

醒吾技術學院
資訊科技應用研究所
碩士論文

應用攝影機於太陽能追日系統研製
Applying Video Camera to the Desing of Sun-Tracking System

研 究 生： 徐政甯

指導教授： 葉思明

中華民國 一〇一 年 六 月

應用攝影機於太陽能追日系統研製

Applying Video Camera to the Desing of Sun-Tracking
System



研 究 生 ： 徐政甯

指導教授 ： 葉思明

醒吾技術學院
資訊科技應用研究所一百學年度
碩士論文

A Thesis
Submitted to
Institute of Applied Information Technology
Hsing-Wu College
in Partial Fulfillment of the Requirements
for the Degree of
Master of Informatics

June 2012
Lin-Ko, Taipei County, Taiwan, Republic of China
中華民國 一〇一 年 六 月

摘 要

太陽能發電效率受到太陽光照射角度影響甚巨，本研究在開發攝影機應用於太陽能追日系統，利用攝影機做為追蹤系統感測器，採用影像處理的方式開發主動式太陽能追日系統。首先在攝影鏡頭前方設置一片濾光片以過濾多餘及較弱光源，將所得太陽光源影像由電腦進行影像處理，計算太陽光源影像偏離所需位置，再透過串列通訊由 RS-232 控制電路將資料送至由 AI 馬達製成的雙軸追蹤機構，移動太陽能板至所需位置，如此不斷的定時執行此一動作，以達到動態修正受光角度，提高太陽能發電效率，經實驗測試結果，平均追蹤精確度良好。由此顯示，所設計應用攝影機於太陽能追日系統是可行且性能優異。

關鍵字：太陽能發電、主動式追日系統、影像處理



ABSTRACT

The quantity of solar power energy is relative to an angle of shine. The research is successful to design a Sun-Tracking system via applying video camera. We use video camera to a sensor of Sun-Tracking systems. Furthermore, the technique of image process is used to active tracking systems. First, we set an attachment lens on the front of video camera to filter an unnecessary light. Next, the collected image of shine is processed via computer to calculate the displacement of light source. Finally, the processed data is transferred to AI motor tracking system by RS-232 serial port to relocate the position of moving solar panels. To increase the efficiency of solar energy, dynamic adjusting the angle of received shine is needed and repeated. The experiment results achieve a good performance. Therefore, the design is a practically and efficient system.

Keywords: Solar power energy, Active tracking system, Image process.



誌 謝

本論文得以順利完成，首先由衷感謝指導教授葉思明老師對於影像處理領域的指導及實驗室蕭棟元老師以認真、用心的態度教導學生系統控制專業知識能力，並提供許多研究上的寶貴經驗與心得，在此致上最誠摯的謝意。

其次感謝蕭棟元和葉大綱教授對本論文的指正與建議，使本論文得以更加完善，在此由衷感謝二位口試委員。

再者感謝實驗室的同學王亮台、鄧為仁、林張暉，他們在生活與課業上給予莫大的幫助，以及感謝學長林品洋，他們在課業上的指導與經驗上的傳承。

最後感謝我的同事在研究所求學期間，對我的協助與鼓勵，另特別感謝老婆詹家欣，在我求學中正值懷孕及生產期間，為了能讓我專心研究，獨自不畏辛勞默默照顧小朋友的一切，使我能無後顧之憂的順利完成學業，在此將完成此論文的喜悅與家人、師長、朋友及關心我的人一同分享。



目 錄

摘 要	i
ABSTRACT.....	ii
誌 謝	iii
目 錄	iv
表 目 錄	vi
圖 目 錄	vii
1. 緒論.....	1
1.1. 研究動機	1
1.2. 文獻探討	1
1.3. 研究內容	1
2. 理論基礎.....	2
2.1. 太陽能追日系統	2
2.1.1. 太陽能電池	2
2.1.2. 太陽能電池種類	3
2.1.3. 太陽光入射角	6
2.1.4. 太陽光方位角	7
2.1.5. 太陽能追蹤系統分類	7
2.2. 影像處理	11
2.2.1. 影響數位拍攝因素	11
2.2.2. 感光度和解析度	11
2.2.3. 影像擷取	11
2.2.4. 顏色偵測與色彩空間	11
2.2.5. 影像二值化	16
3. 追日系統原理與架構.....	17
3.1. 追控系統原理	17
3.2. 追蹤機構設計與製作	17

3.2.1. AI 馬達.....	18
3.2.2. AI 馬達 MPU 控制器.....	22
3.2.3. AI 馬達設定及測試.....	24
3.3. 感測器設計與製作	26
3.3.1. 攝影機改造	27
3.3.2. 攝影機測試	29
3.3.3. 影像門檻值	30
3.3.4. 攝影機可視角測量	33
4. 系統整合與實驗結果	36
4.1. 硬體控制電路設計	36
4.1.1. RS232 簡介.....	36
4.1.2. RS-232 控制電路.....	39
4.2. 軟體邏輯設計	43
4.2.1. C# 語言簡介	43
4.2.2. .NET Framework 簡介.....	44
4.2.3. OpenCV 簡介.....	44
4.2.4. AForge.NET 簡介	45
4.2.5. 系統介面設計	46
4.3. 系統邏輯運作流程	47
4.4. 追控系統性能測試	50
4.4.1. 測試環境	50
4.4.2. 精確度測試	51
4.4.3. 實驗結果	53
5. 討論與結論.....	63
5.1. 討論	63
5.2. 結論	63
參考文獻.....	65
附錄.....	67

表 目 錄

表格 2.1 SC5848 太陽能電池規格表.....	5
表格 3.1 WCK-1108T1AI 馬達規格表.....	20
表格 3.2 馬達控制封包碼表	23
表格 3.3 羅技 C310 網路攝影機規格表	27
表格 4.1 RS232 腳位功能表.....	36
表格 4.2 BAUD RATE 關係表	39
表格 4.3 室內發電量測試一覽表	53
表格 4.4 室外發電量測試一覽表	54



圖 目 錄

圖 2.1 光伏特效應示意圖	3
圖 2.2 太陽能電池材料分類	4
圖 2.3 太陽能電池依構造分類	4
圖 2.4 勝特力 SC5848 非晶矽太陽能電池.....	5
圖 2.5 入射角和方位角示意圖	6
圖 2.6 東-西方向移動單軸追蹤器	8
圖 2.7 方位角轉動單軸追蹤器	8
圖 2.8 東-西與南-北轉動雙軸追蹤器	9
圖 2.9 仰角與方位角轉動雙軸追蹤器	9
圖 2.10 CIELAB 色彩空間示意圖	14
圖 2.11 HSV 色彩空間示意圖一	15
圖 2.12 HSV 色彩空間示意圖二	16
圖 3.1 系統架構方塊示意圖	17
圖 3.2 AI 馬達雙軸式追蹤機構示意圖	18
圖 3.3 AI 馬達之分解圖	19
圖 3.4 AI 馬達為 WCK-1108T1 規格圖	19
圖 3.5 AI 馬達解析度角度示意圖	21
圖 3.6 AI 馬達控制腳位	21
圖 3.7 AI 馬達之結構圖	22
圖 3.8 電腦主控端控制多顆 AI 馬達	23
圖 3.9 馬達角度位置圖	24
圖 3.10 WCK PROGRAMER 程式操作說明圖一	24
圖 3.11 WCK PROGRAMER 程式操作說明圖二	25
圖 3.12 WCK PROGRAMER 程式操作說明圖三	25
圖 3.13 WCK PROGRAMER 程式操作說明圖四	26
圖 3.14 羅技 C310 網路攝影機實體圖	27
圖 3.15 感測器不使用遮光罩實際圖	28
圖 3.16 感測器使用遮光罩實際圖	29
圖 3.17 攝影機實際測試圖	29

圖 3.18 系統測試實際設備圖	30
圖 3.19 灰階值 100 測試圖	31
圖 3.20 灰階值 150 測試圖	31
圖 3.21 灰階值 180 測試圖	32
圖 3.22 灰階值 220 測試圖	32
圖 3.23 灰階值 240 測試圖	33
圖 3.24 直角三角形圖	33
圖 3.25 測量方法示意圖	34
圖 3.26 10 公分實際拍攝圖	35
圖 3.27 20 公分實際拍攝圖	35
圖 4.1 RS232 25 及 9PIN 公頭腳位功能圖	37
圖 4.2 RS232 25 及 9PIN 母頭腳位功能圖	37
圖 4.3 RS232 DATA 傳輸示意圖	38
圖 4.4 位準轉換示意圖	39
圖 4.5 RS-232 控制電路圖	40
圖 4.6 RS-232 控制電路實際完成圖	40
圖 4.7 追蹤機構與控制電路整合實際完成圖	41
圖 4.8 全系統硬體實際完成圖	42
圖 4.17 系統控制介面圖	46
圖 4.18 台灣四季仰角與方位角	47
圖 4.19 影像座標示意圖	48
圖 4.20 系統運作流程圖	49
圖 4.21 室內測試系統設備實際圖	50
圖 4.22 室外測試系統介面實際圖	51
圖 4.23 自製日晷精確度測量示意圖	52
圖 4.24 自製日晷實際完成圖	52
圖 4.25 室內不追蹤模式 0 度發電圖	55
圖 4.26 室內不追蹤模式 18.43 度發電圖	55
圖 4.27 室內不追蹤模式 33.69 度發電圖	56
圖 4.28 室內不追蹤模式 45 度發電圖	56
圖 4.29 室內不追蹤模式 53.13 度發電圖	57
圖 4.30 室內不追蹤模式 59.03 度發電圖	57
圖 4.31 室內追蹤模式 0 度發電圖	58
圖 4.32 室內追蹤模式 33.69 度發電圖	58
圖 4.33 室內追蹤模式 45 度發電圖	59

圖 4.34 室內追蹤模式 59.03 度發電圖	59
圖 4.35 戶外不追蹤模式 0 度發電圖	60
圖 4.36 戶外不追蹤模式 33.6 度發電圖	60
圖 4.37 戶外不追蹤模式 45 度發電圖	61
圖 4.38 戶外追蹤模式 0 度發電圖	61
圖 4.39 戶外追蹤模式 33.6 度發電圖	62
圖 4.40 戶外追蹤模式 45 度發電圖	62



1. 緒論

1.1. 研究動機

由於科技發展造成我們面臨能源短缺的問題，各國正重視再生能源的開發，積極發展新能源，改善能源的利用效率。天然能源或再生能源的利用，根據分析結果，太陽能的潛力無窮[1]，以分析全球性陽光照射地球 40 分鐘帶來的能量，相當於全球一年的能源總消耗量[2]；若單以台灣地區而言，位於亞熱帶地區且靠近赤道，太陽的日照量非常充足，若能取用的太陽能，將不會造成任何污染及噪音，所以相信全面開發太陽能的應用，對於解決能源問題是有幫助的。目前大部分的太陽能系統，都是採用固定式，所以無法隨時讓太陽能電池板與陽光保持垂直，無法使太陽能電池長時間發揮其最大效率[3]，因此要想使太陽能電池可以得到最大的效率，太陽能板可以主動且正確追日，準確的掌握太陽軌跡將是重要的課題。

1.2. 文獻探討

關於太陽能追日系統的研究目前已有相當多研究成果，其中所提出以數學模式計算太陽位置的方法[4]，但是此種方法須透過時區等各種換算，顯得較為麻煩；以固定式太陽能電池安裝於該地區某區間角度，以取得太陽能電池最佳供電效率[3]，所提出的方法為固定式較佳的方案，但無法動態且適時調整太陽能電池角度，取得最佳的發電效率；利用小型太陽能電池為感測器追蹤控制平台來追蹤光源，此做法雖感測器能達到有效偵測角度約 45-135 度，但製作規格與設計方面，會使運轉過程中有誤差[5]；在主動式太陽能追日系統感測器方面，以國立台灣大學機械工程學研究所所發表的集光式太陽光發電追控系統，說明光敏電阻作為感測器[6]是一種便宜可靠度佳又方便的光感測元件，但由於光敏電阻規格性能不一，需篩選後方能使用，且由於雲的種類變化多端，對於太陽輻射的影響也不盡相同，因而造成光線漫射影響，容易使光感測器元件的追蹤精度不佳，由此可見，如何有效的集中太陽光源增加發電效率是值得開發的技術，其中又以主動式且精確太陽能追日系統最為值得開發。

1.3. 研究內容

目前一般的太陽能電池發電，大部分是使用固定式收集太陽光源，如此會失去許多太陽輻射能收集的機會。因此，為了追求研發簡單、且效率高之太陽能追日系統，本研究開發製作不同以往感測器之主動式太陽能追日系統模型機，以攝影機做為感測器的太陽能追日系統，其研究內容包括：追蹤機構設計與製作、感測器設計與製作及系統整合與實際測試。

2. 理論基礎

本章節主要介紹太陽能、太陽能電池、追蹤系統、影像處理、色彩空間轉換。

2.1. 太陽能追日系統

太陽能是人類可利用最豐富的能源，在往後數億年間也不會產生明顯的變化，且隨處可得，能就地開發利用，對於偏遠地區更有利用價值。太陽能也是一種潔淨的能源，對地球不會增加額外的熱負荷，相較其他能源來說是非常環保，但缺點如：太陽能量密度較低，要收集大量的能源，則太陽能裝置需要有相當的面積，表示系統的製作成本相對較高，另太陽能也受環境、氣候與晝夜的影響，裝置無法穩定的輸出電力，因此需要電力或儲熱裝置，增加技術上的困難及成本。

2.1.1. 太陽能電池

任何只要能轉換入射光子的能量而直接產生輸出電壓，就可稱為光伏特效應 (photovoltaic effect)，如圖 2.1 所示。光伏特效應一般是指光子射到半導體 p-n 二極體後，p-n 二極體的二端電極，產生可輸出功率的電壓伏特值。這過程包括光子射到半導體內產生電子-電洞對，電子和電洞因半導體 p-n 接面形成的內建電場作用而分離，電子和電洞往相反的方向各自傳輸至二端電極來輸出，所以光伏特效應一般是跟 p-n 二極體有關的；另入射光子在空間電荷區被吸收產生電子和電洞，因為內建電場的影響而產生從 n-型區向 p-型區的漂移電流，就是所謂的光電流 (photocurrent)。光伏特效應中的光電流，其流向是從 n-型區向 p-型區，這對 p-n 二極體而言，這剛好是反向偏壓 (reverse bias) 的電流方向。光伏特效應中，p-n 接面區的空間電荷區的內建電場之功用就是使入射光子被吸收產生電子-電洞對在複合 (recombination) 前被分開，而產生光電流。

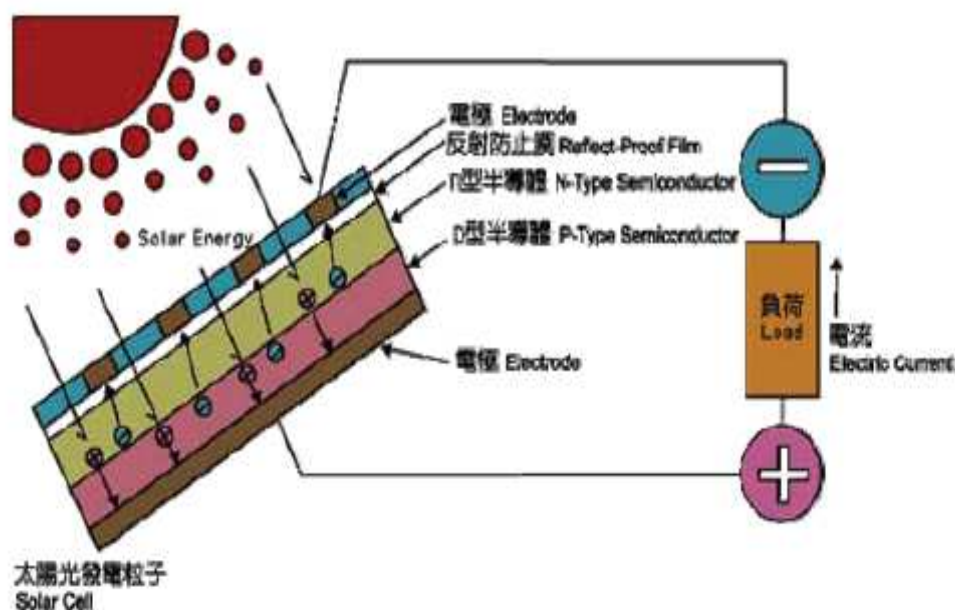


圖 2.1 光伏特效應示意圖

2.1.2. 太陽能電池種類

太陽能電池依材料分類如圖 2.2 所示，以構造分類如圖 2.3 所示。根據國外權威刊物「PV news」報導，全世界太陽能電池的生產總量為 512MW，其中單晶矽電池 150MW，多晶矽電池 279MW，非晶矽電池 28MW，佔全部太陽能電池的 89%。所以目前矽系太陽能電池還是太陽能電池的主流。依穩定性和壽命來看，單晶矽和多晶矽製成的太陽能電池輸出特性非常穩定，而非晶矽太陽能電池經陽光照射後很容易老化。而依材質分析，單晶矽片的厚度在 300 μm 以下，不能彎曲，故需將其裝於鋁合金框架內製成模組，受光面積用透明鋼化玻璃覆蓋，以防止砂石、冰雹的衝擊。相較之下，非晶矽的厚度在 1 μm 以下，可以捲曲，甚至可以在輕質基板上製成薄膜。此外，非晶矽太陽能電池的最大優點是外形似薄膜，在製膜時可以自由控制電壓與電流，依所需的功率製造。由於存在特製的透明電極，故能夠過光線，作為建築物的玻璃窗使用。薄膜型太陽能電池所需材料少，除了矽還可以用砷化鎵等化合物代替。它有兩項優點，一是可用工業製版的印刷法生產薄膜；二是可以與建築工業合作產生特殊的建材，如太陽能電池屋頂或太陽能電池窗口。而化合物半導體太陽能電池中典型代表為砷化鎵，因為它可適應於不同的太陽光譜照射，且砷化鎵比矽的耐放射性更高，所以可應用於宇宙太陽能電池。目前在通信衛星和科學偵測上，砷化鎵太陽能電池已實用化。

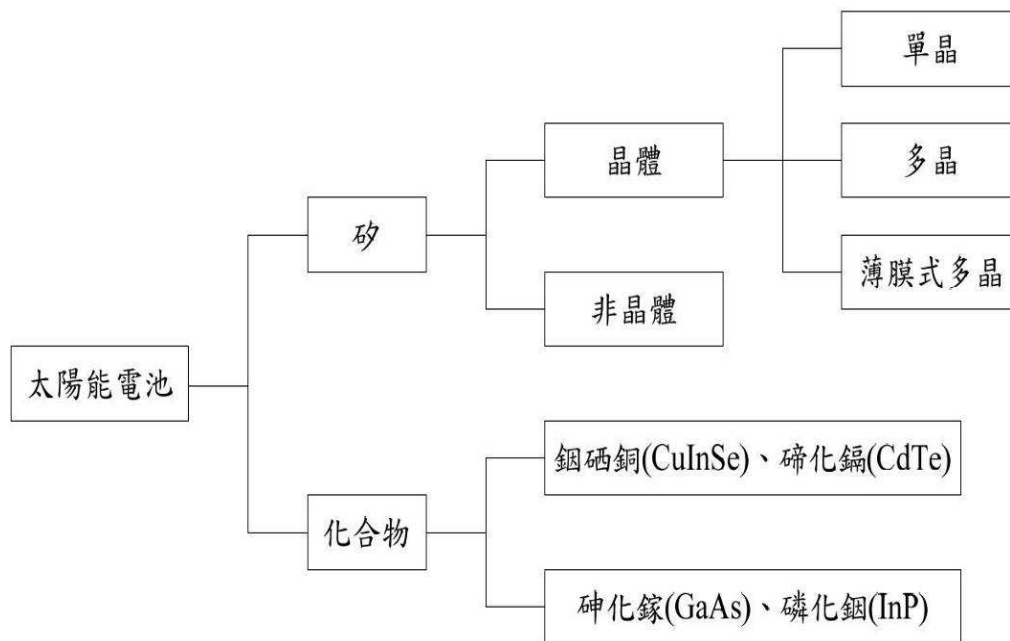


圖2.2 太陽能電池材料分類

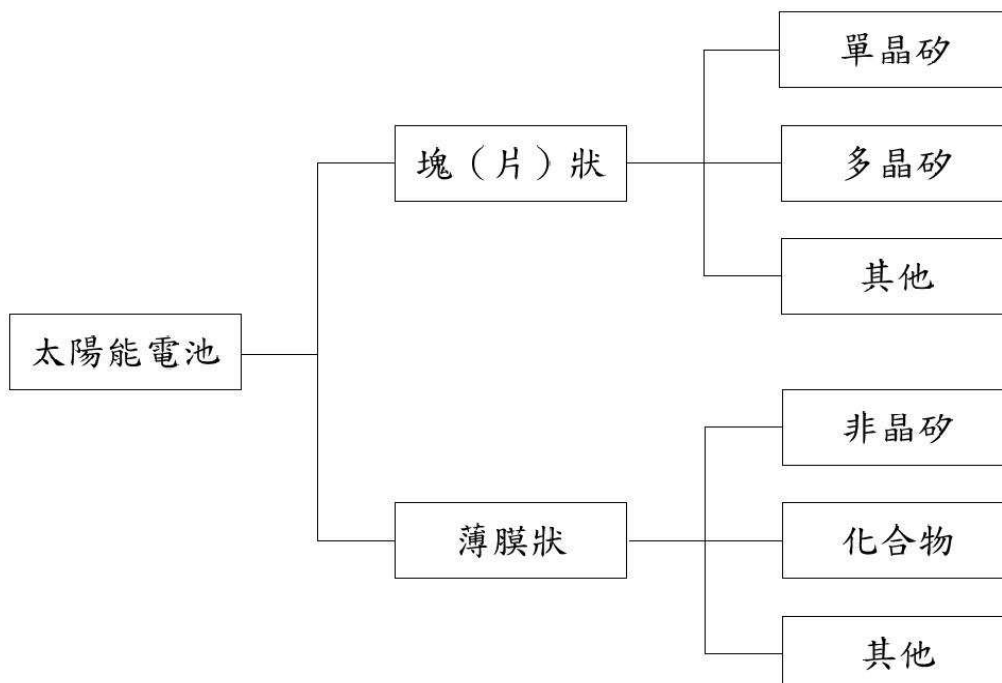


圖 2.3 太陽能電池依構造分類

本系統採用兩塊勝特力公司 SC5848 (58x48mm 8V 8mA) 太陽能電池 (非晶矽) 作為測試，其規格如 2.1 表所示，最大輸出值，需在環境溫度：25℃、光線強度：100mW/cm 達成，溫度及光線強度皆會影響輸出電壓及電流，實際圖如 2.4 所示

表格 2.1

Sc5848 太陽能電池規格表

材質	非晶矽
尺寸 (mm)	58x48mm
工作電壓	8V
工作電流	8mA
開路電壓	12.5V
短路電流	11mA
厚度	2mm

資料來源：<http://shop.cpu.com.tw>(廣華電子)

