雲端智慧窗簾

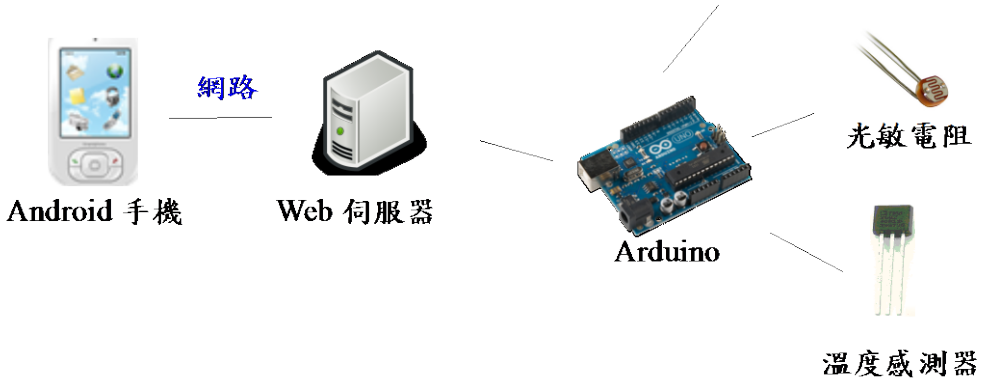
Android系統互動創作

Jun 5, 2013

系統架構

* 原本預計架構

* 實際實作架構



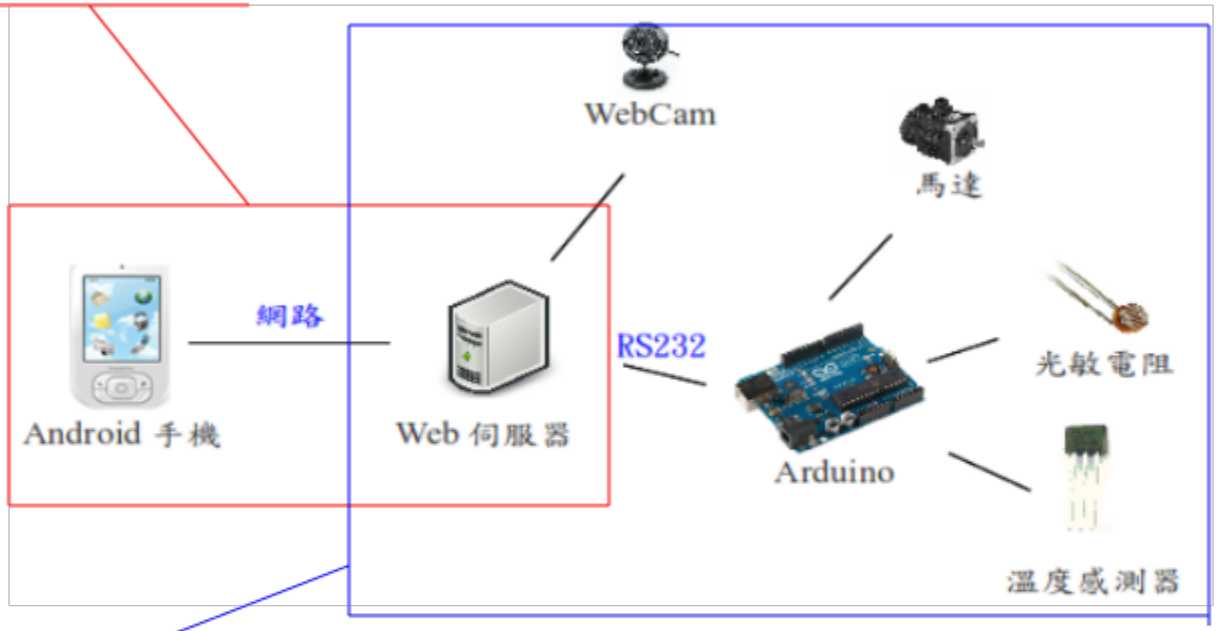
預期功能說明

* 感測器部分
* 窗邊照度太亮,且室溫超過27度C時，＝＞窗自動拉滿
* 窗邊照度太亮,且室溫超過27度C時，＝＞窗自動拉滿
* 窗邊照度太暗，客廳電燈自動開啟　（自動模式時）
* 有人按下關窗簾，窗簾全關
* 窗邊照度太暗，且室溫底於25度時，＝＞窗簾自動收回　（自動模式時）
* 窗簾拉出及收回期間超過10秒時斷電並警告。

