

# 中華民國第 58 屆中小學科學展覽會 作品說明書

---

高級中等學校組 工程學科(一)科

052312

智慧綠能板

學校名稱：新北市立三重商工

作者：  職三 楊子賢  職三 林子豪  職三 翁健翔	指導老師：  林家宇
---	------------------

關鍵詞：太陽能、物聯網、樹莓派

## 壹 • 摘要

現在全球都在探討節能減碳，環保意識崛起，而其中的太陽能又是一項可視性最高的再生能源。由於太陽要直射太陽能板才會得到最大的效率，所以我們設計出了這一項只要啟動就能自行追日的裝置，這個裝置利用四個光敏電阻做偵測感知，並用不透光隔板將光敏電阻切成四個象限，用以偵測四個象限接收到的光源大小，以判別光源的位置，並使用兩顆步進馬達做雙軸全方位轉動，控制太陽能板，能夠隨時在最佳的位置發電，最後以 **lipo rider pro** 做輸出穩壓以供電池充電。

此外，由於本裝置還有搭配樹莓派做設計，所以能夠透過網路即時將輸出電壓、太陽能的發電電壓等資訊上傳回控制裝置讓操作者知道這些即時訊息，以便操作者能夠做後續之監控。

## 貳 • 研究動機

現在環保議題的興起，許多國家相繼開始研究綠色能源的產生方式，例如：風力發電、水力發電以及太陽能許多的發電方式以及後續的電動車發展。從此可見這議題是非常重大與我們息息相關，反之有什麼是渺小的我們能做到的呢？經過我們相繼討論後，決定了這項追日太陽能板，會有這樣想法的原因，是因為傳統太陽能板，都是設立在屋頂上並且是稍微斜面的，不過就有一個問題了，當陽光離開了該位置就無法達到最佳效能，所以我們設計的太陽能板是屬於，能夠感測並追隨光源移動的，找出能發揮最佳效果的位置，如此一來就解決了當陽光位置不同時所造成的效率較差之問題。

## 參 • 主題與課程之相關性或教學單元之說明

程式設計(條件結構、重複結構、陣列)、電子學(電晶體、光敏電阻、RC充放電)  
動力機械概論(能量的來源)

我們在上動力機械概論這門課時，聽到老師講述現在太陽能板的架設大多都是先經過計算出太陽一年四季的軌跡，再找其中間值來架設，這種方法雖然可以提高太陽能板在一年當中的發電量，但卻不能讓太陽能板隨時在最佳的角度上發電。

我們經過討論最後決定**透過光敏電阻來找尋光源**，並利用步進馬達來控制太陽能板轉到適當的位置，隨時使太陽能板發揮最大的發電效率，至於如何控制馬達轉動？我們在老師的建議下**使用了 Raspberry pi 來控制馬達轉動**，因為樹莓派須使用 python 的程式語言來編寫，幸好我們之前有學過 Visual Basic 這套程式語言，才不用讓我們花太多時間重新慢慢學起。

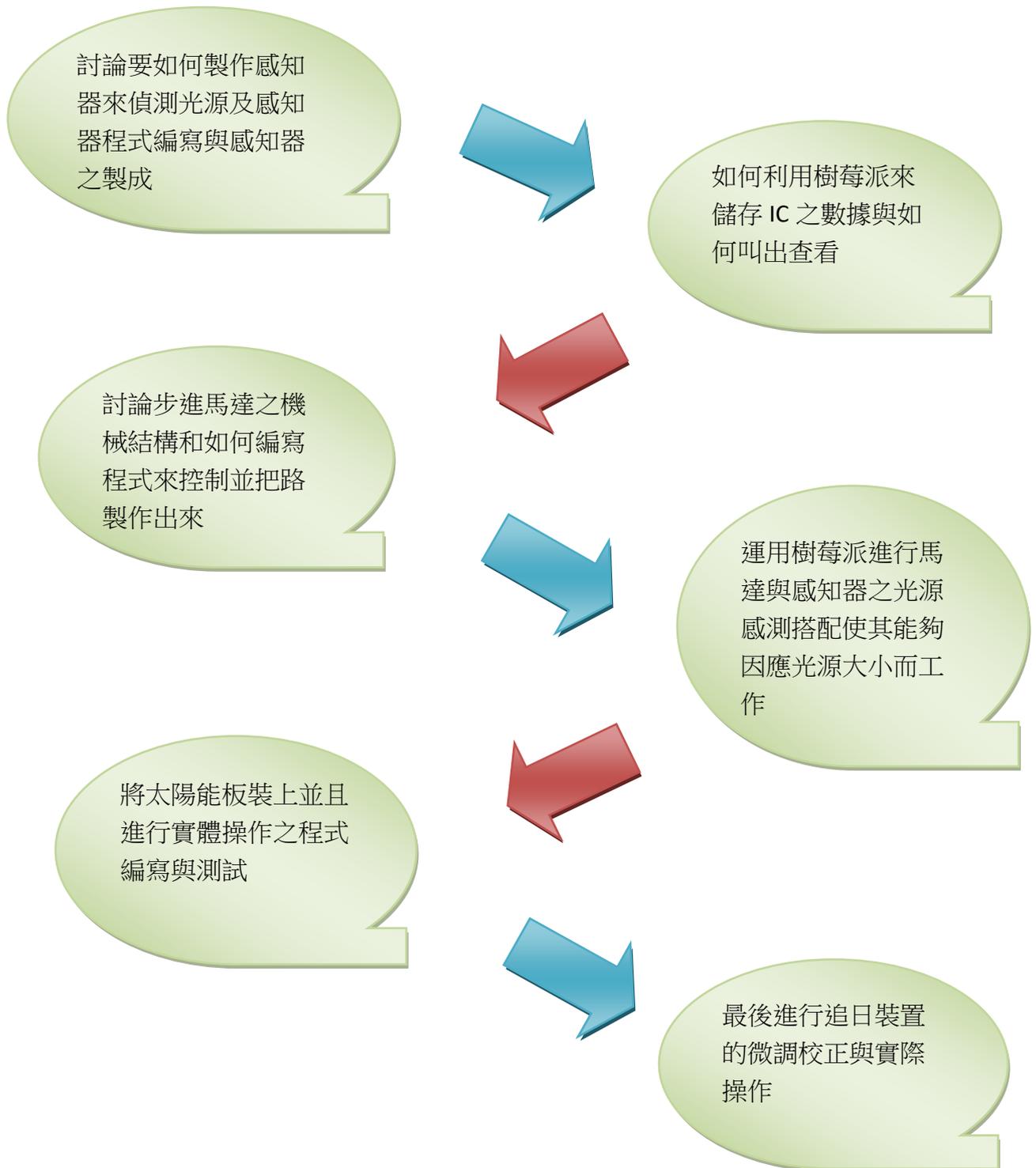
## 肆 • 研究過程

### 一、資料查詢:

Python 的基本語法學習:

