

## 時尚科技文創商品—微型木作機械手臂產品之實作與開發

楊吉仕<sup>1\*</sup>

<sup>1</sup>黎明技術學院時尚設計系

\*Email: yjs@mail.lit.edu.tw

### 摘要

近年來「物聯網(IoT)」、「智慧製造(Smart Manufacturing)」及「文創商品(Cultural and Creative Goods)」等概念風靡了產業界與工商業,前二者需具備諸多高度專業知識與技術,其中又以機械手臂(Robot Arm)最具主流代表性;本研究係結合機械手臂與文創商品,針對「時尚科技文創商品微型木作機械手臂產品之實作與開發」進行硬體開發試製與技術內涵研究,成果將以具文創風及科技感之木作微電腦控制機械手臂來展現其娛樂療癒功能,並依本研究之研發歷程,將技術報告依序以「雷雕機硬體組裝」、「雷雕機軟體控制」、「機械手臂概念設計」、「組件雷雕路徑繪製」、「Arduino 電路配線」、「Arduino 程式設計」、「機械手臂運動學分析」及「Denavit-Hartenberg法則應用探討」等章節來介紹。

**關鍵字:**物聯網、智慧製造、文創商品、機械手臂、Arduino微電腦控制、雷雕機、Denavit-Hartenberg法則、座標轉換矩陣、時尚科技、療癒功能

### 1. 前言

本研究「時尚科技文創商品微型木作機械手臂產品之實作與開發」之技術論文,依據實際研發歷程,製作出第一代與第二代機械手臂,茲將各執行重點摘要說明如下各章節:

- 雷雕機硬體組裝:介紹雷雕機之重要元件、型號與規格、電路主機板、組裝過程、精度校正等細節。
- 雷雕機軟體控制:介紹GBRL雷雕機之主板、專用軟體、視窗畫面及系統參數設定等。
- 機械手臂概念設計:介紹設計概念、自由度分析、細節考量等。
- 組件雷雕路徑繪製:介紹第一代與第二代機械手臂之雷雕路徑繪製細節與設計重點。
- Arduino電路配線:介紹第一代與第二代機械手臂之Arduino主機板、步進馬達、夾爪伺服機、液晶顯示幕...等組件之連接配線圖。
- Arduino程式設計:介紹第一、二代機械手臂Arduino微電腦C語言程式。
- 機械手臂運動學分析:針對機器人運動學,介紹其座標轉換數學基礎、傳統三角函數解析法、由馬達角度求夾爪座標之順向分析、或由夾爪座標來求馬達角度之逆向分析等。

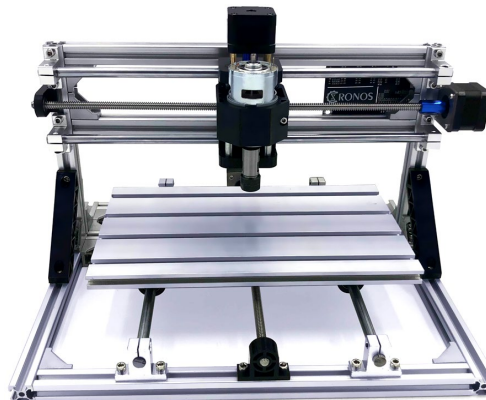
9. Denavit-Hartenberg法則應用探討:針對機械手臂所提出數學分析模式,按各機構關節之相對位置,計算四種D-H參數,以利座標轉換計算。

### 2. 雷雕機硬體組裝

木作商品規劃使用紙質纖維密集板作為原料,為能切割3mm木板,首先須製作雷雕機當作基本切割板材之工具;本案挑選型號CNC3018 mini之桌上型雷射雕銑複合機,並搭配15W藍光晶體雷射頭,其主要配件細部規格如表一,全機外觀如圖一所示:

表一: CNC3018雷射雕銑複合機細部規格

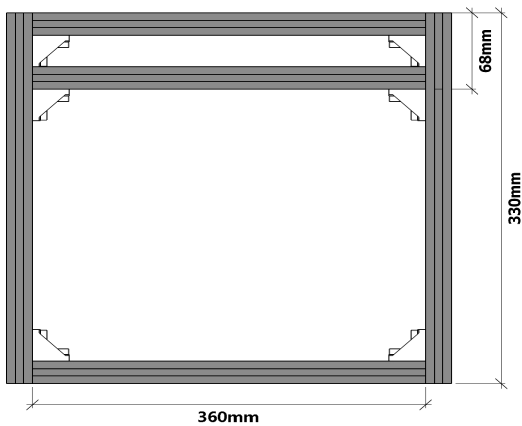
尺寸	400 x 330 x 240 mm
行程	300 x 180 x 45 mm
導螺桿	T8 牙, 導程 4mm(雙銅螺母消回差)
步進馬達	二相四線(42)、電壓 12V、相電流 1.3A、相電阻 2.1、力矩 0.3N.m
主軸晶體雷射	電源 DC12V、波長 450nm 藍光、功率 15W、風冷、焦距 1.8cm、工作溫度 -40~75 攝氏度、PWM 控制電壓(0V 關~5V 開)、控制頻率 0~30Khz、可切割 3mm 木板、可在不鏽鋼或有色合金表面刻字、壽命 1 萬小時
主軸馬達	775(12~36V)、常規 24V 時 1 萬轉、前後雙滾珠軸承
電源	變壓器輸入 AC110~240V、輸出 DC24V 5A
機架	鋁擠型 2020 型材
控制主板	型號 A4988, 電源直流 7~36V, 銑削及雷射加工二用, 附三軸步進驅動模組, 驅動電流可調, 多種微步進模式
脫機控制板	包含機板、TF 卡插槽、1.8 吋液晶顯示幕、按鍵等, 具中英文雙介面(長按 Z+ 切換), 主要功能: 手動控制、文件離線雕刻、儲存文件(當作 USB 隨身碟使用)
步進馬達驅動模組	型號 A4988, 驅動 3~35V, 2A 以下馬達, 五種模式(全、半、1/4、1/8 和 1/16), 可變電阻調節最大電流, 過熱關閉及短路保護



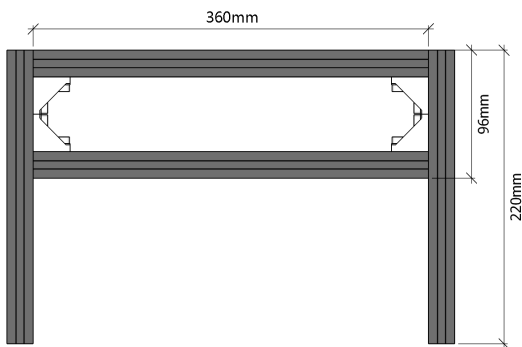
圖一: CNC3018雷射雕銑複合機外觀圖

## 2.1 雷射雕銑複合機之鋁合金機台組裝

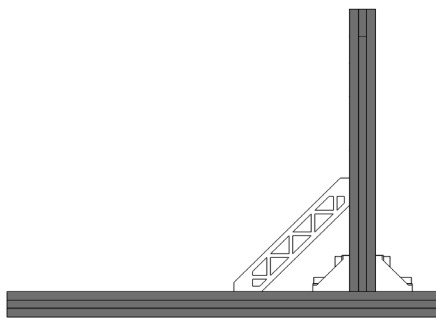
該雷射機係中國河南鄭州之河南倫業(LunYee)進出口貿易有限公司所販售之DIY套件商品，因貨到時為散裝零件，尚需自行組裝；首先進行機架安裝[1]，包含水平底座鋁擠型組立(鋁擠型330mm二根、360mm三根、角鋁六個、M5×10螺絲十二顆、墊片十二個，如圖二)、垂直龍門架鋁擠型組立(鋁擠型220mm二根、鋁擠型300mm二根、角鋁四個、M5×10螺絲八顆、墊片八個，如圖三)、機架結合組立(角鋁六個、M5×10螺絲十六顆、墊片十二個、塑膠把手二個，如圖四)、水準底座Y軸光桿及支撐座組立(支撐座四個、光桿二枝、M5×10螺絲八顆，如圖五)。