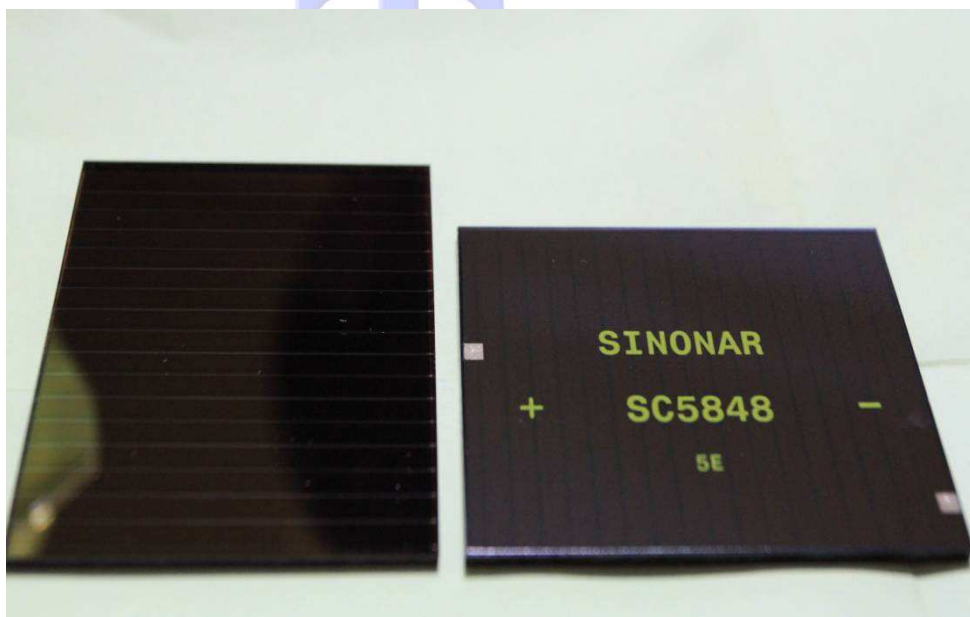


圖 2.4 勝特力 Sc5848 非晶矽太陽能電池

2.1.3. 太陽光入射角

地球自轉以及繞太陽公轉的現象已眾所皆知，地球上任何一個時間與地點，都已能準確的預測太陽與地面所夾的仰角以及東西向的方位。地球在黃道面上以近乎圓形的軌道繞太陽公轉，而地球自轉軸與黃道面的夾角恆維持著 23.45 度的關係。每年夏至，地球在公轉軌道的一端，正午時太陽直射北迴歸線。冬至的中午，地球在公轉軌道呈 180 度的另一端，太陽直射南迴歸線。地球除了繞太陽公轉外也同時在自轉，地球自轉可影響到太陽的仰角以及東西向的方位。以台灣地區而言，夏至時節早上從東北東方升起而傍晚在西北西方落下。冬至時分，太陽從東南東方升起在西南西方落下。仰角及方位角隨著時間在慢速的變化，它的軌跡在天空中劃一個非平面的弧形。可看出入射角就是以站在地球上的觀看者，其頭頂正上方與天體圓的交點為 P 點，即所謂的天頂，觀看者所看到太陽(S 點)與天頂(P 點)之間的夾角 θ_z 就是入射角，也稱為天頂角，如圖 2.5 所示。

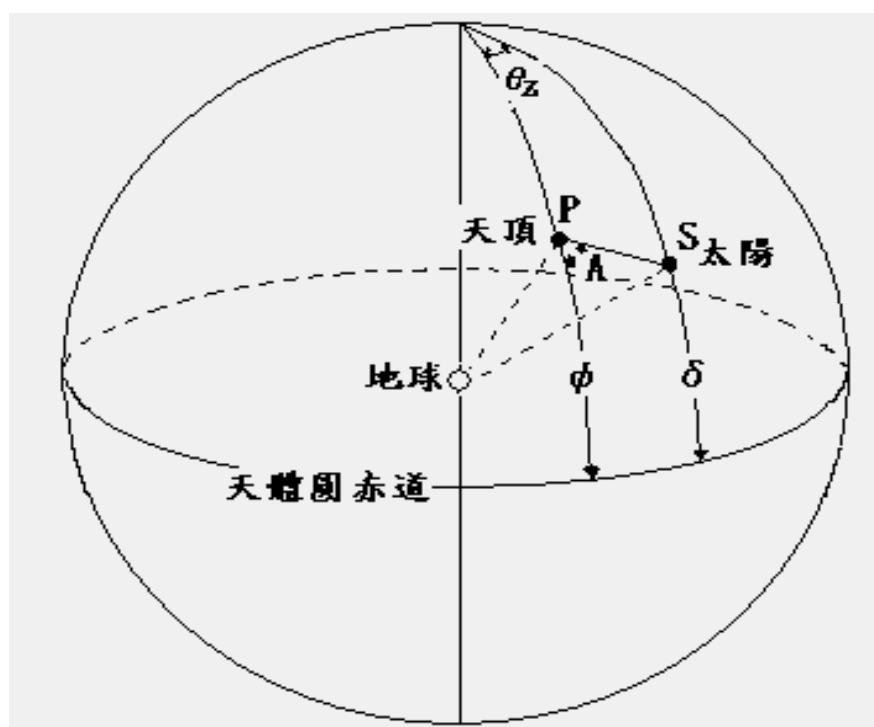


圖 2.5 入射角和方位角示意圖

而太陽的入射角計算公式[7] θ_z 可由(2-1)式求出：

$$\theta_z = \cos^{-1} [\sin \delta \sin \phi + \cos \delta \cos \phi \cos w] \quad (2-1)$$

再求出仰角 α (2-2)式

$$\alpha = 90^\circ - \theta_z \quad (2-2)$$

2.1.4. 太陽光方位角

方位角 A 為觀看者所看到太陽(S點)和天頂(P點)之間的連線，與自天頂(P點)往天體球赤道(正南方)延伸線之間的夾角公式[8]，太陽之方位角 A 為(2-3)式：

$$|A| = \cos^{-1} \left[\frac{\sin \alpha \sin \phi - \delta}{\cos \alpha \cos \phi} \right] \quad (2-3)$$

以上的時間稱為太陽時間，但事實上地球繞行太陽的軌道並非完全圓形，而且地球依各地區經度之不同僅劃分成有限個時區，要更精確的計算個別地點之仰角與方位角。利用以上的方法，同一追日系統無論架設於地球上任一地點，在任何時刻均可準確的計算太陽的仰角及方位角。

2.1.5. 太陽能追蹤系統分類

太陽能追蹤器能有效讓太陽入射光與太陽能板受光面垂直，以提高光源的吸收量，增進發電效率，太陽能追蹤機構可分為單軸式和雙軸式兩種。

一、單軸式太陽能追蹤系統

單軸式主要是指依東-西方向移動或方位角轉動來追蹤太陽，如圖 2.6、圖 2.7 所示。主要應用於不需高度精度追蹤(因為使用單軸式追蹤器會有南北向的誤差)、低製造成本等因素。

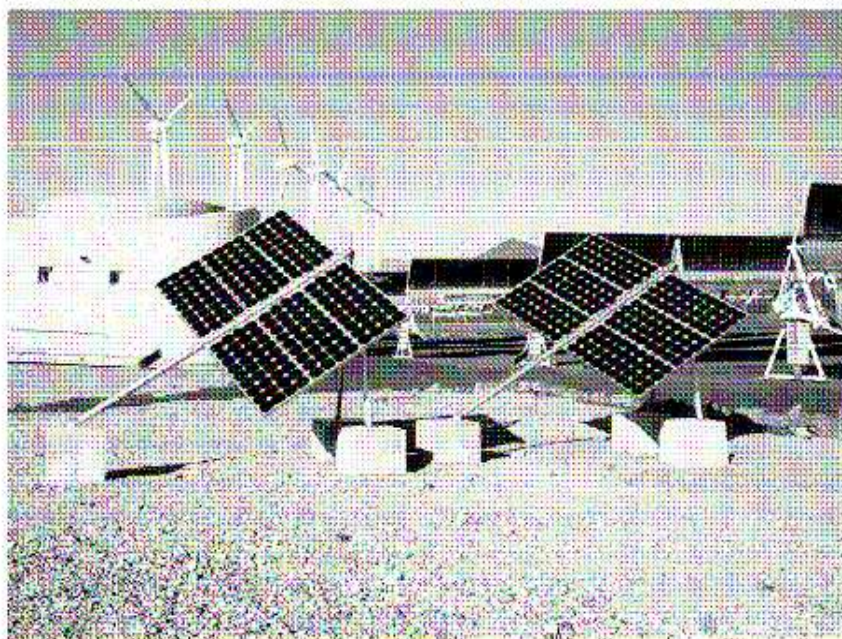


圖 2.6 東-西方向移動單軸追蹤器

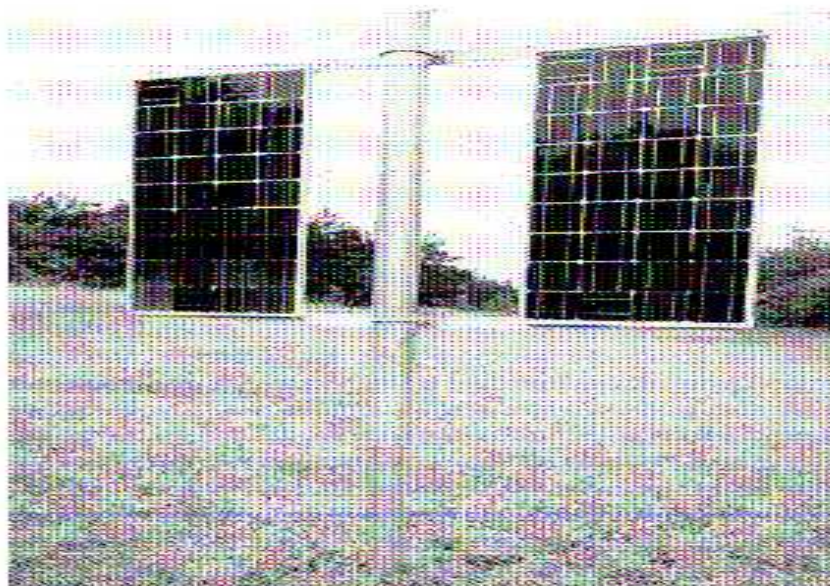


圖 2.7 方位角轉動單軸追蹤器

二、雙軸式太陽能追蹤系統

雙軸追蹤器是指東-西與南-北兩個方向轉動，或是仰角與方位角轉動來追蹤太陽，如圖 2.8、圖 2.9，其主要應用於要求高精度追蹤及高聚光型系統。

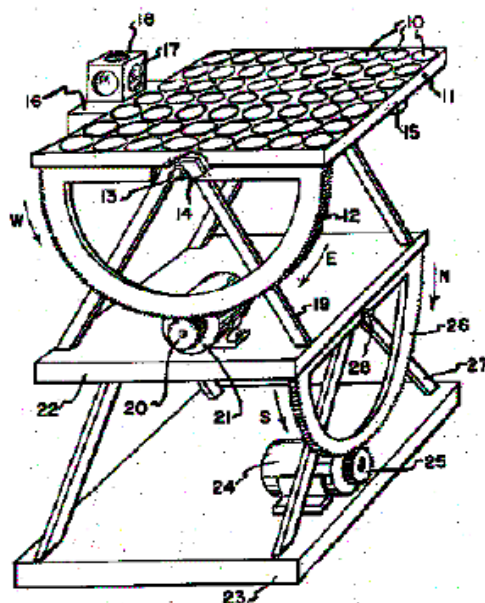


圖 2.8 東-西與南-北轉動雙軸追蹤器

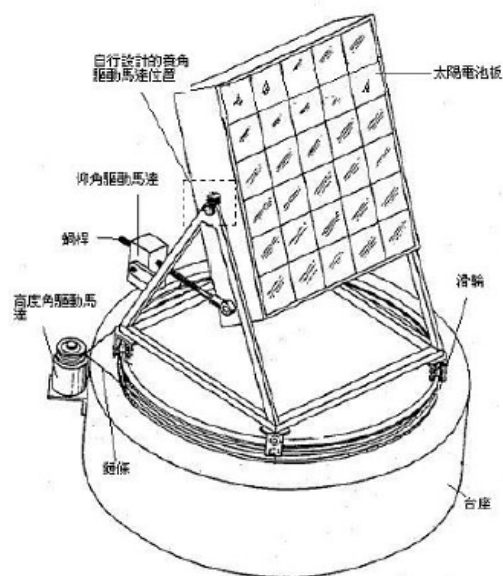


圖 2.9 仰角與方位角轉動雙軸追蹤器

三、主動式追日系統

根據參考文獻[9-10]，主動式追日系統的工作原理是利用光感測器來感測太陽的位置，光感測器可以量測目前方位是否正確，並啟動追蹤機構來修正，來達到追蹤的目的。主動式系統的目的就是要求任何時刻皆能接收最大的日照強度，而提升發電效率。目前已有的主動式系統設計，利用太陽能板兩端架設由光感元件所組成的光量感測裝置，透過感測迴授量的不同進行移動[9]，以及低耗能式主動型追日系統[11]。利用光譜分析原理所研發出主動型追日系統，有效拉長日照能源運用時間，配合聚光核心技術，有效提高轉換效率與降低成本。

四、被動式追日系統

被動式追日系統工作原理以計算太陽仰角及方位角角度，根據帶入的時間、經度及緯度等數據的不同，產生太陽追蹤器需旋轉及傾斜的度數，透過度數去驅動太陽能板來進行追日。



2.2. 影像處理

由於電腦運算速度及儲存技術不斷提升，大多數的研究都開始使用影像來取代傳統的超音波、紅外線偵測等感測器，彩色影像比起傳統的感測器可以捕捉到更多的特徵，且對於區域性分割能提供較為明顯的特徵，本研究將應用數位影像處理方法，分析經由所拍攝之彩色影像，使用主控電腦對彩色影像進行影像處理，再對經過影像處理後的影像進行分析，進而取得有用的資訊，作為太陽能板轉向的依據。

2.2.1. 影響數位拍攝因素

與傳統相機不同，隨著電子產業的進步，數位相機的元件更是多樣且功能亦是不斷地增強，不同產牌的數位相機的其拍攝能力各有不同的表現。其中影響拍攝能力最大因素的就是感光元件 CCD(Charge Coupled Device)，一般而言，CCD 的像素愈高，相機的解析度就愈高，影像的品質也愈好。

2.2.2. 感光度 and 解析度

ISO 感光度代表著傳統相機所用的底片對光線的敏感度，其數值愈大，代表感光程度愈強。為了統一計量單位，數位相機也沿用了這個標準，其 ISO 值愈低所需的曝光量愈高，反之 ISO 值愈高所需的曝光量愈低，在光線較暗的地方可以選用較高的感光度來拍攝。數位相機利用提高感光元件的訊號來增加感光度，但是無形中就提高了訊號中的雜訊，畫質變得粗糙。數位相機的解析度代表著所拍出的影像有多少的像素，以解析度 1600x1200（水平像素 x 垂直像素）為例，代表影像中有 192 萬個像素。

2.2.3. 影像擷取

全彩的 CCD Webcam 來擷取影像，所以在處理色彩之前，必須精確表達色彩，因此有色彩模型之產生，其目的為便於以一般可接受的方式來指定色彩。而所有的色彩皆可視為是三種原色所組成的，三原色分別為紅（R）、綠（G）、藍（B），透過此三原色即可組成人類日常生活所見之各種色彩，而 Webcam 的 CCD 也是將所擷取到之光線分別以三原色的方式來量化為數據，以作為數位化之訊息。

2.2.4. 顏色偵測與色彩空間

顏色偵測是一種簡易影像分割技術，常用於過濾複雜背景並定位出特定顏色範圍之目標物，而由攝影機所擷取得到之影像資訊在儲存像素的色彩時，是使用 RGB 色彩空間來表示一個像素的顏色，這種表示方法可用來傳達一個像素的色彩在紅、

綠、藍三原色中的相關性；但是，從 RGB 色彩空間表示法無法得知像素的明亮度（luminance）、飽和度（saturation）及色相（hue）為何，因為 RGB 色彩空間是屬於非線性的色彩系統，而 RGB 色彩空間的影像顏色很容易因光線強弱的影響產生顏色深淺的變化，也就是說想要抽離出光線對於色彩的影響是非常困難的，所以直接在 RGB 色彩空間做顏色偵測容易產生嚴重的誤差，難以使用在複雜的圖形演算，比較不適合直接進行影像處理。為了減少環境光線變化對顏色偵測的影響，採用對光源強弱較不靈敏的色彩空間，一般顏色偵測常用的色彩轉換空間有 NCC、YUV、YIQ、YCbCr、CIELIB、HSV 等色彩空間[12]。

一、NCC 色彩空間

NCC（Normalized Color Coordinates）色彩空間是解決 RGB 色彩空間中，影像因光源亮度的強弱，造成物體在相同顏色的地方呈現出深淺不同的顏色。分別對 R 與 G 做正規化，就可以使 R 與 G 對亮度的靈敏度降低，其轉換公式為(2-4)、(2-5)式。

$$g = \frac{G}{R+G+B} \quad (2-4)$$

$$r = \frac{R}{R+G+B} \quad (2-5)$$

因為 B 對亮度的靈敏度較小，所以可以忽略。利用 R 與 G 正規化得到 r 與 g，對影像做適當的門檻值選取，就能將特定顏色從影像中分離出來。

二、YUV 色彩空間

YUV 色彩空間是被歐洲電視系統所採用的一種顏色編碼方法（PAL）。在現代彩色電視系統中，通常採用三管彩色攝影機或彩色 CCD 攝影機進行取像，然後把取得的彩色圖像信號經分色、分別放大校正後得到 RGB，再經過矩陣變換電路得到亮度信號 Y 和兩個色差信號 R-Y（即 U）、B-Y（即 V），最後發送端將亮度和色差三個信號分別進行編碼，用同一頻道發送出去。這種色彩的表示方法就是所謂的 YUV 色彩空間表示。

採用 YUV 色彩空間的重要性是它的亮度信號 Y 和色度信號 U、V 是分離的。如果只有 Y 信號分量而沒有 U、V 信號分量，那麼這樣表示的圖像就是黑白灰度圖像。彩色電視採用 YUV 空間正是為了用亮度信號 Y 解決彩色電視機與黑白電視機的相容問題，使黑白電視機也能接收彩色電視信號。

YUV 主要用於優化彩色視頻信號的傳輸，使其向後相容老式黑白電視。與 RGB 視頻信號傳輸相比，它最大的優點在於只需佔用極少的頻寬（RGB 要求三個獨立的

視頻信號同時傳輸)。其中“Y”表示明亮度 (Luminance 或 Luma)，也就是灰階值；而“U”和“V”表示的則是色度 (Chrominance 或 Chroma)，作用是描述影像色彩及飽和度，用於指定像素的顏色。YUV 與 RGB 色彩空間的轉換如 (2-6) 式：

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.14713 & -0.2886 & 0.436 \\ 0.615 & -0.51498 & -0.10001 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2-6)$$

三、YIQ 色彩空間

YIQ 色彩空間，主要為彩色電視系統所採用如 NTSC 系統，Y 向量提供了黑白電視機所需要的顏色資訊，Y 則代表是色彩的明亮度，I 則是表示 In-phase，Q 表示 Quadrature-phase，Q 信號的相位角較 I 信號多 90 度。在彩色電視系統裡，彩色信號是由一向量系統來表示，向量的大小用來表示色彩的 Saturation，相位則是表示色彩的 Hue，I 和 Q 分別為同相位和正交的色彩向量。

YIQ 主要特點是 Y 和 I、Q 是具分離性的，簡單的說，有一個基準的彩色訊號，任何進來的彩色訊號只要與這個基準的彩色訊號相比，就可以知道這個訊號的大小與相位，也就知道這訊號所代表的顏色，所以數位影像中的亮度能單獨處理，不受色彩的干擾。YIQ 與 RGB 色彩空間的轉換如 (2-7) 式：

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2-7)$$

四、YCbCr 色彩空間

YCbCr 色彩空間用於數位視訊中，它是國家無線電諮詢委員會 (Consultative Committee of International Radio, CCIR) 所制訂 MPEG 視訊與 JPEG 影像的標準色彩空間，是由 YUV 色彩系統所開發改良的另外一種色彩空間，YCbCr 最常用於影像資訊壓縮。YCbCr 色彩空間分別代表亮度元素 (Luminance) Y、藍色元素 (Blueness) Cb 成份是藍色成份與參考值的差距，紅色元素 (Redness) Cr 成份是紅色成份與參考值的差距。YCbCr 與 RGB 色彩空間的轉換如 (2-8) 式：

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.1687 & -0.3313 & 0.5 \\ 0.5 & -0.4187 & -0.0813 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} \quad (2-8)$$

五、CIELIB 色彩空間

CIELIB 色彩空間，如圖 2.10 所示，是 1976 年 CIE 國際照明委員會所制定出來的色彩空間，CIELAB 色彩空間具有視覺上的均勻性，也就是說值的變化量和視覺

上感知到的變化量是相同的，例如：值從 100 變為 110 與值從 0 變為 10 的感覺是相同的，這種表示方法較接近人類視覺系統，常用於色彩設計及複製等行業，CIELAB 空間已被普遍使用在彩色電視系統上。CIELAB 色彩空間中的 L 表示顏色亮度，A 指顏色的紅-綠值，B 指顏色的黃-藍值。

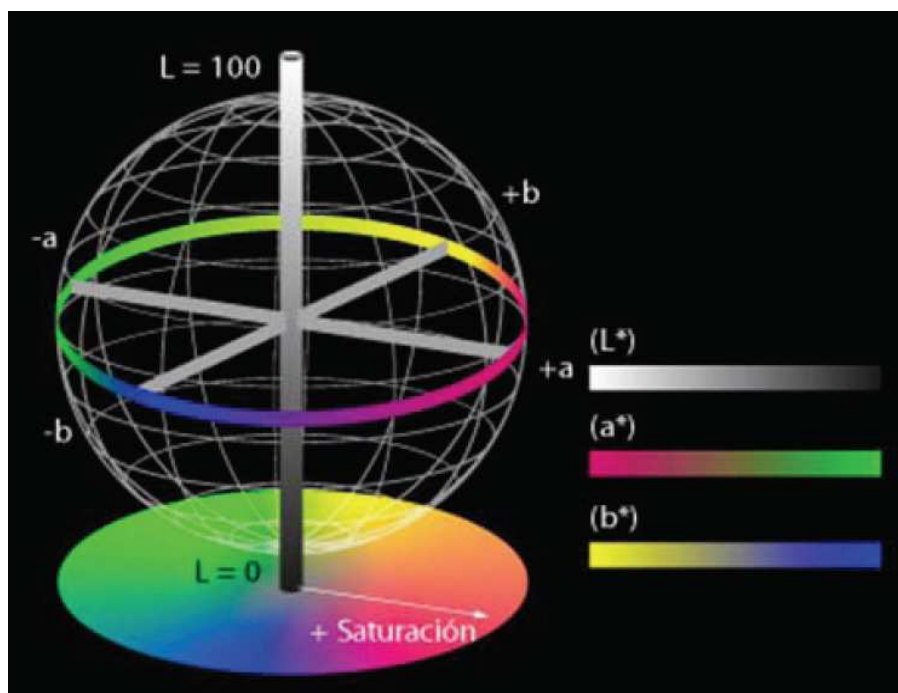


圖 2.10 CIELAB 色彩空間示意圖

六、HSV 色彩空間

HSV 色彩空間，是一個非線性轉換的色彩空間，其中的色彩關係與人類視覺感官較為相似，所以常被用來當作顏色在特徵擷取上的色彩空間。HSV 是將 RGB 轉換成為色調 (Hue)、飽和度 (Saturation)、亮度 (Value) 所組成，如圖 2.11 所示，色相 (H) 是色彩的基本屬性，就是平常所說的顏色名稱，如綠色、紫色等；飽和度 (S) 是指色彩的純度，越高色彩越純，愈低則逐漸變灰，取 0-100% 的數值；亮度 (V) 也叫“明度”，取 0-100%。色調又稱為色相，飽和度又稱為彩度，亮度是指色彩的明暗程度，色調和飽和度合稱為色度。

