

雲端智慧窗簾

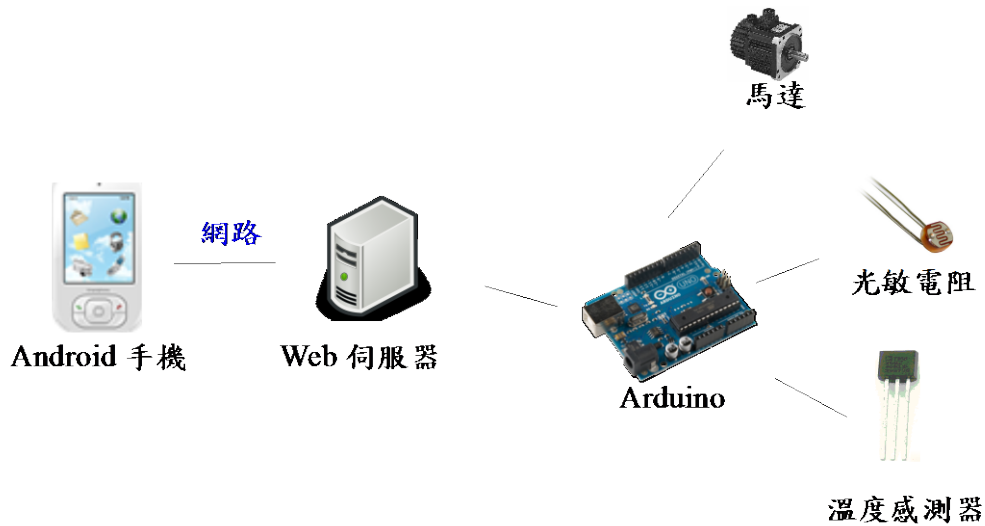
壹、系統架構圖

一、原本預計架構



圖一、原本預計架構圖

二、實際執行架構



圖二、原本預計架構圖

貳、分工配置

政斌、辰希：手機端畫面配置、程式撰寫。

裕民：終端感測器、Web 伺服器程式選寫。

參、端感測器部分

一、終預期功能說明

- (一) 窗邊照度太亮, 且室溫超過 27 度 C 時, =>窗自動拉滿
- (二) 窗邊照度太亮, 且室溫超過 27 度 C 時, =>窗自動拉滿
- (三) 窗邊照度太暗, 客廳電燈自動開啟 (自動模式時)
- (四) 有人按下關窗簾, 窗簾全關
- (五) 窗邊照度太暗, 且室溫低於 25 度時, =>窗簾自動收回 (自動模式時)
- (六) 窗簾拉出及收回期間超過 10 秒時斷電並警告。

太亮=>超過大於 500Lux

拉滿=>關到定位

太暗=>表示低於 100Lux

收回=>移動到定位

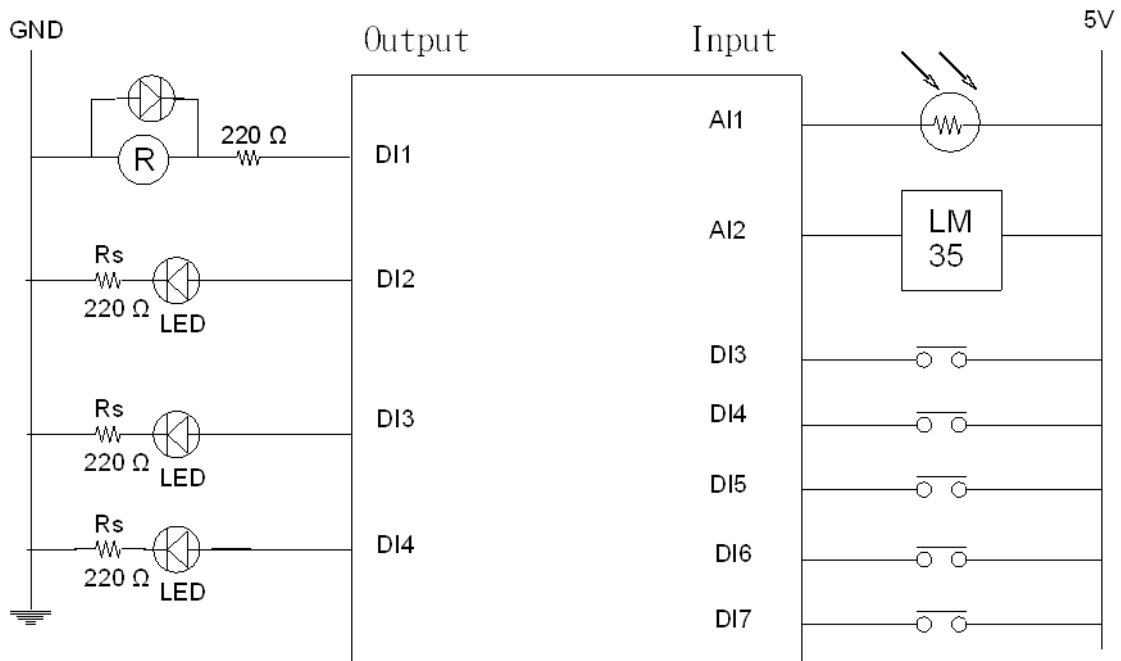
二、I/O 配置表

Input		Output	
A11	光敏電阻 (類別裝置輸入 Analog)	D01	馬達 (數位裝置輸出 Digital)
A12	溫度感測器 (類別裝置輸入 Analog)	D02	電燈點滅(接點) (數位裝置輸出 Digital)
DI3	模式切換開關 (自動/手動模式切換開關) (類別置輸入 Analog)	D03	馬達運作(指示) (數位裝置輸出 Digital)
DI4	窗簾開定位感測 (數位裝置輸入 Digital)	D04	故障指示燈 (數位裝置輸出 Digital)
DI5	窗簾關定位感測 (數位裝置輸入 Digital)		
DI6	窗簾開按鈕 (數位裝置輸入 Digital)		
DI7	窗簾關按鈕 (數位裝置輸入 Digital)		