我們每個人都從網路上找尋資料來練習，因為對我們來講程式語言是一個新的課程，我們在學校也只有打過其他的而已，但是我們要使用的這一種我們並沒有使用過，所以我們找了 Python 條件語句這一網站當作我們的課本學習 Python 的程式語言。

## 二、研究的設計流程



### 三、感知器的設計及校正

(一)我們一開始先討論要用甚麼當感知器的元件，在大家上網找尋各種資料之後，我們就把找來的資料做整理，並且在詢問老師之後，我們決定使用光敏電阻來製作我們的感知器，以及此感知器要做成哪種形式的，後來老師建議我們就以**數學的四象限**去做，而我們也有在找資料時看過，有人是以這種形式製成。



討論過程 (圖.1)

(二)接著我們在買完元件後，先是量測了電阻值的大小，並從中找出 4 個電阻值誤差在 10 歐姆以內的電阻，然後開始把電阻固定在一塊硬質的板面上，並用有一定硬度的紙將其分割成四個象限(建議用消光黑之顏色避免光反射影響測量值)，最後再將其接腳焊上長度足夠的銀線(避免操作時線會拉扯)。



光敏電阻的量測與成品圖 (圖.2)

(三)再來是程式的編寫，這邊運用到了樹莓派來做我們的處理器，在我們從網路上學習程式語言之後，老師找了一套範例叫我們去做改編，讓其能夠配合我們感知器去做執行的程式，接著我們在程式完成後開始校正，讓其可以準確的判斷光源位置。

```
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
GPIO.output(c_pin, True)
tc = time.time()
while not GPIO.input(d_pin):
    pass
td = time.time()
return (td - tc) * 1000000

# =====
#樹莓並沒有類比腳位，利用時間差將光的強弱數字位
#-----CDS-1-----
def analog_read1():
    discharge1()
    return charge_time1()

def read_resistance1():
    n = 20
    total1 = 0;
    for i in range(1, n):
        total1 = total1 + analog_read1()
    reading1 = total1 / float(n)
    resistance1 = reading1 * 6.05 - 939
    return resistance1

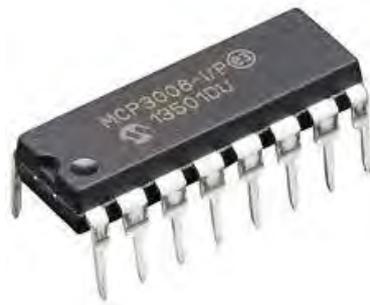
def light_from_r1(R1):
    return math.log(1000000.0/R1) * 10.0
#-----CDS-2-----
```



光敏電阻的校正 (圖.3)

#### 四、IC MCP3008 數據讀取及校正

(一)這次我們會使用這塊 IC 是因為它能夠將類比轉成數位，彌補我們樹莓派無法直接使用類比訊號的缺點，而它在讀取 ADC 數值時，會傳回一個 10-bit 的數字(介於 0 到 1023 之間)，輸入至樹莓派，由此可以根據輸入電壓與傳回值的電壓，計算出類比訊號的值。

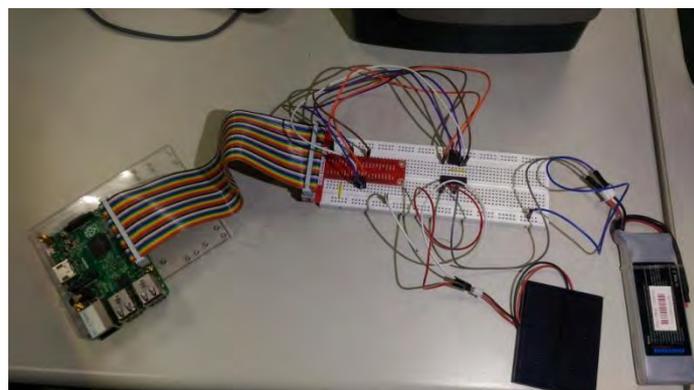


IC MCP3008 (圖.4)

(二)接著我們把 IC 搭配上樹莓派去配置線路，由 IC 提供電源(樹莓派給 IC 5V)與搭鐵給電池及太陽能板，再由 IC 接收回來的訊號轉變成數位給樹莓派，由樹莓派處理及儲存數據讓使用者可以隨時觀看。



樹莓派所記錄的 IC 數據(圖.5)



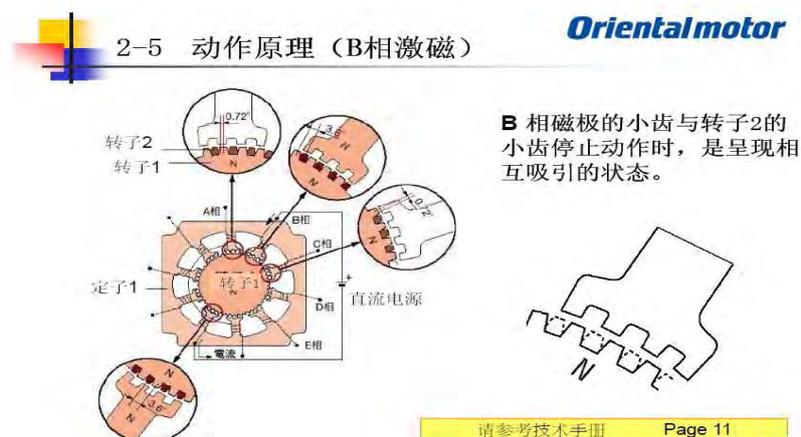
IC 與樹莓派搭配之線路 (圖.6)