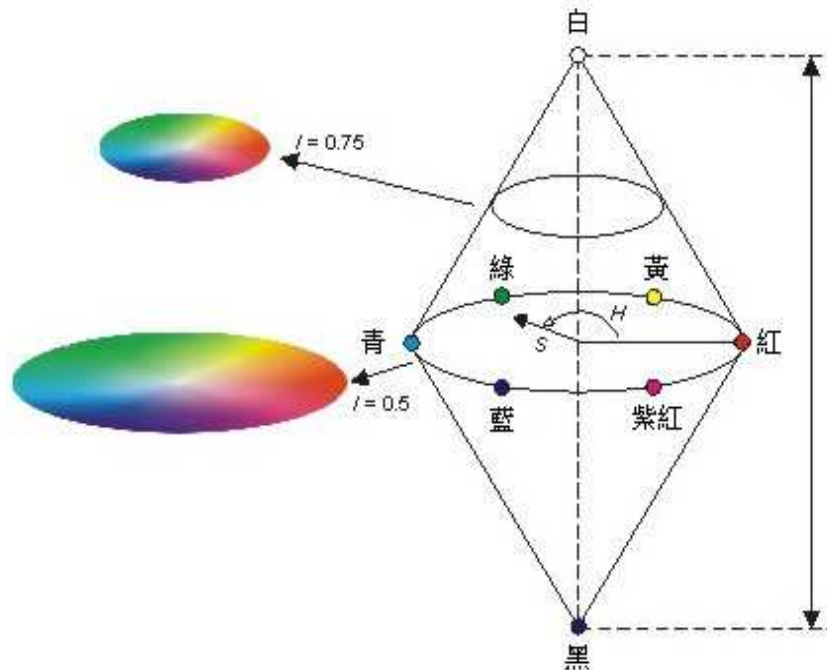


圖 2.11 HSV 色彩空間示意圖一

一般說來，色度和亮度是相互獨立的。在不同的光照條件下，雖然物體顏色的亮度會產生很大的差異，但是它的色度具有一致性，基本保持不變。HSV 與 RGB 色彩空間的轉換如 (2-9) 式、(2-10) 式、(2-11) 式、(2-12) 式、(2-13) 式：

$$\text{MAX} = \max(R, G, B) \quad (2-9)$$

$$\text{MIN} = \min(R, G, B) \quad (2-10)$$

$$H = \begin{cases} 60 \times \frac{G-B}{\text{MAX}-\text{MIN}} + 0, & \text{if MAX} = R, G \geq B \\ 60 \times \frac{G-B}{\text{MAX}-\text{MIN}} + 360, & \text{if MAX} = R, G < B \\ 60 \times \frac{B-R}{\text{MAX}-\text{MIN}} + 120, & \text{if MAX} = G \\ 60 \times \frac{R-G}{\text{MAX}-\text{MIN}} + 240, & \text{if MAX} = B \end{cases} \quad (2-11)$$

$$S = \frac{\text{MAX}-\text{MIN}}{\text{MAX}} \quad (2-12)$$

$$V = \text{MAX} \quad (2-13)$$

此 HSV 色彩空間示意圖二，如圖 2.12 所示：

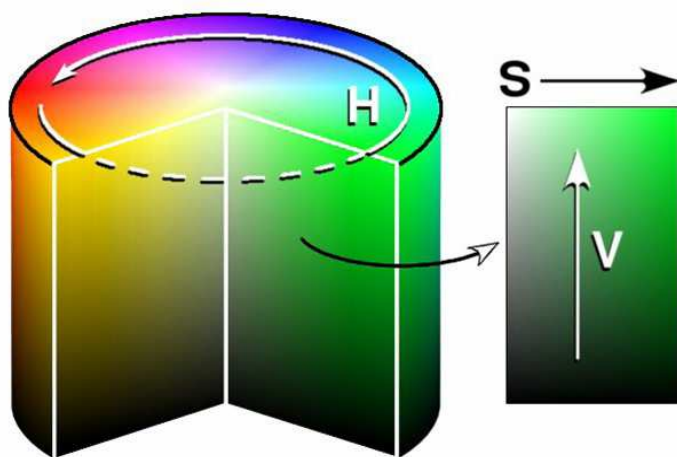


圖 2.12 HSV 色彩空間示意圖二

2.2.5. 影像二值化

影像二值化是將灰階影像透過所設定的門檻值，將影像像素值轉換成 0 或 255 的值。影像二值化是為了將檢測物體與背景資訊作分割，可提高影像處理的速度與記憶體的空間。假設影像資訊為一個二維矩陣 $f(x,y)$ ，且選取之門檻值為 \emptyset ，經過門檻值運算的結果， $f(x,y)$ 值轉換如(2-14)式

$$f(x,y) = \begin{cases} 255 & \text{if } f(x,y) \geq \emptyset \\ 0 & \text{otherwise} \end{cases} \quad (2-14)$$

其 $f(x,y)=0$ 表示像素轉換為黑色，反之為白色。門檻值運算主要是為了從背景影像中擷取物體資訊，門檻值數值設定太高，可能造成物體數量減少。設定值過低，可能會產生背景雜訊，使得影像無法正確判斷。

3. 追日系統原理與架構

本研究係針對攝影機之太陽能發電的應用，設計利用影像處理，並控制雙軸追蹤機構使太陽光與太陽能電池板保持垂直，以提昇太陽能電池發電功率為目的。本章說明的內容依序為追控系統原理、追蹤機構設計與製作、感測器設計與製作。

3.1. 追控系統原理

研究所設計的雙軸式太陽能追日系統，採前方設置一片濾光片過濾多餘及較弱光源的 CCD 攝影鏡頭作為感測器，結合影像處理技術將所得太陽光源影像由電腦進行影像處理，計算影像座標差，透過串列通訊經 RS-232 控制電路將資料傳送至 AI 馬達來控制雙軸追蹤機構，以移動太陽能板至所需位置，定時執行動作達到動態修正受光角度之目的，系統架構方塊如圖 3.1 所示。

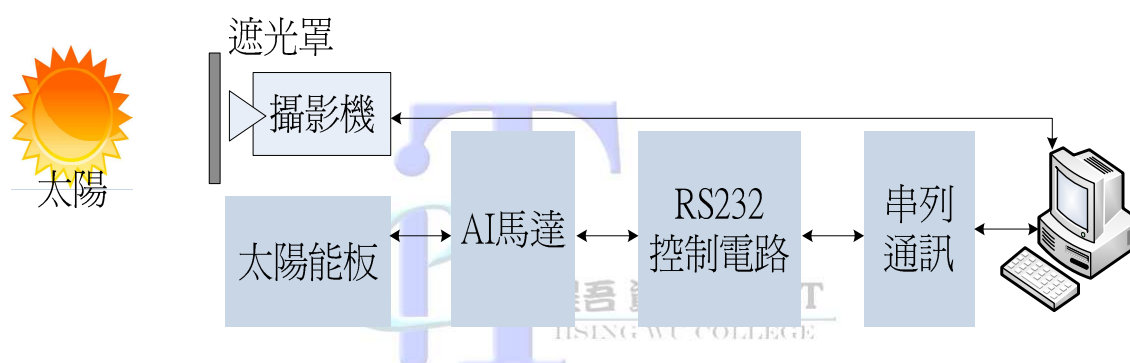


圖 3.1 系統架構方塊示意圖

3.2. 追蹤機構設計與製作

本系統模型機的追蹤機構設計主要是以「簡單」為前提，打造具有雙軸式移動之追蹤系統，由於追蹤機構是直接影響追蹤精確度的重要因素，因此，採用了可簡單控制的 AI 馬達作為基礎，分別使用兩顆 AI 馬達作為東西向（X 軸）與南北向（Y 軸）位置的控制，並設定為高解析度旋轉軸來驅動太陽能發電單元追蹤太陽運動，設計示意圖如圖 3.2 所示。

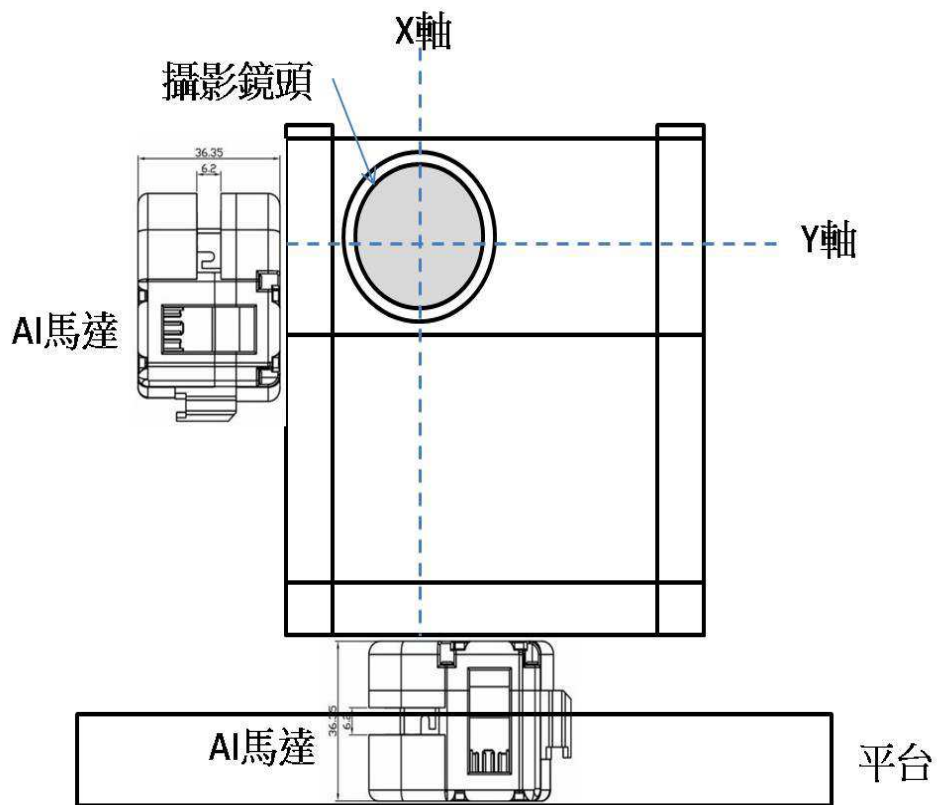


圖 3.2 AI 馬達雙軸式追蹤機構示意圖

3.2.1. AI 馬達

當 AI 馬達控制連接埠接收到 PWM 信號時會使馬達旋轉，馬達齒輪帶動了齒輪 1、齒輪 2、齒輪 3、齒輪 4，所有之連接桿都會接到齒輪 4 上的五角接頭及馬達軸接頭，可讓馬達跟馬達間連接做出更多自由度之動作，馬達分解圖如圖 3.3 所示。

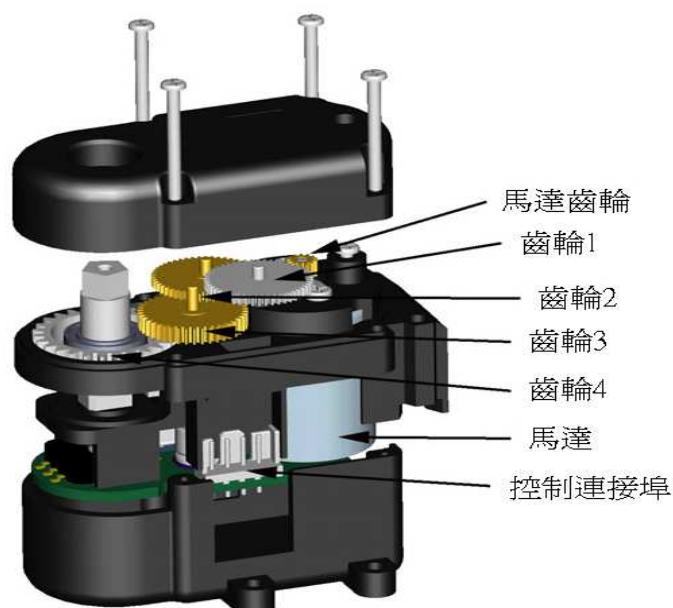


圖 3.3 AI 馬達之分解圖

本文 AI 馬達為 wCK-1108T1 如圖 3.4 所示，重量為 45g，尺寸大小為 51.6mm x 27.1mm x 36.4mm，最大功率都為 1.1W，最大扭力分別為 8kg-cm，工作電壓為 DC6V ~DC10V 之間，建議使用 DC7.4V~DC8.4V 為最佳，最大電流為 400mA~1800mA 當電流過載時馬達內部有逆電流保護裝置及反向電壓保護裝置耐壓為 0~28V 防止馬達損壞，使用 UART 串列通訊與 AI 馬達做溝通，並以串聯的方式與其他 AI 馬達做連結最多可達到 254 顆，傳輸速率 4,800bps~921,600bps 之間，規格如表 3.1 所示。

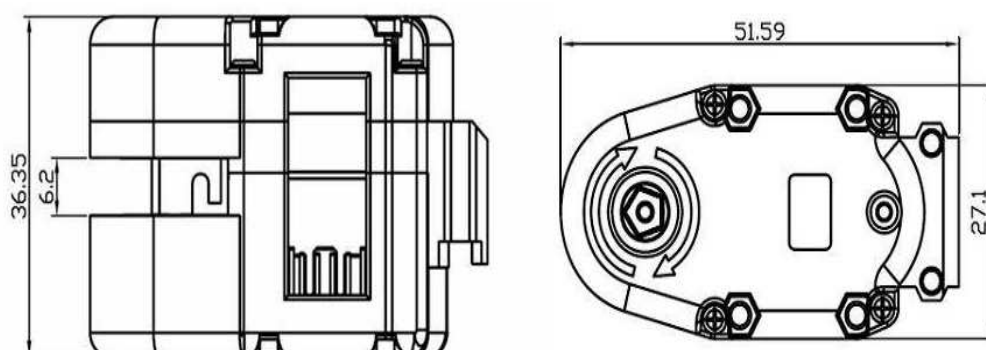


圖 3.4 AI 馬達為 wCK-1108T1 規格圖

表格 3.1

wCK-1108T1AI 馬達規格表

控制通訊方式	全雙功 UART 串列通訊
傳輸包率	4,800bps~921,600bps
馬達擴充數量	最大 254 顆 (ID0~253)
工作電壓	DC6V~DC10V (建議使用 DC7.4V~DC8.4V)
最大電流	400mA~1,800mA (內部有電流保護)
反向電壓保護	DC0V~DC-28V
最大功率	wCK-1108T1
速度模式	30 段速度 (位置模式), 16 段速度 (旋轉模式)
旋轉速度	無負載的最大速度為, 0.15 sec/60 度 (wCK1108T1)
最大扭力	8kgcm (wCK-1108T1)
控制模式	位置控制、速度控制、扭力控制
角度解析	0~254 (標準解析度), 0~1,022 (高解析度)
操作角度	0°~269° (標準解析度), 0°~333° (高解析度)
解析度	8 bit/1.055° (標準解析度), 10 bit/0.325° (高解析度)
裝置控制角度	269°/255=1.055° (標準解析度), 333°/1023=0.325° (高解析度)
誤差角度	±0.8° (8 bit 標準解析度)
P 增益	建議 1~120
D 增益	建議 0~254
I 增益	建議 0~10

資料來源：Intelligent ModularRobot wCK series User`s Manual

AI 馬達 wCK-1108T1 有旋轉模式與定址模式兩種動作，旋轉模式下 AI 馬達會不斷旋轉有 16 段速度，位置模式 AI 馬達有 30 段速度並可選擇高解析度或標準解析度，如圖 3.5 所示。高解析度下，AI 馬達可轉動範圍為 0~333 度，作 1022 格的解析（每格約 0.325 度），而在標準解析度下，AI 馬達可轉動範圍為 0~269 度，作 254 格的解析（每格約 1.055 度），馬達會回傳負載電流及位置資訊給 MPU 作回授控制。

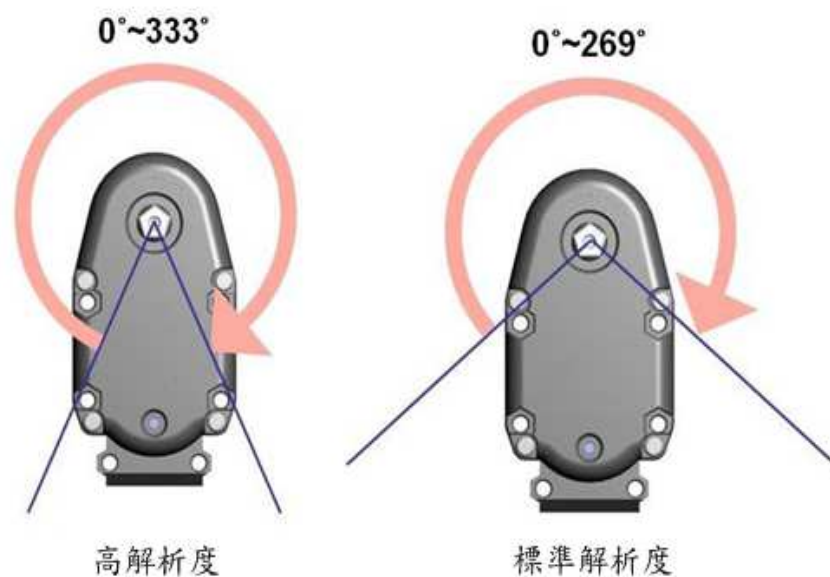


圖 3.5 AI 馬達解析度角度示意圖

AI 馬達控制腳位與連接軸，AI 馬達與 AI 馬達之間連結是用連桿連接五角接頭或馬達軸接頭，控制連接頭的腳位分別為 PIN 1 是 GND、PIN 2 是 TX、PIN 3 是 RX、PIN 4 是 Vcc，將這 4 隻接腳接在 MCS-51 或 RS232 連接主控電腦，對馬達下命令做動作，接腳功能如圖 3.6 所示。

醒吾 資應所 IAIT
HSING WU COLLEGE

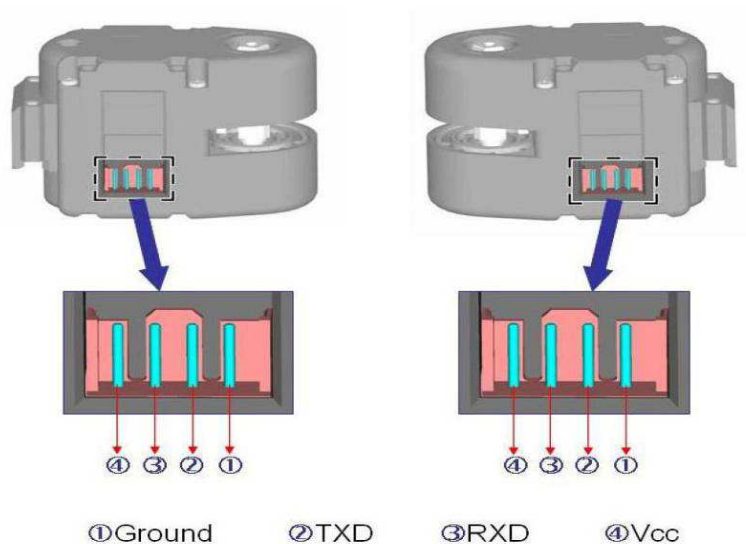


圖 3.6 AI 馬達控制腳位

3.2.2. AI 馬達 MPU 控制器

本系統所採用的 AI 馬達比直流馬達或步進馬達多了許多功能，尤其是內部多加一顆微控制器（MPU），使得 AI 馬達的應用更具有彈性，能夠發揮更大的功能。反向電壓保護裝置則是防止電源接反，變成反向電壓導致馬達燒壞，其結構如圖 3.7 所示。

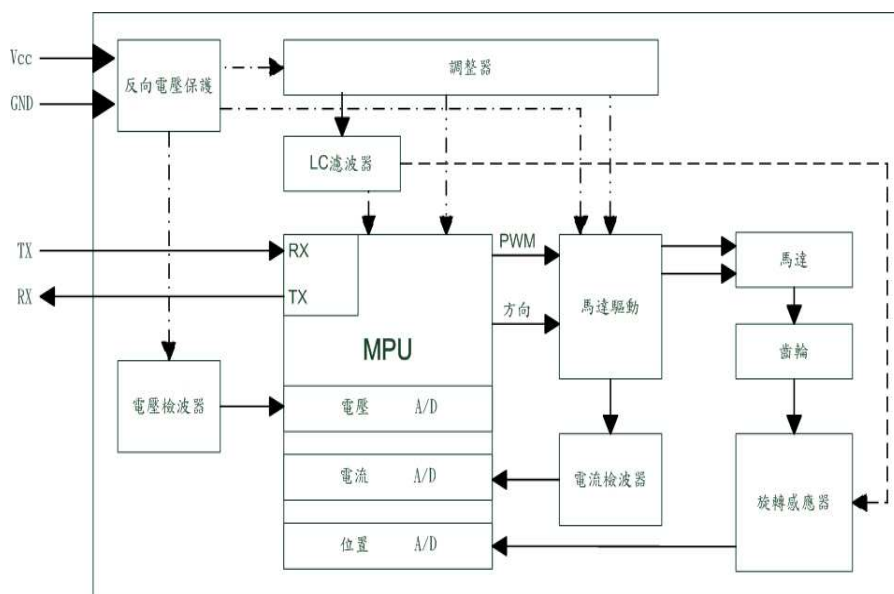


圖 3.7 AI 馬達之結構圖

MPU 主要是透過 TX 與 RX 與外部的 RS-232 作溝通，負責接收控制馬達的封包值（Command Packet），以及回傳反應封包碼（Response Packet），各馬達間以串接的方式連結，最多可串接 254 顆馬達，如圖 3.8 所示，在使用時必須先設定每顆馬達 ID 值，因此 ID 值須不相同才能正常動作，AI 馬達控制封包如 3.2 表所示。

