

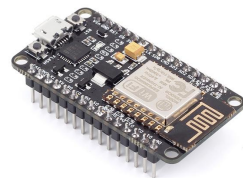


# MODBUS HMI 應用實例

## NodeMCU Wi-Fi

### MODBUS TCP 控制器

(使用 Arduino開發)



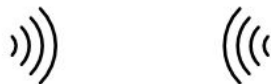
2017/3/31 [元米科技](#)

# 本文目的

以開源的 Arduino 開發環境搭配低廉的 NodeMCU 模組，自製低成本的 Wi-Fi MODBUS TCP 控制器，並透過免費的 MODBUS HMI 方案，自行編輯手機控制介面。從而學會業界常用的 MODBUS TCP 協議，以接軌控制實務，甚至可以搭配溫溼度感測、紅外線控制、PWM 控制等，做出獨一無二的智慧家庭控制系統。



MODBUS HMI



MODBUS TCP



Relay

# 需要設備

1. Android 4.2 以上手機或平板
2. NodeMCU 開發板(拍賣網售價約NT\$120~220)
3. 繼電器模組(選配, 依需求選擇繼電器數量 1~6 個, 拍賣網售價約 NT\$50~200)
4. 5V 手機充電器與充電線(供電給 NodeMCU 與 繼電器模組, 建議充電器應能供電 1.0 A以上, 且充電線應選擇能提供1.0A以上的電源線)
5. Windows 7 /8/10 作業系統電腦

# 需要軟體

1. 手機需安裝 MODBUS HMI APP V3(或較高)版本, 下載處:  
<https://goo.gl/t2g3Gc>
2. 電腦安裝 Arduino 1.6 以上版本, 下載處:  
<https://www.arduino.cc/en/main/software>
3. 電腦安裝 MODBUS HMI 編輯器 V2.0.5.0 以上, 下載  
處:<https://goo.gl/dMciMh>
4. 電腦安裝 ICDT MODBUS TCP Client 工具程式, 下載處:<https://goo.gl/yBLhTV>

# 關於 Arduino

Arduino 是硬體與軟體開放源碼(Open source), 一開始是基於 Atmel AVR 的 MCU (微控制器), 但發展到後來, 許多不同的 MCU 也加支援 Arduino 的開發環境。

Arduino 的開發環境建構容易, 由於開放原始碼在網路上可以獲得許多免費的軟體與學習資源, 而硬體的開源更造就了許多低廉的相容性產品。

關於 Arduino 基本語法類似 C/C++, 可以參考[『小狐狸事務所』](#)的詳盡說明。

或許因為程式碼與硬體來源過多, 因此讓部分控制業界先進有系統不穩定的印象, 其實只要經過可靠的硬體設計與功能驗證, Arduino 也可成為不錯的控制核心。至少在學習與驗證上, Arduino 可以大幅降低進入門檻。

# 關於 MODBUS HMI -1

Modbus HMI 方案為 元米科技 是基於 MODBUS TCP 協議的網頁圖控系統，針對 Android 系統以及其他可執行 JAVA 程式的嵌入式設備開發的低成本、高性能方案。

Modbus HMI 方案的網頁圖控伺服器 採用 Java 程式語言開發，具備高度可攜性，因此可以在 Windows 或 Linux 作業系統的電腦上執行，也可以使用 樹莓派 (Raspberry Pi) 與其他新興低成本高性能的嵌入式 Linux 核心板上順利運作，甚至透過安裝專用 APP 方式在 Android 手機或平板上執行，不需要繁複的網頁伺服器安裝過程，是最佳的嵌入式網頁圖控方案。

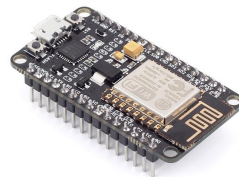
# 關於 MODBUS HMI-2

Modbus HMI 可連接大部分的標準 MODBUS TCP Server 設備，包含 PLC、智慧電表、智慧感測器、智慧驅動器等設備，如果連接設備為 MODBUS RTU 網路，則可以利用市售的 MODBUS Gateway 設備加以轉換。

Android App 版本的 Modbus HMI App 本身具備瀏覽器功能，除了網頁圖控伺服器外，本身也可做為人機介面。

Modbus HMI 在數量 128 點以下可以免費應用於非商業用途，是 MODBUS 標準設備快速完成客製化網頁圖控系統的最經濟選擇。

# 關於 NodeMCU



NodeMCU 是以 ESP8266 Wi-Fi SoC 晶片為基礎的主板，原本使用 Lua 語言開發。在此使用熟悉的 Arduino 環境開發，以獲得更多的應用資源。

利用 NodeMCU 的 Arduino 環境開發不需要外加 Arduino 開發板，只要透過 MicroUSB 線連接 NodeMCU 至電腦，就可上傳程式到 NodeMCU 中獨立運行。

由於 NodeMCU 大量被 Maker 們所使用，因此價位極為低廉，例如本實驗使用的 NodeMCU 主板即以 NT\$120 在 <https://goo.gl/6MEi1d> 購得。您也可以向提供專業技術服務的賣家購買，例如 <https://goo.gl/8FhZR6> 以獲得較好的技術支援。

