

國立雲林科技大學  
電機工程系

實務專題

高階微步進馬達控制應用

**Application of High-level micro-stepping motor**

專題學生： 王志強（四電機四 A B10812033）  
陳昱憲（四電機四 A B10812054）

指導教授： 毛偉龍

中華民國 112 年 1 月

國立雲林科技大學  
電機工程系  
實務專題

本校電機工程系王志強君、陳昱憲君所提實務專題

「實務專題中文題目」製作完成，同意辦理畢業離校。

指導教授 \_\_\_\_\_

112 年 4 月 20 日

# 目錄

目錄.....	I
第一章緒論.....	1
1.1 研究動機與研究目的.....	1
1.2 文獻探討.....	1
第二章系統架構.....	1
2.1 系統架構.....	1
2.2 通訊介面介紹.....	1
2.2.1 通用非同步收發傳輸器 UART.....	1
2.2.2 串列周邊介面 SPI.....	2
2.3 STM32F401RE .....	2
2.4 STM32F401RCT6 CORE BOARD .....	3
2.5 X-NUCLEO-IHM02A1 .....	4
2.5.1 架構.....	4
2.5.2 L6470PD 晶片介紹 .....	4
2.6 光遮斷器介紹.....	4
2.7 光隔離.....	5
2.8 步進馬達.....	5
2.9 人機介面.....	6
第三章系統軟體.....	7
3.1 Keil uVision5 開發軟體.....	7
3.2 STM32CUBE.....	7
3.3 Altium designer.....	8
3.4 VisualStudio 開發工具軟體.....	8
第四章實驗結果.....	10
4.1 電路.....	10
4.1.1 兩軸電路測試.....	10
4.1.2 Altium designer 設計電路.....	10
4.1.3 實體圖.....	12
4.2 動作指令執行.....	13
4.2.1 使用者介面介紹.....	13

4.2.2 指令介紹.....	13
4.2.3 各軸獨立.....	14
4.2.4 多軸同動.....	14
4.2.5 參數讀取與接收.....	15
4.2.6 速度曲線圖.....	16
4.2.7 狀態及異常指示.....	16
4.2.8 極限開關觸發.....	19
第五章結論與未來展望.....	24
5.1 結論.....	24
5.2 未來展望.....	24
參考文獻.....	25

# 第一章緒論

## 1.1 研究動機與研究目的

隨著科技日新月異，定位平台在生產自動化被廣泛運用，例如：半導體產業、平面顯示器產業及印刷電路板產業等，都將與自動控制技術領域息息相關，產品大都朝向精密化、微小化之趨勢，因此在製程及檢測等步驟上就需要更高的精度。

而運動控制也是現今自動化的核心技術，簡單來說就是透過對位置與速度進行即時的控制。以平台在運動控制為例，若想要照所規劃的路徑移動，就需要路徑規劃和控制理論對性能的評估，且考量每個機台本身的特性。在馬達帶動滾珠螺桿使平台移時，必須考量馬達的規格和平台本身的剛性等特性，大部分的系統都有速度和加速度的限制，還有機械結構的不同所產生的摩擦力、推力和馬達轉矩等都不一致，會導致移動時定位、馬達帶動的速度、加減速度所要調整的參數都不相同的，所以想要精准的控制就必須將這些不確定因素考量到所設計之控制器系統方塊中。一般步進的控制器設計也是影響運動控制的關鍵，因此想要設計出一個精准、穩定且按照所規劃軌跡快速移動的實際操作機台，必須在控制器和軌跡規劃方面多加鑽研。

在自動控制領域中，本專案是以回授系統擷取到硬體上的光遮斷器並且經由開發板處理將回授後的資訊做補償，並且適時修正對位的位置使其更精確。所以在自動化製程中，必須經常對位置做修正以利進行製程時的後續動作，應用此技術不僅可以減少人力成本，還可以使產品良率更高、更好，避免人為因素使得產品良率下降。

## 1.2 文獻探討

隨著數位系統的進步，步進馬達的應用也更加普遍。同時步進馬達具有成本低、定位精準、控制方法簡單的優點，因此廣泛地運用在低速定位器材、機械手臂、三軸平台等工業機台上，也常見於電腦周邊硬體，例如印表機等[1-3]。

根據文獻資料[4]致動器指的是類人造機械一種稱呼，它是將電的控制訊號轉成機械行動的裝置。本研究所使用的馬達致動器便是將電氣訊號的電能傳至馬達轉為動能，在經由齒輪及軸承將旋轉運動改變為直線運動的一種類人造機械。線性致動器結構[5]中螺桿與馬達並非一體式，需要使用聯軸器來連結兩軸的動力傳動，聯軸器又分為剛性及撓性兩類，撓性聯軸器的特性是緩和衝擊、吸收平行、偏角、軸相位差、改善位差，改善系統傳統傳動力特性。原點復歸[5]

是許多自動化工程應用當中最為重要的一個功能，檔片觸發到極限開關後會有一個訊號產生，使其找出系統的機械原點。使工作平台回到該設定點，使用極限檢知器與 HOME 檢知器的原點復歸運轉，以系統上已經預設好的定點作為原點的參考點。黃正杰[6]提及到多軸控制步進馬達為個人電腦、UART、主控制板、步進馬達以及個人電腦上的介面程式。使用者可以透過介面程式進行指令操作以及監控系統狀態，指令操作包含各軸馬達的轉動控制和行程資料的儲存，而監控系統狀態的功能包含接收受控系統的感測器訊號、控制板回傳指令以及行程資料的讀取等。而本專案也使用到此系統架構來進行多軸步進馬達控制。吳信成[7]所建構之 X-Y 雙軸運動控制平台是由兩組線性運動模組相互交疊所組成，即在 X 軸線性運動模組的移動平台上搭載 Y 軸線性運動模組，其中線性運動模組是由滾珠導螺桿和 U 型線性滑軌所組成。採用可撓性連軸器以避免因機構裝設的不準確而引起機械上的誤差。並使用極限開關來限制兩個移動平台的行進距離。吳信成所建構的 XY 控制平台，其機械結構非常適合用來作為 XXY 平台機械結構參考與學習。

## 第二章系統架構

### 2.1 系統架構

本研究企劃以開發板 STM32F401RE，馬達驅動板 IHM02A1 與 C#為開發平台，將程式燒入 STM32F401RE 並透過 C#撰寫操作介面，來達成人機介面(HMI)實現，操作介面上可進行步進馬達之控制，將使用者所操控之動作編碼後利用 UART 進行傳送至 STM32F401RE，再由開發板解碼後藉由 SPI，對驅動板下達命令進行多軸步進馬達控制，架構圖如圖 1。

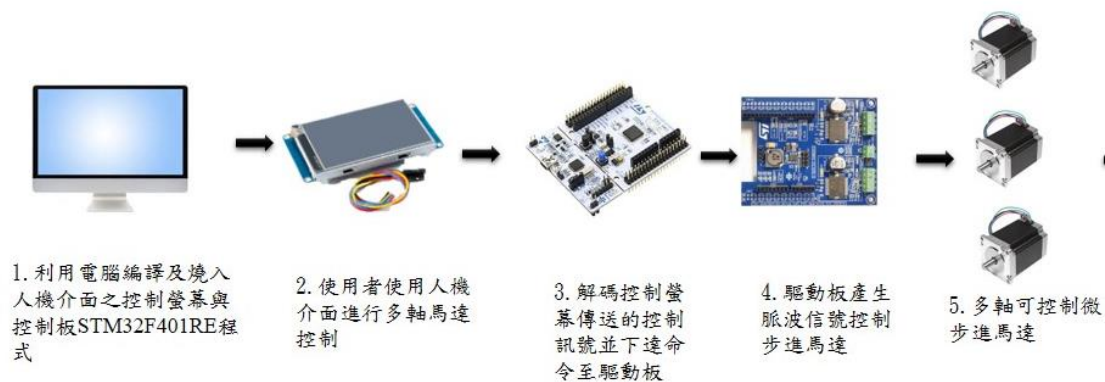


圖 1 系統架構圖

### 2.2 通訊介面介紹

#### 2.2.1 通用非同步收發傳輸器 UART

UART (Universal Asynchronous Receiver/Transmitter) 為常見數據傳送方法之一，其硬體接線也相對簡單，若將電源與接地線除外後，只須雙方各提供 RX 與 TX 兩個腳位即可互相傳送資料，如圖 2 左。UART 提供雙方以此通訊方式進行資料傳送，其傳送特性為，(1)一個起始位元，(2)一到八個位元的資料量，(3)可選擇奇同位元或偶同位元檢測，(4)一到兩個停止位元，傳送資料如圖(2)右。此傳輸方式於觸控介面與開發板 STM32F401RE 互相溝通所使用，需在開發板程式內部加入自定義函式庫用來轉換雙方傳送的資料，轉換成觸控介面與開發板分別能夠處理的數據，實現人機介面功能。