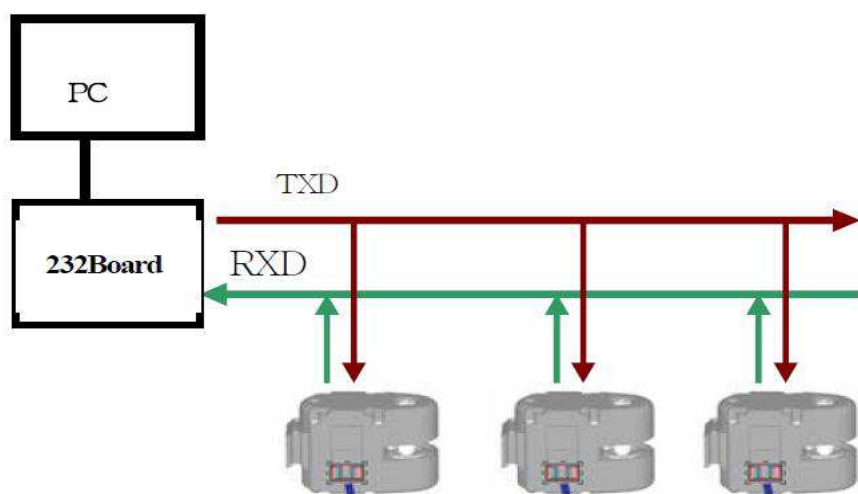


圖 3.8 電腦主控端控制多顆 AI 馬達

表格 3.2
馬達控制封包碼表

1byte	1byte	1byte	1byte				
Header 0xFF	Data1	Data2 0~254 (Target position)	Checksum (Data1 XOR Data2) AND 0x7F				
Torque = 0 (fastest) ~ 4 (slowest)			ID=0~30				
7	6	5	4	3	2	1	0

資料來源：Intelligent ModularRobot wCK series User`s Manual

Data1 為控制馬達的 ID 值及旋轉速度，Data2 為指定轉動到的位置。其中的目標位置 (Data2) 內容並非實際角度，必須經過計算才能把角度求出來。如圖 3.9 所示，左邊 AI 馬達把 Position128 設為實際角度 0 度，而右圖為 Position 213 設為實際角度 -90 度，Position m 轉換成實際角度 θ 如(3-1)式

$$(\theta-0):(-90-0)=(m-128):(213-128) \quad (3-1)$$

$$\theta=(128-m) \times (1.059)$$

由上式 m 代入 Position 213 求得 θ ，經計算後得到 $\theta \cong -90^{\circ}(-90.015^{\circ}-90.015^{\circ})$

與實際馬達所轉的角度差不多。

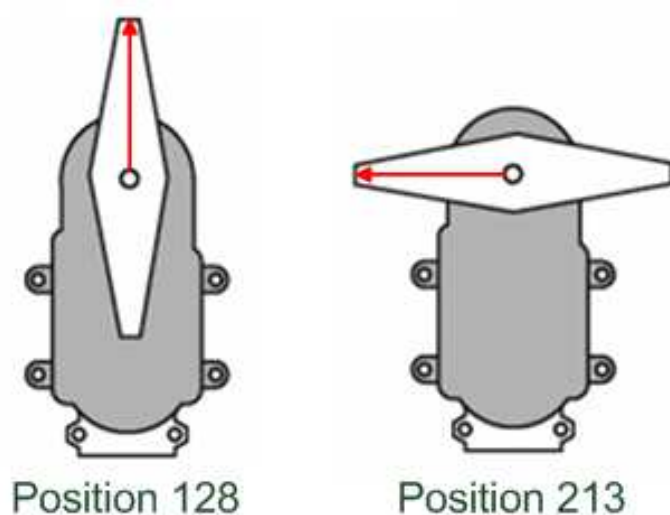


圖 3.9 馬達角度位置圖

3.2.3. AI 馬達設定及測試

首先執行 wCK Programmer 程式，選擇 BasicSetting，如圖 3.10 所示。

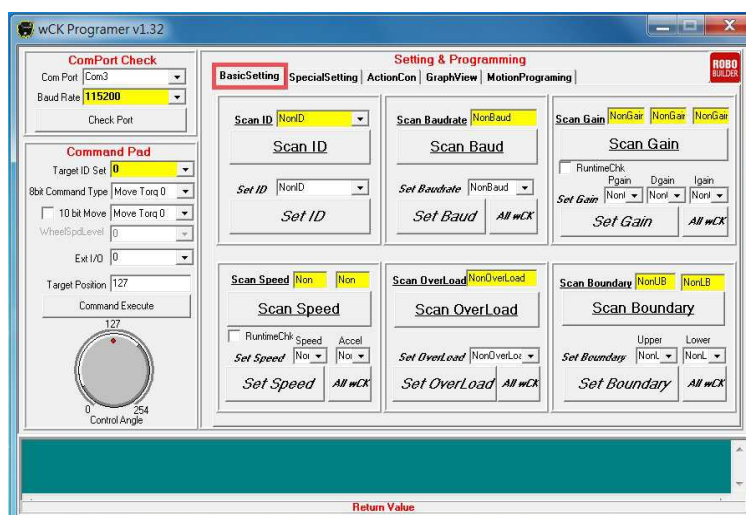


圖 3.10 wCK Programmer 程式操作說明圖一

在 ComPort Check 選擇連接端口（Com3），然後於 ScanBaud 按下掃描鮑率

(115200) 如圖 3.11 所示。

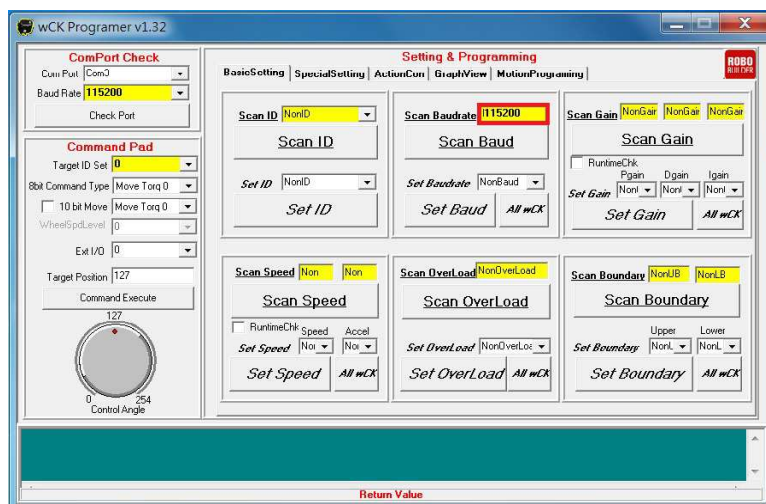


圖 3.11 wCK Programmer 程式操作說明圖二

當軟體不顯示速度，只顯示” Try Again”，則須重新連接 wCK 與 PC 間的電纜，再試一次。點擊” Scan ID”，即可得知 AI 馬達所配置的 ID，如圖 3.12 所示。

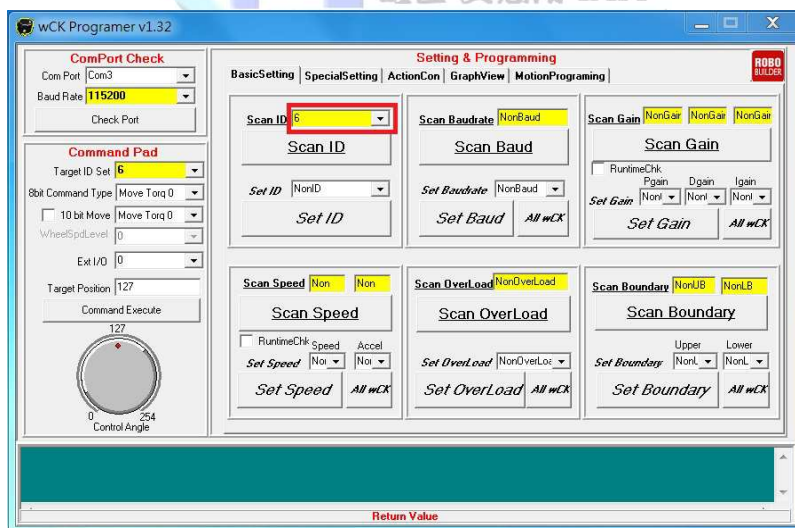


圖 3.12 wCK Programmer 程式操作說明圖三

最後利用 Control Angle 來測試 AI 馬達是否可正常控制，如圖 3.13 所示。

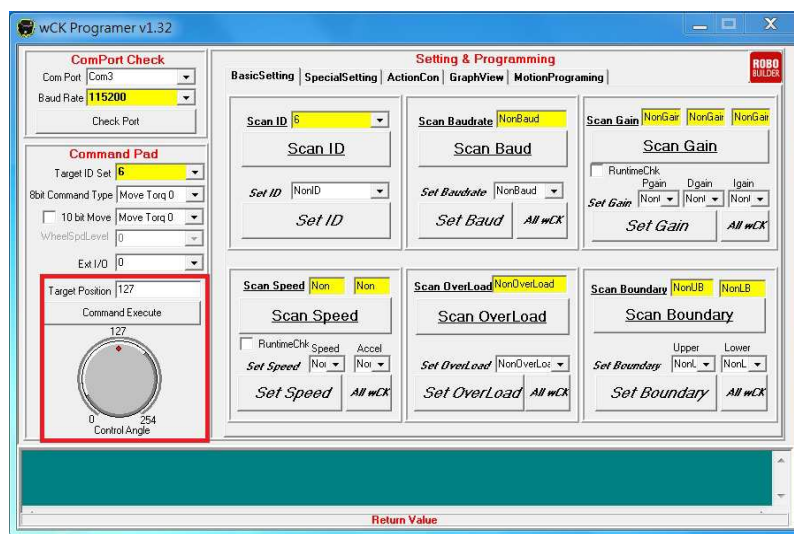


圖 3.13 wCK Programmer 程式操作說明圖四

3.3. 感測器設計與製作

數位攝影基本原理是利用鏡頭將影像投射到影像陣列（Image Array）上，數位相機的感應器，由此一電荷耦合元件，把相機鏡頭所採取到的影像，轉換成數位影像訊號，透過電子電路的信號處理，再把數位影像訊號儲存在儲存媒體中。

Webcam 則是再進一步在 CMOS 影像感應器後面，整合或加入一個影像信號處理晶片，接收來自 CMOS 影像感測器原始資料，並且執行不同的影像處理任務，如色彩更正、伽瑪更正、直方條等化等。並在最後做影像壓縮編碼成 MJPEG 或 MPEG 等影音格式，再經由 USB 介面傳輸至電腦。

Web camera（Webcam）網路攝影機有兩種，一種是利用網路直接將影像放置在於網路上的網路攝影機（又名 IP camera），另一種則需接在電腦上，透過軟體和驅動程式將影像放入電腦中處理。第二類（後面簡稱 Webcam）幾乎都是透過 USB 與電腦連接。這兩類產品大多屬於 CMOS 式攝影機。本系統則使用上述第二種透過 USB 與電腦連接之羅技 C310 網路攝影機做為太陽能追日系統的感測器，實際照片如圖 3.14 所示，規格表如表 3.3 所示。



圖 3.14 羅技 C310 網路攝影機實體圖

表格 3.3

羅技 C310 網路攝影機規格表

影像擷取	最高 1024 X 768 像素
視訊通話	640 X 480 像
羅技	Fluid Crystal™ 技術
照片	最高可達 5 百萬像素 (軟體增強處理)
建麥克風	
高速 USB 2.0	

資料來源：<http://www.logitech.com>(羅技網站)

3.3.1. 攝影機改造

太陽光直接照射亮度效能約有每瓦特 93 流明的輻射通量，其中包括紅外線、可見光和紫外線。明亮的陽光對地球表面上每平方米提供的照度大約是 100,000 勒克司或流明[13]。人眼可看見的可見光波長在 380-780nm 之間(780nm 以後的一部份範圍叫做紅外線，380nm 以前的一部份範圍叫做紫外線)，一般傳統底片設計也是在這個範圍內以對應人類視覺感應，即在可見光範圍內成像。如果相機要做紅外線攝影，必須使用紅外線專用的黑白底片。然而，現一般的 CCD/CMOS 本身就可以感應到紅外線，只是數位裝置為了降低雜訊，會在 CCD 或 CMOS 在出廠前都裝有排除紅外線之濾鏡(IR cut filter)，所以看不出紅外線效應，有些夜視功能的攝影機，就是在夜間拍攝時關掉 IR cut filter，利用紅外線照射來進行拍攝，即使在夜間眼睛

看不見時，仍可利用紅外線拍攝影像成像。

本系統是利用太陽光之紅外線作為偵測的範圍，因此，須將羅技 C310 網路攝影機的鏡頭拆下，將裡面的紅外線濾光器取出以進行紅外線攝影，為了讓攝影機可以捕捉到太陽所發出的紅外線而不被其他可見光所干擾，參考大同大學「多點觸控系統的設計與建置」專題報告中所說明的三種濾光作法，得知使用玻璃紙對 LED 燈的效果很好，但會有過多的可見光通過，而使用底片則會過濾太多的可見光，3.5 吋磁片過濾的效果最符合要求[14]。因此，系統將採用第三種作法，以 3.5 吋磁片作為攝影機的遮光罩，過濾不必要的可見光，如圖 3.15 及圖 3.16 所示。

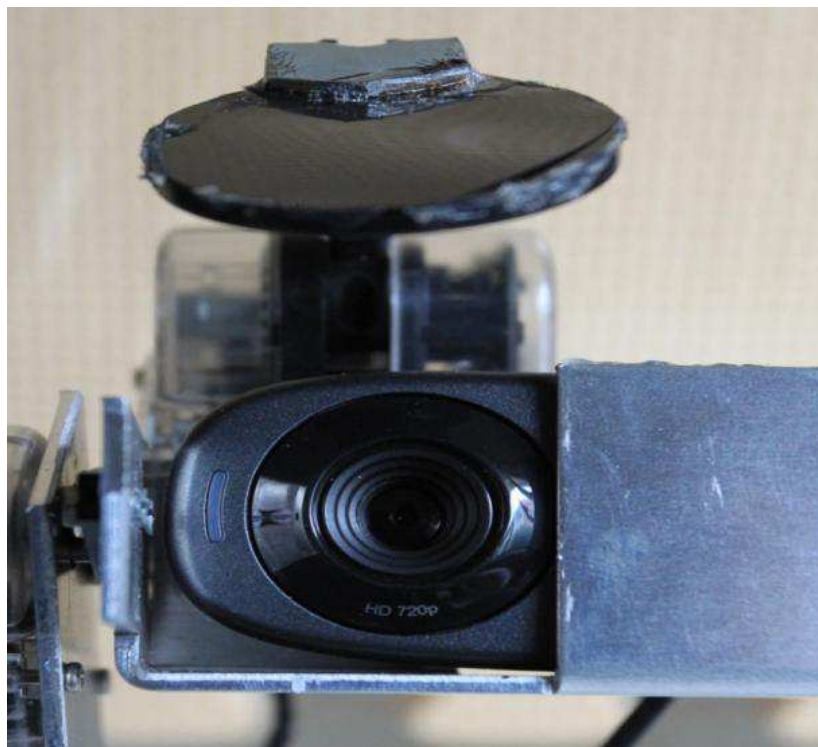


圖 3.15 感測器不使用遮光罩實際圖

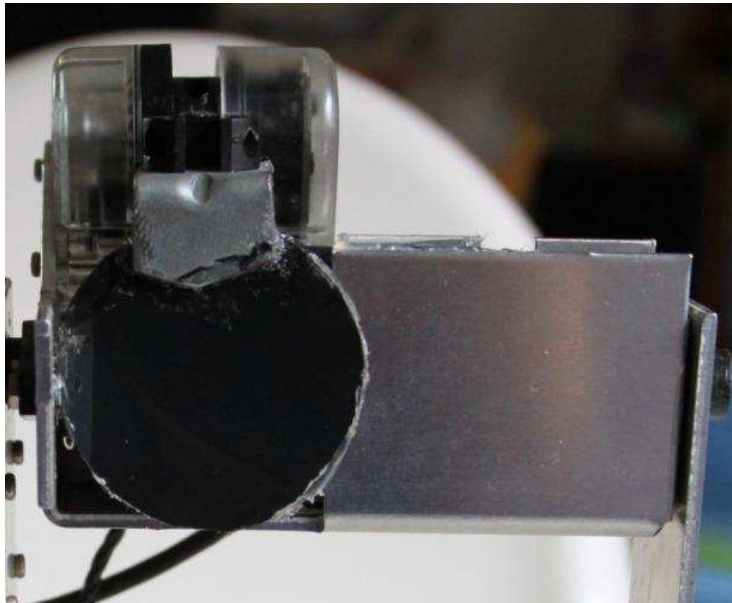


圖 3.16 感測器使用遮光罩實際圖

3.3.2. 攝影機測試

直接與個人電腦 USB 介面連接，在作業系統中開啟「我的電腦」，即可在「掃描器與數位相機」部分發現 USB 視訊裝置，點擊開啟後出現視訊畫面，表示裝置可正常運作，如圖 3.17 所示

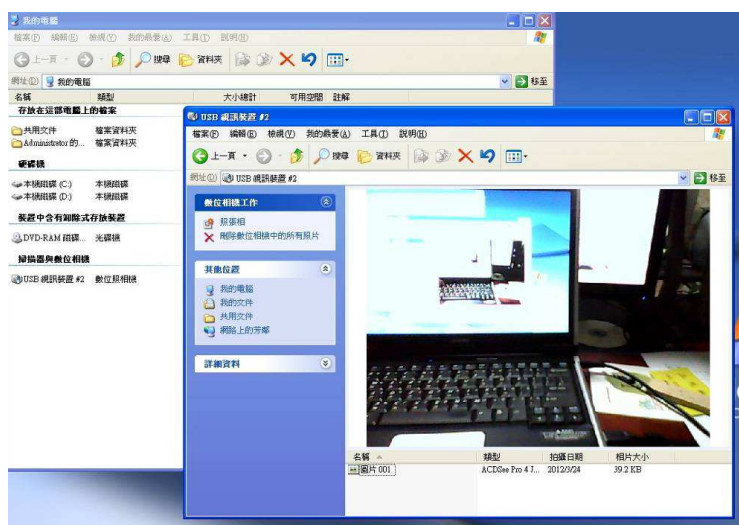


圖 3.17 攝影機實際測試圖

3.3.3. 影像門檻值

在計算機領域中，灰階（Gray scale）每個像素只有一個採樣顏色的圖像。這類圖像通常顯示為從最暗黑色到最亮的白色的灰階，灰階圖像經常是在單個電磁波頻譜如可見光內測量每個像素的亮度得到的，用於顯示的灰階圖像通常用每個採樣像素 8bits 的非線性尺度來保存，這樣可以有 256 種灰階(8bits 就是 2 的 8 次方=256)，本系統進行太陽光源影像灰階值(0-255)門檻值設定，如圖 3.18 所示，並依序設定灰階值為 100、150、180、220 及 240 來測試取得最佳的門檻值，結果發現灰階值為 100 時，因範圍太大，無法標示太陽所在區塊，如圖 3.19 所示；灰階值 150 時，已可標示太陽區塊，但範圍偏大，如圖 3.20 所示；灰階值 180，其範圍已逼近太陽光暈，如圖 3.21 所示；灰階值 220，範圍已進入太陽光暈內，如圖 3.22 所示；最後灰階值 240，已可正確標示出太陽光源，如圖 3.23 所示，惟使用此 240 值，因門檻太高，如遇太陽光源較弱時易出現無法辨識狀況，因此，經實驗將使用灰階值為 220，作為本研究的門檻值。

