

# 指紋辨識模組(YE-UL1701B)使用手冊

版本號: V1.0.5

**YANEN** 研恩科技

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

## 內 容

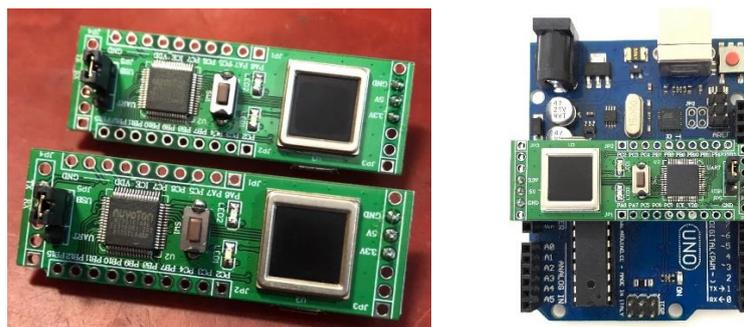
<b>1. 產品介紹</b> .....	<b>3</b>
1.1. 模組特色.....	3
1.2. 規格.....	3
1.3. 特徵.....	4
1.4. 模組介面.....	4
<b>2. 通訊協議</b> .....	<b>5</b>
<b>3. LED 燈號顯示說明</b> .....	<b>6</b>
3.1. 配件模式.....	6
3.2. 展示模式.....	6
<b>4. 操作範例</b> .....	<b>7</b>
4.1. ARDUINO 配件模式.....	7
4.1.1. 硬體安裝.....	7
4.1.2. Arduino 程式碼.....	7
4.1.3. Arduino 操作命令.....	7
4.2. 樹莓派(RASPBERRY PI) 配件模式.....	10
4.2.1. 硬體安裝.....	10
4.2.2. Raspberry Pi 程式碼.....	11
4.2.3. Raspberry Pi 操作命令.....	11
4.3. PC 端配件模式.....	13
4.3.1. 硬體安裝.....	13
4.3.2. 工具程式安裝.....	13
4.3.3. 工具通訊架構.....	13
4.3.4. 工具操作說明.....	14

## 1. 產品介紹

本文件為「指紋辨識評估板(YE-UL1701B)」的操作手冊，以下說明皆稱為 YE-UL1701B。內容包含 YE-UL1701B 的簡介、規格說明、通訊協議、操作說明及不同使用環境下的範例，本手冊內的操作範例亦有提供測試軟體與原始代碼。YE-UL1701B 是一個高效能指紋辨識模組，可提供產品開發廠商應用於指紋辨識需求進行開發(應用範圍如指紋門鎖、保險櫃、工業控制、授權許可、金融...等隱私及安全領域)。此外，YE-UL1701B 也可作為有 UART (Serial Port)的微處理器控制板像是 Arduino 開發板或樹梅派、FPGA、ARM、pcDuino...等開發套件使用，方便使用者開發使用。本規格書與操作說明等內容資料，給予使用者或開發者透過本文件所提供的說明步驟即可輕易完成操作。

### 1.1. 模組特色

- YE-UL1701B 指紋識別辨識模組為一個電容式指紋辨識傳感器，傳感器組成包含高速處理器、高效能指紋演算法與 FLASH 晶片等軟硬體構成；
- 性能穩定，結構簡單，具有指紋比對、圖像掃描傳送、搜索和範本指紋儲存等功能，另系統獨特內碼保護機制。
  - 內含指紋比對展示程式，可輸入註冊 24 枚指紋數來做比對，比對速度快，正確率高，可提供開發者據以開發各式各樣的應用。(進行指紋辨識相關產品開發或測試使用)
  - 系統特有 XOM 機制可以保護程式碼只能[執行]，無法被[讀取]，故能保護重要智慧財產權程式碼不會被讀取後破解。
- 功能完整：獨立完成指紋採集、指紋範本儲存、指紋比對 (1:1) 和指紋搜索 (1:N) 功能。
- 體積小巧：產品體積小巧，內置指紋演算法晶片的電路板，集成一體，方便安裝，故障少。
- 超低功耗：產品整體功耗極低，適用於低功耗要求的場合。
- 應用開發簡單：開發者可根據提供的控制指令，自行指紋相關應用產品的開發，無需具備專業的指紋識別知識。



YE-UL1701B 產品圖(右圖為直接插件在 Arduino 版上)

### 1.2. 規格

- 模組尺寸：53.3x17.8x1.6(mm)
- MCU：Nuvoton M2351, ARM-M0 Plus, 32-bit, 64 MHz
- 指紋感應器：A172
- 影像像素：176\*176pixels
- 影像解析度：508 Dpi
- 取樣頻率：40 frames/s

- 認假率(FAR)：<0.001% (Security Level, 3)
- 拒真率(FRR)：<0.1% (Security Level, 3)
- 比對模式：1:1, 1:N
- 比對時間：< 0.3s
- 影像採集時間：< 0.2s
- 工作溫度：-20 to +60°C / <RH 90%
- 儲存溫度：-40 to +70°C / <RH 90%
- 通訊方式：UART
- 電源：UART 3.3V 或 USB 供電

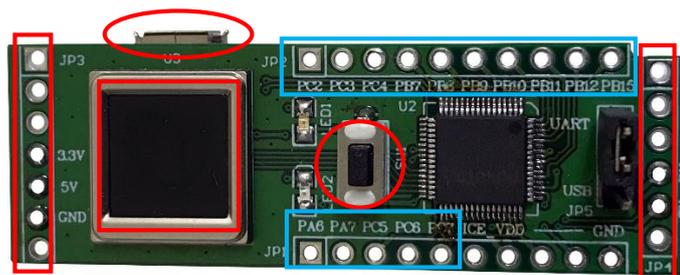
### 1.3. 特徵

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

### 1.4. 模組介面

YE-UL1701B 模組的通信介面為 UART。YE-UL1701B 評估板說明與接線圖如下所示。圖左上方橢圓形紅框處是 USB 接口，正方形紅框處是指紋感測器，中間圓形紅框處是重置鍵，左右長方形紅框內有 UART 接口，UART 腳位為 VCC(3.3V)、GND、RX 和 TX 定義如下表。另有提供 15 個 GPIO 給使用者運用(藍框處部分)。

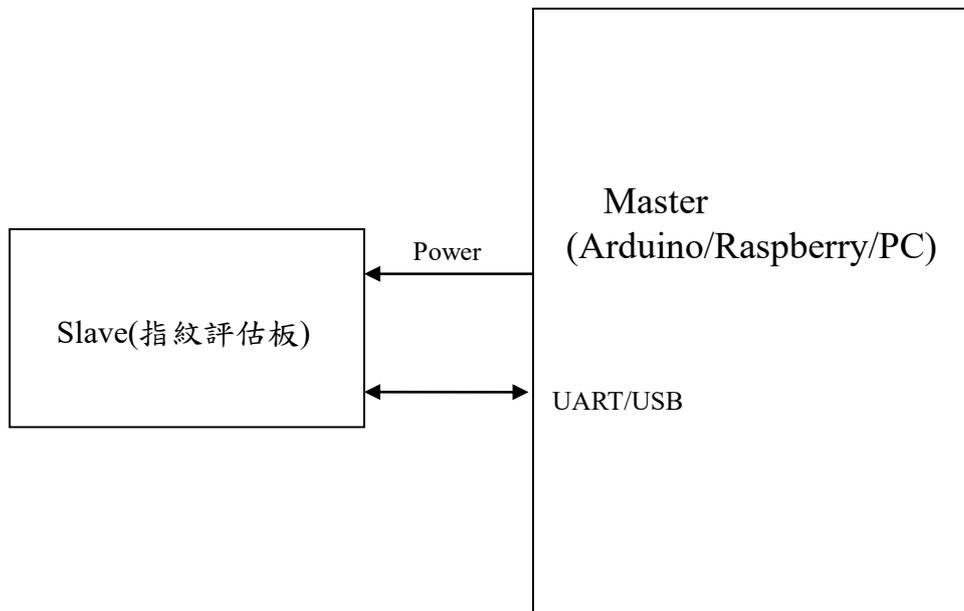
名稱	類型	描述
VCC	In	外接 3.3V
GND	--	外接 GND
RX	In	數位資料輸出腳位
TX	Out	數位資料輸入腳位



YE-UL1701B 產品說明與接線圖

## 2. 通訊協議

說明指紋評估板(YE-UL1701B)與 Arduino, Raspberry PI...等微控制器或 PC 之間的通訊協議。Arduino 或 Raspberry PI 是主動裝置(Master)而指紋評估板是從動裝置(Slave)。Slave 由 USB 或 UART 來供電。



本通訊協議使用的通道是 UART 或 USB，目前支援 UART。

指紋評估板的 UART 出廠時的傳輸速率是 9600,N,8,1，可經由命令更改此設定。

有關通信協議的詳細內容，請另參考 **YE-EVB Communication Protocol** 文件。

## 3. LED 燈號顯示說明

指紋評估板(YE-UL1701B)的 LED 燈號顯示與其當時的工作狀態有關。

指紋評估板上電或重置後，若紅色 LED 和綠色 LED 輪流閃，則表示工作在配件模式，若是單獨是紅色 LED 或綠色 LED 閃爍，則表示工作在展示模式。

### 3.1. 配件模式

上電或重置(reset)後，若紅色 LED 和綠色 LED 輪流閃，表示是在接收命令狀態，可由 Master 端(Arduino 或 Raspberry PI)對它下命令來執行相關功能。

此模式的各種燈號顯示說明如下：

紅色 LED 和綠色 LED 輪流閃爍：

處於接收命令狀態，等待 Master 端的命令。

紅色 LED 和綠色 LED 同時 ON(亮)：

處於提醒狀態，兩三秒後會進入註冊狀態。

紅色 LED 閃爍：

處於註冊狀態，等待指紋按壓來註冊，會註冊兩枚指紋，每一枚指紋需按壓 3 次，每次按壓成功掃描指紋後會讓綠色 LED 亮兩三秒。

綠色 LED 閃爍：

處於輸入指紋狀態，等待指紋按壓來掃描指紋影像以便和已註冊的指紋做比對或上傳指紋影像。若是比對指紋，則比對成功則綠色 LED 亮兩三秒，比對失敗則紅 LED 亮兩三秒。若是上傳指紋影像，則影像上傳期間，紅色 LED 和綠色 LED 輪流各閃 2 次。

