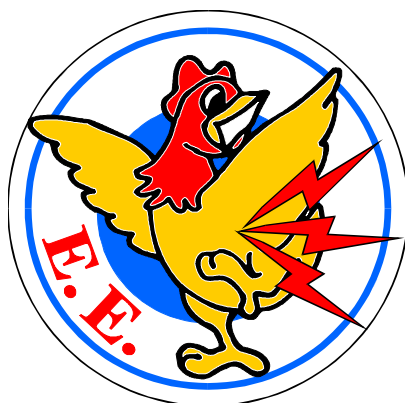
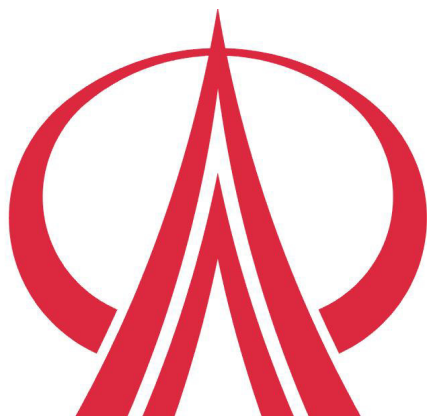


修平科技大學 電機工程系

DEPARTMENT OF ELECTRICAL ENGINEERING
HSIUPING UNIVERSITY OF SCIENCE AND TECHNOLOGY

實務專題報告書

太陽能製程之輸送系統模擬



指導老師：趙維和

專題製作學生：四技電四乙 黃永昌 BD100102

四技電四乙 王晨宇 BD100062

中華民國 一百零三 年 十二 月 二十三 日

摘要

對科技廠的機台運作及回報功能而言，如何使運作上的控管及錯誤偵測回報和產品生產狀況之呈現，要有良好的產能，相對的與此有著密切關係。

各家大廠的生產機台相異不大，那對於製作太陽能板的相關機具，就 Sorter 這方面來說，是製程方面的最後把關，機台所檢測出的數據優劣都一併紀錄於機台面板並會傳給相關主管的電腦作紀錄顯示，那當主管或相關製程部門人員，無電腦或筆電在身時，如何有效去監管及注意廠內產品的大小事。

此專題係以模擬太陽能製程「Sorter」這部分的輸送系統，若能將工廠的機台與普遍的智慧型手機加以運用，並與之結合，進而達到改善其除錯的模式來提升效率，相對提高廠內的產能，而現在的社會是智慧控管的時代，在無線網路的環境下，運用智慧型手機或平板隨時掌握運作狀態，期許對相關業界的幫助能有所助益。

致 謝

感謝 103 學年度科技部科學工業園區人才培育補助計畫 - 智慧綠能系統暨週邊模組之創新應用人才模組課程所提供的實習機會以及專題製作的相關資源補助與師資指導，讓我們的專題可以順利的完成。

也因此有幸到了太陽光電能源股份有限公司，這段業界實習期間，使我們有了不一樣的學習及體驗，非常感謝製程部的潘副理，不時在廠內給我們提出相關問題讓我們去思考，組長們的協助及關照使我們實習順利，其他的技術員更是不厭其煩的教導我們，其中有外籍的夥伴，溝通上更是以外語互通有無，多虧他們，語言能力方面提升了不少。

專題製作裡，有了趙維和老師的悉心指導，並指正及建議我們要改善的部分，歐崇仁老師的技術支援不在話下，不管是硬體或軟體都能適度切入，還有畢業後的學長，展現出卓越的能力從旁協助，讓我們收穫良多。

目 錄

摘要	1
致謝	2
目錄	3
圖目錄	5
表目錄	6
第一章 序論	7
1-1 研究動機.....	7
1-2 目的.....	7
1-3 研究架構.....	8
第二章 系統架構	9
2-1 自走車的應用	9
2-2 感測器.....	9
2-3 通訊方式.....	11
第三章 輸送系統架構介紹	13
3-1 Arduino Uno.....	13
3-2 驅動裝置.....	16
3-3 感測裝置.....	20

3-4 App Inventor	21
第四章 藍芽控制模組	23
4-1 零件介紹.....	23
4-2 運作原理.....	26
4-3 改善方法.....	28
第五章 程式碼	29
第六章 結論	33
零件表	34
參考文獻	35

圖目錄

圖 1.3 研究架構圖	8
圖 2.2 感測器組成示意圖	9
圖 2.3 電阻式感測器（光敏電阻）	10
圖 3.1 Arduino Uno 板	14
圖 3.2 Arduino Uno 基本腳位註解	14
圖 3.3 L298N 電機驅動板	16
圖 3.4 感測器及光源架構	20
圖 3.5 MIT App Inventor 網頁編輯介面	22
圖 4.1 HC-06 藍芽模組	23
圖 4.2 藍芽接線示意圖	24
圖 4.3 手機藍芽控制之程式碼	27
圖 4.4 手機 APP 藍芽控制介面	28

表 目 錄

表 3.2 馬達前進.....	18
表 3.3 馬達後退.....	18
表 3.4 馬達停止.....	19
表 4.1 HC-06 藍芽接線圖說明.....	25
表 4.2 零件表.....	34

