修平科技大學
資訊網路技術系

實務專題報告

物聯網之密室逃脫

賴昱辰
張世育
吳原程
蔡俊軒

指導老師：張瑞淇

修 平 科 技 大 學

資訊網路技術系

物聯網之密室逃脫

指導老師： 張瑞淇老師

學生：BN104102 賴昱辰

BN104099 張世育

BN104081 吳原程

BN104078 蔡俊軒

中華民國 108 年 1 月 9 日

修 平 科 技 大 學

資訊網路技術系

物聯網之密室逃脫

指導老師： 張瑞淇老師

學生：BN104102 賴昱辰

BN104099 張世育

BN104081 吳原程

BN104078 蔡俊軒

指導老師：__________老師

評審老師：__________老師

__________老師

__________老師

中華民國 108 年 1 月 9 日

摘要

近年來物聯網發展日趨成熟， 主要架構為感知、網路、應用，透過各種不同的感知設備（sensors）即時感知物理的狀態，並傳輸至網際網路來實現智慧化識別、定位、跟蹤、監控和管理。以現在熱門的資訊話題物聯網作為本專題的思考大方向。利用多元的網路傳輸技術及通訊元件結合，透過 Arduino 週邊套件包含 LCD 顯示器、超音波感測器、紅外線感測器、伺服馬達將結果表現出來，提供各種智慧物聯應用，以期各應用領域在採用後能達到簡化流程、降低成本、提升安全及節省能源等目的。經由此專題的研究，我們已有累積建置基本物聯網所需的基本技術能力，把相關技術應用在實際的生活與娛樂，可更加便利。

目　　錄

圖 目 錄

表 目 錄

第 1 章 緒論

1.1 研究動機

在實景遊戲中，密室逃脫一般指一種單人或多人在特定場所裡進行娛樂活動。參與這項活動的玩家，一般會被置身於在一個特定的場所，通過裝修與設計，營造逼真的場景，而後賦予玩家不同的身份、任務、及故事劇情，要求玩家在規定的時間內，通過尋找線索、團隊合作、層層解謎，最終完成任務脫離密室。

由於網路時代物聯網新起，手機娛樂性能愈發豐富，想要從現實中，改變設計成我們的手機遊戲。讓更多想玩密室逃脫的玩家能輕鬆的就能闖關。

1.2 研究目的

透過網路的溝通橋樑，來實現物體與物體的連結進而實現物聯網的應用。本專題研究目的包含以下三點

（一）了解物聯網的技術基礎

（二）Arduino 與樹莓派之間的溝通

（三）探討物聯網之密室逃脫遊戲應用

1.3 研究方法

本專題採用 Arduino 與樹莓派 Raspberry Pi 設計闖關遊戲之密室逃脫。系統架構如圖 1 所示，主要元件由伺服馬達、紅外線及超音波感測器。由於 Arduino 沒有無線通訊模組因此無法直接與外界進行連線，我們採用的 ESP8266 NodeMCU 無線模組，經無線網路閘道器來實現 Arduino 所有對外的訊息與樹莓派 Raspberry Pi 做配對溝通。

樹莓派 Raspberry Pi 需架設 MQTT、NodeRed 及 Blynk Server 來實現與手機、平板連線之溝通橋梁。

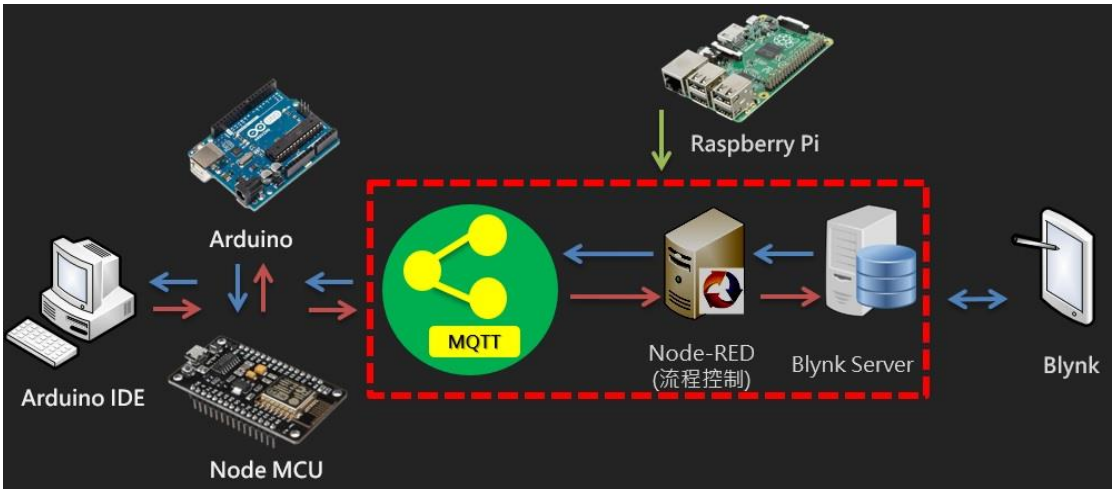


圖 1-1：密室逃脫系統架構流程圖

1.3.1 製作流程

硬體：

構想出 3D 立體圖並繪製後，即可使用 CNC 銑床技術完成，本專題是使用電木材質(不導電、材料價格較便宜)。

電路板主要擴充 NodeMCU。設計使用 Fritzing 軟體做設計電路圖，以顯影劑、蝕刻液及檯燈曝光，來完成電路板。

密室門設計以冰棒棍及丁雙，以熱熔槍黏貼後即可完成。

LCD 顯示器需每塊焊接 I2C 介面(採用 Arduino 的 Scoop 支援多通函式庫）。

ESP8266 NodeMCU 為了提供 Arduino 網路功能。必須先架設好無線基地台。

燈條為密室的外觀布置，使用 LED12V 測發光燈條。

穩壓電源模組，提升穩定密室中所有電力系統。

軟體：

以 Arduino IDE 做為程式撰寫

開發：

Arduino(LCD、伺服馬達、超音波、紅外線)、ESP8266 NodeMCU，使用 Arduino IDE 做編寫。

樹莓派必須先架設 MQTT、NodeRed 及 Blynk Server。

APP Blynk 做創立帳戶後以私有主機登入，創立新專案，選擇設備及連線方式，成功後會寄一組 auth 到信箱。這組 auth 運用於 NodeMCU 及 Node-RED 控制流程

第 2 章 硬體開發簡介

2.1 Arduino

2.1.1 Arduino 介紹

Arduino 是一種開放授權的互動環境開發技術，互動裝置無所不在，簡易 I/O 介面板，並具有使用類似 Java、C 語言的程式開發環境。

2.1.2 Arduino 具備優點

軟體開放源碼(open-source)，硬體也是開放。開發環境可在官方免費下載，而電路設計圖也可從網站下載，依需求進行修改。開發環境下提供了大量的基礎函數。學習門檻較為簡單，很容易學會相關互動裝置的開發。由於 Arduino 以公開共享為基礎，多數人都樂於分享自己的的創品，網路上能找的創作案子非常豐富。以此會基礎，有時只需要參考分享者的作品，依據自身的需求行調整，就可以在短時間內完成自己的創作。

2.1.3 Arduino 種類

Arduino 種類繁多，各有各的應用，主要包括：Arduino Uno 、Arduino Mega、Arduino Duemilanove 、Arduino Fio、Arduino Nano 3.0 、Arduino Mini 、Arduino BT (BlueTooth) 。

2.1.4 Arduino UNO 硬體介紹

微控制器	ATMEGA328
處理時脈速度	16 MHz
工作電壓	5V
輸入電壓	7-12V
數字 I / O 接腳	14（6 個提供 PWM 輸出）

表 2-1：Arduino 規格圖

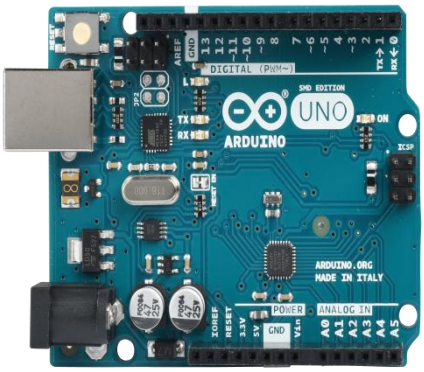


圖 2-1：Arduino 外觀

2.1.5 Arduino 硬體功能說明

(一)　數位接腳

有 14 支數位 I/O 接腳可以當作 input 使用，也可以當作 output 使用，使用方法是透過 pinMode(), digitalWrite(), anddigitalRead() 這幾個函式。

14 支數位 I/O 接腳，其中幾支腳有特殊的功能:

Serial 通訊	0(RX) 和 1 (TX) 這兩支腳。用來接收(RX)與傳輸(TX) TTL 訊號的序列資料。這兩支腳也連接到 USB Converter 晶片中。
外部中斷	2 和 3 這兩支腳。這兩支腳可以利用外部事件觸發中斷。詳細內容請參考 attachInterrupt() 函式。
PWM	3, 5, 6, 9, 10 和 11 共六支腳。透過 analogWrite() 函式可以提供 8-bit 的 PWM 輸出。
SPI	10 (SS), 11 (MOSI), 12 (MISO) 和 13 (SCK) 這四支腳。這四支腳搭配 SPI Library 可提供 SPI 序列通訊，乙太網卡與 SD 卡都需要使用 SPI 協定
LED	13。內建一顆 LED，當 pin 腳為 HIGH 時，LED 打開，當 pin 腳為 LOW 時，LED 關閉。

表 2-2：Arduino 數位特殊接腳

（二） 類比接腳

Arduino UNO 的類比接腳除了可以當類比輸入外，也可以其他特殊用途，最重要的就是支援 I2C 協定。I2C 只需要兩支腳位就可以讓許多週邊設備透過並 聯建立通訊，這兩支腳位一個叫作 SDA (Serial DAta line)，另一個叫作 SCL (Serial CLock Line)。在大部份的 Arduino 板子上，SDA 是在 A4，而 SCL 是在 A5，這兩根接腳需要接 10k 歐姆提升電阻以確保電流的驅動力。在此專題中，所 有的液晶螢幕 LCD1602，都是透過 I2C 界面並聯。

2.2 ESP8266、樹莓派

2.2.1 ESP8266 開發

較簡單開發方式使用 Arduino IDE 軟體來開發 ESP8266 的執行程式，語法可相容使用，如 pinMode、digitalRead、digitalWrite 等。ESP8266 的軟體開發套件源頭就自 GCC 編譯器，C 語言可以提供更接近硬體層的低階控制。

2.2.2 ESP8266(Node MCU)硬體功能說明

Arduino 本身沒有 Wi-Fi 功能，但 ESP8266 有 Wi-Fi 功能，Arduino 需要進行 Wi-Fi 無線收發時，可以透過 ESP8266 來實現。把 ESP8266 當 Arduino 的 Wi-Fi 擴充卡來用，但是在本專題中，NodeMCU 與 Arduino 是獨立運作，Arduino 透過一些 I/O 接腳與 NodeMCU 交換訊息，Node MCU 再與雲端的伺服器交換訊。

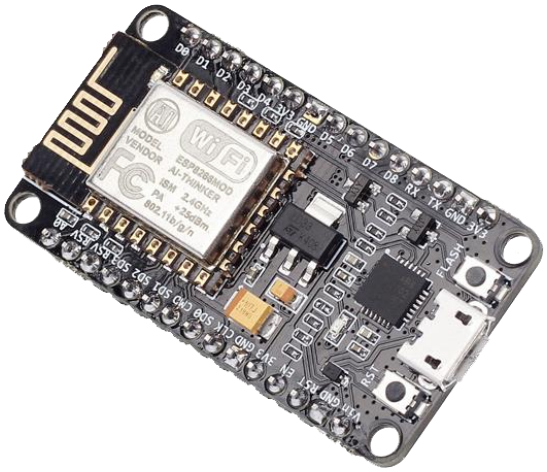


圖 2-2：ESP8266(Node MCU)

2.3.1 樹莓派 Raspberry Pi 介紹

樹莓派是一款基於 Linux 的單晶片電腦。由英國樹莓派基金會所開發。出產 ARM 架構 700MHz BCM2835 處理器，256MB 記憶體，使用 SD 卡當作儲存媒體，且擁有一個乙太網路、兩個 USB 埠、以及 HDMI（支援聲音輸出）和 RCA 端子輸出支援。操作系統採開源的 Linux 系統，在本專題中，所有的雲端伺服器均安裝於樹莓派。

2.3.2 樹莓派 Raspberry Pi 硬體介紹

SoC	Broadcom BCM2837（CPU，GPU DSP 和 SDRAM、USB）
CPU	ARM Cortex-A53 64 位元 1.2GHz（四核心）
記憶體	1024 MB（LPDDR2）
USB 2.0 埠個數	4
影像輸出	HDMI 可分別從 640x350 到 1920x1200 之間
板載儲存	MicroSD 卡插槽
網路埠	乙太網埠，支援 802.11n 無線網路及藍牙 4.1
電源輸入	5V 電壓(通過 MicroUSB 或 GPIO 輸入)

表 2-3：樹莓派硬體介紹

圖 2-3：樹莓派 Raspberry Pi

第 3 章 軟體開發環境

3.1 Arduino IDE

3.1.1 Arduino IDE 安裝

1.在搜尋輸入 Arduino - Home

目前最新版本是 1.8.7，為了穩定與相容性考量，採用 1.8.5 版

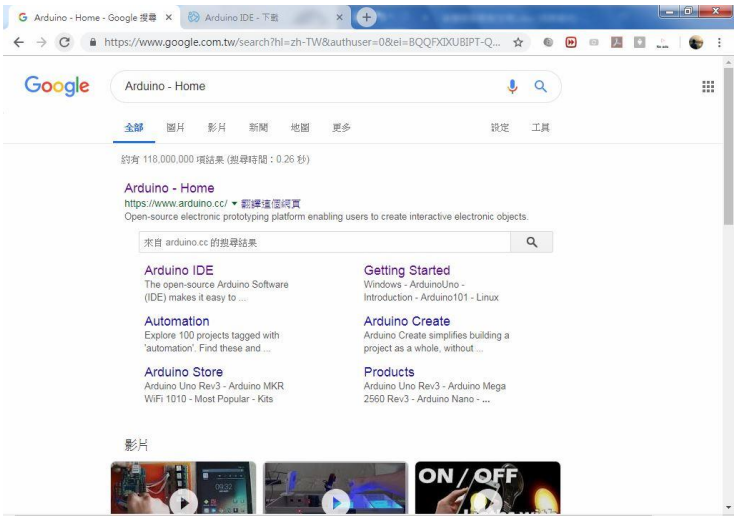


圖 3-1：搜尋輸入 Arduino

2.點入 Arduino IDE

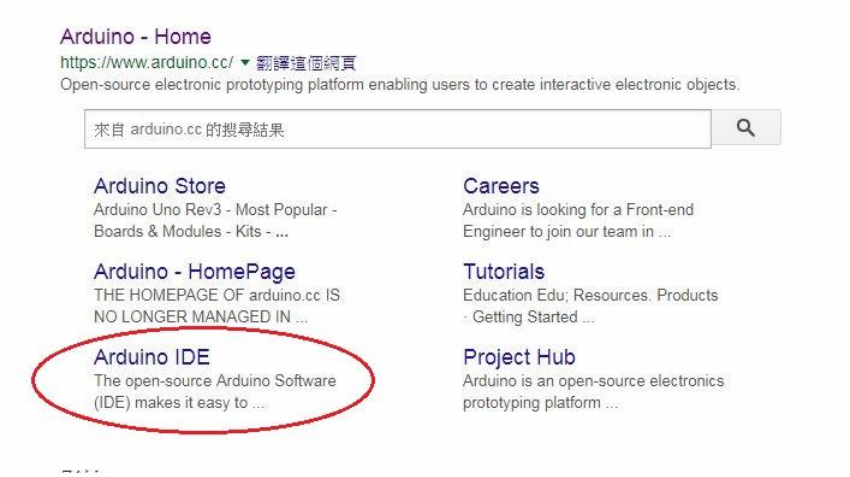


圖 3-2：點擊網頁

3.進入先前版本

圖 3-3：先前版本

4.Windows 下載(選擇自己的電腦系統的軟體)