紅色 LED 和綠色 LED 輪流各閃 2 次：

處於指紋影像上傳狀態。

### 3.2. 展示模式

上電或重置後，若單獨是紅色 LED 或綠色 LED 閃爍，則表示工作在展示模式。此模式的各種燈號顯示說明如下：

紅色 LED 和綠色 LED 同時 ON(亮)：

處於提醒狀態，兩三秒後會進入註冊狀態。

紅色 LED 閃爍：

處於註冊狀態，等待指紋按壓來註冊，會註冊兩枚指紋，每一枚指紋需按壓 3 次，每次按壓成功掃描指紋後會讓綠色 LED 亮兩三秒。

綠色 LED 閃爍：

處於比對狀態，等待指紋按壓來和已註冊的指紋做比對，比對成功則綠色 LED 亮兩三秒，比對失敗則紅 LED 亮兩三秒。連續比對失敗 3 次，則進入提醒狀態。

## 4. 操作範例

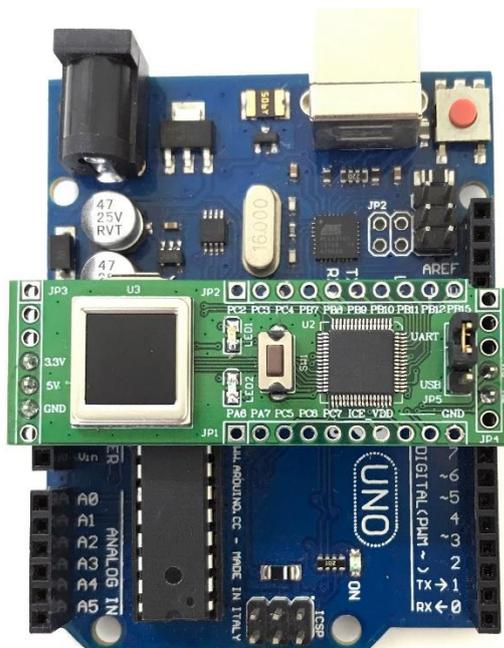
AJ155 指紋機評估板在 Arduino、Raspberry Pi 與 PC 端下的測試環境範例如下：

### 4.1. Arduino 配件模式

本範例以 Arduino UNO 進行操作並說明。

#### 4.1.1. 硬體安裝

YE-UL1701B 可使用接線或直接插件於 Arduino 控制板，如下圖示：



插件對照	Arduino	YE-UL1701B
	3.3V	3.3V
	GND	GND
	GPIO10	RX
	GPIO9	TX

#### 4.1.2. Arduino 程式碼

完整的程式請參考：[Fingerprint\\_Demo.ino](#)。

#### 4.1.3. Arduino 操作命令

當於 Arduino IDE 執行上傳 Fingerprint\_Demo.ino 後，若無出現錯誤畫面將出現以下項目，接續輸入功能代碼即可進行各項操作。

=====

- 1.註冊指紋
- 2.比對指紋
- 3.指紋影像資料
- 4.掃瞄並上傳指紋
- 5.檢查註冊指紋個數

### 1)註冊指紋：

當執行功能註冊指紋(輸入 1)後，使用者須在指紋感測區域連續註冊指紋共 24 枚，或註冊所需的指紋數後按下重置鍵中斷。註冊時每枚指紋註冊按壓 3 次(註冊第一枚指紋時每按一下燈號變為綠燈後即可離開指紋，連續按壓 3 次後，接著註冊下枚指紋重複按壓 3 次，後續動作相同)。以下為註冊 2 枚為例成功執行回應如下：

```

COM5 (Arduino/Genuino Uno)
1.註冊指紋
2.比對指紋
3.指紋影像資料
4.掃瞄並上傳指紋
5.檢查註冊指紋個數

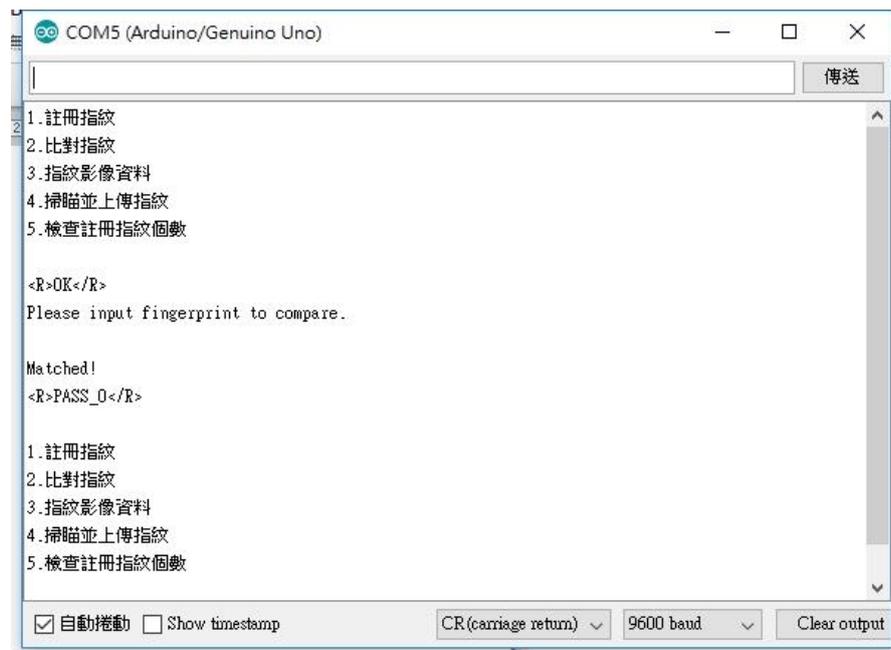
<R>OK</R>
Fingerprint registration state.

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>
1.註冊指紋
2.比對指紋
3.指紋影像資料
4.掃瞄並上傳指紋
5.檢查註冊指紋個數
  
```

### 2)比對指紋：

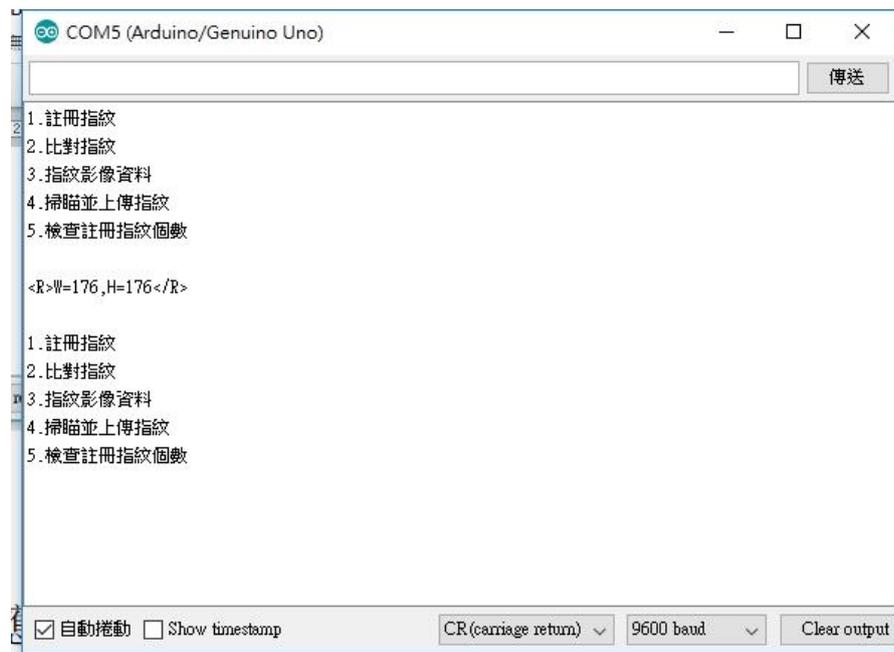
當執行功能比對指紋(輸入 2)後，使用者在指紋感測區域按壓指紋，將顯示出比對錯誤與比對正確的訊息。

序列埠監控視窗回應如下：



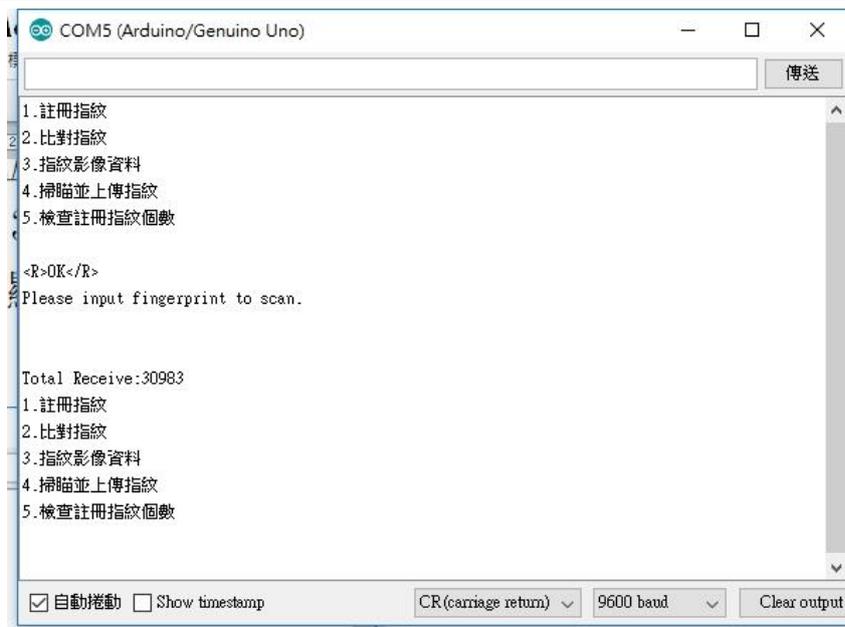
### 3)詢問指紋影像資訊：

當執行功能詢問指紋影像資訊(輸入3)後，成功執行回應如下：



### 4)掃瞄並上傳指紋影像：

當執行功能掃瞄指紋影像(輸入4)後，使用者須在指紋感測區域按壓指紋，成功執行回應如下：



(Note: 由於 Arduino 的 SRAM 大部分小於 30976 Bits(30KBytes)，所以無法將所有影像資料顯示出來，此部分使用者須注意測試時所使用的硬體規格。)

### 5) 檢查註冊指紋個數：

序列埠監控視窗回應如下：

回應直接以<R>和</R>來封包其個數，例如：

<R>0</R> 表示並無註冊資料

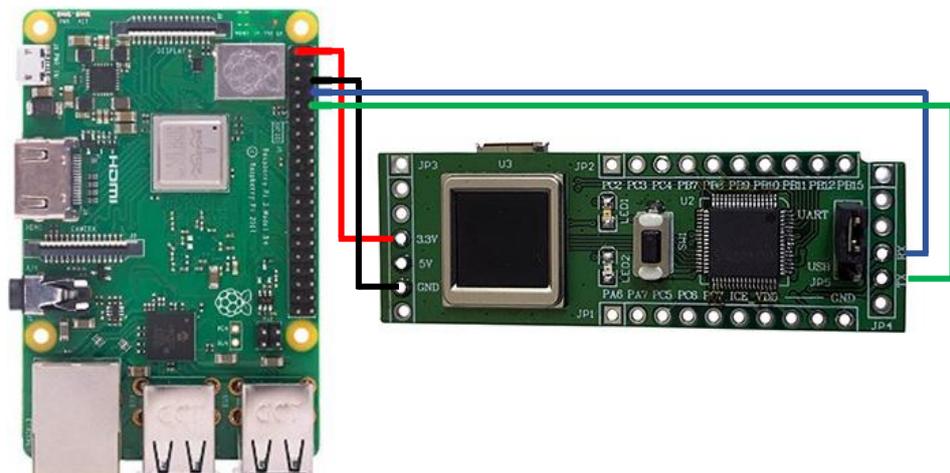
<R>2</R> 表示有 N 枚註冊資料

## 4.2. 樹莓派(Raspberry Pi) 配件模式

Raspberry Pi 以 Raspberry Pi 3 B+ 為範例說明。

### 4.2.1. 硬體安裝

AJ155 與 Raspberry Pi 接線如下圖示：



接線對照	Raspberry Pi	AJ155
紅線	3.3V	Vcc
黑線	GND	GND
藍線	GPIO14	RX
綠線	GPIO15	TX

#### 4.2.2. Raspberry Pi 程式碼

初始測試版本 Python version: 3.5.3

完整的程式請參考：Fingerprint\_Demo.py。

#### 4.2.3. Raspberry Pi 操作命令

當於 Python 執行 Fingerprint\_Demo.py 後，若無出現錯誤畫面將出現以下項目，接續輸入功能代碼即可進行各項操作。

```
=====
1.註冊指紋
2.比對指紋
3.指紋影像資料
4.掃瞄並上傳指紋
5.檢查註冊指紋個數
0.結束
請輸入功能代碼:
=====
```