表一、I/O 配置表

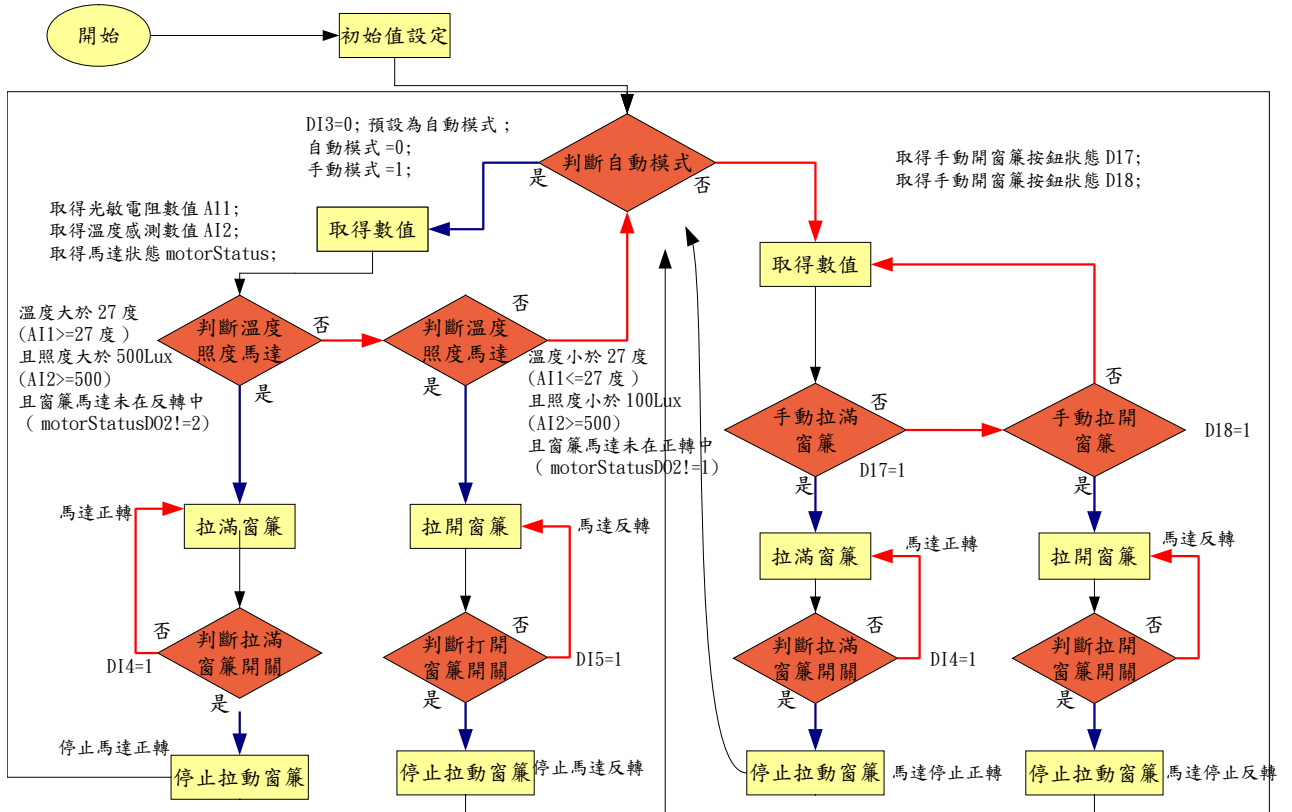
三、填寫 I/O 配置表架構規劃 (含電路板)

表一、各項 I/O 配置圖表



圖三、I/O 配置表

四、流程圖



圖四、系統流程圖

五、變數規劃宣告

```
int photocellPin = A0; // AI1 光敏電阻 pin 腳
int photocellReading; // 紀錄光敏電阻 pin 腳得的數值

int tempPin = A2; // AI2 溫度感測器 pin 腳
float temp = 0; // 紀錄溫度感測器 pin 腳得的數值

//DI3 手動模式切換開關
//紀錄手動模式切換開關數值(0 是自動;1 是手動)

int openSwitchPin = 12; //DI4 窗簾開定位感測
int openSwitch = 0;
int closeSwitchPin = 7; //DI5 窗簾關定位感測
int closeSwitch = 0;

int openBtnPin = 12; //DI7 窗簾開按鈕
int openBtn = 0;

int closeBtnPin = 2; //DI8 窗簾關按鈕
int closeBtn = 0;

int motorEnablePin = 4; //控制馬達啟動腳位
int motorPositivePin = 5; //控制馬達正轉
int motorNavivePin = 6; //控制馬達反轉

int lightPin = 8; //D02 電燈點滅
int motorLightPin = 12; //D02 馬達運作電
int faultLightPin = 13; //D03 故障指示燈
```