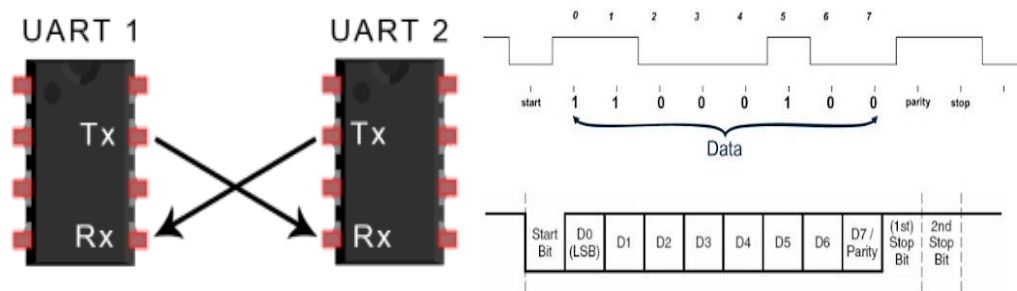


圖 2UART 硬體接線圖與資料格式

### 2.2.2 串列周邊介面 SPI

SPI (Serial Peripheral Interface Bus) 為高速同步串列介面，其低接腳數，簡單結構，簡易使用與高傳輸速度，使其被廣泛應用在單晶片與微控制器之資料傳送上，且有 SPI 模組提供使用。SPI 有四隻接腳，分別為 MOSI(Master Out Slave In)主出從入、MISO(Master In Slave Out)主入從出、SCLK(System Clock)系統時脈、SS (Slave Select)晶片致能，基本傳輸方式為接收端根據 SS 之高低準位決定可否接收資料，資料從傳送端 MOSI 發送由接收端 MISO 接收至位移暫存器內儲存後，完成資料傳送的步驟，傳輸結構之示意圖為圖 3。

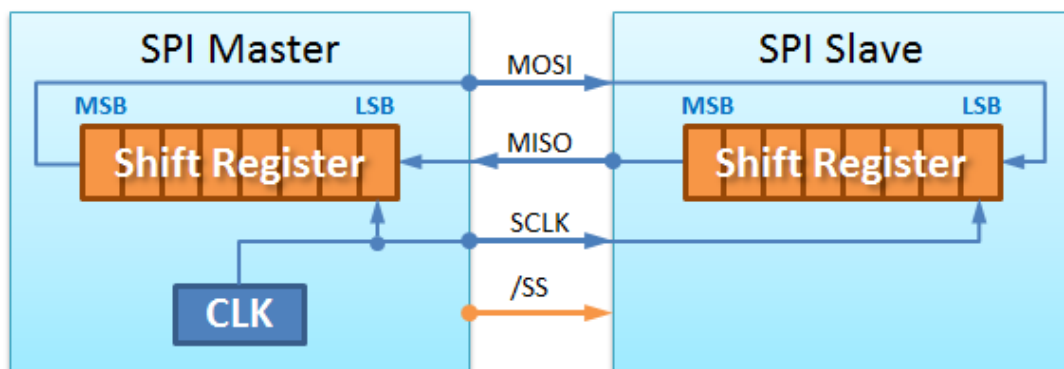


圖 3SPI 傳送結構圖

## 2.3 STM32F401RE

圖 4STM32F401RE 開發板具有 mbed 功能，支持 Arduino 接口，同時還提供 ST Morpho 擴展排針，可連接微控制器的所有周邊外部設備，具有 Arduino 連接功能能夠使此開發板具有 Arduino 之擴充功能，能夠使開發人員輕鬆快速增加特殊功能。此開發板也能夠對 STM32 系列的全面支持，讓開發人員能夠靈活地開發原型設計，在各個開發階段動態微調軟硬件。此開發板具有易用性、靈活性、連接功能和主流工具資源完美地融合在一起，是開發人員的理想的開發板之一。





圖 4 開發板 STM32F401RE

## 2.4 STM32F401RCT6 CORE BOARD

圖 5 為 STM32F401RCT6，相較於 STM32F401RE 在相同 MCU 與功能下，其燒入方式為使用 JTAG，雖燒入方式較 STM32F401RE 複雜，但有體積較小的優點，較適合系統整合，減少使用空間。

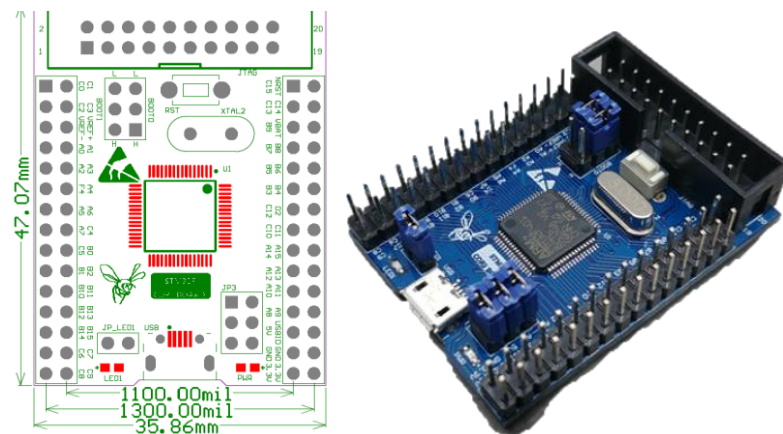


圖 5 驅動板 x-nucleo-ihm02a1(左)和其單組電路圖(右)

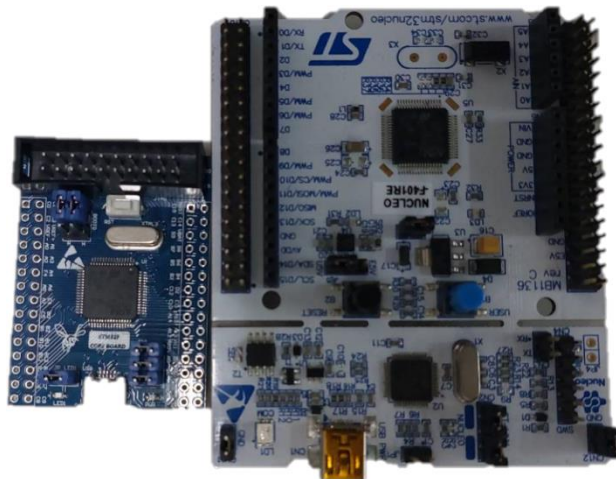


圖 6 原開發板與 STM32F401RCT6 之大小比較圖

## 2.5. X-NUCLEO-IHM02A1

### 2.5.1 架構

此驅動板將使用 AD(Altium designer)進行 PCB 板繪製出 layout 檔再藉由電路板雕刻機雕刻出驅動板，再將驅動板焊接上各式晶片，電子元件，最後完成驅動板。

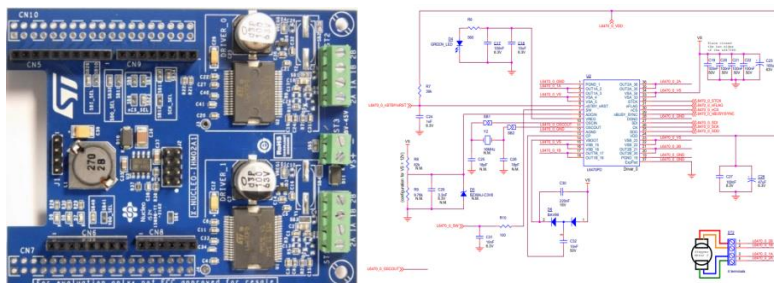


圖 7 驅動板 x-nucleo-ihm02a1(左)和其單組電路圖(右)

### 2.5.2 L6470PD 晶片介紹

圖 8 為 L6470PD 晶片，L6470PD 為兩軸驅動板 x-nucleo-ihm02a1 之主要驅動晶片，主要功能為發送控制步進馬達的脈波信號，此晶片提供 8-45V 工作電壓，128 微步進功能，步進馬達失步檢測，可程式控制的速度曲線和定位能力，SPI 模組，還有良好的功率效能與過溫保護，搭配 STM32F401RE 開發板最多能控制 8 顆步進馬達，適用於開發多軸馬達控制的各項應用。



圖 8 L6470PD 晶片

## 2.6 光遮斷器介紹

圖 9 左，為光遮斷器，光遮斷器是感測器的一種，內部構造是將發光組件與受光組件相對排放，並設置在同封裝內，若有物體通過光遮斷器，將使受光組件接收不到發光組件所發出的光線，若受光組件在受光時能夠觸發電晶體如圖(9)右，並維持穩態，未受光時將產生一個訊號，利用此方式來實現光遮斷器檢測功能，