第一章 序論

1-1. 研究動機

能在以生產 Solar Cell 為主的太陽光電這間公司實習，是一個難能可貴的經驗，那藉由在實習中所學之技術與所學的知識技能相互結合，是很重要的。期望使廠區的作業能更加便利，相關主管單位如:PIE（製程整合部）能及時掌握生產現況、產線上的 ENG（工程師）在機台 Alarm（警報）時，便利處理現況。

1-2. 目的

本專題以研究如何使輸送系統的運轉狀態結合手機 app 加以監控。再以 Arduino 和 MIT APP Inventor 做出智慧控管的介面及連接，使其達到以 Android 手機能控制此系統，讓感測器接收 LED 所照射在太陽能板上的光譜數據呈現在 PC，進而檢測其優劣。

1-3. 研究架構

本專題將太陽能製程之輸送系統主體完成後，依其運作模式，把各個所需之控制元件將其分析，從中不斷嘗試及改善，以達到起始的需求。

我們的專題研究架構表示如下圖 1.3 所示

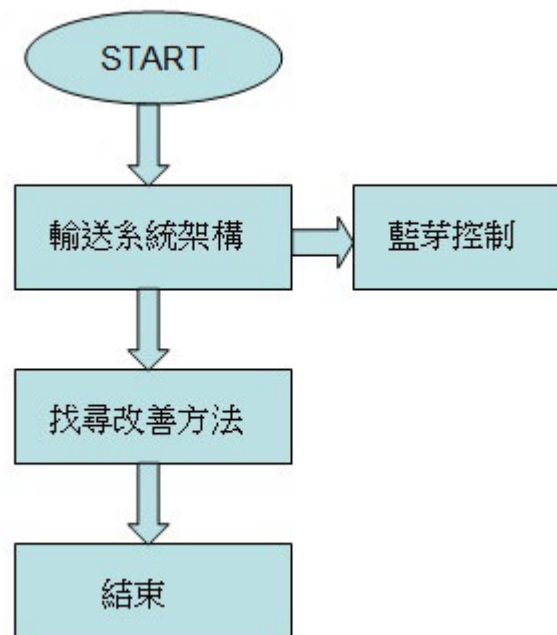


圖 1.3 研究架構圖

第二章 系統架構

2-1. 自走車的應用

科技的進步，人們的工作也漸漸得由機器代勞，轉成自動化的控制模式，自走車就是自動控制方面的一項，它是利用單晶片微電腦的的程式設計來控制馬達、驅動模組和感測器，並互相配合，製作出來的。

那我們以一台普通的自走車，將車體反轉，進而聯想到能夠將其長度加長，並套用上黏扣帶來當輸送的皮帶，讓車子的馬達、輪子那一部份，當作主要帶動此系統的輸出動力，讓平凡的自走車能夠以不同的方式來呈現我們所需的裝置樣貌。

2-2. 感測器

感測器是能接收規定的信號或刺激，並按一定規律將待測物理量或化學量轉換成可用輸出信號的器件或裝置，通常有敏感元件和轉換元件組成。（如圖2.2所示）

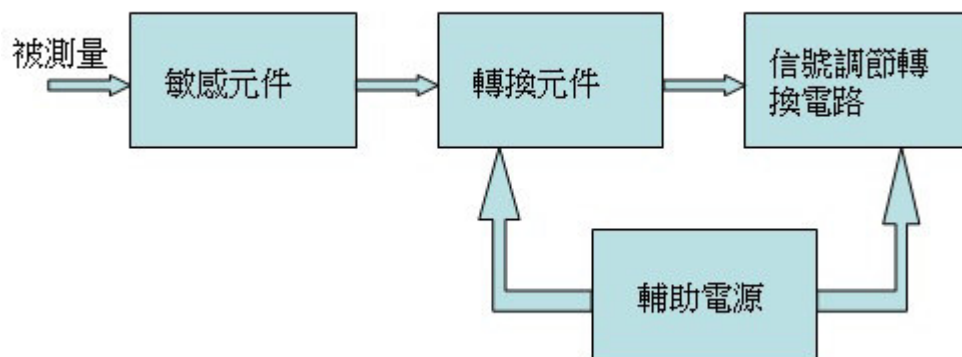


圖 2.2 感測器組成示意圖

目前常見的感測器有：

電阻式感測器(例：光敏電阻)、電容式感測器、熱電式感測器、電感式感測器、壓電式感測器、超音波、溫度(例：PT100)、濕度、氣體、壓力(例：荷重元感測器)…等

那我們此系統所用的感測器如圖 2.3 所示



圖 2.3 電阻式感測器 (光敏電阻)

2-3. 通訊方式

通訊是發送者透過某種媒體以某種形式來傳遞訊息到收信者以達到目的。從古至今，隨著科技的迅速發展，基本通訊完全利用有線或無線電完成，陸續出現了固定電話、無線電話、網際網路甚至視訊電話等方式。先進的技術拉近了人與人之間的距離，提高了通訊的效率，深深的改變了傳遞的方式。下面將探討幾個通訊技術：藍芽、WI-FI、近場通訊、ZigBee。

