the first registered fingerprint (count from index 0). After user press “CONFIRM” to the above PASS message box, the device gets into unlocked state, and the Setting menu will be enabled.

[Setting]

After device is unlocked, user could do all the setting items below:



Password

It has 3 sub-functions:

- Get Password
To get the current password saved in device.
- Set Password
To set a new password to the device, the previous one will be overwritten.
- Clear Password
To clear the current password saved in device.

Register All Fingerprints

To register fingerprint, it can register 24 fingerprints at max. User needs to enroll his fingerprint one by one to replace the previous registered one from beginning, each registration needs 3 enrolls. User follows the instructions on screen to enroll fingerprint.

User can stop this process by press Reset button on EVB at any time. For example, after register 3 fingerprints, user can press Reset button to make device only have 3 registered fingerprints.

Clear All Registered Fingerprint

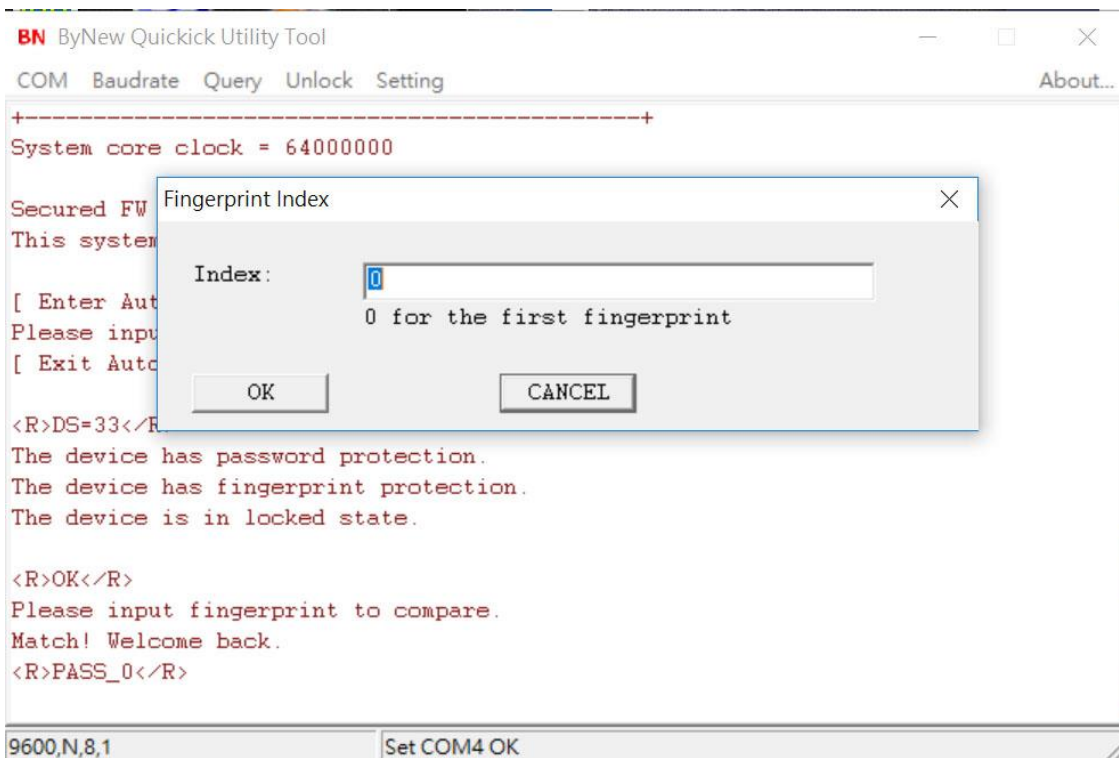
To clear all the registered fingerprint.

Compare Fingerprint (1:N)

Input fingerprint to compare with all the registered fingerprint. User follow the instruction to enroll his fingerprint. It will display compare result on screen.

Register One Fingerprint

To register only one fingerprint, it will popup a message box for user to input the index of fingerprint to register. The index is from 0 to 23.