此實驗利用 NodeMCU 低價、具備 Wi-Fi 功能、支援 Arduino 開發特性，當成 MODBUS TCP Server 控制器，以連接 Relay 模組，直接或定時控制外部設備



# 關於繼電器模組



繼電器(Relay) 是用來將 NodeMCU 主控板 3.3V 的控制信號轉成接點以控制外部設備(例如電燈、馬達等)。控制信將繼電器的電磁線圈激磁使接點閉合，相當於手按電燈開關的動作。繼電器接點必須外加驅動外部設備的電源，因此必須注意繼電器接點容量與電源電壓、電流等與驅動設備的關係。

另外由於驅動電路設計的差異，繼電器驅動可能有高電位驅動或低電為驅動等，依照選用的不同，程式的驅動選項必須隨著改變，本次實驗用的是先前購入的8路 Relay(但實際只用到6路)

(補充說明:附圖皆附有賣家連結網址，僅供參考)



# 電源與連接線

NodeMCU 使用 5VDC 的電源，在沒有額外負載下 0.5A 的電源供應器足可因應，但外加多個 Relay、LED燈、伺服馬達 (Servo motor) 後，則必須納入考量。除了電源供應器，MicroUSB 線徑不足也可能造成壓降過大，影響穩定度。也可考慮使用樹莓派購買專用含線的電源供應器如右圖。

本次實驗的 MicroUSB 自拍賣網購得單價僅 NT\$38，線徑達 24AWG，效果不錯。

(補充說明：附圖皆附有賣家連結網址，僅供參考)



# 動手做之前

本文包含了 MODBUS HMI APP 的編輯設定與 NodeMCU 程式，為了避免篇幅過多，關於 MODBUS HMI APP 的部分，請先參閱“MODBUS HMI APP 應用實作”(<https://goo.gl/vncIEs>) 進行實際驗證。接下來的內容預設讀者已熟悉 MODBUS HMI 操作，重複部分將不再說明。

# 下載並安裝 Arduino IDE

下載位

置:<https://www.arduino.cc/en/Main/Donate>

點選 **JUST DONLOAD** (當然願意捐款更好)

安裝過程中，請選擇安裝所有驅動程式

Arduino - Donate

安全 | <https://www.arduino.cc/en/Main/Donate>

Buy Software Products Learning Forum Support Blog LOG IN SIGN UP

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). Learn more on how your contribution will be used.

SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED **14,551,141** TIMES. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!