1. 藍芽技術是一種無線個人局域網（Wireless PAN），最初由易利信創製，後來由藍芽技術聯盟訂定技術標準。這個詞的來源是10世紀丹麥和挪威國王藍芽哈拉爾（丹麥語：Harald Blåtand Gormsen），借國王的綽號「Blåtand」當名稱，直接翻譯成中文為「藍牙」（blå=藍，tand=牙）。藍芽是一種通用的射頻頻率是2.40GHZ的無線電介面，它使帶有電子埠的裝置能夠在小範圍內進行無線通訊。
2. Wi-Fi是Wi-Fi聯盟製造商的商標可做為產品的品牌認證，是建立於IEEE 802.11 標準的無線區域網路設備。基於兩套系統的密切相關，也常有人把Wi-Fi當做IEEE 802.11 標準的同義術語。其用途包括：網路連結、校園的Wi-Fi覆蓋、城市的Wi-Fi覆蓋、電腦對電腦直接通訊。
3. 近場通訊（Near Field Communication，簡稱NFC），又稱近距離無線

通訊，是一種短距離的高頻無線通訊技術，允許電子設備之間進行非接觸式點對點資料傳輸，在十公分（3.9 英吋）內交換資料。

4. ZigBee（也稱紫蜂）是一種低速短距離傳輸的無線網路協定，底層是採用 IEEE802.15.4 標準規範的媒體存取層與實體層。主要特色有低速、低耗電、低成本、支援大量網路節點、支援多種網路拓撲、低複雜度、快速、可靠、安全。

第三章 輸送系統架構介紹

3-1. Arduino Uno

Arduino 的微處理器使用類似 Java 的 C 語言的開發環境，我們可在開發板上外接各類的電子元件及各種通訊模組等。

我們以Arduino Uno(如圖3.1)作為此輸送系統的邏輯運算單元，而Uno的處理器核心是ATmega328，同時具有14路數字輸入/輸出口(其中6路可作為PWM輸出)，一個USB口，6路模擬輸入，一個電源插座，一個16MHz晶體振盪器，一個ICSP header 及一個複位按鈕。

在接線部分，基本腳位(如圖3.2所示)的認知能夠方便我們對該腳位要對應的是哪個部分，該跟另一裝置如何做接線，清楚的做配合，接錯時也方便查詢錯誤點在哪。

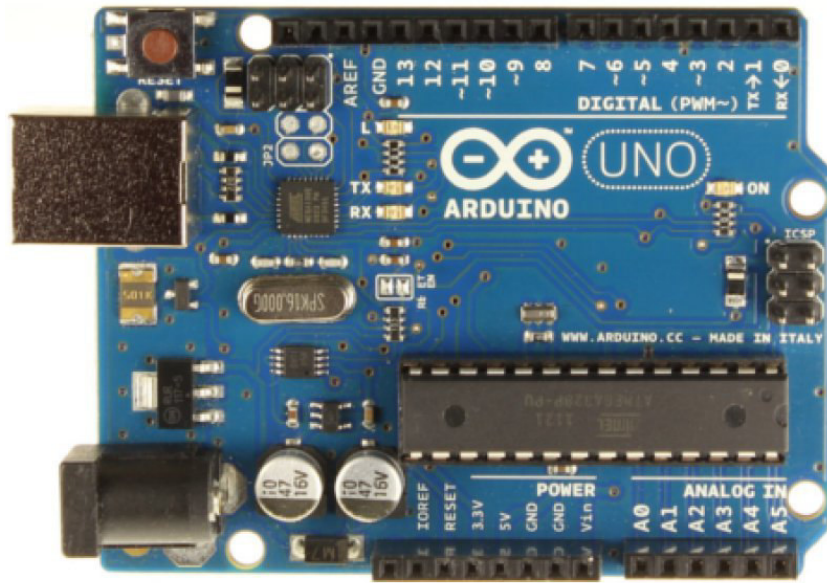


圖 3.1 Arduino Uno 板

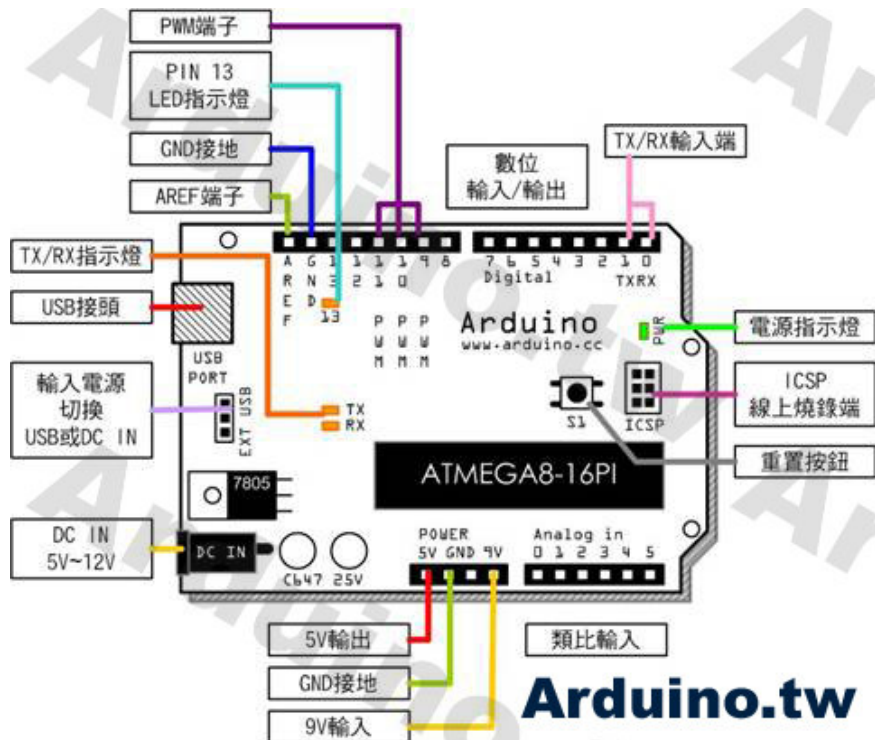


圖 3.2 Arduino Uno 基本腳位註解

特色描述

- (1)開放原始碼的電路圖設計，介面免費下載，可依需求自行修改。
- (2)可與各式各樣的電子元件連接。
- (3)使用高速的微處理控制器（ATMEGA328）
- (4)開發語言及環境都容易上手。