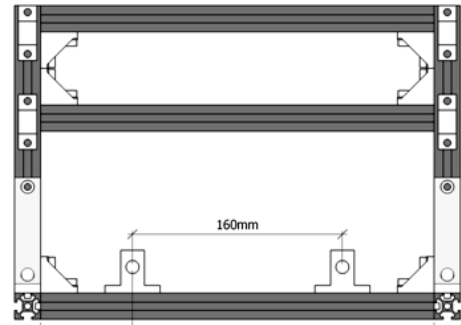
圖二：水平底座鋁擠型俯視圖



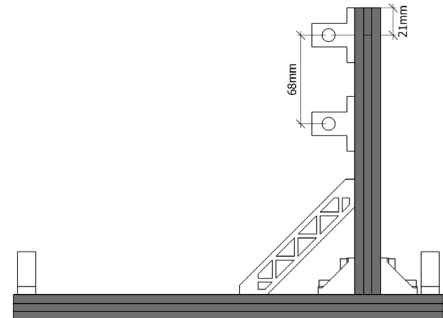
圖三：垂直龍門架鋁擠型前視圖



圖四：機架結合右側視圖

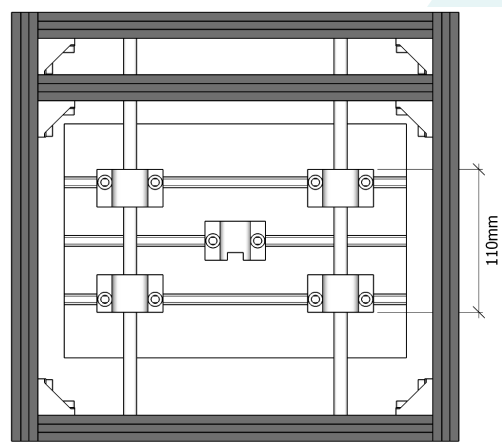


圖五：水平底座Y軸光桿及支撐座前視圖

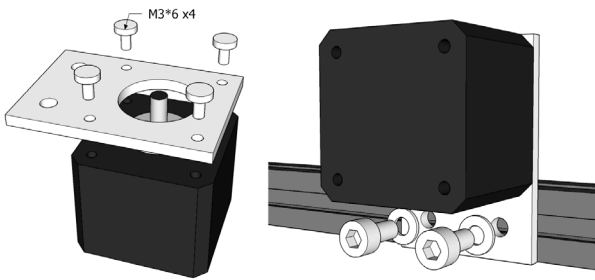


圖六：垂直龍門架X軸光桿及支撐座右側視圖

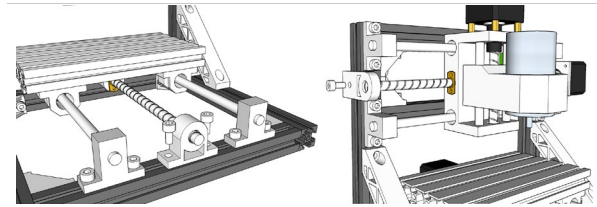
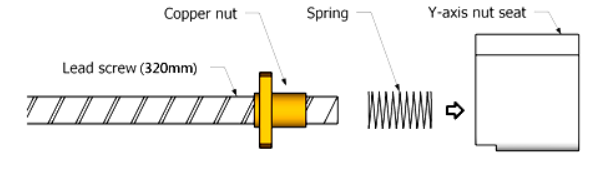
接著進行垂直龍門架X軸光桿及支撐座組立(支撐座四個、光桿二枝、M5×10螺絲八顆，如圖六)、工作檯面組立(鋁擠型工作檯面一片、M6×10螺絲十顆、光桿滑塊五個，如圖七)、X及Y軸步進馬達安裝(馬達二顆、馬達安裝板二片、M3×6螺絲八顆、M5×10螺絲四顆、墊片四個，如圖八)、Z軸加工總成於X軸光桿上組立(Z軸加工總成一組，安裝時需保證X軸二光桿互相平行，如圖九)、X及Y軸導螺桿及軸承座安裝(軸承座四個、M5×10螺絲八顆、導螺桿二枝、銅螺母套筒二個、緩衝彈簧2個，如圖十)，最後則是CRONOS公司之GRBL控制主機板安裝，如圖十一。



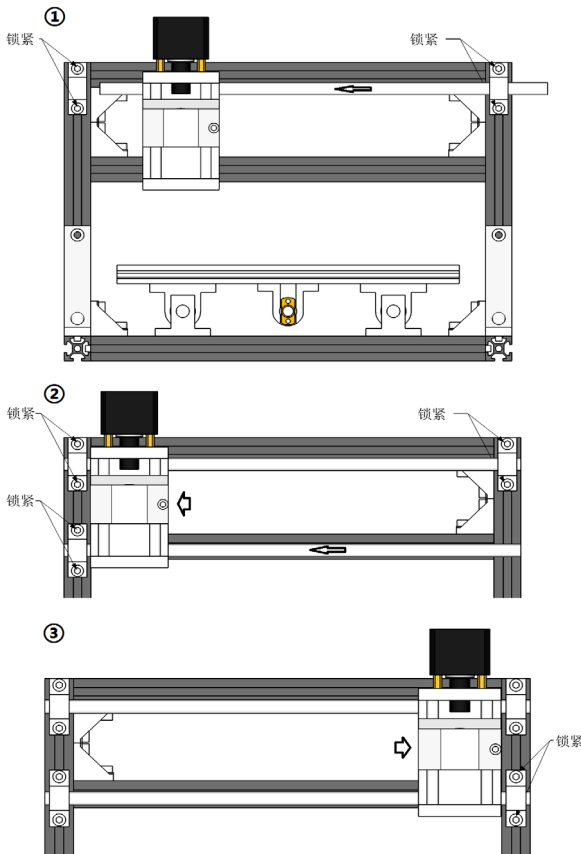
圖七：工作檯面組立底視圖



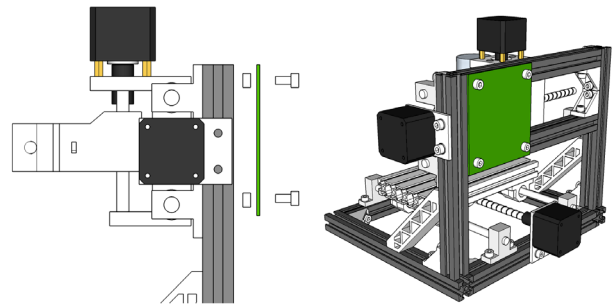
圖八：X及Y軸步進馬達安裝圖



圖十：X及Y軸導螺桿及軸承座安裝圖



圖九：Z軸加工總成於X軸光桿上組立之前視圖



圖十一：GRBL控制主機板安裝圖

## 2.2 主軸馬達及藍光晶體雷射組裝

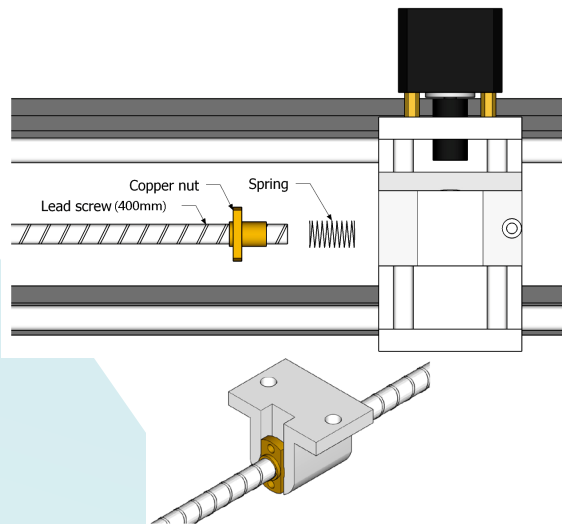
雷射雕銑複合機隨附有一顆以12~36V供電之主軸馬達，以及一顆以12V供電、5V PWM控制之藍光晶體雷射，可視加工現場需求安裝主軸馬達或藍光晶體雷射，Z軸上的主軸安裝座均可緊密容納馬達或雷射，主軸馬達775外觀如圖十二，藍光晶體雷射外觀如圖十三，裝入Z軸組好之外觀如圖十四所示。



圖十二：主軸馬達775外觀圖



圖十三：15W藍光晶體雷射外觀圖

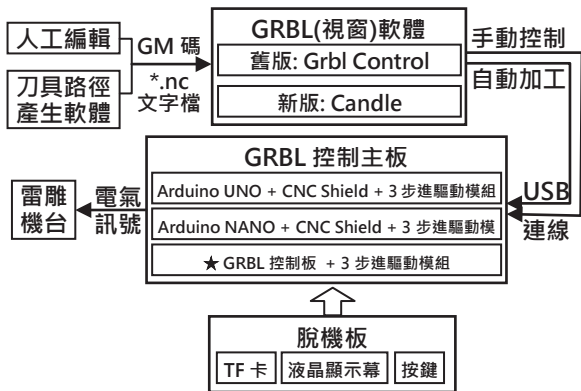




圖十四：主軸馬達與Z軸之組裝圖

### 3. 雷射機軟體控制

欲了解雷射機之控制方式，首先說明GRBL軟體與控制主板之間的流程關係，如圖十五所示。



圖十五：GRBL軟體與控制主板之控制流程圖

使用者可用人工文書編輯或以刀具路徑軟體自動產生文字格式GM碼(GM Code)之NC檔，再用GRBL軟體讀取此NC檔，該軟體舊版叫Grbl Control(ler)[3]，新版稱為Candle，其二大功能一為負責將GM碼轉成馬達控制之電氣訊號，二為在視窗介面上直接手動控制機台；控制主板通常有三個微型步進驅動模組附加其上，亦有廠商以Arduino微處理板當作主板使用，將特殊韌體程式用Arduino Builder燒錄至Arduino內，如圖十五之三種控制主板。此外，因為長時間加工為了省電緣故，甚至可加裝「脫機板」來擺脫使用筆電或桌機之限制，以利讀取TF卡之NC檔，其外觀如下圖十六所示。

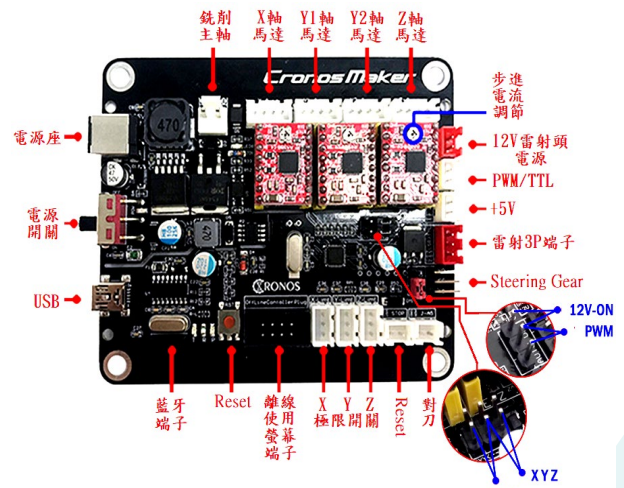


圖十六：脫機(離線)控制板之外觀圖

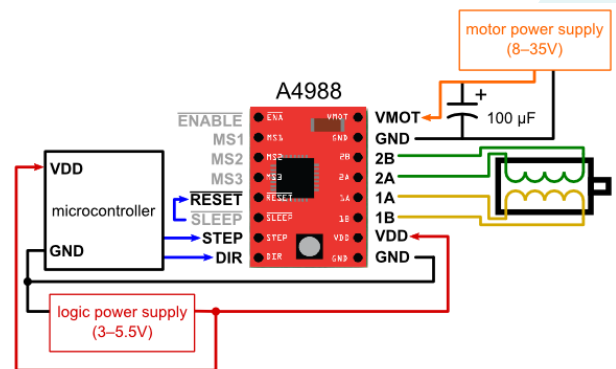
### 3.1 雷射雕銑複合機之控制主板介紹

CNC3018雷射雕銑複合機採用Cronos Maker公司所生產之GRBL控制主板(型號GRBL ATmega328P-MU-V11)[2]，其主要功能係接收GRBL軟體或脫機控制板之運行GM碼或手控命令所轉換之電氣訊號，傳達並驅動諸如：各軸步進馬達、PWM藍光晶體雷射或主軸馬達等周邊，並可接收安裝於三軸上六顆極限開關之警示電壓訊號；同時考慮到龍門雷射機應用之可能性，Y軸可同時指揮Y1及Y2軸二顆步進馬達，並附加對刀及藍牙傳輸所需之端子，其控制主板周邊之配電連接關係如圖十七所示。