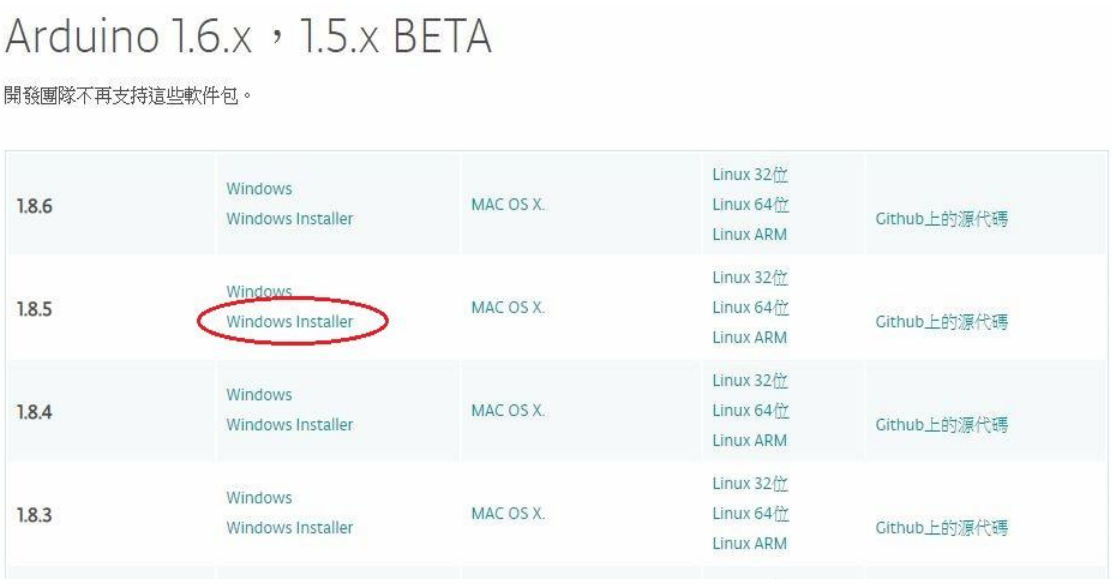


圖 3-4：選擇系統軟體

5.Arduino - 捐贈 - 點選只是下載

圖 3-5：下載 Arduino

6.安裝流程介紹

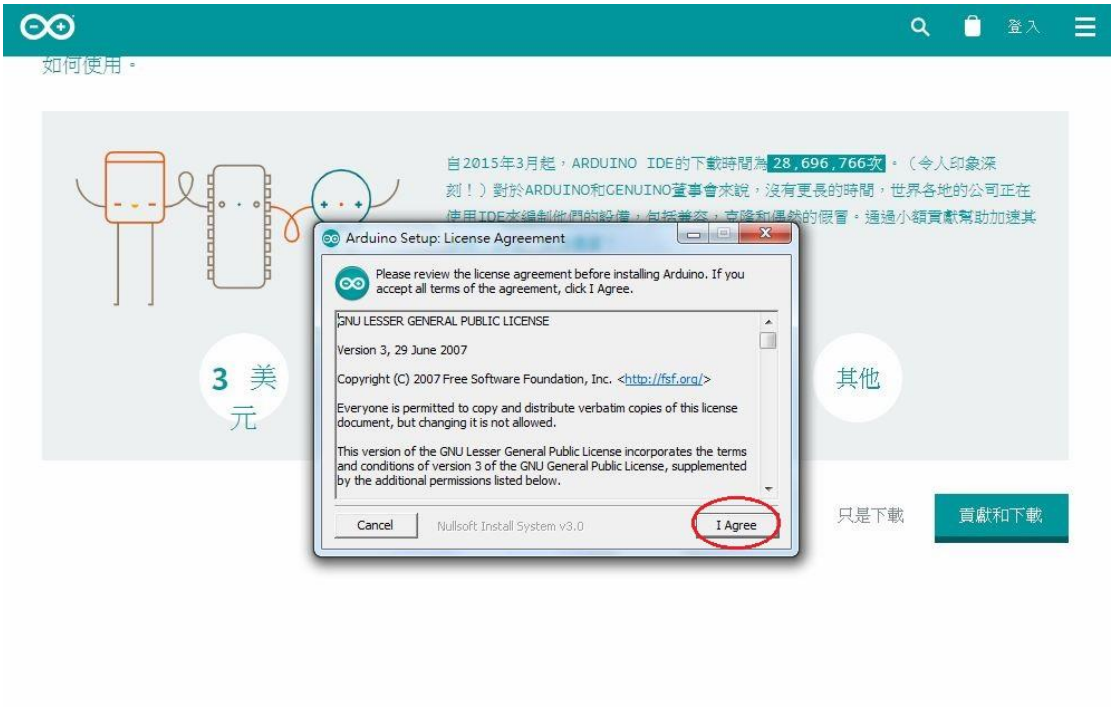


圖 3-6：安裝流程 1

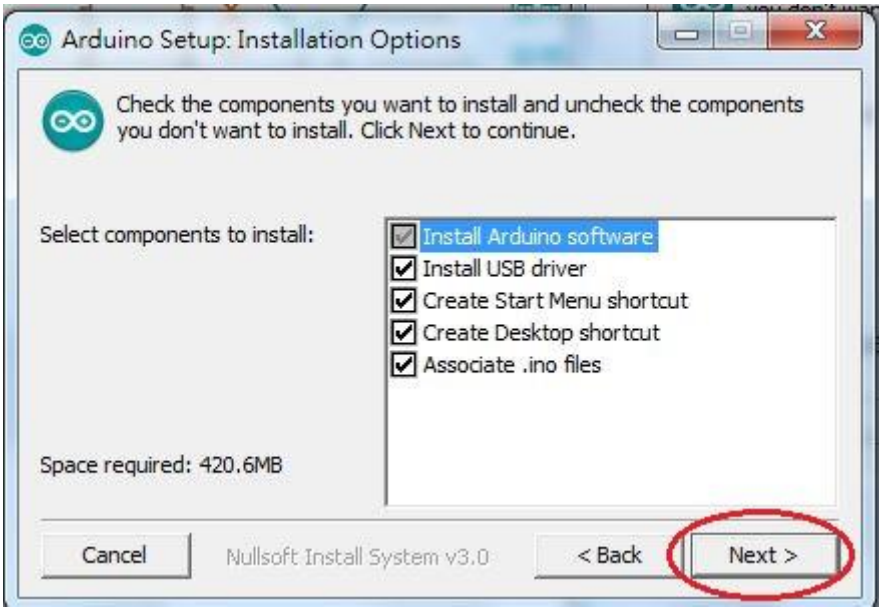


圖 3-7：安裝流程 2

7.儲存於你想要的位置

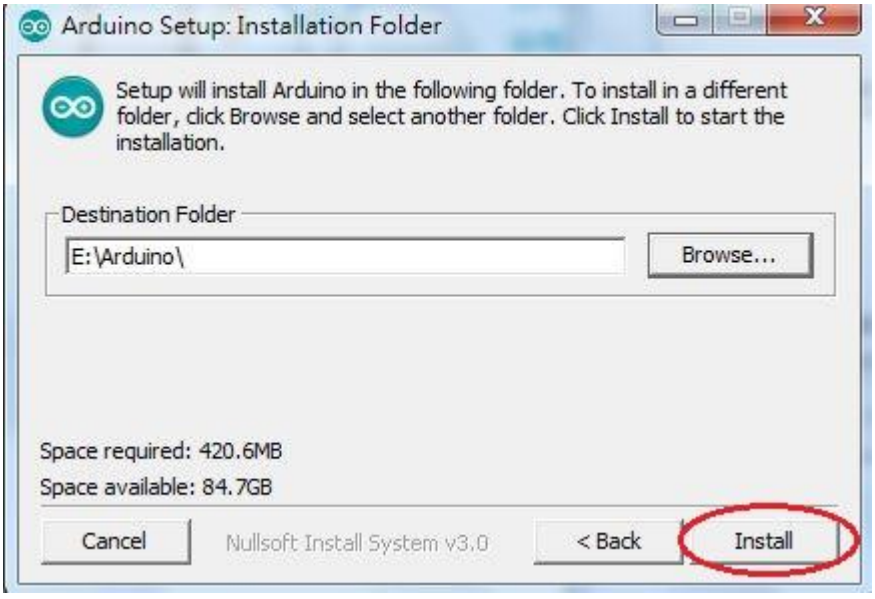


圖 3-8：安裝流程 3

8.安裝畫面

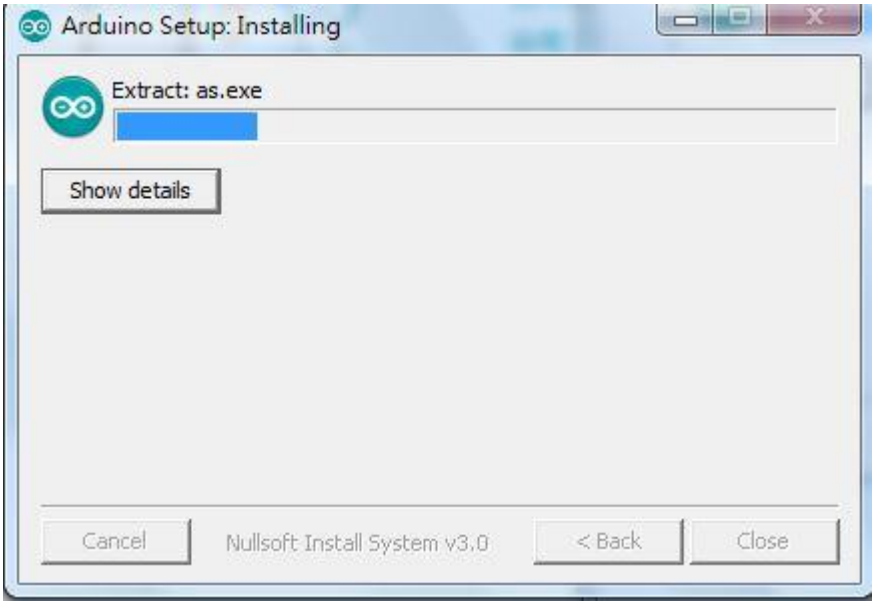


圖 3-9：安裝畫面

9.安裝完成

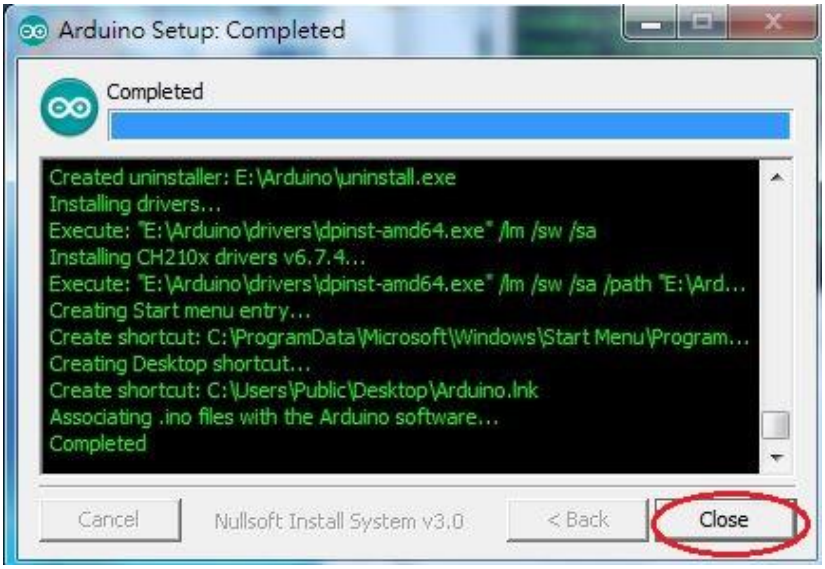


圖 3-10：安裝完成

3.1.2 Arduino IDE 開發環境介面

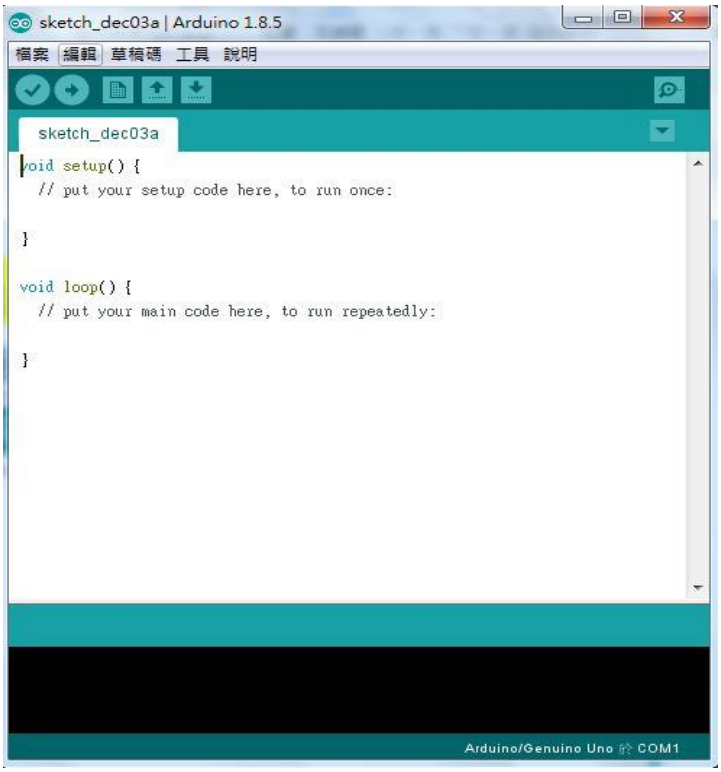


圖 3-11：Arduino 介面

	Verity 檢查參數設定或引入程式是否產生錯誤。
	Upload 程式編譯，將程式碼透過 USB 介面燒錄至 Arduino 控制板。
	New 產生新的腳本。
	開啟腳本，顯示在同一頁面上。點擊不同腳本便顯示不同腳本。
	Save 腳本儲存。
	Serial Monitor 開啟監視器頁面，監視 Arduino I/O 介面。

表 3-1：Arduino IDE 常用功能

Arduino 程式主要由 setup() 和 loop() 這兩個函式組成:

```
void setup(){
  //setup 初始環境設定
}
void loop(){
  //loop 程式執行主體
}
```

圖 3-12：函式 setup loop

因為setup()和 loop()是每支 Arduino 程式都會用到的兩個函式，為了方便，Arduino 已經幫大家準備好程式基本架構了，編寫Arduino程式時，可直接點選 File > Examples > 1.Basics > Digital ReadSerial 這個範本檔，另存新檔後，然後再修改setup()和 loop()兩個函式的內容即可。

3.2 無線基地台設定

本專題採用 華碩 RT-AC66U 無線基地台。此 WiFi 基地台支援 IEEE 802.11ac 標準、2.4GHz 與 5GHz 兩個頻段，評價頗高。

圖 3-13：華碩 RT-AC66U

<u>無線基地台設定</u>

1. 網址輸入 192.168.1.1

圖 3-14：基地台初始畫面

2.登入此基地台管理者使用帳號密碼

圖 3-15：管理者使用帳號密碼

3.簡易設定無線網路{SSID(ASUS-HUST)/PW}

圖 3-16：設定無線網路

4.樹莓派 IP(192.168.1.187)設定於 DHCP 中配送

圖 3-17：設定樹莓派 DHCP

3.3 樹莓派 Raspberry Pi 作業系統安裝與設定

1. 燒錄好 SD Card（安裝好 Pi 3 Image）。

2. 將設備接到 Pi3：USB 鍵盤、滑鼠、網路線(原有電腦網路線）。

3.HDMI 轉 VGA 線來連接 VGA 螢幕，連接 com port, 就可進入終端機模式開始進行設定或操作；當確認 IP Address 之後，即可進行視窗遠端操作。

4.開機完成後，會自動出現 Pi3 桌面 ；預設帳號/密碼為 pi/raspberry。

5.可用 vnc 透過網路連線至樹莓派，如圖 3-18 來開啟終端機文字視窗，以便修改網路參數。

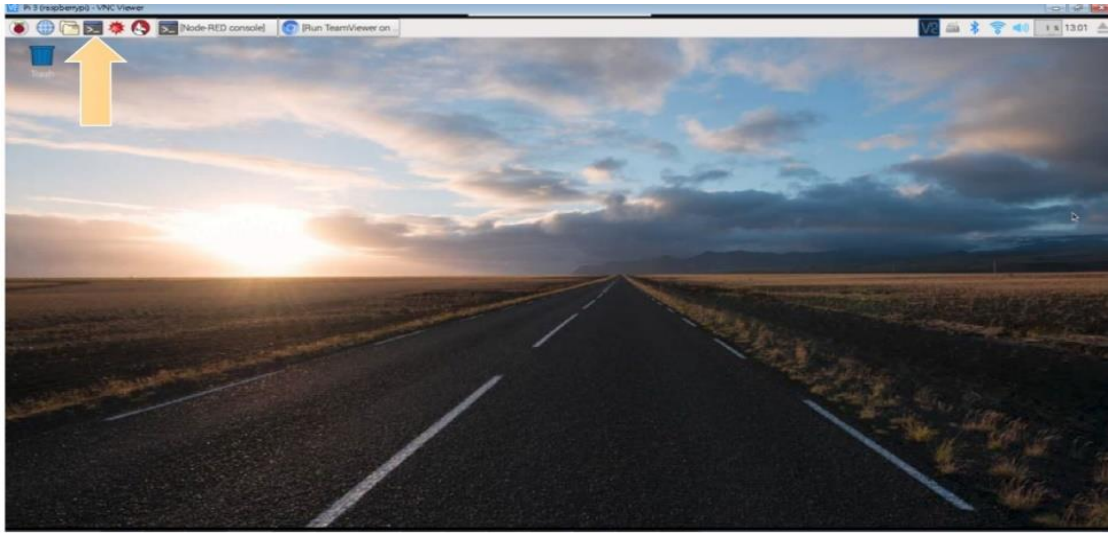


圖 3-18：開啟終端機

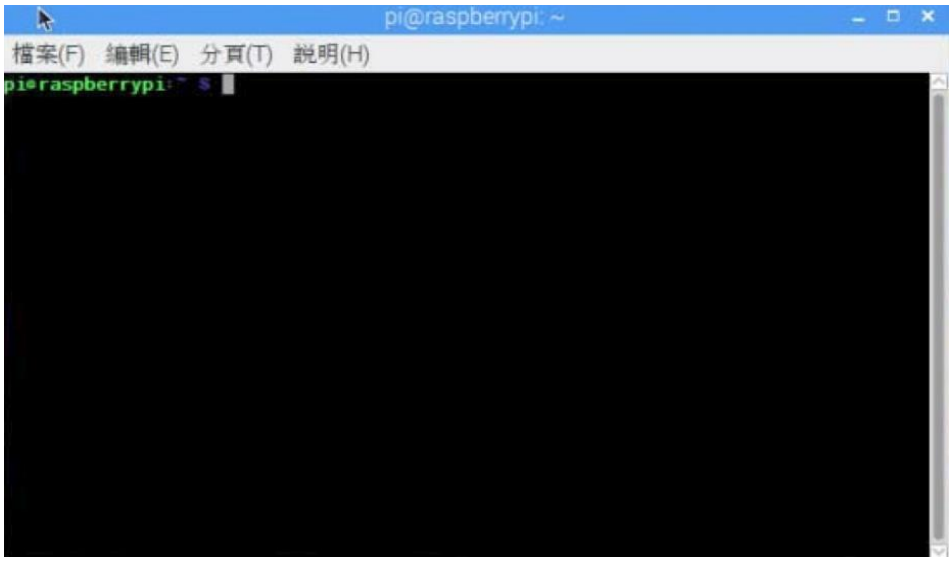


圖 3-19：使用終端機

6.修改網路參數(IP Address)：根據實際環境來設定

請執行以下指令:

```
# sudo nano /etc/dhcpcd.conf
# static IP configuration
  interface eth0
  static ip_address=192.168.1.187 /24
  static routers=192.168.1.1
  static domain_name_servers=168.95.1.1
  static wpa-ssid ASUS-HUST
  static wpa-psk 12345678
```