## 五、步進馬達正轉逆轉程式編寫及測試

(一)我們接著討論到如何使太陽能板進行所謂的追日移動，在找尋了許多資料後，我們決定使用步進馬達，由於步進馬達可以達到 **360 度** 的轉向，所以我們使用到兩顆來做全方位的轉向，來使我們的追日裝置可以最無死角的追蹤。

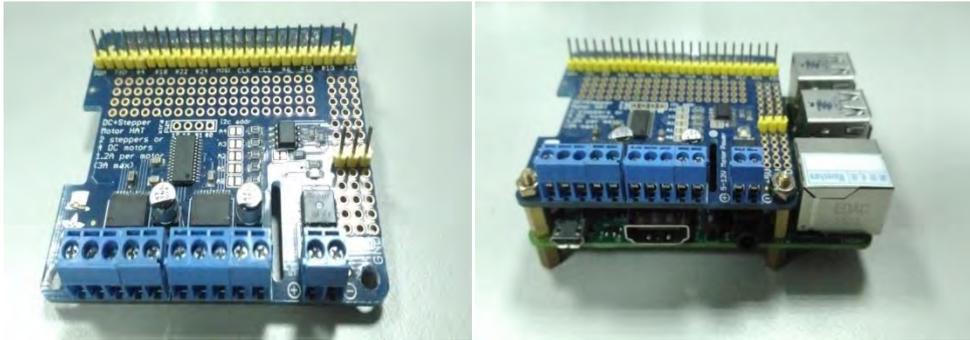


討論馬達驅動電路圖(圖.7)



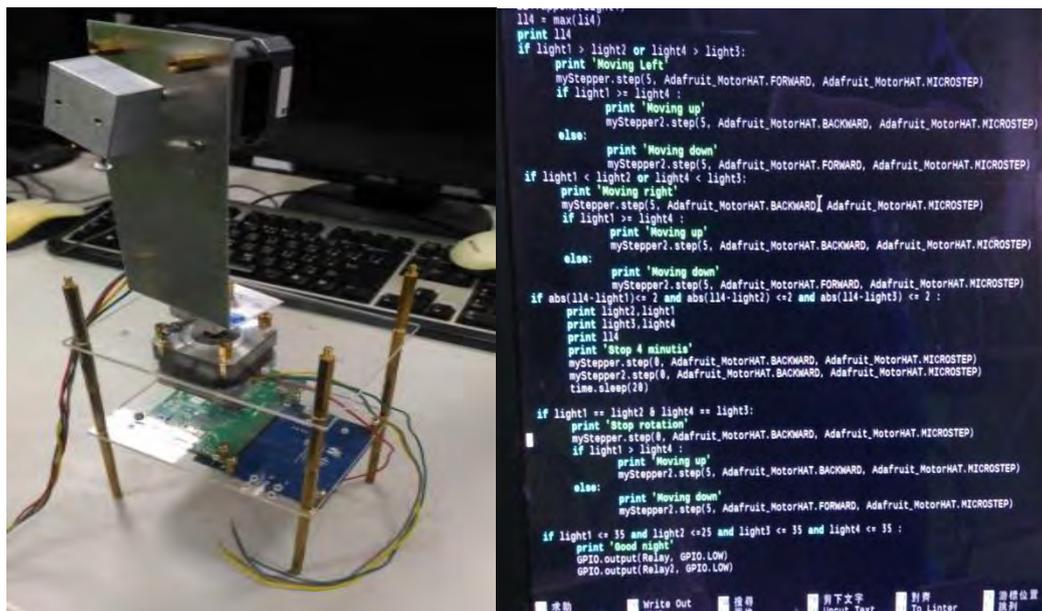
步進馬達的做動原理 (圖.8)

(二)我們這次因為使用到兩個步進馬達，而樹莓派沒有足夠的 PWM 腳位，所以我們用到這塊驅動板，因為這塊驅動板只需要使用兩個引腳(SDA 和 SCL)，就可用來驅動多個馬達。



Adafruit DC&StepperMotor HAT 驅動板(左)和裝在樹莓派上圖示(右) (圖.9)

(三)最後是我們這階段的程式編寫及測試，老師跟感知器那時一樣有範例讓我們進行改寫，再修改完後我們將元件先裝到我們的支架上，好進行程式的執行來判斷程式有沒有錯誤。



馬達測試圖(左)及驅動程式(右) (圖.10)

## 六、光源感測測試

(一)討論如何讓感知器與步進馬達搭配，讓其能夠按照我們的想法去執行操作。我們先是利用搜尋到的資料討論出一個概論，接著尋求老師的協助，詢問老師應該如何製作及程式的編寫，因為是把先前所作的兩樣元件組在一起，所以沒有花太多時間在編寫程式碼，只是微調程式有點花時間。



老師教我們感測方式 (圖.11)



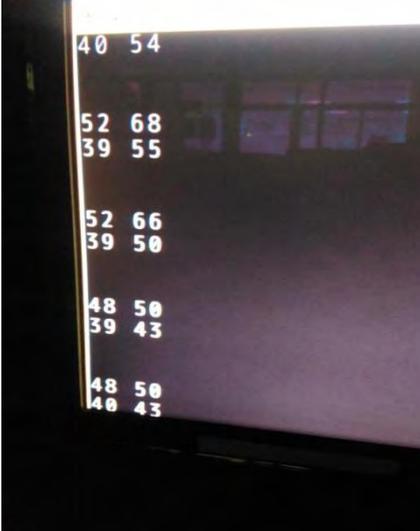
組裝元件到支架 (圖.12)



馬達與感知器組裝完成 (圖.13)