性能描述

- (1) Digital I/O 數位式輸入/輸出端共 1~13。
- (2) Analog I/O 類比式輸入/輸出端共 0~5。
- (3)輸入電壓:接上 USB 時無須外部供電或外部以 5~9V 直流電壓輸入
- (4)支持 ISP 下載功能

3-2. 驅動裝置

1. L298N電機驅動板

在驅動裝置這部分，我們以L298N電機驅動板（如圖3.3）來使用，做為驅動系統上的兩個直流馬達的裝置，它採用SMT工藝穩定性高、高品質鋁電解電容，使電路穩定工作。

而控制此區動裝置的訊號方面可以用Arduino或其它微控制器輸入。並提供了5 V 輸出介面（輸入最低只要6 V），可方便控制直流電機速度和方向並方便與Arduino板或其它微控制器作連接。

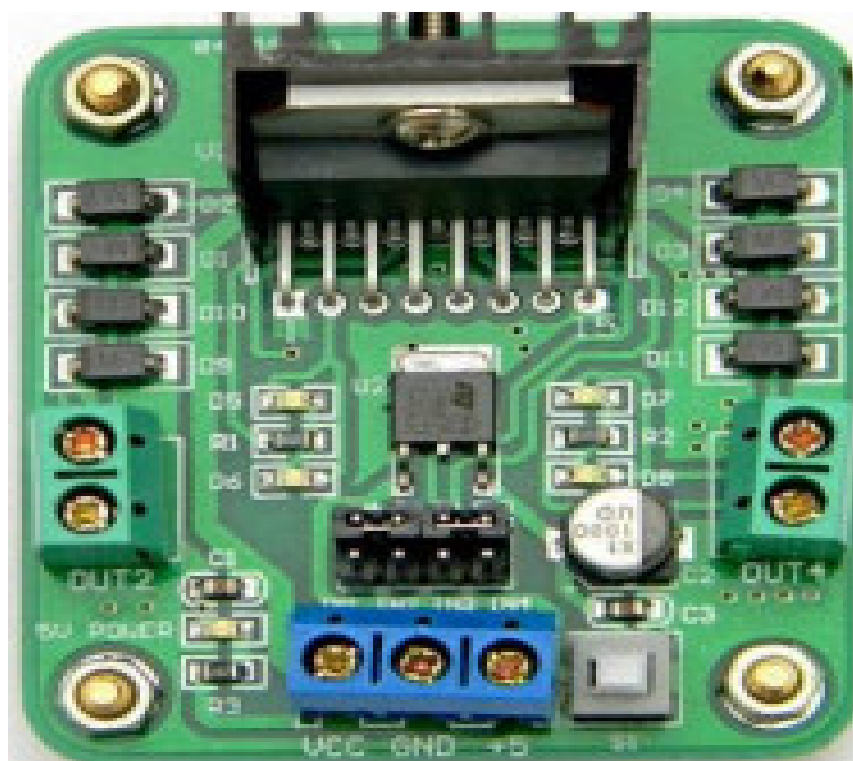


圖3.3 L298N電機驅動板

基本描述

- (1) 驅動晶片以 L298N 雙 H 橋直流電機驅動晶片。
- (2) 驅動部分端子的供電範圍 $V_s: +5V \sim +12V$ 。
- (3) 邏輯部分工作電流範圍 $0 \sim 36mA$ 。
- (4) 邏輯部分端子的供電範圍 $V_{ss}: +5V \sim +7V$ (可板內取電 $+5V$)

2. 馬達驅動

我們專題使用的是直流馬達，好處在為控速方面比較簡單，只須控制電壓大小即可控制轉速，但是直流馬達不適合於高溫或是易燃的工作環境。且一般直流馬達都是有碳刷做為電流變換器，所以直流馬達需要定期清理摩擦產生的髒污以利於馬達正常運作。

當我們發送數位訊號和類比訊號給左右馬達分別是LOW、160時，馬達就會正轉，除非有其他指令，否則馬達就會一直執行正轉。在Arduino的開發環境中，1 秒=1000 毫秒；0.1 秒=100 毫秒；0.01 秒=10 毫秒。

表3.2 馬達前進

<pre>void forward()//前進</pre>	<pre>digitalWrite(4, LOW); analogWrite(5, 160); analogWrite(6, 160); digitalWrite(7, LOW); digitalWrite(VR, HIGH);</pre>
-------------------------------	--------------------------------------------------------------------------------------------------------------------------