接著，按下【Ctil】+【O】，然後再下【Enter】完成原檔案的存檔

7.關機指令：sudo shutdown -h now

8.重新啟動指令: sudo reboot

3.4 樹莓派 Raspberry Pi 安裝 MQTT Server

是一種輕量級通訊協定，可讓各種設備互相溝通，運算與傳輸頻寬很低，適合物聯網中各種應用。

MQTT broker有諸多種，例如ActiveMQ Apollo、 HiveMQ、 Mosca、 Mosquitto、 RabbitMQ等。我們以Mosquitto這一個MQTT實作版本的安裝與使用。

- 在樹莓派的中我們可以使用apt直接安裝mosquitto套件

apt-get install mosquitto

- 我們可以使用service指令檢查mosquitto服務狀態

service mosquitto status

服務狀態呈綠色的active就表示Mosquitto有在正常運作

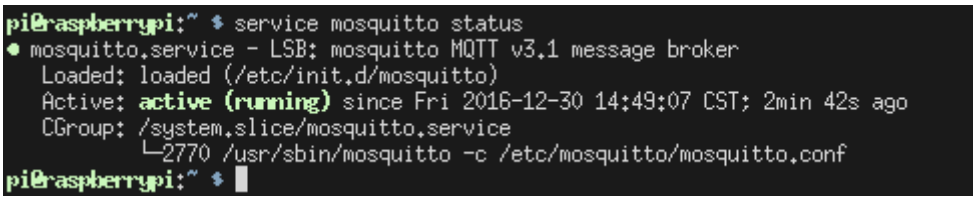


圖 3-20：mosquitto 服務狀態

3.5 樹莓派 Raspberry Pi 安裝 Node-RED Server

建議把樹莓派更新到最新版後才安裝Node-RED

>> bash <(curl

-sLhttps://raw.githubusercontent.com/node-red/raspbian-deb-

package/master/resources/update-nodejs-and-nodered)

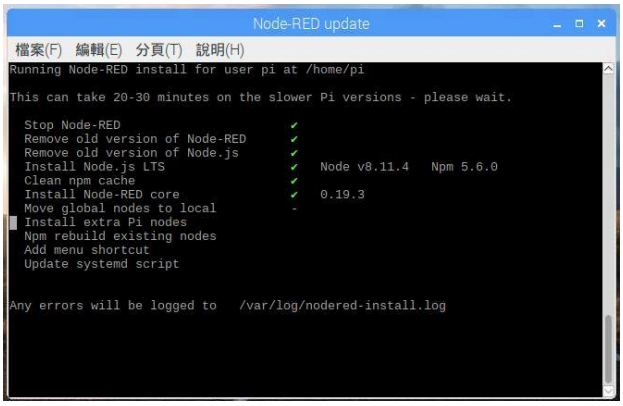


圖 3-21：Node-RED 安裝

Node-RED 環境設定

開啟瀏覽器在網址上輸入Raspberry Pi的IP及PORT 1880，如果未啟動，可設定開機後自動啟動。

在開機後自動將Node-RED的服務啟動：

cd ~/.node-red

npm rebuild

npm ls –depth=0

sudo systemctl enable nodered.service

3.6 樹莓派 Raspberry Pi 安裝 Blynk Server

安裝JAVA

- 因Blynk Server是JAVA包裝的JAR 檔案

```
$sudo apt-get install oracle-java8-jdk
```

- 下載Blynk伺服器程式

```
$wget"https://github.com/blynkkk/blynk-server/releases/down
load/v0.31.0/server-0.31.0-java8.jar"
```

- 啟動伺服器指令

```
$java -jar server-0.31.0-java8.jar -dataFolder /home/pi/Blynk
```

Blynk管理網頁https://IP:9443/admin

預設登入帳號：admin@blynk.cc 預設密碼：admin

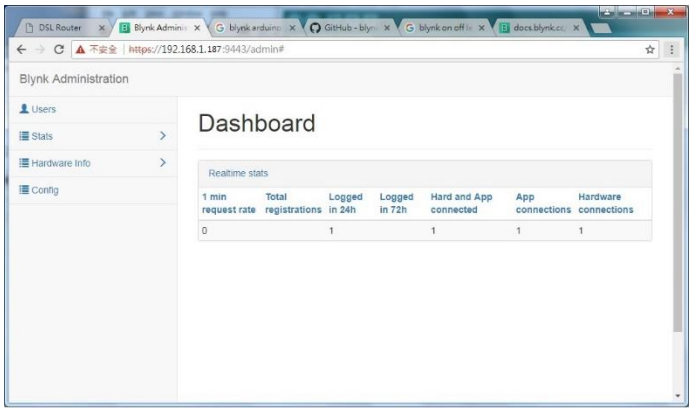


圖3-22：顯示目前連線Blynk APP

目前使用的專題環境，還不需要使用到付費的 Blynk 環境。

如下表為付費與免費的差異限制

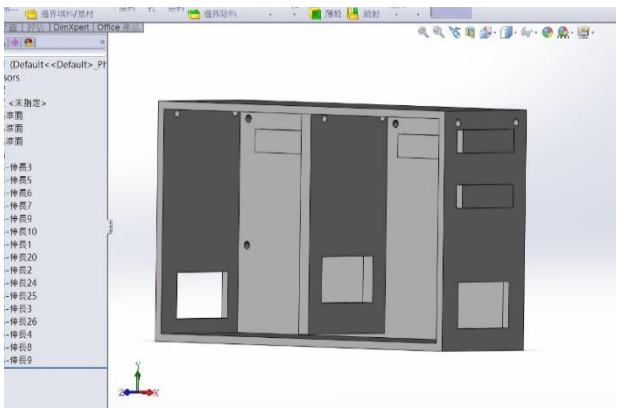
	Free	Startup 199 美金	Business 499 美金	Enterprise 聯繫公司
移動 App Builder	Y	Y	Y	Y
App Publishing	-	Y	Y	Y
應用品牌	-	-	Y	Y
應用更新	-	1	3	無限
最大設備數量	20	無限	無限	無限

表 3-2：Blynk 限制

第 4 章 物聯網之密室逃脫建置與相關零件

4.1 密室設計結構

利用 3D 軟體開模把構想圖 4-1 中的外型繪製後，使用 CNC 銑床技術圖 4-2 完成每一片加工圖 4-3，即可組裝起來完成密室逃脫外觀結構。



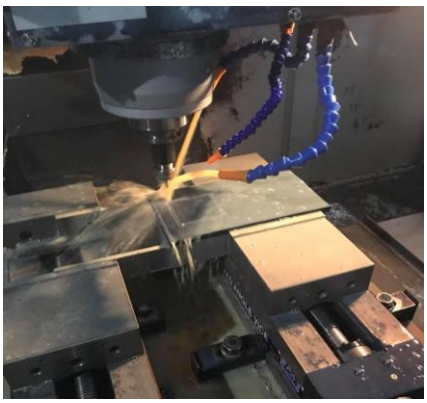


圖 4-1：3D 設計

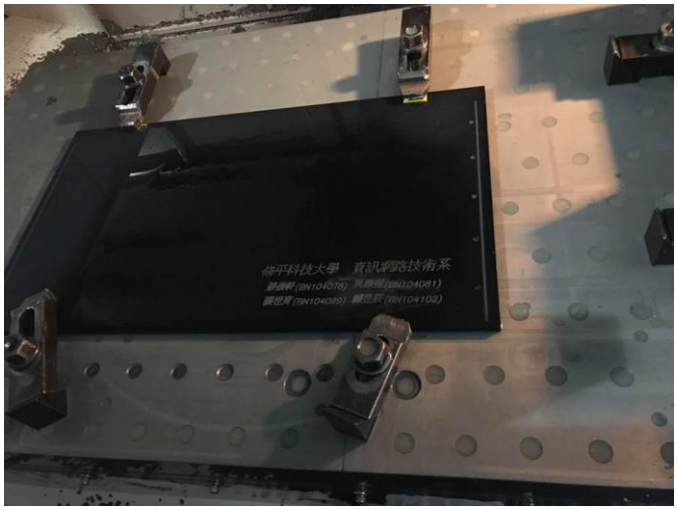


圖 4-2：CNC 加工

使用線上開關來控制 DC12V 白光 LED 側發光燈條，讓我們的密室外觀顯得更加醒目圖 4-3。

圖 4-3：LED 側發光燈條

密室的門我們利用冰棒棍裁減預設定大小後圖 4-4，黏上丁雙後，使用熱熔槍互相黏著即可完成門之設計。

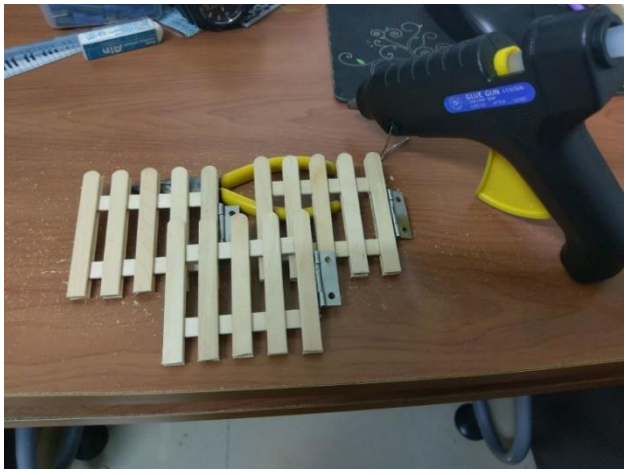


圖 4-4：使用冰棒棍設計密室門

4.2 電路板

利用 Fritzing 軟體設計繪製電路板線路圖後圖 4-5，使用顯影劑及蝕刻液的功能，下去洗電路板後圖 4-6，以光的照射來完成電路板圖 4-7。此電路板搭配穩壓電源模組，提供本密室逃脫所有的電源供應系統。

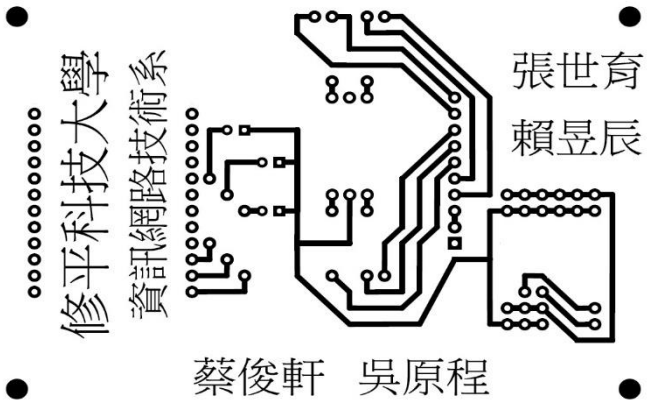


圖 4-5：電路線路圖

圖 4-6：洗電路板

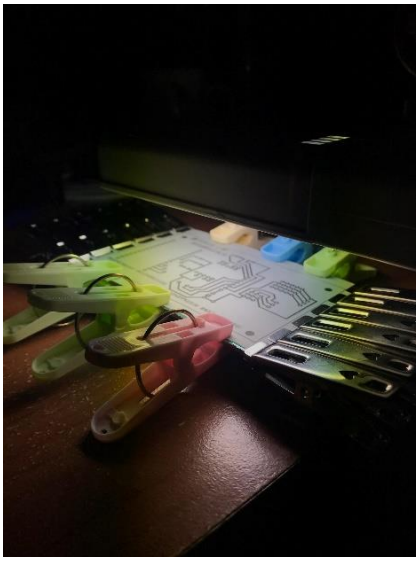


圖 4-7：光照

4.3 電路板元件功能

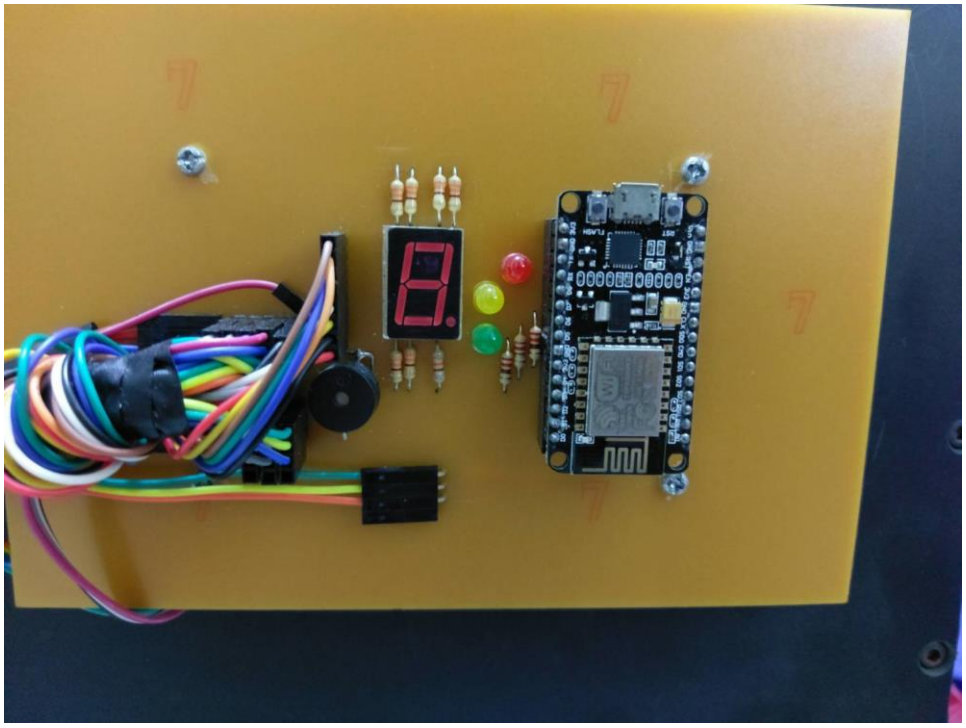


圖 4-8：電路板元件說明

ESP8266 NodeMCU

Arduino 沒提供網路功能，所以擴充採用 NodeMCU 無線模組。

紅黃綠燈

在 NodeRED 控制流程編輯中，做後台測試使用。

七段顯示器

每過一關會加 1，闖關成功，即會歸 0。

蜂鳴器

過一關逼一聲。結束逼三次通知玩家成功闖關。

4.4 遊戲關卡零件介紹

本物聯網之密室逃脫專題使用 Arduino 模組(蜂鳴器、LCD 顯示器、伺服馬達、超音波感測器、紅外線感測器)、7 段顯示器及 ESP8266(NodeMCU)