* 手機端部分
* 可以接收到感測器傳來的數值

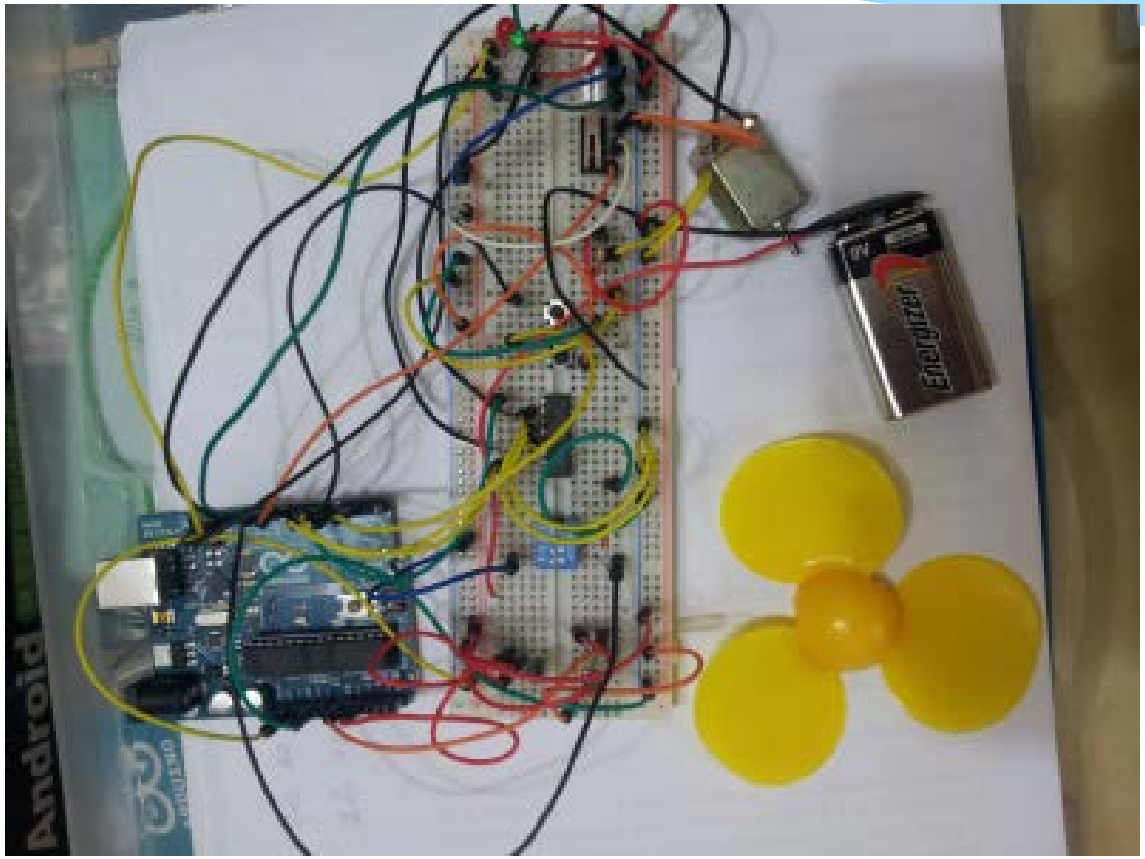
分工配置

政斌、辰希：手機端畫面配置、程式撰寫。。



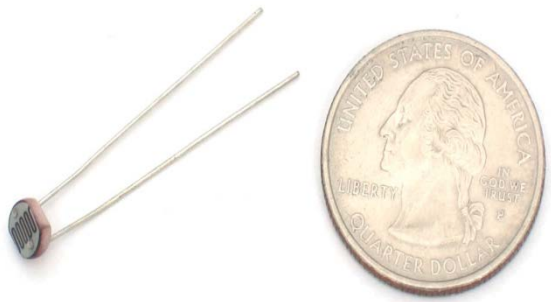
裕民：終端感測器、Web伺服器程式選寫。

終端感測器部分

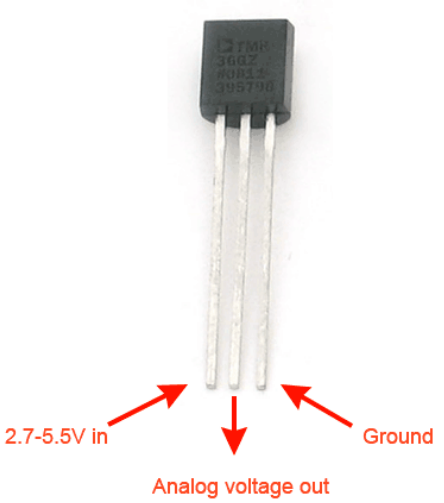


電子元件

* 光敏電阻



* 溫度感測器



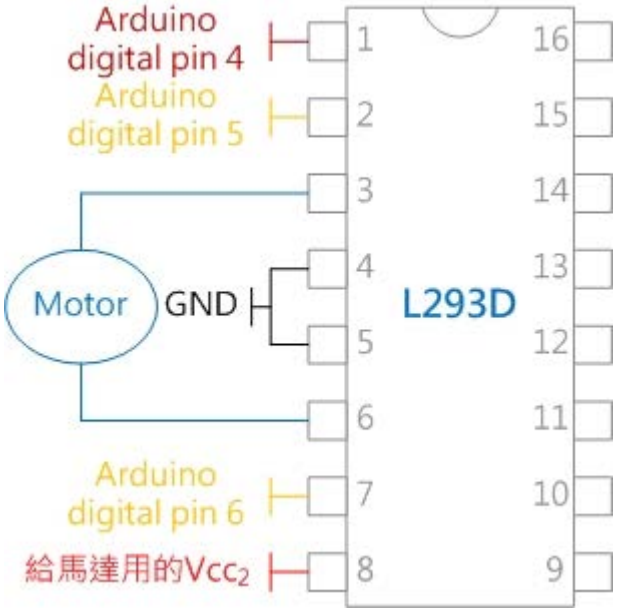
電子元件

* 微動開關

* 指播開關

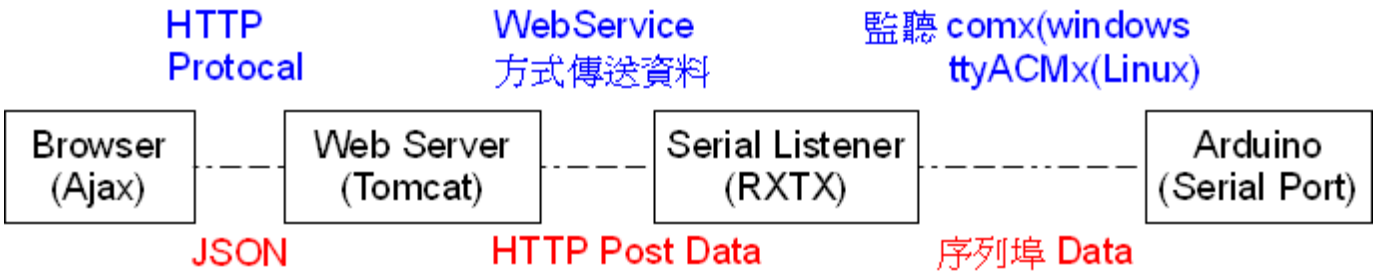
電子元件

* H-Birdge



* 直流馬達

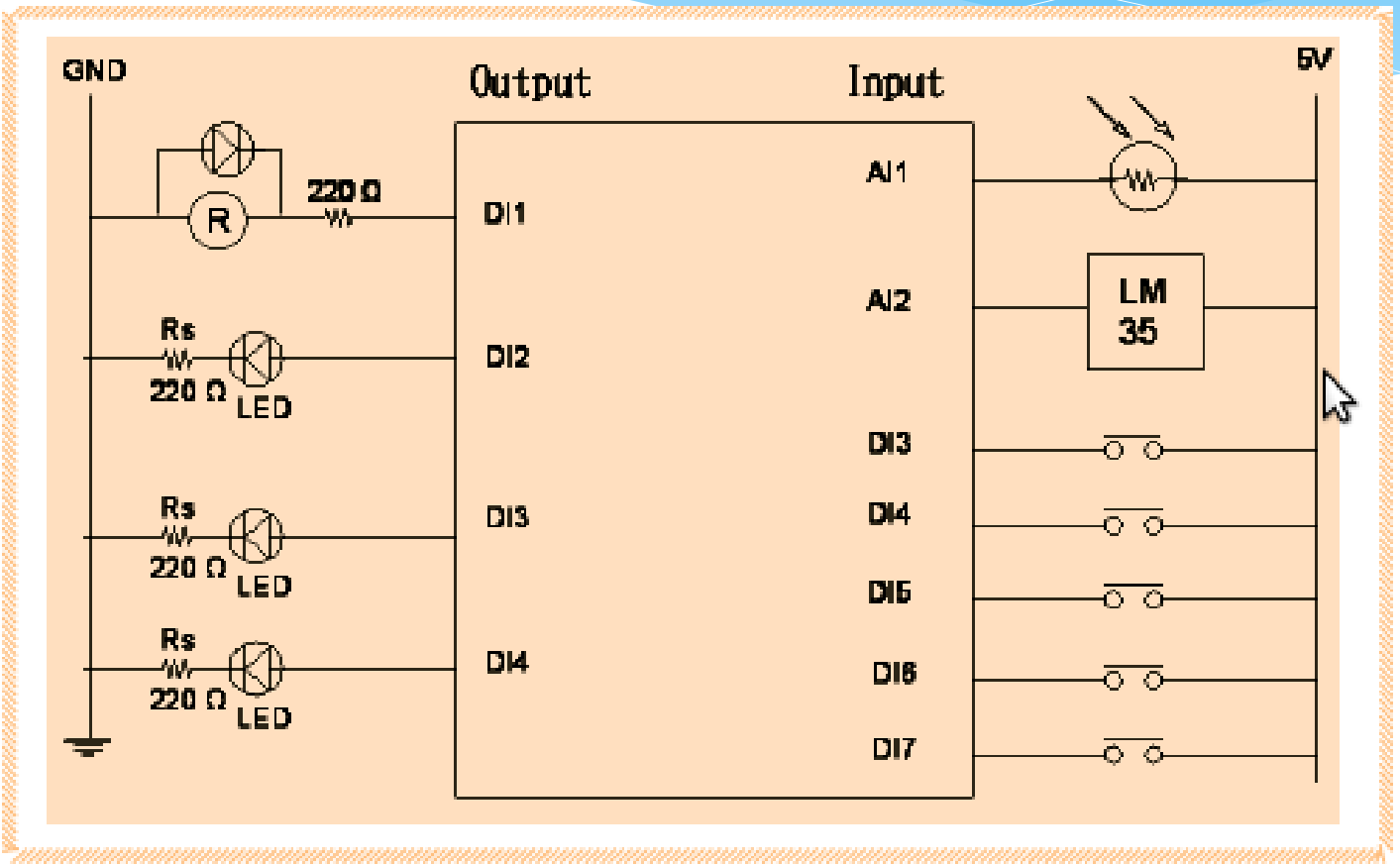
感測器軟體架構


HTTP