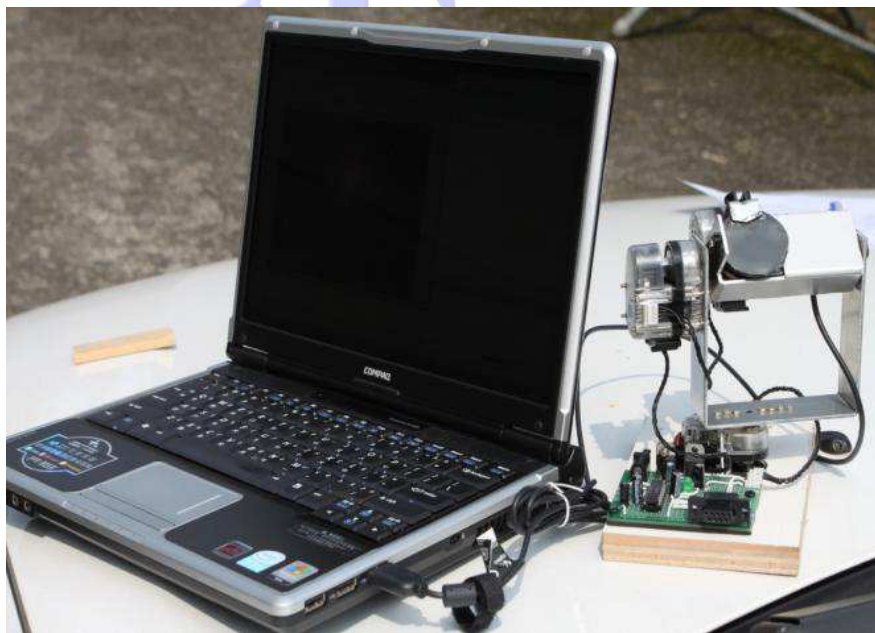


圖 3.18 系統測試實際設備圖

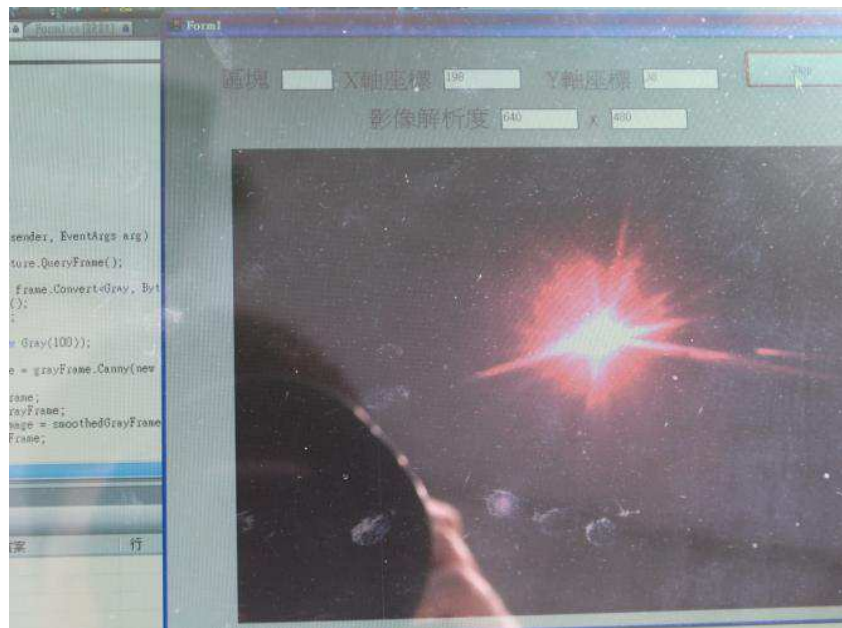


圖 3.19 灰階值 100 測試圖



圖 3.20 灰階值 150 測試圖

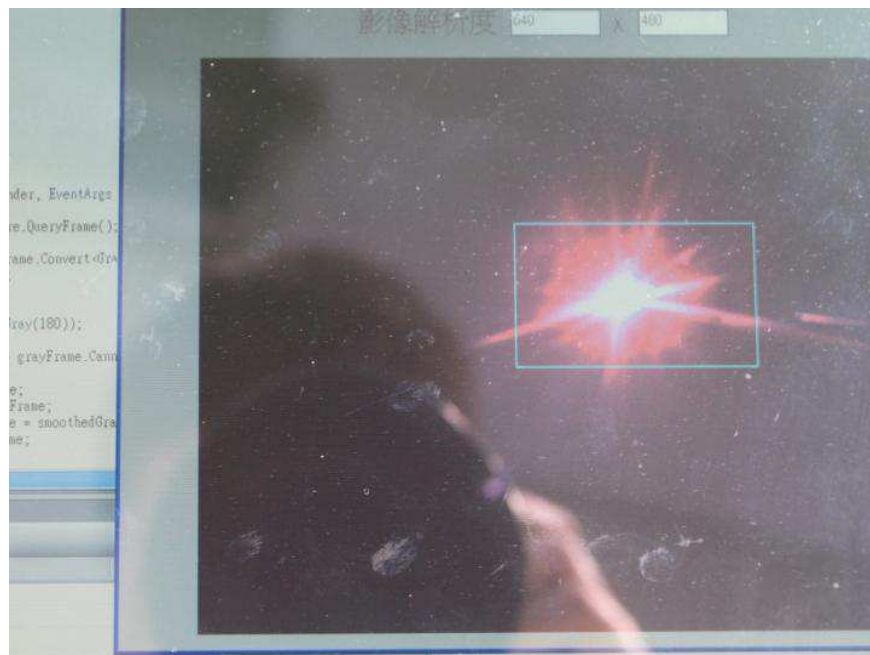


圖 3.21 灰階值 180 測試圖

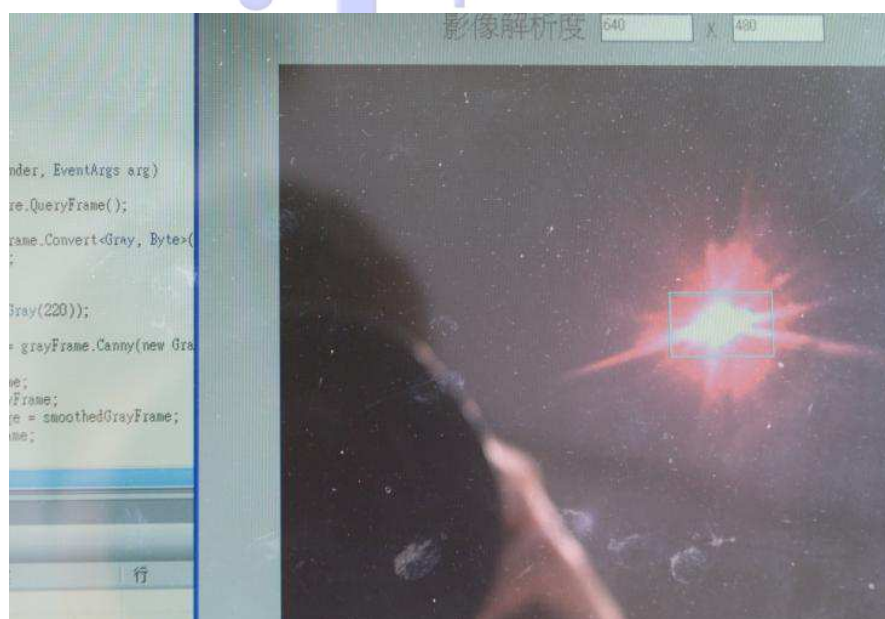


圖 3.22 灰階值 220 測試圖

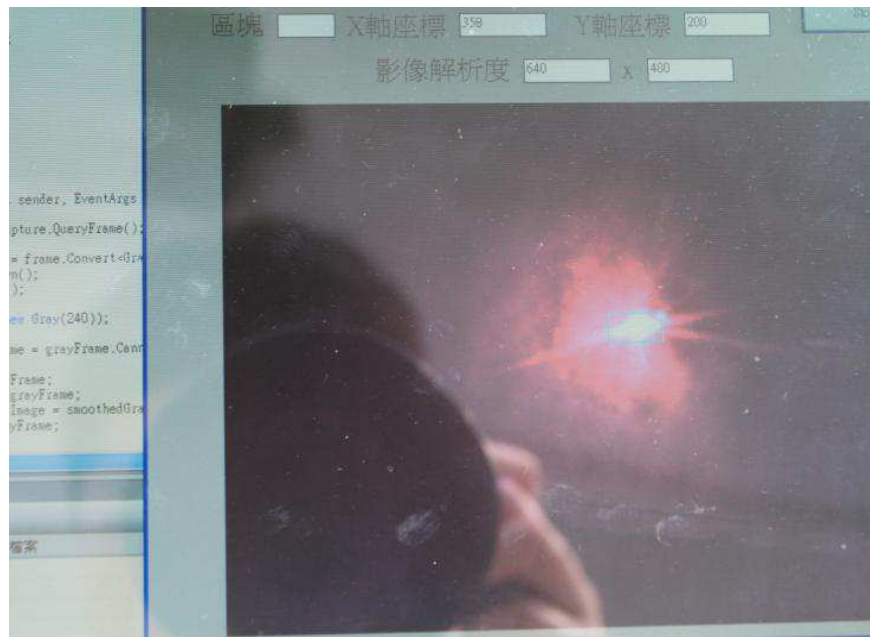


圖 3.23 灰階值 240 測試圖

3.3.4. 攝影機可視角測量

由於攝影機原廠並未提供該攝影機的可視角度，攝影機的可視角度影響本追日系統的可偵測範圍，為取得攝影機可視角度，利用直角三角形角度不會隨著三角形的大小而有所改變的性質，如圖 3.24 所示。

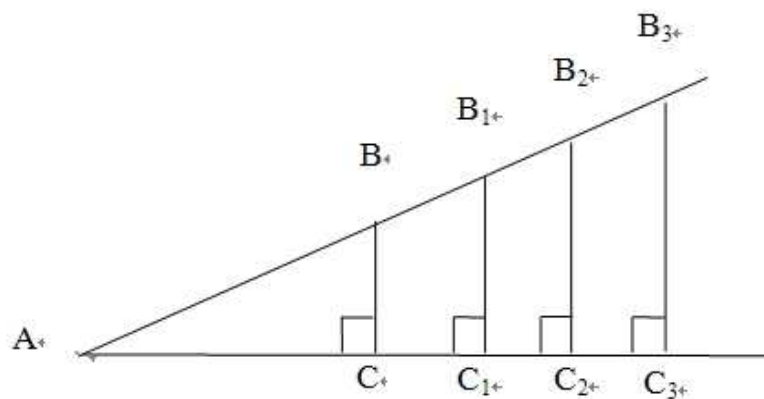


圖 3.24 直角三角形圖

在紙張劃上十字座標，並以每公分為一刻度，將攝影機分別置於 10 公分及 20 公分處進行拍攝，所得影像範圍利用求直角三角形夾角函數計算出攝影機之可視角度，方法如圖 3.25 所示，經由實際拍攝，攝影機分別置於 10 公分及 20 公分處得知可視範圍為 10 公分及 20 公分，如圖 3.26、3.27 所示。將所得範圍除以二，代入直角三角夾角公式，如 (3-2) 式。得知該攝影機寬可視角為 53.14 度，高可視角為 43.6 度因此，只要控制系統未超出此一範圍均可有效追蹤。

$$i = \tan^{-1}(W/L) \quad (3-2)$$

$$\text{寬} = \tan^{-1}(5/10) = 26.57^\circ$$

$$26.57^\circ \times 2 = 53.14^\circ$$

$$\text{高} = \tan^{-1}(4/10) = 21.8^\circ$$

$$21.8^\circ \times 2 = 43.6^\circ$$

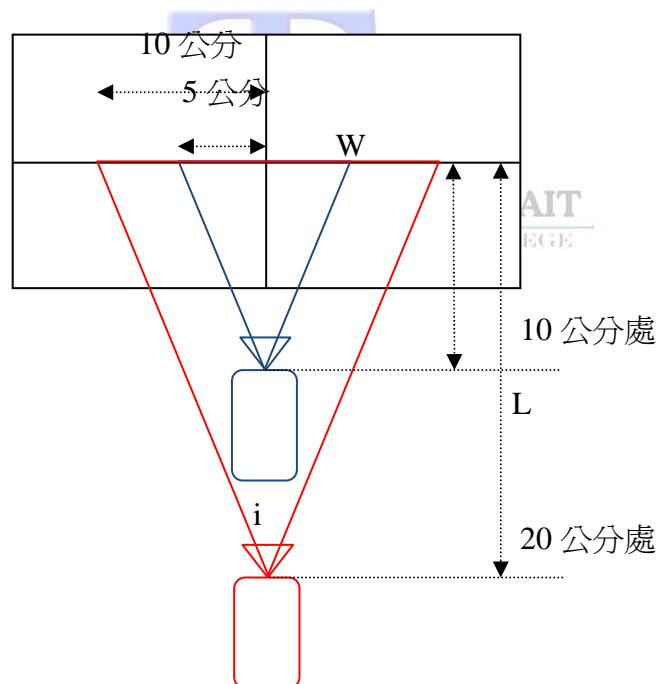


圖 3.25 測量方法示意圖

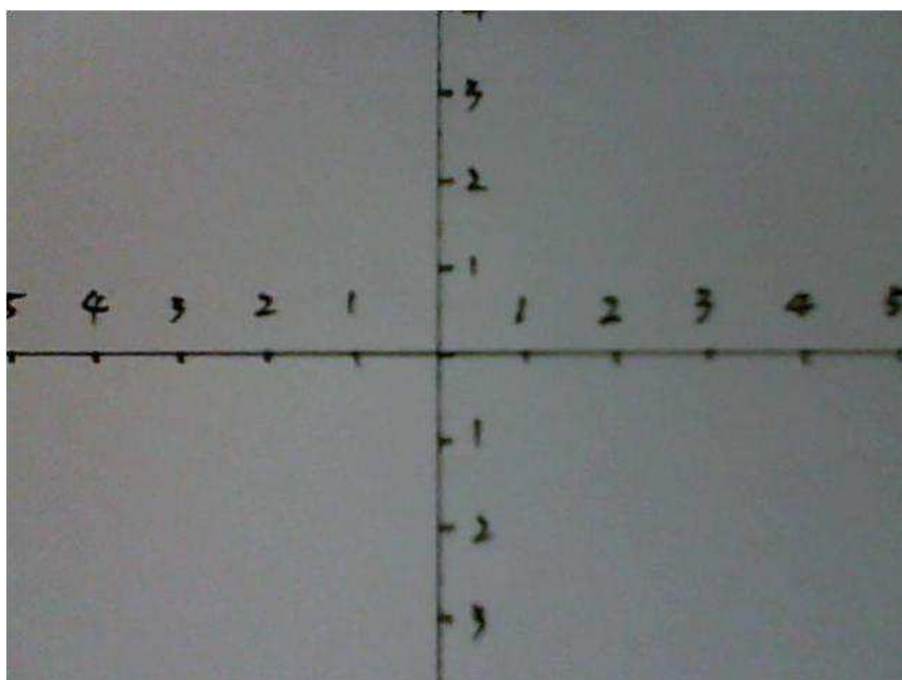


圖 3.26 10 公分實際拍攝圖

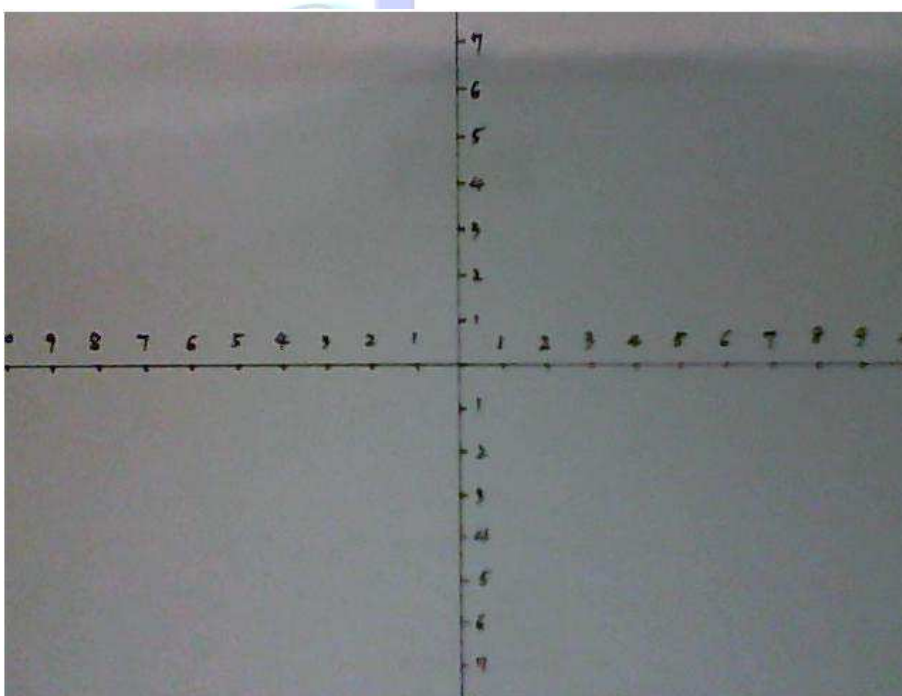


圖 3.27 20 公分實際拍攝圖

4. 系統整合與實驗結果

本研究所建立之系統包含硬體和軟體兩部份，以下將對硬體及軟體部份做詳細的介紹：硬體方面是以 PC 作為控制器，透過串列通訊經 RS-232 控制電路來控制 AI 馬達所組成的追蹤機構；軟體的方面則是所用 C#、OpenCV 及 AForge.NET 作為程式開發工具，配合自行撰寫的程式加以整合應用。

4.1. 硬體控制電路設計

4.1.1. RS232 簡介

RS-232 為一種 serial port standard，目前所普遍使用的為 RS-232-C，於 1969 年由 EIA（Electronic Industries Association）所制定。目前在 PC（IBM 相容或是 Mac）上尚有符合 RS-232-C standard 的 serial port 的裝置，因此就算在有其他 serial port 如 USB、IEEE1394 的今日，RS-232-C standard serial port 在工業上仍佔有重要的地位，腳位功能如表 4.1 所示。

表格 4.1
RS232 腳位功能表

腳位	簡寫	功能
Pin1	CD	載波偵測
Pin2	RXD	接收字元
Pin3	TXD	傳送字元
Pin4	DTR	資料端備妥
Pin5	GND	地線
Pin6	DSR	資料備妥
Pin7	RTS	要求傳送
Pin8	CTS	清除以傳送
Pin9	RI	響鈴偵測

資料來源：「單晶片實習-PIC 與電腦之非同步串列通訊」，洪永杰，2002，pp.2

除了標準的 25 pin 以外，還有 9 pin 及 10 pin 等其他規格的接頭，如圖 4.1 所示、圖 4.2 所示

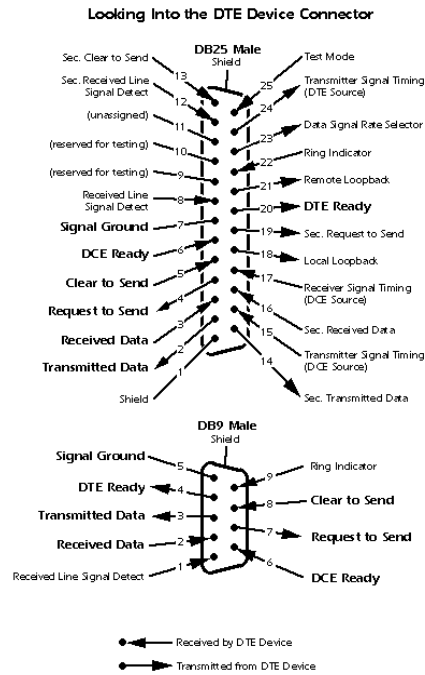


圖 4.1 RS232 25 及 9pin 公頭腳位功能圖

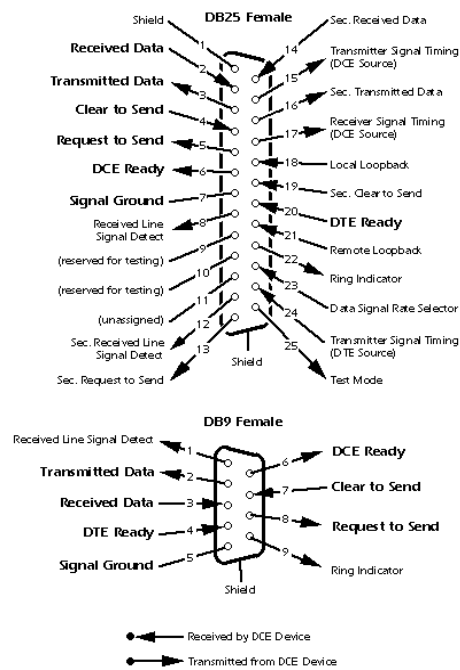


圖 4.2 RS232 25 及 9pin 母頭腳位功能圖

一、Data 傳輸格式：

RS-232-C standard 並沒有硬性規定 data 的傳輸格式（如幾個 bits, start/end bit 為何…等），但業界普遍使用一套規則，以 1 bit 的” 0”（低電位）作為 start bit，以 1bit 的” 1”（高電位）作為 stop bit。在兩者之間使用 5~8 bits 傳送 data，並加上 1~2 bits 作為 optional。如圖 4.3 所示：

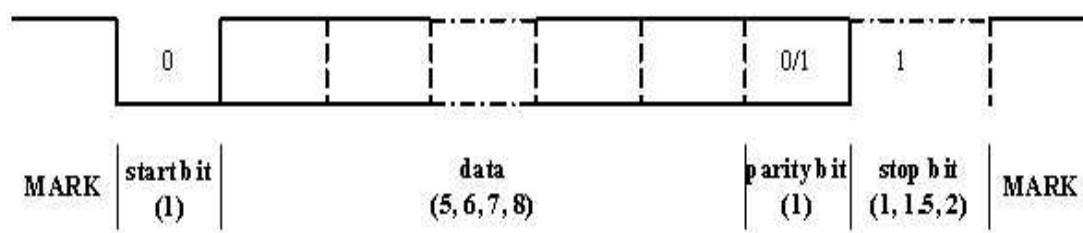


圖 4.3 Rs232 Data 傳輸示意圖

二、Baud Rate：

Baud rate（俗稱鮑率）為計算傳輸速率的單位，假設 baud rate 為 900，其背後的涵義為 900 bits/sec，或是傳輸 1 bit 需要 $1/900$ sec，在 RS-232-C standard 中並沒有硬性規定必須要使用多少的 baud rate。故許多 devices 都可以利用選擇訊號來設定所使用的 baud rate，選擇信號與 baud rate 的關係如表 4.2 所示：

表格 4.2

baud rate 關係表

選擇信號	鮑率 (bclk)	8 倍鮑率 (bclkx8)
000	38400	307200
001	19200	153600
010	9600	76800
011	4800	38400
100	2400	19200
101	1200	9600
110	600	4800
111	300	2400

資料來源：<http://www.csie.ntu.edu.tw/~b94108/2008dsd/RS232.htm>(台灣大學)

4.1.2. RS-232 控制電路

AI 馬達和電腦間是透過串列通訊埠 RS-232 的協定來溝通，由於電腦端 RS-232 的邏輯位準定義是 +5V ~ +15V 之間為 0，-5V ~ -15V 之間為 1，而 AI 馬達端是屬於電路 TTL 的位準，規格顯示邏輯位準之定義為 High (2.57V ~ 5.2V)，Low (-0.5V ~ 0.94V)，因此必須轉換電腦 RS-232 的位準與 TTL 的位準，使雙方能夠溝通，如圖 4.4 位準轉換示意圖所示。在此以 IC (MAX-232) 完成此需求，控制電路如圖 4.5 所示及實際完成如圖 4.6 所示，本系統控制電路及追控機構組合完成如圖 4.7 所示，全系統硬體完成如圖 4.8 所示。