六、程式設計

```
int photocellPin = A0; // 光敏電阻使用 A0 腳位
int tempPin = A2; // 溫度感測器 A2 腳位

int autoAndManualSwitch = 10; // 自動與手動轉換(自動訊號為 0;手動訊號為 1)

int openSwitchPin = 8; // 窗簾拉開感測腳位,使用微動開關
int closeSwitchPin = 9; // 窗簾拉關感測腳位,使用微動開關

int openBtnPin = 2; //手動拉開窗簾
int closeBtnPin = 3; //手動拉關窗簾

int motorEnablePin = 4; //啟動馬達腳位
int motorPositivePin = 5; //控制馬達正轉腳位
int motorNavivePin = 6; //控制馬達反轉腳位
int motorStatus = 0; // 紀錄馬運作狀態(default stop = 0; native = 2;positive = 1;)

int lightPin = 11; //室內電燈打開
int motorLightPin = 12; //馬達運作指示燈
int faultLightPin = 13; //系統發生錯誤警示燈
int faultWarm = 0; //記錄系統發生錯誤狀態

void setup(void) {
    // We'll send debugging information via the Serial monitor
    Serial.begin(9600);
    // 設定 g_nEnablePin 為數位輸出，
    pinMode( motorEnablePin, OUTPUT );
    //
    // 設定 g_nEnablePin 高低電位來控制是否啟用馬達輸出
    digitalWrite(motorEnablePin, HIGH );

    pinMode(autoAndManualSwitch, INPUT);
```

```
pinMode(lightPin, OUTPUT);
pinMode(motorLightPin, OUTPUT);
pinMode(faultLightPin, OUTPUT);

pinMode(openBtnPin, INPUT);
pinMode(closeBtnPin, INPUT);
}
```

```
float nowTemp = 0;
int nowLight = 0;
int mode = 0;//0 : auto;1:manual
int nowMotorStatus = 0;//0:stop;1:open;2:close
int curtainStatus = 0;//0:open;1:close
```

```
void loop() {;
```

```
if ( getAutoOrManualSwitch() == 0 ) { //自動模式
```

```
float temp = getTemperature();
nowTemp = temp;
int light = getLDR();
nowLight = light;
```

```
delay(3000);
printMsg();
mode = 0;
//Serial.print(" auto mode");
//Serial.println();
//Serial.print(motorStatus+'\0');
```

```
//照度大於 500Lux 以及 溫度超過 29 度 以及馬達目前狀態不能在反轉狀態 以及現在就是不是開啟
狀態
```



```

if ( (light > 300) && (temp > 29) && (motorStatus != 2) && (curtainStatus != 0) ) {
    digitalWrite(lightPin, LOW);
    //窗簾開定位感測
    while ( getOpenSwitch() != 1) { //沒有定位到就繼續馬達轉動
        curtainStatus = 0; //open status
        printMsg();
        setMotorNativie(); //馬達反轉(關閉窗簾)
    }
    nowMotorStatus = 0;
}
//照度小於 500Lux 以及 溫度低過 27 度 以及馬達目前狀態不能在正轉狀態 以及現在就是不是關
開狀態
else if ( (light < 200) && (temp < 27) && (motorStatus != 1) && (curtainStatus != 1) ) {
    //室內燈關閉
    digitalWrite(lightPin, HIGH);
    //窗簾關定位感測
    while ( getCloseSwitch() != 1) {
        curtainStatus = 1; //open status
        printMsg();
        setMotorPositive(); //馬達正轉(打開窗簾)
    }
    nowMotorStatus = 0;
}
}
else { //手動模式
    // Serial.print("manual mode");
    // Serial.println();
    mode = 1;
    printMsg();
    //讀取目前窗簾開按鈕是否被按下
    if ( getButton(openBtnPin) == 1 ) {
        // Serial.print("openButton : 1"); Serial.println();
        digitalWrite(lightPin, LOW);

```

```

while ( getOpenSwitch() != 1) {
    curtainStatus = 0;//open status
    printMsg();
    setMotorPositive();
}
nowMotorStatus = 0;
}
//讀取目前窗簾關按鈕是否被按下
else if ( getButton(closeBtnPin) == 1) {
// Serial.print("closeBtnPin : 1"); Serial.println();
digitalWrite(lightPin, HIGH);

while ( getCloseSwitch() != 1) {
    curtainStatus = 1;//close status
    printMsg();
    setMotorNativie();
}
nowMotorStatus = 0;
}
}
}
}

```

```

void printMsg() {

```

```

//motorStatus = 1; //1:開啟正轉狀態；2:關閉反轉；0：停止
Serial.print("temp:");
Serial.print(nowTemp);
Serial.print(";");
Serial.print("light:");
Serial.print( nowLight);
Serial.print(";");

```

```
Serial.print("mode:");  
Serial.print(mode);  
Serial.print( ");");  
Serial.print("motorStatus:");  
Serial.print(nowMotorStatus);  
Serial.print( ");");  
Serial.print("curtainStatus:");  
Serial.print( curtainStatus);  
Serial.print( ");");  
Serial.println();  
}
```

//取得自動/手動模式

```
int getAutoOrManualSwitch() {  
    int openSwitch = digitalRead(autoAndManualSwitch);  
    // Serial.println(openSwitch);  
    // delay(1000);  
    return openSwitch;  
}
```

//取得目前照度

```
int getLDR() {  
    int photocellReading = analogRead(photocellPin);  
    //Serial.print("Analog reading = ");  
    //Serial.print(photocellReading); // the raw analog reading  
    // We'll have a few thresholds, qualitatively determined  
    if (photocellReading < 10) {  
        //Serial.println(" - Dark");  
    } else if (photocellReading < 200) {  
        //Serial.println(" - Dim");  
    } else if (photocellReading < 500) {  
        // Serial.println(" - Light");  
    } else if (photocellReading < 800) {  
        //Serial.println(" - Bright");  
    }
```

```

    } else {
        // Serial.println(" - Very bright");
    }
    delay(1000);
    return photocellReading;
}

//取得目前溫度
float getTemperature() {
    float temp = analogRead(tempPin);
    temp = temp * 0.48828125;
    // Serial.print("Temp.");
    // Serial.println(temp);
    delay(1000);
    return temp;
}

//取得窗簾開定位感測狀態
int getOpenSwitch() {
    int openSwitch = digitalRead(openSwitchPin);
    //Serial.println(openSwitch);
    //若為 1,表示窗簾已達定點
    if ( openSwitch == 1 ) {
        //停止馬達轉動
        digitalWrite(motorLightPin, LOW);
        analogWrite( motorPositivePin, 0 );
        analogWrite( motorNavivePin, 0 );
        motorStatus = 0;//馬達狀態為停止
    }
    curtainStatus = 0;//窗簾開啟
    delay(1000);
    return openSwitch;
}