(二)再來是實際操作與程式微調，我們在這裡邊執行程式確定沒有錯誤後，開始依照讀出的數據進行程式校正，我們是以每 5 秒就驅動一次的方式來執行程式，並判斷裝置是否有達到我們要的位置，藉此半段我們的矯正是否可以。

```
# =====  
# 基本腳位設定  
#GPIO.cleanup()  
GPIO.setmode(GPIO.BCM)  
GPIO.setwarnings(False)  
sunrise = 60  
a_pin =27  
b_pin =22  
x=0  
# =====  
#設定放電函數  
#-----CDS-1-----  
def discharge_1():  
    GPIO.setup(a_pin, GPIO.IN)  
    GPIO.setup(b_pin, GPIO.OUT)  
    GPIO.output(b_pin, False)  
    time.sleep(0.01)  
# =====  
#設定充電函數並回傳充電時間  
#-----CDS-1-----  
def charge_time_1():  
    GPIO.setup(b_pin, GPIO.IN)  
    GPIO.setup(a_pin, GPIO.OUT)  
    GPIO.output(a_pin, True)  
    ta = time.time()  
    while not GPIO.input(b_pin):  
        pass  
    tb = time.time()  
    return (tb - ta) * 1000000  
# =====  
#樹莓並沒有類比腳位，利用時間差將光的強弱數字位  
#-----CDS-1-----  
def analog_read_1():  
    discharge_1()  
    return charge_time_1()  
def read_resistance_1():  
    n_1 = 20  
    total_1 = 0;  
    for i in range(1, n_1):  
        total_1 = total_1 + analog_read_1()  
    reading_1 = total_1 / float(n_1)  
    resistance_1 = reading_1 * 6.05 - 939  
    return resistance_1  
def light_from_r_1(R):  
    return math.log(1000000.0/R) * 10.0  
# =====  
try:  
    while True:
```



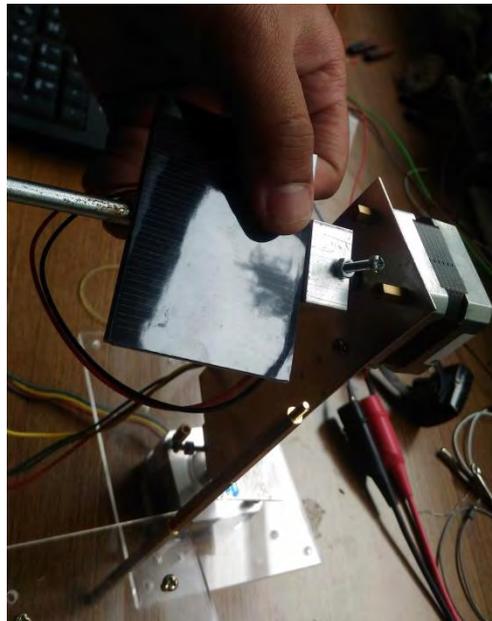
程式校正(左)與讀取出的數據(右) (圖.14)

## 七、裝置追日程式編寫及測試

(一)在這個階段我們就開始把太陽能板及電池加裝到裝置上，由於我們要讓太陽能板能夠充電，並讓電池可以放電我們在這邊則用到了一塊驅動版，這塊驅動版可以將不管是鋰電池或 USB 的電源，都轉變成穩定的輸出 5V 電壓及最大 1A 電流負載輸出，使我們在後續的程式編寫與電路配置不用考慮到太多東西。

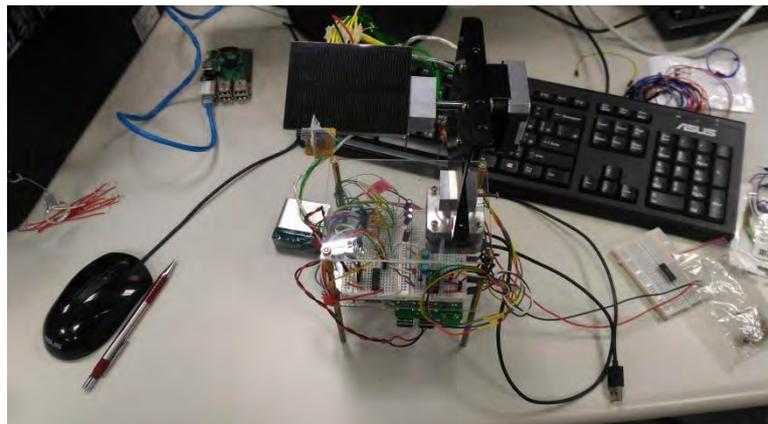


Lipo Rider Pro 驅動版 (圖.15)



太陽能板之安裝 (圖.16)

(二)接著將元件都裝配好之後，我們先是把程式加寫了電量那些因素後，再把感知器與太陽能板對好使其位於同一平面上，然後開始進行第一次的裝置操作，然而第一次的操作卻沒有想像中那麼好，裝置所追的光源與真實的光源相距太大，而我們在一次次的調整和請教老師之下這個裝置終於有成形了。



裝置組裝完成(圖.17)

```
l14 = max(l14)
print l14
if light1 > light2 or light4 > light3:
    print 'Moving Left'
    myStepper.step(5, Adafruit_MotorHAT.FORWARD, Adafruit_MotorHAT.MICROSTEP)
    if light1 >= light4:
        print 'Moving up'
        myStepper2.step(5, Adafruit_MotorHAT.BACKWARD, Adafruit_MotorHAT.MICROSTEP)
    else:
        print 'Moving down'
        myStepper2.step(5, Adafruit_MotorHAT.FORWARD, Adafruit_MotorHAT.MICROSTEP)
if light1 < light2 or light4 < light3:
    print 'Moving right'
    myStepper.step(5, Adafruit_MotorHAT.BACKWARD, Adafruit_MotorHAT.MICROSTEP)
    if light1 >= light4:
        print 'Moving up'
        myStepper2.step(5, Adafruit_MotorHAT.BACKWARD, Adafruit_MotorHAT.MICROSTEP)
    else:
        print 'Moving down'
        myStepper2.step(5, Adafruit_MotorHAT.FORWARD, Adafruit_MotorHAT.MICROSTEP)
if abs(l14-light1)<= 2 and abs(l14-light2) <=2 and abs(l14-light3) <= 2 :
    print light2,light1
    print light3,light4
    print l14
    print 'Stop 4 minutis'
    myStepper.step(0, Adafruit_MotorHAT.BACKWARD, Adafruit_MotorHAT.MICROSTEP)
    myStepper2.step(0, Adafruit_MotorHAT.BACKWARD, Adafruit_MotorHAT.MICROSTEP)
    time.sleep(24)

if light1 == light2 & light4 == light3:
    print 'Stop rotation'
    myStepper.step(0, Adafruit_MotorHAT.BACKWARD, Adafruit_MotorHAT.MICROSTEP)
    if light1 >= light4 :
        print 'moving up'
        myStepper2.step(5, Adafruit_MotorHAT.BACKWARD, Adafruit_MotorHAT.MICROSTEP)
    else:
        print 'Moving down'
        myStepper2.step(5, Adafruit_MotorHAT.FORWARD, Adafruit_MotorHAT.MICROSTEP)

if light1 <= 35 and light2 <=25 and light3 <= 35 and light4 <= 35 :
    print 'Good night'
    GPIO.output(Relay, GPIO.LOW)
    GPIO.output(Relay2, GPIO.LOW)
```