##### 1)註冊指紋：

當執行功能註冊指紋(輸入 1)後，使用者須在指紋感測區域連續註冊指紋共 24 枚，或註冊所需的指紋數後按下重置鍵中斷。註冊時每枚指紋註冊按壓 3 次(註冊第一枚指紋時每按一下燈號變為綠燈後即可離開指紋，連續按壓 3 次後，接著註冊下枚指紋重複按壓 3 次，後續動作相同)。以下為註冊 2 枚為例成功執行回應如下：

```
<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>

註冊完成

## 2) 比對指紋：

當執行功能比對指紋(輸入 2)後，使用者在指紋感測區域按壓指紋，將顯示出比對錯誤與比對正確的訊息。

成功執行回應如下：

比對錯誤(錯誤指紋)

<R>OK</R>

Please input fingerprint to compare.

Unmatched!

<R>FAIL</R>

或

比對正確(正確指紋)

<R>OK</R>

Please input fingerprint to compare.

Unmatched!

<R>FAIL</R>

## 3) 詢問指紋影像資訊：

當執行功能詢問指紋影像資訊(輸入 3)後，成功執行回應如下：

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

## 4) 掃瞄並上傳指紋影像：

當執行功能掃瞄指紋影像(輸入 4)後，使用者須在指紋感測區域按壓指紋。成功執行回應如下：

Please input fingerprint to scan.

開始接收資料

Total Bytes=30983

資料接收完成

請輸入儲存路徑及檔名，例如 /home/pi/Finger/Image1.pgm，之後指紋影像將被儲存於指定路徑中。

## 5) 檢查註冊指紋個數：

當執行功能詢問檢查註冊指紋個數(輸入 5)後，執行回應如下：

<R>0<R> 表示並無註冊資料

or

<R>2<R> 表示有 N 枚註冊資料