\$3 \$5 \$10 \$25 \$50 OTHER

JUST DONLOAD CONTRIBUTE & DOWNLOAD

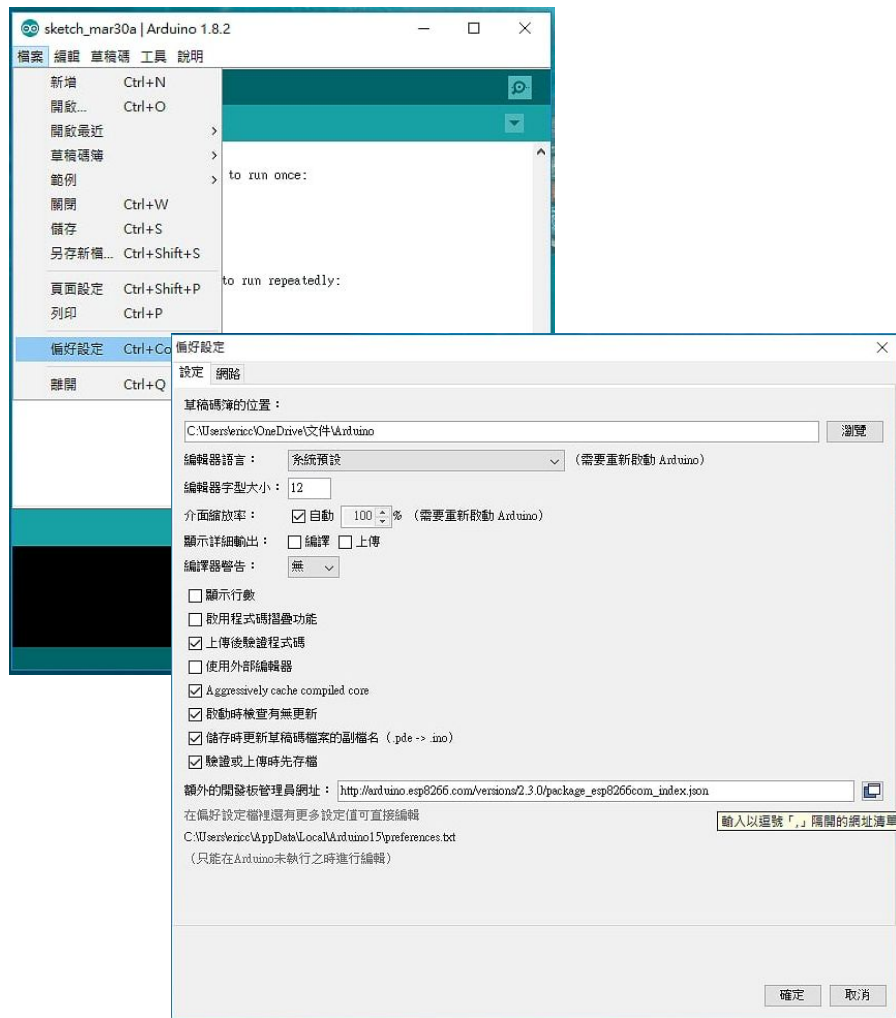
arduino-1.8.2-wi...exe

下午 03:40  
2017/3/30

# 安裝 NodeMCU開發板

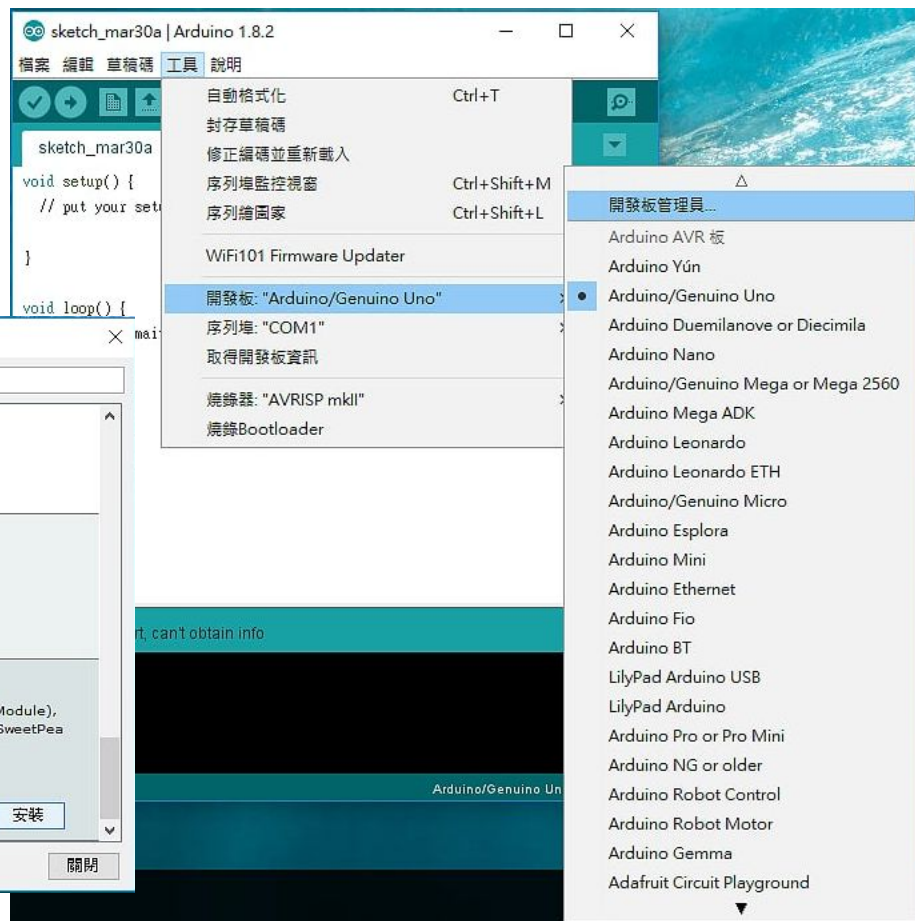
執行程式，選擇“檔案”->“偏好設定”，在額外的開發板管理員網址輸入：

[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json) 按下確定



# 開發板管理員-安裝

選取"工具"->"開發板管理員", 找到  
esp8266 by ... 點選後按下安裝



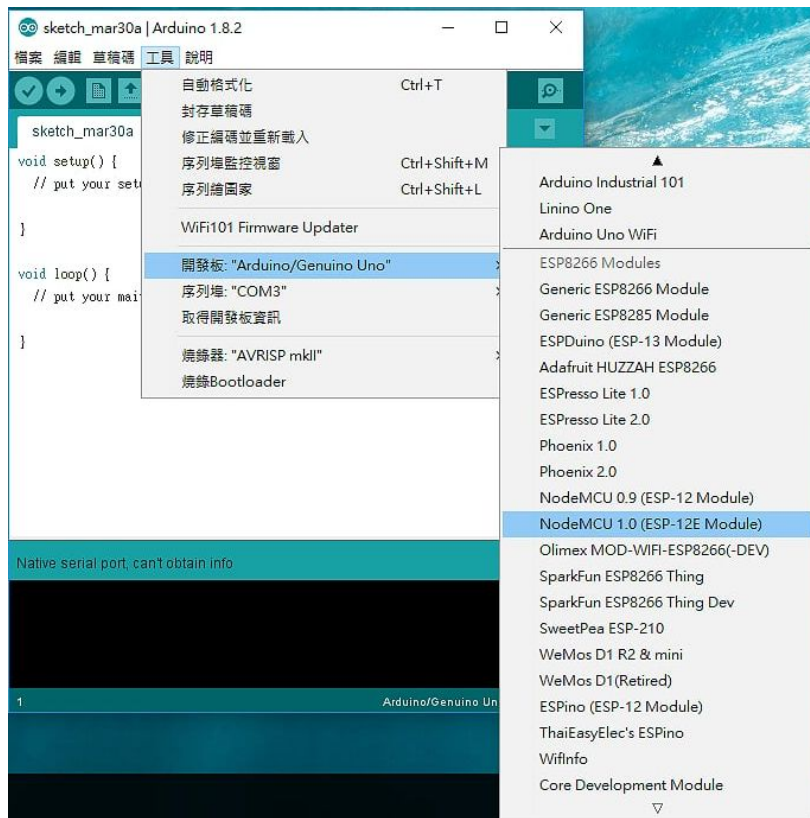
# 將 NodeMCU 連到電腦

將 NodeMCU 以 MicroUSB 線連接到 Arduino IDE 作業電腦，順利的話，可以在 "工具"->"序列埠"找到新的虛擬序列埠，如果無法自動安裝驅動程式，則可能需要至 <https://github.com/nodemcu/nodemcu-devkit/tree/master/Drivers> 下載 CH341SER\_WINDOWS.zip 安裝



# 開發板選擇 NodeMCU


點選“工具”->“開發板”後下拉，應可順利找到 NodeMCU 1.0。





# 下載並開啟程式

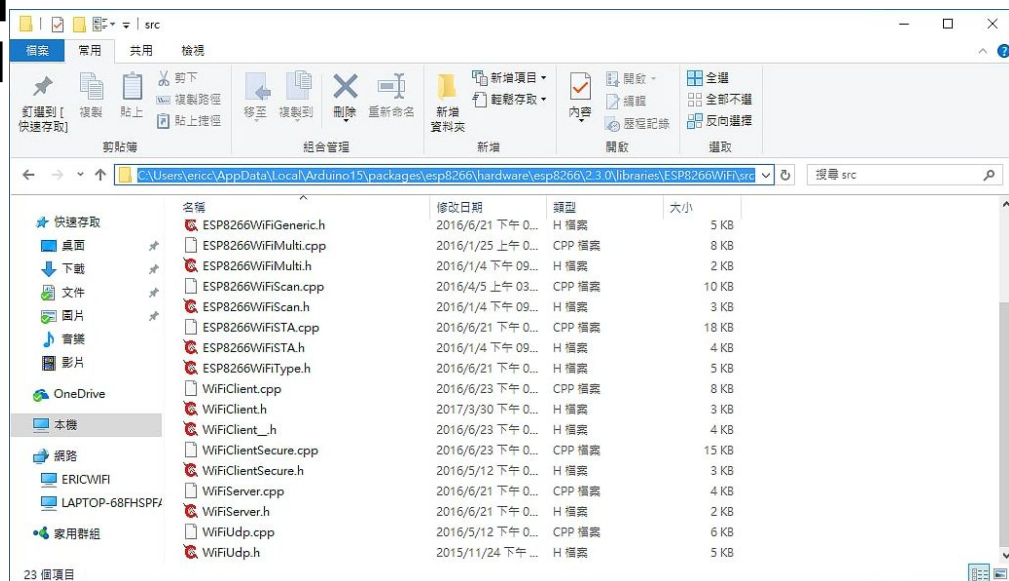
下載 ESP8266MODBUSTCP.ino (<https://goo.gl/ZElAfq>) 程式，並移至桌面或工作路徑，點兩下進行編輯，此時跳出建立同名路徑並移動的提示，選擇"確定"。