最終程式(圖.18)

## 八、最終校正與操作

(一)最後是確認都沒問題後，在陽光下實際執行操作，在真正的太陽光下，我們進行程式的校正與實際數據的紀錄，並且使用遠端控自來確認物聯網的功用，在這幾次的確認後我們覺得差不多了，我們所設計的裝置做完了，我們在看他執行最後一次測試時，心中有股莫名的喜悅感。



追日裝置操作執行(圖.19)

```
print 'Moving up'  
myStepper2.step(5, Adafruit_MotorHAT.BACKWARD, Adafruit_MotorHAT.MICROSTEP)  
else:  
print 'Moving down'  
myStepper2.step(5, Adafruit_MotorHAT.FORWARD, Adafruit_MotorHAT.MICROSTEP)
```

校正時所微調的程式(圖.20)

(二)最後的最後是我們的另一個主角**樹莓派**，我們這個裝置的最主要大腦，我們所有的東西都是經由它處理，它擁有可以連上網路這項功能，才可以形成我們這項裝置的物聯網概念，形成我們這裝置缺一不可的物件，但因為它並不是我們本次最主要的功能，所以擺到最後才來了解它。

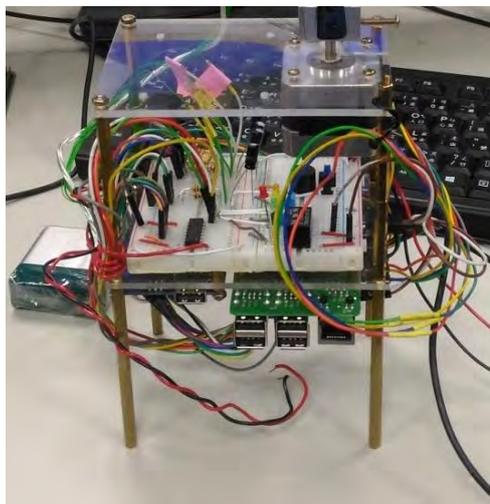


Raspberry Pi 2 Model B (樹莓派) (圖.21)

## 伍・研究結果

這個太陽追日裝置，主要分成兩大項功能，第一個就是追蹤太陽光源的追日太陽能板，第二個是能夠利用樹莓派進行物聯網功能。

首先我們是利用四象限感知器感測出光源大小，然後讓步進馬達配合各顆光敏電阻的大小進行判斷，然後由一顆控制上下 一顆控制左右，讓其作正反轉已達到無死角偵測，而在充放電方面我們利用 IC 與 Lipo Rider Pro 驅動板來操作，由 IC 的九號腳座共搭鐵，15、16 號腳做 5V 電壓輸出到太陽能板及電池，10~13 號腳與樹莓派做資料傳輸，而 Lipo Rider Pro 驅動版和太陽能板與電池連接，讓裝置可以直接由太陽能轉為電壓充進電池或直接經由面板的 USB 孔作輸出，接著在連上網路就可以成為我們要的遠端監控功能，讓我們的裝置完成研製。



成品電路圖 (圖.22)



從遠端進行監控 (圖.23)

## 陸•討論

### 一、為何不使用 Arduino ?

我們本次研究會使用樹莓派，是因為樹莓派有一個比 Arduino 要好的功能那就是連網，在現在這個甚麼都可以靠網路的時代，只要東西可以上網就能遠端控制或查看，再加上本裝置是要能夠使用者隨時隨地想用時就能運用，如果是使用 Arduino 的話就沒辦法做到這點了。

### 二、步進馬達與伺服馬達之選擇 ?

會使用步進馬達最主要是因為它可以轉動 360 度，而伺服馬達只能轉動 180 度，在轉動角動上就相差整整 180 度，對於我們想做的裝置來說實在是一大困惱，因為我們希望做到最精準的光源位置感測，那難免會有可能發生光源在一些角度是使用伺服馬達到不了的，而且還有一點是兩顆馬達的運用方式不同，步進馬達是屬於相對位置，而伺服馬達是絕對位置，在我們的設計上這一點就不同了，而且真的需要的話步進馬達也是可以做到絕對位置的。

### 三、為何感知器要分成四個象限?

我們使用四象限感知器，是因為它能夠分割四個光敏電阻讓光源不會互相干擾，使感知器能夠準確的讀取每個電阻的差異，讓我們在尋找光源大小時能準確判斷，只要光源一跑掉就能使裝置立即性的做出反應，去尋找正確的光源。

#### 四、本裝置能不能不接外部電源?

能，但是必須要換太陽能板及電池，我們現在上面的太陽能板輸出電流與電壓太小，沒有辦法達到理想的充電速度，而電池也是因為輸出電壓太小導致無法進行充電，會無法充電主要還是因為我們這個裝置本身需要的電壓就滿高的，樹莓派就要 12V 的電壓，而馬達也需要 10V 左右的電壓，就這些原件來說，以我們現在裝置上的太陽能板與電池來說是完全負荷不了的。

所以結論是本裝置可以不接外部電源，但是需要改裝能夠負荷此裝置的電池與太陽能板。

## 柒• 結論

本次的研究我們以每個人都能使用此系統為理想，所以我們以壓低經費的方式來考慮，研發出太陽能板追日系統以提高發電效率，並且結合樹莓派讓裝置達到物聯網之效用，藉此希望能讓太陽能源在未來普及化。在研究的過程中，我們學習到許多新技能，雖然遇到重重的難關，但我們覺得這樣才有研究的意義，就像那些創造出許多發明的人們，就是有他們這樣鏗而不捨的研究精神，現代人才能夠過的如此便利，我們所做的這些都只是一個契機，還是需要有人把它發揚光大。