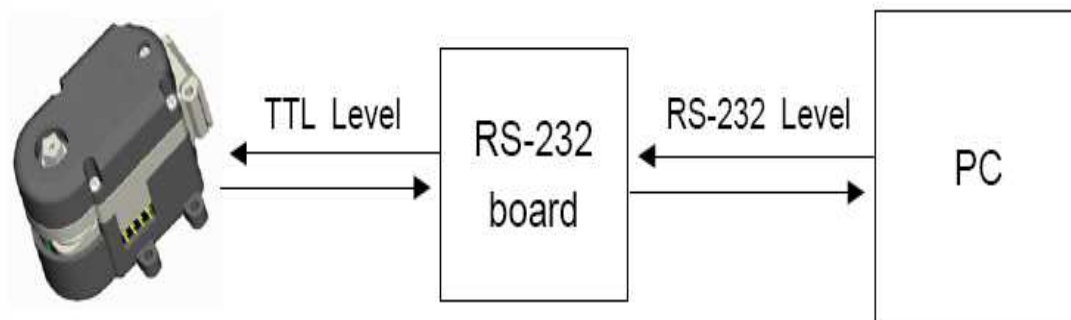


圖 4.4 位準轉換示意圖

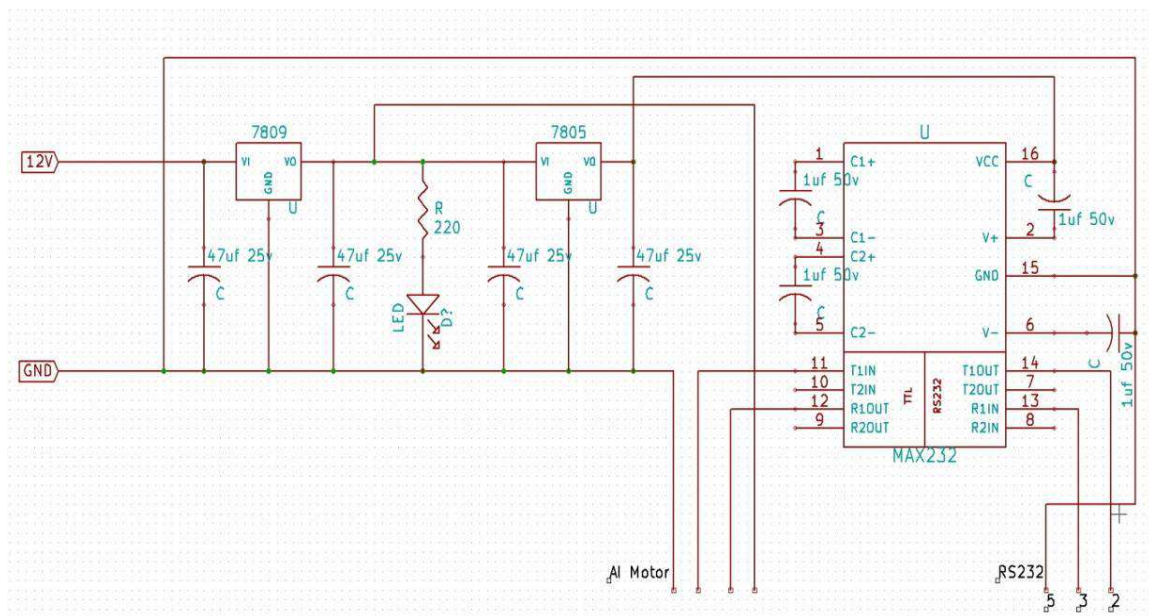


圖 4.5 RS-232 控制電路圖

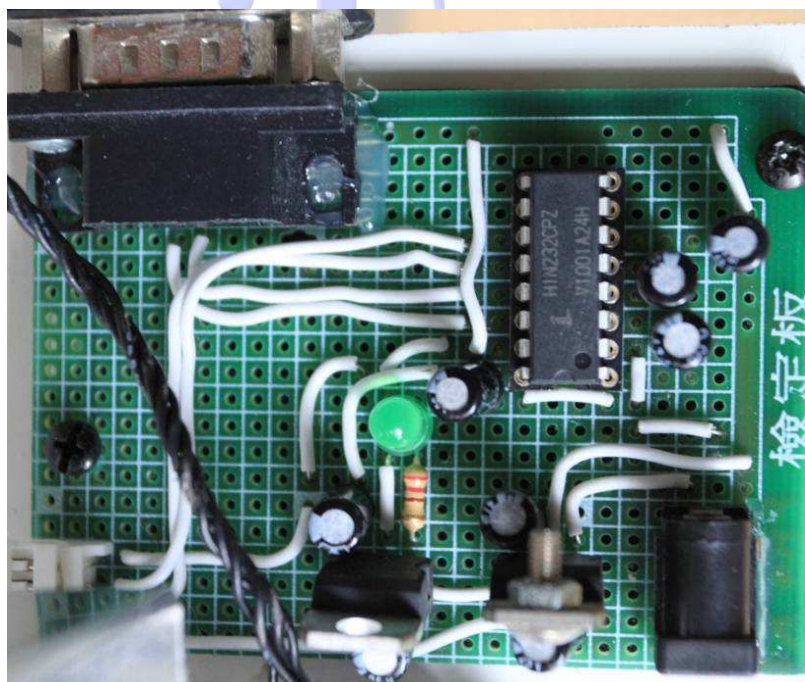


圖 4.6 RS-232 控制電路實際完成圖

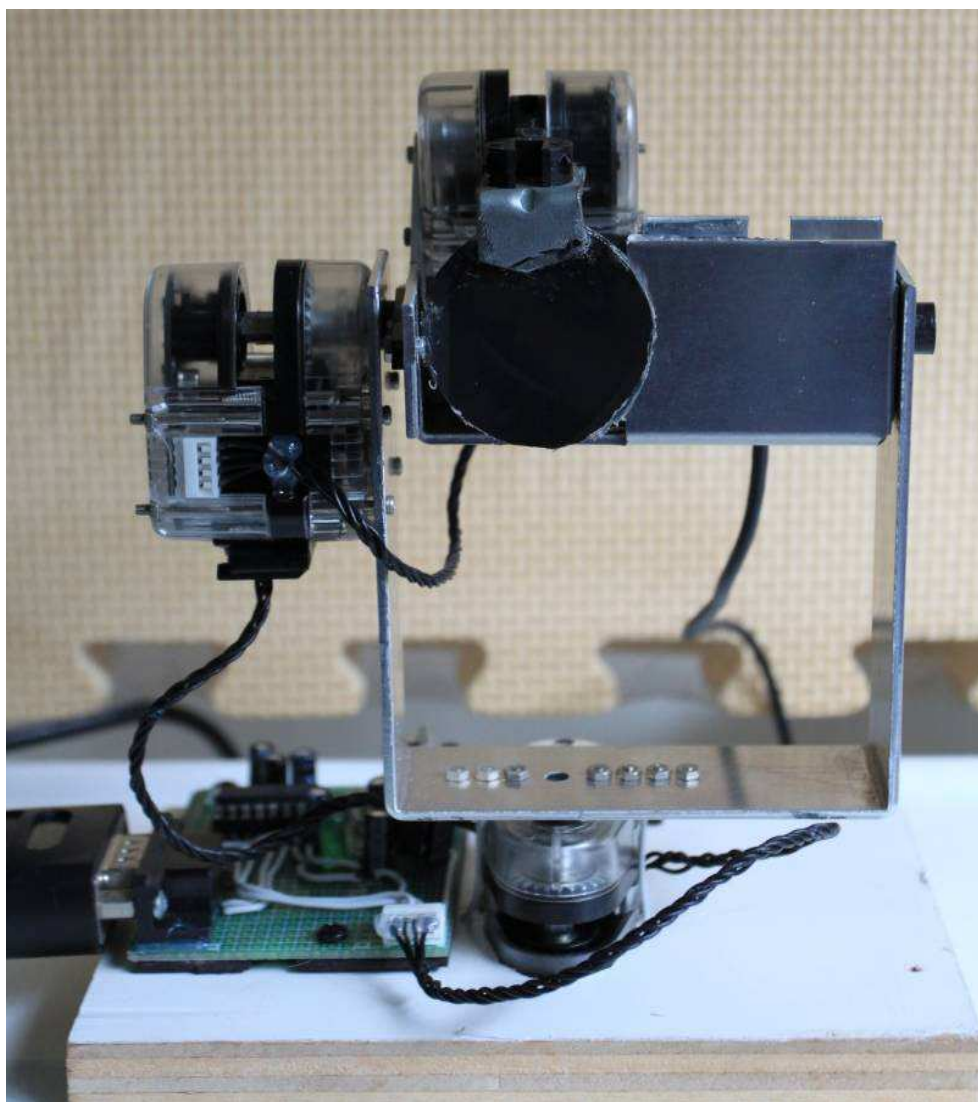


圖 4.7 追蹤機構與控制電路整合實際完成圖

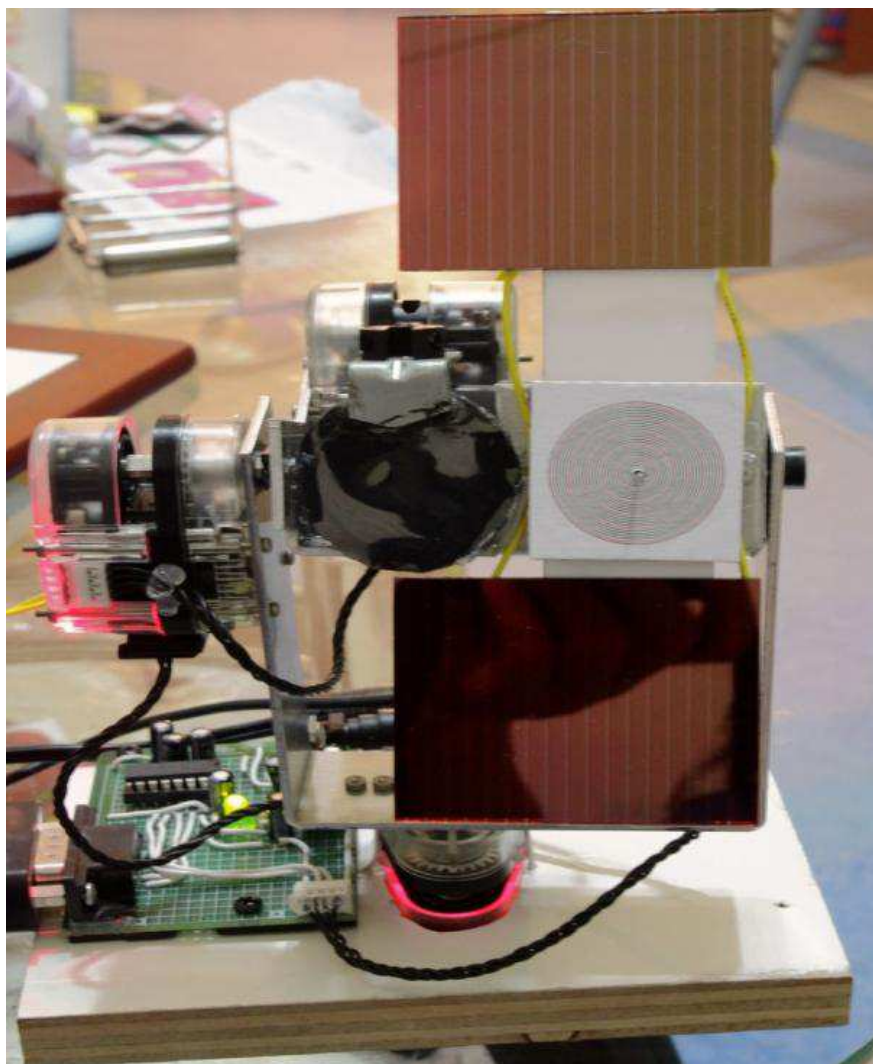


圖 4.8 全系統硬體實際完成圖

4.2. 軟體邏輯設計

4.2.1. C# 語言簡介

C#是一種簡潔、型別安全的物件導向語言，讓開發人員能夠建置一系列可以在 .NET Framework 上執行安全且強固的應用程式。C#可以用來建立傳統 Windows 用戶端應用程式、XML Web Service、分散式元件、主從式應用程式、資料庫應用程式及更多程式，其語法具有高度表達能力，但關鍵字卻少於 90 個，是相當簡單並容易學習的語法，也提供強大的功能，例如可為 Null 的型別（Nullable Type）、列舉型別（Enumeration）、委派、匿名方法（AnonymousMethod）及直接記憶體存取。C#也支援泛型方法和型別（會提供增強的型別安全和效能）以及 Iterator（可讓集合類別的實作器定義自訂反覆運算行為，由用戶端程式碼輕鬆運用）。做為物件導向語言，C#支援封裝（Encapsulation）、繼承（Inheritance）和多型（Polymorphism）的概念。所有的變數和方法，包括 Main 方法，也就是應用程式的進入點（Entry Point），都封裝在類別定義之內。類別可能直接從一個父類別繼承，不過可以實作任何數目的介面。覆寫父類別中之虛擬方法的方法，都需要用 override 關鍵字做為避免意外重新定義的方式。在 C#中，結構（Structure）就像輕量的類別；是一種能夠實作介面，卻不支援繼承的堆疊配置型別。除了這些基本的物件導向原則之外，C#還透過許多創新的語言建構來促進軟體元件的開發，包括：

- 一、委派的封裝方法簽名碼（Signature），可以啟用型別安全事件通知。
- 二、多種屬性（Property），做為私用成員變數的存取子。
- 三、多種屬性（Attribute），在執行階段提供關於型別的宣告式中繼資料。
- 四、內嵌（Inline）XML 文件註解。

如果需要與其它 Windows 軟體（例如 COM 物件或原生 Win32 DLL）互動，可以透過稱為 "Interop" 的執行緒在 C#中進行互動。Interop 使 C#程式幾乎可以執行所有原生 C++ 應用程式能夠執行的功能。C#甚至在直接記憶體存取為絕對關鍵因素的情況下，支援指標和 "Unsafe" 程式碼的概念。C#建置程序與 C 和 C++ 相較之下更為簡單，而且比 Java 更有彈性。由於沒有分隔的標頭檔（Header File），因此不需要以特定的順序宣告方法和型別。C#原始程式檔（Source File）可以定義任何數目的類別、結構、介面及事件。C#所撰寫的程式是運行於在 .NET Framework，屬於較特別的非機械碼。

4.2.2. .NET Framework 簡介

.NET Framework 為不可或缺的 Windows 元件，它可支援建置和執行下一代的應用程式和 XML Web Service。 .NET Framework 是專為實現以下目標所設計的：

- 一、提供一致的物件導向程式設計環境，允許程式在本機中儲存及執行或分散至網際網路或在遠端執行。
- 二、提供可減少軟體部署和版本控制衝突的程式碼執行環境。
- 三、提供加強程式碼安全執行的程式碼執行環境，包括未知或非完全信任之協力廠商所建立的程式碼。
- 四、提供可消除編寫指令碼或解譯環境效能問題的程式碼執行環境。

讓開發人員在使用各式各樣的應用程式時仍能體驗一致性，例如 Windows 架構的應用程式和 Web 架構的應用程式。業界標準建置所有通訊，以 .NET Framework 為基礎的程式碼能夠與其他程式碼整合。 .NET Framework 有兩個主要元件：執行階段 Common Language Runtime (CLR) 和 .NET Framework 類別庫。 Common Language Runtime 是 .NET Framework 的基礎。可以將執行階段視為在執行時間管理程式碼的代理程式，提供類似像記憶體管理、執行緒管理和遠端處理等核心服務，同時執行嚴格的型別安全 (TypeSafety) 以及加強安全性和強固性的其它形式的程式碼正確率。 .NET Framework 的另一個主要元件—類別庫，則是範圍廣泛、物件導向、可重複使用型別的集合，可用它來開發的應用程式，範圍從傳統命令列或圖形使用者介面 (GUI) 應用程式到以 ASP.NET 所提供最新創新方式為基礎的應用程式，例如 Web Form 和 XML Web Service，都包括在內。

4.2.3. OpenCV 簡介

OpenCV 是 Intel® 開源電腦視覺庫。它由一系列 C 函數和少量 C++ 類構成，實現了圖像處理和電腦視覺方面的很多通用演算法。 OpenCV 重要特性：

- 一、OpenCV 擁有包括 300 多個 C 函數的跨平臺的中、高層 API。它不依賴於其它的外部庫—也可以使用某些外部庫。
- 二、OpenCV 對非商業應用和商業應用都是免費的。
- 三、OpenCV 為 Intel® Integrated Performance Primitives (IPP) 提供了透明介面。這意味著如果有為特定處理器優化的 IPP 庫，OpenCV 將在運行時自動載入這些外部庫。

4.2.4. AForge.NET 簡介

AForge.Imaging，主要為幫助一些電腦計算機視覺任務，其包含不同影像處理函式：

- 一、顏色篩選器（灰度、深褐色反向、旋轉管道、通道提取、通道替換、通道篩選，顏色篩選級別線性，重新映射擴展，對比度的顏色的篩選，歐幾裡德顏色 長條圖均衡化，gamma 修正）。
- 二、HSL 篩選器（線性更正、亮度／對比度／飽和度更正、順化修飾符 HSL 篩選）。
- 三、YCbCr 篩選器（線性更正、YCbCr 篩選、通道（提取更換））。
- 四、binarization 篩選器（閾值，閾值與攜帶，burkes 抖動，Sierra 抖動，Ninke Stucki 抖動）。
- 五、Adaptive binarization（簡單圖像統計、反覆運算閾值、大津閾值）。
- 六、數學形態學篩選器（侵蝕，dilatation，關閉，hit-and-miss 細化 thickening）。
- 七、convolution 篩選器（表示、模糊、提高邊緣、Gaussian 模糊、自訂 convolution 篩選器）。
- 八、2 源篩選器（合併、相交、添加、減去、差異，邁向 morph）。
邊緣探測器（homogeneity, difference、sobel，canny）。
平面欄位照明改正。
- 七、pixellate，抖動、收縮、油繪畫畫布作物／填充／移動，Mirroring。
- 十一、中位數的篩選器、Adaptive 平滑、保守平滑。
- 十二、紋理篩選器（texturer、紋理篩選、紋理合併）。
- 十三、傅裡葉轉換（低通及高通篩選器）。
- 十四、blob 計數器，標籤篩選最大 blob 提取，Blob 的連接元件。
- 十五、調整大小和旋轉（最接近相鄰、雙線性，bicubic）。
- 十六、轉角探測器（莫拉韋克，Susan）。
- 十七、霍夫 Line 和 Circle 轉換。

十八、圖像顏色統計（RGB HSL，YCbCr）和垂直／水準統計（RGB）。

AForge.Video 函式庫提供對視頻資料的支援。並考慮到的影像處理 Framework 中的方法。

一、存取 JPEG 和 MJPEG 串流，使存取 IP 攝錄機。

二、透過 DirectShow 介面存取 USB Web 相機、擷取裝置和視頻檔。

三、使用音訊 for Windows 介面讀／寫 AVI 檔。

4.2.5. 系統介面設計

本系統使用一個 ImageBox 來擷取影像、三個 butan 分別啟動、重新校對及離開，另可顯示相關資料，如太陽影像座標、馬達目前角度、影像解析度、季節及現在時間等等，系統控制介面如圖 4.17 所示。程式碼如附錄。



圖 4.9 系統控制介面圖

4.3. 系統邏輯運作流程

首先本系統在開機時先取得系統時，判斷是否為系統運作時間，預設為 08-16 時，若不是在 08-16 時間內則判斷為夜晚，系統不運作；接著系統參照中央氣象局所提供之台灣四季仰角與方位角[15]如圖 4.18 所示，取得時間並驅動 AI 馬達移動太陽能電池板至太陽所在位置，達到系統初始位置。

台 北 （北緯25.03度，東經121.5度）															
季節	時	6	7	8	9	10	11	中天	12	13	14	15	16	17	18
夏至	仰角	11.3	24.1	37.3	50.7	64.2	77.8	88.5	87.8	74.7	61.1	47.6	34.2	21.1	08.4
	方位	069.1	074.1	078.6	082.9	087.4	094.4	180.0	226.4	267.8	273.7	278.1	282.4	287.0	292.1
春分	仰角	01.9	15.1	28.5	41.3	53.0	61.9	65.0	64.9	60.2	50.4	38.4	25.4	12.1	
秋分	方位	090.7	097.3	104.7	114.2	128.2	150.9	180.0	184.1	215.4	235.6	248.3	257.2	264.3	
冬至	仰角		04.2	15.6	25.9	34.3	39.8	41.5	41.4	38.8	32.6	23.6	13.1	01.7	
	方位		118.3	125.9	135.7	148.3	164.1	180.0	182.1	199.8	214.9	226.8	236.0	243.2	
台 中 （北緯24.15度，東經120.68度）															
季節	時	6	7	8	9	10	11	中天	12	13	14	15	16	17	18
夏至	仰角	10.3	23.1	36.4	49.8	63.4	77.1	89.3	88.9	75.4	61.8	48.2	34.7	21.5	08.7
	方位	068.7	073.5	077.7	081.6	085.4	090.0	180.0	232.4	269.3	275.1	278.9	282.8	287.0	292.0
春分	仰角	01.3	14.5	28.0	41.0	52.9	62.3	65.8	65.8	61.3	51.5	39.4	26.3	12.9	
秋分	方位	090.4	096.7	103.8	112.9	126.4	148.5	180.0	182.3	214.9	235.7	248.4	257.2	264.1	
冬至	仰角		04.0	15.5	26.0	34.6	40.4	42.3	42.3	39.9	33.7	24.8	14.1	02.6	
	方位		117.9	125.3	134.8	147.2	162.9	180.0	181.1	199.1	214.5	226.5	235.7	242.9	
台 南 （北緯23.00度，東經120.2度）															
季節	時	6	7	8	9	10	11	中天	12	13	14	15	16	17	18
夏至	仰角	09.4	22.4	35.7	49.2	62.9	76.6	89.5	89.4	75.8	62.1	48.4	34.9	21.6	08.7
	方位	068.3	072.9	076.8	080.1	083.0	085.0	0	218.7	264.9	277.1	280.0	283.4	287.3	292.0
春分	仰角	00.9	14.2	27.8	41.0	53.2	63.0	67.0	67.0	62.5	52.5	40.2	27.0	13.4	00.1
秋分	方位	090.2	096.2	102.9	111.7	124.6	146.4	180.0	181.2	215.3	236.4	249.0	257.5	264.2	269.8
冬至	仰角		04.1	15.8	26.5	35.3	41.4	43.5	43.5	41.1	34.9	25.9	15.1	03.5	
	方位		117.6	124.7	134.0	146.3	162.1	180.0	180.5	198.9	214.5	226.6	235.7	242.7	
恆 春 （北緯22.00度，東經120.73度）															
季節	時	6	7	8	9	10	11	中天	12	13	14	15	16	17	18
夏至	仰角	09.5	22.5	35.9	49.5	63.2	77.0	88.5	88.3	75.2	61.5	47.8	34.2	20.8	07.9
	方位	068.4	072.7	076.2	079.1	081.1	080.7	0	329.4	278.9	279.0	281.2	284.2	287.8	292.3
春分	仰角	01.3	14.8	28.5	41.8	54.2	64.1	68.0	68.0	63.0	52.6	40.1	26.7	13.1	
秋分	方位	090.4	096.1	102.7	111.2	124.0	146.3	180.0	182.6	217.4	238.0	250.1	258.3	264.7	
冬至	仰角		05.0	16.7	27.5	36.4	42.5	44.5	44.5	41.9	35.4	26.2	15.3	03.5	
	方位		117.8	124.8	134.0	146.4	162.4	180.0	181.2	199.8	215.4	227.3	236.2	243.0	

圖 4.10 台灣四季仰角與方位角

系統完成初始角度後，攝影機即取得太陽光源影像，並判斷光源影像與螢幕中心點之位置，由於本系統採用羅技 C310 攝影機，其影像解析度為 640X480，因此可以得知中心位置 X 軸為 320，Y 軸為 240，如圖 4.19 所示。由於攝影機所得影像與所顯示之影像方向相反，故所得光源若位於中心點左方，則移動 X 軸 AI 馬達向左移至 320 座標位置，反之則向右移動；若光源位於中心點上方則移動 Y 軸 AI 馬達向上移至 240 座標位置，反之則向下，如此不斷偵測光源位置及移控 X、Y 軸 AI 馬達，藉此轉動太陽能電池板，以達到動態修正太陽能電池板至最佳受光位置；若無法取得太陽光源影像，則系統不運作；另以避免諸多原因造成影像超出攝影機可視範圍，系統利用地球自轉速度每小時 15 度[16]及攝影機可視角為 53.14 及 43.6 度，設計每兩小時依中央氣象局所提供之台灣四季仰角與方位角再進行校對校，系統運作流程如圖 4.20 所示。