After registration, the original registered fingerprint with the same index is overwritten.

Clear One Registered Fingerprint

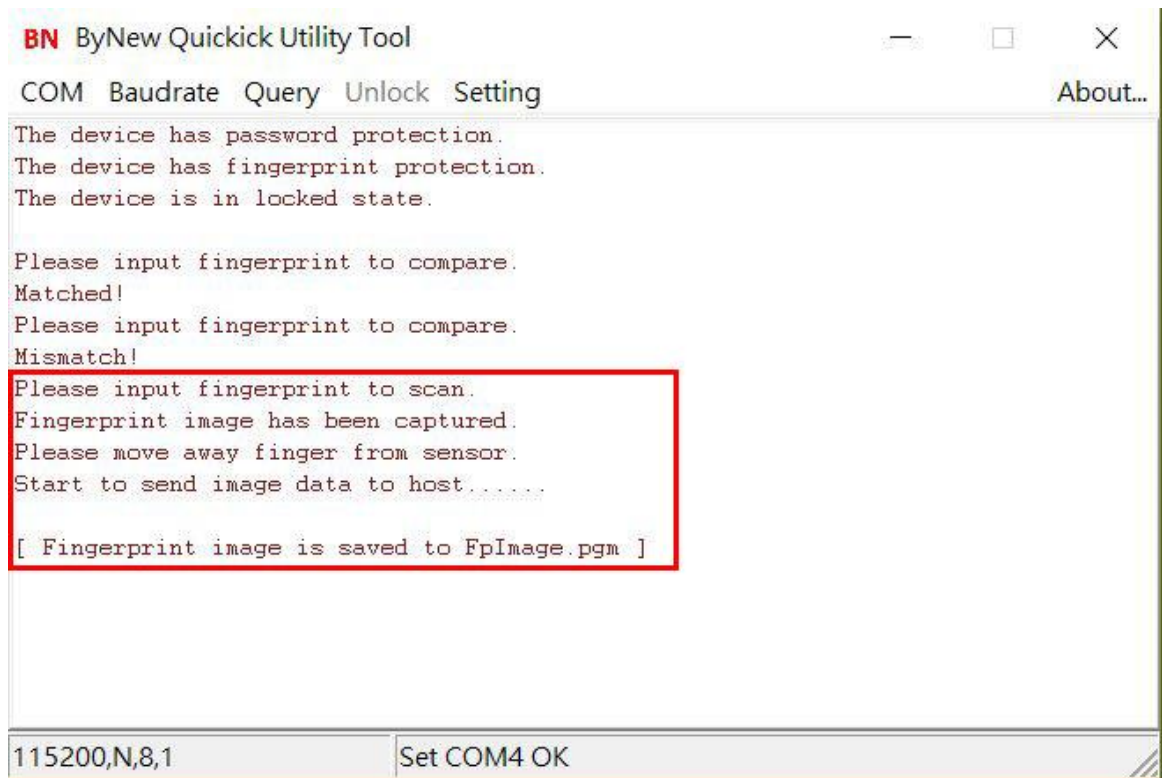
To clear some registered fingerprint, it will popup a message box for user to input the index of fingerprint to clear. The index is from 0 to 23

Compare Fingerprint (1:1)