在開啟的程式中確認"工具"->"開發板"為NodeMCU 1.0後按下  以嘗試編譯，此時可能出現錯誤訊息，這是因為 ESP8266 程式庫中 WiFiClient.h 有誤，先複製錯誤訊息



# 變更 WiFiClient.h 檔

從複製的錯誤訊息中可以找到 WiFiClient.h 檔案存存放路徑，下載 WiFiClient.h(<https://goo.gl/pOa6el>) 後置換，再次編譯應該可以成功。



錯誤訊息範

例:C:\Users\eric\AppData\Local\Arduino15\packages\esp8266\hardware\esp8266\2.3.0\libraries\ESP8266WiFi\src\WiFiClient.h:127:5: error: request for member 'read' in 'source', which is of non-class type 'unsigned char [256]'

# 修改程式網路參數

程式有兩個地方必須修改，才能順利連線


將 ssid 後的 "mywifi" 依照實際連線的無線 AP 名稱  
;pass 後的 "mypassword" 則改為無線 AP 密碼。

IP 設定的部分，必須將 IP 設定與無線 AP、工作電腦、手機相同的區域網路，如果無法確定應該設的 IP 地址，而單位內有 MIS(網管人員)，可向其洽詢，以避免造成其他設備不能連線。家中網路或許可以利用手機安裝 [Fing](#) 之類 APP 掃描網路，避免用到重複 IP。當然如果有 AP 管理權限者，可查閱並避開 DHCP 的 IP 範圍。

```
const char* ssid = "mywifi";  
const char* pass = "mypassword";
```

```
// IP 設定，必須與 AP 相同區網  
IPAddress myIp(192,168,0,133);  
IPAddress gw(192,168,0,1);  
IPAddress subnet(255,255,255,0);
```