生活上許多發明都得來不易，都是各個不同的人在非一時一地一人之所做，對我們來講我們明明常常看見別人在探討或研究各種事務，但是我們卻是到參加科展的時候才開始轉動我們的大腦，從一開始討論主題到最後做出成品，對我們來說這都是一個難忘的回憶，在作品上所花費的時間及心力，都將會是我們一輩子的收穫。

## 捌 • 參考資料及其他

一、MCP3008 之介紹。

<https://learn.adafruit.com/raspberry-pi-analog-to-digital-converters/mcp3008>

二、馬達驅動板之介紹。

<https://learn.adafruit.com/adafruit-dc-and-stepper-motor-hat-for-raspberry-pi?view=all>

三、步進馬達驅動板功能介紹 PDF

<https://cdn-learn.adafruit.com/downloads/pdf/adafruit-dc-and-stepper-motor-hat-for-raspberry-pi.pdf>

四、IC MCP 3008 功能介紹 PDF

<https://cdn-learn.adafruit.com/downloads/pdf/raspberry-pi-analog-to-digital-converters.pdf>

五、LiPo Rider Pro 功能規格

<https://www.seeedstudio.com/LiPo-Rider-Pro-p-992.html>

六、Arduino 追日系統

<http://blog.xuite.net/m0923678421/development/508263070-%5B%E4%BA%82%E7%8E%A9%5DArduino+%E8%BF%BD%E6%97%A5%E7%B3%BB%E7%B5%B1+%23001>

七、太陽能發電與追日系統

[http://web.nanya.edu.tw/mechanic/research/2\\_2\\_1021a/2\\_2\\_1021a\\_08.pdf](http://web.nanya.edu.tw/mechanic/research/2_2_1021a/2_2_1021a_08.pdf)

八、太陽照明系統追日系統

[http://web.tiit.edu.tw/mechanic/research/2\\_2\\_1021b/2\\_2\\_1021b\\_09.pdf](http://web.tiit.edu.tw/mechanic/research/2_2_1021b/2_2_1021b_09.pdf)

## 【評語】 052312

1. 作品標題定義不明確，按內容評估應為太陽能板之追日系統研發。
2. 相關系統已多有發展，創意性略不足。同時，應多探討現有追日系統所使用之方法與原因，避免閉門造車。
3. 以光敏電阻方式進行日照方向判定具有創意，但是否可行則需進一步的實驗驗證。另外，若太陽能板安裝在固定地點(在特定經緯位置下)，太陽每月每日每時的軌跡為固定可預測的，此時，是否需要導入即時的日照偵測？或是直接依據月日時設定太陽能板方位即可？可仔細思考一下。
4. 系統大致完成，但缺實驗驗證數據。
5. 相較於傳統固定式的太陽能板，追日太陽能板是否更能儲能，應由每日可達到的總發電量來評估(假設日照狀態不變)。若導入追日系統所需消耗電量超過了太陽能板因追日所能增加儲存的電量(馬達相對於系統中其他機電元件非常耗電)，追日不具意義。另外，追日系統也需要額外的建置成本和維護成本。

## 摘要

現在全球都在探討節能減碳環保意識崛起，而其中的太陽能又是一項可視性最高的再生能源。由於太陽要直射太陽能板才會得到最大的效率，所以我們設計出了這一項只要啟動就能自己追日的裝置，這個裝置利用四個光敏電阻做成感知器，並用隔板將光敏電阻切成四個象限，用以偵測四個象限接收到的光源大小，以判別出光源的位置，並以兩顆步進馬達做雙軸控制控制太陽能板，能夠隨時在最佳的位置發電，最後以lipo rider pro 作輸出穩壓以供電池充電。

此外，由於本裝置還有搭配樹莓派做設計，所以能夠透過網路及時將輸出電壓、太陽能的發電電壓等資訊上傳回控制裝置讓操作者知道這些即時訊息，以便操作者能夠做後續之使用。

## 壹、研究動機

因環保議題興起，替代能源是現今社會不可或缺的，可再生能源絕對是在好不過了，因此我們一人想了一種來討論，最後我們是以太陽能為主去收集資料，因為太陽會一直存在，所以我們覺得太陽能是很有未來趨勢的，在搜尋了很多資料下我們得到了一個結論，就是現在的效率實在是太差了，所以我們就打算以增加效率為主去做，在大家想著如何增加它的效率時，有人發現最簡單方法就是跟著太陽移動，由此動機我們就紛紛搜尋資料，丟出自己的意見並提出個案的優點，經表決後決定了要使用樹莓派當作我們的基點來去製作，打算做出一個能夠自己追著太陽的裝置，並且還因為樹莓派而使我們可以不管在何處，只要我們可以連上網路，就能即時接收或察看，現在這個時間點的太陽能發電電壓及電池電壓。

## 貳、研究目的

由於現在太陽能發電之效率普遍沒有想像中來的高，因此我們想要做出一種能提高此效率之裝置，而且希望此裝置之研發成本可以壓低，好讓其能普遍發展，以利於未來之能源發展。

此裝置再配合樹莓派來讓其能上網，以便能隨時查看數據，也能達到未來遠端遙控之目的。

## 參、研究設備及器材

本次所使用到之主要零件及設備：



## 肆、研究過程或方法

### 一、資料查詢:

Python 的基本語法學習:

我們每個人都從網路上找尋資料來練習，因為對我們來講程式語言是一個新的課程，我們在學校也只有打過其他的而已，但是我們要使用的這一種我們並沒有使用過，所以我們找了Python 條件語句這一網站當作我們的課本學習Python的程式語言。