### 4.3. PC 端配件模式

**Note:** 本測試範例使用者需自備一條 USB 轉 UART(或 USB 轉 TTL)線。

#### 4.3.1. 硬體安裝

除了指紋評估板外，另需要一條 USB 轉 UART(或 USB 轉 TTL)線把評估板接上 PC，USB 轉 UART 線接上評估板時，TX/RX 需交換對接：

指紋評估板	PC 端 (USB 轉 UART 線)	說明
VCC (3.3V)	VCC (3.3V)	若 USB 轉 UART 線的 VCC 是 5V 則不接，請把評估板插入 PC 的 USB 插孔，由 USB 來供電。
TX	RX	交換對接
RX	TX	交換對接
GND	GND	--

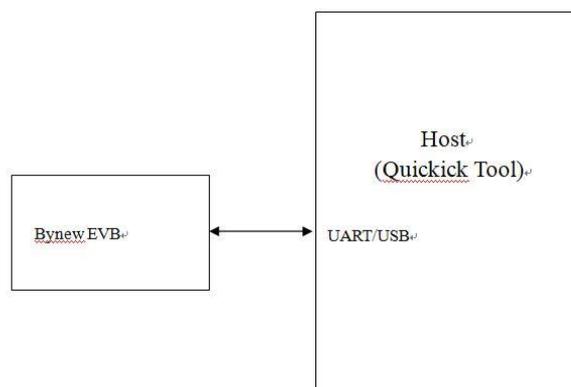
USB 轉 UART 至少會有 VCC/GND/TX/RX 四條線，接法如上表，再強調若 USB 轉 UART 線的 VCC 是 3.3V 則可直接地接上評估板 VCC，若其 VCC 是 5V，則不要把此 5V 的 VCC 接上評估板，以免評估板容易燒毀，必需改由 USB 接口來供電。

上電後評估板 LED 會輪流閃爍，在配件模式下的燈號顯示請見：指紋評估板 LED 燈號顯示說明。

#### 4.3.2. 工具程式安裝

BNQuickkick 工具程式，在 Windows 環境下執行，直接把它(BNQuickkick.exe, CxImage.exe and FreeImage.dll)放在某個資料夾即可執行。

#### 4.3.3. 工具通訊架構

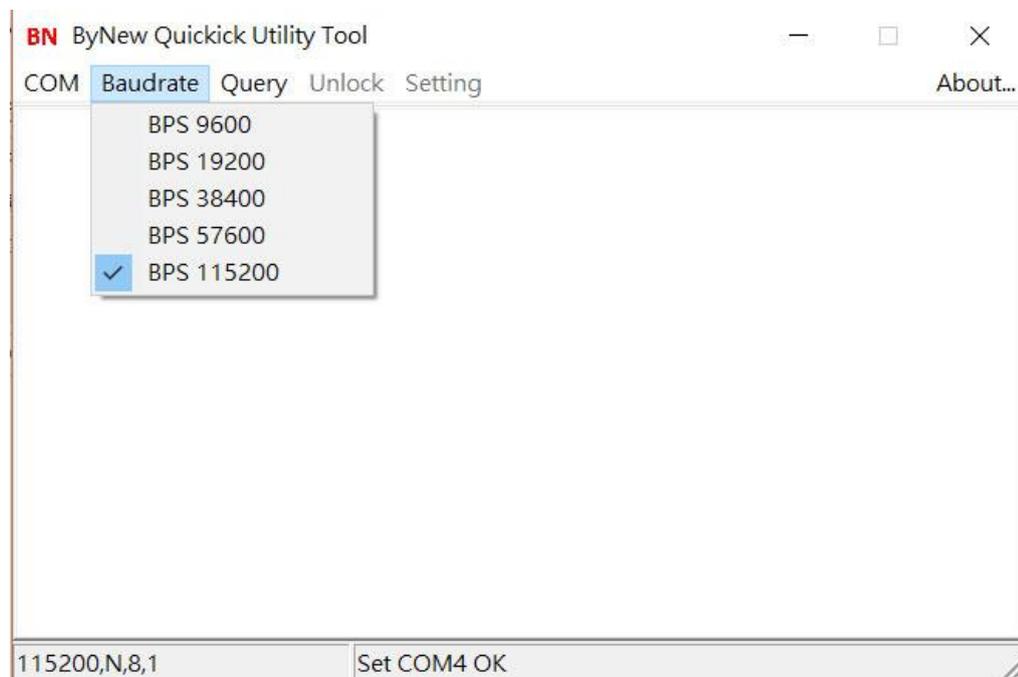
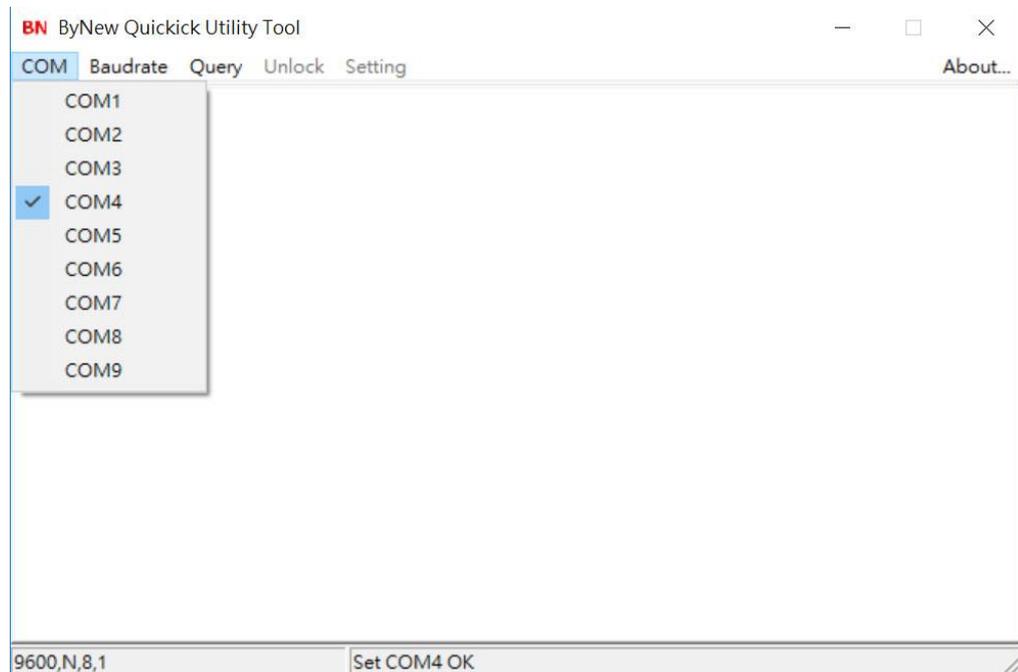


當前的通訊接口為 UART.

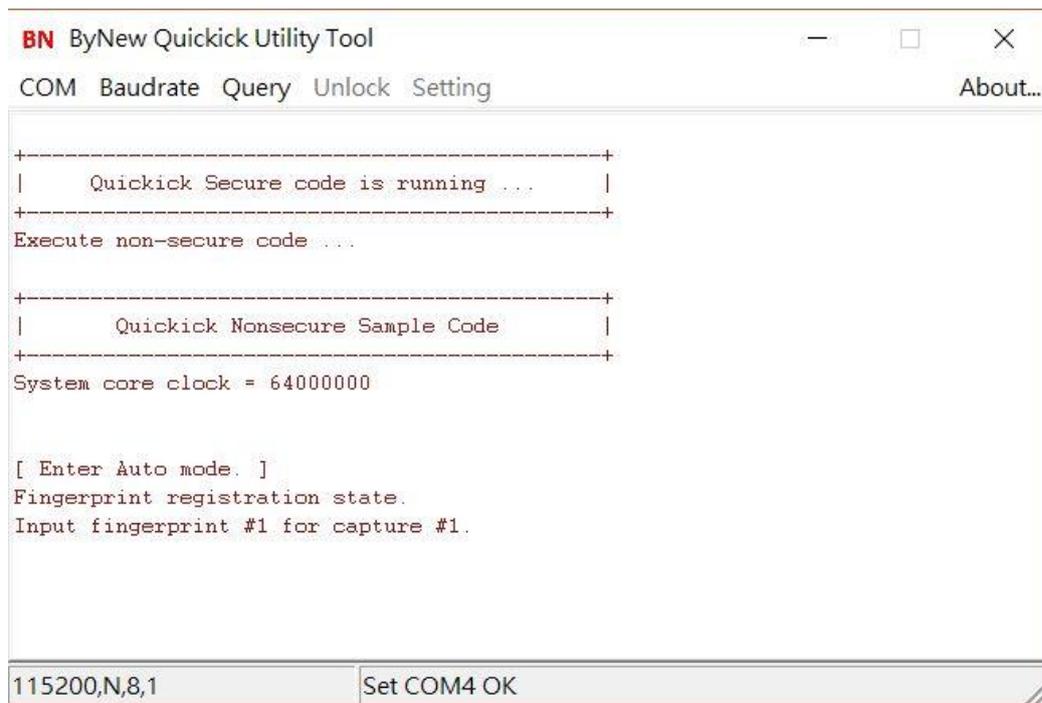
#### 4.3.4. 工具操作說明

### [COM and Baudrate]

執行 Quickkick 工具後，首先動作是確認並設置 Baudrate。它在 Quickkick 上有 COM 和 Baudrate 項目選單，如下所示：



在裝置管理員中找到 COM Port 並試著改變 Baudrate(從 9600 到 115200)，查看哪一個 Baudrate 設定可顯示出啟動畫面訊息，如下圖所示。