此外，若僅當作雕銑機時，輸入控制主板之直流電源電壓關係著其加工能力，本研究所使用控制主板之直流電源電壓，其接受範圍廣達7~36V，表示最大能推動36V之銑削主軸馬達，大幅增加銑削扭矩。



圖十七：控制主板與周邊之配電關係圖



MS1	MS2	MS3	Microstep Resolution	Excitation Mode
L	L	L	Full Step	1 2 Phase
H	L	L	Half Step	1/2 1-2 Phase
L	H	L	Quarter Step	1/4 W1-2 Phase
H	H	L	Eighth Step	1/8 2W1-2 Phase
H	H	H	Sixteenth Step	1/16 4W1-2 Phase

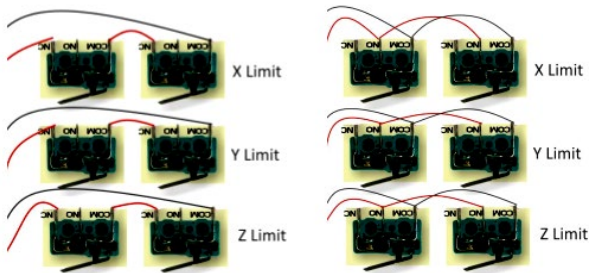
圖十八：步進馬達驅動模組接腳圖以及微步進細分設置參照圖

另有三片步進馬達驅動模組(型號A4988)，係安



裝於控制主板上層，其特色為具有微步進模式，可增加馬達進給之精度；通常2A、2B為一組接馬達紅、綠線，1A、1B為一組接馬達藍、黃線，其微步進應用細節如圖十八所示，下方之主板有三組對應之短路插腳，每組短路帽從上往下依次是MS3、MS2、MS1(驅動模組上下顛倒插入，高度位於主板上，下層短路插腳均被上層之驅動模組擋到，且短路帽插上為H、拔除為L)。如果2相4極步進馬達採用1/2半步進模式，走完一圈將由原來200步倍增至400步，同時該驅動模組還附有半固定可變電阻，以便來調整驅動馬達驅動電流之大小。

此外，為防止各軸自撞，每軸兩端建議安裝二極限開關保護之，如圖十九所示。控制主板的下方有三個X、Y、Z軸極限開關之插槽，每個插槽有三根插腳，由上至下分別代表+(5V)、-(GND)與S(極限開關信號)，其中極限開關信號S與GND需接在極限開關之二側；安裝於各軸兩側之二類極限開關有串聯及並聯二種接線方式，串聯乃接線到二極限開關之COM及NC腳，並聯乃接線到二極限開關之COM及NO腳。



圖十九：極限開關串(左)及並聯(右)二種接線示意圖

本章節開頭所提及之「脫機板」，所附按鍵意義如前圖十六，若將其排線插入控制主板上層插槽時，會致使桌機或筆電與控制主板上層之連接失效，此時脫機板取得主導控制權；在甫開機狀態下，如圖二十所示，主頁(Main Page)會顯示手控(Ctrl)、檔案(File)、速率(Sset)等三個主選單，表示脫機板可手動控制三軸運動、選擇SD卡之NC檔案進行加工或設定進給速率及傳輸速率。

若按下Ctrl手控頁(Move Page)，如圖二十一所示，則可手動控制XYZ三軸正負方向運動、主軸馬達啟閉或停止；若按下File檔案頁(File Select)，如圖二十二所示，則可選擇SD卡之NC檔、確認開始加工或監看加工進度百分比；若按下Sset速率頁(Setting)，如圖二十三所示，則可設定進給速率為每轉多少毫米或設定傳輸速率。



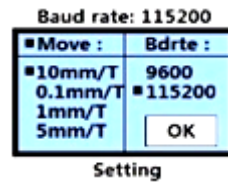
圖二十：脫機板液晶之主頁畫面

功能	面板按鍵
XYZ三軸移動	藍色按鈕
主軸開/關	中央紅色按鈕
步進	右下紅色按鈕
畫面返回	長按右下紅色按鈕

圖二十一：脫機板液晶之手控頁畫面

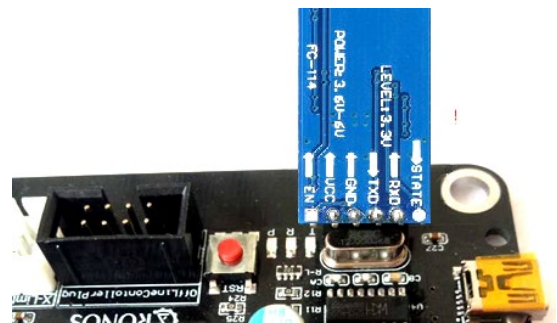


圖二十二：脫機板液晶之檔案頁畫面

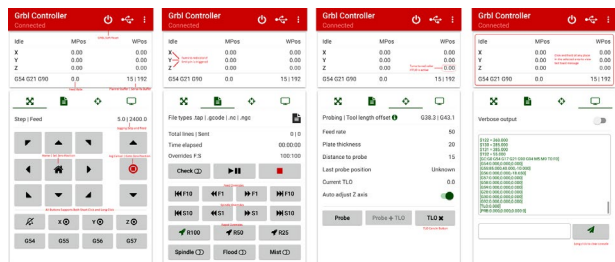


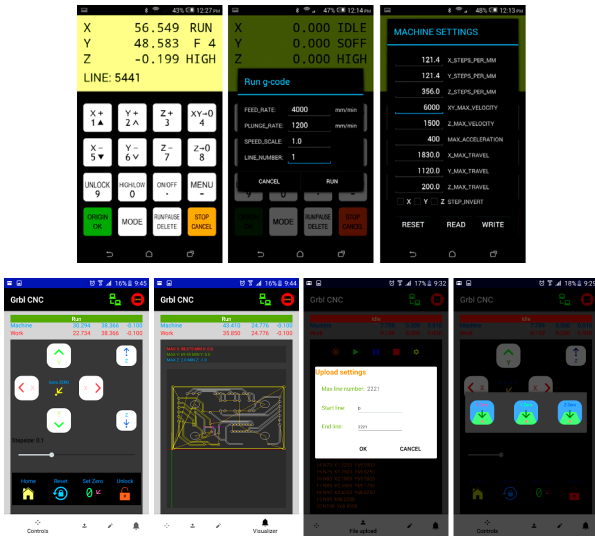
圖二十三：脫機板液晶之速率頁畫面

為了增加使用方便性，控制主板上層提供一藍牙插槽，如圖二十四所示，以利手機App進行手動操控或GM檔案傳輸，只需準備HC系列藍芽通訊模組(6根針腳之HC-05為主從一體、或4根針腳之HC-06為從端均可使用)，以HC-05為例，倒插入控制主板上層之藍芽插槽，同時保留二最外側插腳不用；在HC-05安裝前，需先用AT指令將速率設為115200、無校正、1位停止位元，再透過手機進行藍芽配對，通常預設配對密碼為0000或1234；安裝好藍芽通訊模組之後，便可在手機Play商店中搜尋App免費應用程式(諸如Grbl Control、DroidCNC、Grbl CNC附路徑預覽...等)，如圖二十五所示，並下載安裝至手機，執行時經過藍芽配對成功後，便可在手機上遠端手動控制雷雕機，或傳nc檔之GM碼給雷雕機加工。



圖二十四：安裝HC-05藍芽通訊模組至控制主板上



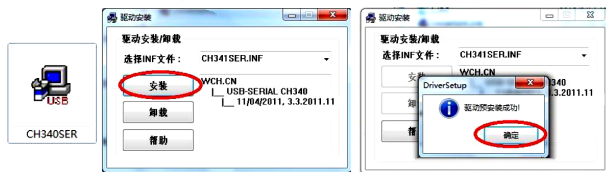


圖二十五：各類手機免費App程式介面外觀  
(上：Grbl Control、中：DroidCNC、下：Grbl CNC)

### 3.2 GRBL眾軟體介紹

了解控制主板、脫機(離線)控制板、三軸步進馬達驅動模組等韌體之後，接著可以開始詳細介紹GRBL軟體了；因為除了硬體機台組裝、韌體電路板配線之外，正確地設定GRBL軟體是雷雕機能正常運作最重要之環節。廠家建議使用桌機版Grbl Control[3]或微雕管家[4]來搭配Cronos控制主板，通常Grbl Control較適宜進行三維銑削及二維雷雕，微雕管家則限用於二維雷雕。

不管使用何種軟體，首先要針對XP 32/64或WIN7 32/64之視窗作業系統安裝CH340驅動程式，以產生控制主板與電腦間USB的COM連接埠，安裝點選步驟如圖二十六。接著以滑鼠右鍵點擊我的電腦(或本機)、內容、裝置管理員之後，看看是否出現CH340埠號，如果安裝成功就會出現COMx，記下後面數字以備待會在Grbl Control設定時使用。



圖二十六：安裝驅動程式以產生主板COM連接埠

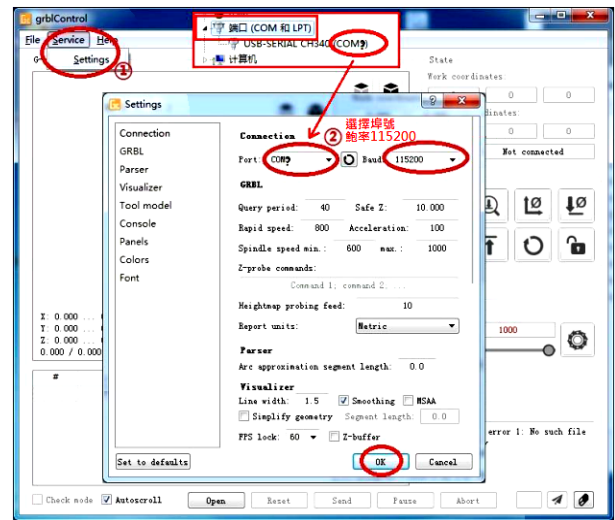
#### 3.2.1 Grbl Control軟體設定

Grbl Control軟體適用於控制機台進行3軸運動，例如三維銑削及二維雷雕。在安裝Grbl Control軟體並執行後，會先出現主視窗，如圖二十七所示，主視窗內中央有刀具路徑預覽區、下方有GM碼加工程式區、右側有操控面板區。