當發送數位訊號和類比訊號給左右馬達分別是 130、LOW 時，則馬達就會反轉。

表 3.3 馬達後退

<pre>void backward()//後退</pre>	<pre>analogWrite(4, 130); digitalWrite(5, LOW); digitalWrite(6, LOW); analogWrite(7, 130); digitalWrite(VR, HIGH);</pre>
--------------------------------	--------------------------------------------------------------------------------------------------------------------------

當發送數位訊號和類比訊號給左右馬達分別是LOW、LOW時，則馬達就會停止。

表3.4 馬達停止

void motorstop()//停止	digitalWrite(4, LOW); digitalWrite(5, LOW); digitalWrite(6, LOW); digitalWrite(7, LOW);
----------------------	--------------------------------------------------------------------------------------------------

3-3. 感測裝置

光學感測器是利用光敏元件將光訊號轉換為電訊號的感測器，感應波長在可見光波長附近，如紅外線波長和紫外線波長。光學感測器不只是應用於光的測量，常用於作為探測元件，組成其它類型的感測器，對非電量（如溫度等）進行檢測，只要將這些非電量轉換為光訊號的變化便可實現對非電量的檢測。（如圖3.4 為本專題感測器及光源的架構）

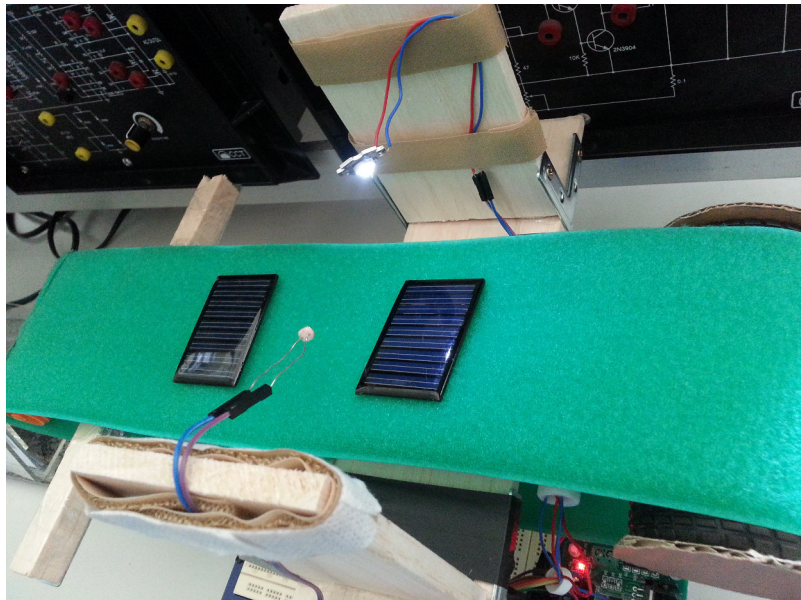


圖 3.4 感測器及光源的架構

3-4. App Inventor

這是一個完全線上開發的 Android 程式環境，拋棄複雜的程式碼而使用樂高積木式的堆疊法來完成需要的 Android 程式。而對於使用手機來控制的使用者來說，它不大需要太華麗的介面，只要使用基本元件例如按鈕、文字輸入輸出即可。

優點及適性

- (1)所有作業都在瀏覽器完成。
- (2)適合無 java 基礎或想學習手機程式設計的入門者。
- (3)伺服器會儲存您的工作進度並協助管理專案進度。
- (4)設定程式的行為就像玩樂高積木一樣有趣，增進學習方式。

我們所使用的 MIT App Inventor 的網頁編輯介面如圖 3.5 所示，左邊的區塊是使用者介面，要是想加什麼按鍵方塊或連線方塊，有想當多的物件供給選擇，而 Viewer 則是目前的配置情形，旁邊的 Components 則顯示目前有構成 Viewer 的相關物件訊息，都能點入對其進行文字、大小、位置的更改。