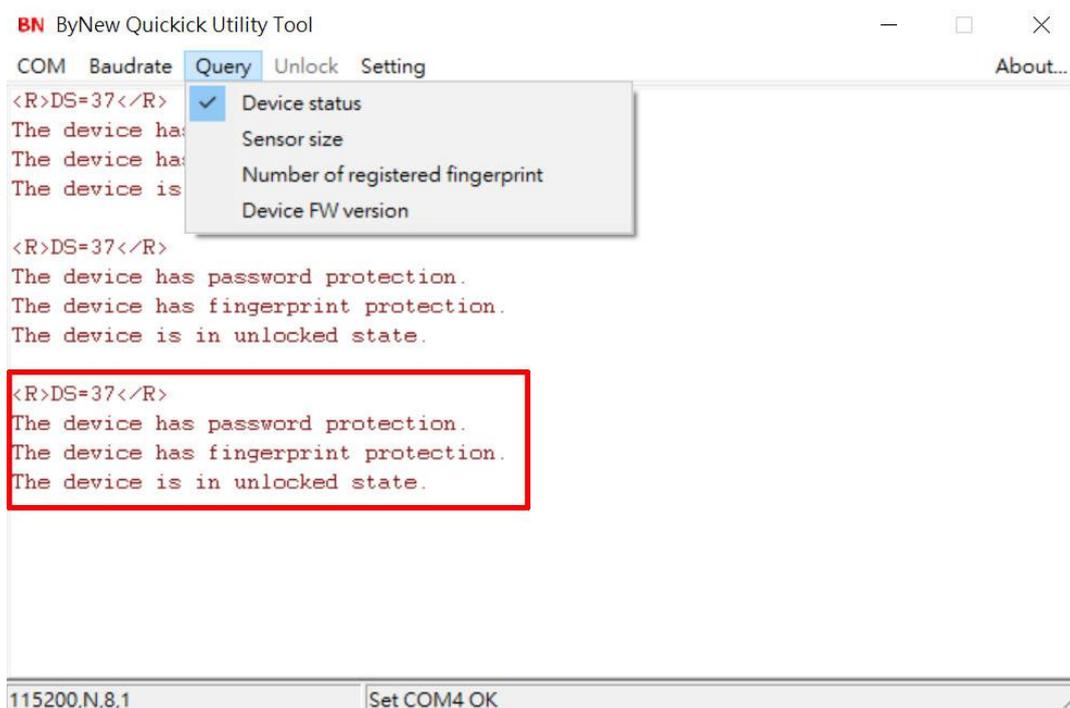


當使用者看見圖示的啟動訊息出現，即表示 Quickkick 與 YE-UL1701B 的通訊已成功設置，左下角字段顯示當前 Baudrate，中下區域顯示其執行的 COM 端口。通信設置成功後，現在使用者可以通過點選 Baudrate 選單中的 Baudrate 項來進行更改。

## [詢問指令]

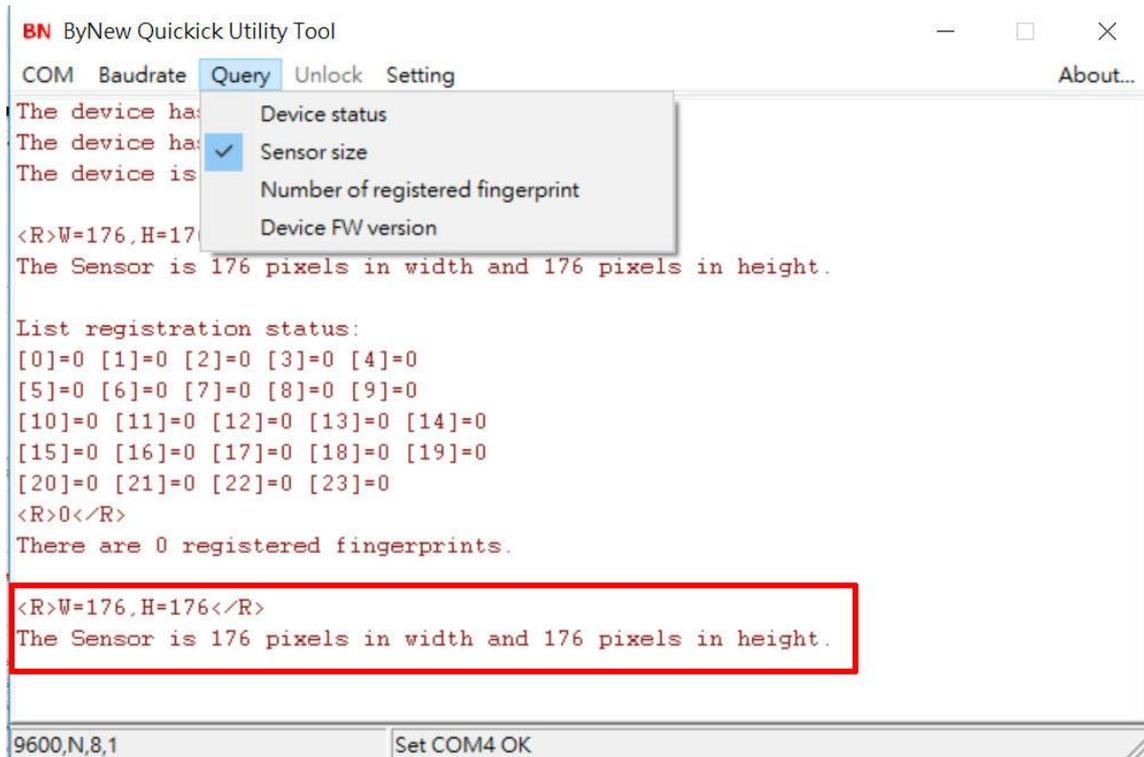
### # 設備狀態(device status)

這是一個 YE-UL1701B 設備檢查指令，執行後將返回目前是否具有密碼/指紋保護，並且是否處於鎖定或解鎖的狀態，如下圖所示。



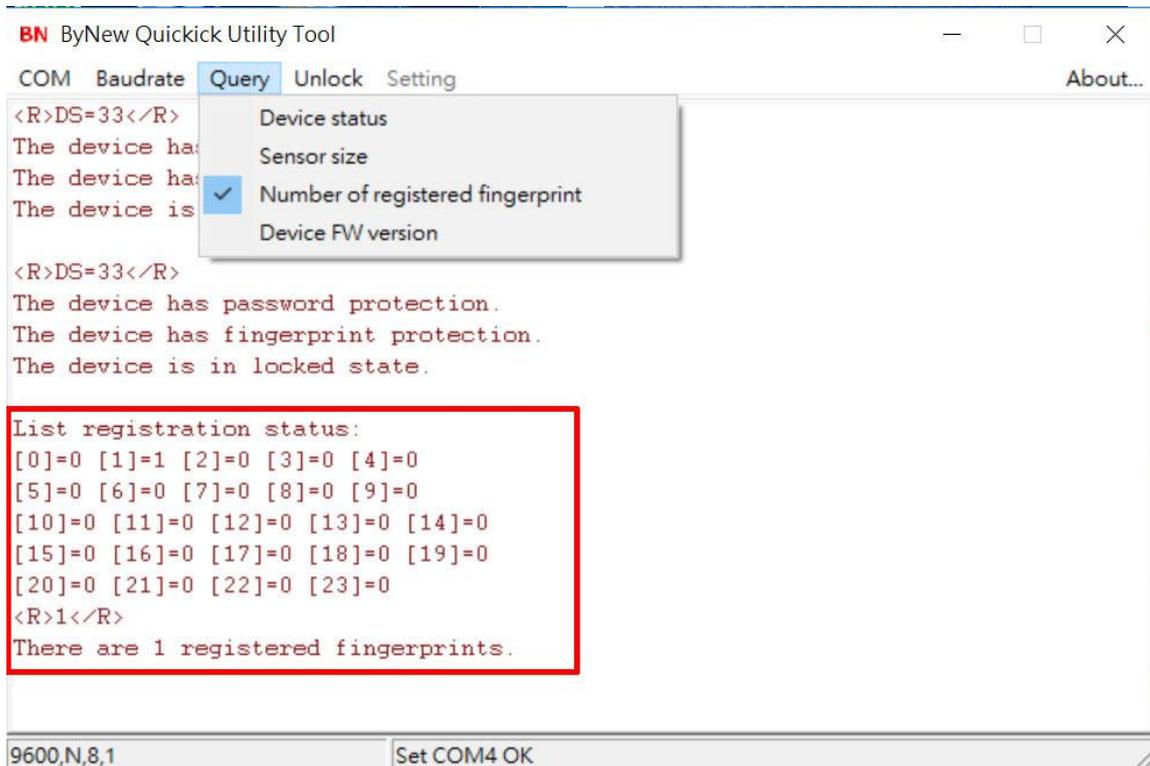
圖例中顯示裝置中存在密碼和指紋保護，裝置處於解鎖狀態，因此解鎖選單呈灰色，並且啟用設定選單以供使用者進行設定操作。

## # 感應尺寸(Sensor size)



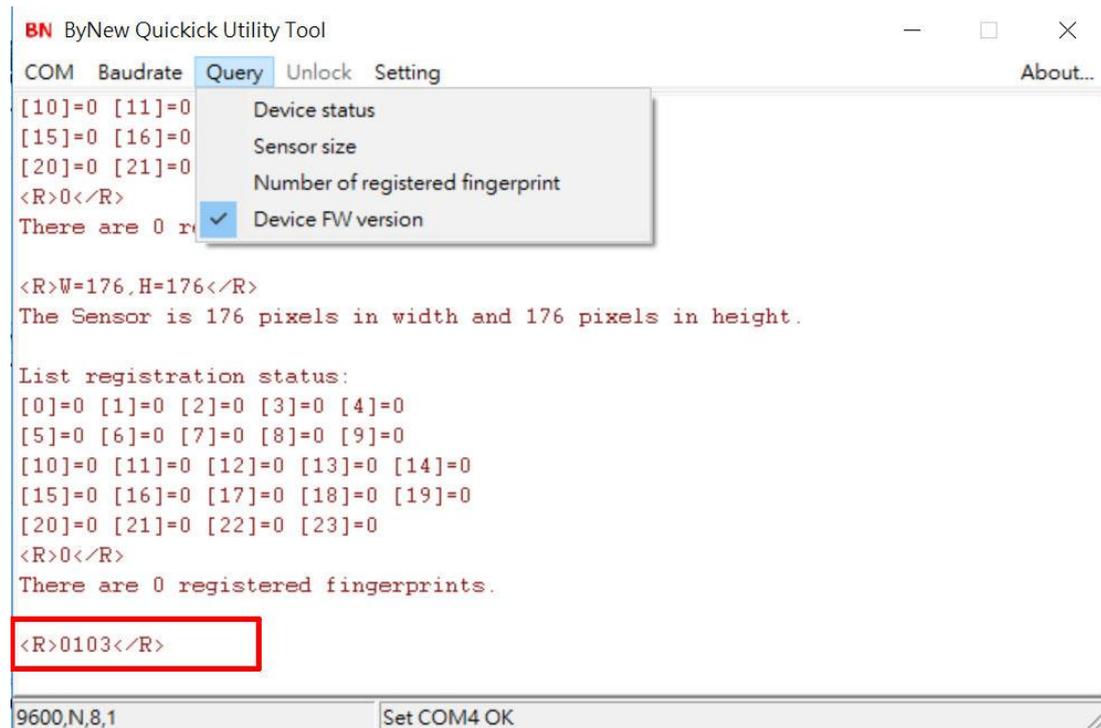
查詢指紋感測的影像尺寸。它將顯示指紋感測的寬度和高度的像素數。

## # 註冊指紋個數(Number of registered fingerprint)



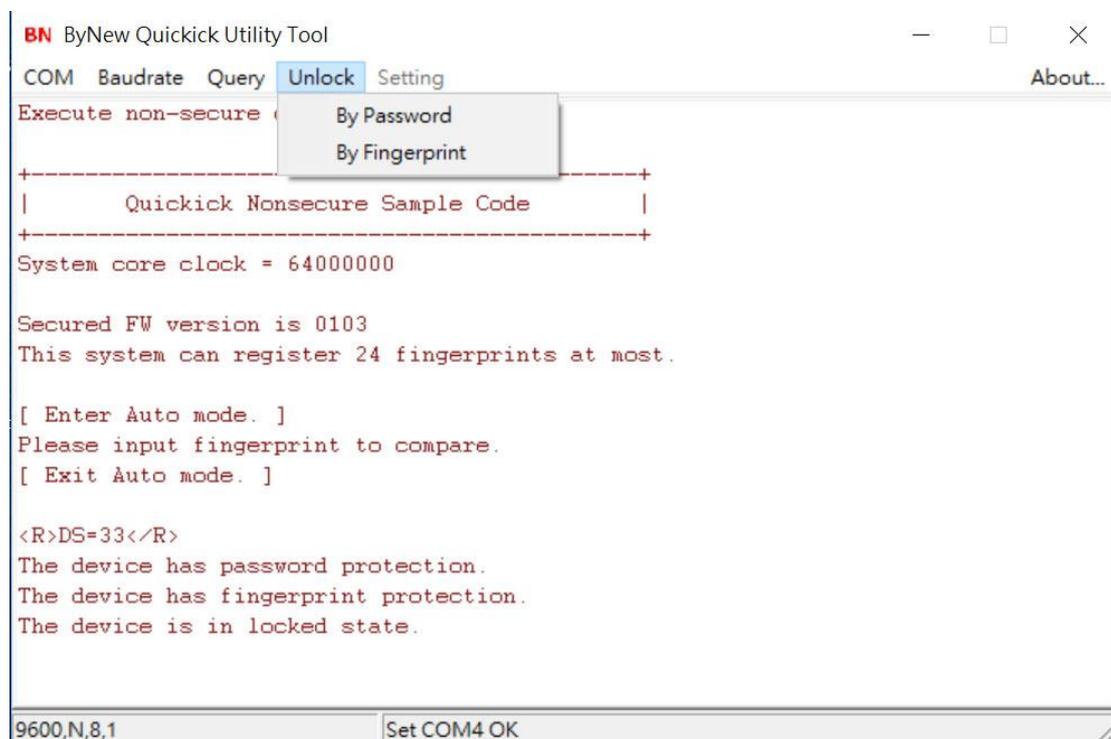
它將列出所有已註冊的指紋，如上圖所示，它顯示只有 1 個註冊指紋，並且它在索引 1 處註冊，這是第二個位置（它從索引 0 開始計數）。本裝置最多可以有 24 個註冊指紋。

## # 韌體版本(Device FW version)



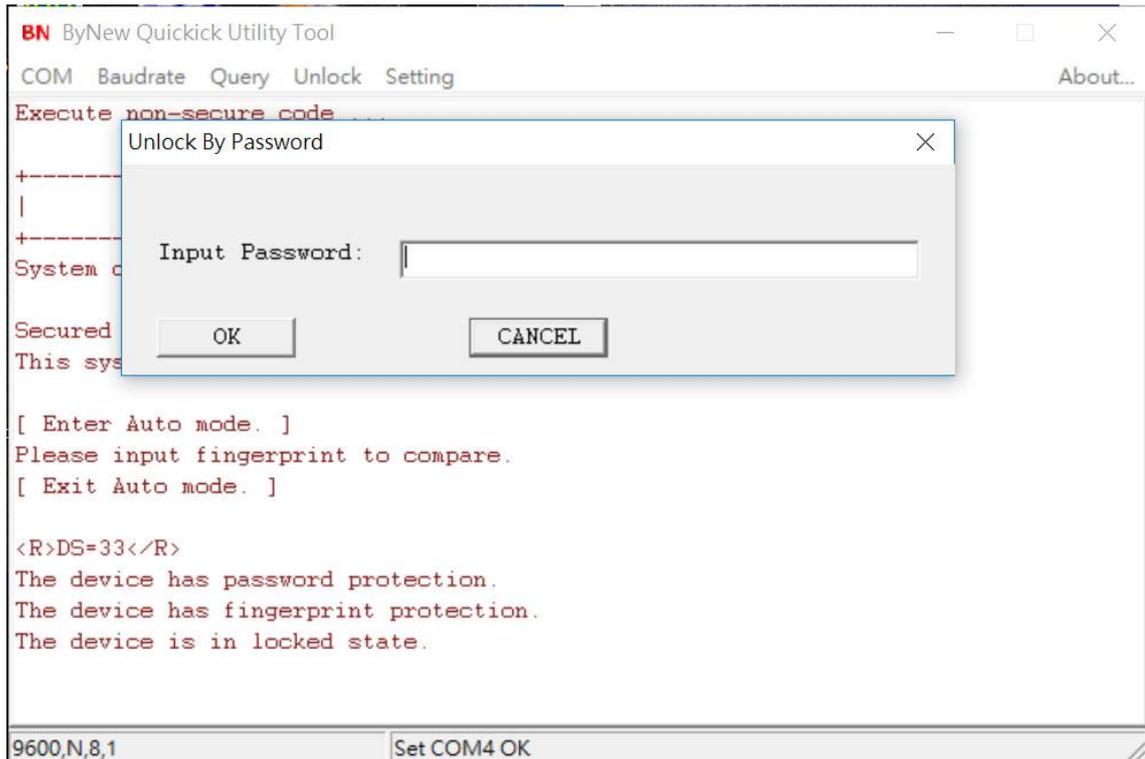
這裡顯示了裝置的韌體(Firmware)版本，如上圖所示，FW 版本為 0103，即版本 1.03。

## [解鎖]



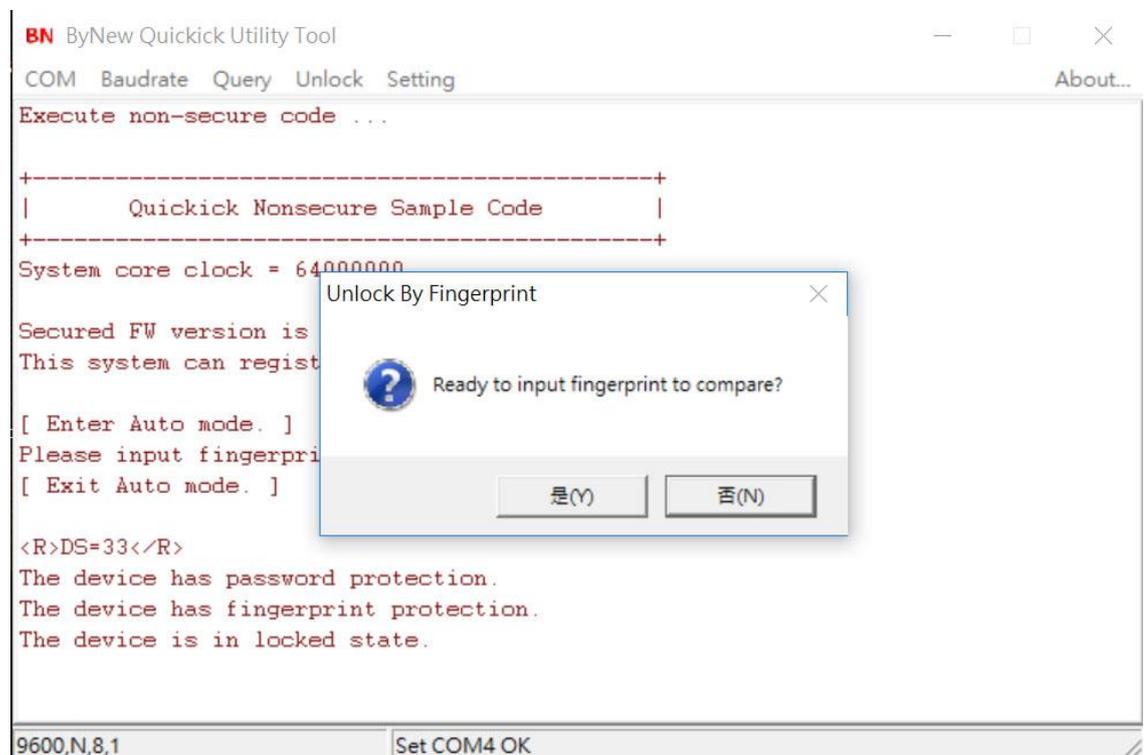