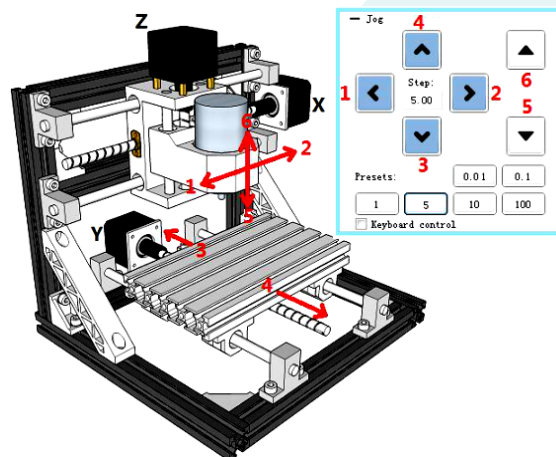


圖二十七：Grbl Control軟體主視窗外觀圖

如圖二十八所示，一開始先選主功能表Service之Setting分頁，在Connection項下先設置埠號Port及鮑率Baud，埠號須選驅動程式所產生之COMx，鮑率需設為115200，設置好後返回主介面，使軟體連接機台；若連線成功，操控面板區之指令輸入終端台會顯示[CTRL+X] < Grbl 0.9j ['\$' for help]-之提示，若否則表示連線失敗，需進一步關閉軟體、拔除USB線、再重新插入USB線並重啟軟體，再重複之前步驟。

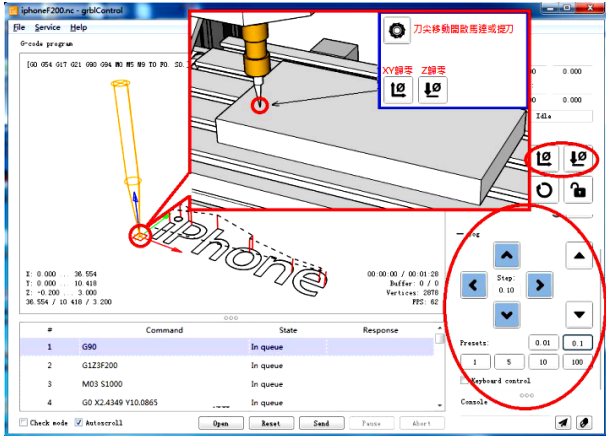


圖二十八：Connection項下設置埠號及鮑率



圖二十九：使用手動控制面板進行方向核對

接著進行方向核對(圖二十九)，點選手動控制以開啟手動控制面板，並按壓XYZ軸六個箭號圖示之移動按鈕，若圖示箭號方向與機台實際移動方向相反，表示步進馬達接線順序顛倒，需將排線翻面再插入。



圖三十：校刀執行工作座標歸零

方向核對正確後，GRBL Control設定就基本完成，接著加工前尚需進行校刀，如圖三十所示，校刀主要係設定加工路徑檔之工作原點在機器座標中的實際位置。例如在木料正中心刻字，工作原點若設在文字左15mm下5mm處，則木料可隨意擺放，然後將刀頭移到木料中心之左15mm(X)、下5mm(Y)處，並執行將工作座標XYZ歸零，便完成對刀工作；找Z零點高度需用銑刀尖輕抵工件Z方表面，傳統尚可隔張紙慢慢下降，或使用與主板連接之校刀器尋找適當高度；需注意對刀時要先提刀，因刀尖在移動過程中可能有撞刀風險。

茲以Grbl Controller 3.4.6版本為例，整理條列機台加工前設定之簡化步驟如下：

- (1) 重設連線，自動識別調整連接埠(port name)和鮑率(baud rate)。
- (2) 會顯示現在默認的設定和其他信息。
- (3) 點擊spindle on勾選框以點亮雷射；點擊方向移動，以檢查馬達線路是否正常。
- (4) 走至初始加工點，點擊zero position按鈕，把各軸現在位置定為工作原點。
- (5) 進入options進行參數設定(例：invert axis勾選框、use millimeter勾選框)。
- (6) 點GRBL settings按鈕，設3軸步進馬達1mm之脈衝數(此數字至關重要)。
- (7) 選擇一數控指令檔來測試。

### 3.2.2 微雕管家軟體設定

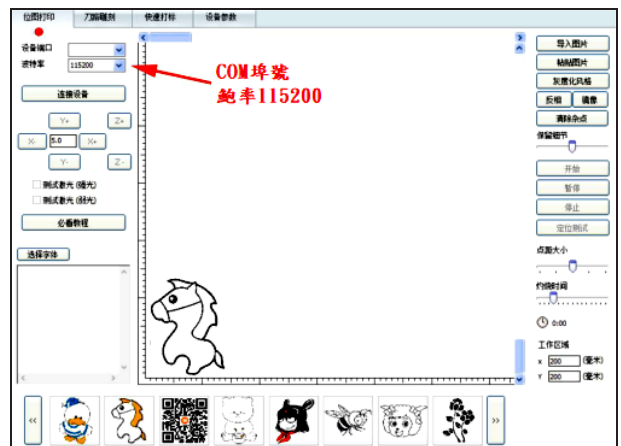
微雕管家能簡單快速進行工件表面之二維圖案雷雕加工，如圖三十一所示，首先進入「位圖打印」分頁，選好連線COM埠號，並將鮑率設為115200。接著在「設備參數」分頁進行雷射焦距高度調校，如

圖三十二所示，目的是要將工件表面的雷射光點集束到最小面積，以求最佳加工效能；為保護眼睛、預防傷眼，故先設定弱光等級18(一般18~22)，功率100%，其次才是視實際需要暫設加速度為200mm/sec<sup>2</sup>、速度500mm/min，視作動需要勾選三軸反向，另外要計算馬達步進數，其公式如式(1)所示，上述設置完成便可按壓儲存設置按鈕。

馬達走1mm之脈衝數

$$= ((360/\text{步距角}) \times \text{驅動板細分數}) / \text{牙桿導程}$$

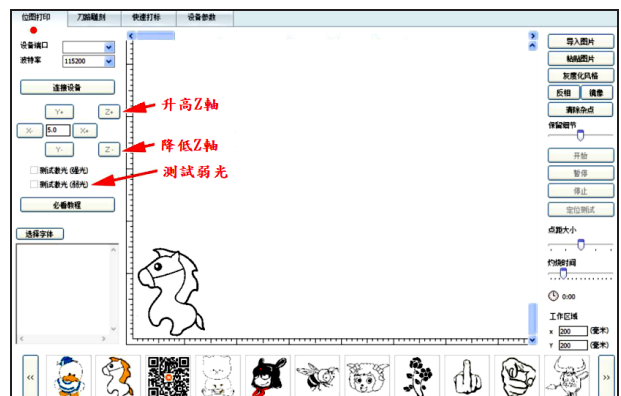
$$= (\text{原始脈衝數} \times \text{驅動板細分數}) / \text{牙桿導程} \quad (1)$$



圖三十一：微雕管家外觀圖、設置埠號與鮑率



圖三十二：設定雷射弱光等級



圖三十三：弱光設定後，再進行Z軸升降



再返回到「位圖打印」分頁，如圖三十三，先勾選測試雷射弱光，再進行適當Z軸高度調整，按Z+上升、Z-下降，觀察雷射光點大小，直至最小圓點面積；熟練者可戴護目鏡直接勾選強光，並水平移動被加工物，同時調整Z軸高度，使其表面燒灼痕跡線條達到最細。

### 3.2.3 Grbl軟體進階設定

所謂工欲善其事、必先利其器，舊版Grbl Control、微雕管家或其他廠家之雷雕軟體，均附有GBRL進階設定之功能，例如在微雕管家軟體的「設備參數(或設備設定)」分頁下，有一「設備設定(專家)>>>」之按鈕，點擊後便會出現進階GRBL參數列表，有些軟體甚至可下Grbl指令來更改參數，例如Grbl Control指令輸入終端台可下Grbl指令…等，如表二所示之指令格式：

表二 Grbl指令格式表

指令格式	功能及作用
\$\$	顯示所有 Grbl 參數
\$#	顯示某號參數
\$G	顯示此版本   所有 G 指令
\$I	顯示版本資訊
\$N	顯示 startup 啟動多行 G 指令區塊
\$x=值	儲存參數設定並保存到 Rom
\$Nx=line	儲存 startup 啟動執行區塊
\$C	檢查 GM 碼指令的運行過程
\$X	解除 alarm 警告鎖定
\$H	執行原點復歸(home)，使三軸歸位(必須限位功能致能)
~	[即時指令]加工執行開始
!	[即時指令]進給(feed)保持( hold)
?	[即時指令]顯示目前狀態資訊
ctrl-x	[即時指令]重置 Grbl(參數或位置會丟失)
\$RST=\$	恢復原廠預設值

還記得若想讓電腦串列終端(serial terminal)連上Grbl控制主板時，需將鮑率設為115200以及8-N-1(即8位元、無奇偶校驗、1停止位元)，連線成功後，指令輸入終端台會得到提示訊息(例：Grbl 0.9i ['\$' for help])，便可進行Grbl參數設定，其所有機台參數清單及其預設值(各家不一)如表三、四之所示[5]：

表三：Grbl參數預設值清單列表

參數號碼與設定值	英文意義	單位
\$0=10	step pulse	usec
\$1=25	step idle delay	msec
\$2=0	step port invert mask	00000000
\$3=6	dir port invert mask	00000110
\$4=0	step enable invert	bool
\$5=0	limit pins invert	bool
\$6=0	probe pin invert	bool
\$10=3	status report mask	00000011
\$11=0.020	junction deviation	mm
\$12=0.002	arc tolerance	mm
\$13=0	report inches	bool
\$20=0	soft limits	bool
\$21=0	hard limits	bool