Protocal
WebService
方式傳送資料
監聽 comx(windows
ttyACMx(Linux)
Browser
(Ajax)
Web Server
(Tomcat)
Serial Listener
(RXTX)
Arduino
(Serial Port)
JSON
HTTP Post Data
序列埠 Data

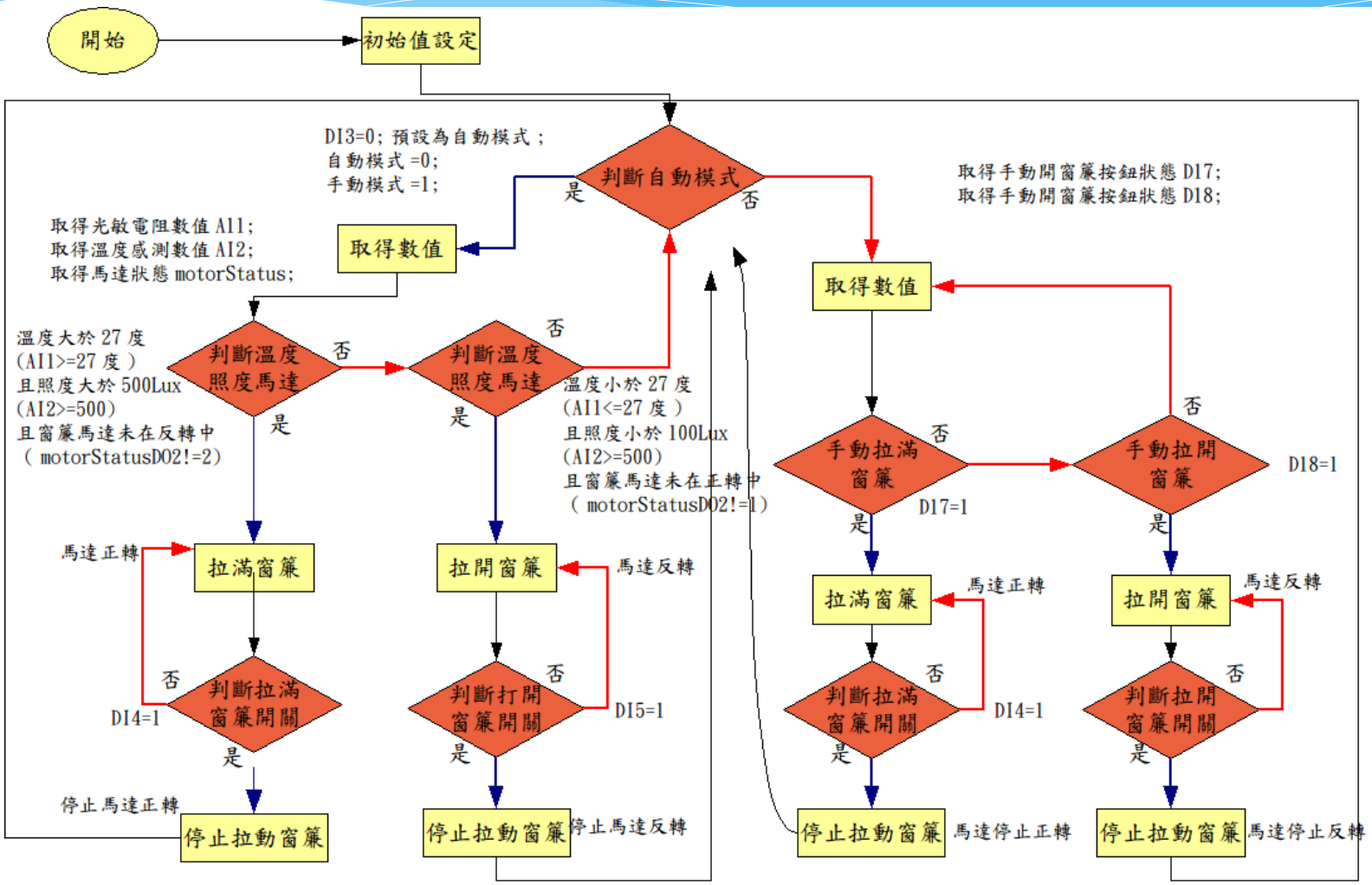
I/O配置表

Input		Output	
AI1	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)		

I/O配置表架構規劃


GND
Output
Input
5V
220 Ω
R
DI1
AI1
AI2
LM
35
Rs
220 Ω
LED
DI2
DI3
Rs
220 Ω
LED
DI3
DI4
DI5
Rs
220 Ω
LED
DI4
DI6
DI7