```

```

//取得窗簾關定位感測狀態
int getCloseSwitch() {
    int closeSwitch = digitalRead(closeSwitchPin);
    // Serial.print("closeSwitch:");
    // Serial.println(closeSwitch);
    //若為 1,表示窗簾已達定點
    if ( closeSwitch == 1 ) {
        digitalWrite(motorLightPin, LOW);
        analogWrite( motorPositivePin, 0 );
        analogWrite( motorNavivePin, 0 );
        motorStatus = 0;
    }
    curtainStatus = 1;//窗簾關閉
    delay(1000);
    return closeSwitch;
}

```

```

//取得指定腳位數位訊號狀態
int getButton(int pin) {
    int openBtn = digitalRead(pin);
    // Serial.print(openBtn);
    delay(1000);
    return openBtn;
}

```

```

//設定馬達正轉//open
void setMotorPositive() {
    // Serial.print("motor positive");Serial.println();
    //馬達正轉
    digitalWrite(motorLightPin, HIGH);
    analogWrite( motorPositivePin, 60 );
    analogWrite( motorNavivePin, 0 );
    motorStatus = 1;//設定馬達為正轉狀態
    nowMotorStatus = 1;
}

```

```
}
```

```
//設定馬達反轉//close
```

```
void setMotorNativie() {
```

```
    // Serial.print("motor native");Serial.println();
```

```
    //馬達反轉
```

```
    digitalWrite(motorLightPin, HIGH);
```

```
    analogWrite( motorPositivePin, 0 );
```

```
    analogWrite( motorNavivePin, 60 );
```

```
    motorStatus = 2; //設定馬達為反轉狀態
```

```
    nowMotorStatus = 2;
```

```
}
```

七、電子元件
(一) 微動開關

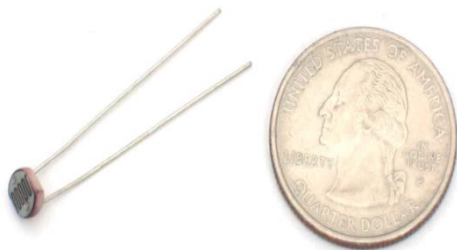


圖五、微動開關

數值	狀態
1	開關按下狀態
0	開關放開狀態

表二、微動開關狀態

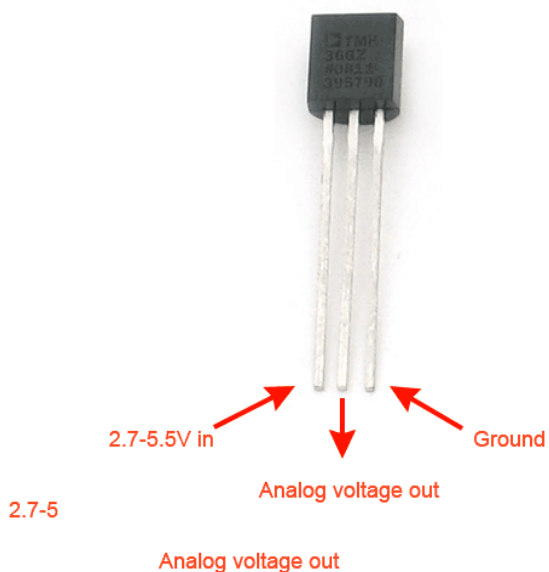
(二) 光敏電阻



圖六、光敏電阻

Ambient light like...	Ambient light (lux)	Photocell resistance (Ω)	LDR + R (Ω)	Current thru LDR + R	Voltage across R
Dim hallway	0.1 lux	600K Ω	610 K Ω	0.008 mA	0.1 V
Moonlit night	1 lux	70 K Ω	80 K Ω	0.07 mA	0.6 V
Dark room	10 lux	10 K Ω	20 K Ω	0.25 mA	2.5 V
Dark overcast day / Bright room	100 lux	1.5 K Ω	11.5 K Ω	0.43 mA	4.3 V
Overcast day	1000 lux	300 Ω	10.03 K Ω	0.5 mA	5V

(三) 溫度感測器



圖七、LM38 溫度感測器

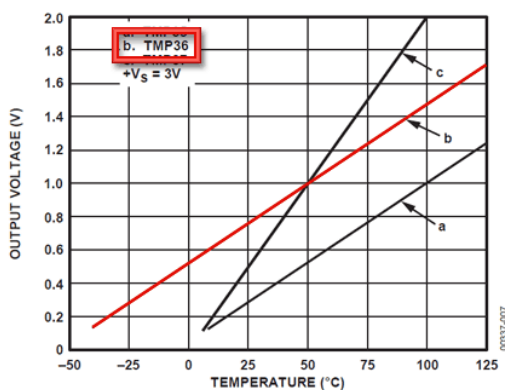


Figure 6. Output Voltage vs. Temperature

表三、光敏電阻電阻與溫度線性對照圖

規則如下：

$$\text{milliVolts} = (\text{reading from ADC}) * (5000/1024) ;$$

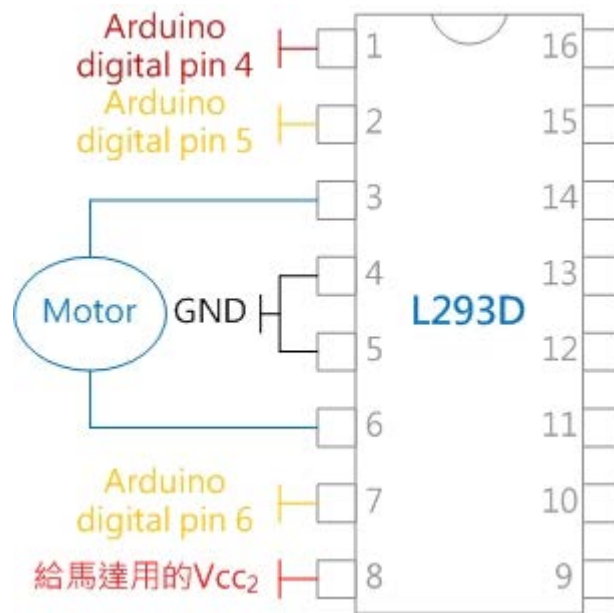
$$\text{Temp} = [(\text{milliVolts}) - 500] / 10 ;$$