\$22=0	homing cycle	bool
\$23=1	homing dir invert mask	00000001
\$24=50.000	homing feed	mm/min
\$25=635.000	homing seek	mm/min
\$26=250	homing debounce	msec
\$27=1.000	homing pull-off,	mm
\$100=314.961	x	step/mm
\$101=314.961	y	step/mm
\$102=314.961	z	step/mm
\$110=635.000	x max rate	mm/min
\$111=635.000	y max rate	mm/min
\$112=635.000	z max rate	mm/min
\$120=50.000	x accel	mm/sec <sup>2</sup>
\$121=50.000	y accel	mm/sec <sup>2</sup>
\$122=50.000	z accel	mm/sec <sup>2</sup>
\$130=225.000	x max travel	mm
\$131=125.000	y max travel	mm
\$132=170.000	z max travel	mm

每個參數號碼對於機台硬體而言，都有其細部控制之特殊意義，設定上之中文意義及單位請詳見表四，須注意要明白其單位與機台硬體、或每個遮單位元欄位與該軸之對應關係，才不會給錯數值。

表四：Grbl參數值之單位列表

參數號碼/單位	中文意義	硬體控制說明
\$0= us 值	步進脈衝時間	最小步進脈衝長度，太長則不適於高進給，默認值為 10 微秒
\$1= ms 值	步進延遲時間	轉停後，Grbl 延遲停電之毫秒時間
\$2= mask	步進埠反壓遮罩	正常時，電壓為 LH(\$0)L 反壓時，電壓為 HL(\$0)H
\$3= mask	方向埠反轉遮罩	正常時，軸正向為 L、軸反向為 H；反轉時，軸正向為 H、軸反向為 L
\$4= bool	步進致能反相	預設時，H 禁用、L 致能 相反時，L 禁用、H 致能
\$5= bool	極限引腳反相 (limit pin)	預設時，引腳因 Arduino 提升電阻為 H；引腳為 L，Grbl 視為觸發
\$6= bool	探頭引腳反相 (probe pin)	預設時，引腳因 Arduino 提升電阻為 H；引腳為 L，Grbl 視為觸發
\$10= mask	?狀態回報遮罩 (低限度以節約資源)	機器位置 1、工作位置 2、規劃緩衝區 4、RX 緩衝區 8、極限引腳 16
\$11= mm 值	線段接點偏差 (G-code 角落路徑)	大則導致走速快、失步多、定位差；小則導致走速慢、失步少、定位準
\$12= mm 值	圓弧容錯偏差 (G2 / G3)	若圓弧過於粗糙或走速緩慢，請調低，但可能過多強使 Grbl 過載
\$13= bool	毫米改英寸回報	實時坐標及參數回報單位系統切換
\$20= bool	軟極限設定	檢查 G 代碼是否超出各軸最大行程限制，是則立即發出進給保持、關閉主軸&冷卻液、設警報指示
\$21= bool	硬極限設定	極限引腳透過內部提升電阻保持 H，在引腳和地線間接並聯二個常開開關
\$22= bool	原點復歸循環	通常極限開關設在 +x、+y、+z 的最遠點，用機器零參考點來復歸時，首先 Grbl 停止所有 G 代碼命令，然後 Z 軸正向移動、再 +x、+y 方向同時移動
\$23= mask	復歸方向反向遮罩	預設時，認定極限開關處於軸正方向；相反時，認定極限開關處於軸負方向
\$24= mm/min 值	復歸進給速率 (出發搜索)	首先以較高速率搜索極限開關，然後以較慢(本)速率復歸到機器零點
\$25= mm/min 值	復歸進給速率 (回去原點)	首先以較高(本)速率搜索極限開關，然後以較慢速率復歸到機器零點





\$26=ms 值	極限開關去彈跳延遲	開關觸發時需要電氣彈跳時間信號，一般為 5-25 毫秒
\$27=mm 值	復歸移出距離	原點復歸循環後，離開極限開關避免碰觸
\$100=步/mm 值	三軸螺桿步距比值	軸解析度 = [每轉步數(通常 200)*微步(通常 1,2,4,8 或 16)] / 導螺桿螺距
\$101=步/mm 值		
\$102=步/mm 值		
\$110=mm/min 值	三軸最大速度	移動時檢查是否超出任何一軸之最大速率，是則減慢以確保未超出
\$111=mm/min 值		
\$112=mm/min 值		
\$120=mm/sec <sup>2</sup> 值	三軸加速度	低值運動緩慢，高值移動快速
\$121=mm/sec <sup>2</sup> 值		
\$122=mm/sec <sup>2</sup> 值		
\$130=mm 值	三軸最大行程	檢查運動指令是否通出機器限制
\$131=mm 值		
\$132=mm 值		
\$#	查看 g-code 參數	包括：G54-G59 設定工作座標、G28 / G30 預定復歸位置，G92 坐標偏移，刀具長度偏移和探測的坐標偏移值

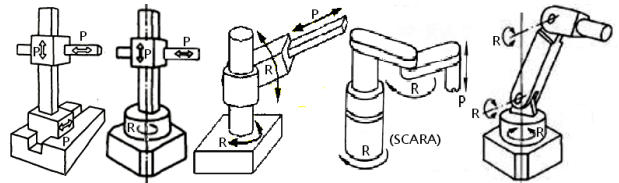
在下達指令時，需遵照Grbl指令格式去撰寫，以得到OK正確之回應，茲列舉一些常用指令之範例，如下面表五所示，以利做為下達指令之參考。

表五：Grbl指令注意事項之範例

指令	注意事項
\$x=值	微秒步進脈衝更改為 10us，例：\$0=10
\$1=ms 值	設最大 255 來保持軸啟用(通電以握持位置)，例：\$1=255
\$2=mask	反轉 X、Z 軸，例：\$2=5，(即 0000X0Z)
\$3=mask	只反轉 Y 方向，例：\$3=2，(即 00000Y0)
\$4=bool	相反，例：\$4=1
\$5=bool	相反，例：\$5=1
\$6=bool	相反，例：\$5=1
\$10=mask	需要機器和工作位置，例：\$10=3 僅要機器位置及極限引腳，例：\$10=17
\$11=mm 值	查閱：Grbl 轉彎演算法 (Grbl cornering algorithm) (不需調此設定，因 0.002mm 遠低於多數 CNC 精度)
\$12=mm 值	\$12=0 (mm)、\$13=1 (inches)
\$13=bool	\$20=1 (啟用)、\$20=0 (禁用)
\$20=bool	注意：因 Grbl 需知位置，需準確之軸最大行程及回原點致能
\$21=bool	\$21=1 (啟用)、\$21=0 (禁用) 注意：硬事件為關鍵事件，會立即停止步進，並丟失座標，將進入無窮循環警報模式
\$22=bool	注意：查看 config.h 更多復歸細部選項，例：啟動時禁用原點復歸、何軸先移動、何種順序移動
\$23=mask	只反轉 X、Y 軸，例：\$23=6，(即 00000XY0)
\$27=mm 值	pull-off travel
\$100=步/mm 值	steps_per_mm =
\$101=步/mm 值	(steps_per_revolution*microsteps)/mm_per_rev
\$102=步/mm 值	
\$110=mm/min 值	設 X 軸，則發送 G0 X50 足夠遠距離，以達最大速度
\$111=mm/min 值	直至馬達失步(可能伴隨噪音)，輸入此值 10-20%
\$112=mm/min 值	
\$120=mm/sec <sup>2</sup> 值	最簡單方法是逐步增加，單獨測試各軸，直到馬達停轉，然後使用此最大值 10-20% 設定加速度
\$121=mm/sec <sup>2</sup> 值	
\$122=mm/sec <sup>2</sup> 值	
\$130=mm 值	僅於啟用軟限制(和回原點)時才有效，屬 Grbl 的軟限制功能
\$131=mm 值	
\$132=mm 值	

#### 4. 機械手臂概念設計

通常機械手臂常見之關節(Joint)，分為旋轉對(Revolute Pair)或滑動對(Prismatic Pair)二種，各以英文字母R及P來代表；若進一步以關節組合來分類，則機械手臂不考慮夾爪情況下，可區分為直角坐標PPP、圓柱座標RPP、球座標(SCARA)RRP、關節(或迴轉)座標RRR等五種，如圖三十四所示[6]：



圖三十四：各類型關節組合之機械手臂

假設每個關節在三維空間中有6個自由度(3軸平移、3軸旋轉)，則旋轉對R及滑動對P均喪失5個自由度、僅剩1個自由度，若帶入三維自由度公式，如式(2)所示，其中N為所有組件數量、J為關節數量、F為某關節剩餘自由度，減6為扣除機台地板之自由度，依此公式，前述這五種機械手臂均為N=4、J=3、F=3，故PPP、RPP、RRP及RRR之自由度均為6(4-3)+3-6=3，剛好以三顆馬達或線性缸來驅動時，可得唯一之位置座標。本研究第一代機械手臂採用RRRR結構，其自由度經計算後得6(5-4)+4-6=4，第二代機械手臂採用RRR改良結構，其自由度為3。

$$DOF = 6 \times (N - J) + \sum_{i=1}^J F_i - 6 \quad (2)$$

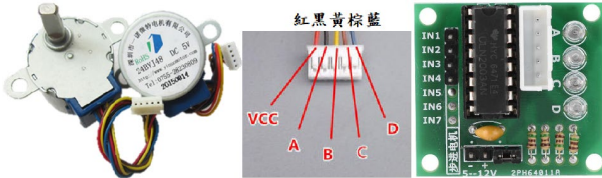
因微型電動線性缸較為昂貴且不普及，故本研究採用步進馬達來進行第一代RRRR及第二代RRR結構之機械手臂製作，茲將第一、二代機械手臂實作要點條列整理如下：