流程圖

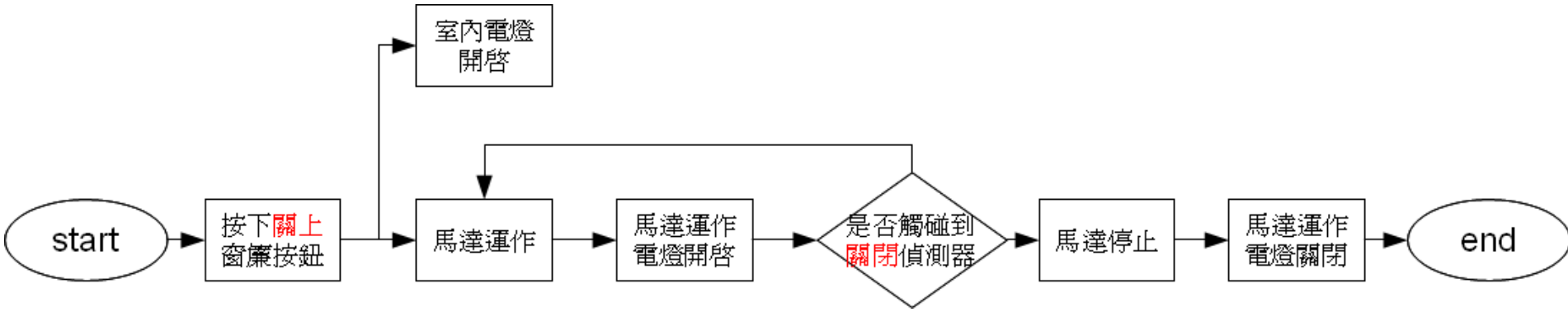

開始
初始值設定
DI3=0; 預設為自動模式；
自動模式 =0;
手動模式 =1;
判斷自動模式
是
否
取得光敏電阻數值 AI1;
取得溫度感測數值 AI2;
取得馬達狀態 motorStatus;
取得數值
溫度大於 27 度
(AI1>=27 度)
且照度大於 500Lux
(AI2>=500)
且窗簾馬達未在反轉中
(motorStatusDO2!=2)
判斷溫度
照度馬達
否
是
判斷溫度
照度馬達
否
是
溫度小於 27 度
(AI1<=27 度)
且照度小於 100Lux
(AI2>=500)
且窗簾馬達未在正轉中
(motorStatusDO2!=1)
馬達正轉
拉滿窗簾
判斷拉滿
窗簾開關
否
DI4=1
是
停止馬達正轉
停止拉動窗簾
拉開窗簾
馬達反轉
判斷打開
窗簾開關
否
DI5=1
是
停止拉動窗簾
停止馬達反轉
取得手動開窗簾按鈕狀態 DI7;
取得手動開窗簾按鈕狀態 DI8;
取得數值
手動拉滿
窗簾
否
DI7=1
是
手動拉開
窗簾
否
DI8=1
是
拉滿窗簾
馬達正轉
判斷拉滿
窗簾開關
否
DI4=1
是
停止拉動窗簾
馬達停止正轉
拉開窗簾
馬達反轉
判斷拉開
窗簾開關
否
是
停止拉動窗簾
馬達停止反轉

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

int autoAndManualSwitch = 10;           // DI3 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 = 9;          // DI4 窗簾開定位感測
int openSwitch = 0;
int closeSwitchPin = 8;         // DI5 窗簾關定位感測
int closeSwitch = 0;