蜂鳴器：每過一關會逼一聲；闖關成功即會逼逼逼三聲來結束遊戲流程。

LCD 顯示器：利用顯示器來提供玩家闖關的進度。圖 4-9

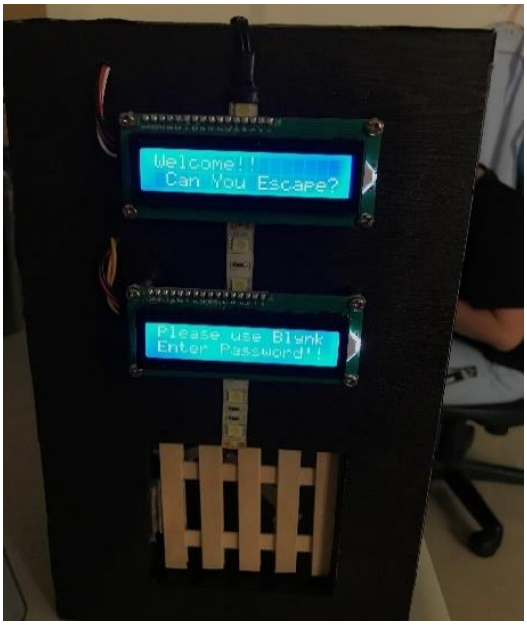


圖 4-9：LCD 顯示器

伺服馬達：驅動馬達後及可開啟密室闖關的門。圖 4-10

圖 4-10：伺服馬達

超音波感測器：此感測器需離感測 5 公分以內測 3 次即可過關。

圖 4-11

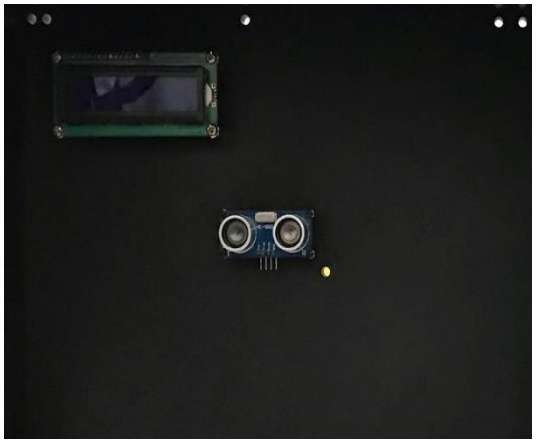


圖 4-11：超音波感測器

紅外線感測器：需觸碰此感測器 3 次即可過關。圖 4-12

圖 4-12：紅外線感測器

7段顯示器:過一關會提醒顯示玩家闖關關卡,闖關成功即歸零。圖 4-13

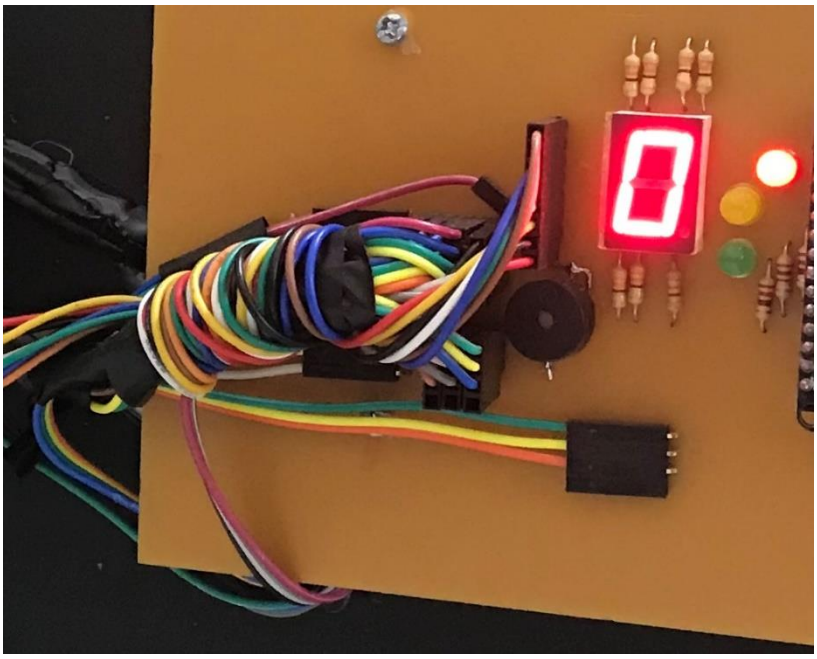


圖 4-13：七段顯示器

第 5 章 物聯網之密室逃脫程式開發

5.1 Arduino 周邊設備的擴充

5.1.1 I2C 介面並聯 6 個 LCD 顯示器

本專題使用 6 個 I2C 1602 LCD 點陣液晶模組，如圖 5-1。來顯示每一關卡中過關訊息，I2C 模組以焊接方式焊於 LCD 液晶模組上

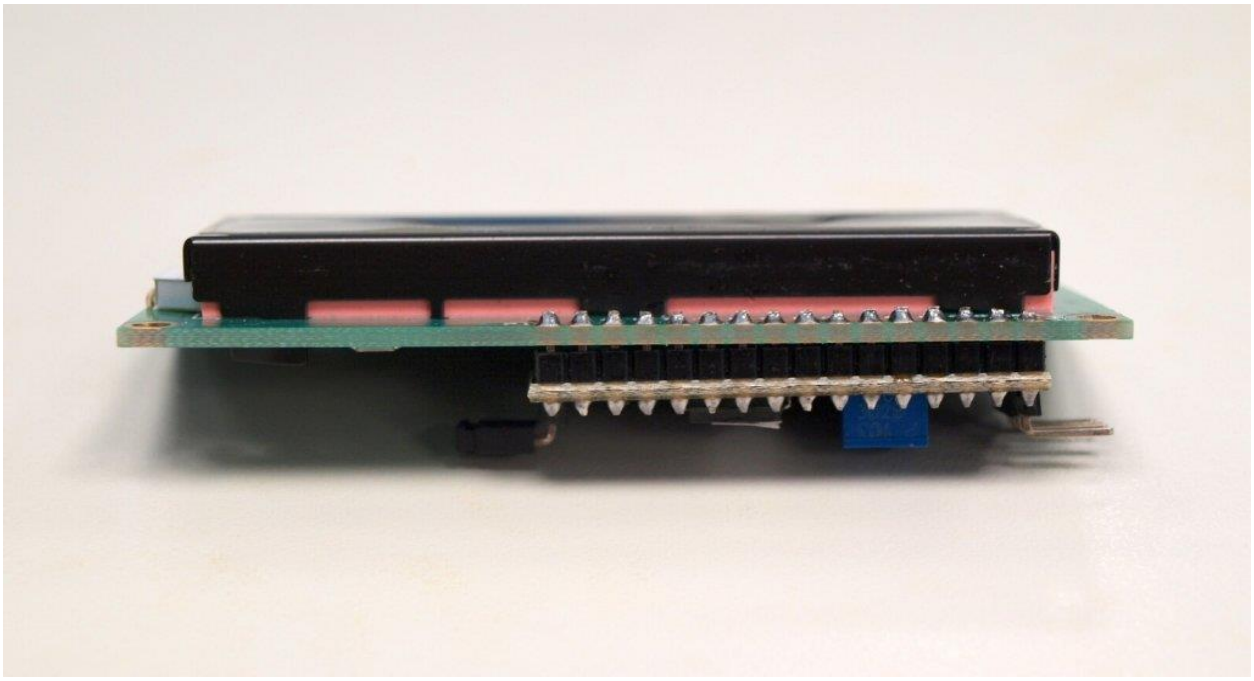


圖 5-1：LCD 液晶模組

5.1.2 伺服馬達

程式中如圖 5-2。我們必須定義此模組的接腳，再定義哪些顯示器要顯示的文字，最後設定多久時間更新此液晶面板一次，讓玩家可以很明確的知道，目前闖關的進度。

圖 5-2：LCD 顯示器程式

5.1.3 超音波感測器

本專題使用 3 個伺服馬達，此馬達的功能在於每闖關成功一關，感應模組就會驅動此馬達，讓密室中的門打開，讓玩家能順利達到下個關卡。程式中如圖 5-3。主要在伺服馬達的旋轉角度大小以及延遲時間。

M1

```
#include <Servo.h>

Servo myservo;  // 建立一個 servo 物件，最多可建立 12個 servo

int pos = 35;    // 設定 Servo 位置的變數

void setup() {
  myservo.attach(10);  // 將 servo 物件連接到 pin 9
}

void loop() {
  // 正轉 180度
    for (pos = 35; pos <= 95; pos +=60  ) // 從 0 度旋轉到 180 度，每次 1 度
  {
    myservo.write(pos);              // 告訴 servo 走到 'pos' 的位置
    delay(850);                       // 等待 15ms 讓 servo 走到指定位置
  }

  // 反轉 180度
  for (pos = 95; pos >= 35; pos -=60 ) // 從 180 度旋轉到 0 度，每次 1 度
  {
    myservo.write(pos);              // 告訴 servo 走到 'pos' 的位置

    delay(850);                       // 等待 15ms 讓 servo 走到指定位置
```

圖 5-3：伺服馬達程式

5.1.4 紅外線與七段顯示器

本專題使用 1 個超音波感測器。程式中如圖 5-4。主要落在感測器的數位腳位的定義以及感測中的距離設定。

圖 5-4：超音波感測器程式

本專題使用 1 個紅外線感測器及 1 個 7 段顯示器。因為此感測器在密室逃脫遊戲中，是最後一個關卡，在闖關成功後 7 段顯示器須歸零，我們把程式合併在一起。圖 5-5：腳位程式定義紅外線感測器與 7 段顯示器的腳位。如圖 5-5

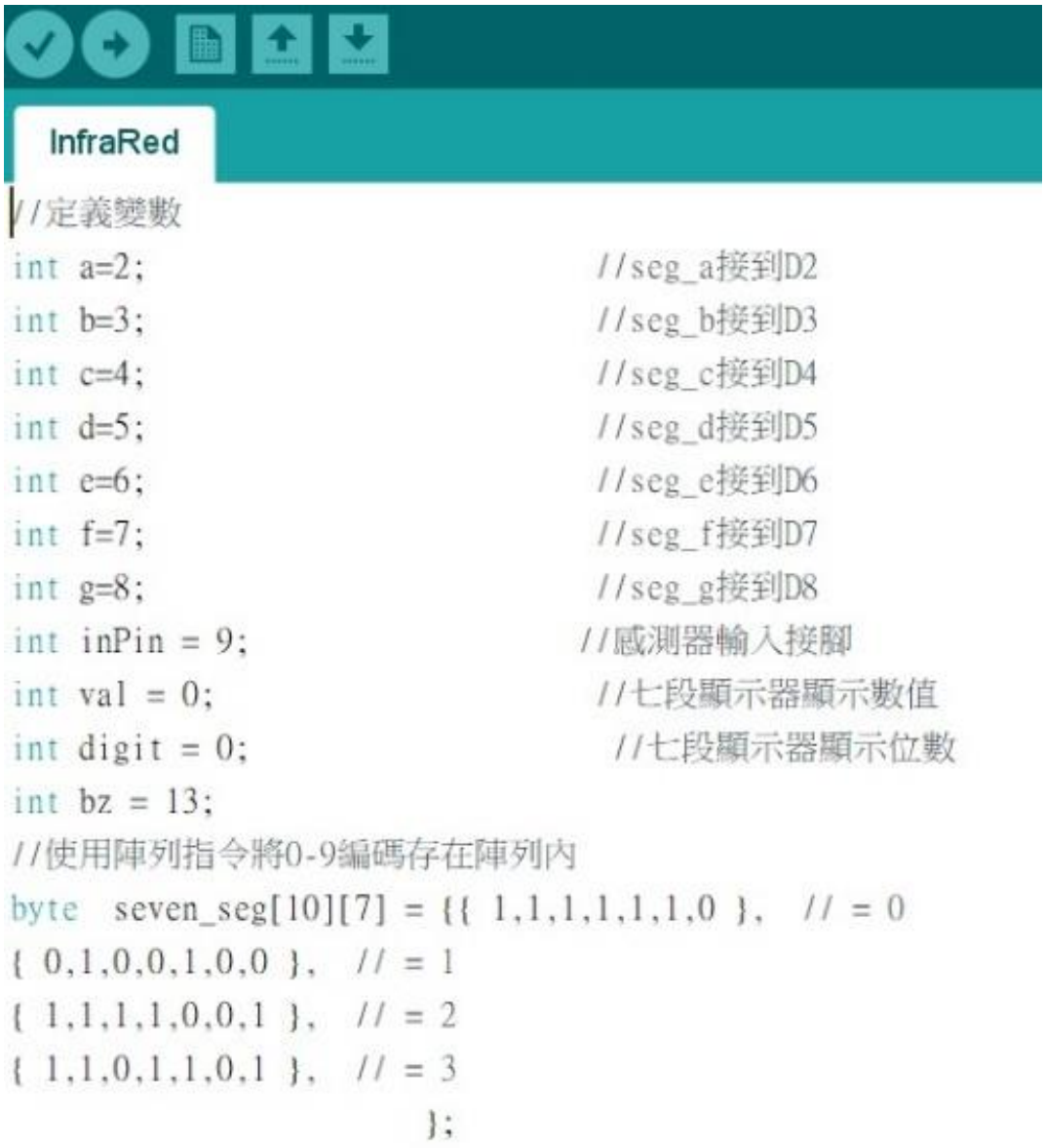
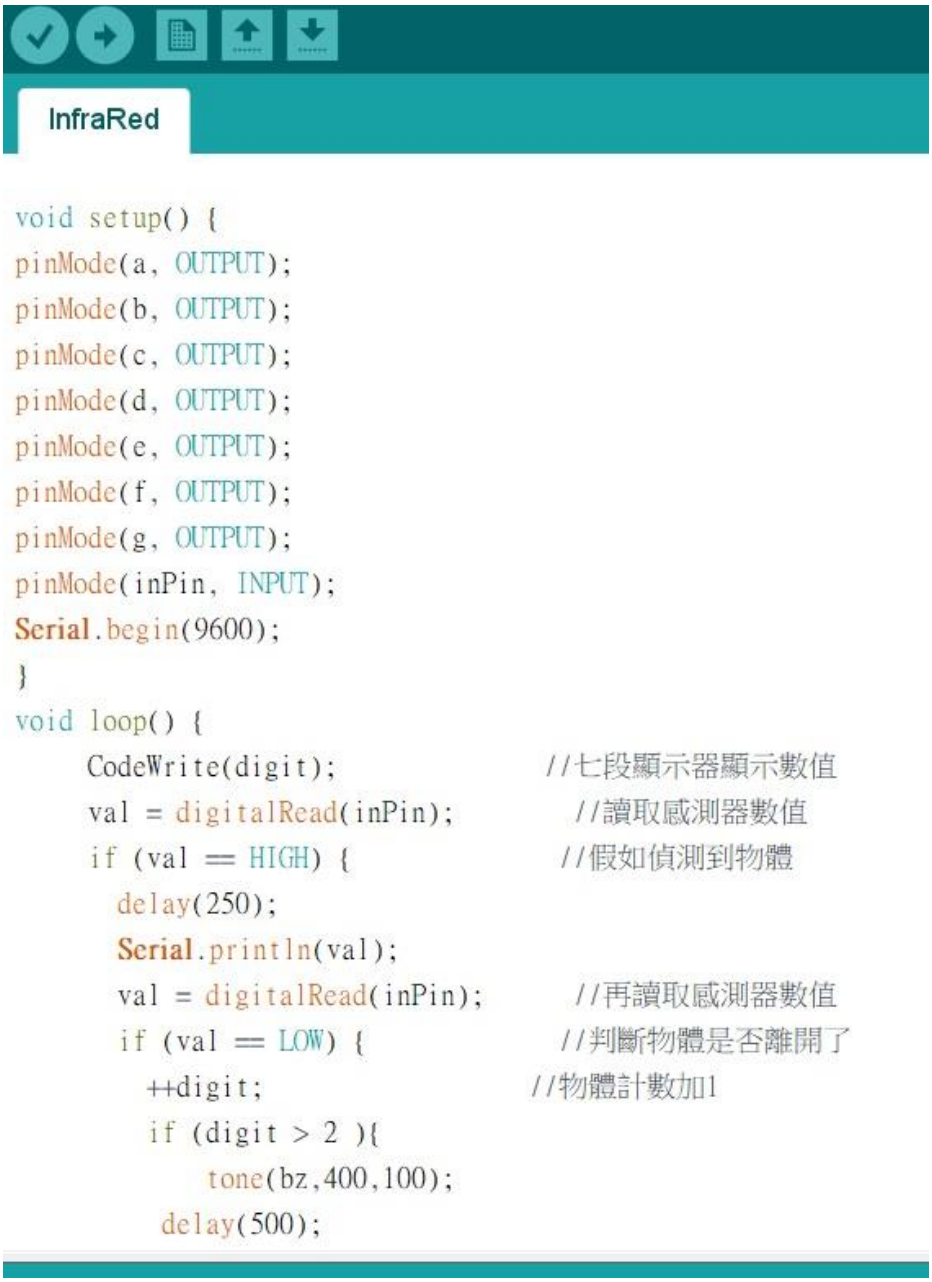


圖 5-5：腳位程式定義紅外線感測器與 7 段顯示器的腳位

紅外線感測器主要只讀取到 3 次的數值即可，此時 7 段顯示器即歸零，表示密室逃脫遊戲已經闖關完畢。圖 5-6：紅外線感測器、7 段顯示器程式所示。



InfraRed

```
void setup() {
pinMode(a, OUTPUT);
pinMode(b, OUTPUT);
pinMode(c, OUTPUT);
pinMode(d, OUTPUT);
pinMode(e, OUTPUT);
pinMode(f, OUTPUT);
pinMode(g, OUTPUT);
pinMode(inPin, INPUT);
Serial.begin(9600);
}
void loop() {
     CodeWrite(digit);                  //七段顯示器顯示數值
     val = digitalRead(inPin);           //讀取感測器數值
     if (val == HIGH) {                  //假如偵測到物體
       delay(250);
       Serial.println(val);
       val = digitalRead(inPin);         //再讀取感測器數值
       if (val == LOW) {                 //判斷物體是否離開了
         ++digit;                       //物體計數加1
         if (digit > 2 ){
             tone(bz,400,100);
          delay(500);
```

圖 5-6：紅外線感測器、7 段顯示器程式

5.2 使用樹莓派 MQTT 作為資料交換中心

MQTT 扮演三種腳色	Publisher：發送訊息者
	Subscriber：接收訊息
	Broker：轉送訊息者

表 5-1：MQTT 腳色

本專題 MQTT 扮演 Broker 轉送訊息者的腳色圖 5-7 所示

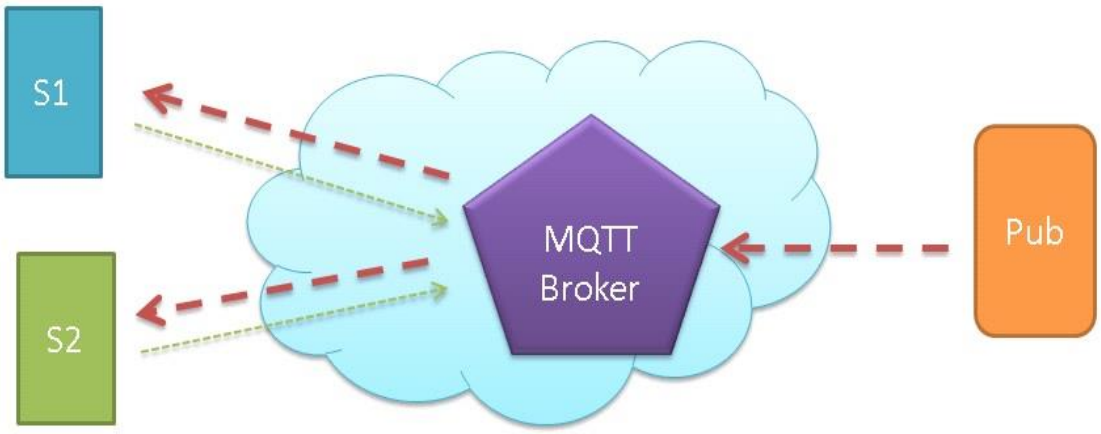


圖 5-7：Broker 適意圖

韌體參數修改：圖 5-8：修改參數程式

WiFi	SSID 與 Password
MQTT	Address, port, username, password, topic

表 5-2：參數修改項目

檔案 編輯 草稿碼 工具 說明

ESCAPE_MCU

```
int pin14=14;
//-----------------------------------------------------------------------------------------------
// (*) Wifi AP 連線設定
const char* ssid = "ASUS-HUST";
const char* password = "hust1234";
//-----------------------------------------------------------------------------------------------
// (*) MQTT Broker IP Address/username/password/topic
#define MQTT_SERVER                 "192.168.1.187"
#define MQTT_PORT                   1883
#define USERNAME                    "user001"
#define PASSWORD                    "12345678"
#define TOPIC                       "1b9d4d14eda046b0a755194c2b65e666"    // IoT - Node-MCU 授權碼
#define TOPIC_ANY_PIN               TOPIC"/pin/#"      // +, # ：是萬用字元（for單一主題、多重主題）
```