此計畫利用光遮斷器產生的訊號來檢測馬達是否超過工作範圍，檢測點若觸發光遮斷器，STM32F401RE 開發板將立即接收到訊號並停止步進馬達繼續移動。

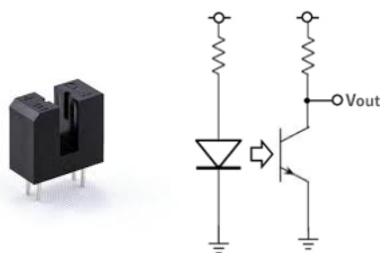


圖 9 光遮斷器(左)與受光組件電路(右)

## 2.7 光隔離

光電耦合元件是以光（含可見光、紅外線等）作為媒介來傳輸電信號的一組裝置，其功能是平時讓輸入電路及輸出電路之間隔離，在需要時可以使電信號通過隔離層的傳送方式。光電耦合元件可以在二個不共地的電路之間傳遞信號，二電路之間即使有高壓也不會影響。

本專題使用圖 ILQ620GB 作為極限開關信號的隔離，內建 4 組光隔離開關，用於做電器隔離，主要功能是要阻隔高電壓以及電壓暫態，不讓這些電路中的電壓暫態影響電路的其他部份。

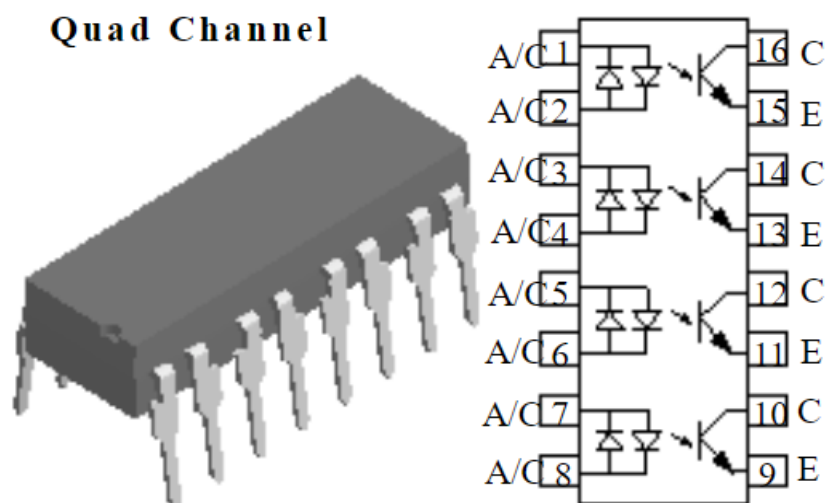


圖 10 四路光隔離外型及內部構造

## 2.8 步進馬達

圖 11 步進馬達是直流無刷電動機的一種，為擁有如齒輪狀突起相銜合的定子和轉子，可藉由切換流向定子線圈中的電流，以一定角度逐步轉動的馬達。步進馬達的特徵是採用開迴路控制處理且切換電流觸發器的是脈衝信號，不需要位

置檢出和速度檢出的回授裝置，所以步進電機可正確地依比例隨脈衝信號而轉動，因此達成精確的位置和速度控制，且穩定性佳。



圖 11 步進馬達

2.9 人機介面

圖 12 為使用 C#設計的操作頁面，其架構參考 ARCUS 4 軸 UI 架構，並加入適當的調整，能夠藉由電腦對馬達下達控制指令，並讀取馬達狀態和極限開關，進行馬達運動控制。

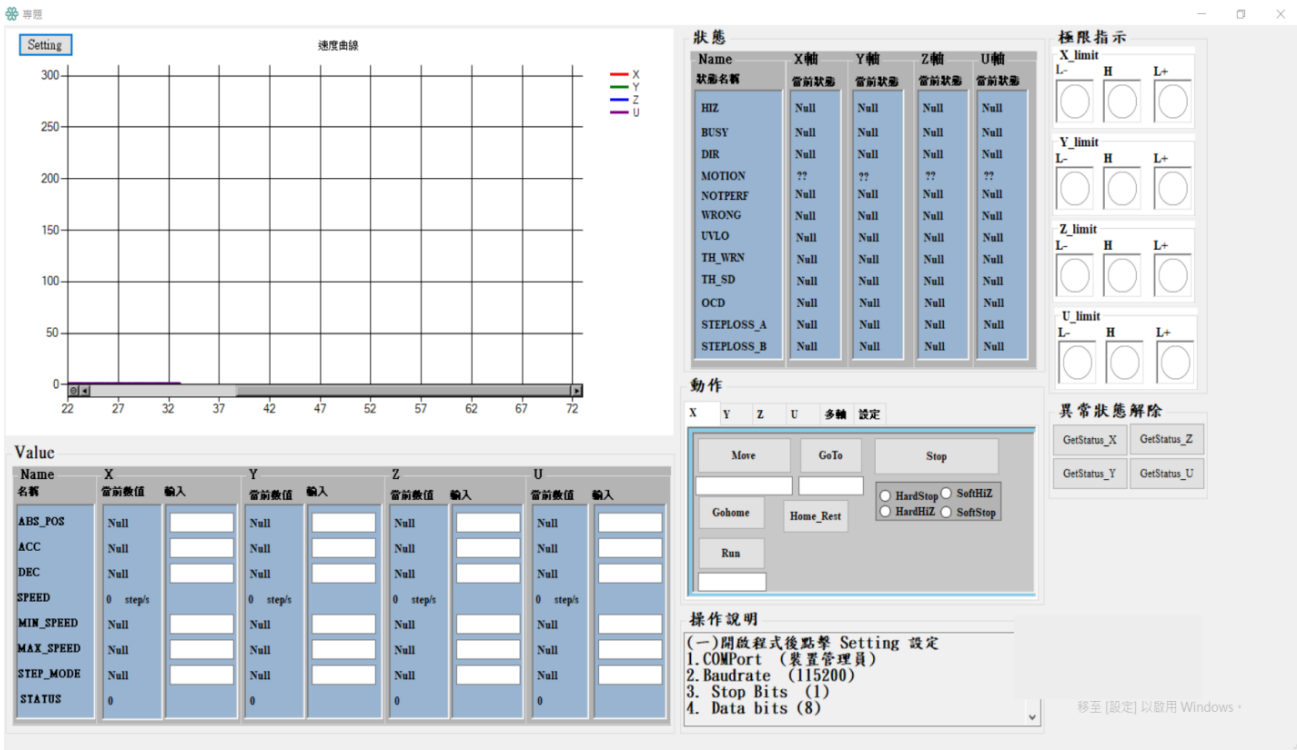


圖 12 C#操作介面

## 第三章系統軟體

### 3.1 Keil uVision5 開發軟體

圖 13 Keil uVision5 是德國知名軟體公司 Keil（現已併入 ARM 公司）開發的微控制器軟體開發平台，是目前 ARM 核心微控制器開發的主流工具。Keil 提供了包括 C 編譯器、巨集彙編、聯結器、庫管理和模擬偵錯程式在內的完整開發專案，通過一個整合開發環境將這些功能組合在一起。



圖 13 Keil uVision5 軟體

### 3.2 STM32CUBE

圖 14 STM32CubeMX 為意法半導體公司開發軟體，在設計其公司所生產的晶片時，利用此軟體能夠大幅縮減開發流程，軟體功能為圖形化腳位設置功能、系統時脈調整功能、能夠自動除錯重複設定，可根據各晶片產生其專有程式庫，設計流程為：1.選擇晶片型號 2.設定晶片腳位功能 3.設定晶片工作頻率 4.設定轉出程式碼格式 5.生成程式專案，如軟體編寫程式 Keil MDK uvision5 相符之 C 程式。



圖 14 STM32CubeMX 軟體

### 3.3 Altium designer

圖 15 Altium Designer 是原 Protel 軟體開發商 Altium 公司推出的一體化的電子產品開發系統，主要運行在 Windows 作業系統。這套軟體通過把原理圖設計、電路仿真、PCB 繪製編輯、拓撲邏輯自動布線、信號完整性分析和設計輸出等技術的完美融合，為設計者提供了全新的設計解決方案，使設計者可以輕鬆進行設計，熟練使用這一軟體使電路設計的質量和效率大大提高。



圖 15 Altium Designer 軟體

### 3.4 Visual Studio 開發工具軟體

Microsoft Visual Studio 是微軟公司所開發的一套工具套件開發軟體，為一個基本且完整的開發工具集，Visual Studio 是一套方便、免費且功能強大的開發工具，可以將許多不同的混和語言建立在同一個開發平台上，包括了 Visual Basic、Visual C++、Visual C# 和 Visual J#。此外，也可以利用 .NET Framework 的功能，簡化 ASP Web 應用程式與 XML Web Services 開發的工作，下面介紹幾個 Visual Studio 常用的工具技術：

#### 1. Windows Form

Windows Form 是用來在 .NET Framework 上建立 Microsoft Windows 應用程式，他提供清楚的物件導向擴充工具集，讓使用者開發多樣的 Windows 應用程式，此外，Windows Form 還可做為多層分散式方案中的本機使用者介面。