當裝置處於鎖定狀態時，需要通過密碼或指紋解鎖，以便對設備進行一些設定。裝置中只能保存 1 個密碼，但最多可以有 24 個註冊指紋。如果用戶選擇指紋解鎖，則必須是任何已註冊的指紋進行解鎖。

## # 密碼解鎖(By Password)

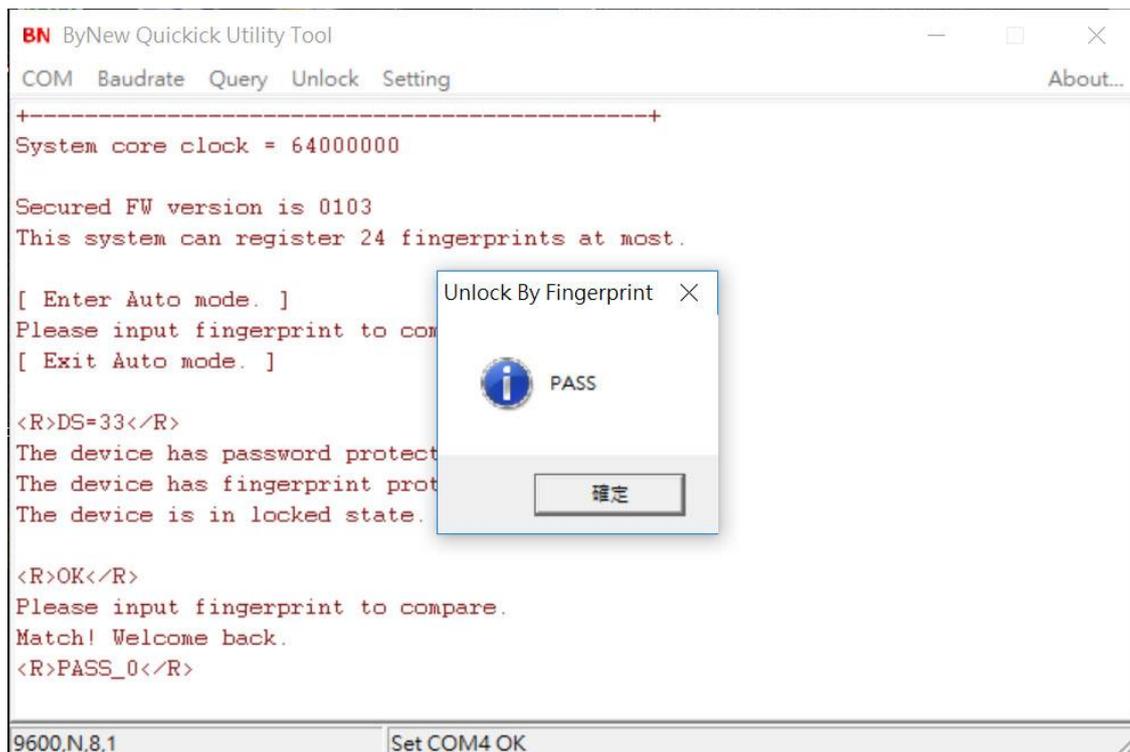


此功能將彈出一個訊息視窗供使用者輸入密碼，比較後將會回應“失敗”或“通過”。

## # 指紋解鎖(By Fingerprint)



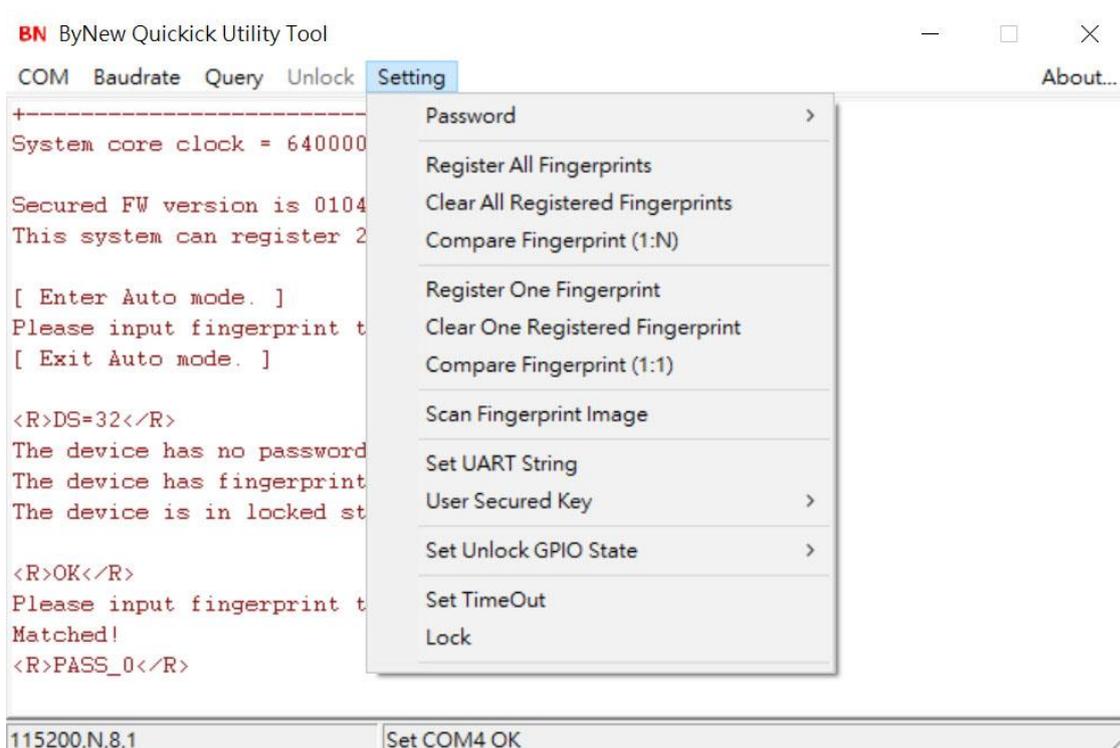
此功能將彈出一個訊息視窗，告知使用者準備輸入指紋，按“Y”後，它會指示用戶註冊他的指紋進行比較。



如果輸入指紋與任何已註冊的指紋匹配，它將如上圖所示回應“PASS”，並且在視窗上，它還將顯示與哪一個已註冊的指紋匹配。對於上圖所示，它顯示“PASS\_0”，因此它與第一個註冊的指紋（索引 0 的計數）相匹配。在使用者按“確認”的 PASS 消息框後，裝置進入解鎖狀態，並啟用設定選單。

### [功能設定]

裝置解鎖後，使用者可以執行以下所有設置項：



---

## # 密碼(Password)

這裡有 3 個次選項:

- Get Password  
獲得目前保存在裝置中的密碼。
- Set Password  
設定新密碼，先前的密碼將被覆蓋。
- Clear Password  
清除目前儲存在裝置裡的密碼。

## # 註冊全部指紋(Register All Fingerprints)

指紋註冊，最多可註冊 24 枚指紋。從開始註冊，使用者需要逐個(一個註冊完成後接著下一個)註冊指紋，註冊時也會替換掉先前已註冊的指紋，每一枚指紋註冊需按壓 3 次，使用者可依視窗顯示說明進行指紋註冊。

使用者也可以隨時按下裝置上的“重置”按鈕來停止此註冊過程。例如，在註冊 3 個指紋後，用戶可以按下 Reset 按鈕，使設備只有 3 枚註冊指紋。

## # 清除全部已註冊指紋(Clear All Registered Fingerprint)

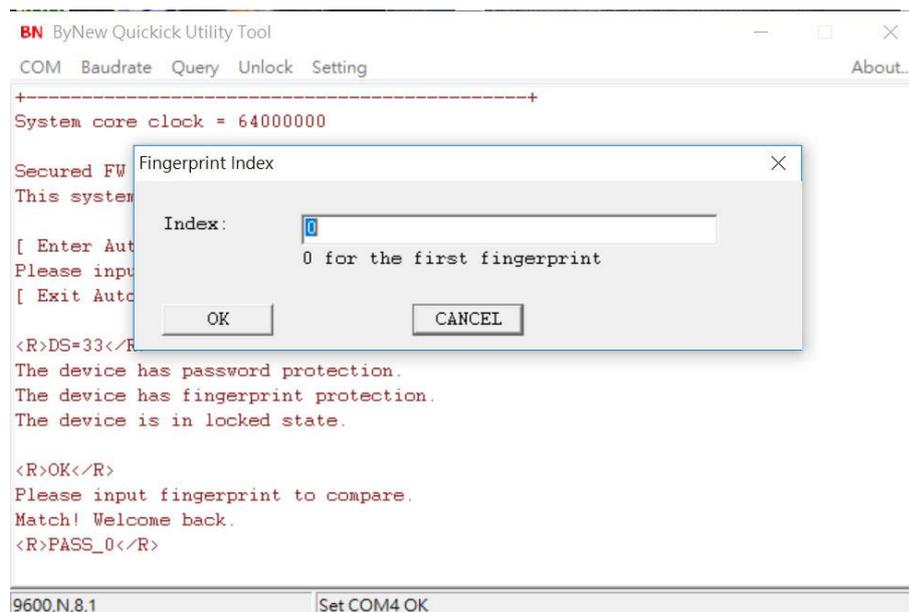
清除全部裝置內已註冊的指紋。