int openBtnPin = 2;             // DI7 窗簾開按鈕
int openBtn = 0;
int closeBtnPin = 3;            // 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 = 11;              // DO1 電燈點滅
int motorLightPin = 12;         // DO2 馬達運作電燈
int faultLightPin = 13;         // DO3 故障指示燈
int faultWarm = 0;
```

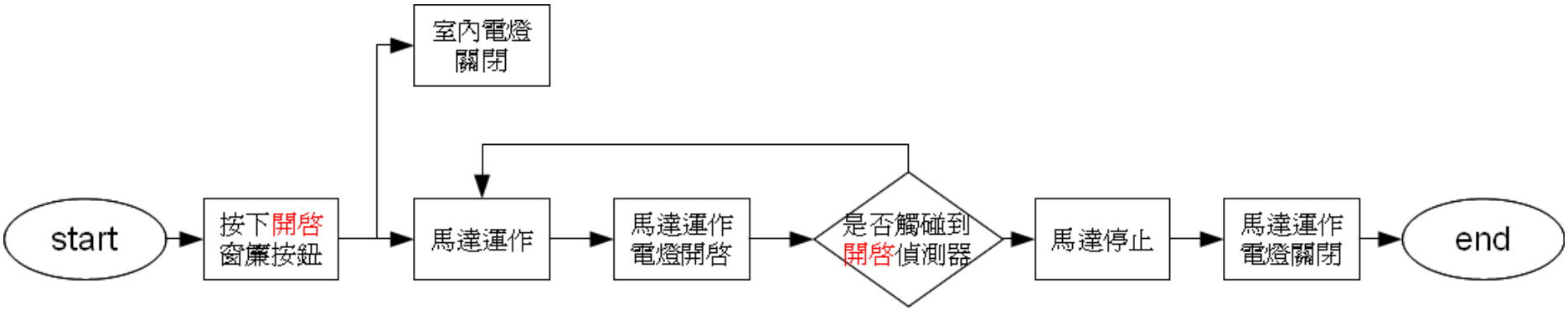
實際展示

* 手動模式-拉上窗簾



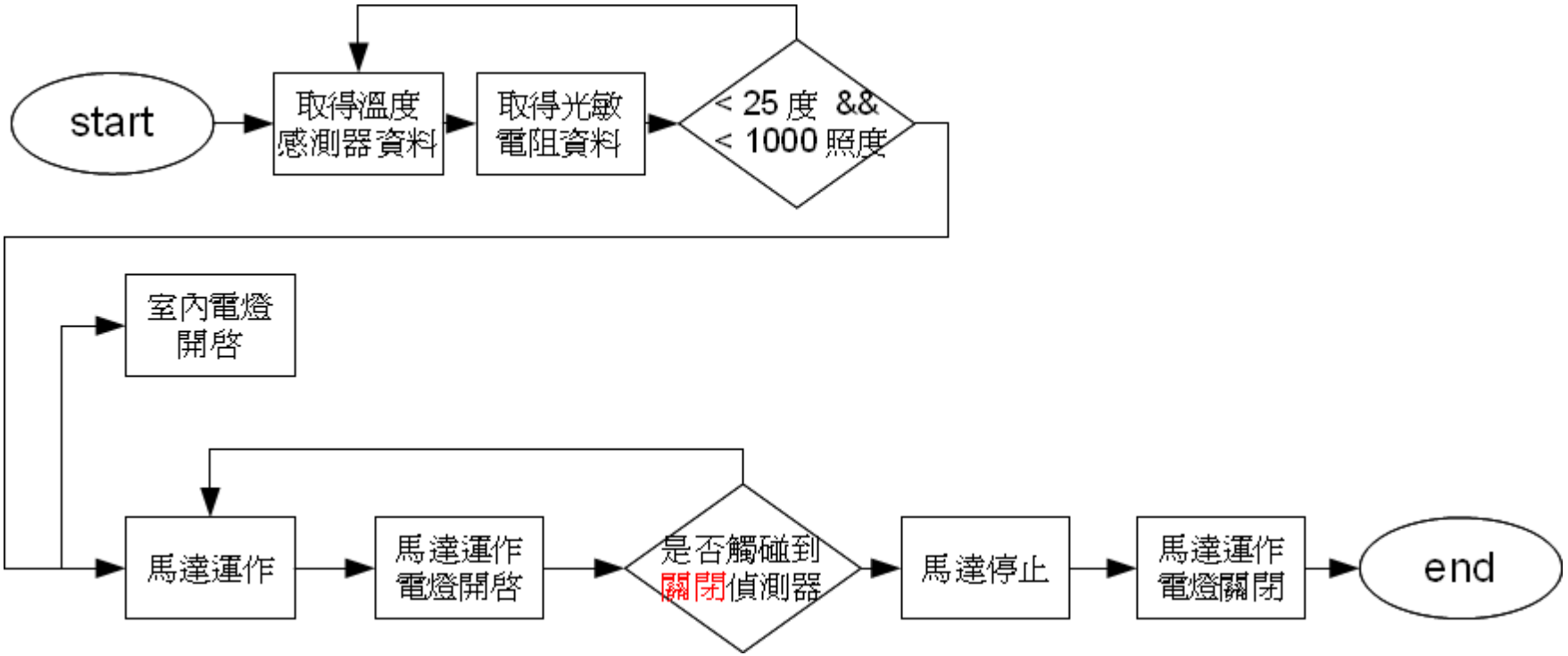
實際展示

* 手動模式-開啟窗簾



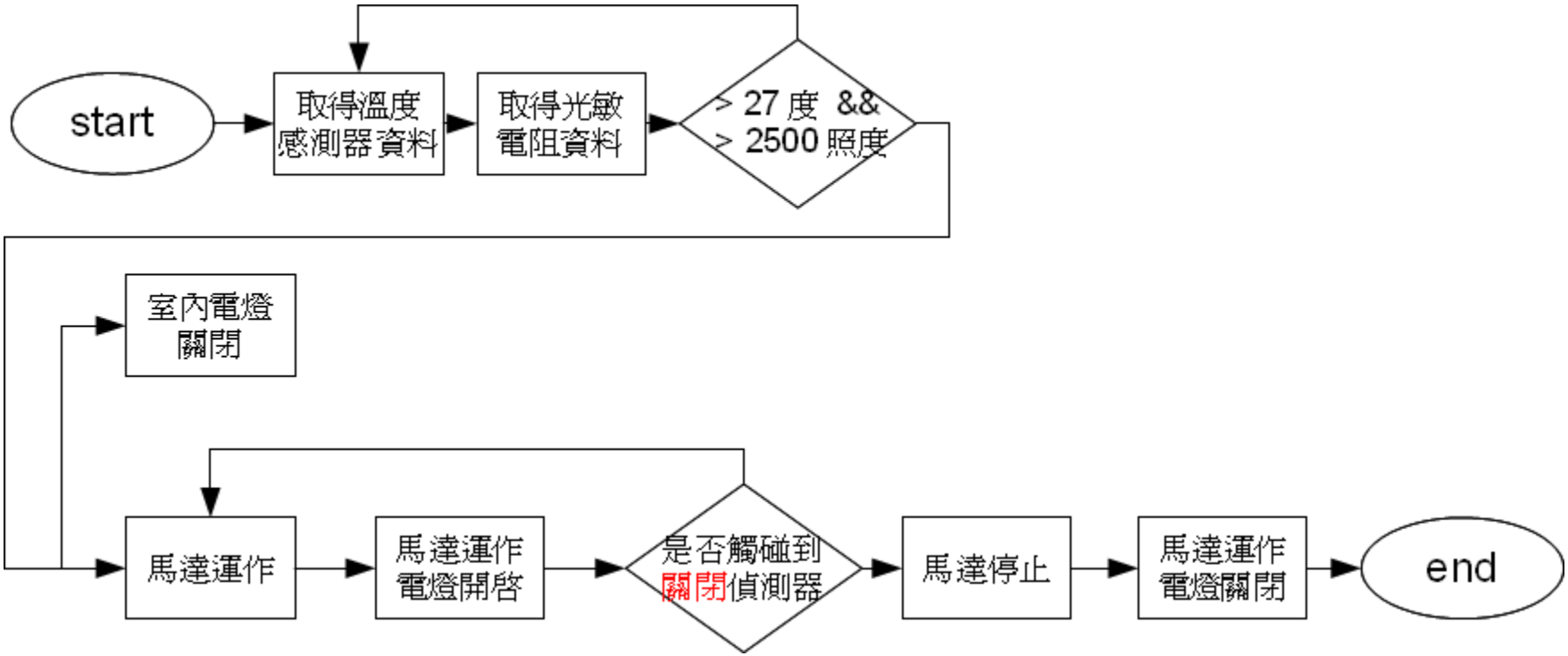
實際展示

* 自動模式-拉上窗簾



實際展示

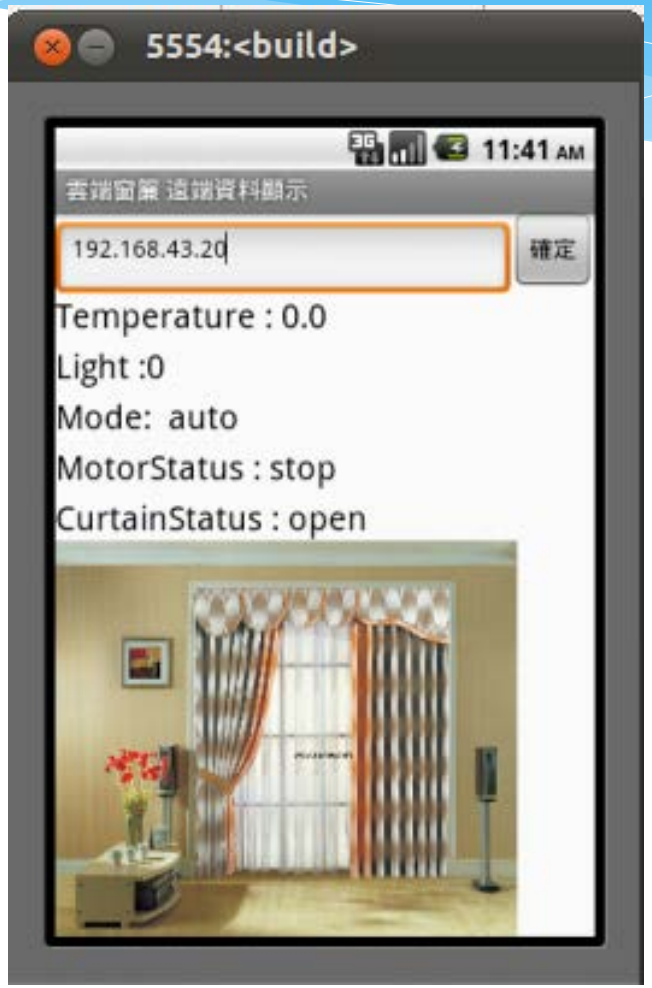
* 自動模式-開啟窗簾



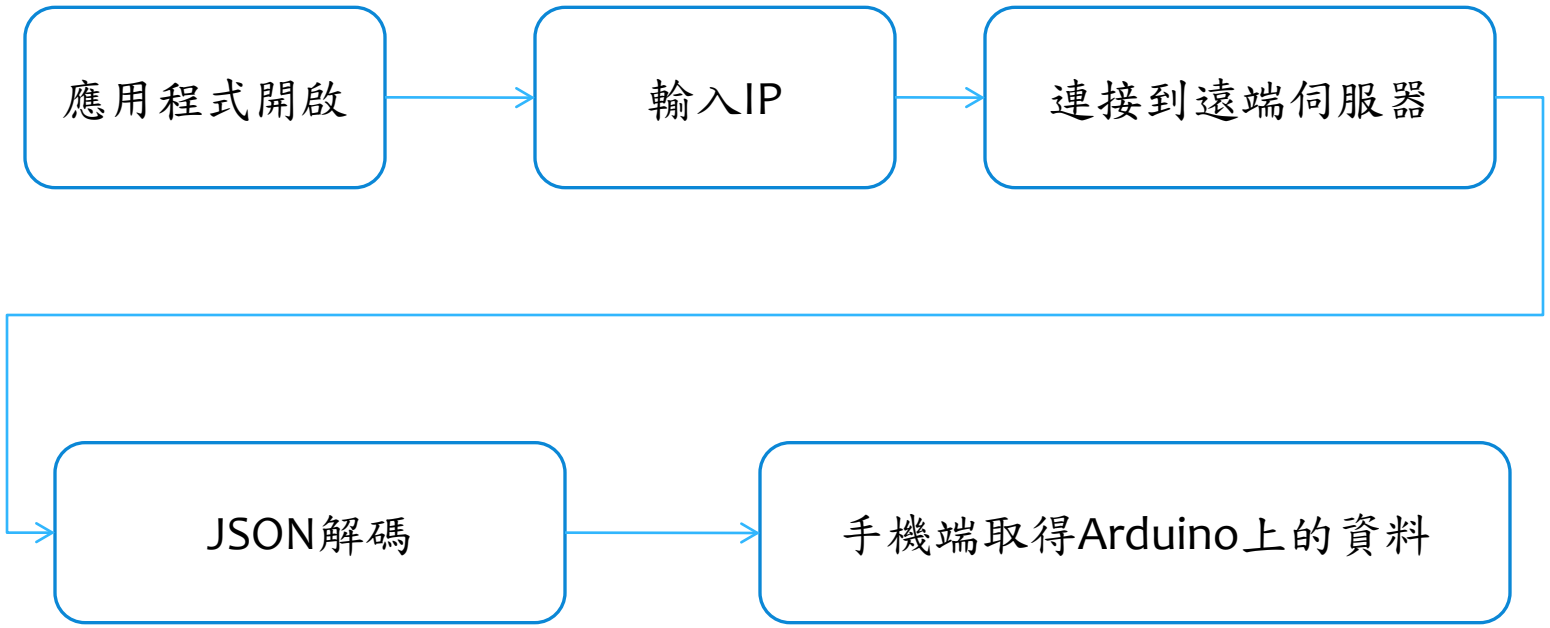
實際展示

* Web Server - JSON

手機部分


5554:<build>
11:41 AM
雲端窗簾 遠端資料顯示
192.168.43.20
確定
Temperature : 0.0
Light :0
Mode: auto
MotorStatus : stop