#### 2. Web Form

Web Form 是用來建立可程式化網頁的 ASP.NET 技術，可以把程式碼轉譯為瀏覽器所相容的 HTML 與指令碼，可讓所有瀏覽器檢視網頁。使用 Web Form 時，只要將控制項拖放至設計工具，然後加入程式碼，就可以建立 Web 網頁，類似建立 Visual Basic 表單的方式。

#### 3. XML Web Service

XML Web Services 是可以使用 XML 透過 HTTP 接收要求和資料的



應用程式，XML Web Services 不會與特定物件呼叫慣例或元件技術繫結，因此可由語言、作業系統或元件模型存取。在 Visual Studio 中，可以使用 Visual Basic、Visual C#、JavaScript 或 ATL Server。

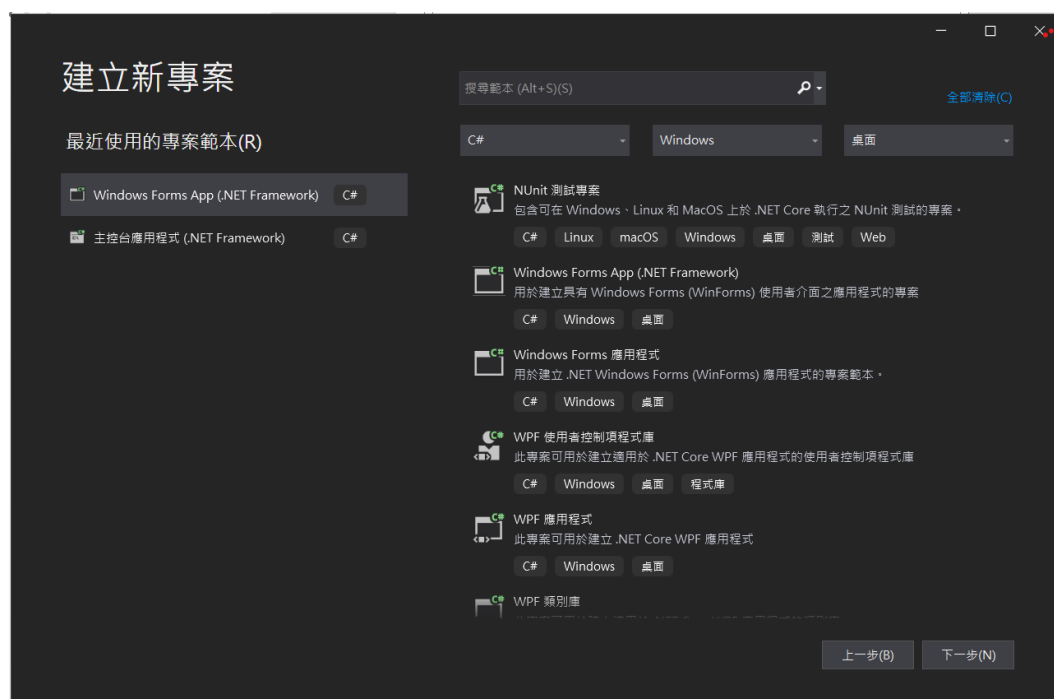


圖 16 VisualStudio

## 第四章實驗結果

### 4.1 電路

#### 4.1.1 兩軸電路測試

再利用 Altium designer 設計電路前，由於原驅動板之部分功能並未使用，經過研究及討論後，決定移除部分電路簡化電路及成本，並透過電木板與 IC 轉接板進行驅動電路焊接後，進行測試。

由於電路焊接後，發現驅動板上之異常指示燈偶爾會亮，透過 C# 介面之異常狀態表查看後，發現供電不足，經檢查後發現電路焊接電源處接觸不良，修正後即可正常使用所有功能，並能與原驅動板配合進行多軸控制。

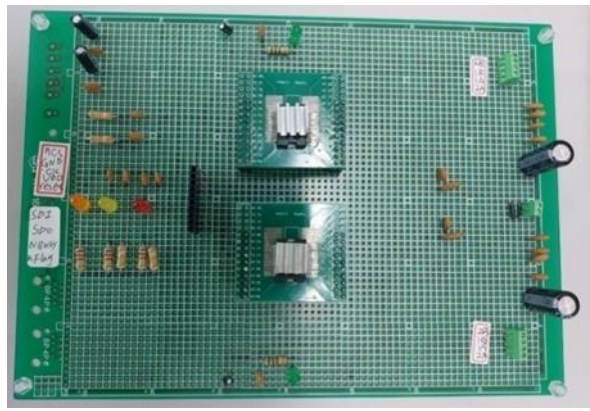


圖 17 兩軸電路電木板焊接測試

#### 4.1.2 Altium designer 設計電路

本次專題四軸步進馬達驅動之 PCB 利用 X-NUCLEO-IHM02A1 的 Gerber File 萃取成 PCB 檔進行參考與設計，其控制電路將 STM32F401RCT6 CORE BOARD 放置進下板設計中後，與步進馬達驅動電路進行整合，再參考 ARCUS 四軸馬達軸卡硬體的上板與下板配置，將步進馬達輸出線與極限開關接線，還有狀態指示燈與電源指示燈從上板拉出，再將上下板進行利用排針排母進行連接，由於使用雕刻機進行電路板雕刻，TOP layer 與 BOTTOM layer 的導孔需要進行焊接來導通，所以在導孔的設計部分有特殊規劃過，避免之後的電路功能出現問題。