## # 比對指紋(Compare Fingerprint (1:N))

輸入指紋與所有已註冊的指紋進行比較。使用者依照說明註冊指紋。這將在視窗上顯示比對結果。

## # 註冊單枚指紋(Register One Fingerprint)

只註冊一個指紋，這會彈出一個訊息視窗，供使用者輸入指紋索引進行註冊。索引從 0 到 23。



指紋註冊後，具有相同索引的原始註冊指紋將被覆蓋。

### # 清除單枚已註冊指紋 Clear One Registered Fingerprint

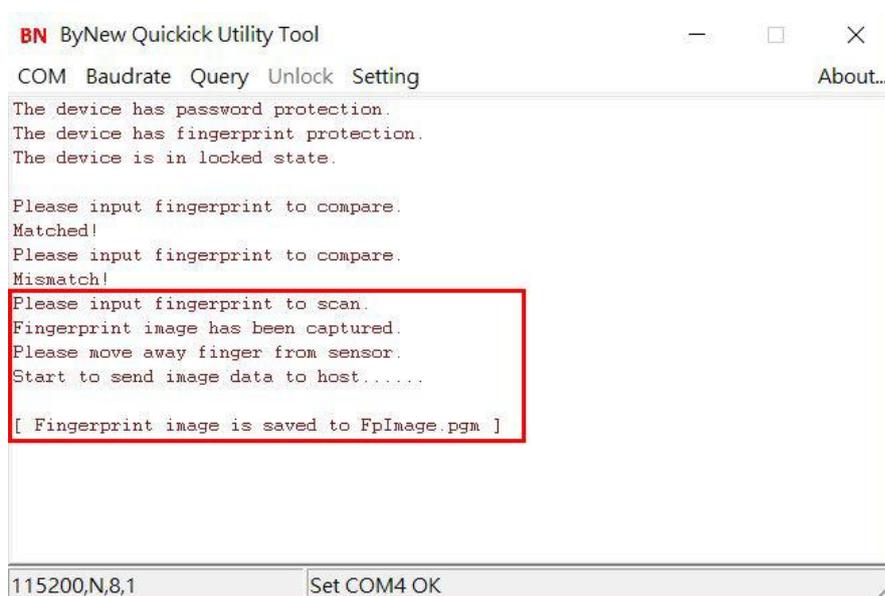
清除一些已註冊的指紋，這會彈出一個訊息視窗，提供使用者輸入指紋索引清除，索引從 0 到 23。

### # 比對指紋(Compare Fingerprint (1:1))

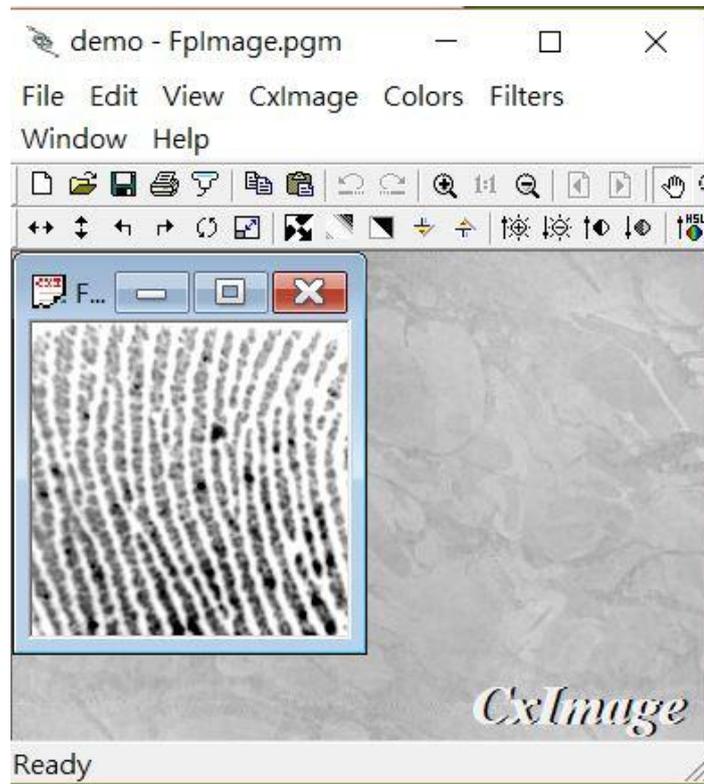
輸入指紋與指定的註冊指紋進行比較，彈出一個訊息視窗供使用者輸入指紋索引進行比較，索引數從 0 到 23。

### # 掃描指紋影像(Scan Fingerprint Image)

將指紋圖像掃描到主機(如 PC 端)。使用者按照視窗上的說明操作，掃描的圖像將儲存到文件 FpImage.pgm 中。

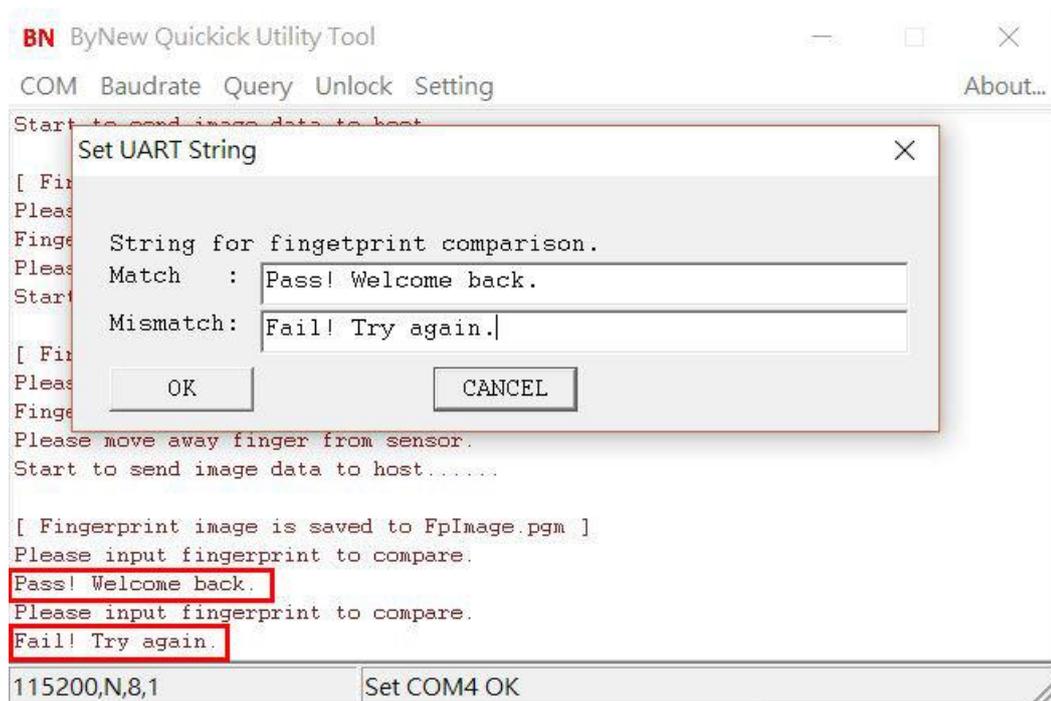


若 CxImage.exe 和 FreeImage.dll 與 BNQuickkick.exe 在相同文件夾中，將自動顯示掃描圖像，如下圖所示。



### # 設置 UART 字串(Set UART String)

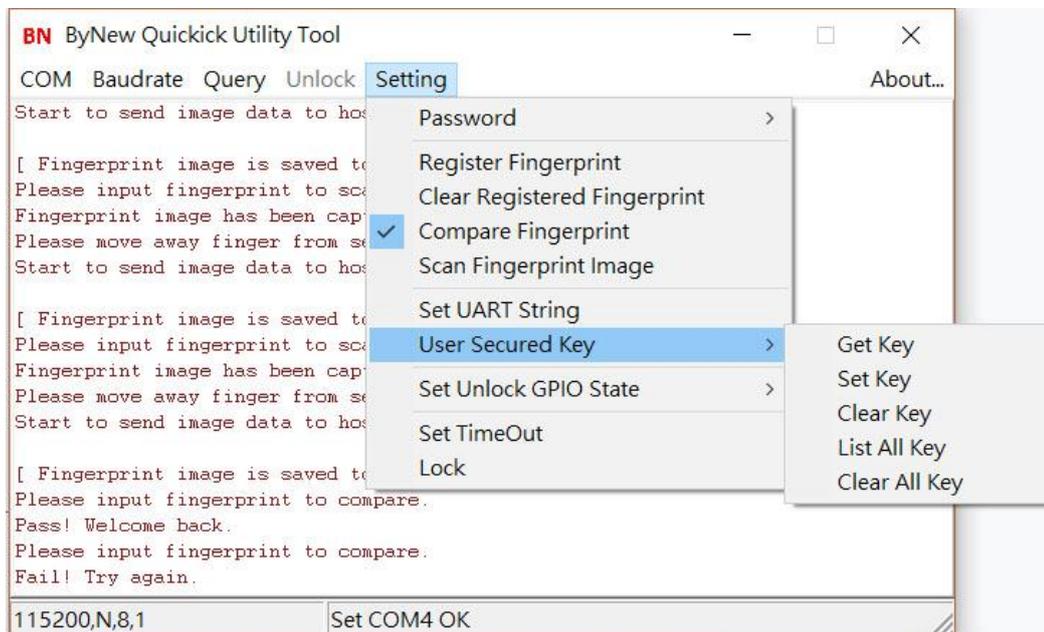
這是在比較指紋時從設備 UART 設定的輸出字串。使用者可以定義匹配和不匹配的輸出字串：



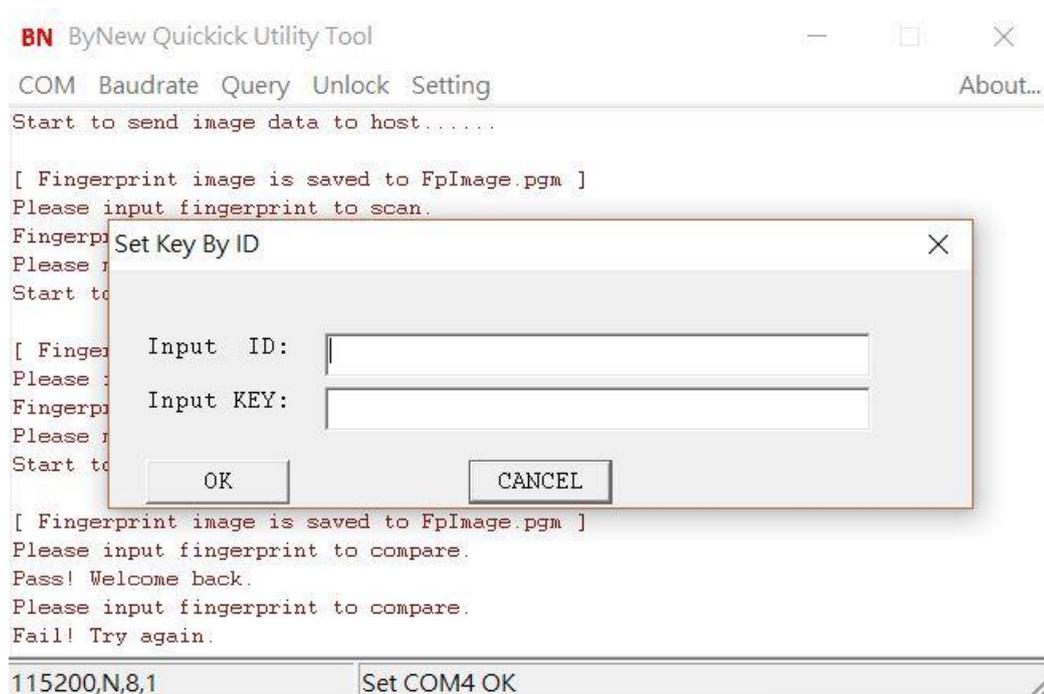
所有的指紋比較，輸出訊息將是 UART 字串的設定，如上圖所示。

## # 使用者安全資料(User Secured Key)

使用者將其安全數據保存在安全區域，用戶安全數據可以是電子郵件帳戶的密碼，保管箱密碼或任何其他重要事項。這裡有 Get Key / Set Key / Clear Key / List All Key / Clear All Key 等功能。