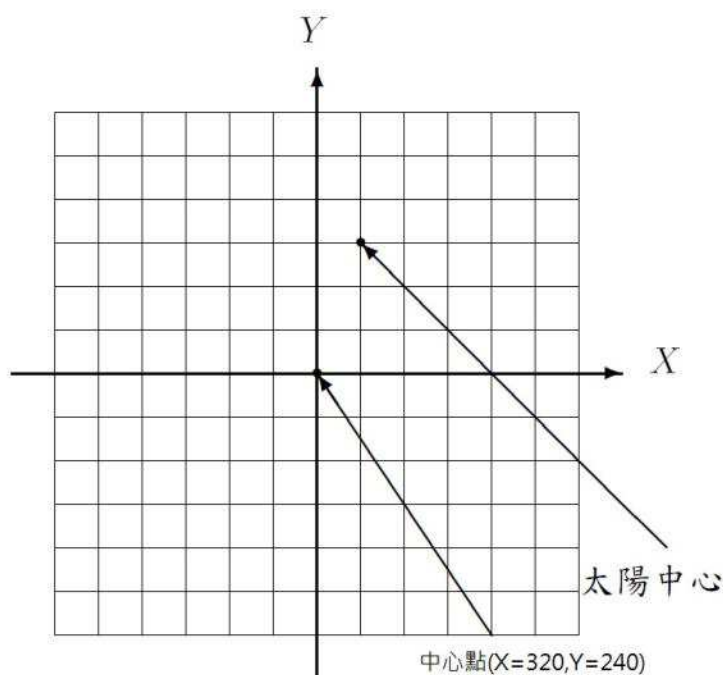


圖 4.11 影像座標示意圖

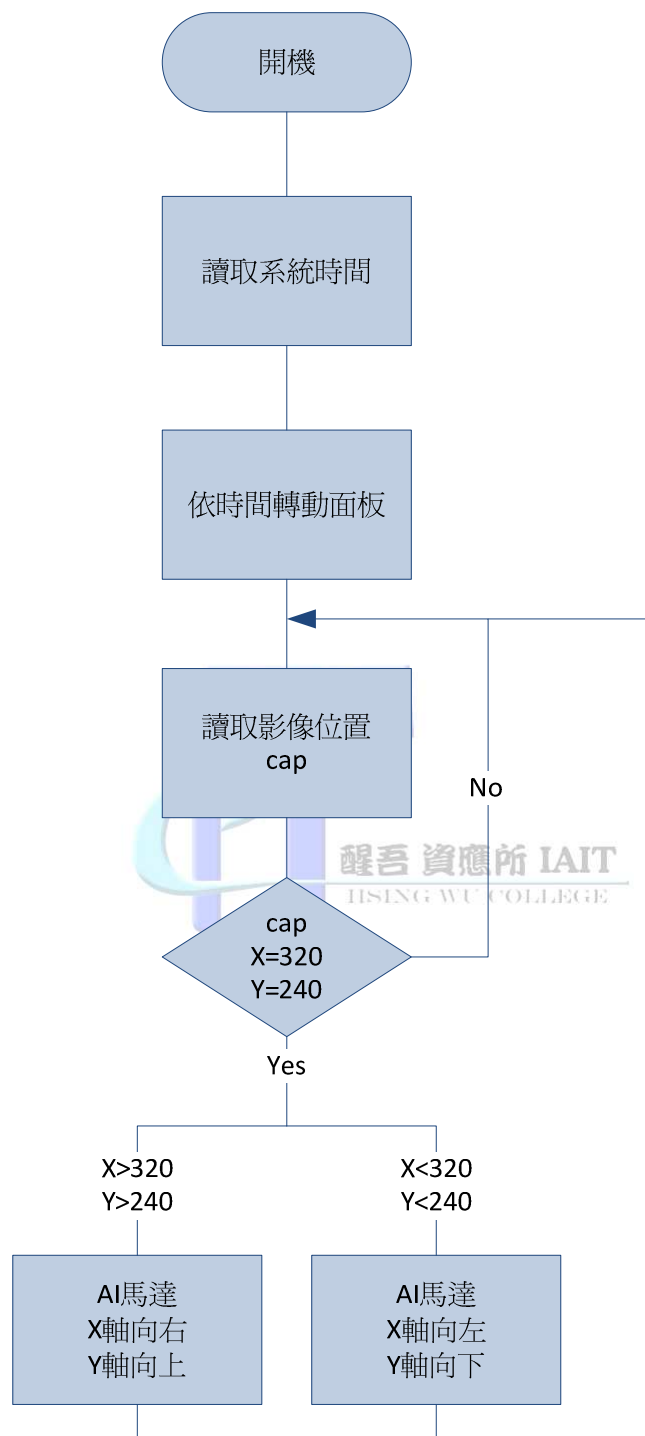


圖 4.12 系統運作流程圖

4.4. 追控系統性能測試

本小節將說明系統測試環境，並利用自製日晷與固定模式來驗正其精確度及比較發電量

4.4.1. 測試環境

本實驗分別於室內及室外進行，室內是利用 200w 燈泡模擬太陽光源並置於系統 2 公尺外進行測試，由於空間有限，無法針對移動燈泡來模擬太陽位移，而採用轉動系統自身平台及搭配指南針進行測試及角度測量，並使用數位型三用電表實際量測發電量，如圖 4.21 所示；戶外測試則於戶外以實際太陽進行測試，系統介面如圖 4.22 所示。



圖 4.13 室內測試系統設備實際圖

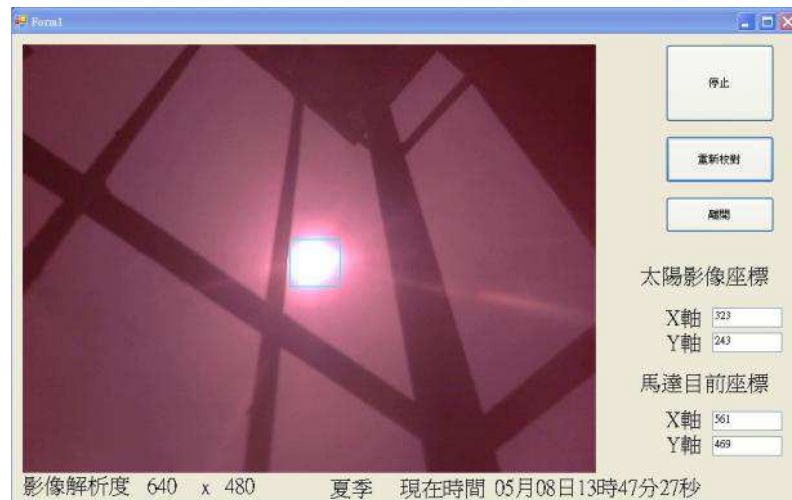


圖 4.14 室外測試系統介面實際圖

4.4.2. 精確度測試

為了解本系統當時的追蹤準確度，參考了國立台灣大學機械工程學研究所集光式太陽光發電追控系統研製碩士論文內所設置的自製日晷機構[6]，以日影長度判斷太陽光與太陽電池法線的入射角度，以辨別追蹤系統的精確程度。其日影判斷追蹤精度的設計圖，如圖 4.23 所示，圖中圓柱為高 $L=15\text{mm}$ 的自製日晷，其下為以 1mm 相間隔的刻度圓，設置在上層框架的機構上。當陽光以 i 的角度斜射進入受光平面時，將在平面上產生一長度為 R 的日影，因此太陽光的入射角 $i=\tan^{-1}(R/L)$ ，如(4-1)式，實際完成如圖 4.24 所示。

$$i=\tan^{-1}(R/L) \quad (4-1)$$

其中符號說明如下：

L : 為量測日晷圓柱長度

R : 為當時從日晷圓柱外周至陰影末端的長度

i : 為當時太陽入射角，其定義為直進光線與日晷圓棒小於 90 度的交角

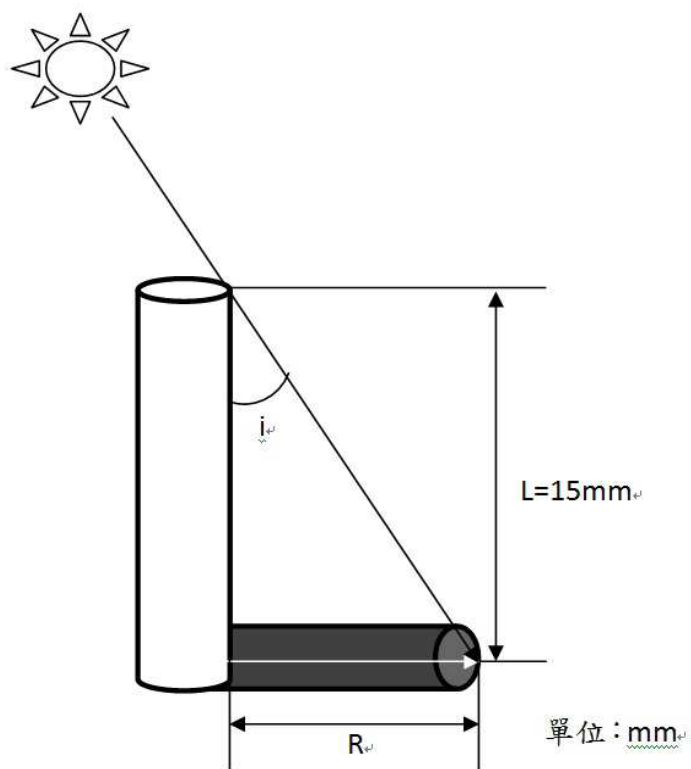


圖 4.15 自製日晷精確度測量示意圖

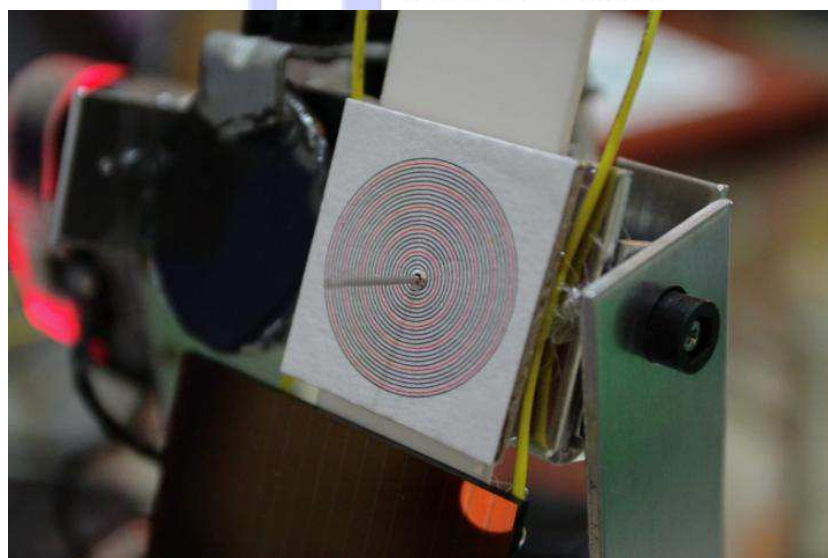


圖 4.16 自製日晷實際完成圖

4.4.3. 實驗結果

本實驗配合自製日晷機構與固定模式進行發電量測試，室內不追蹤實體測試圖如圖 4.25、圖 4.26、圖 4.27、圖 4.28、圖 4.29 及圖 4.30；室內追蹤實體測試圖如圖 4.31、圖 4.32、圖 4.33 及圖 4.34。戶外不追蹤實體測試如圖 4.35、圖 4.36 及圖 4.37 所示，戶外追蹤實體測試如圖 4.38、圖 4.39 及圖 4.40 所示。實驗結果如表 4.3、表 4.4 所示。

表格 4.3

室內發電量測試一覽表

測試方式 平台移動角度	不追蹤 發電量	影像追蹤 發電量
0度	17.6	17.9
10.6度	17.6	17.9
18.43度	17.5	17.8
20.5度	17.5	17.8
33.69度	17.4	17.7
41.4度	17.2	17.6
45度度	17.1	17.8
49.6度	16.5	17.7
53.13度	16.7	17.7
59.03度	16.3	17.1
平均發電量	17.1	17.7

資料來源：本實驗結果

表格 4.4

室外發電量測試一覽表

測試方式 平台移動角度	不追蹤 發電量	影像追蹤 發電量
0度	28.4	27.6
10.5度	28.4	27.6
15.6度	28.3	27.5
22.4度	28.2	27.7
27.6度	28.3	27.5
30.4度	28.2	27.7
33.69度	28.2	27.7
38.6度	28.1	27.6
41.5度	28.1	27.6
45度	28	27.5
平均發電效率	28.2	27.6

資料來源：本實驗結果

根據表 4.3 及 4.4 的測試，此實驗有以下 3 點結論分析：

- 一、室內發電測試，影像追蹤時比不追蹤時之平均發電量高出 0.6 伏特，其發電效益高出 2.85 個百分點，測試結果證明本系統發電效益優於傳統不追蹤式。
- 二、室外發電測試，因太陽光強弱變化時所測數據，不追蹤方式發電量雖較影像追蹤方式高，但其發電量變化比較大，表示不追蹤式發電量受太陽方向移動而有所變化，而本追蹤系統其發電量變化則較小，說明本系統可有效追蹤指向太陽光，發電量穩定而不受平台移動所影響。
- 三、觀察室內及室外量測時，自製日晷機構圓柱外周至陰影末端長度均未超過所設置以 1mm 做為刻度的最內圓圈，因此由公式計算，此時之太陽入射角 $i = \tan^{-1}(1/15) = 3.814$ 度，表示本系統無論在室內或室外期追蹤精度優良，皆可控制追蹤精度小於 3.814 度，換算 AI 馬達步數(每步 0.325 度)約在 12 步內。