經過換算為

$$\text{Temp} = \text{Arduino 讀出的數值} * 0.48828125;$$

Temp 即為現在量測的溫度。

(四) 溫度感測器直流馬達控制器(LD293D)

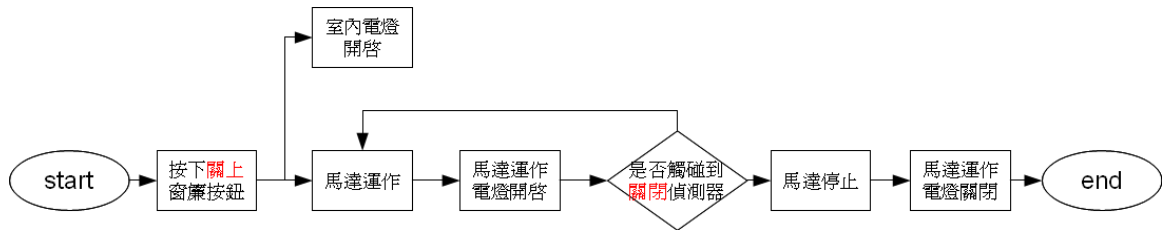


圖八、馬達控制器

使用 Arduino Pin 4 插在 L293D 晶片的 Pin 1 來控制 HIGH = Enable 或 LOW = Disable 決定是否啟用馬達控制。使用 Arduino Pin 5(PWM) 插在 L293D 晶片的 Pin 2 和 Arduino Pin 6(PWM) 插在 L293D 晶片的 Pin 7 來作為訊號輸入。L293D 晶片的 Pin 4 和 Pin 5 要和 Arduino 的 Pin GND 接在一起共地。L293D 晶片的 Pin 8 要外接獨立電源給晶片使用，驅動電源要 4.5V ~ 7V，使用兩顆乾電池(3V)，馬達可以正常運作。

