

iParking

此章節解說智能停車格(iParking)的實作方法，應用於停車位上偵測車輛距離小於30 CM時，以紅燈表示；以綠燈表示，並回報資訊給中控中心。

如果對於建置環境不了解，先參考「NodeMCU_HelloWorld」章節。

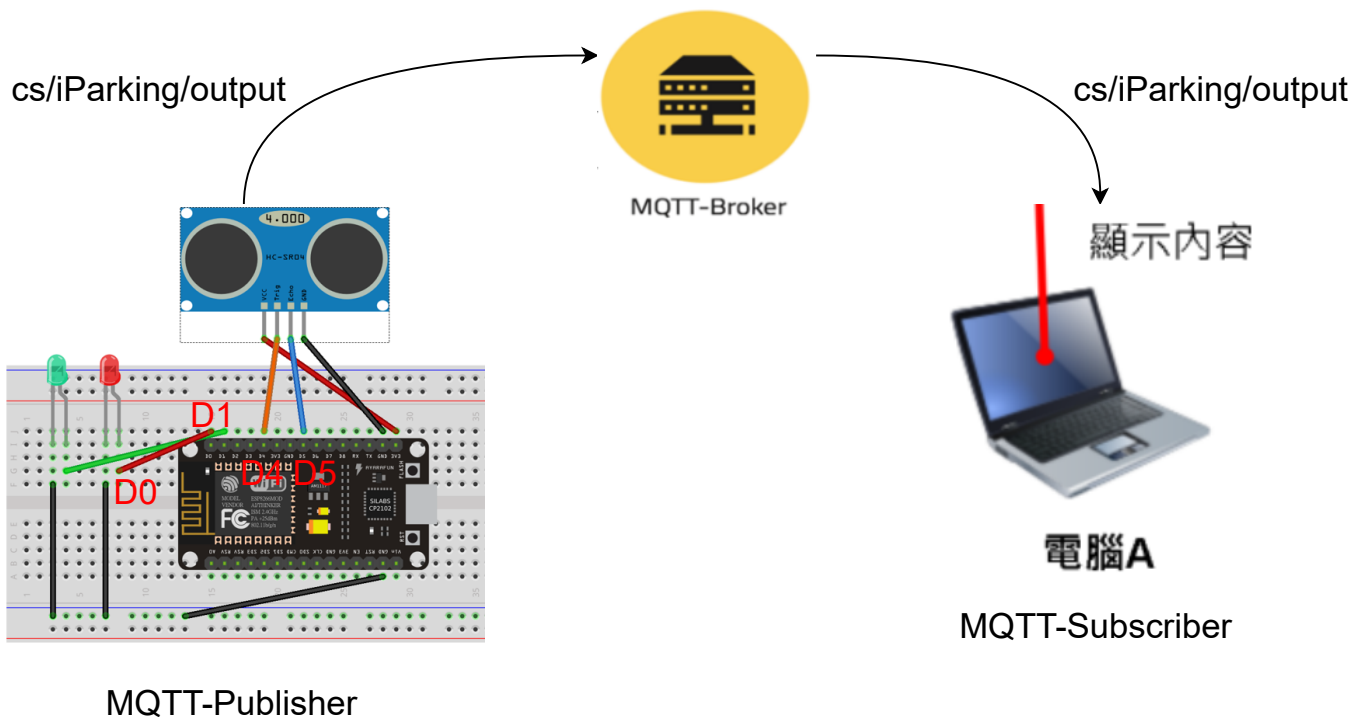
如果對於HC-SR04不了解，先參考「HC_SR04」章節。

如果對於LED不了解，先參考「LED」章節。

如果對於MQTT不了解，先參考「NodeMCU_MQTT」章節。

架構圖：

- (1) 將 VCC 與 GND接上
- (2) 發送聲波接在 D4 Pin 腳，接收聲波接在 D5 Pin 腳
- (3) 紅燈正極接在 D0 Pin 腳
- (4) 綠燈正極接在 D1 Pin 腳