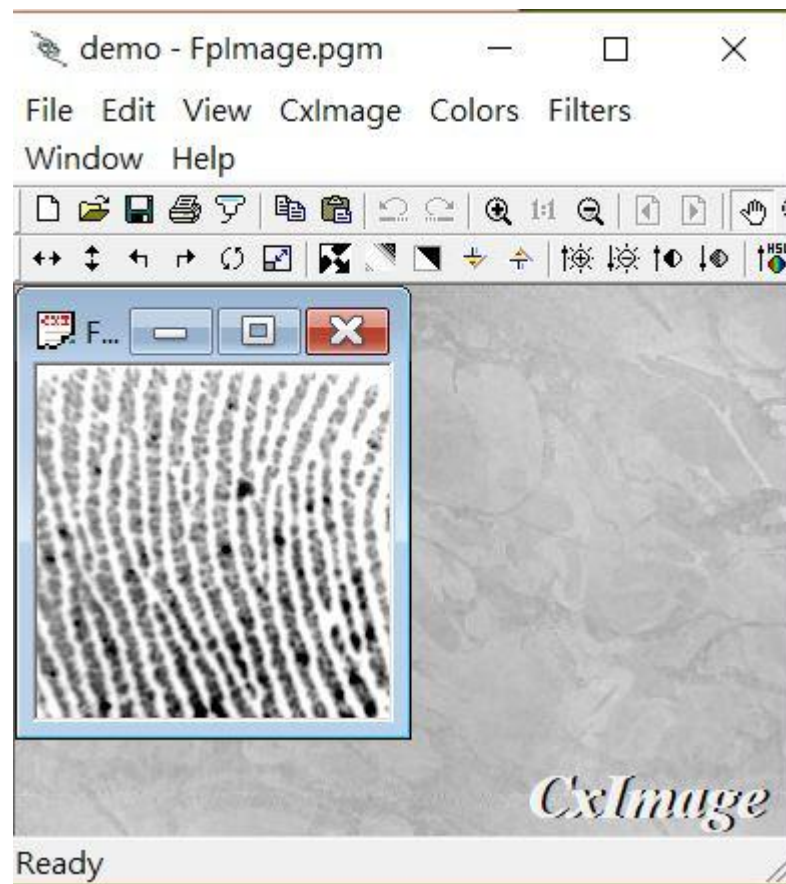
Input fingerprint to compare with a specified registered fingerprint, it will popup a message box for user to input the index of fingerprint to compare. The index is from 0 to 23..

Scan Fingerprint Image

To scan a fingerprint image to host. User follows the instruction on screen, the scanned image will be saved to a file, FpImage.pgm.