八、操作流程

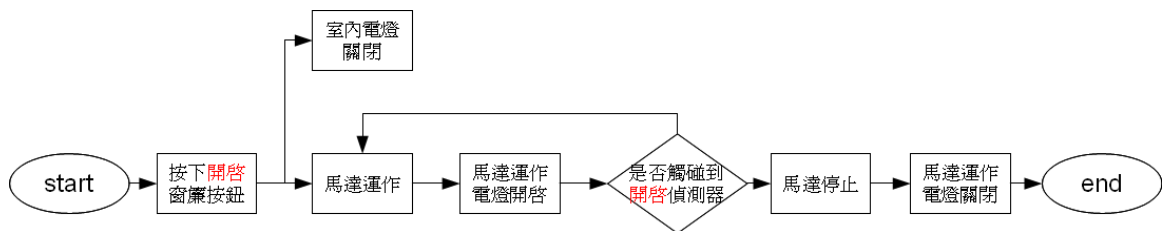
(一) 手動模式-拉上窗簾



圖九、手動流程-拉上窗簾

按下關閉按鈕後，分為開啟室內電燈以及窗簾馬達運作，窗簾馬達運作指示燈啟動，一定判斷到窗簾到達關閉的定位點後，再停止馬達運作，馬達指示燈關閉，結束動作。

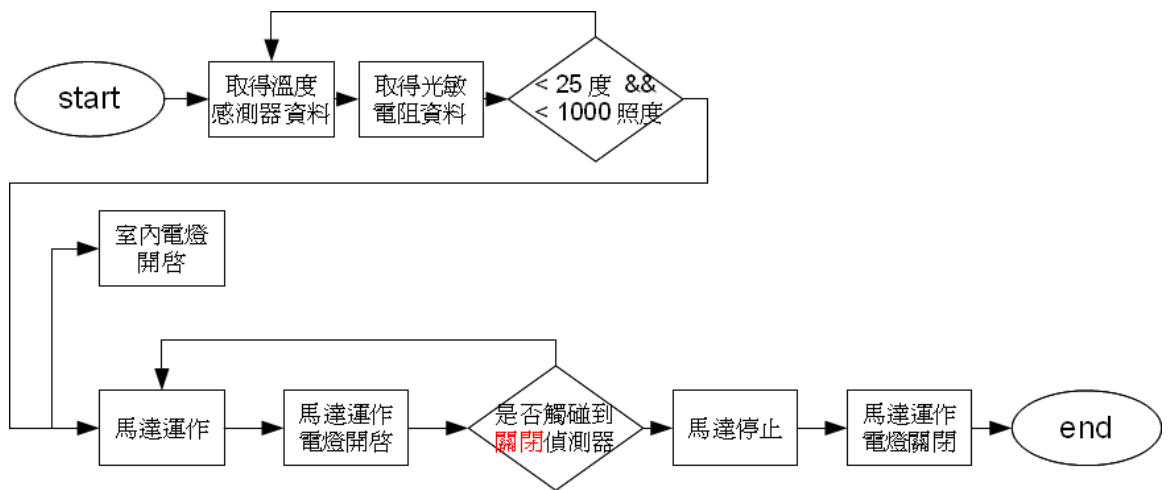
(二) 手動模式-開啟窗簾



圖十、手動流程-開啟窗簾

按下開啟按鈕後，分為關閉室內電燈以及窗簾馬達運作，窗簾馬達運作指示燈啟動，一定判斷到窗簾到達關閉的定位點後，再停止馬達運作，馬達指示燈關閉，結束動作。

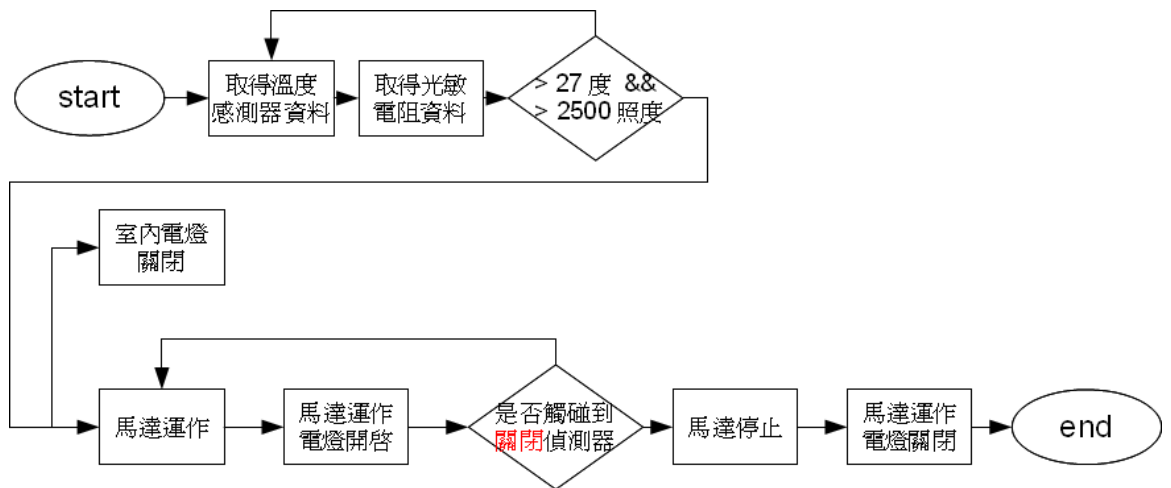
(三) 自動模式-拉上窗簾



圖十一、手動流程-拉上窗簾

自動取得溫度感測器以及光敏電阻上的資訊，判斷是否大於 27 度以及 2500 照度，若是表示溫度太高以及日照過度，天氣為大太陽天，自動開啟窗簾，降低室內溫度。

(四) 自動模式-開啟窗簾

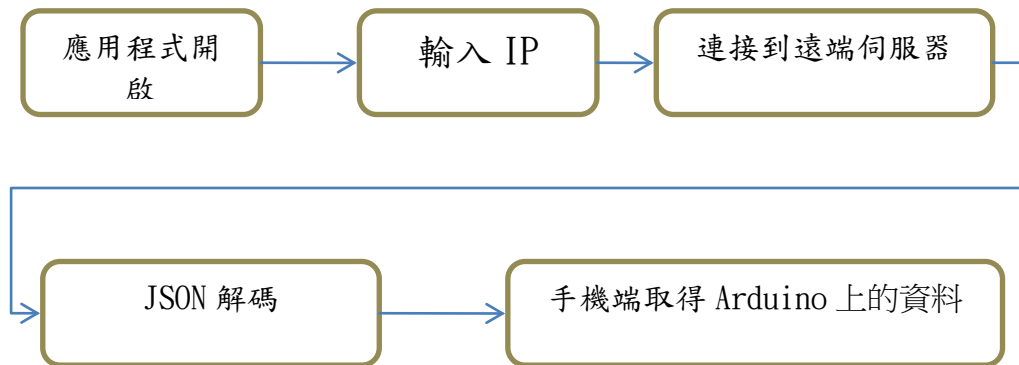


圖十二、手動流程-開啟窗簾

自動取得溫度感測器以及光敏電阻上的資訊，判斷是否小於 25 度以及 1000 照度，若是表示溫度降低以及日照度減度，天氣為陰天，自動拉上窗簾。

肆、手機端應用程式部分

一、動作流程圖



圖十三、動作流程圖

程式開啟後，要先輸入 Sever IP 進行連接，之後會對網頁上的資料做 JSON 解碼，並且將解碼後資料傳回手機端。

二、What is JSON?

(一) JSON 應用在哪些地方?