### 二、研究的設計流程



## 三、感知器的設計及校正

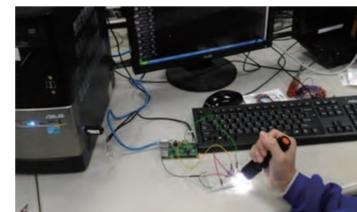
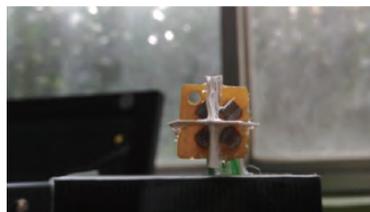
- (一) 我們一開始先討論要用甚麼當感知器的元件，在大家上網找尋各種資料之後，我們就把找來的資料做整理，並且在詢問老師之後，我們決定使用光敏電阻來製作我們的感知器，以及此感知器要做成哪種形式的，後來老師建議我們就以數學的四象限去做，而我們也有在找資料時看過，有人是以這種形式製成。
- (二) 接著我們在買完元件後，先是量測了電阻值的大小，並從中找出4個電阻值誤差在10歐姆以內的電阻，然後開始把電阻固定在一塊硬質的板面上，並用有一定硬度的紙將其分割成四個象限(建議用消光黑之顏色避免光反射影響測量值)，最後再將其接腳焊上長度足夠的銀線(避免操作時線會拉扯)。
- (三) 再來是程式的編寫，這邊運用到了樹莓派來做我們的處理器，在我們從網路上學習程式語言之後，老師找了一套範例叫我們去做改編，讓其能夠配合我們感知器去做執行的程式，接著我們在程式完成後開始校正，讓其可以準確的判斷光源位置。



討論過程 (圖1)



光敏電阻的量測與成品圖 (圖2)



光敏電阻的校正 (圖3)

## 四、IC MCP3008數據讀取及校正

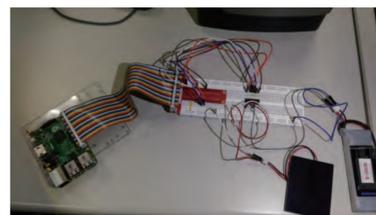
- (一) 這次我們會使用這塊IC是因為它能夠將類比轉成數位，彌補我們樹莓派無法直接使用類比訊號的缺點，而它在讀取ADC數值時，會傳回一個10-bit的數字(介於0到1023之間)，輸入至樹莓派，由此可以根據輸入電壓與傳回值的電壓，計算出類比訊號的值。
- (二) 接著我們把IC搭配上樹莓派去配置線路，由IC提供電源(樹莓派給IC 5V)與搭鐵給電池及太陽能板，再由IC接收回來的訊號轉變成數位給樹莓派，由樹莓派處理及儲存數據讓使用者可以隨時觀看。



IC MCP3008 (圖4)



樹莓派所記錄的IC數據 (圖5)



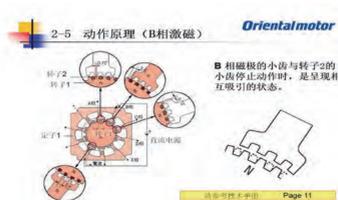
IC與樹莓派搭配之線路 (圖6)

## 五、步進馬達正轉逆轉程式編寫及測試

- (一) 我們接著討論到如何使太陽能板進行所謂的追日移動，在找尋了許多資料後，我們決定使用步進馬達，由於步進馬達可以達到360度的轉向，所以我們使用到兩顆來做全方位的轉向，來使我們的追日裝置可以最無死角的追蹤。
- (二) 我們這次因為使用到兩個步進馬達，而樹莓派沒有足夠的PWM腳位，所以我們用到這塊驅動板，因為這塊驅動板只需要使用兩個引腳(SDA和SCL)，就可用來驅動多個馬達。
- (三) 最後是我們這階段的程式編寫及測試，老師跟感知器那時一樣有範例讓我們進行改寫，再修改完後我們將元件先裝到我們的支架上，好進行程式的執行來判斷程式有沒有錯誤。



討論馬達驅動電路圖 (圖7)



步進馬達的做動原理 (圖8)



Adafruit DC&StepperMotor HAT驅動板(左)和裝在樹莓派上圖示(右) (圖9)



馬達測試圖(左)及驅動程式(右) (圖10)

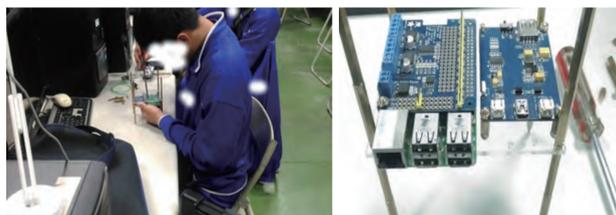


## 六、光源感測測試

- (一) 討論如何讓感知器與步進馬達搭配，讓其能夠按照我們的想法去執行操作。我們先是利用搜尋到的資料討論出一個概論，接著尋求老師的協助，詢問老師應該如何製作及程式的編寫，因為是把先前所作的兩樣元件組在一起，所以沒有花太多時間在編寫程式碼，只是微調程式有點花時間。



老師教我們感測方式 (圖11)



組裝元件到支架 (圖12)



馬達與感知器組裝完成 (圖13)

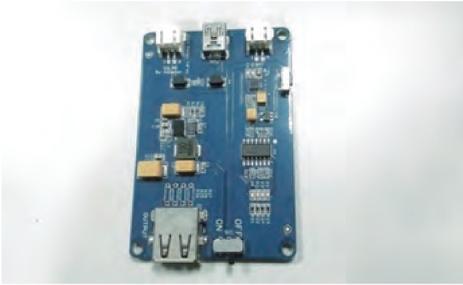
- (二) 再來是實際操作與程式微調，我們在這裡邊執行程式確定沒有錯誤後，開始依照讀出的數據進行程式校正，我們是以每5秒就驅動一次的方式來執行程式，並判斷裝置是否有達到我們的位置，藉此半段我們的矯正是否可以。



程式校正(左)與讀取出的數據(右) (圖14)

## 七、裝置追日程式編寫及測試

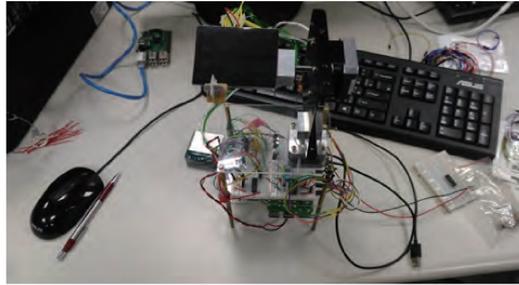
- (一) 在這個階段我們就開始把太陽能板及電池加裝到裝置上，由於我們要讓太陽能板能夠充電，並讓電池可以放電我們在這邊則用到了塊驅動版，這塊驅動版可以將不管是鋰電池或 USB 的電源，都轉變成穩定的輸出 5V 電壓及最大 1A 電流負載輸出，使我們在後續的程式編寫與電路配置不用考慮到太多東西。
- (二) 接著將元件都裝配好之後，我們先是把程式加寫了電量那些因素後，再把感知器與太陽能板對好使其位於同一平面上，然後開始進行第一次的裝置操作，然而第一次的操作卻沒有想像中那麼好，裝置所追的光源與真實的光源相距太大，而我們在一次次的調整和請教老師之下這個裝置終於有成形了。