CurtainStatus : open

動作流程圖



What is JSON?

JSON(JavaScript Object Notation) 是個以純文字為基底去儲存和傳送簡單結構資料，你可以透過特定的格式去儲存任何資料(字串, 數字, 陣列, 物件)，也可以透過物件或陣列來傳送較複雜的資料。一旦建立了您的 JSON 資料，就可以非常簡單的跟其他程式溝通或交換資料，因為 JSON 就只是純文字個格式。

JSON 應用在哪些地方?

JSON 最常用用在 Web 網頁程式從 Server 端傳送資料給 browser，底下簡單舉個範例:

1. 伺服器端收到 ID，將產品資料（ex 價格, 描述）編碼成 JSON 資料，並且回傳給瀏覽器
2. JavaScript 收到 JSON 資料，將其解碼 (decode) 並且將資料顯示在網頁上

如何建立 JSON 字串

一. 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

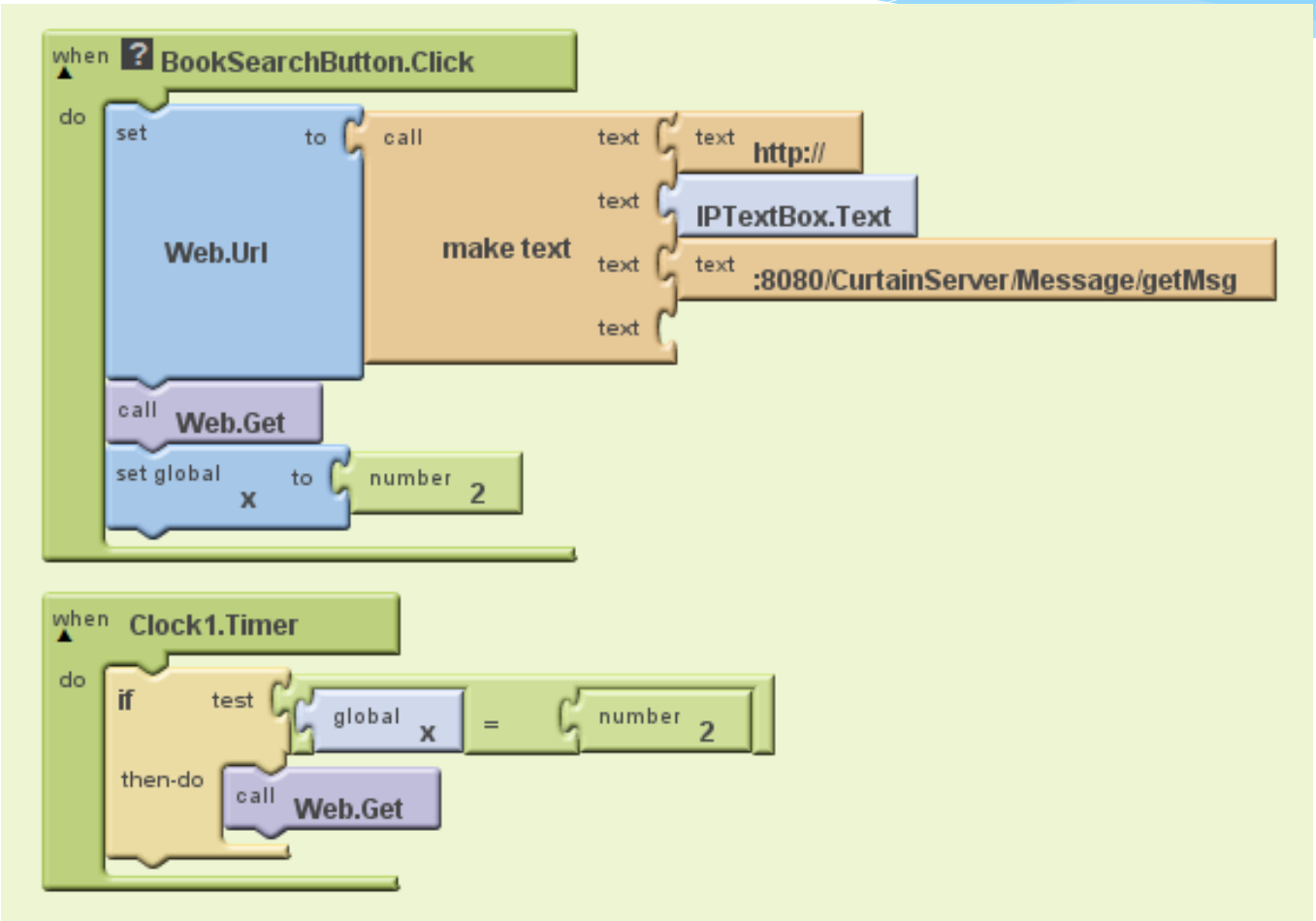
如何建立 JSON 字串

二. 物件或陣列的 value 值可以如下:

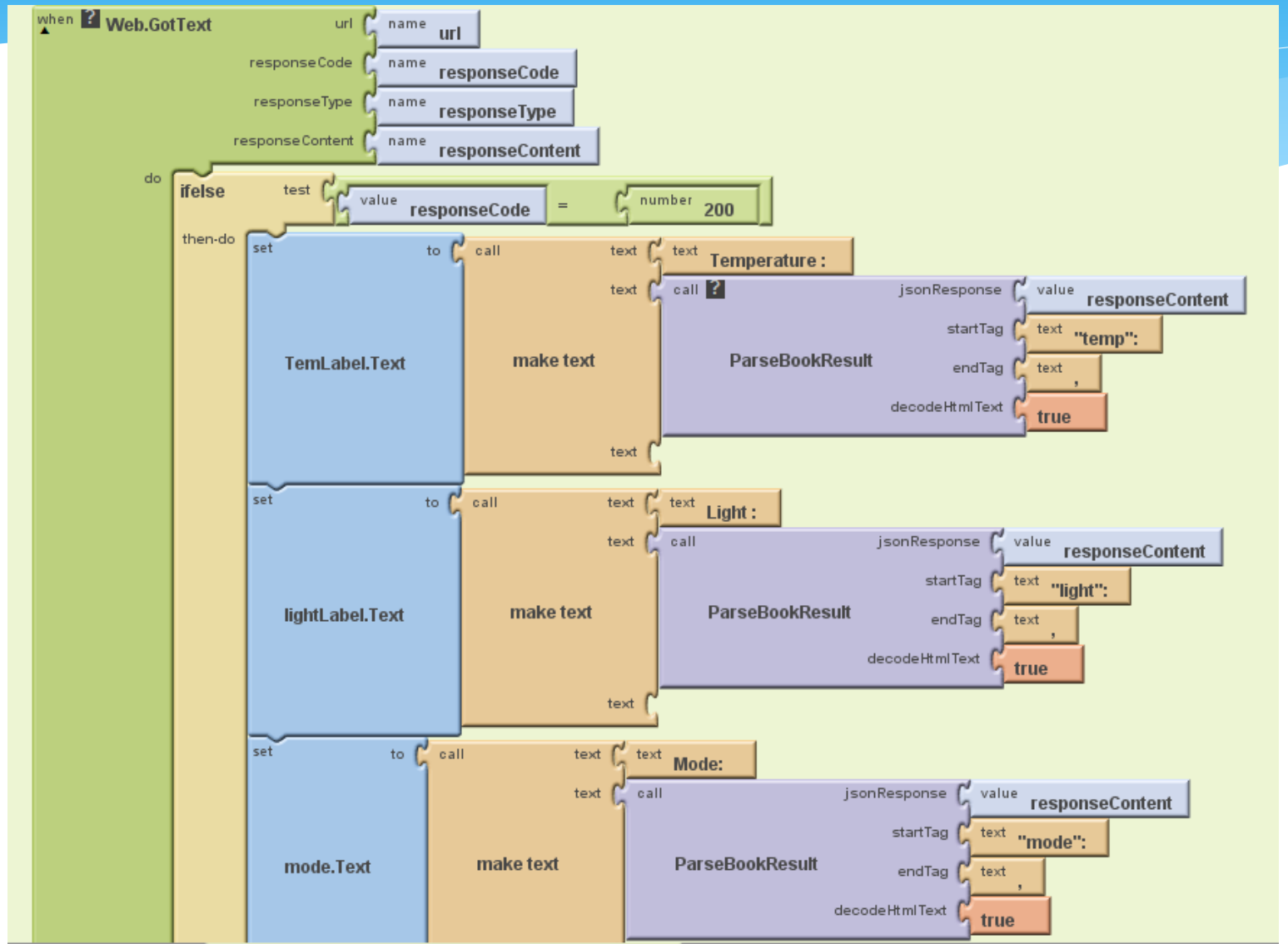
1. 數字 (整數或浮點數)
2. 字串 (請用 “” 括號)
3. 布林函數 (boolean) (true 或 false)
4. 陣列 (請用 [ ])
5. 物件 (請用 { })
6. NULL

程式碼圖

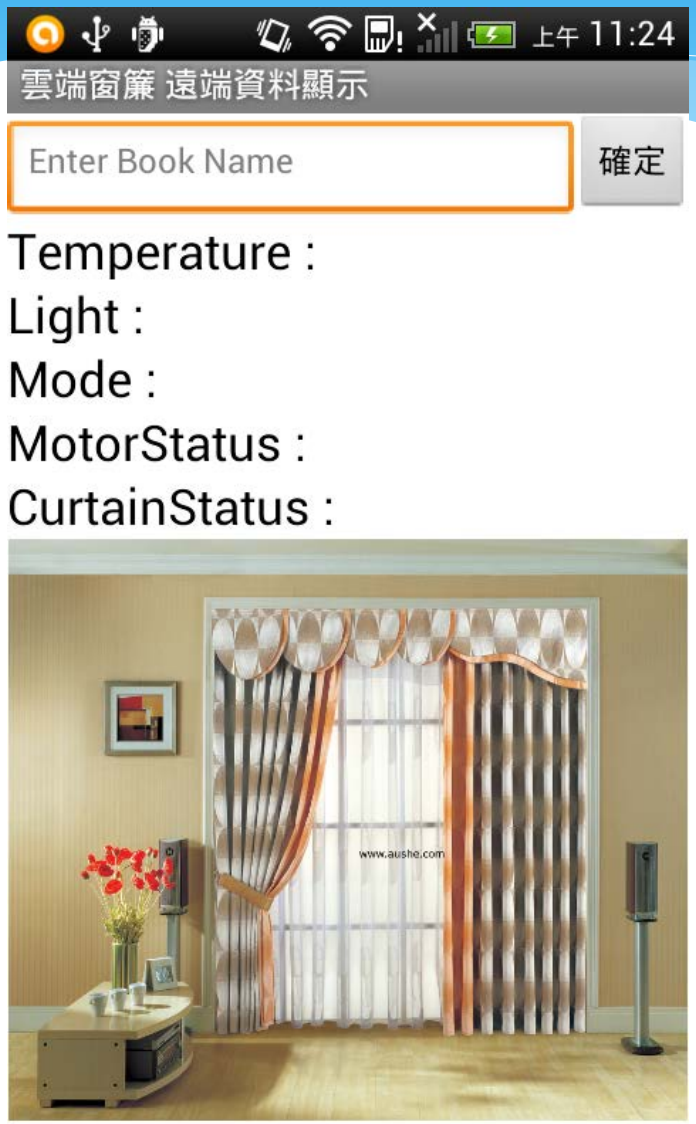