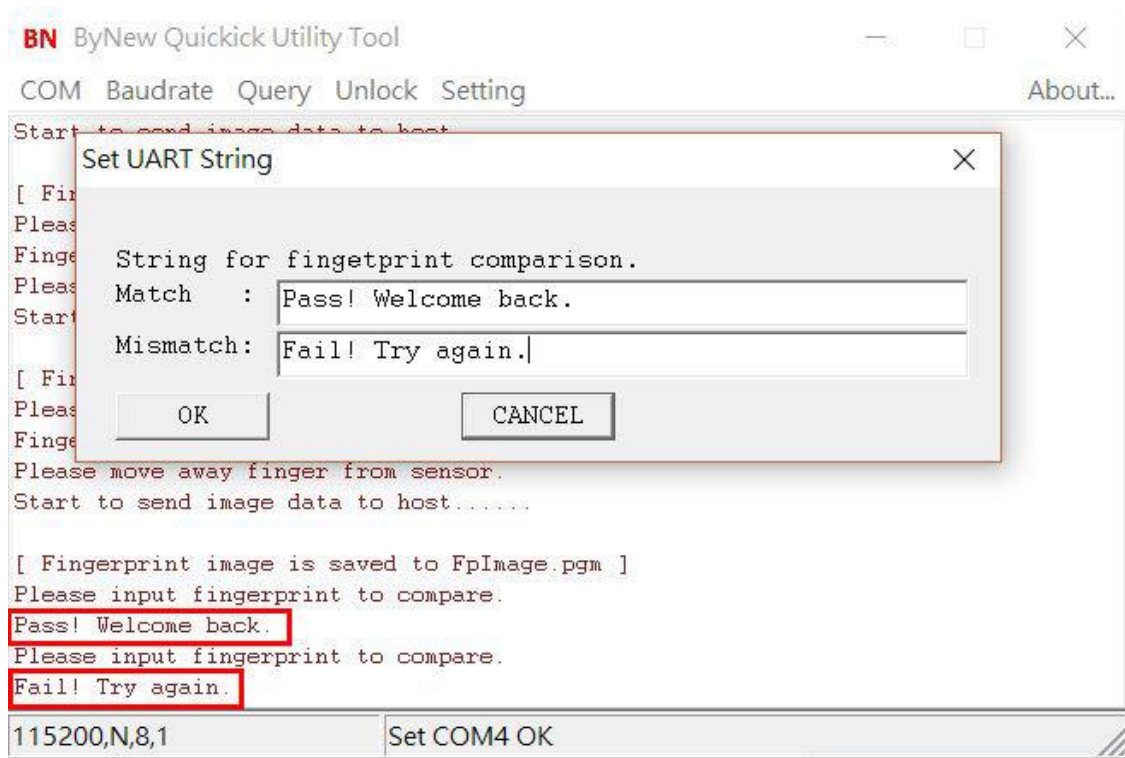


If CxImage.exe and FreeImage.dll are at the same folder with BNQuickkick.exe, it will display the scanned image automatically as below.



Set UART String

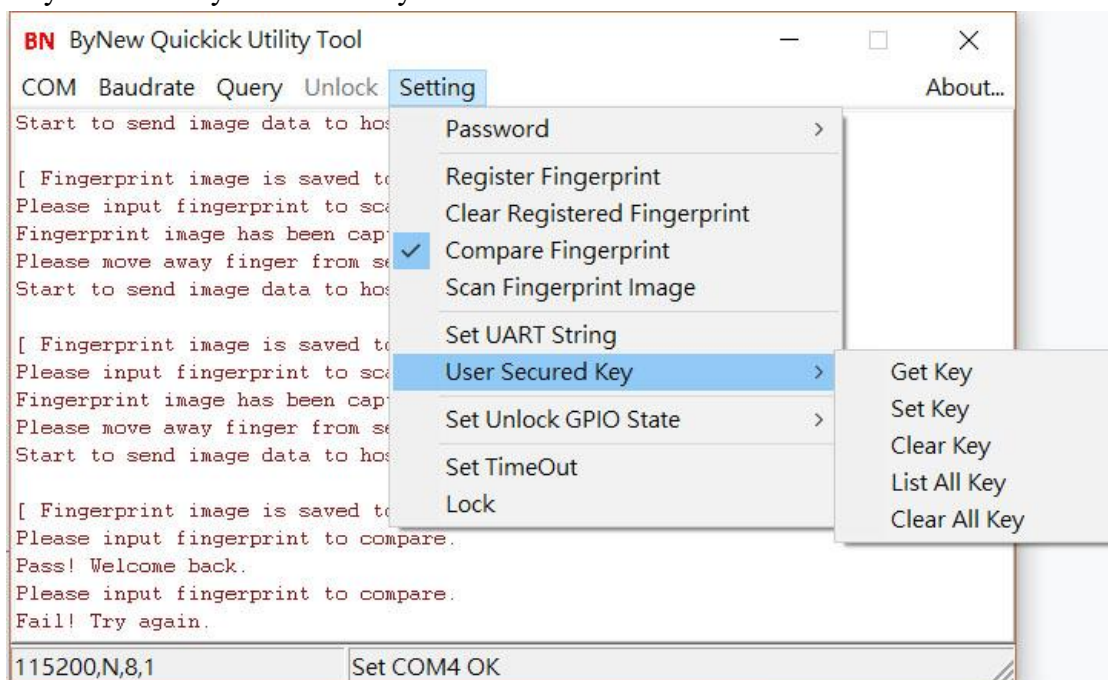
It is to set the output string from device UART when compare fingerprint. User can define the output string for match and mismatch:



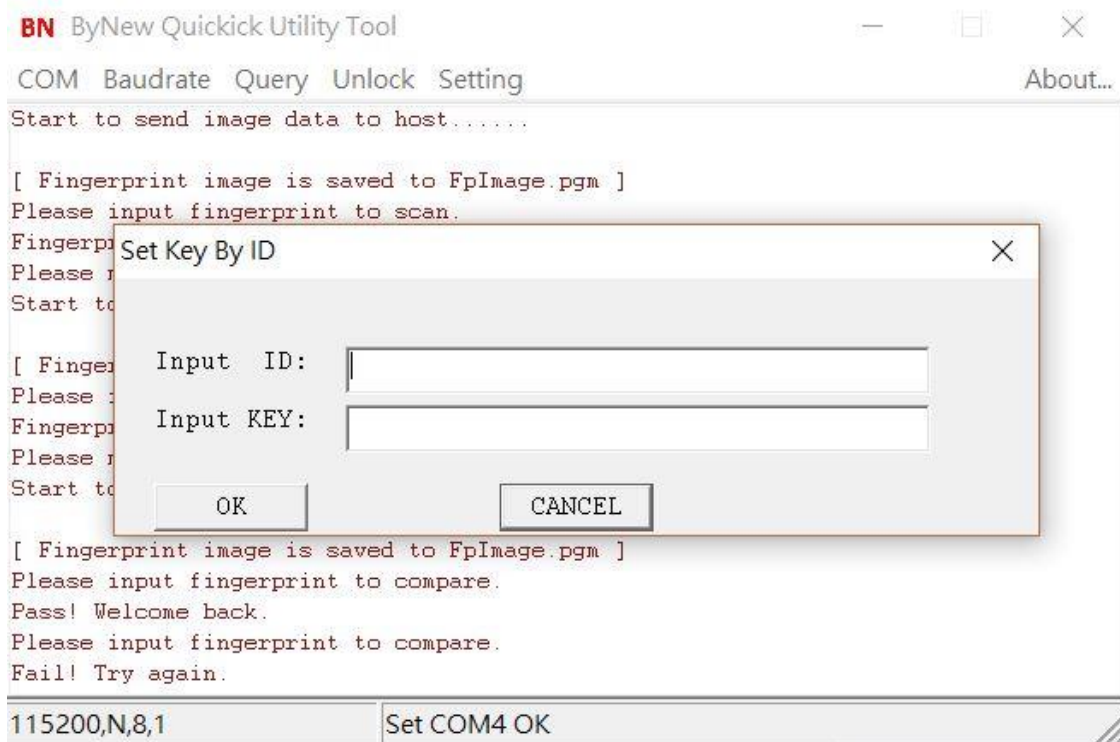
And then all the fingerprint comparison, the output message will be the setting of UART string as the above example.

User Secured Key

User keep his secured data in secured area, user secured data can be password of an e-mail account, safe box password, or any other important thing. Here it supports functions of Get Key/Set Key/Clear Key/List All Key/Clear All Key.



Each secured data includes “ID” and “KEY”, both are string format.



- Get Key
User input ID to query its KEY.
- Set Key
User set a new secured data with its ID and KEY.
As an example, user can input an e-mail account as ID and its password as KEY.
- Clear Key
User input an ID to clear its secured data.
- List All Key
List all stored secured data on screen.
- Clear All Key
User clear all secured data.