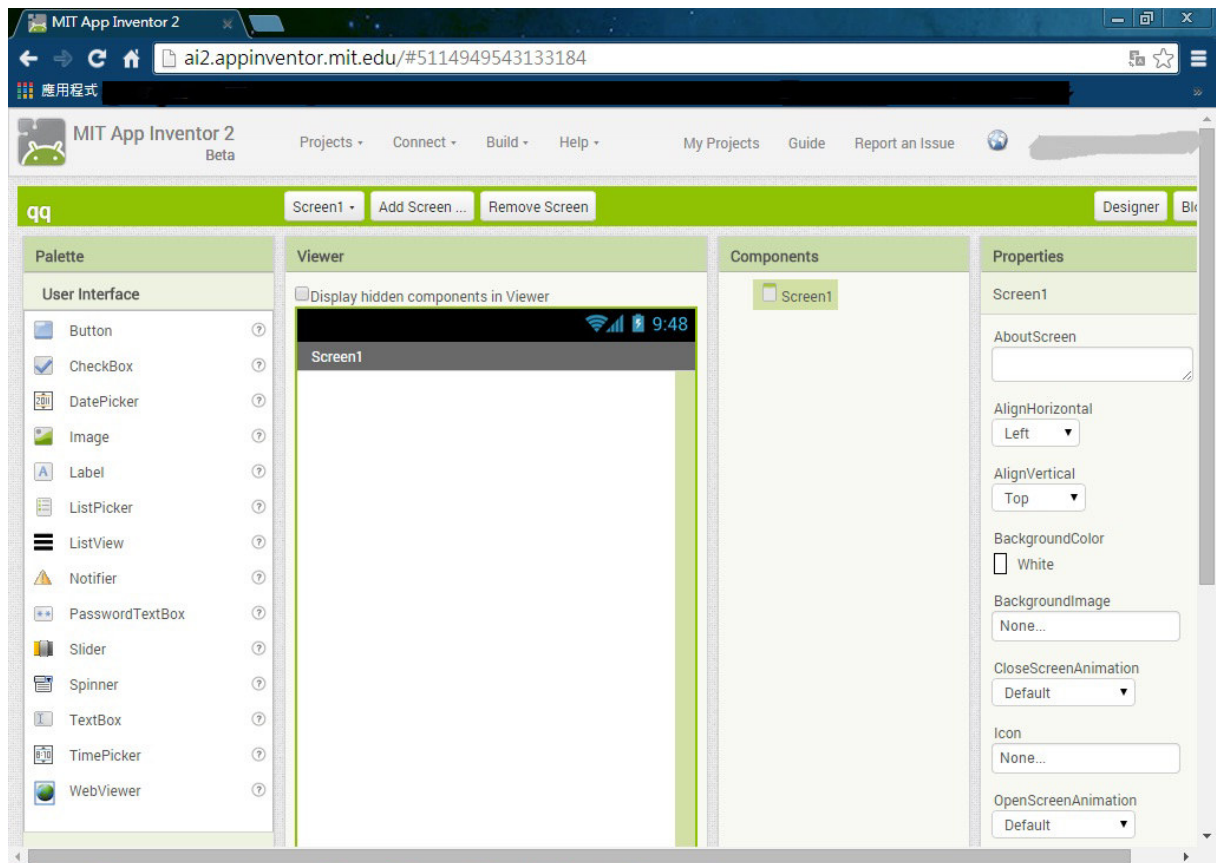


圖 3.5 MIT App Inventor 網頁編輯介面

第四章 藍芽控制模組

4-1. 零件介紹

藍芽操控分別由二個端點組成，主端（Server）與從端（Client），由主端負責控制從端，那我們是使用HC-06這一個藍芽模組（如圖4.1所示），屬於從端，其在偵測的有效距離是10公尺，最大距離可超過十公尺，電源電壓：3.6V~6V，未配對時電流約30mA，配對後約8mA，接口電平3.3V，可以直接連接各種單片機（51，AVR，PIC，ARM，MSP430 等），5V 單片機最好串聯一個1K 電阻再與模塊直接連接，採用CSR主流藍芽晶片，藍芽V2.0 協議標準。

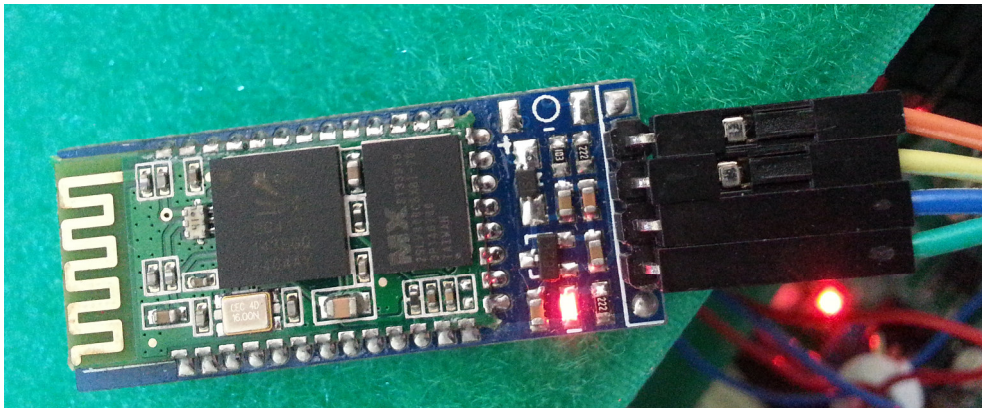


圖 4.1 HC-06 藍芽模組

圖 4.2 為此次的藍芽接線示意圖，要接的四條線分別為 VCC（正極）、GND（接地）、TXD（發送端）及 RXD（接收端）。我們將此 HC-06 接線圖的功能於表 4.1 上說明。