* 輸入IP部分

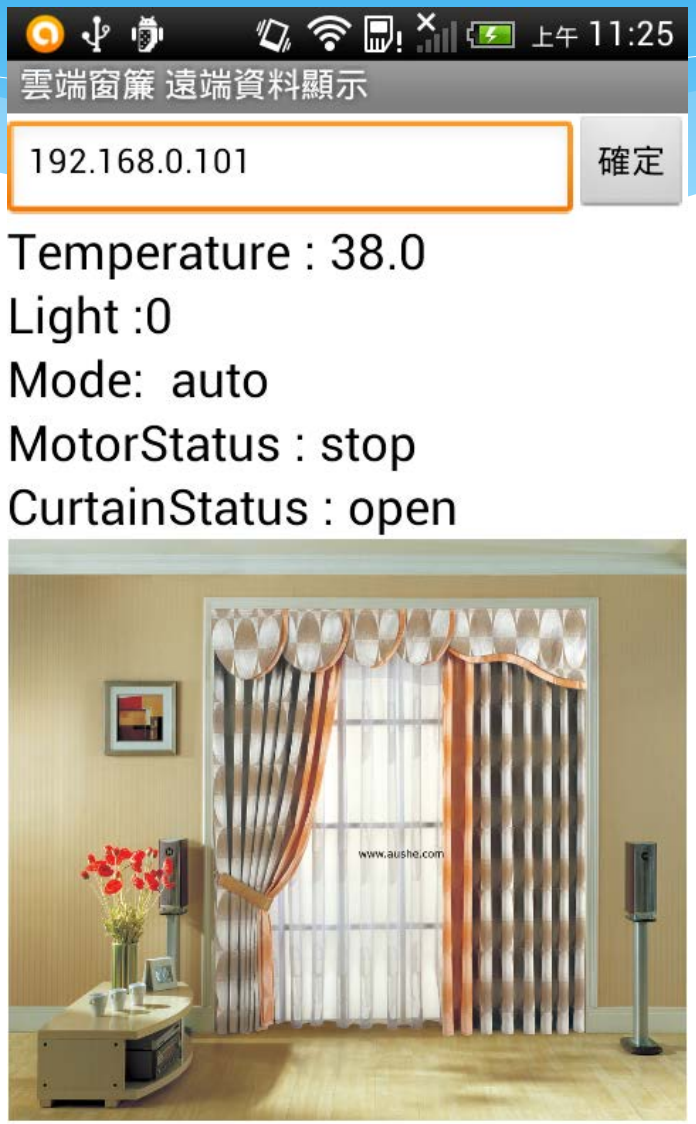

when BookSearchButton.Click
do
set Web.Url to
call make text
text http://
text IPTextBox.Text
text :8080/CurtainServer/Message/getMsg
text
call Web.Get
set global x to
number 2
when Clock1.Timer
do
if test
global x = number 2
then-do
call Web.Get

程式碼圖


when Web.GotText
url
responseCode
responseType
responseContent
name url
name responseCode
name responseType
name responseContent
do
ifelse
test
value responseCode = number 200
then-do
set TemLabel.Text to
call make text
text Temperature :
call ParseBookResult
jsonResponse value responseContent
startTag "temp":
endTag ,
decodeHtmlText true
set lightLabel.Text to
call make text
text Light :
call ParseBookResult