(二) 如何建立 JSON 字串

可以透過底下規則來建立 JSON 字串

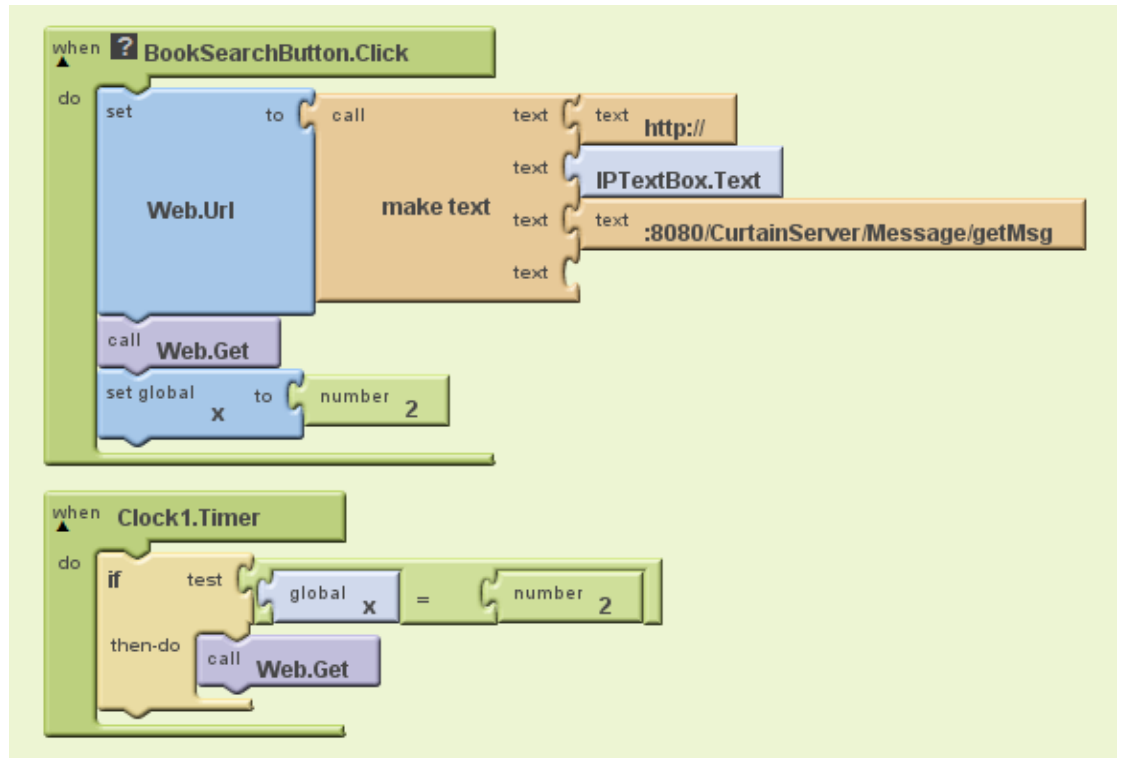
1. JSON 字串可以包含陣列 Array 資料或者是物件 Object 資料
2. 陣列可以用 [] 來寫入資料
3. 物件可以用 {} 來寫入資料
4. name / value 是成對的，中間透過 (:) 來區隔

物件或陣列的 value 值可以如下：

1. 數字（整數或浮點數）
2. 字串（請用 “” 括號）
3. 布林函數 (boolean) (true 或 false)
4. 陣列（請用 []）
5. 物件（請用 {}）
6. NULL