- (1) 原材使用3mm合成木漿板(又稱椴木板)，進行雷射切割；除裝配上採緊配公差外，亦用白膠輔助黏合，以求最大強度。
- (2) 第一代機械手臂採用RRRR結構，四組步進馬達使用最普及之四相五線帶減速機之5VDC步進馬達(型號24BYJ48)，轉子轉一圈64步、減速機也是64:1、牽入扭矩29.4mN·m，搭配ULN2003步進馬達驅動板(該板可接受5~12VDC供電)；另Arduino控制訊號接到驅動板IN1~4腳隔著IC對側分別為馬達之藍、棕、黃、黑及共陽紅線，如圖三十五所示。
- (3) 第一代機械手臂採RRRR結構，外觀如圖三十六所示，組件由下而上分別為底板基座、立柱、上臂、下臂與手腕，除底板基座為水平正方形板，其餘組件為全包覆式六面長方體，各面以城垛形相拼接；除立柱之轉軸方向為垂直外，其餘模組化組件之轉軸方向為水平；底板基座含一水平齒輪由馬達帶動，與立柱底部相同大小之齒輪相契合，以達成垂直轉軸之效果。
- (4) 第一代設計過於笨重，導致肩膀關節處馬達難以揚升，故第二代主要針對減輕整體重量、加大馬達扭矩、增加電控能力等三大方向進行改善。
- (5) 第二代機械手臂採RRR結構，外觀如圖三十七所示，少了末段手腕，但加了伺服機夾爪，並採用鏤空雙隔板式，同時特意加大了步進馬達(型號28BYJ48，步進角與減速比和24BYJ48相同，但牽入扭矩可增至37.2 mN·m)，以改善第一代組件過重及馬達扭矩不足之問題；另Arduino接到驅動板

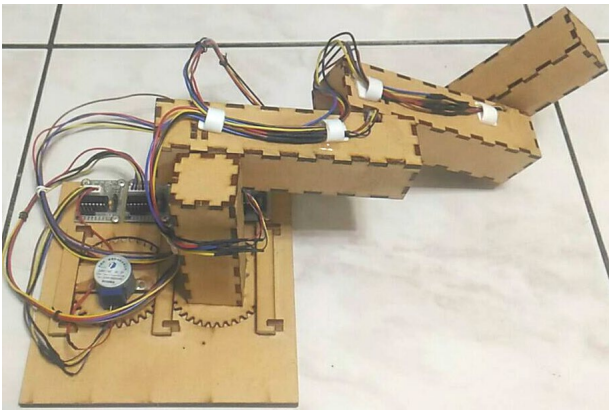


IN1~4腳隔著IC對側分別為馬達之藍、粉紅、黃、橙及共陽紅線，如圖三十八所示。

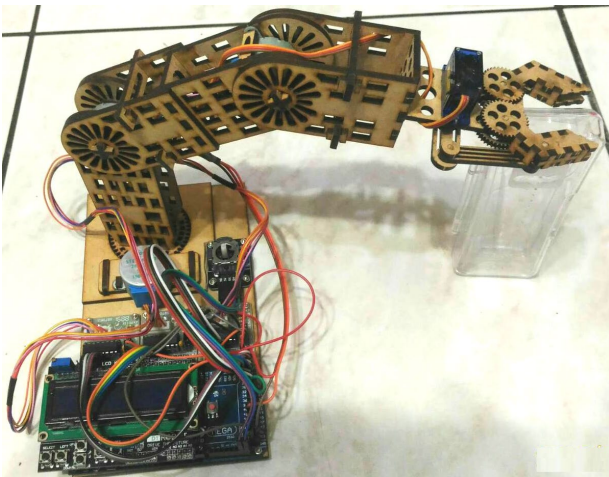
- (6) 第二代機械手臂之電控部分，增加了液晶文字顯示幕，以利看到受控軸狀態及座標之改變；同時也新增了手控搖桿，二者可提高手控之提示性與操控性，大大增加了遊戲樂趣。



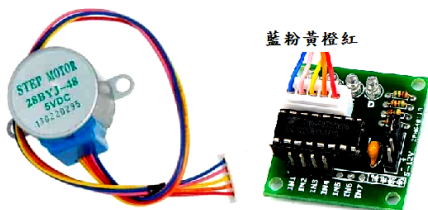
圖三十五：24BYJ48步進小馬達與ULN2003驅動板



圖三十六：第一代全包覆式RRRR結構之機械手臂



圖三十七：第二代改良輕量化鏤空雙隔板式RRR結構之機械手臂



圖三十八：28BYJ48步進大馬達與ULN2003驅動板

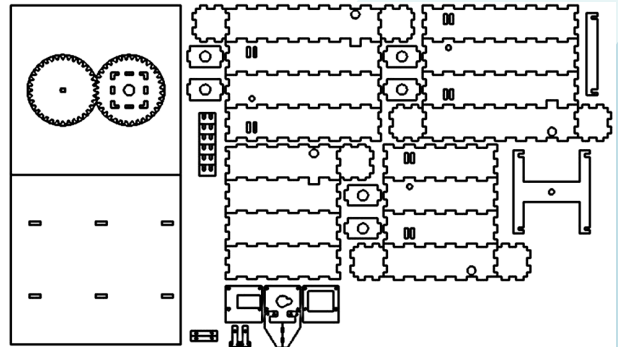
## 5. 組件雷雕路徑繪製

雷雕路徑之產生，必須在AutoCAD電腦輔助繪圖軟體上，進行二維切割刀具路徑之設計，再用轉檔功能將DXF檔轉成NC檔。在繪製過程中，有些注意事項必須知道，以求切割路徑設計妥適，如下條列：

- (1) 因雷射切割木料板材，會造成木料上表面較下底面焦黑，因切割線會有細微寬度損耗，且上表面之損耗會大於下底面，雷射單點切割會產生倒立圓錐狀之燒焦體積，故零件不可太細小、切割路徑之線條外型不可過於繁瑣密集。
- (2) 路徑繪製以mm為單位，先進行理論尺寸繪製，因木料切割損耗，成品組裝時會搖動鬆脫，故於整張圖面完成繪製後，還要進行緊配尺寸微調(使用偏移複製offset功能)，一般會在二塊板材凹凸裝配處，單側尺寸會蓄意增加0.05~0.1mm，以使裝配處有緊配公差之效果。
- (3) 為預防切割作業完成後，各組件之零件散落在雷雕機之工作台面上，除非加裝蜂巢式台面，否則細小零件會因為中途傾斜而影響到後續切割準度，甚至會發生掉落台面鋁擠型凹槽而漏割小零件內的加工路徑。通常會增加連接髮絲以解決之，作法是適當增加數個未完全切斷之連接處，加工完成後便可輕鬆單手提取整片板材，否則就要一塊塊撿拾小零件，且繪製時盡量將各零件相疊合，以共享切割邊緣，並減少加工時間。
- (4) 承上，實務上可在AutoCAD中繪製一個極小圓，其直徑可為0.2mm，再大量使用多重複製(multiple copy)及鎖點到最近點(snap on line)功能，在某零件周長上平均分散這些小圓，利用線段裁剪(trim)功能刪去圓內切割路徑線段，使其線段具有極小未切割連接處，最後再利用圖層管理把所有小圓一口氣刪掉。
- (5) 承上，須注意盡量不要鎖點到線段中點，以免造成後續尺寸量測時，鎖不到中點之困難。

### 5.1 第一代機械手臂雷雕路徑繪製

雷雕路徑設計針對第一代機械手臂，如圖三十九所示，簡述其缺點條列如後：



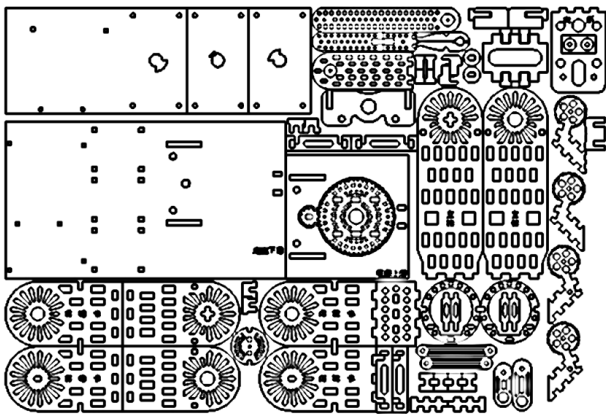
圖三十九：第一代機械手臂雷雕路徑繪製

- (1) 第一代底板基座之二個大正齒輪，其半徑相等，

- 加上立柱總成重壓，因齒輪與底板互相摩擦，會產生馬達機械效益(=1)偶發不足以克服靜摩擦力。
- (2) 第一代每節組件採長方體全包覆式設計，造成組件重量過重，導致中段二顆馬達失步或顫動；另六片零件互相以城垛外型緊配接合成為六面體，導致拆或裝都不方便。
  - (3) 第一代馬達置於前段長方體末端，再用轉動軸穿過後段長方體前端，藉以帶動後段長方體；因轉動軸穿過二層板厚，加上所剩軸長僅數公厘，導致連接容易鬆脫，最後係利用車製之金屬軸件套合馬達轉動軸，藉以延伸軸長，並搭配塑鋼土補強前後段之接合面，才不致產生鬆脫現象。
  - (4) 第一代馬達安裝於長方體一端，無特殊機構防止馬達本體原地自旋或鬆搖，須將馬達二側金屬安裝翼緣用黏合方式固定，使其不會鬆脫自旋。
  - (5) 第一代轉動軸穿過後段長方體前端之異形孔，強度不足以支持轉動後段總成之所有重量，故額外增設二塊隔間板於後段長方體之內部空間，藉以迫緊轉動軸之延伸軸長。
  - (6) 因小馬達自帶有減速齒輪，加上不似傳統圓形剖面，造成轉動軸軸心位置沒有對正長方體矩形中央，導致後續連接段於轉動時不美觀及偏旋。

### 5.2 第二代機械手臂雷雕路徑繪製

改良後第二代機械手臂之雷雕路徑設計，如圖四十所示，簡述其相對應之優點條列如後：



圖四十：第二代機械手臂雷雕路徑繪製

- (1) 第二代底板基座乘載大小正齒輪各一，馬達安裝於小齒輪上，當帶動大齒輪轉動時，具有較優之機械效益(>1)，容易克服立柱總成重壓之靜摩擦力。
- (2) 第二代每節組件採鏤空雙隔板式設計，加上少了末段手腕，大幅地減輕組件重量，馬達不易失步或顫動；另雙隔板互相以條狀扣件及插銷組合，拆裝方便。
- (3) 第二代馬達置於前段雙隔板末端，以馬達本體當作轉動軸，短軸馬達不需額外自製延伸金屬軸件，亦無需塑鋼土強化。