jsonResponse value responseContent
startTag "light":
endTag ,
decodeHtmlText true
set mode.Text to
call make text
text Mode:
call ParseBookResult
jsonResponse value responseContent
startTag "mode":
endTag ,
decodeHtmlText true

手機實體圖


上午 11:24
雲端窗簾 遠端資料顯示
Enter Book Name
確定
Temperature :
Light :
Mode :
MotorStatus :
CurtainStatus :

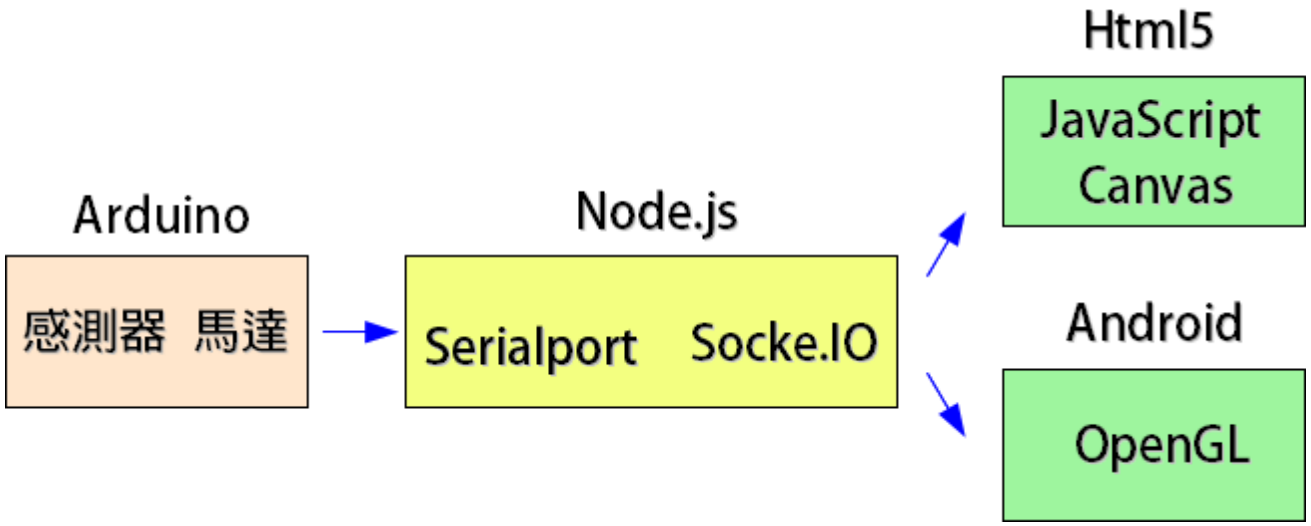

上午 11:25
雲端窗簾 遠端資料顯示
192.168.0.101
確定
Temperature : 38.0
Light :0
Mode: auto
MotorStatus : stop
CurtainStatus : open

未來擴充功能

* 設定偵測的溫度以及亮度。
* 加入中央氣象局目前的溫度以提供確認感測器的正確性。
* 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。
* 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。

未來擴充功能

* 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。



謝謝各位