圖 5-8：修改參數程式

發佈 MQTT：主要在樹莓派與 NodeMCU 之間做資料的轉運站。程式中先處理觸發發佈的時間，觸發後嘗試跟樹莓派連線，連線成功後即可成功當轉送訊息者的腳色。圖 5-9：發佈 MQTT 程式

圖 5-9：發佈 MQTT 程式

5.3 使用樹莓派 Node-RED 連線 Blynk

樹莓派中 Node-RED 流程控制。圖 5-10：Node-RED。基於 Node.js 所開發的系統，以瀏覽器為主的圖控式流程開發工具。利用 Browser 開啟 http://127.0.0.1:1880 即可啟用 Node-RED 功能。在做 Node-RED 本專題前先要有 NodeMCU 的 auth，以及在 Blynk 介面按鈕腳位圖 5-11：Blynk 數位按鈕做相互的呼應，即可輕鬆的完成。

圖 5-10：Node-RED

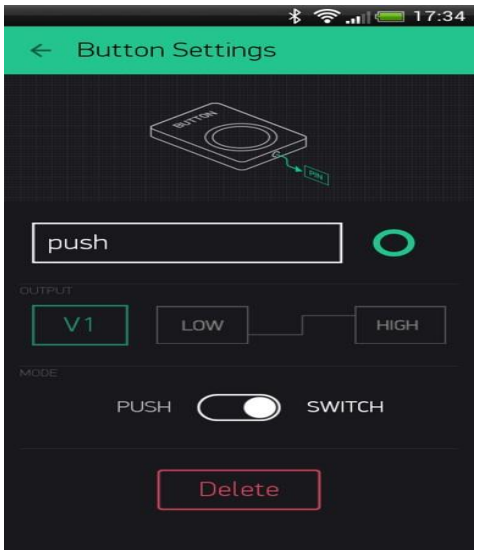


圖 5-11：Blynk 數位按鈕

5.4 Blynk 連線與設計專案

Blynk App:設定服務主機(Pi3)

先切換到 Server 模式，選 CUSTOM 私有主機，並輸入(Pi3)網址:192.168.1.187 後再點擊 Create New Account，輸入個人 Email 與密碼，以建立帳號，如圖 5-12：修改 Blynk Server

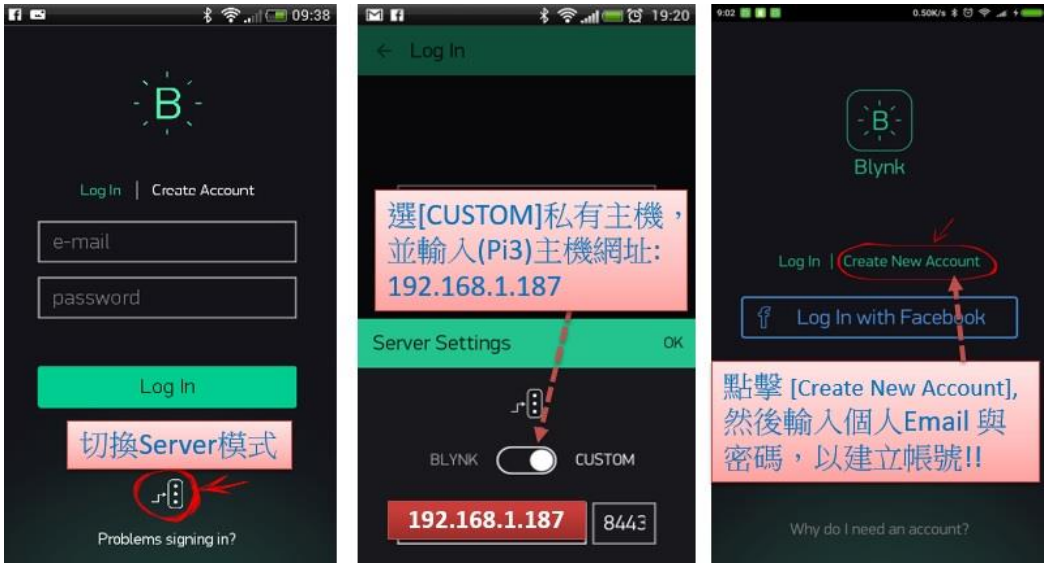


圖 5-12：修改 Blynk Server

Blynk App 專案建立:

創一個專案，設備選擇 NodeMCU，連線方式選 WiFi，成功後會寄一組 auth 到信箱，此 auth 會應用在 NodeMCU 及 Node-RED 的連線控制，如圖 5-13，Blynk 建立專案時會產生一組授權碼 Auth Token

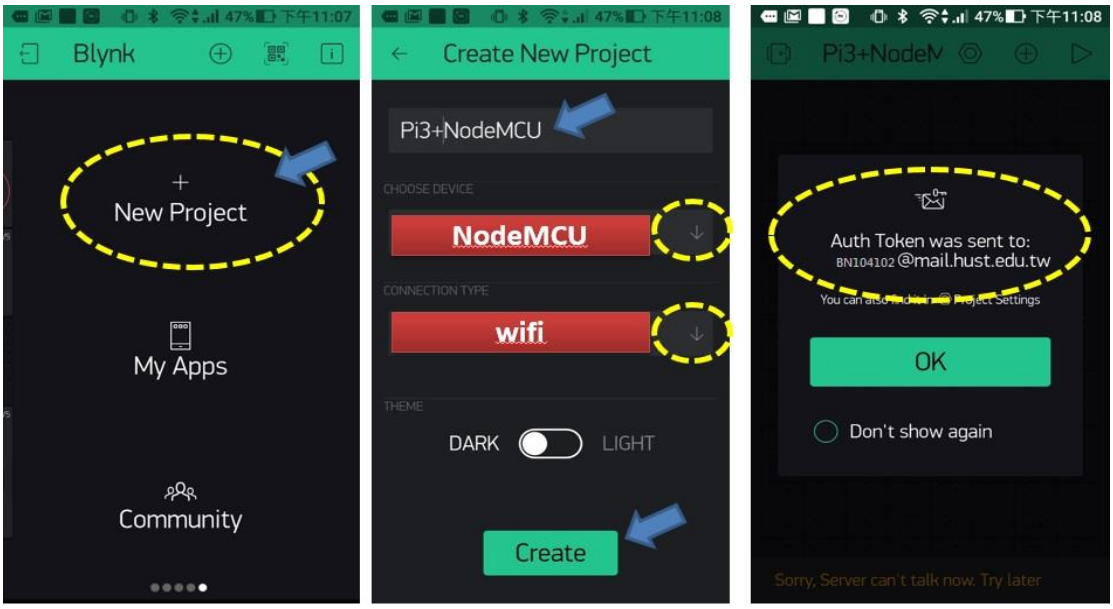


圖 5-13：Blynk 建立專案

5.5 NodeRed 程式設計與 Blynk、MQTT 的結合

介面採數位式按鈕由 1 到 9、清除及確認鍵，以一個文字控制項的設計，帶給玩家能夠明確的是否闖關正確。圖 5-14：Blynk 介面設計

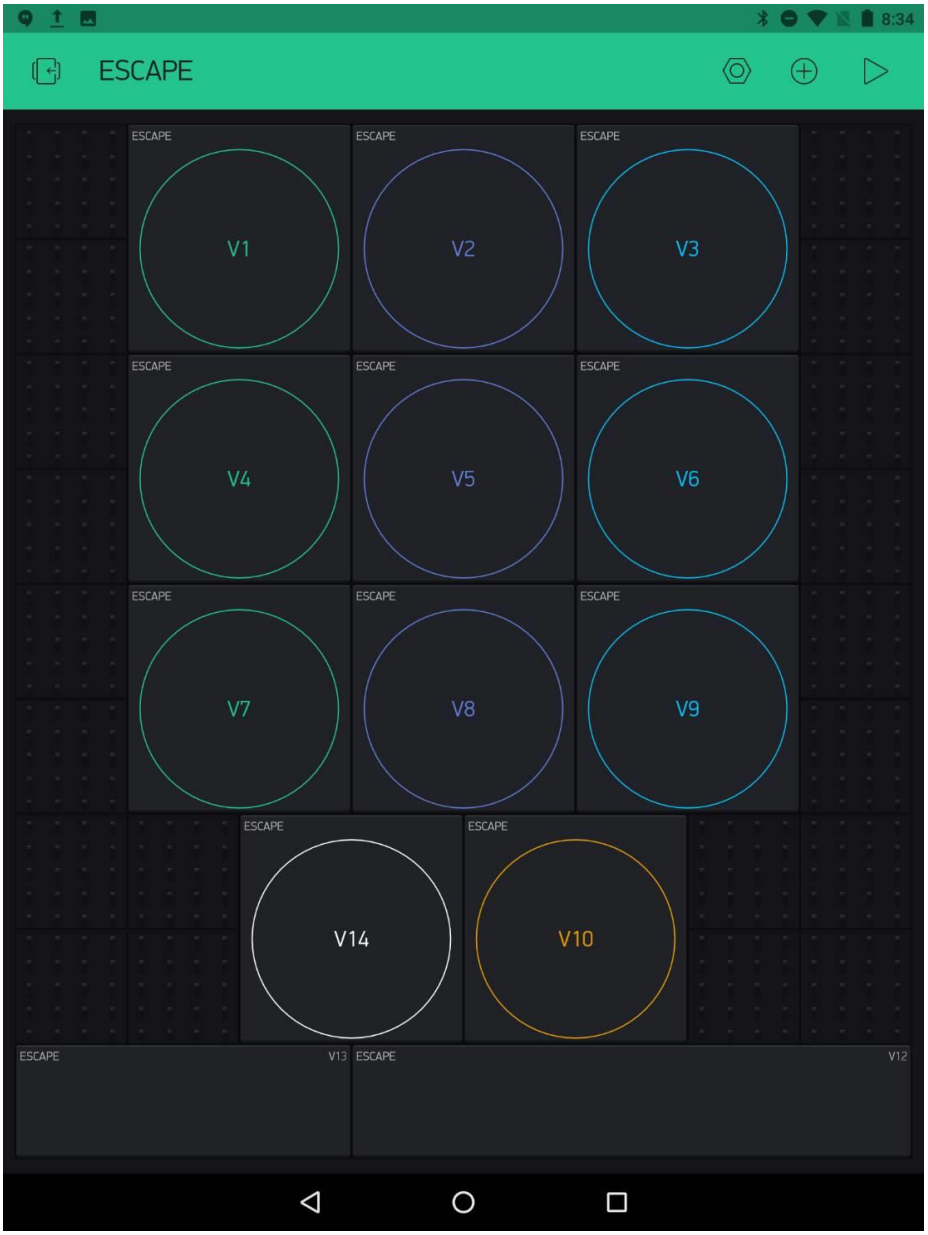


圖 5-14：Blynk 介面設計

5.6 NodeRed 流程控制設計

圖 5-15 是 NODE-RED 的內建節點清單，在專題 RED 設計階段，也有使用到其中幾項節點。

圖 5-15：NODE-RED 的內建節點清單

圖 5-16 左邊節點為 MQTT 訂閱制定主題，並且使用/分隔符號，在設定中必須輸入 BlynkAPP 的授權碼及狀態主題，授權碼一致才能接收 BlynkAPP 的訊息，使用/分隔符號。

中間節點為函式方塊，可以撰寫小程式碼或是想執行的指令。

右邊節點為設計時 Debug 所作為顯示使用，可以選擇是否顯示。

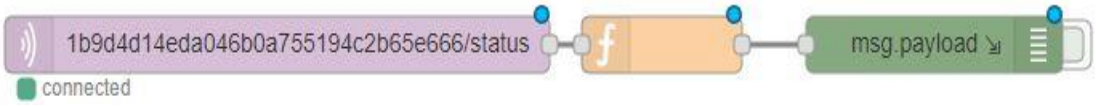


圖 5-16：MQTT 訂閱制定主題

圖 5-17 左邊第一排 11 個綠色節點是使用 write event，並且設定腳位的值，這邊設定的腳位值必須與 BlynkAPP 所設計的按鈕腳位值一致，NODE-RED 才能夠透過授權碼收到到輸入的值，同時在 BlynkAPP 的開關也設定了 01 高低電位，判斷是否選擇。

第二排 11 個橘色節點是函式方塊，其中有 9 個數字，利用收到腳位觸發訊號,執行函式節點判斷按到哪些按鈕,並且傳到彙整函式節點，定義一個 val 暫存目前輸入的字串。中間黃色開關節點，判斷 OK 鍵是否有按壓，如果有按壓就將暫存字串送出，判斷字串與預設字串是否符合，如果符合就會送出正確的訊息，透過 MQTT 發佈節點送出 NO 訊息，如果不符合就會送出錯誤的訊息，並且透過 MQTT 發佈節點送出 OFF 訊息，同時將結果傳到 BlynkAPP 畫面的狀態條。

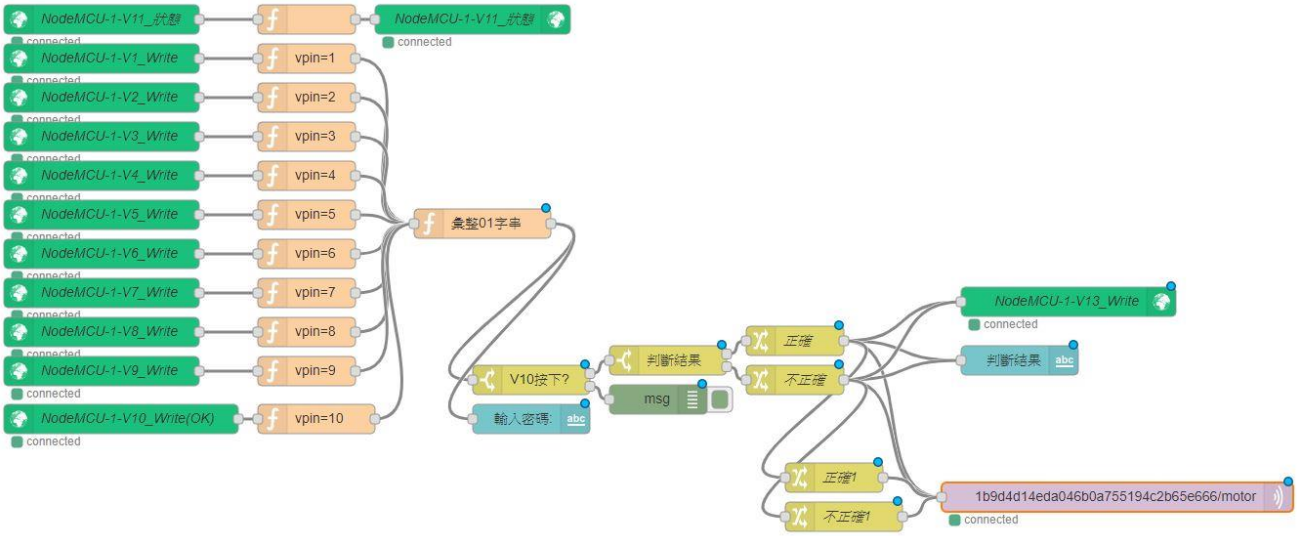


圖 5-17：NodeRED 程式設計

圖 5-18 為判斷如果清除鍵按壓，將會清除 BlynkAPP 畫面的狀態條訊息。

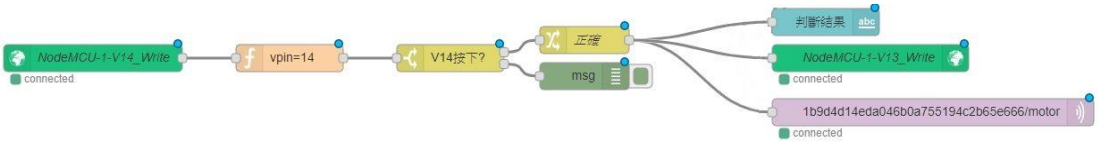


圖 5-18：判斷清除鍵按壓

Node-Red 應用虛擬接腳配合 Blynk App 上的元件接腳

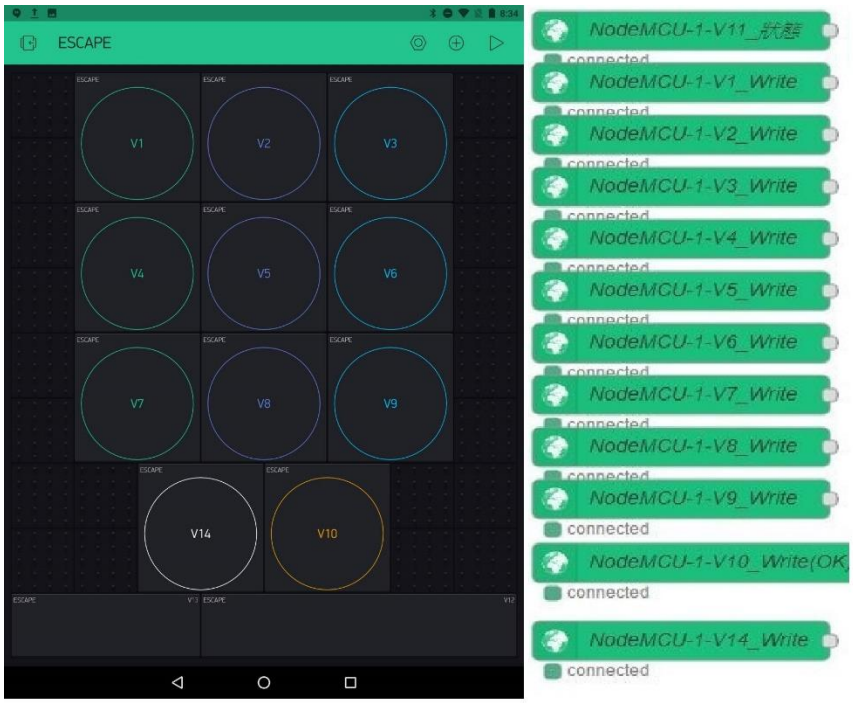


圖 5-19：Blynk 元件接腳

判斷結果正確，顯示正確 Pass；不正確，顯示 Wrong。

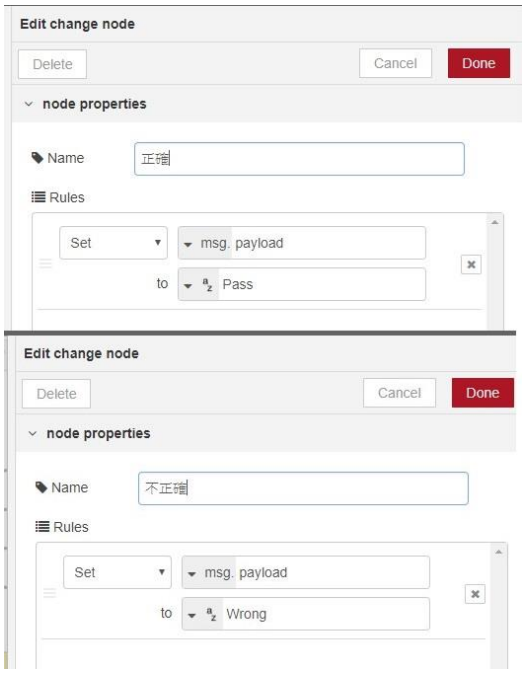


圖 5-20：NodeRED 判斷

5.7 Blynk App 設計

圖 5-19 為 BlynkAPP 元件庫的畫面，餘額額度是用電力及電池圖案來表示，拉元件是需要付出電力，私人的 BlynkSever 一開始就有 10 萬額度，如果是使用官方的 Sever 一開始就只有 200 額度，必須儲值才能拉更多元件，這次我們使用到 11 個按鈕及 1 個標籤。