- (4) 第二代馬達安裝於雙隔板之一端，條狀扣件已穿過馬達二金屬安裝翼緣，馬達自然不會原地自旋或鬆搖。
- (5) 第二代以馬達本體當作轉動軸，加上前後段以二層板厚之隔板間距之差，來緊密包容套疊，且馬達短軸末端有增設一異形孔補強墊圈，強度足以支持轉動後段總成之所有重量。
- (6) 因大馬達自帶有減速齒輪，偏心之轉動軸軸心之位置，已經對正雙隔板末端之圓心，後續連接段轉動時美觀且不偏旋。

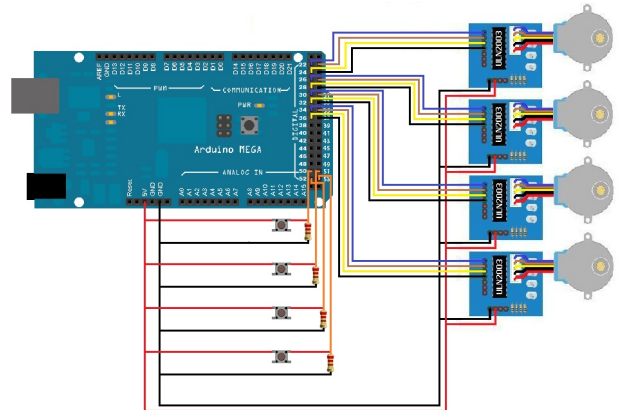
## 6. Arduino 電路配線

第一代機械手臂採用RRRR結構，電控零件包括四顆24BYJ48步進馬達、四片ULN2003步進馬達驅動板、一顆夾爪伺服機(因改良至二代故棄之)、四顆微動開關(隨附四顆220Ω限流電阻)以及一片Arduino MEGA2560控制主板。

至於第二代機械手臂採用RRR結構，電控零件包括三顆28BYJ48步進馬達、三片ULN2003步進馬達驅動板、一顆夾爪伺服機以及一片Arduino MEGA2560控制主板。另外為了增加操控性與可讀性，還增添一塊16×2雙排文字型液晶顯示螢幕(帶有LCD Keypad Shield，上有六個按鍵)以及雙軸按鍵搖桿模組作為手控使用。

### 6.1 第一代機械手臂Arduino電路配線

第一代機械手臂因有四組24BYJ48步進馬達及ULN2003步進馬達驅動板，原本使用Arduino UNO控制主板，但馬達4×4=16個以及4個開關之數位輸出腳總需求遠遠超過Arduino UNO之數位腳數，故改用Arduino MEGA 2560，如圖四十一所示：

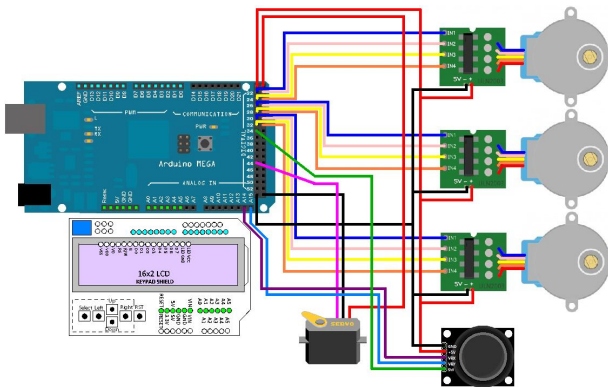


圖四十一：第一代機械手臂之電路配線圖

### 6.2 第二代機械手臂Arduino電路配線

第二代機械手臂除了三顆步進馬達之接線腳位與第一代相同，其餘還新增了夾爪伺服機、液晶顯示螢幕以及雙軸按鍵搖桿，均與第一代大幅相異，如圖四十二所示：





圖四十二：第二代機械手臂之電路配線圖

## 7. Arduino程式設計

本節所指C語言程式設計，係針對第一代及第二代之機械手臂而言，非指雷雕機上之Arduino控制主板，合先敘明。

### 7.1 第一代機械手臂Arduino程式設計

因有四顆步進馬達，每顆馬達又有四隻相位腳，故驅動前先利用pin[4][4]二維陣列指定腳位，同時先制定resolution解析度為360度除以64分割、再除以減速機降速比64；驅動時可呼叫oneStep(motor,dir)函數來驅動馬達走一步，走前先查看phase[motor]各馬達之相位(1~4)，以便能持續運轉，走完須更新該馬達相位；至於要走幾步才能達到指定浮點角度，則須靠rotate(motor,dir,deg)函數來計算並重複呼叫oneStep函數；值得一提的是，rotate函數能下達最靠近目標浮點角度之整數步數，再用浮點數remain[4]來記錄上次四顆馬達的浮點剩餘誤差，以利在下次驅動時，再把浮點剩餘誤差加回到下次轉的角度，且能預防密集驅動造成之誤差累積；由於第一代機械手臂靠四個按鈕來控制四顆馬達正反轉，故每押一次按鈕，則馬達反方向走，放開按鈕則馬達停止。整個手動控制第一代機械手臂之Arduino程式隨附如下：

```
const int pin[4][4] = {{22, 23, 24, 25},
                      {26, 27, 28, 29},
                      {30, 31, 32, 33},
                      {34, 35, 36, 37}};

const int sw[4] = {50, 51, 52, 53};
#define CW +1
#define CCW -1

float resolution = (360.0/64) / 64;
int motor; //作用焦點馬達(0~3)
int dir[4]; //四顆馬達目前轉向(CCW, CW)
float remain[4]; //四顆馬達剩餘步數(不足1步)
int phase[4]; //四顆馬達激磁相位(整數 1~4)
int press[4]; //四顆開關按壓電位(按 LOW, 放 HIGH)

void oneStep(int motor, int dir)
{
    int p = phase[motor],
        p1 = pin[motor][0],
        p2 = pin[motor][1],
        p3 = pin[motor][2],
        p4 = pin[motor][3];

    if(p==0) {digitalWrite(p1,1); digitalWrite(p2,1); digitalWrite(p3,0); digitalWrite(p4,0); delay(5);}
    else if(p==1) {digitalWrite(p1,0); digitalWrite(p2,1); digitalWrite(p3,1); digitalWrite(p4,0); delay(5);}
    else if(p==2) {digitalWrite(p1,0); digitalWrite(p2,0); digitalWrite(p3,0); digitalWrite(p4,1); delay(5);}
    else if(p==3) {digitalWrite(p1,1); digitalWrite(p2,0); digitalWrite(p3,0); digitalWrite(p4,1); delay(5);}

    if(dir == CW) { phase[motor]=p+1; if(phase[motor]== 4) phase[motor]=0; }
    else if(dir == CCW) { phase[motor]=p-1; if(phase[motor]== -1) phase[motor]=3; }
}

void rotate(int motor, int dir, float deg)
{
    float steps = deg / resolution;
    float amend = dir*steps + remain[motor];
    int truth = (amend-floor(amend) < ceil(amend)-amend)? floor(amend) : ceil(amend);

    for(int i=1; i<=abs(truth); i++) oneStep(motor, dir);
    remain[motor] = amend - truth;
}

void setup()
{
    for(int i=0; i<4; i++)
        for(int j=0; j<4; j++) pinMode(pin[i][j], OUTPUT);
    for(int i=0; i<4; i++) pinMode(sw[i], INPUT);
    for(int i=0; i<4; i++) digitalWrite(sw[i], HIGH);

    for(int i=0; i<4; i++)
    {
        dir[i] = CCW;
        remain[i] = 0.0;
        phase[i] = 0;
    }
}

void loop()
{
    for(int i=0; i<4; i++) press[i] = digitalRead( sw[i] );
    for(int i=0; i<4; i++)
    {
        if(press[i] == LOW)
        {
            motor = i;
            rotate(motor, dir[i], 1.0);

            if(digitalRead(sw[i]) == HIGH)
            {
                if(dir[i] == CCW) dir[i] = CW;
                else if(dir[i] == CW) dir[i] = CCW;
            }
        }
    }
}
```

```
else if(p==1) {digitalWrite(p1,0); digitalWrite(p2,1); digitalWrite(p3,1); digitalWrite(p4,0); delay(5);}
else if(p==2) {digitalWrite(p1,0); digitalWrite(p2,0); digitalWrite(p3,0); digitalWrite(p4,1); delay(5);}
else if(p==3) {digitalWrite(p1,1); digitalWrite(p2,0); digitalWrite(p3,0); digitalWrite(p4,1); delay(5);}

    if(dir == CW) { phase[motor]=p+1; if(phase[motor]== 4) phase[motor]=0; }
    else if(dir == CCW) { phase[motor]=p-1; if(phase[motor]== -1) phase[motor]=3; }
}

void rotate(int motor, int dir, float deg)
{
    float steps = deg / resolution;
    float amend = dir*steps + remain[motor];
    int truth = (amend-floor(amend) < ceil(amend)-amend)? floor(amend) : ceil(amend);

    for(int i=1; i<=abs(truth); i++) oneStep(motor, dir);
    remain[motor] = amend - truth;
}

void setup()
{
    for(int i=0; i<4; i++)
        for(int j=0; j<4; j++) pinMode(pin[i][j], OUTPUT);
    for(int i=0; i<4; i++) pinMode(sw[i], INPUT);
    for(int i=0; i<4; i++) digitalWrite(sw[i], HIGH);

    for(int i=0; i<4; i++)
    {
        dir[i] = CCW;
        remain[i] = 0.0;
        phase[i] = 0;
    }
}

void loop()
{
    for(int i=0; i<4; i++) press[i] = digitalRead( sw[i] );
    for(int i=0; i<4; i++)
    {
        if(press[i] == LOW)
        {
            motor = i;
            rotate(motor, dir[i], 1.0);

            if(digitalRead(sw[i]) == HIGH)
            {
                if(dir[i] == CCW) dir[i] = CW;
                else if(dir[i] == CW) dir[i] = CCW;
            }
        }
    }
}
```