極限開關在電源與觸發電位的要求與此系統電源不匹配，所以需要使用光隔離來進行電氣隔絕，故在上板增加光隔離電路用來確保極限開關的功能正常。



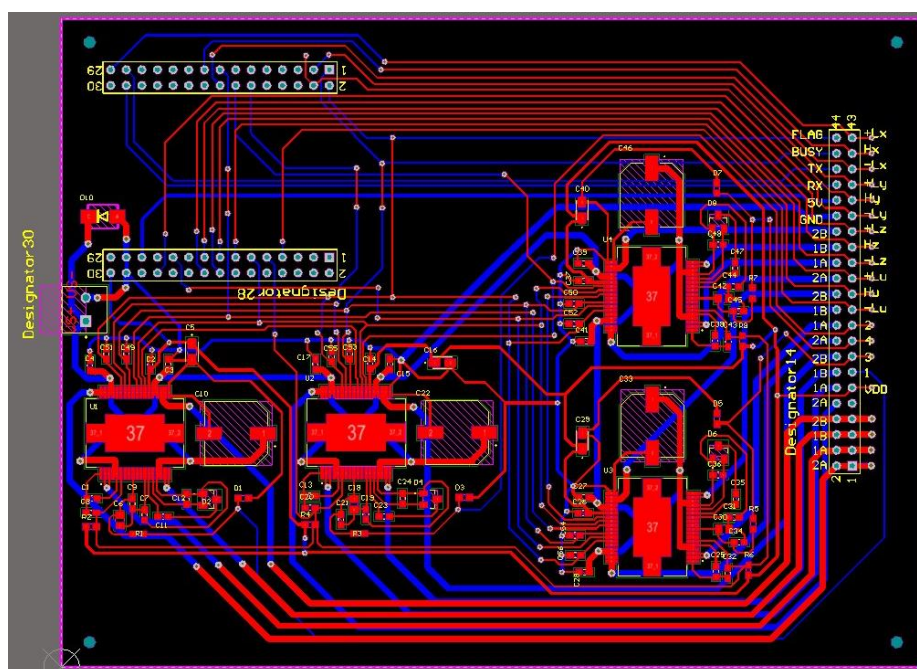


圖 18 Altium designer 設計電路下板

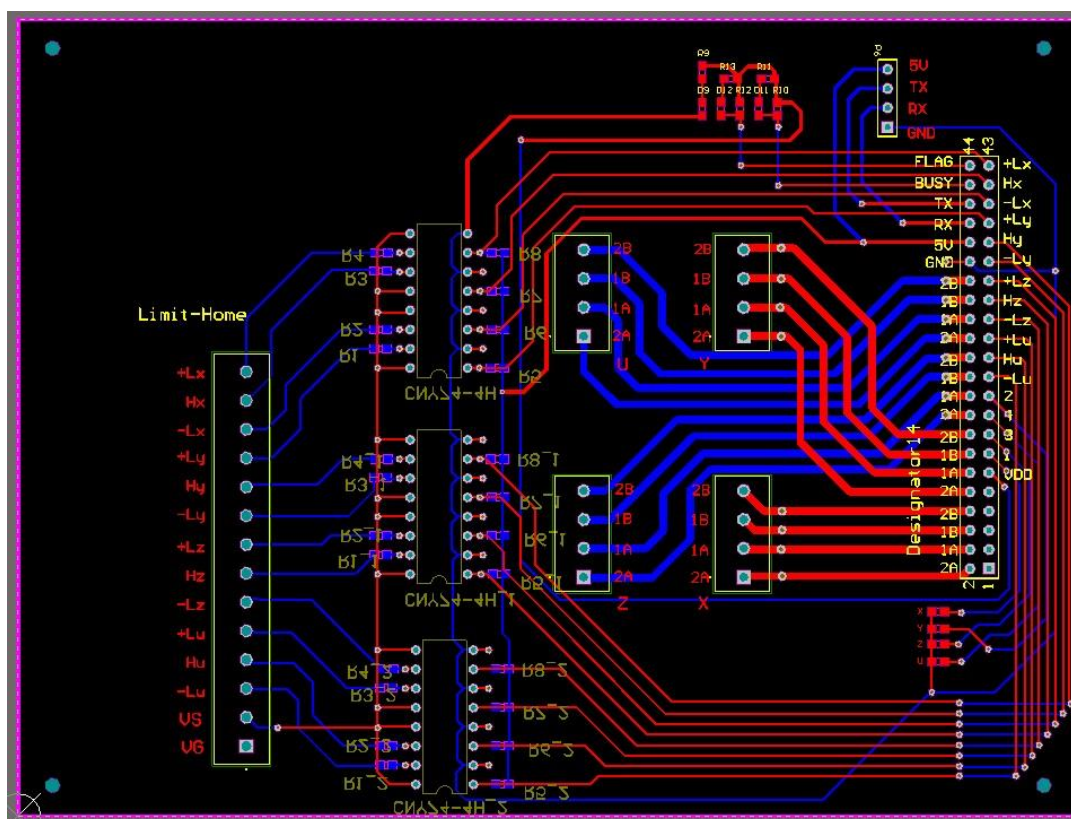


圖 19 Altium designer 設計電路上板

#### 4.1.3 實體圖

電路板設計完後，經由雕刻機進行電路板雕刻，雕刻後完成品如圖 20、圖 21，目前進度在電路板焊接，根據目前的觀察，發現可能會遇到下列以上問題：

1. 電路過於脆弱，焊接時容易發生損壞電路板
2. 雕刻後，有部分銅片未被刮除乾淨，容易發生短路之情形。
3. 由於電路設計相較於電路複雜，在量測及除錯時，較難排除異常。

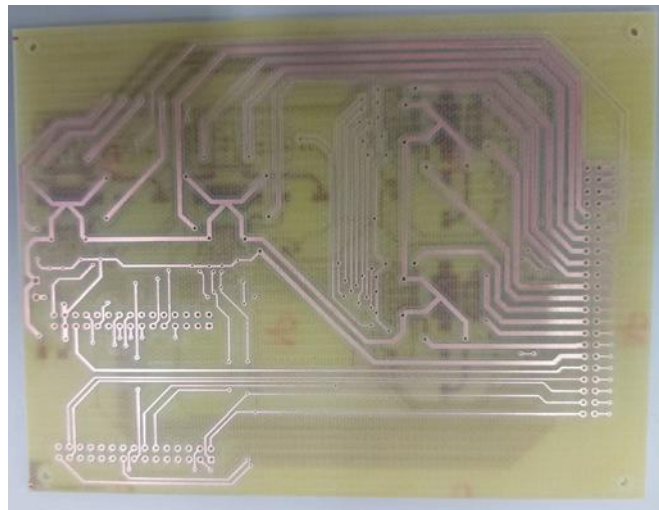


圖 20 雕刻板上板

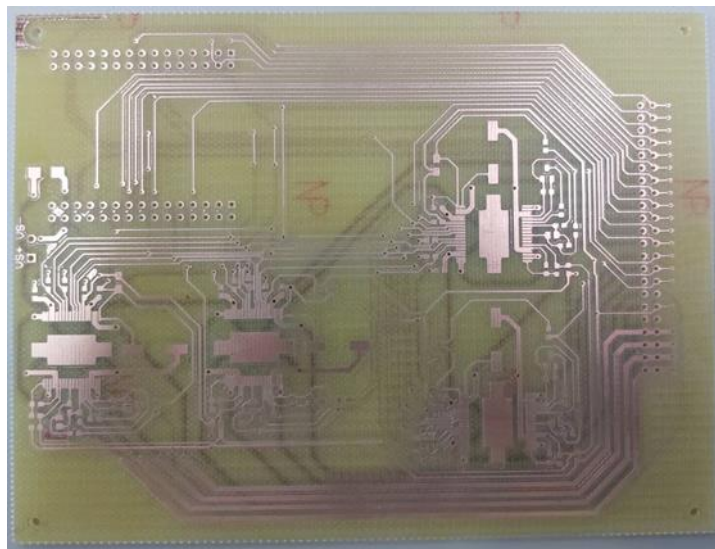


圖 21 雕刻板下板

## 4.2 動作指令執行

### 4.2.1 使用者介面介紹

為了讓使用者方便的對馬達進行控制，因此本文中設計了一套能控制多軸馬達的介面(如圖 22)，透過這個介面能夠實現以下幾種功能：

1. 各軸獨立控制
2. 多軸同時動作
3. 參數讀取與接收
4. 速度曲線圖
5. 狀態及異常指示
6. 位置限制及原點復歸

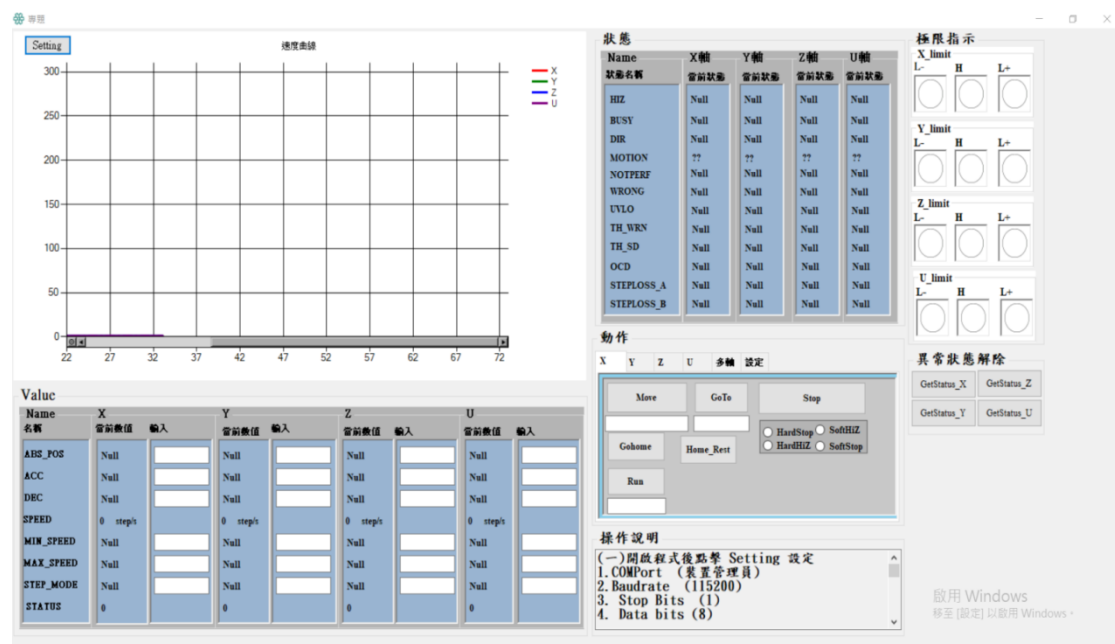


圖 22 C#操作介面

### 4.2.2 指令介紹

透過以下幾個指令，使馬達進行運動

1. Move: 移動指定的步數(順時針為正，逆時針為負)
2. GoTo: 以最短路徑到達指定之 ABS\_POS
3. Stop: 立即停止(速度瞬間歸 0)，減速度停止(以減速度方式讓速度為 0)，停止後再分為無激磁及激磁。
4. Gohome: 抵達 ABS\_POS 的位置。
5. Home\_Reset: 根據極限開關信號，重新尋找原點位置。