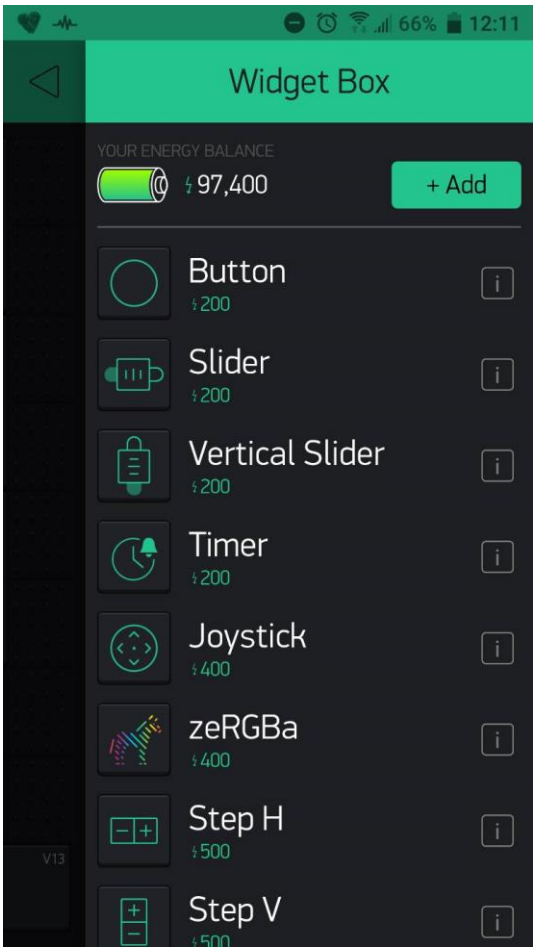


圖 5-21：Blynk APP 元件庫的畫面

圖 5-20 為按鈕元件的設定介面，可以設定顏色、名稱及按壓前後的按鈕顯示文字，最重要的是輸出腳位設定，必須與 NODE-RED 的節點腳位一致才行。

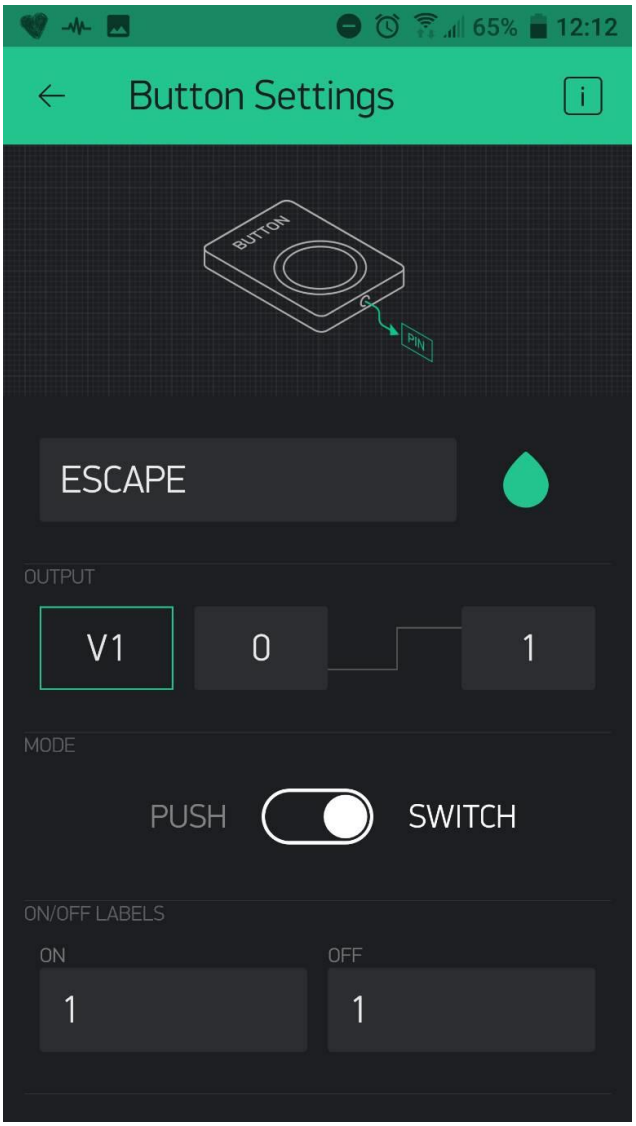


圖 5-22：Blynk APP 按鈕元件的設定介面

圖 5-21 為標籤元件設定介面，跟按鈕一樣可以設定名稱、顏色及接腳，不過標籤元件還有多兩項，就是文字的對齊方式和刷新的頻率。

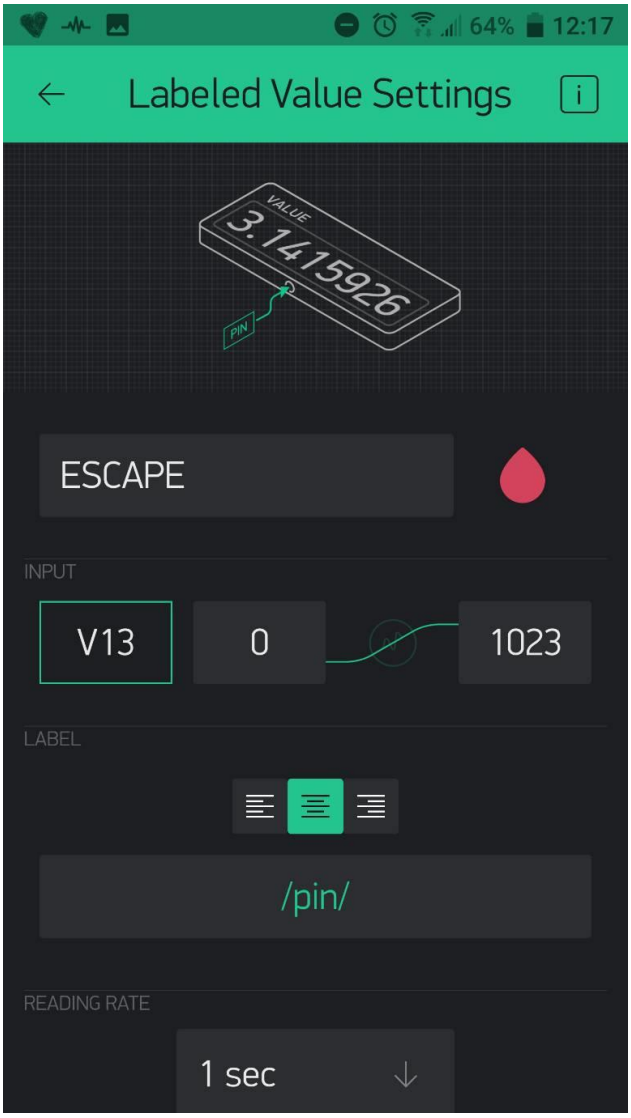


圖 5-23：Blynk APP 標籤元件設定介面

圖 5-22 為我們設計的主畫面，V1~V9 是數字鍵 1~9，輸入密碼後，按下 V10 確認鍵，將會把結果顯示在下方的 V13 標籤欄，如果像要清除標籤欄的狀態文字可以按下 V14 按鈕將會清除。

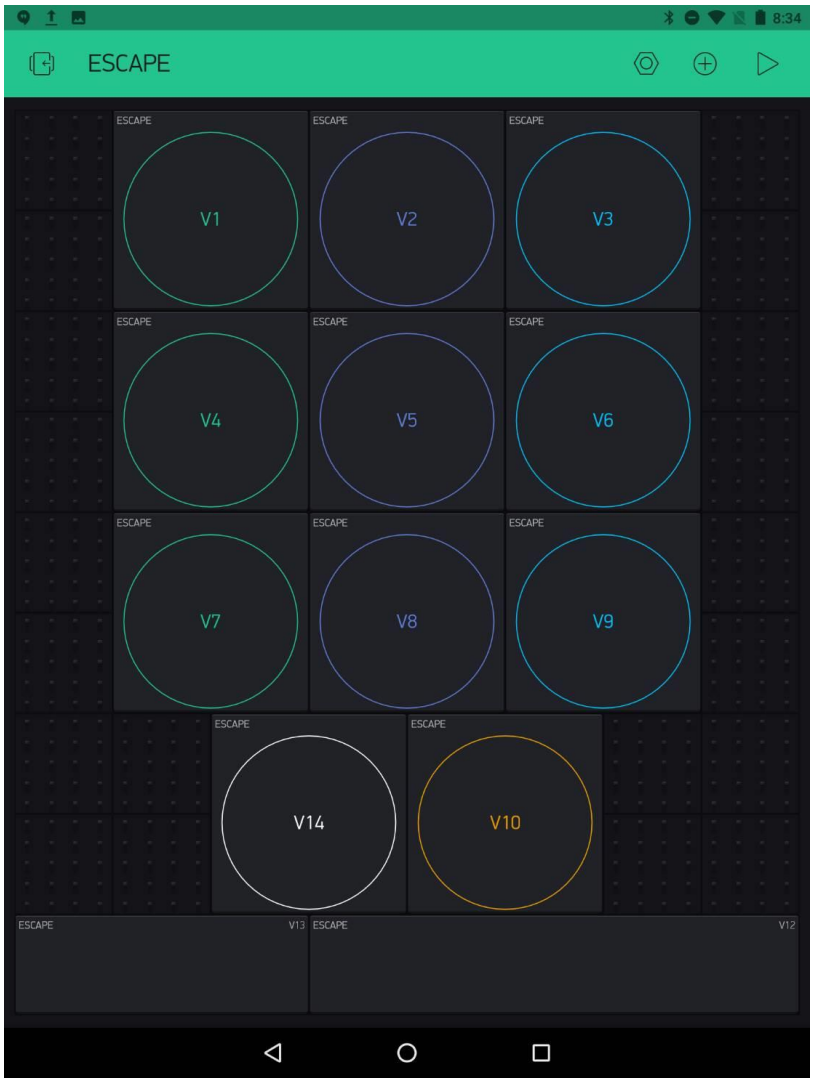


圖 5-24：Blynk APP 密室逃脫闖關畫面

5.8 每一關程式流程

以樹莓派 NODE-RED 來設計 如圖 5-25：NODE-RED 設計

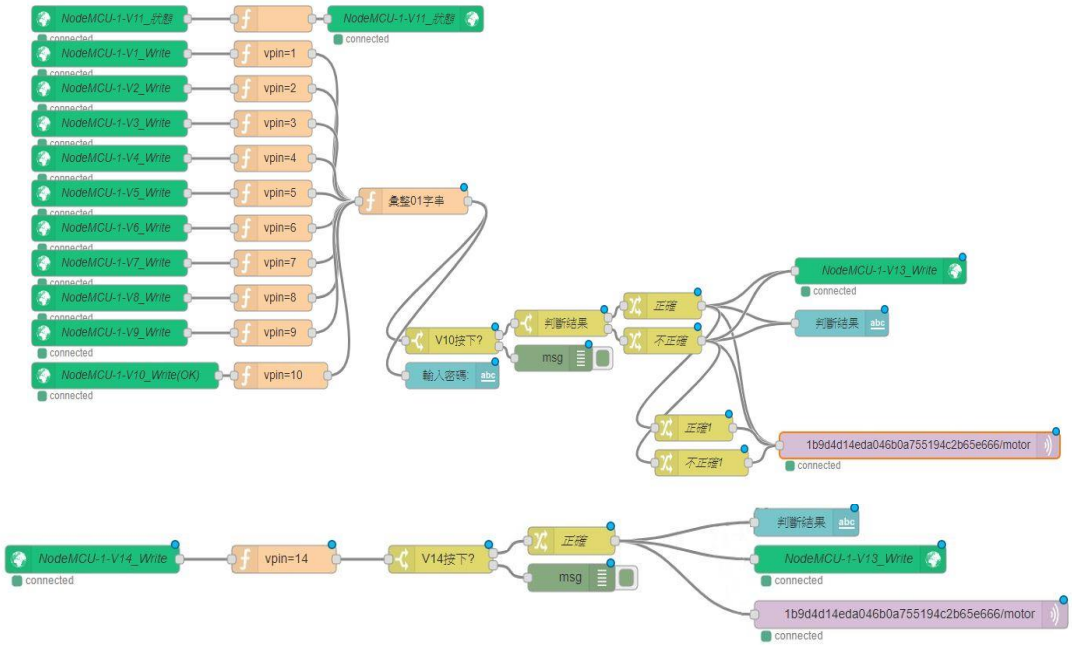


圖 5-25：NODE-RED 設計

第二、三關以超音波感測器 如圖 5-26 及紅外線感測器

如圖 5-27。Arduino 會驅動伺服馬達後，便都會回傳到 Blynk APP 上。

```
defineTaskLoop(T2) //超音波
{
  long cm;
  servopin11.write(40);
  digitalWrite(trip,LOW);
  delayMicroseconds(5);
  digitalWrite(trip,HIGH);
  delayMicroseconds(18);
  digitalWrite(trip,LOW);
  delayMicroseconds(5);
  duration = pulseIn(echo,HIGH);
  cm = duration / 58;
  if(cm<=3 && sta==1) {
    ++digit;
    tone(bz,400,100);
    sleep(1000);
      if (digit >= 3){
          ++sta;
          tone(bz,400,100);
          sleep(500);
          tone(bz,400,100);
          digit=0;
          sleep(100);
          servopin11.write(135);
          sleep(3000);
          servopin11.write(40);
          sleep(250);
          }
  }
}
```

圖 5-26：超音波程式

```
defineTaskLoop(T3) //紅外線
{
  servopin12.write(100);
   CodeWrite(digit2);
   val = digitalRead(inPin);
   if (val == HIGH) {
      sleep(250);
      Serial.println(val);
      val = digitalRead(inPin);
      if (val == LOW && sta==2) {
        ++digit2;
        tone(bz,400,100);
           sleep(1000);
        if (digit2 >= 3 ){
          ++sta;
         tone(bz,400,100);
         sleep(500);
         tone(bz,400,100);
         digit2=0;
         sleep(100);
         servopin12.write(10);
         sleep(3000);
         servopin12.write(100);
         sleep(250);
         }
       }
      }
}
```