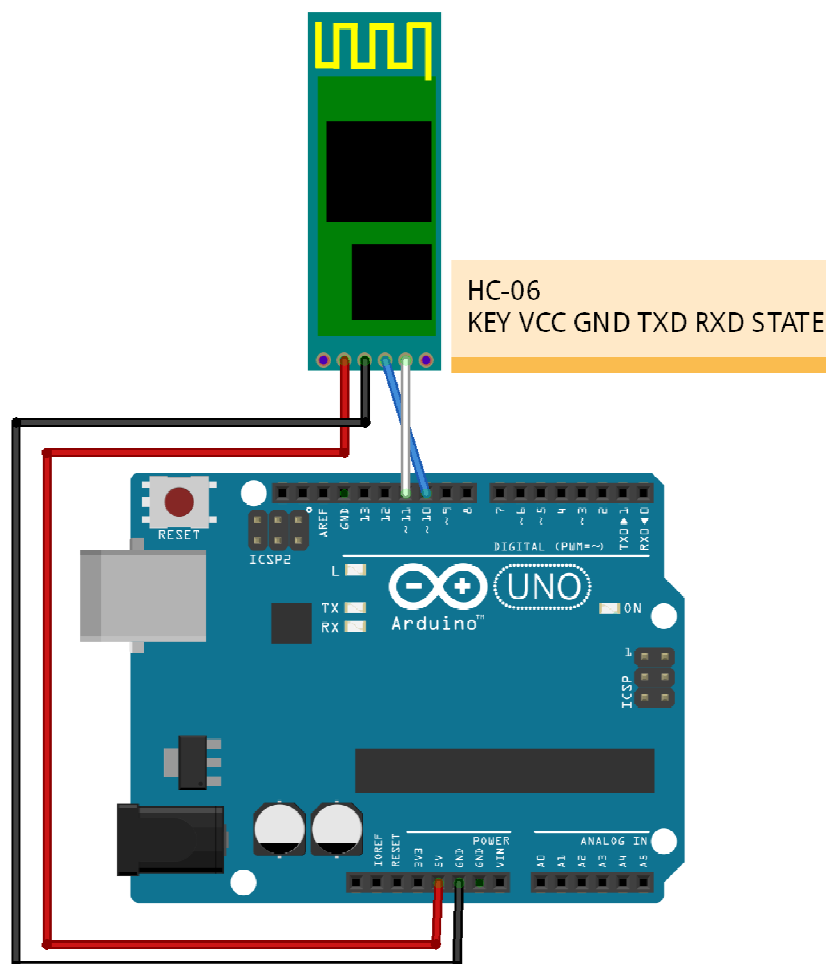


圖 4.2 藍芽接線示意圖

表 4.1 HC-06 藍芽接線圖說明

電流導通	功能
VCC (正極)	接電源
GND (負極)	接地
訊號傳輸	功能
TXD (發送端)	發送端若要傳輸必須與接收端做連結
RXD (接收端)	接收端若要傳輸必須與發送端做連結

4-2. 運作原理

將藍芽 HC-06 連接 Server 端，在這邊我們是以 Android 的作業系統之手機作為 Server 端，來發送信號。為什麼會選 Android 作業系統的手機來使用呢？因為能夠在前面所提到的 MIT App Inventor 的網頁來進行手機控制界面的相關編輯。

下圖 4.3 為我們手機控制太陽能製程之輸送系統的程式模塊，每個模塊所代表的位置都表示其應有的功能，當初在拼拼圖時，對各個方塊所要宣告的意義並不熟，多拼幾次，試到後來其實不難懂。

```
when forward .Click
do
  call BluetoothClient1 .SendText
  text "1"
  set Label1 .Text to "前進"

when backward .Click
do
  call BluetoothClient1 .SendText
  text "2"
  set Label1 .Text to "後退"

when motorstop .Click
do
  call BluetoothClient1 .SendText
  text "5"
  set Label1 .Text to "停止"

when ListPicker1 .AfterPicking
do
  set ListPicker1 . Selection to call BluetoothClient1 .Connect
  address ListPicker1 . Selection
  set Label1 .Text to "連接"

when ListPicker1 .BeforePicking
do
  set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames
```

0 0
Show Warnings

圖 4.3 手機藍芽控制之程式碼

4-3. 改善方法

起初做的手機藍芽控制的 APP 要跟太陽能製程輸送系統的藍芽模組作連接時，沒什麼問題，但按了指令，怎麼樣就是不給動，查了許多關於此問題的文章，也嘗試著修改內部的程式碼，還是不行，所以後來請教功力高身的學長，經他輕輕一點，在去用 MIT App Inventor 去重新審視，發現到與要控制的裝置腳位的配對沒設對，導致連上一直無法動作。所以圖 4.4 是成功過後的 APP 控制介面。



圖 4.4 手機 APP 藍芽控制介面

第五章 程式碼

```
#include <SoftwareSerial.h>

#define VR A0          //VR 信號輸入端

SoftwareSerial BTSerial(10, 11);    // RX | TX

int val;              //宣告 val 為整數

int c =0;             //宣告變數 c 為整數預設值為 0

void setup()         //副程式開始
{
    pinMode(VR,INPUT);    //初始化 VR 為輸入
    digitalWrite(VR, HIGH); //輸入 VR 狀態會跑
    pinMode(4, OUTPUT);   //初始化輸出 4
    digitalWrite(4, LOW); //輸出腳位 4 狀態,預設不動
    pinMode(5, OUTPUT);   //初始化輸出 5
    digitalWrite(5, LOW); //輸出腳位 5 狀態,預設不動
    pinMode(6, OUTPUT);   //初始化輸出 6
    digitalWrite(6, LOW); //輸出腳位 6 狀態,預設不動
    pinMode(7, OUTPUT);   //初始化輸出 7
    digitalWrite(7, LOW); //輸出腳位 7 狀態,預設不動

    Serial.println("Enter AT commands:");

    Serial.begin(9600);    //啟用串列埠
    BTSerial.begin(9600); //藍芽串列為 9600
}

void loop()           //此 loop 程序會一直重複執行
```

```

{
    val=analogRead(VR);    //讀取光敏電阻的分壓值
    Serial.print("VR value=");
    Serial.println(val);
    if(BTSerial.available()) {    //如果藍芽可用的
        c = BTSerial.read();    //c=藍芽讀取數值
        switch(c){
            case 49:            //案件 49
                Serial.print("forward\n");
                motorstop();    //馬達停止
                delay(1000);    //延遲 1 秒
                forward();    //前進
            break;            //休息
            case 50:            //案件 50
                motorstop();    //馬達停止
                delay(1000);    //延遲 1 秒
                Serial.print("backward\n");
                backward();    //後退
            break;            //休息
            case 53:            //案件 53
                Serial.print("motorstop\n");
                motorstop();    //馬達停止
            break;            //休息