#### 4.2.3 各軸獨立

切換至要執行動作的軸，輸入目標參數即可動作。(如圖 23)  
以此圖為例，按下 Move 按鈕後，Z 軸將順時針移動 128000 微步。

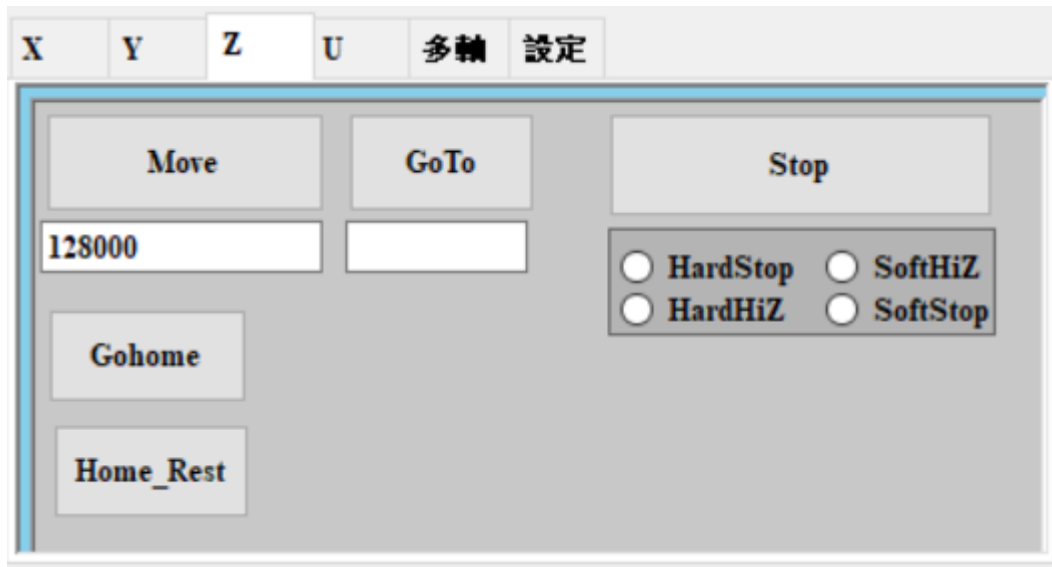


圖 23 單軸控制介面

#### 4.2.4 多軸同動

勾選多個或單個要執行動作的軸，即可進行動作(如圖 24)  
以此圖為例，按下 Goto 按鈕後，Z 軸將以最短路徑移動至絕對位置 12800。U 軸將以最短路徑移動至絕對位置-25600。



圖 24 多軸控制介面

#### 4.2.5 參數讀取與接收

透過讀取 L6470 內，對應暫存器(如圖 25)的值，了解馬達當前配置。也可以透過 C#修改數值，使馬達動作的參數改變，本次使用的馬達參數如下:

1. 當前位置 ABS\_POS
2. 加速度 ACC
3. 減速度 DEC
3. 當前速度 SPEED
4. 最低速度 MIN\_SPEED
5. 最高速度 MAX\_SPEED
6. 微步進 STEP\_MODE

Table 9. Registers map

Address [Hex]	Register name	Register function	Len. [bit]	Reset Hex	Reset value	Remarks (1)
h01	ABS_POS	Current position	22	000000	0	R, WS
h02	EL_POS	Electrical position	9	000	0	R, WS
h03	MARK	Mark position	22	000000	0	R, WR
h04	SPEED	Current speed	20	00000	0 step/tick (0 step/s)	R
h05	ACC	Acceleration	12	08A	125.5e-12 step/tick <sup>2</sup> (2008 step/s <sup>2</sup> )	R, WS
h06	DEC	Deceleration	12	08A	125.5e-12 step/tick <sup>2</sup> (2008 step/s <sup>2</sup> )	R, WS
h07	MAX_SPEED	Maximum speed	10	041	248e-6 step/tick (991.8 step/s)	R, WR
h08	MIN_SPEED	Minimum speed	13	000	0 step/tick (0 step/s)	R, WS
h16	STEP_MODE	Step mode	8	7	128 microsteps	R, WH

圖 25 驅動 IC 之暫存器對應表

初始化後各軸顯示預設初值，如圖。透過輸入欄可改變參數值，如圖 26(Z 軸及 U 軸)。

Name 名稱	X		Y		Z		U	
	當前數值	輸入	當前數值	輸入	當前數值	輸入	當前數值	輸入
ABS_POS	0	<input type="text"/>	0	<input type="text"/>	0	<input type="text"/>	0	<input type="text"/>
ACC	0 step/s <sup>2</sup>	<input type="text"/>	0 step/s <sup>2</sup>	<input type="text"/>	392 step/s <sup>2</sup>	<input type="text"/>	392 step/s <sup>2</sup>	<input type="text"/>
DEC	0 step/s <sup>2</sup>	<input type="text"/>	0 step/s <sup>2</sup>	<input type="text"/>	392 step/s <sup>2</sup>	<input type="text"/>	392 step/s <sup>2</sup>	<input type="text"/>
SPEED	0 step/s	<input type="text"/>	0 step/s	<input type="text"/>	0 step/s	<input type="text"/>	0 step/s	<input type="text"/>
MIN_SPEED	0 step/s	<input type="text"/>	0 step/s	<input type="text"/>	0 step/s	<input type="text"/>	0 step/s	<input type="text"/>
MAX_SPEED	0 step/s	<input type="text"/>	0 step/s	<input type="text"/>	305 step/s	<input type="text"/>	305 step/s	<input type="text"/>
STEP_MODE	1	<input type="text"/>	1	<input type="text"/>	128	<input type="text"/>	128	<input type="text"/>
STATUS	0	<input type="text"/>	0	<input type="text"/>	32258	<input type="text"/>	32274	<input type="text"/>

圖 26 參數設定介面



Value								
Name 名稱	X		Y		Z		U	
	當前數值	輸入	當前數值	輸入	當前數值	輸入	當前數值	輸入
ABS_POS	0	<input type="text"/>	0	<input type="text"/>	12800	<input type="text"/>	25600	<input type="text"/>
ACC	0 step/s <sup>2</sup>	<input type="text"/>	0 step/s <sup>2</sup>	<input type="text"/>	72step/s <sup>2</sup>	<input type="text" value="80"/>	392 step/s <sup>2</sup>	<input type="text"/>
DEC	0 step/s <sup>2</sup>	<input type="text"/>	0 step/s <sup>2</sup>	<input type="text"/>	392 step/s <sup>2</sup>	<input type="text"/>	392 step/s <sup>2</sup>	<input type="text"/>
SPEED	0 step/s	<input type="text"/>	0 step/s	<input type="text"/>	0 step/s	<input type="text"/>	0 step/s	<input type="text"/>
MIN_SPEED	0 step/s	<input type="text"/>	0 step/s	<input type="text"/>	0 step/s	<input type="text"/>	0 step/s	<input type="text"/>
MAX_SPEED	0 step/s	<input type="text"/>	0 step/s	<input type="text"/>	305 step/s	<input type="text"/>	305 step/s	<input type="text"/>
STEP_MODE	1	<input type="text"/>	1	<input type="text"/>	128	<input type="text"/>	128	<input type="text"/>
STATUS	0	<input type="text"/>	0	<input type="text"/>	32274	<input type="text"/>	32274	<input type="text"/>

圖 26 參數設定介面(設定後)

#### 4.2.6 速度曲線圖

馬達運轉時，會將馬達的速度變化以曲線圖表示(如圖 27)

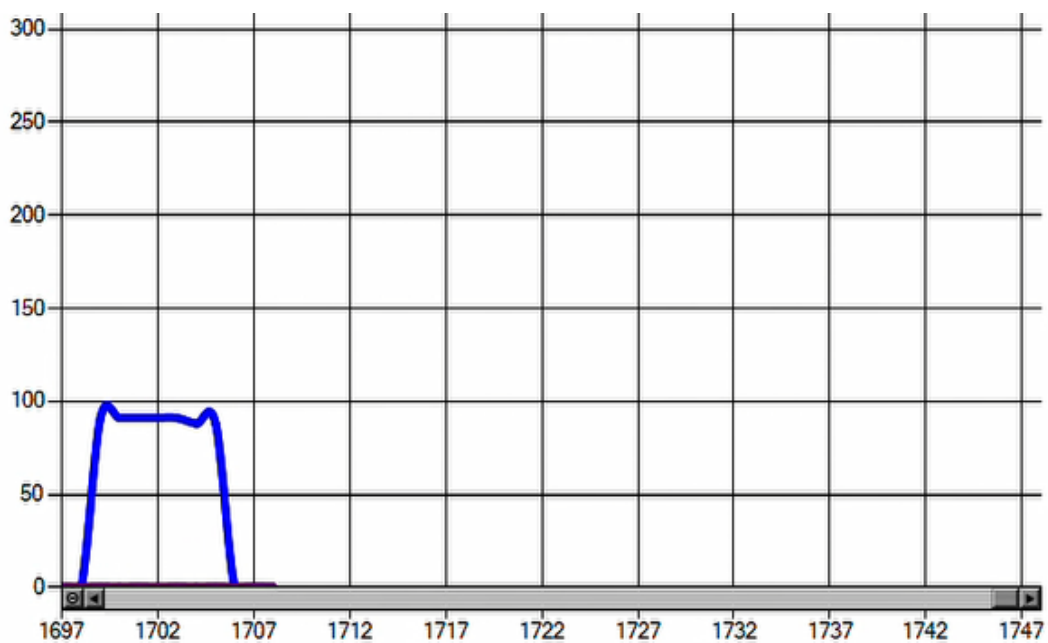


圖 27 速度曲線圖

#### 4.2.7 狀態及異常指示

透過讀取 L6470 內，STATUS register 對應位元的值，了解馬達當前狀態，並判斷是否異常。

Table 34. STATUS register

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
SCK_MOD	STEP_LOSS_B	STEP_LOSS_A	OCD	TH_SD	TH_WRN	UVLO	WRONG_CMD
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
NOTPERF_CMD	MOT_STATUS		DIR	SW_EVN	SW_F	BUSY	HIZ