# 編譯並上傳

完成程式變更後按下  重新編譯並上傳，下方顯示傳送進度，傳送完成後 NodeMCU 將重新啟動。

(如果無法順利上傳，請確認開發板選擇是否正確，序列埠選擇是否正確，序列埠選擇是否正確，或驅動程式是否正常安裝)



The screenshot shows the Arduino IDE interface for an ESP8266. The title bar reads "ESP8266MODBUSTCP | Arduino 1.8.2". The menu bar includes "檔案", "編輯", "草叢碼", "工具", and "說明". The toolbar contains icons for "Verify", "Compile and Upload", "Save", "Upload", and "Download". The main editor area shows the following code:

```
ESP8266MODBUSTCP
/*
 * 使用 ESP8266 (NodeMCU) 使用 Arduino 基於 MODBUS TCP 的範例程式
 * 實做了 Function 3/4 & Function 6
 * 具備 64 個 Register(40001-40064)，UID 0-255皆可
 * 範例使用 前 6 個 Register 控制 6 個 Relay，分別接至 GPIO 5,4,2,14,12,13。也就是 NodeMCU 的D1,D2,D4,D5,D6 & D7
 */

#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <stdio.h>

const char* ssid = "ericwifi";
const char* pass = " ";

//modbus 異常代碼
#define ILLEGAL_FUNCTION 1
#define ILLEGAL_DATA_ADDRESS 2
#define ILLEGAL_DATA_VALUE 3
#define STATE_DEVICE_FAILURE 4
```

At the bottom, the upload progress bar is visible, showing the upload is complete with a progress of 100%.

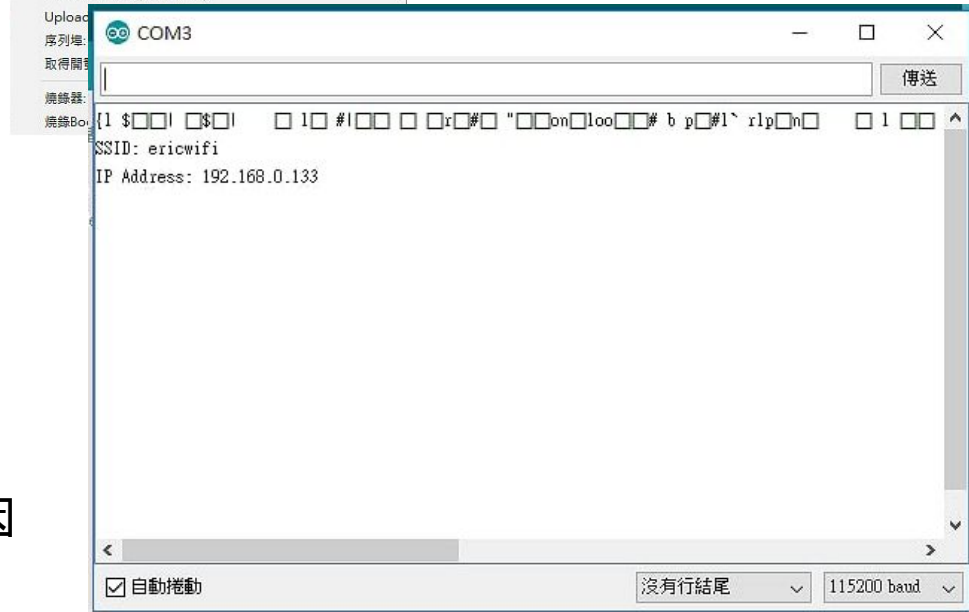
# 開啟通訊視窗

點選"工具"->"序列埠監控視窗", 把速率調整為 115200 baud, 可以看到 SSID:.... IP Address:.... 表示連接到 Wi-Fi 正常。