圖 4.17 室內不追蹤模式 0 度發電圖



圖 4.18 室內不追蹤模式 18.43 度發電圖



圖 4.19室內不追蹤模式33.69度發電圖



圖 4.20室內不追蹤模式45度發電圖

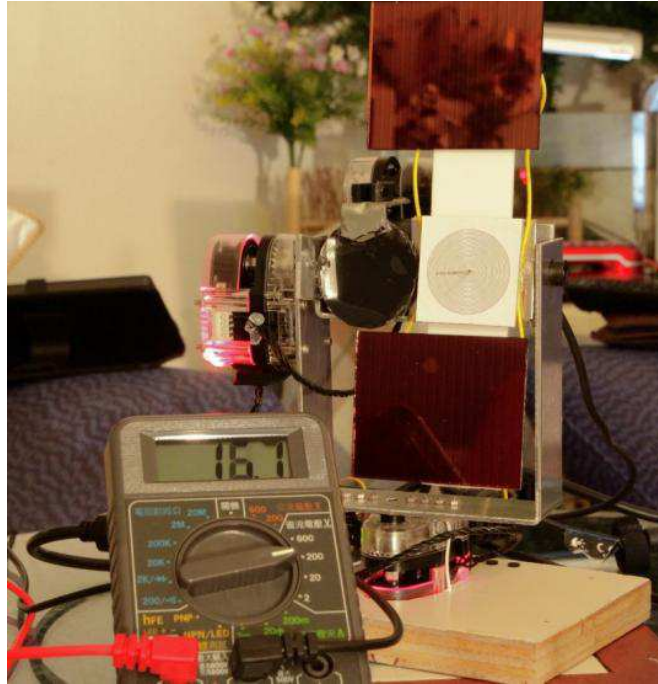


圖 4.21室內不追蹤模式53.13度發電圖

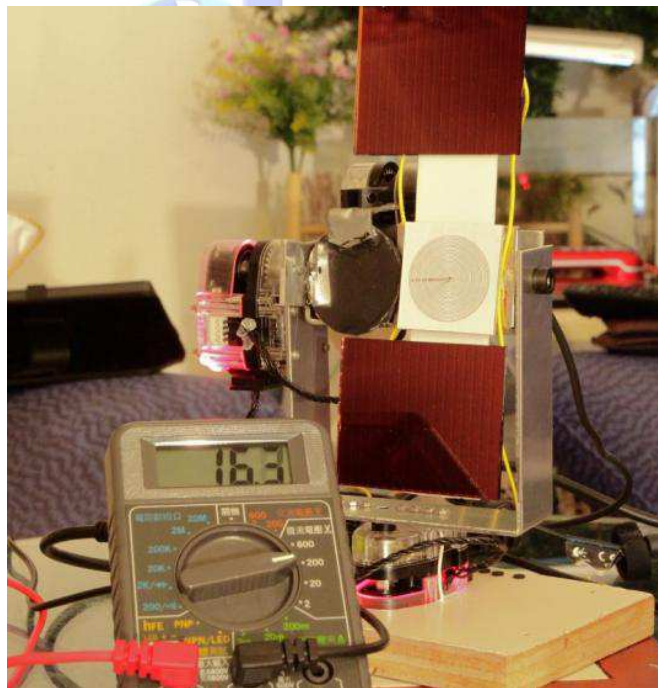


圖 4.22室內不追蹤模式59.03度發電圖

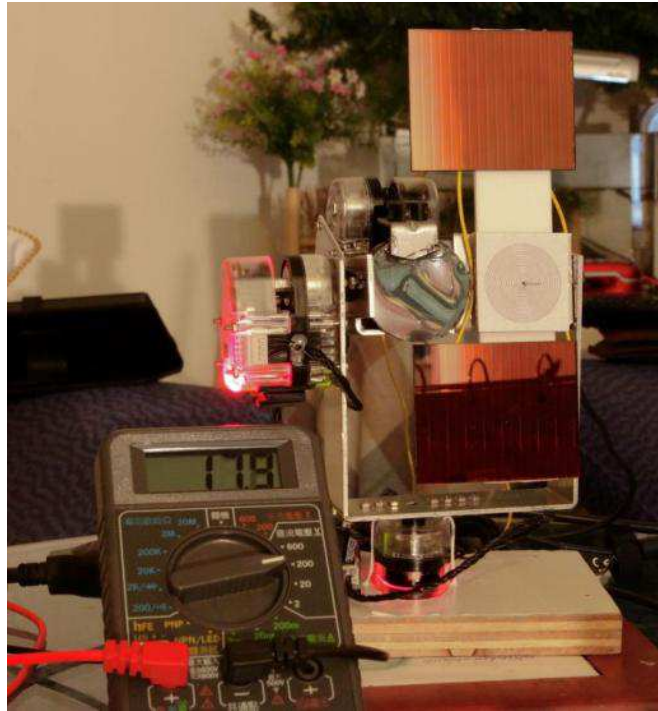


圖 4.23室內追蹤模式0度發電圖



圖 4.24室內追蹤模式33.69度發電圖

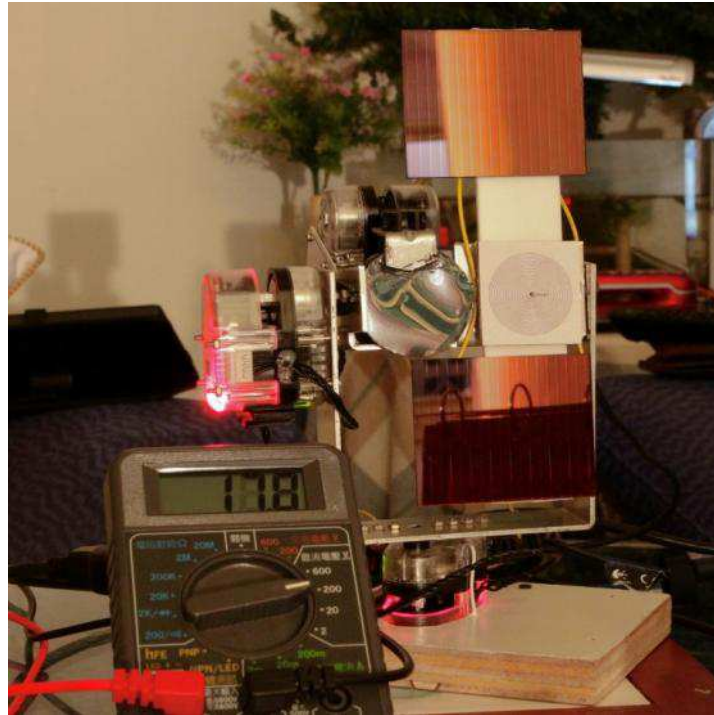


圖 4.25室內追蹤模式45度發電圖

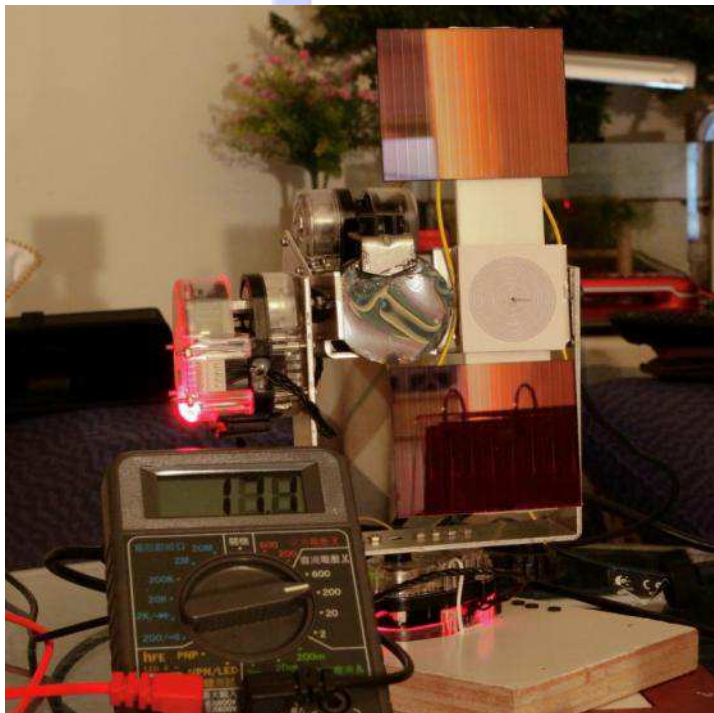


圖 4.26室內追蹤模式59.03度發電圖

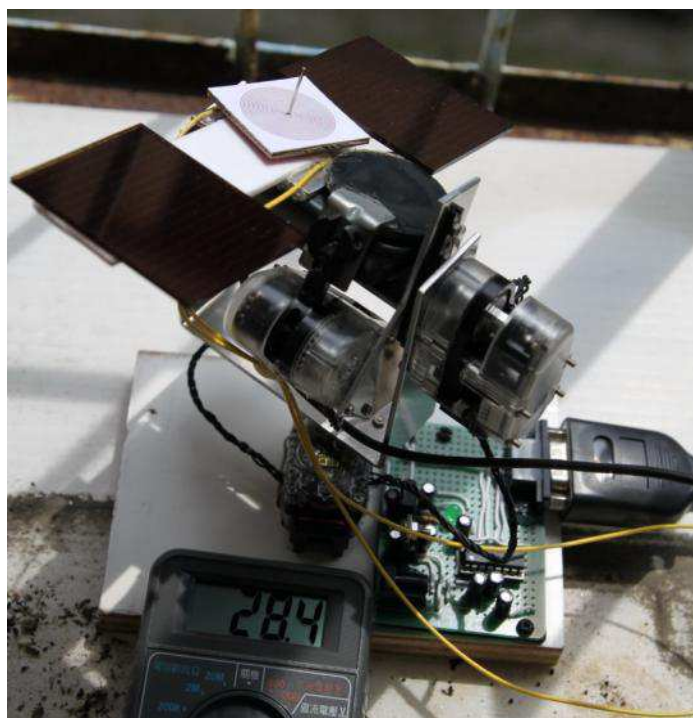


圖 4.27戶外不追蹤模式0度發電圖

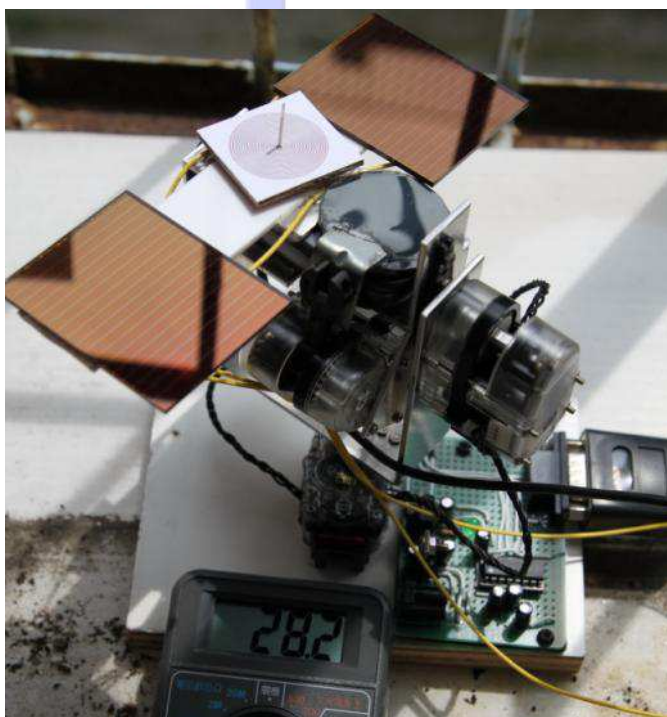


圖 4.28戶外不追蹤模式33.6度發電圖

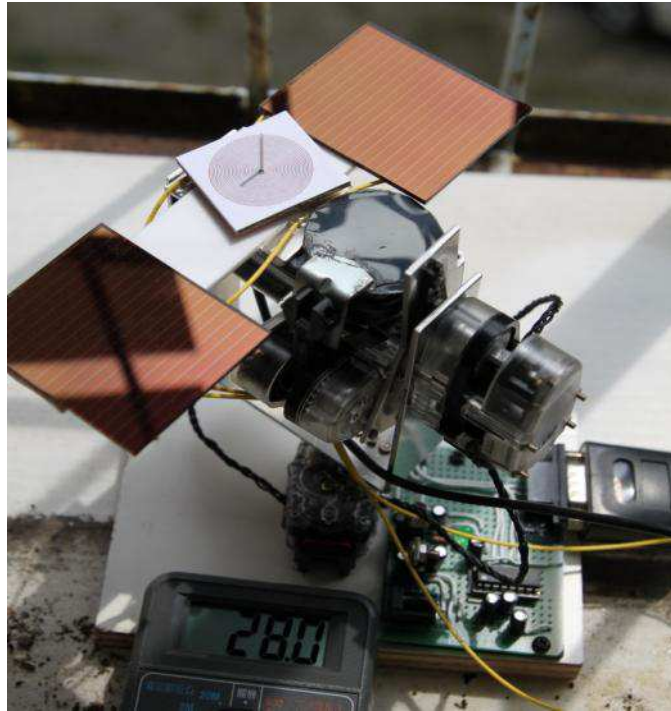


圖 4.29戶外不追蹤模式45度發電圖



圖 4.30戶外追蹤模式0度發電圖

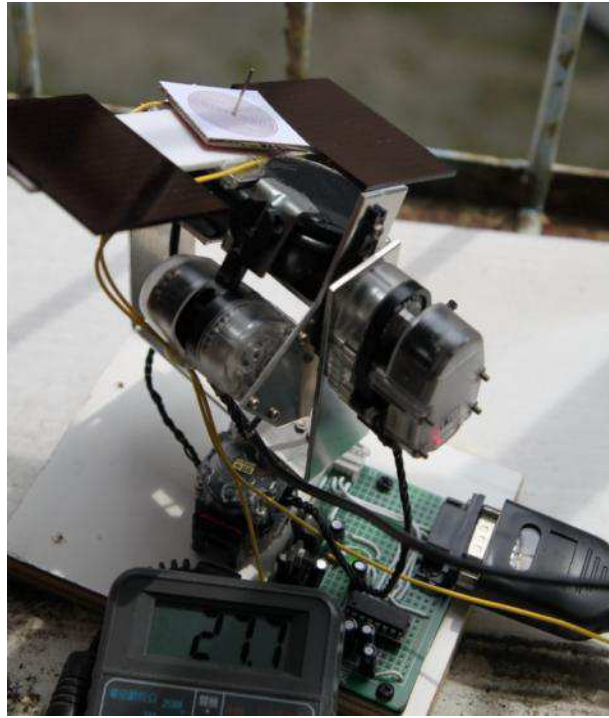


圖 4.31戶外追蹤模式33.6度發電圖

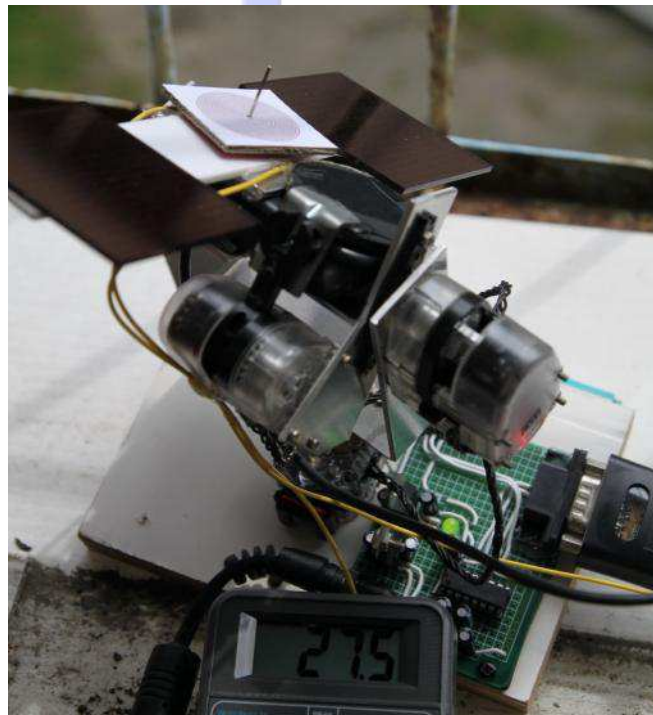


圖 4.32戶外追蹤模式45度發電圖

5. 討論與結論

5.1. 討論

本研究所設計製作的應用攝影機於太陽能追日系統，符合當初所設計的能有效的追蹤太陽光源，並提高發電效率的設定目標，但是仍有可以在加強改良的地方，分別敘述如下所示。

一、雙層式追蹤機構方面

此機構框架設置未能配合 AI 馬達角度做精確的安裝，導致控制追蹤機構角度上，無法以 AI 馬達的角度做為控制機構所需的角度的，改善方法須以程式設計上修正其誤差值，或加強機構安裝時的精確度，依 AI 馬達自身角度作為機構控制角，減少程式換算角度的問題。

二、感測器方面

使用羅技 C310 網路攝影機作為感測器，其可視角約為 53 度，系統在追控或搜尋目標時依可視角範圍大小而有所不同，若能改用可視角度較大之攝影機，將可增加追控及搜尋範圍，亦能減少系統校對的步驟。

三、控制電路方面

本追蹤控制電路及馬達均未針對水氣做處理，若遇雨天將導致系統損毀。其改善方法在攝影機部分，須使用不影響攝影之透明材質進行防水措施，而馬達及電路部分則無此限制，採用一般防水材質進行防水措施即可，如此可達降低水氣對本系統的影響。

5.2. 結論

本研究目的是以攝影機作為主動式太陽能追日系統的感測器，配合雙軸式移動機構，並應用影像辨識的技術，來取得太陽光源最佳的位置，增加發電效率，經實驗結果，可獲得以下結論。

一、應用攝影機做為主動式太陽能追日系統的感測器，並配合雙軸式移動機構的設計，經實驗，其發電量較固定式太陽能系統高，證實可有效提升太陽能電池的發電量。

二、以攝影機為感測器配合雙軸機構，經實驗，其追蹤效果平均可達追蹤精確度 3.814 度以內，相較台灣大學所開發之「集光式太陽光發電系統」[6]平均追蹤精度 4.29 度高出 0.476 度，表示平均追蹤精確度優良。

由此，所設計應用攝影機於太陽能追日系統是可行且性能優異，是作為主動式太陽能追日系統另一項可行的方案。



參考文獻

- [1]. 汪宗煌, “我國太陽光電產業發展現況與展望,” 新聞稿, 2009, pp. 3.
- [2]. 伍柏霖、林信普、賴忠進, “應用人工智慧理論於雙軸太陽能追日系統之研究,” 遠東學報, vol.26, no.2009, pp. 219-232.
- [3]. 沈仲晃, “太陽能電池安裝角度與電能輸出之研究,” 南開技術學院技術學刊, vol.20, 2005, pp. 15-20.
- [4]. 何賢中, “太陽能追日系統研製,” 碩士論文, 聖約翰科技大學, 新北市, 2010.
- [5]. 柯昱光, “太陽能追日系統之感測器研製,” 碩士論文, 東華大學, 花蓮縣, 2009.
- [6]. 葉益男, “集光式太陽光發電追日系統研製,” 碩士論文, 台灣大學, 新北市, 2003.
- [7]. 李明博、黃聖澤、劉宏益、林文德、卓胡誼, “太陽能追日系統之研究,” 2004, pp.2148-2153.
- [8]. 張智凱, “被動式雙軸太陽追蹤器之追控系統開發,” 碩士論文, 中央大學, 桃園縣, 2008.
- [9]. 王文宏, “高效率太陽光電能原轉換系統,” 碩士論文, 元智大學, 桃園縣, 2006.
- [10]. 莊榮瀚, “太陽追蹤器之設計與測試,” 碩士論文, 中央大學, 桃園縣, 2008.
- [11]. 瀚昱能源科技, “瀚昱發展高轉換效率 III-V 聚光型太陽能模組,”
http://www.digitimes.com.tw/tw/dt/n/shwnws.asp?id=0000127088_QXU16NMN75OYTJ93B134G&tid=7&dt=t
- [12]. R.C.Gonzalez and R.E.Woods 著, 數位影像處理, 台灣培生教育出版, 2004。
- [13]. 維基百科 <http://zh.wikipedia.org/wiki/%E5%A4%AA%E9%99%BD%E5%85%89>
- [14]. 魏正翰, 高愷懃, “多點觸控系統的設計與建置,” 學位專題, 大同大學, 新北市, 2009.
- [15]. 交通部中央氣象局
<http://www.cwb.gov.tw/V7/knowledge/astrometry/cdata/season.htm>
- [16]. 維基百科
<http://zh.wikipedia.org/wiki/%E5%9C%B0%E7%90%83%E8%87%AA%E8%BD%AC>
- [17]. 張興政, “太陽能動態偵測與尋跡定位技術,” 碩士論文, 逢甲大學, 台中市, 2007.
- [18]. 張建偉, “太陽能電池最大功率點追蹤之研究,” 碩士論文, 成功大學, 台南市, 2009.

- [19]. 紀卓志明,“史特林引擎太陽能發電,” 碩士論文, 萬能科技大學, 桃園縣, 2009.
- [20]. 高政良,“具火災偵測之追蹤式數位錄影系統之研究,” 碩士論文, 南台科技大學, 台南市, 2002.
- [21]. 許博勝,“雷射針灸與穴道點自動定位系統,” 碩士論文, 中央大學, 桃園縣, 2008.
- [22]. 謝宗明,“自動化影像定位系統及其定位標記之應用,” 碩士論文, 成功大學, 台南市, 2005.
- [23]. 郭桓甫,“應用即時影像辨識技術於物體追蹤之無線遙控自走車的設計製作,” 碩士論文, 成功大學, 台南市, 2006.



附錄

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.IO.Ports;
using System.Threading;
using Emgu.CV;
using Emgu.CV.Structure;
using Emgu.Util;
using Emgu.CV.UI;
using Emgu.CV.CvEnum;
using System.Runtime.InteropServices;
using AForge;
using AForge.Imaging;
using AForge.Math.Geometry;
using System.Drawing.Imaging;

namespace Cap_test1
{
    public partial class Form1 : Form
    {
        private BlobCounter blobCounter = new BlobCounter();
        private Blob[] blobs;
        Bitmap image2;
        private Capture _capture;
        private bool _captureInProgress;
        comport_list motor = new comport_list();
        private Thread workerThread = null;

        public Form1()
```

```

{
    InitializeComponent();
    _capture = new Capture(0);
    motor.Danger += new workEvent(motor_position);
    motor.Open("COM5", 115200, 8, Parity.None, StopBits.One);
    timer1.Interval = 1000; //設定每一秒執行一次(顯示時間)
    timer1.Start();
}

private void ProcessFrame(object sender, EventArgs arg)
{
    Image<Bgr, Byte> frame = _capture.QueryFrame();
    Image<Gray, Byte> grayFrame = frame.Convert<Gray, Byte>();
    grayFrame = grayFrame.PyrDown();
    grayFrame = grayFrame.PyrUp();
    //設定門檻值
    grayFrame._ThresholdToZero(new Gray(240));
    captureImageBox.Image = grayFrame;
    image2 = captureImageBox.Image.Bitmap;
    image2 = AForge.Imaging.Image.Clone(image2,
    PixelFormat.Format24bppRgb);
    blobCounter.ProcessImage(image2);
    blobs = blobCounter.GetObjectsInformation();

    if (blobCounter.ObjectsCount > 0)
    {
        PointF[] pts = new PointF[] {
            new PointF(blobs[0].Rectangle.Left, blobs[0].Rectangle.Bottom),
            new PointF(blobs[0].Rectangle.Right, blobs[0].Rectangle.Bottom),
            new PointF(blobs[0].Rectangle.Right, blobs[0].Rectangle.Top),
            new PointF(blobs[0].Rectangle.Left, blobs[0].Rectangle.Top)};
        frame.DrawPolyline(Array.ConvertAll<PointF, Point>(pts,
            Point.Round), true, new Bgr(255, 255, 0), 1);
        textBox2.Text = blobs[0].CenterOfGravity.Y.ToString();
        textBox3.Text = blobs[0].CenterOfGravity.X.ToString();
    }
}

```



```

        label8.Text = frame.Width.ToString();
        label9.Text = frame.Height.ToString();
        captureImageBox.Image = frame;
    }
}

private void captureButton_Click(object sender, EventArgs e)
{
    #region if capture is not created, create it now
    if (_capture == null)
    {
        try
        {
            _capture = new Capture();
        }
        catch (NullReferenceException excpt)
        {
            MessageBox.Show(excpt.Message);
        }
    }
}
#endregion

//先執行一次 Button1
button1.PerformClick();
timer2.Interval = 100;
timer2.Start();
//每小時執行校對 1 次
timer3.Interval = 3600000;
timer3.Start();
if (_capture != null)
{
    if (_captureInProgress)
    {
        //stop the capture
        captureButton.Text = "啟動";
        Application.Idle -= ProcessFrame;
    }
}

```

```

        timer2.Stop();
    }
    else
    {
        //start the capture
        captureButton.Text = "停止";
        Application.Idle += ProcessFrame;
    }
    _captureInProgress = !_captureInProgress;
}

}

private void ReleaseData()
{
    if (_capture != null)
        _capture.Dispose();
}

private void captureImageBox_Click(object sender, EventArgs e)
{
}

private void Form1_Load(object sender, EventArgs e)
{
}

private void label2_Click(object sender, EventArgs e)
{
}

private void label5_Click(object sender, EventArgs e)
{
}

```

```

private void textBox5_TextChanged(object sender, EventArgs e)
{
}

private void motor_position()
{
    SetText(label6, motor.position_ID1.ToString());
}

// Delegates to enable async calls for setting controls properties
private delegate void SetTextCallback(System.Windows.Forms.Control control,
string text);

// Thread safe updating of control's text property
public void SetText(System.Windows.Forms.Control control, string text)
{
    if (control.InvokeRequired)
    {
        SetTextCallback d = new SetTextCallback(SetText);
        Invoke(d, new object[] { control, text });
    }
    else
    {
        control.Text = text;
    }
}

private void button1_Click(object sender, EventArgs e)
{
    textBox1.Text = motor.PosRead_10bit(8).ToString();
    textBox4.Text = motor.PosRead_10bit(12).ToString();
    string mm = DateTime.Now.ToString("MM");
    int mmd = Int32.Parse(mm);
    //判斷四季

```

//判斷是否為夏季 4 至 6 月(使用夏季仰角及方位角)

if (mmd >= 4 && mmd <= 6)

{

 string dd = DateTime.Now.ToString("HH");

 label13.Text = "夏季";

 switch (dd)

 {

 case "08":

 motor.PosSend_10bit(8, 0, 160);

 motor.PosSend_10bit(12, 0, 380);

 break;

 case "09":

 motor.PosSend_10bit(8, 0, 180);

 motor.PosSend_10bit(12, 0, 422);

 break;

 case "10":

 motor.PosSend_10bit(8, 0, 200);

 motor.PosSend_10bit(12, 0, 465);

 break;

 case "11":

 motor.PosSend_10bit(8, 0, 356);

 motor.PosSend_10bit(12, 0, 507);

 break;

 case "12":

 motor.PosSend_10bit(8, 0, 512);

 motor.PosSend_10bit(12, 0, 550);

 break;

 case "13":

 motor.PosSend_10bit(8, 0, 668);

 motor.PosSend_10bit(12, 0, 507);

 break;

 case "14":

 motor.PosSend_10bit(8, 0, 824);

 motor.PosSend_10bit(12, 0, 465);

 break;

```

        case "15":
            motor.PosSend_10bit(8, 0, 844);
            motor.PosSend_10bit(12, 0, 420);
            break;
        case "16":
            motor.PosSend_10bit(8, 0, 864);
            motor.PosSend_10bit(12, 0, 375);
            break;
    }
}
//判斷是否為冬季 10 至 12 月(使用冬季仰角及方位角)
if (mmd >= 10 && mmd <= 12)
{
    string dd = DateTime.Now.ToString("HH");
    label13.Text = "冬季";
    switch (dd)
    {
        case "08":
            motor.PosSend_10bit(8, 0, 380);
            motor.PosSend_10bit(12, 0, 330);
            break;
        case "09":
            motor.PosSend_10bit(8, 0, 395);
            motor.PosSend_10bit(12, 0, 355);
            break;
        case "10":
            motor.PosSend_10bit(8, 0, 410);
            motor.PosSend_10bit(12, 0, 380);
            break;
        case "11":
            motor.PosSend_10bit(8, 0, 461);
            motor.PosSend_10bit(12, 0, 385);
            break;
        case "12":
            motor.PosSend_10bit(8, 0, 512);

```

```

        motor.PosSend_10bit(12, 0, 390);
        break;
    case "13":
        motor.PosSend_10bit(8, 0, 668);
        motor.PosSend_10bit(12, 0, 380);
        break;
    case "14":
        motor.PosSend_10bit(8, 0, 824);
        motor.PosSend_10bit(12, 0, 370);
        break;
    case "15":
        motor.PosSend_10bit(8, 0, 844);
        motor.PosSend_10bit(12, 0, 345);
        break;
    case "16":
        motor.PosSend_10bit(8, 0, 864);
        motor.PosSend_10bit(12, 0, 320);
        break;
    }
}
//判斷是否為春秋兩季 1 至 3 月及 7 至 8 月
if (mmd >= 1 && mmd <= 3 || mmd >= 7 && mmd <= 8)
{
    string dd = DateTime.Now.ToString("HH");
    label13.Text = "春秋季";
    switch (dd)
    {
        case "08":
            motor.PosSend_10bit(8, 0, 290);
            motor.PosSend_10bit(12, 0, 360);
            break;
        case "09":
            motor.PosSend_10bit(8, 0, 347);
            motor.PosSend_10bit(12, 0, 400);
            break;
    }
}

```



```

        case "10":
            motor.PosSend_10bit(8, 0, 405);
            motor.PosSend_10bit(12, 0, 440);
            break;
        case "11":
            motor.PosSend_10bit(8, 0, 458);
            motor.PosSend_10bit(12, 0, 460);
            break;
        case "12":
            motor.PosSend_10bit(8, 0, 512);
            motor.PosSend_10bit(12, 0, 480);
            break;
        case "13":
            motor.PosSend_10bit(8, 0, 591);
            motor.PosSend_10bit(12, 0, 455);
            break;
        case "14":
            motor.PosSend_10bit(8, 0, 670);
            motor.PosSend_10bit(12, 0, 430);
            break;
        case "15":
            motor.PosSend_10bit(8, 0, 767);
            motor.PosSend_10bit(12, 0, 395);
            break;
        case "16":
            motor.PosSend_10bit(8, 0, 864);
            motor.PosSend_10bit(12, 0, 360);
            break;
    }
}

}

private void button2_Click(object sender, EventArgs e)
{

```

```

    }

    private void textBox2_TextChanged(object sender, EventArgs e)
    {
    }

    private void label4_Click(object sender, EventArgs e)
    {
    }

    private void textBox4_TextChanged(object sender, EventArgs e)
    {
    }

    private void button3_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }

    private void label7_Click(object sender, EventArgs e)
    {
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        //顯示系統月日時分秒
        label7.Text = DateTime.Now.ToString("MM 月 dd 日 HH 時 mm 分 ss 秒");
    }

    private void timer2_Tick(object sender, EventArgs e)
    {
        //-----馬達的控制-----
        try
        {
            int capx = blobs[0].CenterOfGravity.X;

```

```

        int capy = blobs[0].CenterOfGravity.Y;
        int motx = motor.PosRead_10bit(8);
        int moty = motor.PosRead_10bit(12);
        //x 軸中心點 320 正負 5 個像素為範圍，馬達每次移動 3
        if (capx < 315)
        {
            motx = motx - 3;
            motor.PosSend_10bit(8, 0, motx);
        }
        else if (capx > 325)
        {
            motx = motx + 3;
            motor.PosSend_10bit(8, 0, motx);
        }

        //y 軸中心點 240 正負 5 個像素為範圍，馬達每次移動 3
        if (capy < 235)
        {
            moty = moty + 3;
            motor.PosSend_10bit(12, 0, moty);
        }
        else if (capy > 245)
        {
            moty = moty - 3;
            motor.PosSend_10bit(12, 0, moty);
        }

        //顯示馬達現在角度
        textBox1.Text = motx.ToString();
        textBox4.Text = moty.ToString();
    }
    catch (System.IndexOutOfRangeException)
    {
        timer2.Start();
    }

```

```

    }
    }

    private void timer3_Tick(object sender, EventArgs e)
    {
    }

    private void textBox3_TextChanged(object sender, EventArgs e)
    {
    }

    private void textBox4_TextChanged_1(object sender, EventArgs e)
    {
    }

    private void label13_Click(object sender, EventArgs e)
    {
    }

    private void timer3_Tick_1(object sender, EventArgs e)
    {
        button1.PerformClick();
    }
}
}

```