每個安全數據包括“ID”和“KEY”，兩者都是字串格式。



- Get Key  
使用者輸入 ID 以查詢其 KEY。
- Set Key  
使用者使用其 ID 和 KEY 設定新的安全數據。  
例如，使用者可以輸入電子郵件帳戶作為 ID，密碼輸入為 KEY。
- Clear Key

使用者輸入 ID 以清除其安全數據。

- List All Key  
在視窗上列出所有存儲的安全數據。
- Clear All Key  
使用者清除全部安全數據。

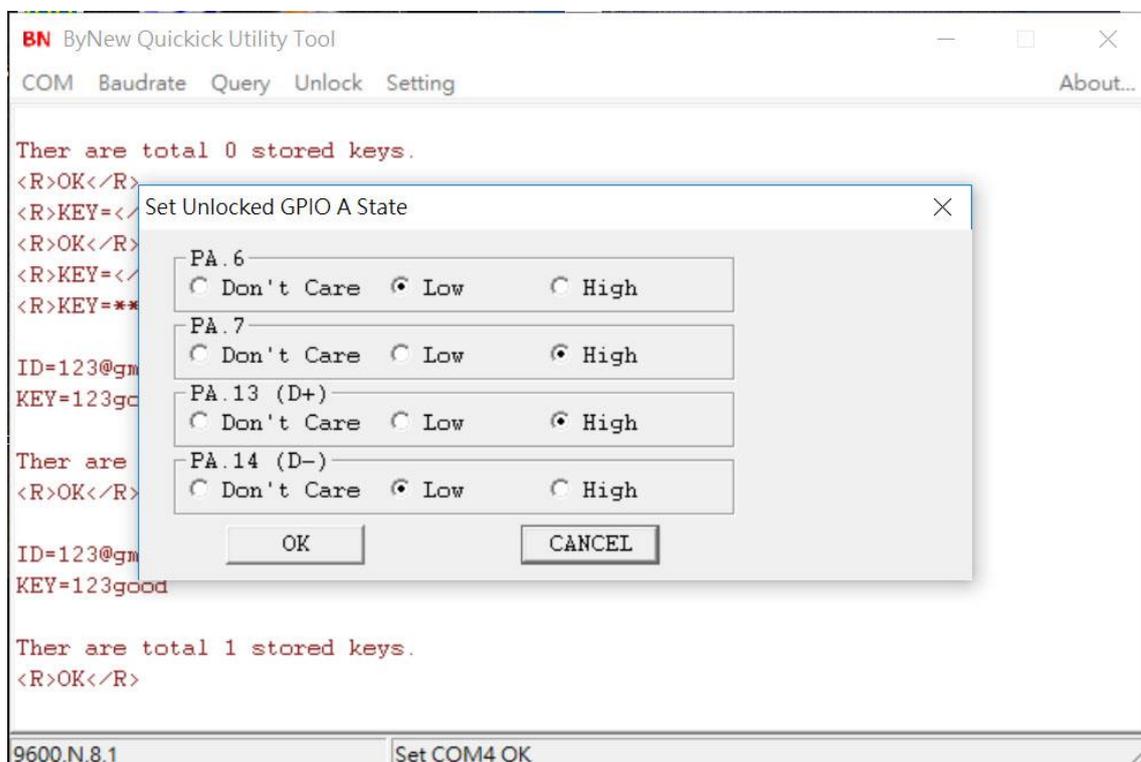
## # 解鎖 GPIO 狀態設置(Set Unlock GPIO State)

當裝置處於解鎖狀態時，它可以設定 GPIO 狀態，當裝置處於鎖定狀態時，它是相反的狀態。當設備處於解鎖狀態時，使用者可以定義 GPIO 是“Don't Care”，Low 或 High。

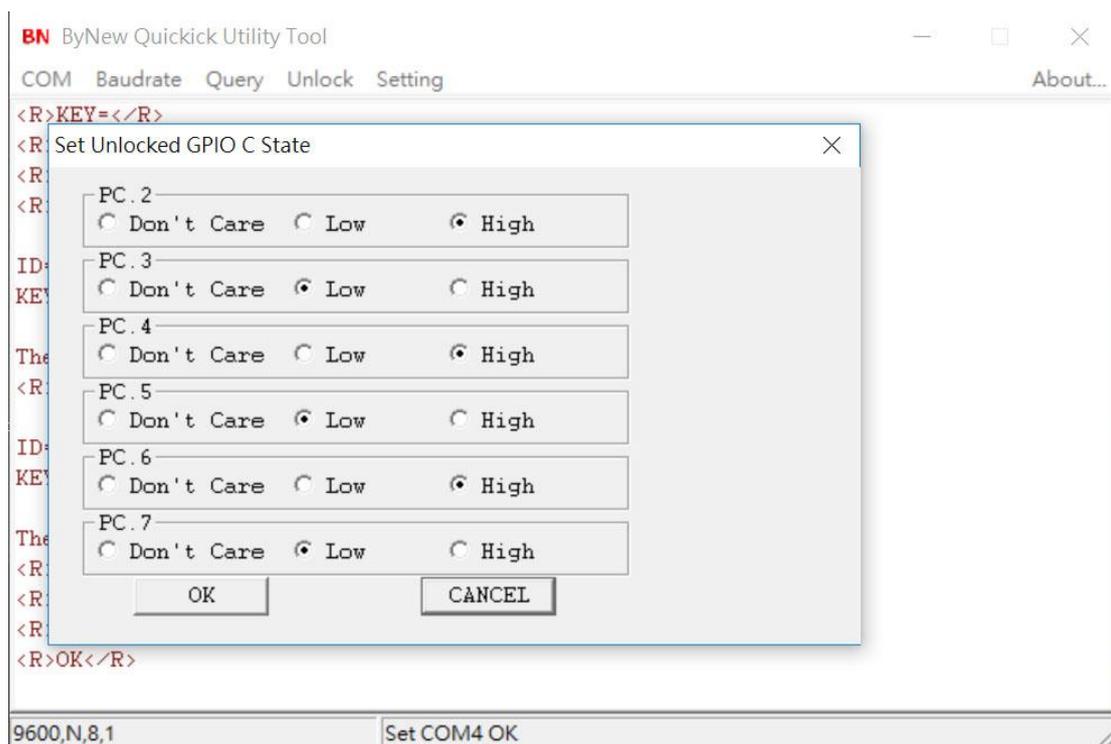
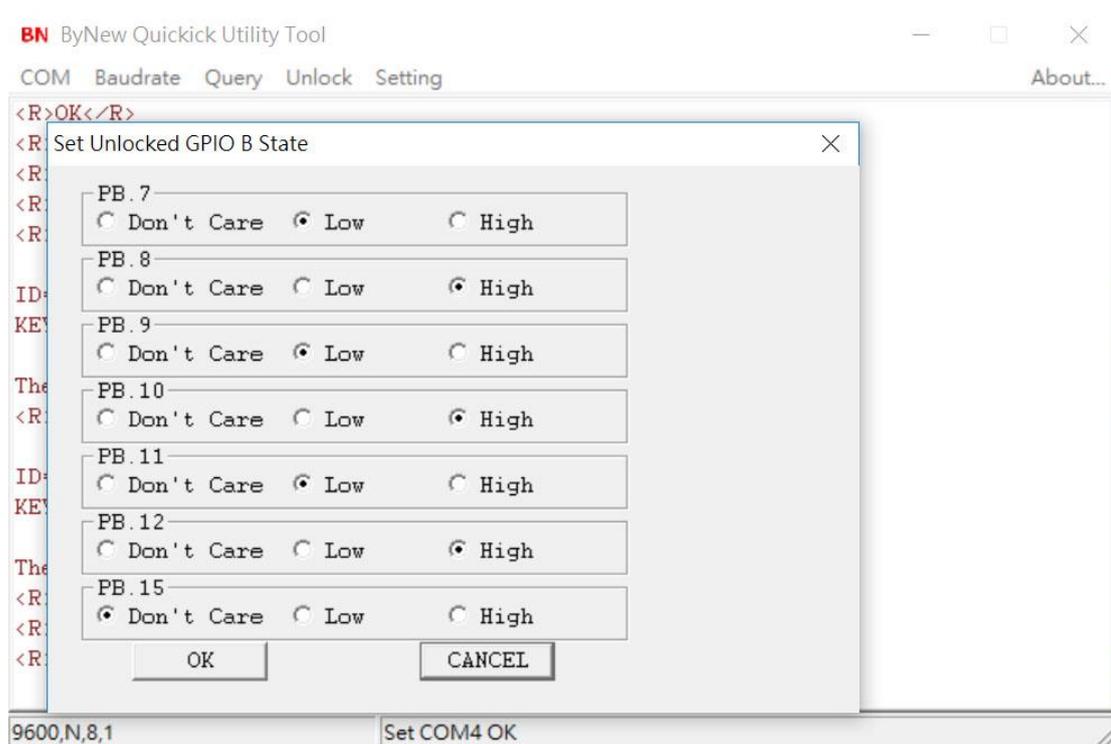
支援 PA、PB、PC 的一些 GPIO

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



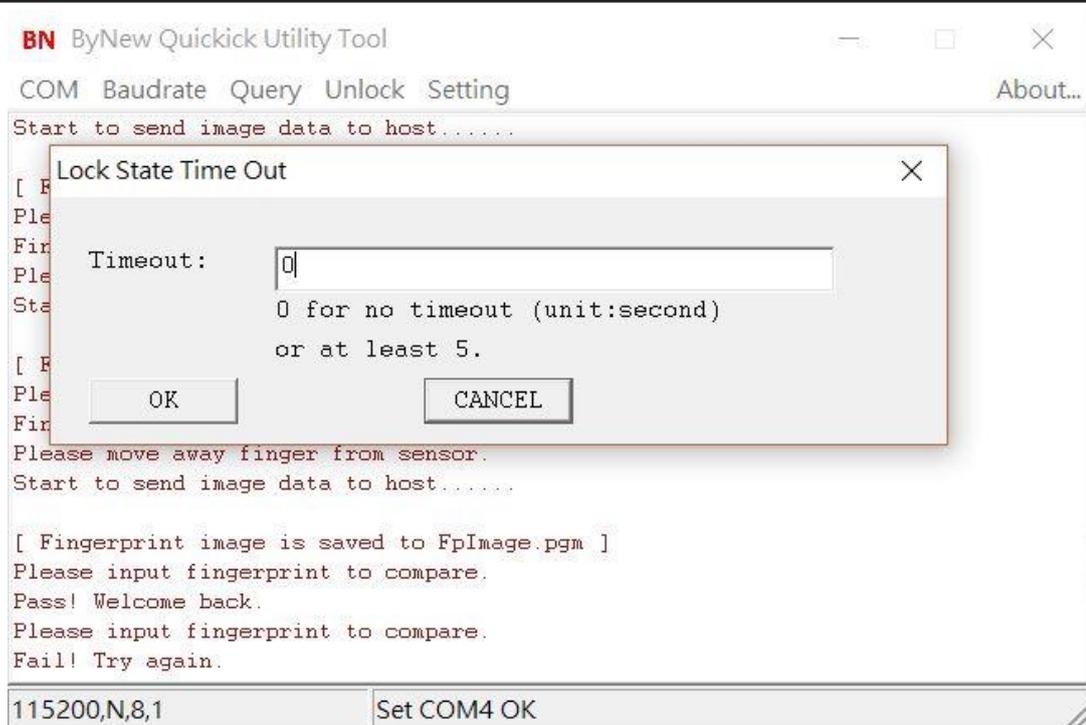
PA.13 和 PA.14 是 USB Bus 的 D+和 D-信號，這裡只需將它們作為 GPIO 輸出接腳。



請注意，並非上述所有 GPIO 都可以設定和工作，請查看裝置電路板以獲取可用的 GPIO。

## # 超時設置(Set Timeout)

為解鎖狀態所設定超時。



當設備處於解鎖狀態並已超時，這時將返回鎖定狀態。

#### # 鎖定(Lock)

選擇執行此功能將使設備立即從解鎖狀態恢復為鎖定狀態。設置選單將顯示為灰色，且解鎖選單再次啟用。使用者在進行任何設置項目之前需要再次解鎖。