序列埠速率是 NodeMCU 程式決定的, 查看程式可以找到 `Serial.begin(115200)`。

由於 Arduino 並沒有使用 ICE 除錯, 因此程式的進度與除錯, 可自行透過 `Serial.printf()` 等方式提供除錯資訊。

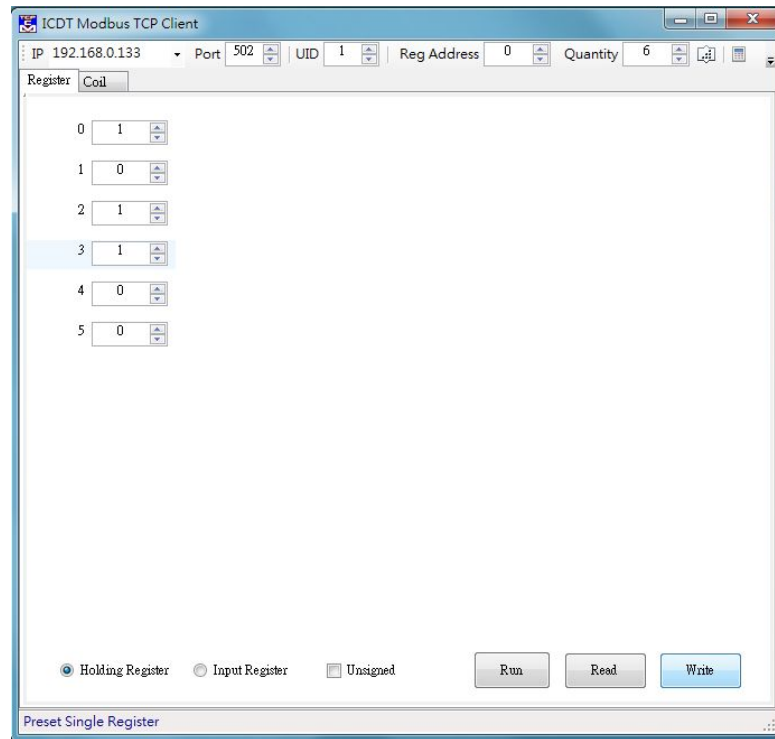
3U5TCP | Arduino 1.8.2



# 以 ICDT MODBUS TCP Client 驗證

電腦安裝 ICDT MODBUS TCP Client 程式(<https://goo.gl/yBLhTV>), 開啟後將 IP 處輸入 NodeMCU 更改後的 IP, 按下 Read 可以正常讀取, 修改後按下 Write 可以改變數值。

注意此處電腦必須與 NodeMCU 位在相同區域網路中(例如 子網路遮罩是 255.255.255.0 時, 每個 IP 前 3 組數值必須相同, 本例中為 "192.168.0")。



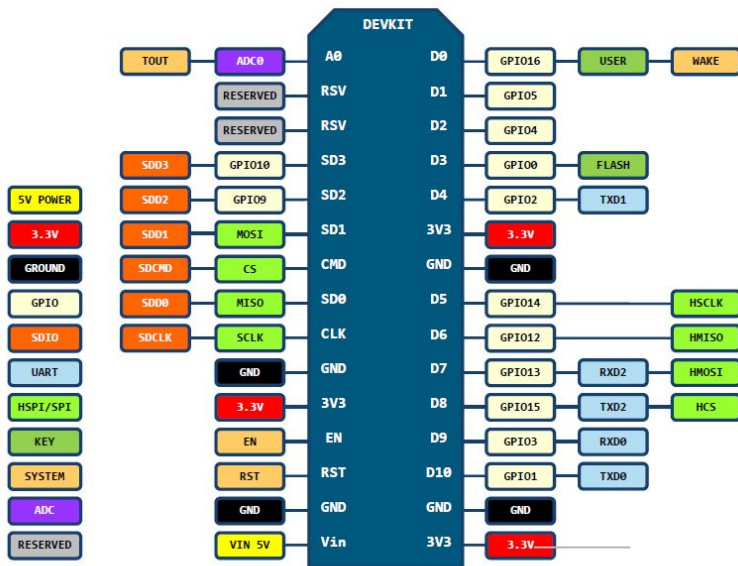
# 程式的 GPIO 輸出

由程式中可以發現輸出腳(GPIO) gpioPin[6] 共使用了 5,4,2,14,12,13, 也就是在 NodeMCU 上面的 D1,D2,D4,D5,D6, 程式中將位置 0~5 的 Register 對應到這 6 個腳位做輸出。

程式中#define OutputReverse 有效時, Register 對應數值 0 GPIO 輸出 3.3V, 數值 1或更大則輸出 0V。否則數值 1 輸出 3.3V, 數值 0 輸出 0V。

```
//定義 Relay GPIO 腳編號  
//相當 NodeMCU D1,D2,D4,D5,D6,D7  
uint8_t gpioPin[6]={5,4,2,14,12,13};
```

## PIN DEFINITION



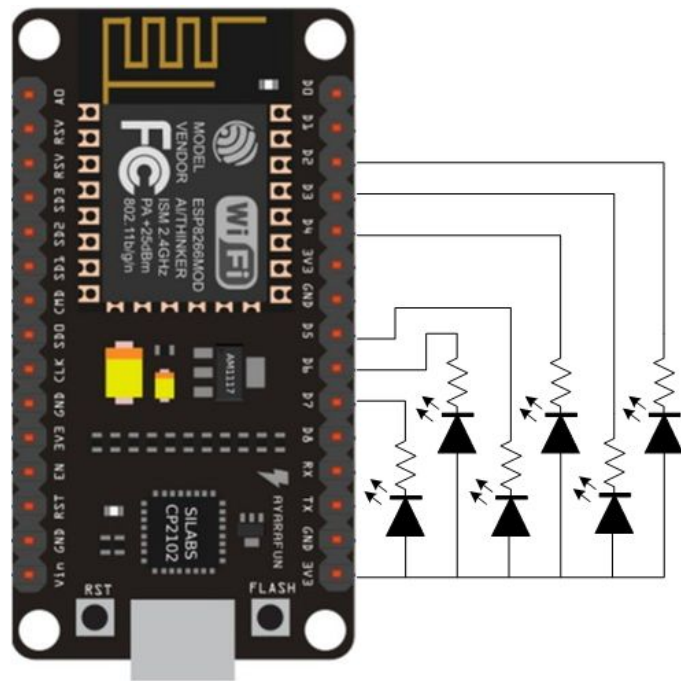
*D0(GPIO16) can only be used as gpio read/write, no interrupt supported, no pwm/i2c/iw supported.*