上圖，NodeMCU為Publisher角色，電腦A為Subscriber角色。NodeMCU發送的topic為cs/iParking/output，反之電腦A接收的topic為cs/iParking/output

架構介紹：

1. NodeMCU連上WiFi AP(2.4G only)，也連上MQTT-Broker
2. NodeMCU每2秒發送訊息至topic(cs/iParking/output)
3. 電腦A連上WiFi AP，也連上MQTT-Broker
4. 電腦A接收topic(cs/iParking/output)，知道該停車位狀態

1. 原理解說

HC-SR04 透過 40kHz的超聲波運作，此頻率人類無法聽到。當 Trigger 發送超聲波後，超聲波遇到障礙物(**Object**)而反射，此時Echo接收到反射的超聲波，得到測量時間(**Duration**)。我們知道聲音的速度約 340M/sec，因此可從公式得出聲音走的距離(**Sonar Distance**)：

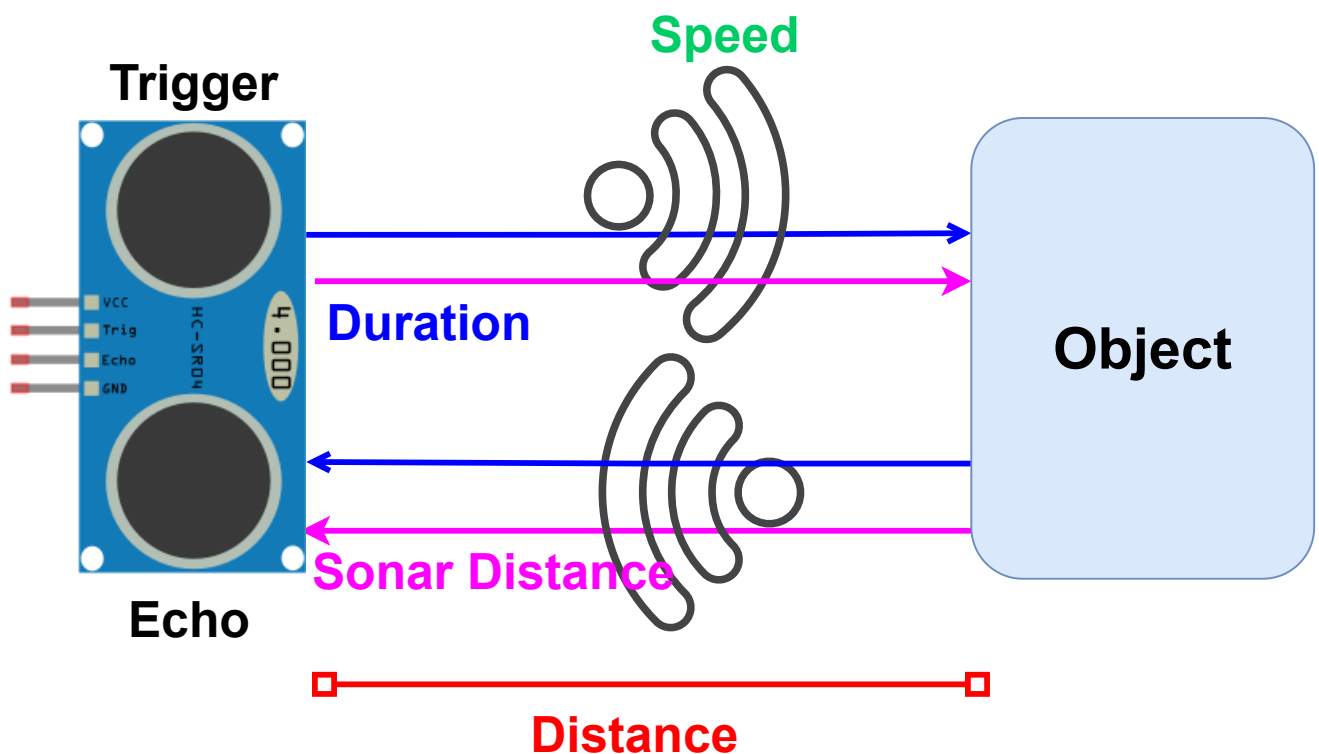
$$\text{Sonar Distance} = \text{Speed} * \text{Duration}$$