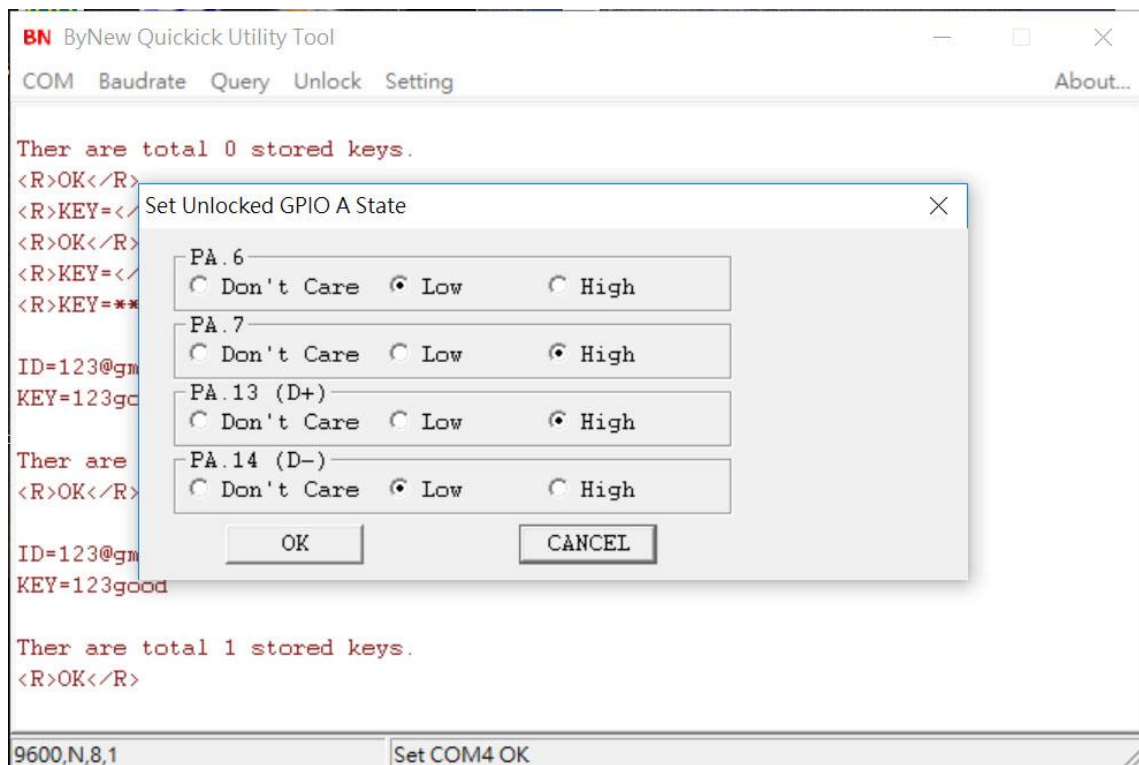
Set Unlock GPIO State

It can set GPIO state when device is in unlocked state, and it is the opposite state when device is in locked state. User can define the GPIO is “Don’t care”, Low or High when device is in unlocked state.

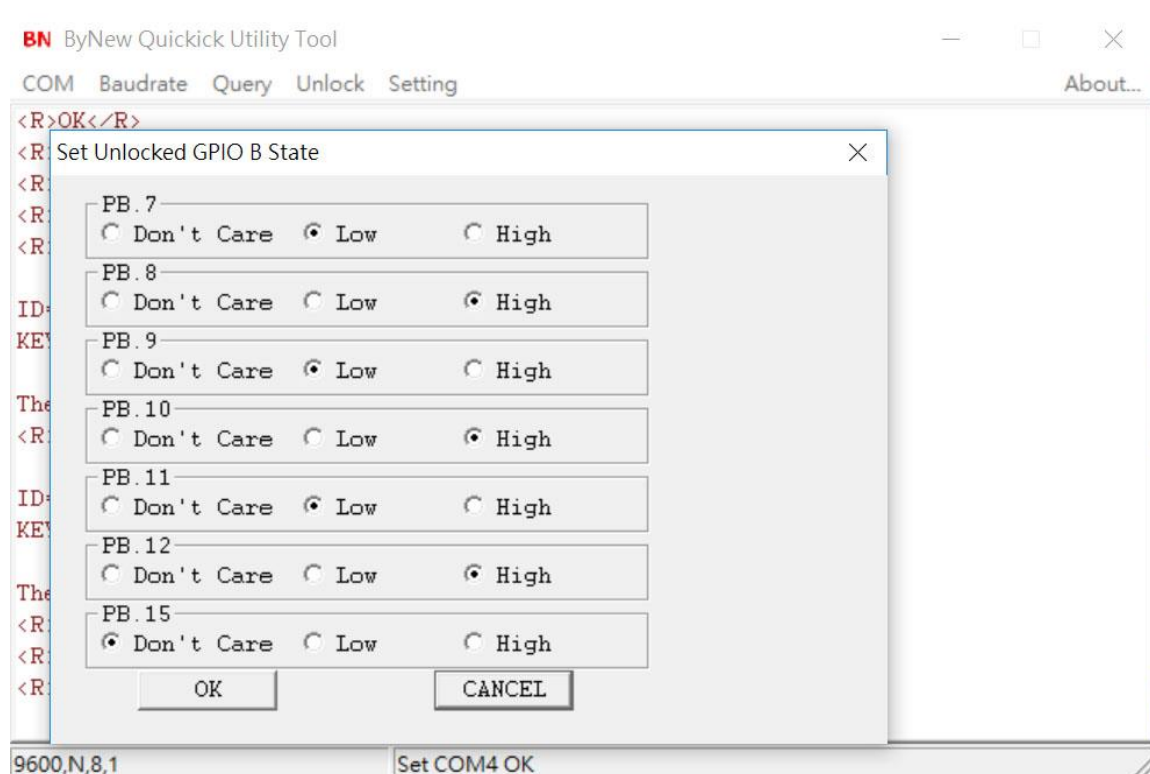
It supports some GPIOs of PA, PB and PC:

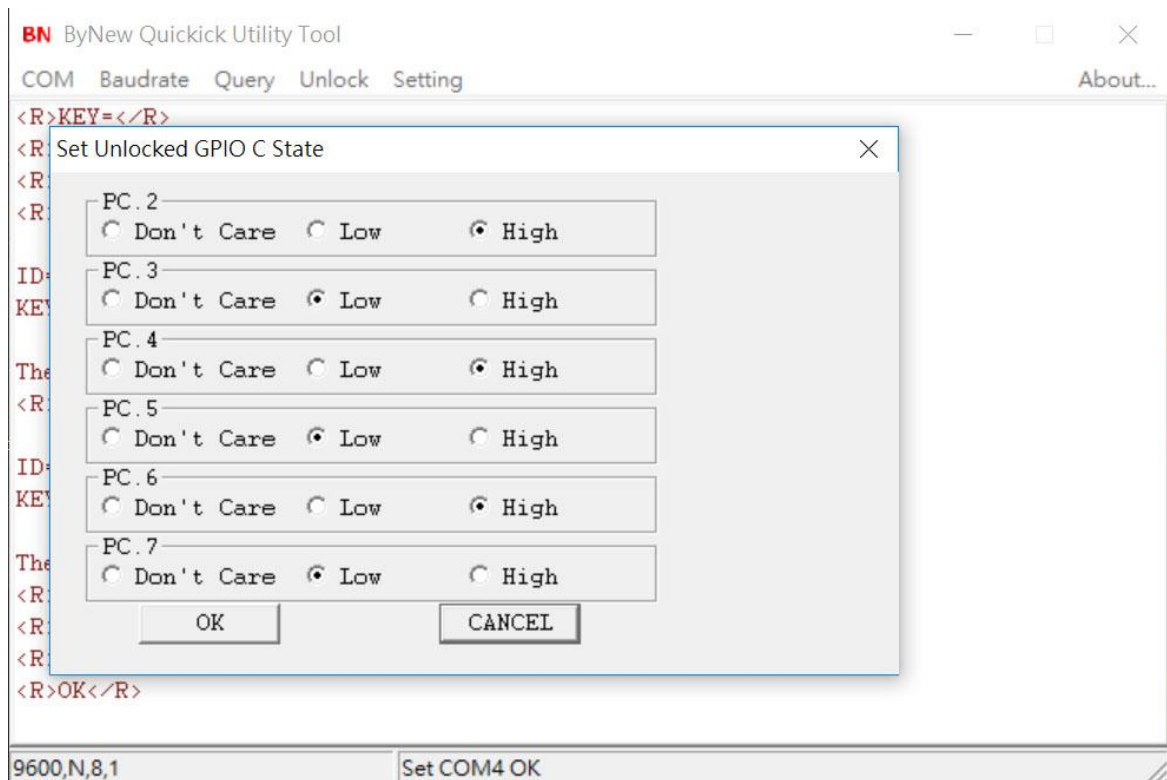
GPIO A: PA.6, PA.7, PA.13, PA.14

GPIO B: PB.7, PB.8, PB.9, PB.10, PB.11, PB.12, PB.15



PA.13 and PA.14 are the D+ and D- signals of USB bus, here just take them as GPIO output pins.

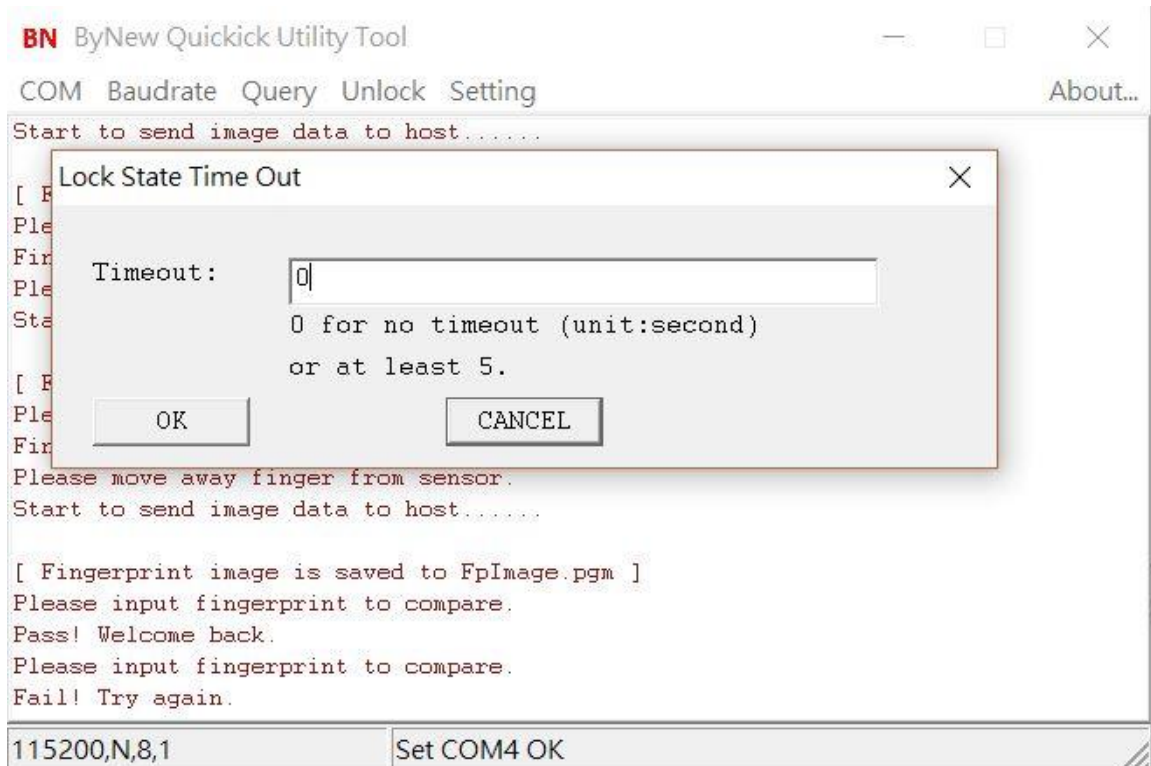




Please be noted that NOT all the GPIOs above can be set and work, please check the EVB board for the available GPIOs.

Set Timeout

It is to set timeout for unlocked state.



When device is in unlocked state and timeout, it will go back to locked state.

Lock

Click this item will make device from unlocked state back to locked state immediately. The Setting menu will be grayed and Unlock menu be enabled again. User needs to unlock again before doing any Setting item.