圖 5-27：紅外線程式

第 6 章 物聯網之密室逃脫關卡解說

密室逃脫闖關關卡 本遊戲分為三大關卡

6.1 關卡一：密碼通關

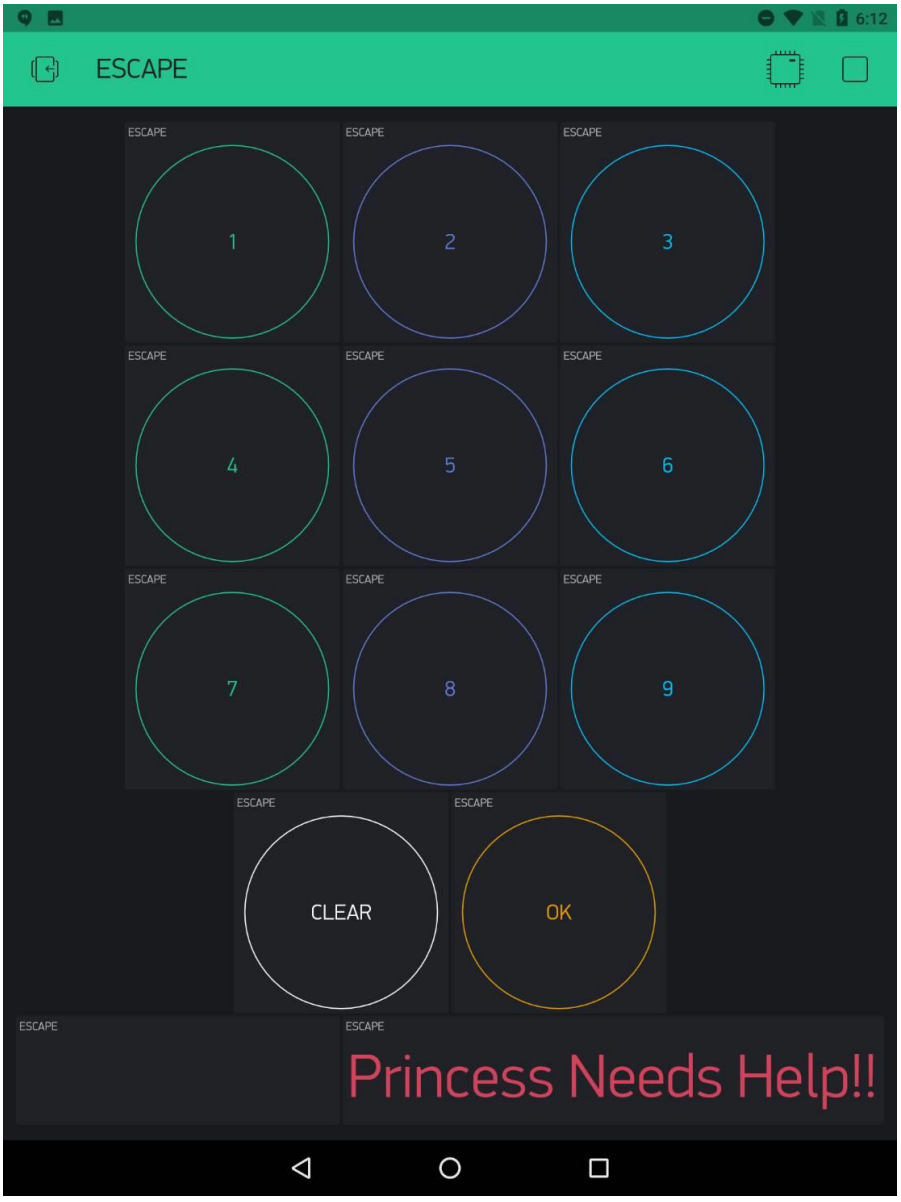


圖 6-1：輸入密碼

如果輸入錯誤顯示 Wrong；正確顯示 Pass 如圖 6-2

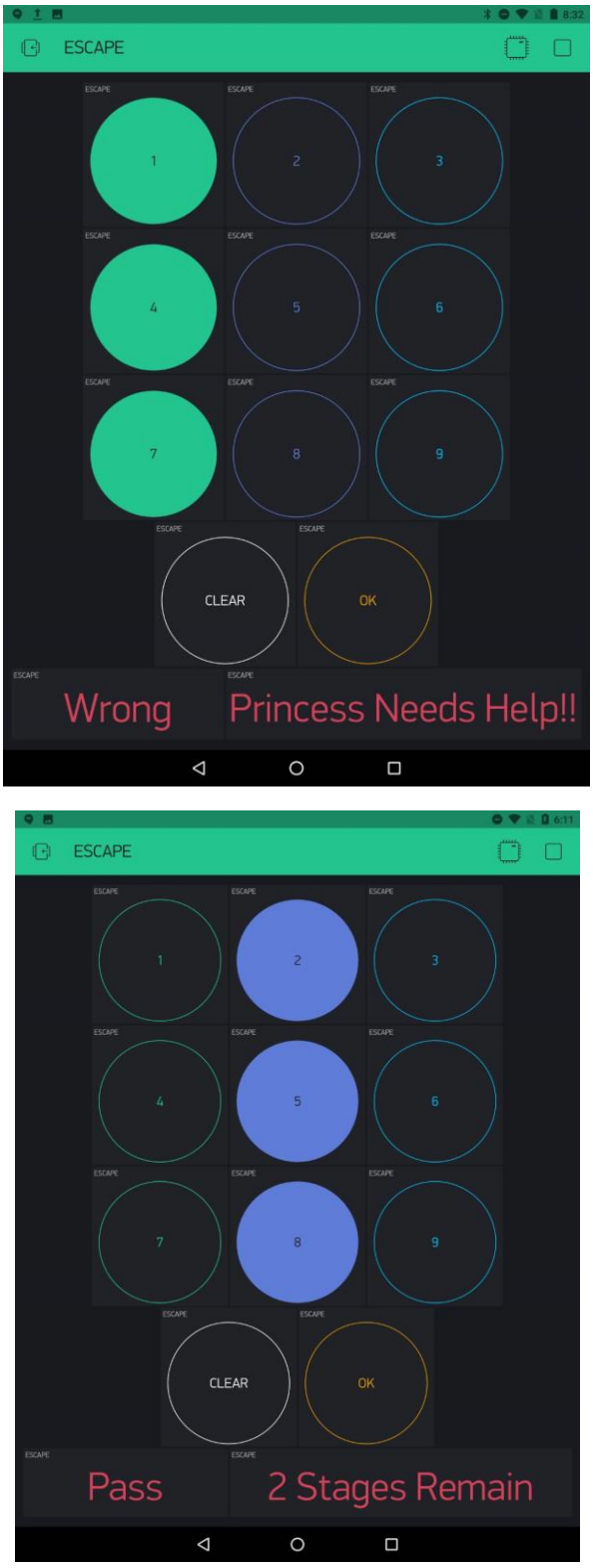


圖 6-2：密碼正確、錯誤

圖 6-3 為第一關初始介面

輸入密碼成功即可開啟密室一的門圖 6-4：第一關成功過關。

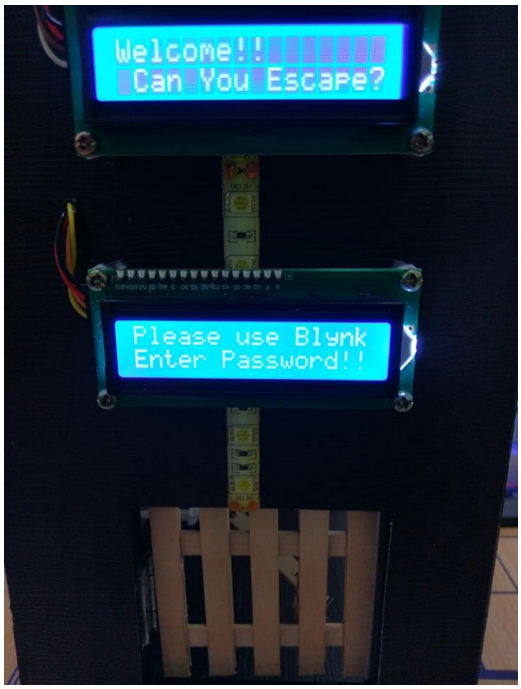


圖 6-3：第一關初始介面

圖 6-4：第一關成功過關

6.2 關卡二: 超音波通關

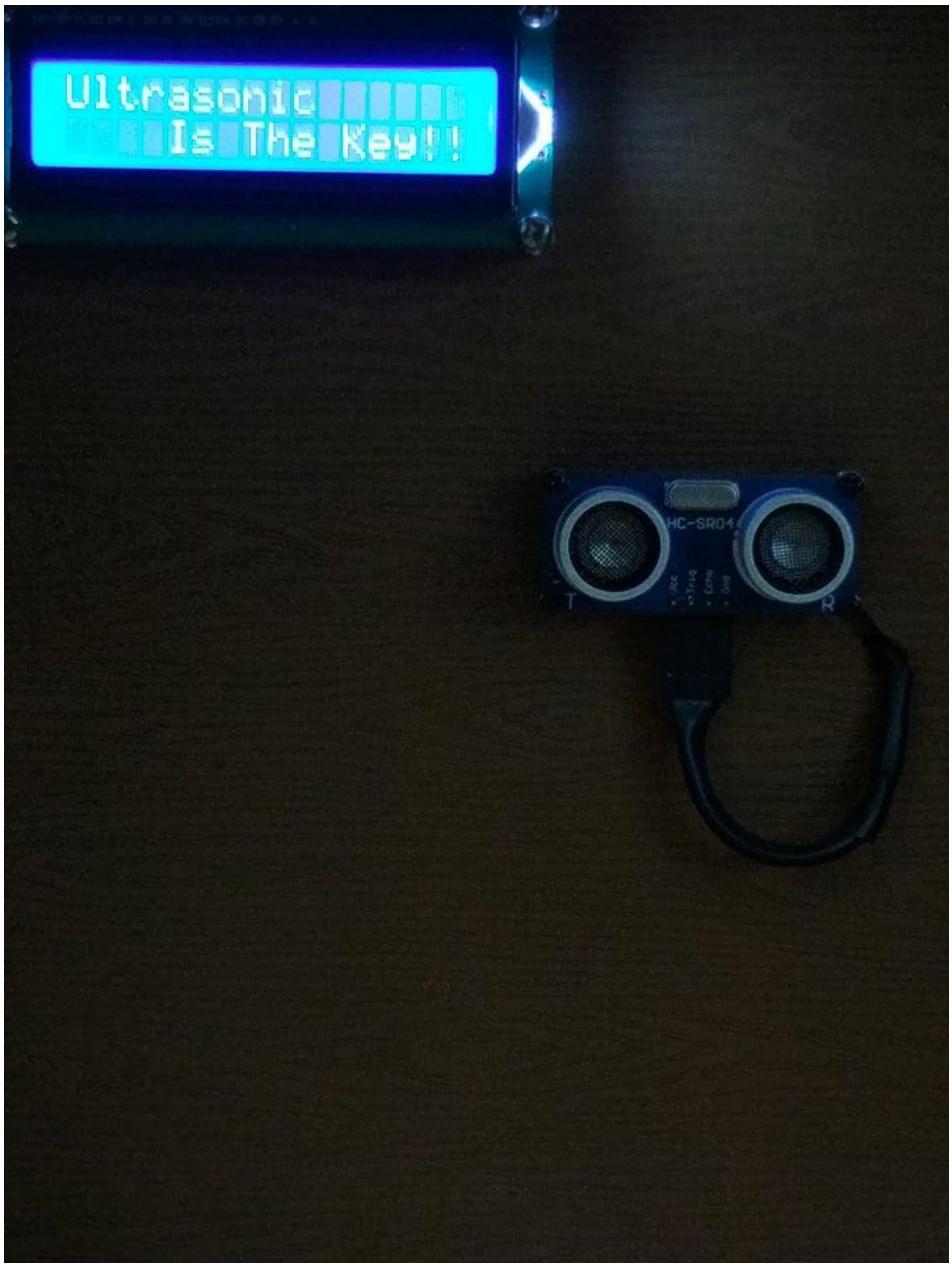


圖 6-5：第二關初始介面

利用超音波感測器，5 公分距離，感測 3 次如圖 6-6

即可開啟密室二的門如圖 6-7：第二關成功過關

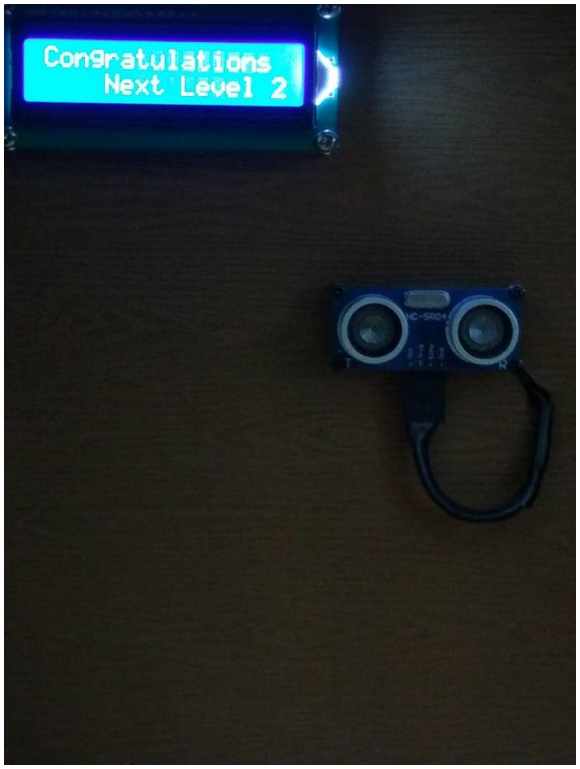


圖 6-6：超音波感測

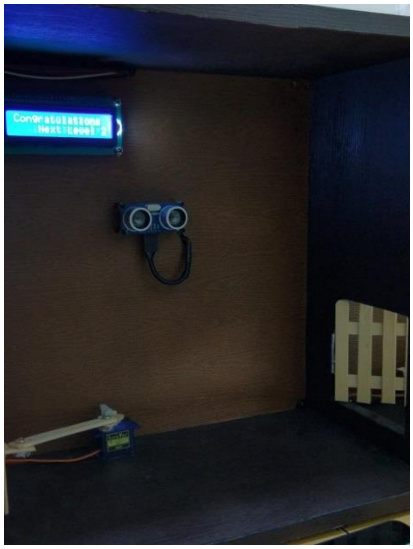


圖 6-7：第二關成功過關

第二關過關後回傳訊息。如圖 6-8

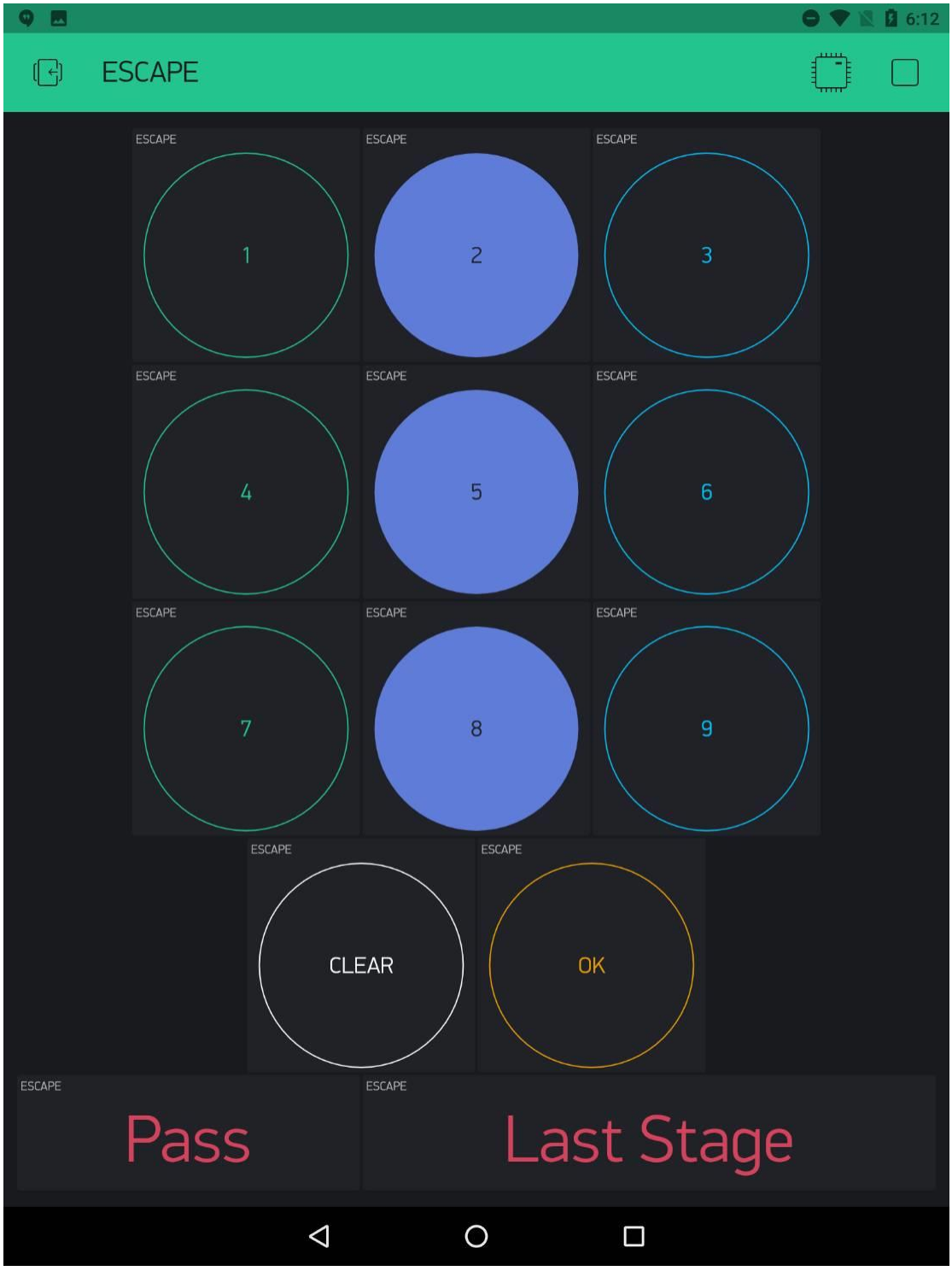


圖 6-8：第二關回傳訊息

6.3 關卡三: 紅外線通關

圖 6-9：第三關初始介面

利用紅外線，感測 3 次即可開啟密室三的門順利闖關成功

圖 6-10。如未闖關成功表示紅外線未偵測到指定的次數

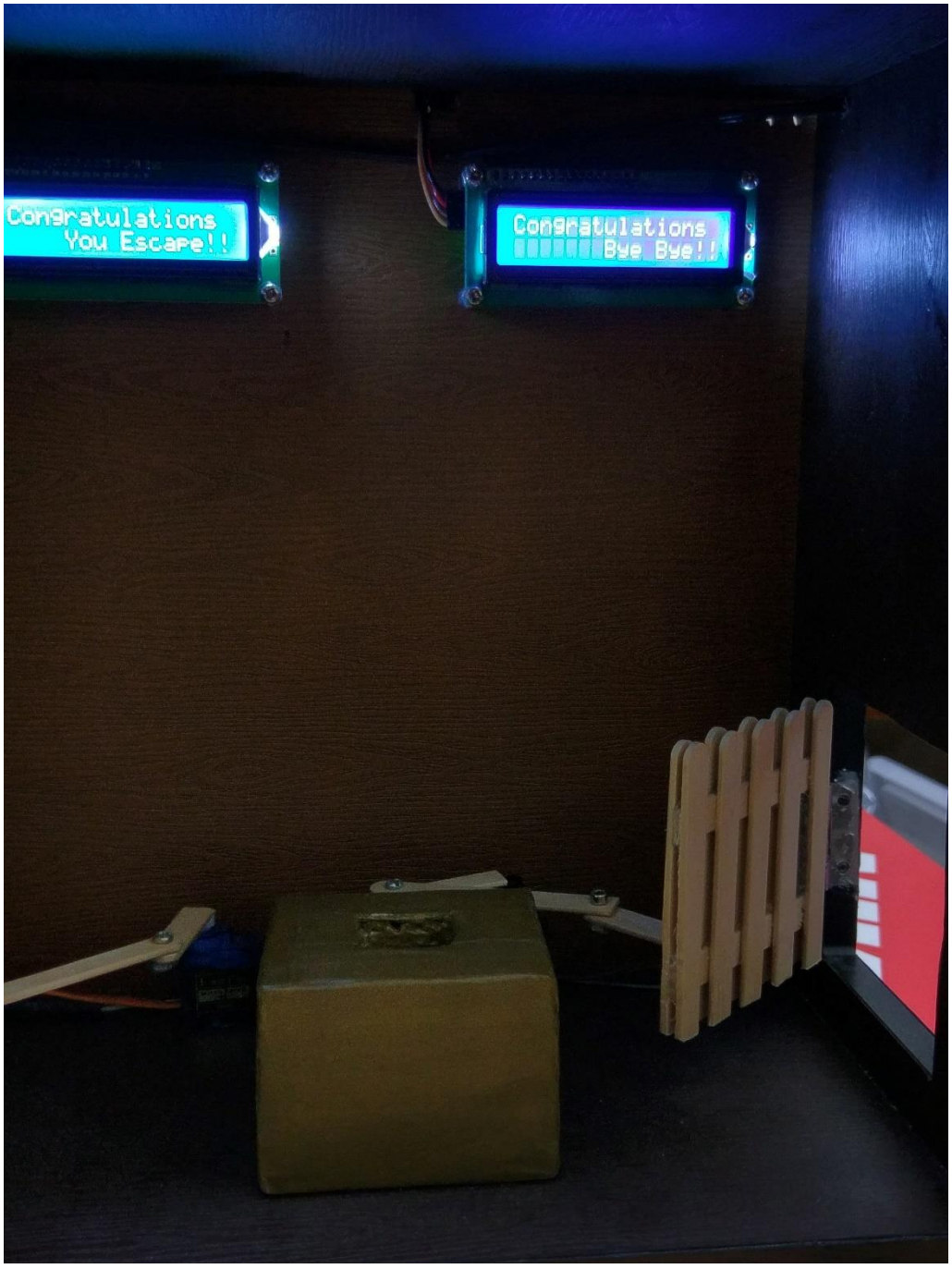


圖 6-10：遊戲結束

第三關過關後回傳訊息。如圖 6-11

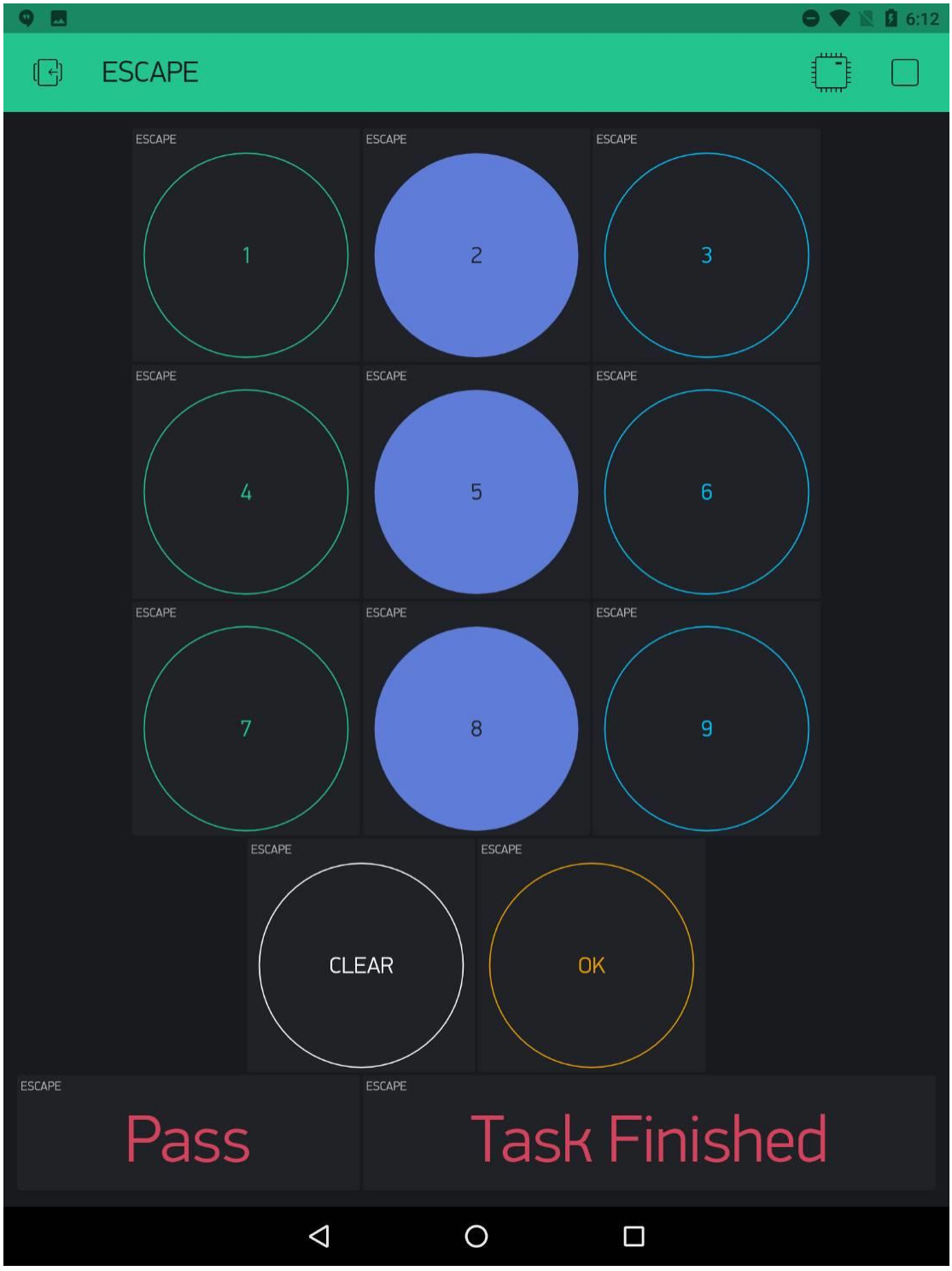


圖 6-11：第三關回傳訊息

PS. 本物聯網之密室逃脫闖關遊戲已有加設防止越關功能。

第 7 章 專題開發心得與問題討論

7.1 專題製作心得

組員一：賴昱辰(BN104102)

經過一整年的研究再研究修改再修改，終於把我們專題物聯網密室逃脫完成了。途中遇到很多困難，嘗試千百種不同的方式，雖然心神疲勞，曾經想要有放棄的念頭，但有著團隊組員合作及互相扶持下，都把疲勞、放棄已經擱在一旁。本專題有著多方向學習，想要跟別人不一樣，我們一心想要有著獨一無二的作品，當然付出的時間與精力本來就會較多；如何正確的製作 PCB 電路板、Arduino 及樹莓派之間的溝通來實現物聯網實際的運用。目前科技的趨勢；物聯網已經成為當紅焦點，藉由本專題也能讓我們更加以了解物聯網，更與科技融合接軌。

組員二：張世育(BN104099)

整體上來說我覺得專題製作的報告書寫起來是有難度的跟每個團員息息相關，團員之間要清楚知道自己所做的部份更重要的是對方做的也要瞭解清楚。

組員三：吳原程(BN104081)

藉由做這份專題，需要了解以及學習很多的東西，從實做的部分到文件報告書等等的工作項目，並不像剛開始想的那麼簡單，要一步一步地去了解，每個地方需要注意的細節。

組員四：蔡俊軒(BN104078)

參與這次的文案製作，讓我了解原來文案的製作、內容及格式有著很大的學問，並且在過程中發現平時沒有注意的細節問題，也許就會發現新的問題，讓我可以修改和了解問題，也是一次尋找錯誤問題的學習經驗。

7.2 製作電路板

我們並沒有專業的藥水及光照機，所以嘗試了很多次，使用顯影劑及蝕刻液的濃度比例以及光照感光板的距離的拿捏。圖 7-1：失敗例子。

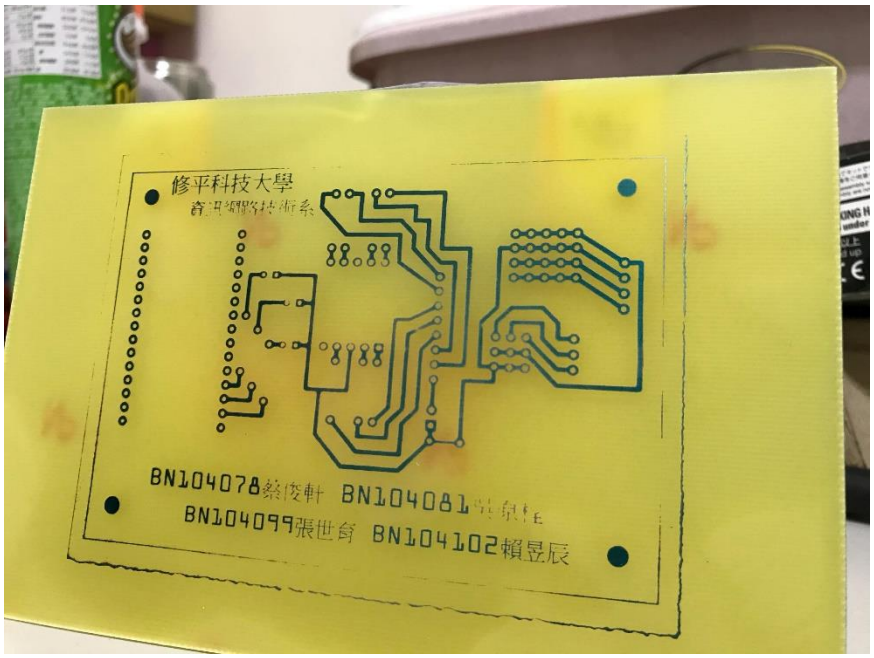


圖 7-1：失敗例子

正確製作感光板電路方式如下：

- 曝光-以透明膠帶固定好，上方 5~10 公分放置一盞日光燈進行曝光，如使用投影機屬於平行光源，光源強度足夠，可直接將感光電路板的綠色轉為藍色即曝光完成。
- 2.顯影劑-調製顯影溶液，劑與水比例 1：20，約 5 秒後可見些許綠色微泡，之後，每隔數秒輕搖一下容器使微泡散開，搖至線路非常清楚且不再有微粒冒起為止，整個顯影過程即完成。
- 3.蝕刻液-將氯化鐵液注入塑膠容器，並把感光板膜面朝上，輕搖塑膠容器，蝕刻至非線路部份的銅箔完全清靜。氯化鐵液的溫度以 20℃~50℃，溫度高速度快，加溫以間接加熱法。

7.3 多重 LCD 液晶顯示器接 I2C 與多工作業系統