圖 28 狀態暫存器

1. HIZ:馬達是否激磁。
2. BUSY:馬達是否運轉。
3. DIR:馬達運轉方向如圖 29。
4. MOTION:馬達當前狀態(停止、加速運轉、減速運轉、定速運轉)如圖 30。
5. NOTPERF:馬達運轉中寫入指令，導致異常。
6. WRONG:指令缺少參數或參數錯誤。
7. UVOL:電壓過低(6.6V~7.8V)。
8. TH\_WRN:裝置內部到達 130°C。
9. TH\_SD:裝置內部到達 160°C (會強制鎖住馬達的驅動電路)。
10. OCD:電流過大(可自行設定預設 3.38A)。
11. STEPLOSS\_A 和 STEPLOSS\_B:馬達失步。

Table 35. STATUS register DIR bit

DIR	Motor direction
1	Forward
0	Reverse

圖 29 方向位元對應表

Table 36. STATUS register MOT\_STATE bits

MOT_STATUS		Motor status
0	0	Stopped
0	1	Acceleration
1	0	Deceleration
1	1	Constant speed

圖 30 運動狀態位元對應表

初始化後，讀取馬達狀態，如圖 31。當發生異常時，如圖 32，會顯示異常問題，需進行故障排除後按下異常狀態解除，方可繼續控制，如圖 33。

狀態				
Name	X軸	Y軸	Z軸	U軸
狀態名稱	當前狀態	當前狀態	當前狀態	當前狀態
HIZ	N High Z	N High Z	N High Z	High Z
BUSY	忙碌中	忙碌中	動作完成	動作完成
DIR	逆時針	逆時針	順時針	逆時針
MOTION	停止	停止	停止	停止
NOTPERF	正常	正常	正常	正常
WRONG	正常	正常	正常	正常
UVLO	不正常	不正常	正常	正常
TH_WRN	不正常	不正常	正常	正常
TH_SD	不正常	不正常	正常	正常
OCD	不正常	不正常	正常	正常
STEPLOSS_A	不正常	不正常	正常	正常
STEPLOSS_B	不正常	不正常	正常	正常

圖 31 當前狀態指示介面(Z、U 軸為例)

狀態				
Name	X軸	Y軸	Z軸	U軸
狀態名稱	當前狀態	當前狀態	當前狀態	當前狀態
HIZ	N High Z	N High Z	High Z	High Z
BUSY	忙碌中	忙碌中	動作完成	動作完成
DIR	逆時針	逆時針	順時針	逆時針
MOTION	停止	停止	停止	停止
NOTPERF	正常	正常	正常	正常
WRONG	正常	正常	正常	正常
UVLO	不正常	不正常	不正常	不正常
TH_WRN	不正常	不正常	正常	正常
TH_SD	不正常	不正常	正常	正常
OCD	不正常	不正常	正常	正常
STEPLOSS_A	不正常	不正常	正常	正常
STEPLOSS_B	不正常	不正常	正常	正常

圖 32 當前狀態指示介面異常時(Z、U 軸為例)



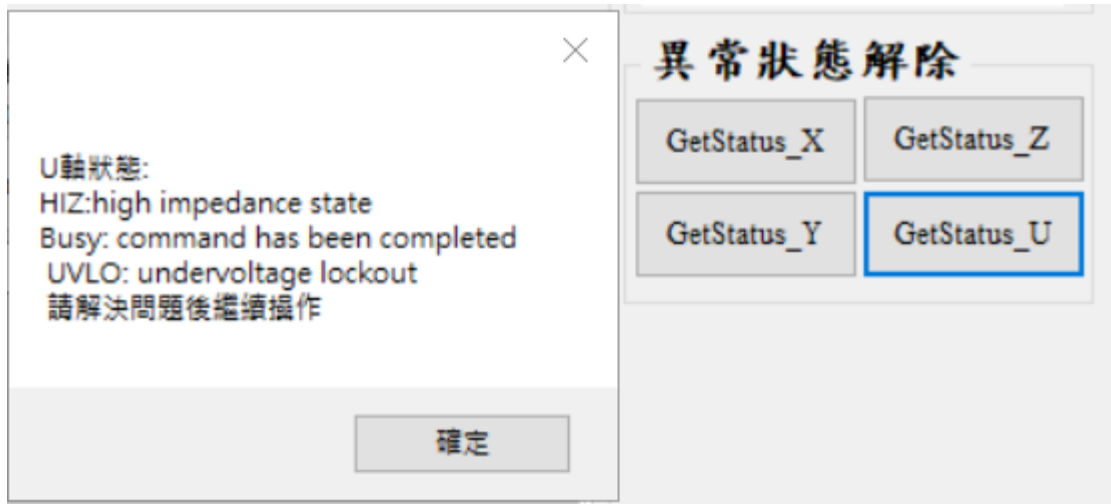


圖 33 異常狀態解除鈕

#### 4.2.8 極限開關觸發

利用極限開關達到(1)位置限制，(2)原點復歸，以實現精準的位置控制。因此會有一般模式及復規模式，一般模式時，原點之極限開關無任何功能，在復規模式時原點之極限開關須觸發兩次才可完成復歸。

##### 4.2.8.1 位置限制

正常狀態下極限指示燈，如圖 34 (右)。運轉中若觸碰到左右極限時馬達立即停止，此時指示燈亮起紅燈如圖 34 (左)，點擊紅燈後，馬達往原點方向轉十分之一圈，離開極限位置，指示燈轉為綠燈，如圖 35。

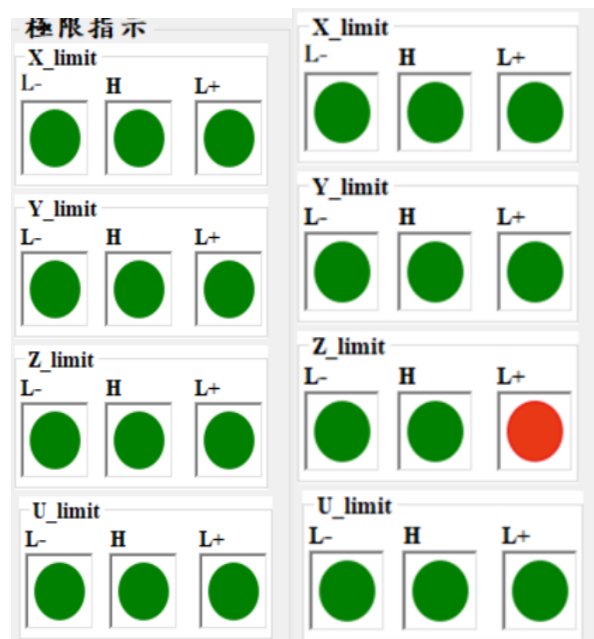


圖 34 極限觸發指示燈，(右) 未觸發 (左)觸發

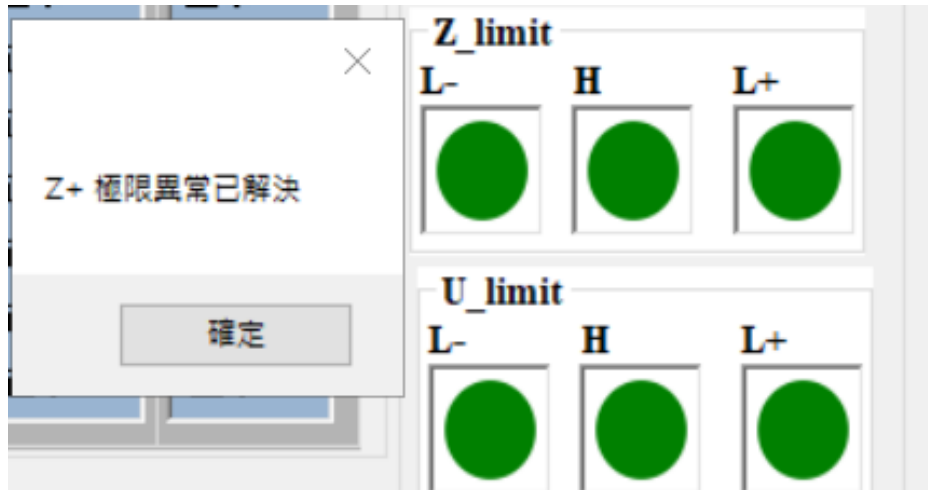


圖 35 復歸通知

#### 4.2.8.2 原點復歸

按下 Home\_Reset 後(圖 36)，根據設定中勾選模式(圖 37)，進行原點復歸，原點復歸方式根據終點位置及初始運動方向分為以下四種:



圖 36 原點復歸鈕



圖 37 模式設定勾選

一. 初始運動方向:左終點位置:左 (如圖 38)

狀況 A:

- 1.逆時針轉至 L-後停止，
- 2.在順時針轉至 Home 後停止
- 3.逆時針轉 0.5 圈
- 4.順時針轉至 Home 後停止，完成復歸

狀況 B:

- 1.逆時針轉至 Home 後停止
- 3.逆時針轉 0.5 圈
- 4.順時針轉至 Home 後停止，完成復歸

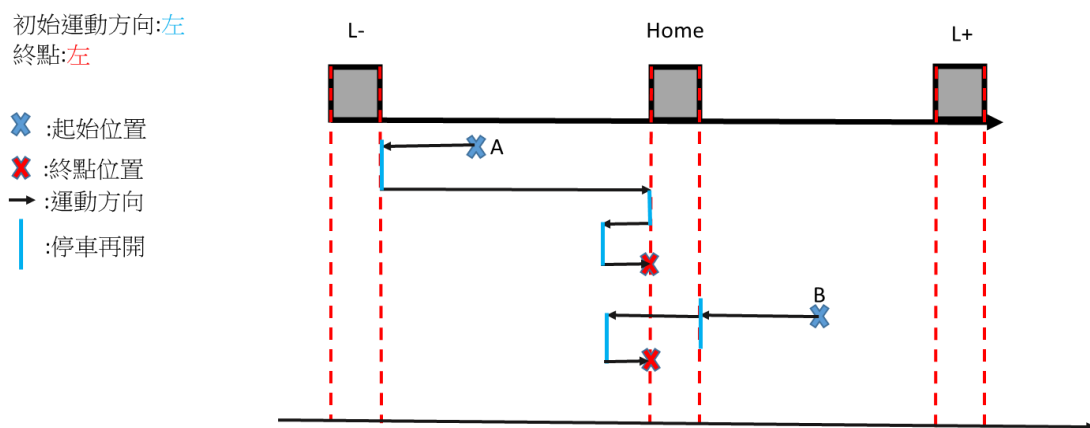


圖 38 原點復歸行程圖(一)

二. 初始運動方向:左終點位置:右 (如圖 39)

狀況 A:

- 1.逆時針轉至 L-後停止，
- 2.順時針轉至 Home 後停止
- 3.順時針轉 0.5 圈
- 4.逆時針轉至 Home 後停止，完成復歸

狀況 B:

- 1.逆時針轉至 Home 後停止
- 2.順時針轉 0.5 圈

### 3.逆時針轉至 Home 後停止，完成復歸

初始運動方向:左

終點位置:右

✕:起始位置

✕:終點位置

→:運動方向

|:停車再開

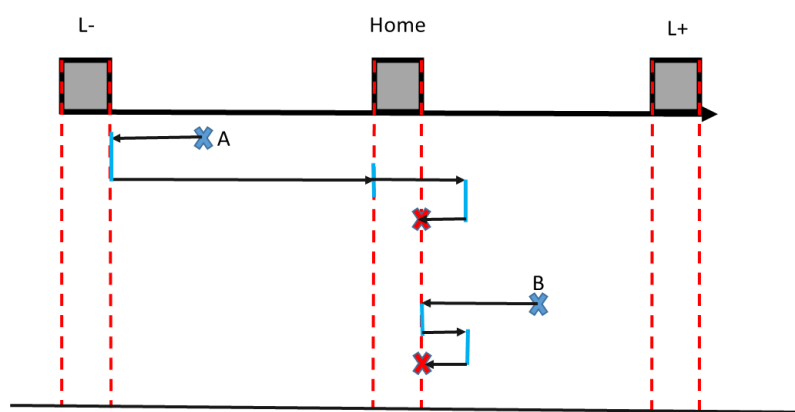


圖 39 原點復歸行程圖(二)

三.初始運動方向:右終點位置:左 (如圖 40)

狀況 A:

- 1.順時針轉至 Home 後停止，
- 2.逆時針轉轉 0.5 圈
- 3.順時針轉至 Home 後停止，完成復歸

狀況 B:

- 1.順時針轉至 L+後停止
- 3.逆時針轉至 Home 後停止
- 2.逆時針轉轉 0.5 圈
- 3.順時針轉至 Home 後停止，完成復歸

初始運動方向:右

終點位置:左

✕:起始位置

✕:終點位置

→:運動方向

|:停車再開

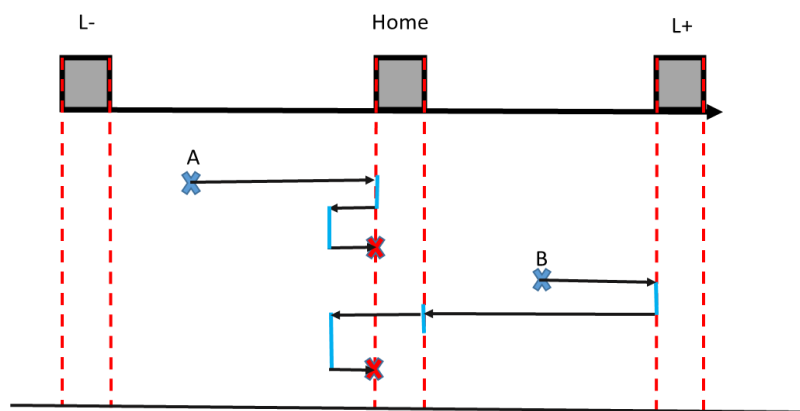


圖 40 原點復歸行程圖(三)

四.初始運動方向:右終點位置:右 (如圖 41)

狀況 A:



## 第五章結論與未來展望

### 5.1 結論

本實驗透過義法半導體之驅動板與開發板，進行高階多軸微步進馬達之位置控制，並透過 C# 設計人機介面，讓使用者能利用一目了然的介面進行控制，並透過極限開關的負回授系統，限制馬達位置，防止超出工作範圍，對於安全性及儀器之壽命有相當的好處。接著透過 Altium Designer 設計電路板，將所有系統整合，以完成本次實驗。

步進馬達系統相關參數表			
參數名稱	數值範圍	參數名稱	數值範圍
工作電壓	8-45V	位置暫存器	$-2^{21} \sim 2^{21} - 1$
微步進解析度	Full step1/128	減速度	14.55-59590step/s <sup>2</sup>
加速度	14.55-59590step/s <sup>2</sup>	最高速度	15.25-15610step/s
最低速度	0-976.3step/s	鮑率	115200 bits/s

專題實務預期進度	已完成	未完成
資料查詢		
微控制器之程式設計		
馬達驅動電路整合與繪製		
C#設計及程式撰寫		
驅動電路之實現與測試		
電路焊接及測試		

### 5.2 未來展望

- 未來希望能夠減少電路板面積，更容易安裝，佔用空間更小，對後續應用之便利性提升。
- 加入更多感測器，使其能應用範圍更加廣泛。
- 改善電路使其能控制不同規格步進馬達，以應用在各式各樣的需求。

## 參考文獻

1. B.C. Kuo, Theory and Applications of the Step Motors. New York: West Publishing Co., 1991.
2. P. Acarnley, Stepping Motors: a Guide to Modern Theory and Practice, 4th ed., Stevenage, UK: IET, 2002, pp.48–51.
3. T. Kenjo and A. Sugawara, Stepping Motors and Microprocessor Control, 2nd ed., Oxford, UK: Clarendon Press, 2003.
4. 黃鈞成, “雙軸線性致動器同步控制研究”, 碩士論文, 國立雲林科技大學電機工程系, 2015。
5. 徐偉傑, “測微器結合小型步進馬達結構開發”, 碩士論文, 國立勤益科技大學機械工程系碩士班, 2015。
6. 黃正杰, “二相步進馬達多軸同動控制系統設計與研製”, 碩士論文, 國立交通大學電控工程研究所, 2016。
7. 吳信成, “X-Y 平台及時運動控制系統的設計與實現”, 碩士論文, 國立成功大學工程科學系碩博士班, 2006。。
8. 蔡炳昌, “影像伺服於 XXY 平台控制之應用”, 碩士論文, 國立雲林科技大學電機工程系, 2014。
9. UART 通訊介面, <https://www.circuitbasics.com/basics-uart-communication/>
10. SPI 周邊介面, <https://magicjackting.pixnet.net/blog/post/164725144/>
11. start [USART HMI 資料中心], <http://wiki.tjc1688.com/doku.php>
12. 意法半導體公司官方網站, [https://www.st.com/content/st\\_com/en.html](https://www.st.com/content/st_com/en.html)
13. 白中和, 步進馬達控制電路設計, 台北, 建興出版社, 2000

。