Lipo Rider Pro 驅動版 (圖15)



太陽能板之安裝 (圖16)



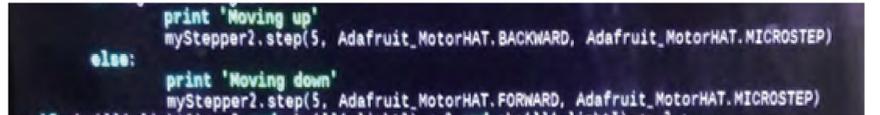
裝置組裝完成 (圖17)



最終程式 (圖18)

## 八、最終校正與操作

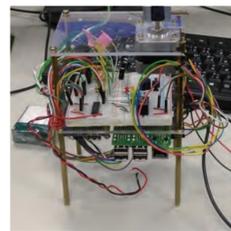
- (一) 最後是確認都沒問題後，在陽光下實際執行操作，在真正的太陽光下，我們進行程式的校正與實際數據的紀錄，並且使用遠端控自來確認物聯網的功用，在這幾次的確認後我們覺得差不多了，我們所設計的裝置做完了，我們在看他執行最後一次測試時，心中有股莫名的喜悅感。
- (二) 最後的最後是我們的另一個主角樹莓派，我們這個裝置的最主要大腦，我們所有的東西都是經由它處理，它擁有可以連上網路這項功能，才可以形成我們這項裝置的物聯網概念，形成我們這裝置缺一不可的物件，但因為它並不是我們本次最主要的功能，所以擺到最後才來了解它。



追日裝置操作執行 (圖19) Raspberry Pi 2 Model B (樹莓派) (圖21) 校正時所微調的程式 (圖20)

## 伍、研究結果

這個太陽追日裝置，主要分成兩大項功能，第一個就是追蹤太陽光源的追日太陽能板，第二個是能夠利用樹莓派進行上網功能。首先我們是利用四象限感知器感測出光源大小，然後讓步進馬達配合各顆光敏電阻的大小進行判斷，然後由一顆控制上下 一顆控制左右，讓其作正反轉已達到無死角偵測，而在充放電方面我們利用 IC 與 Lipo Rider Pro 驅動板來操作，由 IC 的九號腳座共搭鐵，15、16號腳做 5V 電壓輸出到太陽能板及電池，10~13號腳與樹莓派做資料傳輸，而 Lipo Rider Pro 驅動版和太陽能板與電池連接，讓裝置可以直接由太陽能轉為電壓充進電池或直接經由面板的 USB 孔作輸出，接著在連上網路就可以成為我們要的遠端監控功能，讓我們的裝置完成研製。



成品電路圖 (圖22) 從遠端進行監控 (圖23)

## 陸、討論

### 一、為何不使用 Arduino ?

我們本次研究會使用樹莓派，是因為樹莓派有一個比 Arduino 要好的功能那就是上網，在現在這個甚麼都可以靠網路的時代，只要東西可以上網就能遠端控制或查看，再加上本裝置是要能夠使用者隨時隨地想用時就能運用，如果是使用 Arduino 的話就沒辦法做到這點了

### 二、步進馬達與伺服馬達之選擇？

會使用步進馬達最主要是因為它可以轉動 360 度，而伺服馬達只能轉動 180 度，在轉動角動上就相差整整 180 度，對於我們想做的裝置來說實在是一大困惱，因為我們希望做到最精準的光源位置感測，那難免會有可能發生光源在一些角度是使用伺服馬達到不了的，而且還有一點是兩顆馬達的運用方式不同，步進馬達是屬於相對位置，而伺服馬達是絕對位置，在我們的設計上這一點就又不一样了，而且真的需要的話步進馬達也是可以做到絕對位置的。

### 三、為何感知器要分成四個象限？

我們使用四象限感知器，是因為它能夠分割四個光敏電阻讓光源不會互相干擾，使感知器能夠準確的讀取每個電阻的差異，讓我們在尋找光源大小時能準確判斷，只要光源一跑掉就能使裝置立即性的做出反應，去尋找正確的光源。

### 四、本裝置能不能不接外部電源？

能，但是必須要換太陽能板及電池，我們現在上面的太陽能板輸出電流與電壓太小，沒有辦法達到理想的充電速度，而電池也是因為輸出電壓太小導致無法進行充電，會無法充電主要還是因為我們這個裝置本身需要的電壓就滿高的，樹莓派就要 12V 的電壓，而馬達也需要 10V 左右的電壓，就這些原件來說，以我們現在裝置上的太陽能板與電池來說是完全負荷不了的。

所以結論是本裝置可以不接外部電源，但是需要改裝能夠負荷此裝置的電池與太陽能板。

## 柒、結論

本次的研究我們以每個人都能使用此系統為理想，所以我們以壓低經費的方式來考慮，研發出太陽能板追日系統以提高發電效率，並且結合樹莓派讓裝置達到物聯網之效用，藉此希望能讓太陽能源在未來普及化。在研究的過程中，我們學習到許多新技能，雖然遇到重重的難關，但我們覺得這樣才有研究的意義，就像那些創造出許多發明的人們，就是有他們這樣鍥而不舍的研究精神，現代人才能夠過的如此便利，我們所做的這些都只是一個契機，還是需要有人把它發揚光大。

生活上許多發明都得來不易，都是各個不同的人在非一時一地一人之所做，對我們來講我們明明常常看見別人在探討或研究各種事務，但是我們卻是到參加科展的時候才開始轉動我們的大腦，從一開始討論主題到最後做出成品，對我們來說這都是一個難忘的回憶，在作品上所花費的時間及心力，都將會是我們一輩子的收穫。