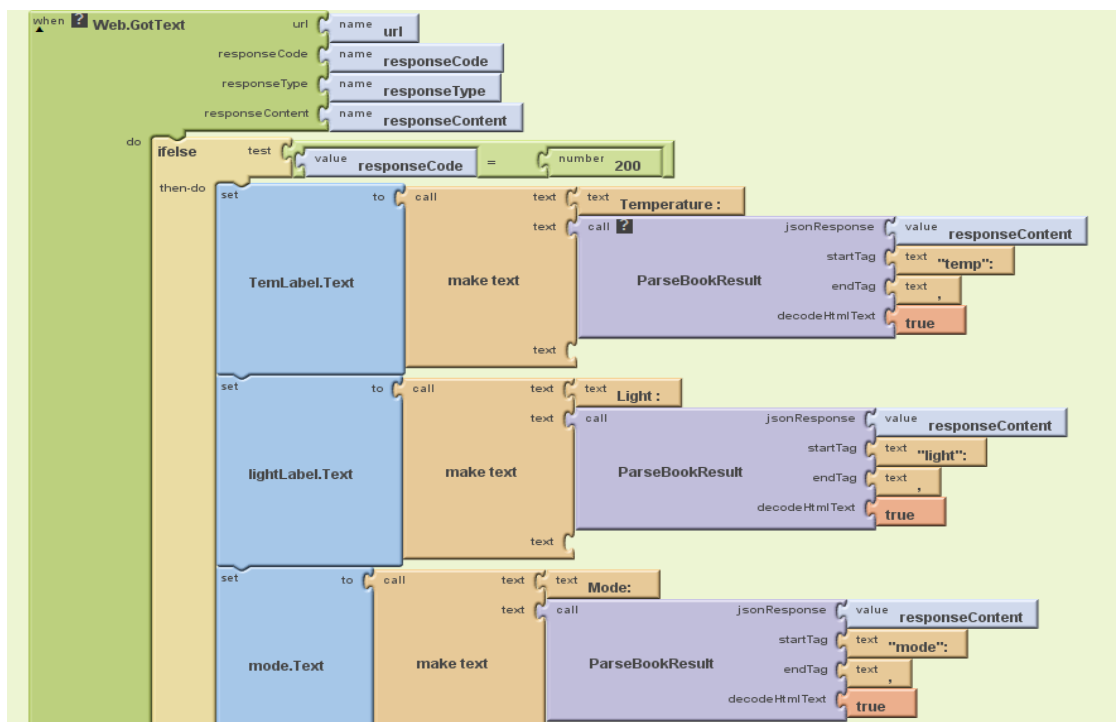
三、 程式碼圖

(一) 輸入 IP



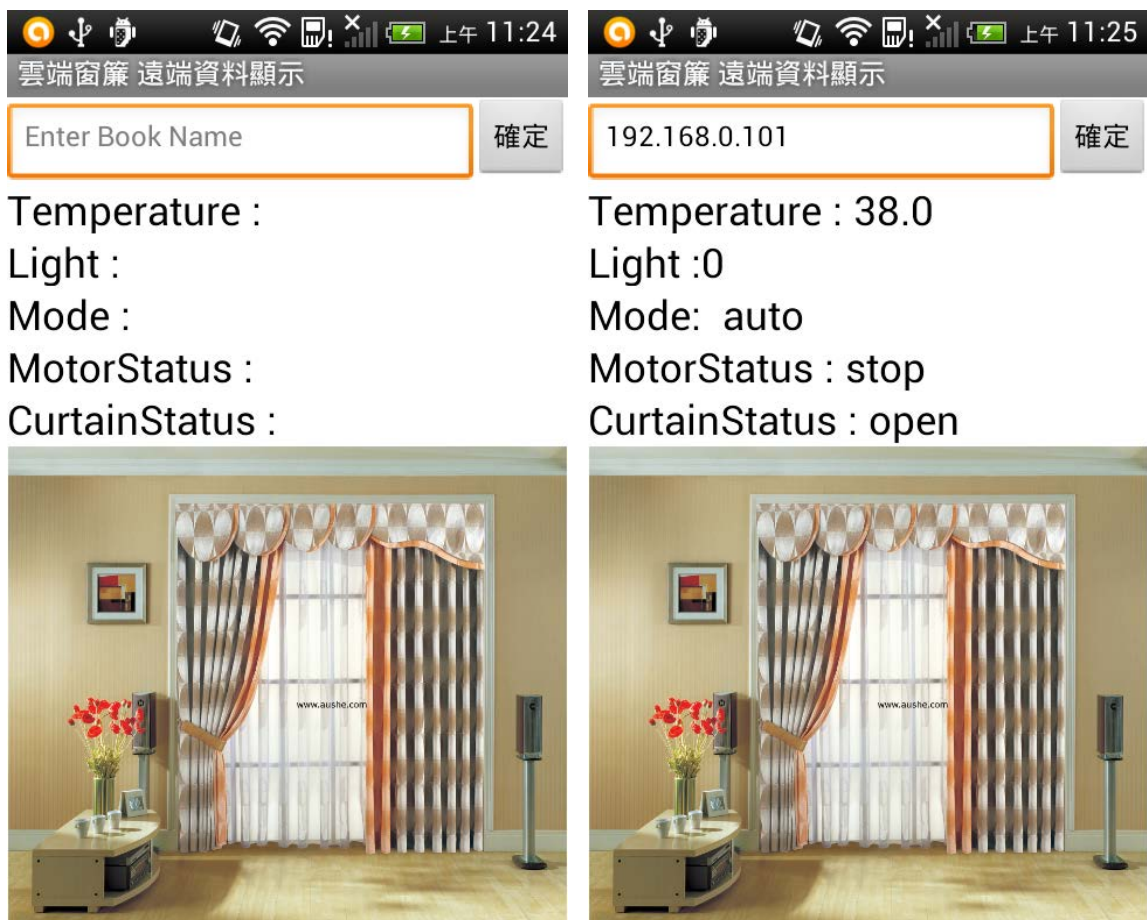
圖十四、輸入 IP

(二) JSON 解碼取得資料



圖十五、JSON 解碼取得資料

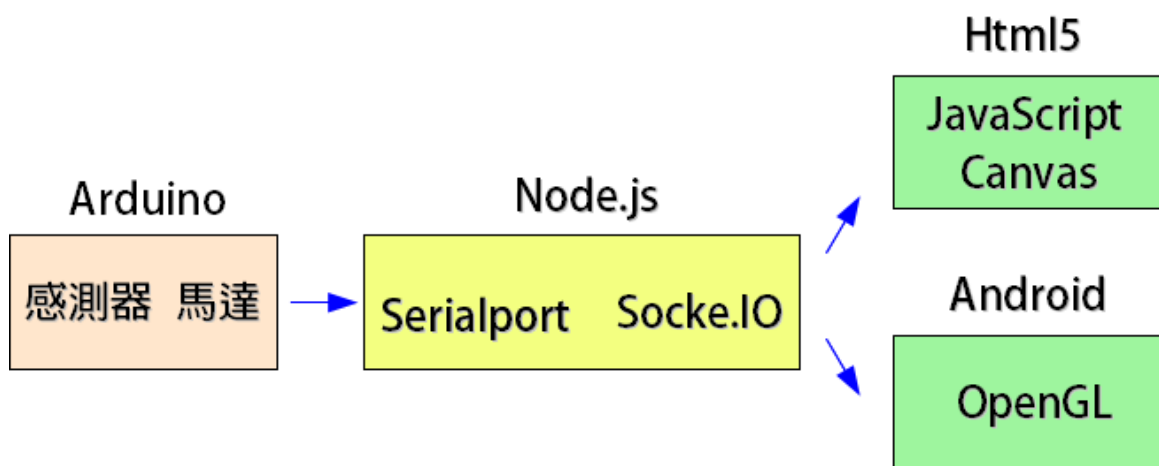
四、 手機端實體圖



圖十六、實機實際畫面

伍、未來擴充功能

- 一、可以設定偵測的溫度以及亮度。
- 二、加入中央氣象局目前的溫度以提供確認感測器的正確性。
- 三、使用 GCM(Google Cloud Message)，在自動模式下，窗簾有打開或者關上的動作時，以簡訊發送通知。
- 四、目前使用 Serial Port Listener+Web Server 方式連結，由手機定時送出要求取得資料模式，可以改成以 Node.js 為平台，讓手機與 Node.js 上架構的 Web Server，以 socket io 方式連結，即快速又正確。



圖十七、架構圖

陸、專案進度甘特圖

	3月20日	3月27日	4月10日	4月17日	4月24日	5月1日	5月8日	5月15日	5月22日	5月29日	6月5日	6月12日
前置作業												
專案討論												
Android手機介面												
Arduino程式編寫												
手機與面板結合												
測試與調整												
專案完成												

柒、參考資料

[1]甘特圖 : <http://zh.wikipedia.org/zh-tw/%E7%94%98%E7%89%B9%E5%9B%BE>

[2]arduino : <http://arduino.tw/>

[3]arduino : <http://zh.wikipedia.org/zh-tw/Arduino>

[4]Android : <http://www.appinventor.tw/>