# 連接 LED 圖例

如果一開始不打算購入 Relay 模組，可以用 6 個 LED 燈串電組取代。電阻 220 歐姆以上，組值越大 LED 亮度越低。

此接線方式為 GPIO 輸出 LOW(0V)時點亮。另一種方式 LED 方向反過來，共電一起接在 GND，則是輸出 HIGH(3.3V)時點亮。如果是共點接地的方式，要將程式 #define OutputReverse 前加 //使其失效

```
//Relay 輸出反向(視 Relay 電路而定)    //Relay 輸出反向(視 Relay 電路而定)
#define OutputReverse                      //#define OutputReverse
```

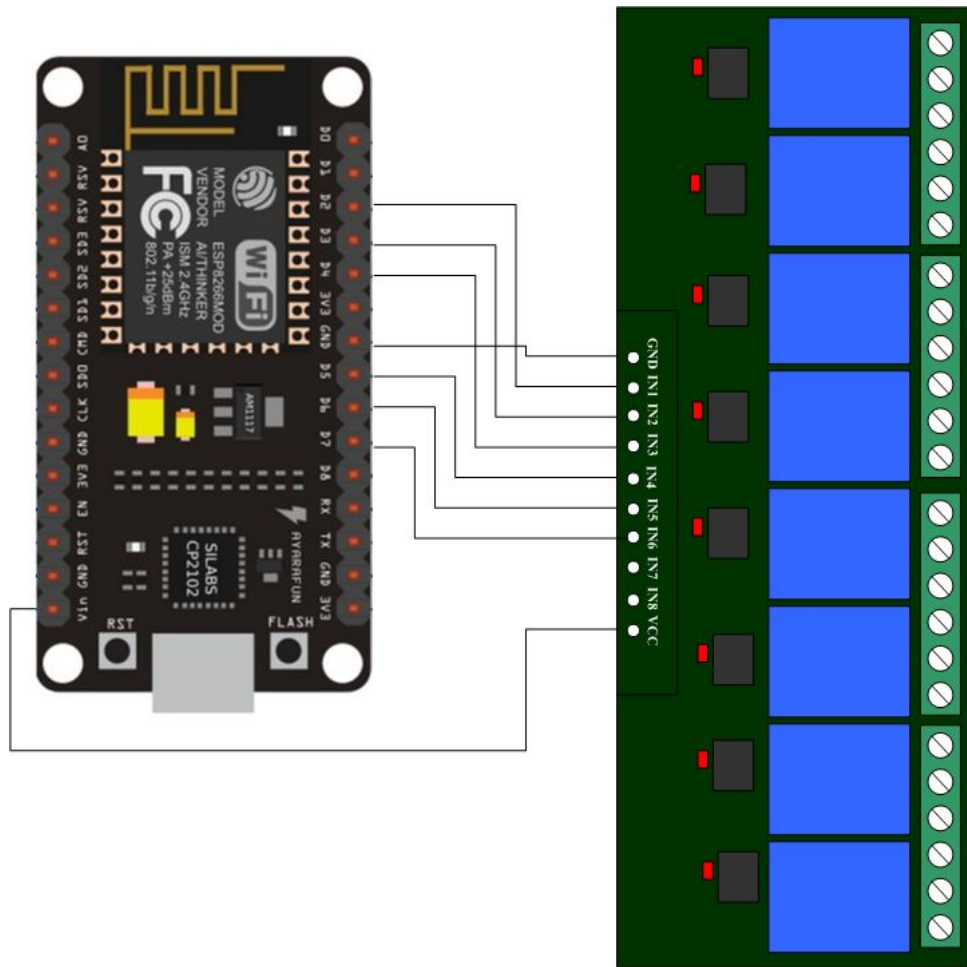




# 接繼電器圖例

右圖為連接 KSRobot 8 路 5V 光耦合隔離記電器模組圖例，此處只接其中前 6 個點。Relay 電源 VCC 連接 Vin 以取用 5V 電源，避免 Relay 動作造成 3V3 電源過大的波動。

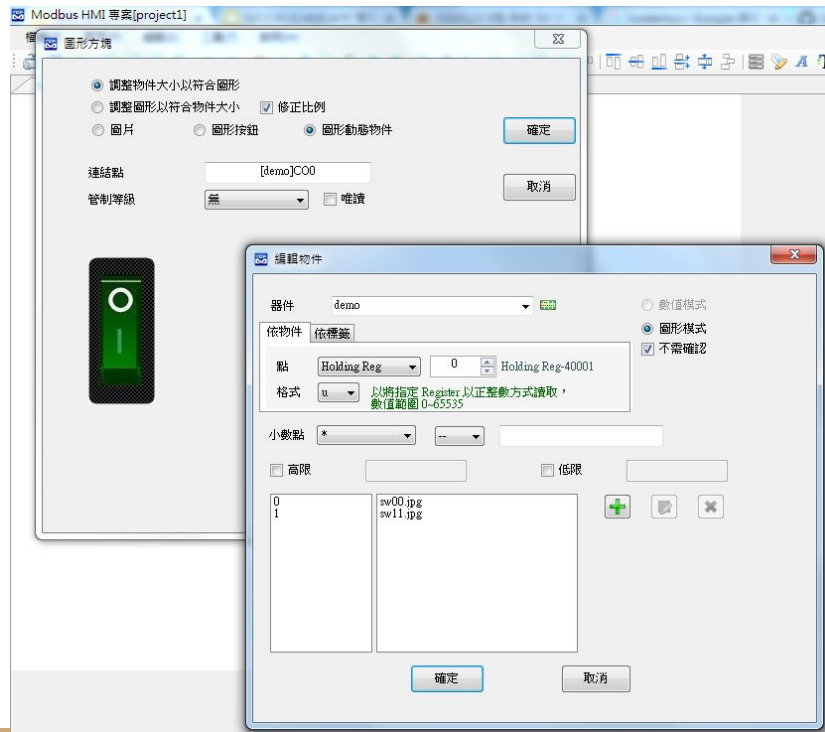
完成接線後送電，重新以 ICDT MODBUS TCP Client 程式測試，可以發現程式中 Register 0~5 寫入 0 與 1 時，Relay 動作將會改變