```

```

    }
}
}
void motorstop(){           //停止
    digitalWrite(4, LOW);    //輸出 4 為 0
    digitalWrite(5, LOW);    //輸出 5 為 0
    digitalWrite(6, LOW);    //輸出 6 為 0
    digitalWrite(7, LOW);    //輸出 7 為 0
    digitalWrite(VR, HIGH);  //輸出 VR 為 1
}
void forward()              //前進
{
    digitalWrite(4, LOW);    //輸出 4 為 0
    analogWrite(5, 180);     //模擬類比輸出的 PWM 指令,5 為指定用來產生
PWM 輸出的接腳.
    //value 用來定義輸出的工作週期,可設定數值為 0~255.設定為 180 為正轉
    analogWrite(6, 180);     //模擬類比輸出的 PWM 指令,6 為指定用來產生
PWM 輸出的接腳.
    //value 用來定義輸出的工作週期,可設定數值為 0~255.設定為 180 為正轉
    digitalWrite(7, LOW);    //輸出 7 為 0
    digitalWrite(VR, HIGH);  //輸出 VR 為 1
}
void backward()             //後退

```



```
{  
    analogWrite(4, 130);    //模擬類比輸出的 PWM 指令,4 為指定用來產生  
PWM 輸出的接腳.  
    //value 用來定義輸出的工作週期,可設定數值為 0~255.設定為 130 為逆轉  
digitalWrite(5, LOW); //輸出 5 為 0  
digitalWrite(6, LOW); //輸出 6 為 0  
    analogWrite(7, 130);    //模擬類比輸出的 PWM 指令,7 為指定用來產生  
PWM 輸出的接腳.  
    //value 用來定義輸出的工作週期,可設定數值為 0~255.設定為 130 為逆轉  
digitalWrite(VR, HIGH);    //輸出 VR 為 1  
}
```

第六章 結論

起初就是因為科學工業園區人才培育計畫，才有這機會到科技廠去實習磨練，並把所學與之結合，當然，為了這專題，我們也多方去學習各種應用程式，從自我摸索到向專家請益，心態上有了不一樣的昇華，讓我們有多方面的提升。而將實務結合實習方面的經驗，擺脫既有的現狀，研擬出更利於社會的應用實務與技術層面。

當然這專題還有許多待加強的元素，並不是這樣就結束，因為有著智慧的控管，所以在傳輸連結上，必須考慮到環境及距離，我們以藍芽作連結，方便且距離較近，那要是相距較遠，那連結的媒介就要有所改變，換以 WIFI 連結的方式來克服這方面的問題，若能將相關數據進而傳輸給手機裝置來存取，可供相關人員隨時掌握進度，往後這都還有待研究並使其變的更加完善。

零件表

項目	名稱	數量	備註
1	Arduino Uno	1 塊	
2	藍芽模組	1 塊	
3	L298N 電機驅動板	1 塊	
4	自走車 DC 馬達 (含輪子)	2 顆	
5	光敏電阻	1 顆	
6	外角鐵	1 盒	固定用
7	內角鐵	2 盒	固定用
8	Tamiya 減速齒輪組	2 組	自行組裝、有 4 種變速型態
9	方型電池盒	1 個	3 號電池、4 入
10	杜邦線 公對母 40PIN	1 包	
11	白光 LED	1 顆	高亮度
12	DC 線 2.1 孔	1 條	保留接頭及部分線以利於連接電池盒 (注意正負)
13	飛機木	4 條	整體支架
14	黏扣帶	1 條	約 30 公分 (毛面)
15	太陽能電池	3 片	

表 4.2 零件表

參考文獻

1. APP Inventor TW 中文學習網

<http://www.appinventor.tw/whatis/>

2. 凱斯電子 L298N 馬達驅動模組

3. 元智大學機械系機械設計課程 Arduino 使用教材

作者:張凱維 推薦:徐業良

[http://designer.mech.yzu.edu.tw/articlesystem/article/compressedfile/\(2011-11-18\)%20Arduino%20%E7%B0%A1%E4%BB%8B%E8%88%87%E8%BB%9F%E9%AB%94%E5%AE%89%E8%A3%9D.pdf](http://designer.mech.yzu.edu.tw/articlesystem/article/compressedfile/(2011-11-18)%20Arduino%20%E7%B0%A1%E4%BB%8B%E8%88%87%E8%BB%9F%E9%AB%94%E5%AE%89%E8%A3%9D.pdf)

4. Arduino 微電腦控制實習 (OZONE 適用) — 邁向 AMA 中級先進微控制器應用認證

編著者:梅克 2 工作室

台科大圖書股份有限公司

5. 顏國雄老師 Arduino 部落格

<http://gsyan888.blogspot.tw/2014/03/arduino-hc-06-at-command.html>

6. 維基百科：<http://zh.wikipedia.org/wiki/Arduino>