進一步得到測量距離(**Distance**)：

$$\text{Distance} = \text{Sonar Distance} / 2$$

公式總結為：

$$\text{Distance} = \text{Speed} * \text{Duration} / 2$$



2. 應用說明

從步驟1可得到感測HC-SR04與障礙物的距離，設計出每2秒偵測與障礙物的距離，距離單位為公分(CM)，並設定當距離小於 30 CM時，紅燈亮起；反之綠燈亮起。
紅燈代表：車位有車 / 離障礙物很近。
綠燈代表：車位無車 / 離障礙物很遠。

3. 編寫草稿碼 -> 上傳至 NodeMCU 開發板

```
iParking | Arduino 1.8.14
檔案 編輯 草稿碼 工具 說明

iParking
7 #include <ESP8266WiFi.h>
8 #include <PubSubClient.h>
9
10 #define MSG_BUFFER_SIZE (1024)
11
12 //ESP8266 名稱
13 #define DUTNAME "iParking"
14
15 //LED
16 const int RED_LED = D0; //D0 (1)
17 const int GREEN_LED = D1; //D1
18
19 //距離感測器 (HC-SR04)
20 const int TRIGGER = D4; //D4 (2)
21 const int ECHO = D5; //D5
22
23 //回報間隔
24 const int report interval = 2000; (3)
25
26 //mac string
27 char dut_mac_str[17+1] = {0};
28
29 //WIFI網路的 SSID, 密碼
30 const char* ssid = "your_wifi_ssid"; (4)
31 const char* password = "your_wifi_key";
32
33 //Third-party MQTT Broker Domain Name & Port
34 const char* mqtt_server = "broker.emqx.io"; (5)
35 const int mqtt_port = 1883;
36
37 //MQTT topic & message
38 char send_topic[]="cs/"DUTNAME"/output";
39 char msg[MSG_BUFFER_SIZE];
40
41 //Distance threshold (6)
42 int VALID_DIS = 30;
43
44 WiFiClient espClient;
45 PubSubClient client(espClient);
46
47 <
上傳完畢。
Leaving...
Hard resetting via RTS pin...
```

- (1) 定義紅綠燈pin腳為 D0、D1
- (2) 定義HC-SR04 trigger pin腳為D4、echo pin腳為D5
- (3) 設定回報間隔時間，單位毫秒(ms)
- (4) 設定WiFi SSID與密碼
- (5) 設定MQTT-Broker、Port、及發送的topic
- (6) 定義與物體的距離，單位公分(cm)