使用 I2C 介面方便並連多個周邊硬體，提升系統的執行效率。採用 Arduino 的 Scoop 支援多通函式庫，並用有限狀態機器的模型撰寫程式，使系統執行更簡潔與效率。

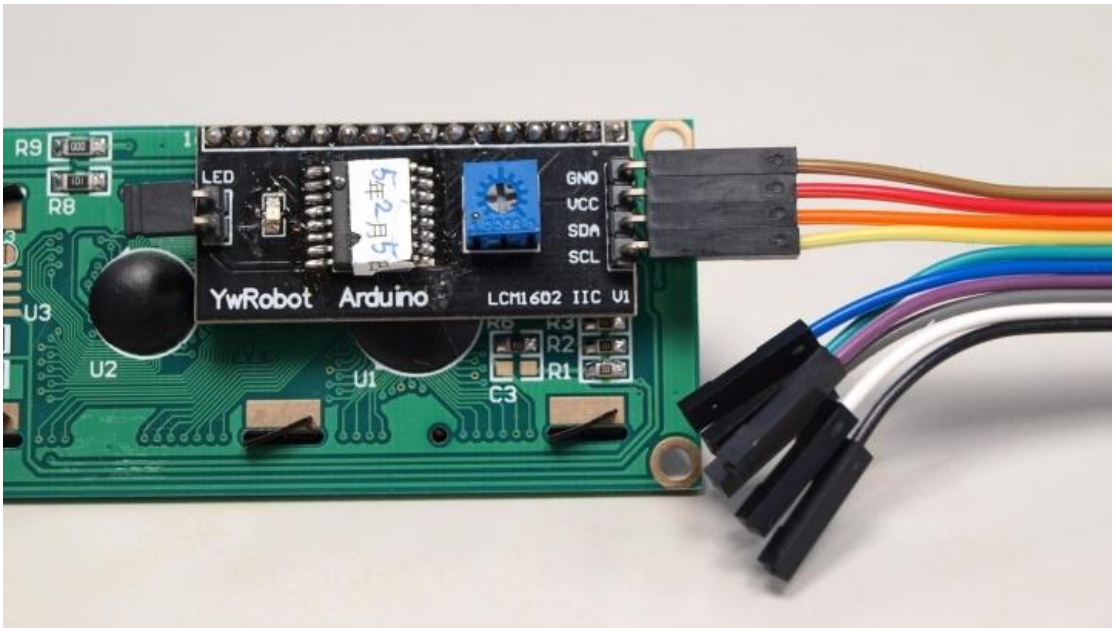


圖 7-2：I2C 接線圖

7.4 WiFi 基地台的外部網路(WAN)

本專題原本利用學校網路製作，遇到在 Blynk Server 無法重工建立專案，後來發現因學校網路會阻擋傳送認證憑證(Auth Token 的 E-MAIL)，後來以組員 4G LTE 手機做為 WiFi RT-AC66U 的 WAN Port 網路，才解決此專題的網際網路連線。

7.5 電源供應配置

因本專題實務中，具備諸多的 LCD 顯示器、感測模組以及外觀布置的 LED 測發光燈條，供電系統以安全起見，我們多安裝了一個可調式穩壓電源模組。

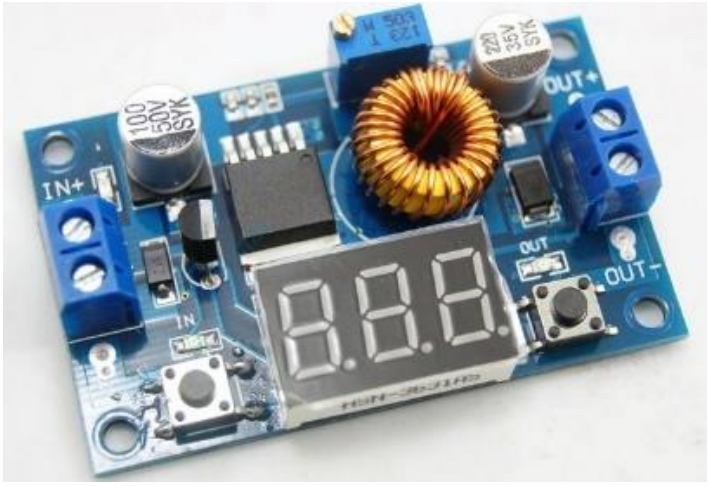


圖 7-3：電源模組

7.6 Blynk APP 更新

本專題 Blynk Server 中只支援到版本 Blynk 1.16.5。如果更新到更高版本，將無法登入 Server。更新 Blynk Server Java 到最新版本，也無法連線。

參考文獻

[1] 物聯網，讓萬物互聯暢通，
https://www-07.ibm.com/tw/blueview/2011apr/pdf/4_web.pdf

[2] 物聯網，(Internet of Things) 是什麼，
https://www.salesforce.com/tw/crm/internet-of-things/

[3] 張志勇、翁仲銘、石貴平、廖文華，物聯網概論，台北，碁峰資訊，2013

[4] 趙英傑，超圖解 Arduino 互動設計入門(第3版)，台北，旗標，2016

[5] 柯博文，Raspberry Pi最佳入門與實戰應用(第二版)，台北，碁峰，2015

[6] ESP8266指南，角色、功效及開發途徑，
https://makerpro.cc/2015/08/esp8266-guideline/

[7] 樹莓派，https://www.raspberrypi.com.tw/tag/%E6%87%89%E7%94%A8/

[8] Firebase 與 Raspberry Pi 製作簡單的物聯網 iOS 程式，
https://www.appcoda.com.tw/swift-raspberry-pi/

[9] 電路板的製作流程，
http://ezphysics.nchu.edu.tw/prophys/ael/File/101/board.pdf

[10] 輔助科技，快速製作電路板，智慧型機電輔具實驗室，
http://www.ym.edu.tw/~chyu/projects-pcb.htm

[11] Arduino，官網，https://www.arduino.cc/en/Main/Software

[12] 樹莓派，https://www.raspberrypi.com.tw/

[13] Arduino到AVR晶片，
http://programmermagazine.github.io/201406/htm/article1.html

[14] 硬Pi製作，下載PiOS作業系統 ，
https://sites.google.com/site/raspberrypidiy/raspberry-pi/images

[15] 給你滿滿的樹莓派教學、實作與應用大平台，
http://blog.itist.tw/p/how-to-study-raspberry-pi.html

[16] 硬Pi製作，映像檔工具拷貝到SD卡，
https://sites.google.com/site/raspberrypidiy/raspberry-pi/os2

[17] MQTT，應用手冊，
http://www.phelipu.com.tw/download/MQTT_UserManual__tw.pdf

[18] Arduino，基本介紹，
http://newsletter.ascc.sinica.edu.tw/news/read_news.php?nid=2782

[19] 從Arduino到AVR晶片，
http://programmermagazine.github.io/201406/htm/article1.html