# 編輯 MODBUS HMI 專案並上傳至手機

參考“MODBUS HMI APP 應用實作”

(<https://goo.gl/vncIEs>), 將設備 IP 改設為 NodeMCU 的 IP, 器件地址(UID)隨意。製作 6 個按鈕, 物件點不使用 Coil, 而改成 Holding Reg 0 ~ Holding Reg 5, 並點選"不需確認"(操作時不彈出確認視窗)。以電腦執行確認新增的按鈕可以正常操作 Relay 或 LED 燈後, 再依照說明將檔案複製到手機中, 使 MODBUS HMI APP 直接操作 Relay 動作。



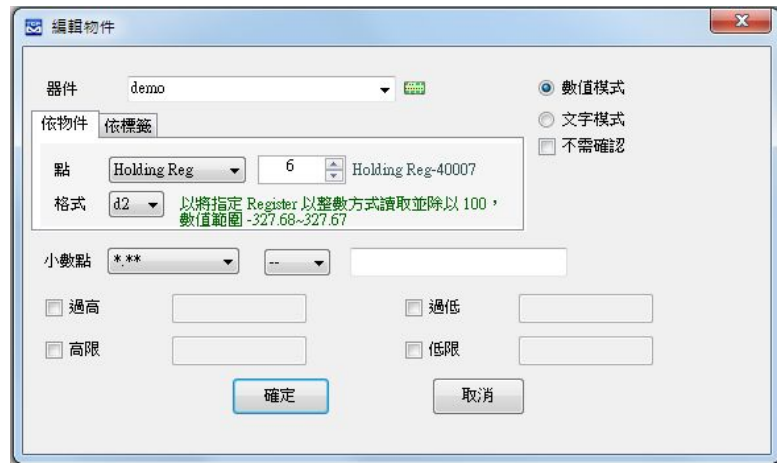
# 補充說明-1

- 此 NodeMCU 的 MODBUS TCP Server 範例同一個時間只能一個 TCP Client 連接, 因此不能同一時間多個 MODBUS HMI APP 同時連線。
- MODBUS HMI APP 限制 128 點, 但沒限制 TCP Server 的數量, 因此可以同時控制多個 NodeMCU。
- 為了簡化MODBUS 範例程式碼, 只完成讀取 Register 的 Function 3 與單點寫入 Register 的 Function, 其餘 Function 都將回 ILLEGAL FUNCTION 的異常代碼。可自行參考 MODBUS 協議標準增加其餘功能。
- NodeMCU 上的 LED 是接到 D0 (GPIO16), 可以看到程式多次對該位置進行操作 (digitalWrite)。如果網路連線, 但沒有 Client 連接, 將亮兩秒滅兩秒; 有 Client 連接則快速閃爍。

```
//NodeMCU LED 腳位  
#define red_LED 16
```

# 補充說明-2

- 除了直接使用手機 APP，以可以用樹莓派做為 Web Server，可以參考這篇：<https://goo.gl/T88nlb>



- 如果對 Arduino 有更多心得，可以外接感測器並依相同方式，把溫、溼度感測值存入程式 Register 陣列中，就可以變成 MODBUS TCP 的溫溼度感測器。但由於 Register 內只能存整數，因此可以先將數值 \*100 後存入，並且於 MODBUS HMI 物件編輯時，格式選擇 d2，小數點選擇 \*.\*。如此就可以得到小數點後兩位的感測值(如果感測器解析度達到0.01的話)

# 更多進階功能與應用

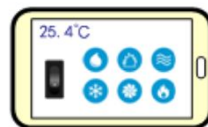
MODBUS HMI 有許多進階花俏的用法，等待您發揮創意。Arduino的想像空間更是無限。動手實現您的想像，未來在您手中。

(本文歡迎學界或業界做為學習教材，部分圖片取材自網路，如有侵權疑慮請告知)

Intelligent Control Design & Technology Co., Ltd <http://www.icdt.com.tw>

Based on JAVA environment 128 points free MODBUS TCP Web HMI

Check Box



MODBUS HMI

MODBUS TCP



MODBUS TCP

MODBUS TCP



PLC



PLC



PLC



Online Menu



WYSIWYG Project Builder



Button

Image

Text

Frame

Password

Registers

Coils