4. 觀看結果

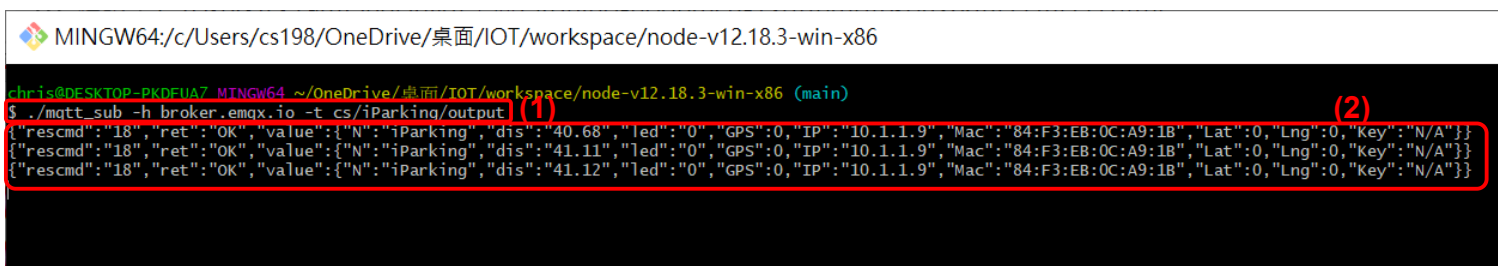
4.1 命令列



```
COM7
.....WiFi connected, IP:10.1.1.9 (1)
Attempting MQTT connection...
client_id:ESP8266Client-aaa4
MQTT connected (2)
Publish message: {"rescmd":"18","ret":"OK","value":{"N":"iParking","dis":"41.38","led":"0","GPS":0,"IP":"10.1.1.9","Mac":"84:F3:EB:0C:A9:1B","Lat":0,"Lng":0,"Key":"N/A"}} (3)
Publish message: {"rescmd":"18","ret":"OK","value":{"N":"iParking","dis":"0.00","led":"2","GPS":0,"IP":"10.1.1.9","Mac":"84:F3:EB:0C:A9:1B","Lat":0,"Lng":0,"Key":"N/A"}}
Publish message: {"rescmd":"18","ret":"OK","value":{"N":"iParking","dis":"39.97","led":"0","GPS":0,"IP":"10.1.1.9","Mac":"84:F3:EB:0C:A9:1B","Lat":0,"Lng":0,"Key":"N/A"}}
Publish message: {"rescmd":"18","ret":"OK","value":{"N":"iParking","dis":"0.00","led":"2","GPS":0,"IP":"10.1.1.9","Mac":"84:F3:EB:0C:A9:1B","Lat":0,"Lng":0,"Key":"N/A"}}
Publish message: {"rescmd":"18","ret":"OK","value":{"N":"iParking","dis":"41.12","led":"0","GPS":0,"IP":"10.1.1.9","Mac":"84:F3:EB:0C:A9:1B","Lat":0,"Lng":0,"Key":"N/A"}}
Publish message: {"rescmd":"18","ret":"OK","value":{"N":"iParking","dis":"40.66","led":"0","GPS":0,"IP":"10.1.1.9","Mac":"84:F3:EB:0C:A9:1B","Lat":0,"Lng":0,"Key":"N/A"}}
Publish message: {"rescmd":"18","ret":"OK","value":{"N":"iParking","dis":"40.68","led":"0","GPS":0,"IP":"10.1.1.9","Mac":"84:F3:EB:0C:A9:1B","Lat":0,"Lng":0,"Key":"N/A"}}
Publish message: {"rescmd":"18","ret":"OK","value":{"N":"iParking","dis":"41.14","led":"0","GPS":0,"IP":"10.1.1.9","Mac":"84:F3:EB:0C:A9:1B","Lat":0,"Lng":0,"Key":"N/A"}}
[ ] 自動捲動 [ ] Show timestamp NL(newline) 115200 baud Clear output
```

- (1) NodeMCU 連上 WiFi，且拿到IP
- (2) NodeMCU 連上 MQTT-Broker
- (3) 每2秒發送訊息至 topic(cs/iParking/output)

4.2 電腦A - Git Bash介面



```
MINGW64:/c:/Users/cs198/OneDrive/桌面/IOT/workspace/node-v12.18.3-win-x86
chris@DESKTOP-PKDEUA7 MINGW64 ~/OneDrive/桌面/IOT/workspace/node-v12.18.3-win-x86 (main)
$ ./mqttsub -h broker.emqx.io -t cs/iParking/output (1)
{"rescmd":"18","ret":"OK","value":{"N":"iParking","dis":"40.68","led":"0","GPS":0,"IP":"10.1.1.9","Mac":"84:F3:EB:0C:A9:1B","Lat":0,"Lng":0,"Key":"N/A"}} (2)
{"rescmd":"18","ret":"OK","value":{"N":"iParking","dis":"41.11","led":"0","GPS":0,"IP":"10.1.1.9","Mac":"84:F3:EB:0C:A9:1B","Lat":0,"Lng":0,"Key":"N/A"}}
{"rescmd":"18","ret":"OK","value":{"N":"iParking","dis":"41.12","led":"0","GPS":0,"IP":"10.1.1.9","Mac":"84:F3:EB:0C:A9:1B","Lat":0,"Lng":0,"Key":"N/A"}}
```

- (1) 用Node.js - mqtt 工具接收 topic(cs/iParking/output)訊息
- (2) 接收到的訊息內容帶有dis欄位，代表與車輛的距離