### 7.2 第二代機械手臂Arduino程式設計

第二代之程式與第一代類似，都是利用函數oneStep(dir)及rotate(dir,deg)來驅動步進馬達，唯一差異便是將2相激磁的四步，改為1-2相激磁的八步；此外，第二代多了液晶螢幕、夾爪伺服機以及雙軸搖桿之相關函數，例如：readLCDkeys為讀取液晶螢幕六顆按鍵的函數，在loop執行迴圈中，若押RIGHT鍵表示轉動焦點在手肘ELBO、押UP鍵表示轉動方向改為CCW、押DOWN鍵表示轉動方向改為CW、押LEFT鍵表示轉動焦點在上臂UARM、押SELECT鍵表示轉動焦點在基座BASE；按壓到搖桿Fire按鈕表示夾爪gripper由開變關或由關變開，其搖動雙軸搖桿，傳回0~1023之類比電壓值，故本程式以300、700作為有否上下左右操控搖桿之分界點。整個手動控制第二代機械手臂之Arduino程式隨附如下：

```
#include <LiquidCrystal.h>
#include <Servo.h>

#define CW -1 //馬達順轉
#define CCW +1 //馬達逆轉
```

```

#define OPEN +1 //夾爪開
#define CLOSE -1 //夾爪閉

#define BASE 0 //基座馬達
#define UARM 1 //上臂馬達
#define ELBO 2 //手肘馬達
#define GRIP 3 //夾爪馬達

#define btnRIGHT 0 //ELBO
#define btnUP 1 //CCW
#define btnDOWN 2 //CW
#define btnLEFT 3 //UARM
#define btnSELECT 4 //BASE
#define btnNONE 5

LiquidCrystal textLCD(8, 9, 4, 5, 6, 7); //液晶螢幕
Servo gripper; //機械夾爪
const int pinStep[3][4] = {{22, 23, 24, 25}, //基座步進
                          {26, 27, 28, 29}, //上臂步進
                          {30, 31, 32, 33}}; //手肘步進
const int pinFire=34, pinVrx=A14, pinVry=A15; //雙軸搖桿
const int pinKey=A0; //液晶按鍵
const int pinSvo=44; //夾爪伺服

float resolution = (360.0/64) / 64;
int focus; //作用焦點馬達(0~2,3)
int dir[3]; //3 顆馬達目前轉向(CCW,CW)
float remain[3]; //3 顆馬達剩餘步數(不足 1 步)
int phase[3]; //3 顆馬達激磁相位(整數 1~4)
int keyin; //液晶螢幕擴展板控制按鍵
int grip; //夾爪目前 OPEN 或 CLOSE
int volx, voly; //搖桿類比電壓讀入值

void oneStep(int dir)
{
  int p = phase[focus],
      p1 = pinStep[focus][0],
      p2 = pinStep[focus][1],
      p3 = pinStep[focus][2],
      p4 = pinStep[focus][3];

  if(p==0){ digitalWrite(p1,1); digitalWrite(p2,0); digitalWrite(p3,0); digitalWrite(p4,0); delay(2); }
  else if(p==1){ digitalWrite(p1,1); digitalWrite(p2,1); digitalWrite(p3,0); digitalWrite(p4,0); delay(2); }
  else if(p==2){ digitalWrite(p1,0); digitalWrite(p2,1); digitalWrite(p3,0); digitalWrite(p4,0); delay(2); }
  else if(p==3){ digitalWrite(p1,0); digitalWrite(p2,1); digitalWrite(p3,1); digitalWrite(p4,0); delay(2); }
  else if(p==4){ digitalWrite(p1,0); digitalWrite(p2,0); digitalWrite(p3,1); digitalWrite(p4,0); delay(2); }
  else if(p==5){ digitalWrite(p1,0); digitalWrite(p2,0); digitalWrite(p3,1); digitalWrite(p4,1); delay(2); }
  else if(p==6){ digitalWrite(p1,0); digitalWrite(p2,0); digitalWrite(p3,0); digitalWrite(p4,1); delay(2); }
  else if(p==7){ digitalWrite(p1,1); digitalWrite(p2,0); digitalWrite(p3,0); digitalWrite(p4,1); delay(2); }
  else { digitalWrite(p1,0); digitalWrite(p2,0); digitalWrite(p3,0); digitalWrite(p4,0); delay(2); }
  if(dir==CW){ phase[focus]=p-1; if(phase[focus]==-1) phase[focus]=7; }
  else if(dir==CCW){ phase[focus]=p+1; if(phase[focus]==8) phase[focus]=0; }
}

void rotate(int dir, float deg)
{
  float steps = deg / resolution;
  float amend = dir*steps + remain[focus];
  int truth = (amend-floor(amend) < ceil(amend)-amend)? floor(amend) : ceil(amend);

  for(int i=1 ; i<=abs(truth) ; i++) oneStep(dir);
  remain[focus] = amend - truth;
}

int readLCDkeys()
{
  int volt = analogRead( pinKey ); //讀按鍵類比電壓
  if(volt > 1000 ) return( btnNONE ); //省時間放首位
  if(volt < 50 ) return( btnRIGHT );
  if(volt < 250 ) return( btnUP );
  if(volt < 450 ) return( btnDOWN );
  if(volt < 650 ) return( btnLEFT );
  if(volt < 850 ) return( btnSELECT );
  return( btnNONE ); //例外處理
}

void setup()
{
  for(int i=0 ; i<3 ; i++)
  {
    for(int j=0 ; j<4 ; j++) pinMode(pinStep[i][j], OUTPUT);
    dir[i] = CCW;
    remain[i] = 0.0;
  }
  phase[i] = 0;
  focus = BASE;

  textLCD.begin(16, 2); //初始液晶螢幕
  textLCD.clear(); //清空液晶螢幕
  textLCD.setCursor(0, 0); //游標位置歸零
  textLCD.print("FOCUS : BASE");

  gripper.attach(44, 500, 2400); //修正脈波寬度
  gripper.write( 0 ); //夾爪狀態閉合
  grip = CLOSE;
  pinMode(pinFire, INPUT_PULLUP); //二軸搖桿按鍵
}

void loop()
{
  /// 有壓按鍵 ///
  keyin = readLCDkeys();
  switch( keyin )
  {
    case btnRIGHT : textLCD.setCursor( 8,0); textLCD.print("ELBO" ); focus = ELBO; break;
    case btnUP : textLCD.setCursor(13,0); textLCD.print("CCW" ); dir[focus]=CCW ; break;
    case btnDOWN : textLCD.setCursor(13,0); textLCD.print("CW" ); dir[focus]=CW ; break;
    case btnLEFT : textLCD.setCursor( 8,0); textLCD.print("UARM" ); focus = UARM; break;
    case btnSELECT : textLCD.setCursor( 8,0); textLCD.print("BASE" ); focus = BASE; break;
  }
  if(keyin==btnUP || keyin==btnDOWN) rotate(dir[focus], 0.087890625*2.0);

  /// 有開閉夾爪 ///
  if(digitalRead(pinFire) == LOW)
  {
    if(grip == CLOSE){ gripper.write(45); grip=OPEN ; }
    else if(grip == OPEN){ gripper.write( 0); grip=CLOSE; }
    delay( 1000 );
  }

  /// 有拉搖桿 ///
  volx = analogRead( pinVrx );
  voly = analogRead( pinVry );
  if(voly < 300) { focus=ELBO; dir[ELBO]=CW; rotate(dir[focus], 0.087890625*2.0); }
  if(voly > 700) { focus=ELBO; dir[ELBO]=CCW; rotate(dir[focus], 0.087890625*2.0); }
  if(volx < 300) { focus=UARM; dir[UARM]=CW; rotate(dir[focus], 0.087890625*2.0); }
  if(volx > 700) { focus=UARM; dir[UARM]=CCW; rotate(dir[focus], 0.087890625*2.0); }
}

```

## 8. 機械手臂運動學分析

截至目前，本研究已經預計完成之基本目標，達成諸如：機構概念設計、自由度分析、零組件採購、雷雕路徑繪製、步進馬達安裝、木作組件組裝、訊號線配線工程、馬達正反轉程式撰寫、消除馬達步數累積誤差程式撰寫、夾爪開合程式撰寫…等機械手臂控制之基本要求；接著為了延伸手動控制之基本要求成果，本研究額外延伸至夾爪座標自動控制功能。

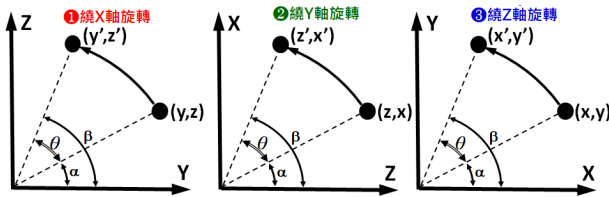
### 8.1 二維旋轉之座標轉換

因機械手臂由基座馬達一直向上驅動至夾爪，各軸前端馬達軸之旋轉必定造成末端座標之轉換，故有必要先行研究機器人運動學之座標轉換理論。

假設座標系XYZ三座標軸符合右手法則，亦即右手四指由X軸掃到Y軸，則拇指指示方向則為Z軸；若有一座標點(x,y,z)經過某一單軸θ角度之旋轉轉換，則該座標轉動方式允許採用「繞軸旋轉」或「軸自旋轉」達到相同之旋轉效果。

繞軸旋轉如圖四十三所示，三種座標點xyz之相互極座標關係如式(3)，其中α、β各為旋轉前後與座標軸之夾角，則利用和差化積可得式(4)，再轉成二維矩陣表示式，可分別得到繞X、Y、Z軸之二維旋轉之轉換矩陣如式(5)。





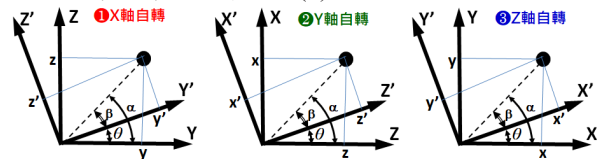
圖四十三 座標點「繞軸旋轉」示意圖

$$\begin{cases} r^2 = y^2 + z^2 \\ y = r \cos \alpha \\ z = r \sin \alpha \end{cases} \quad \begin{cases} r^2 = z^2 + x^2 \\ z = r \cos \alpha \\ x = r \sin \alpha \end{cases} \quad \begin{cases} r^2 = x^2 + y^2 \\ x = r \cos \alpha \\ y = r \sin \alpha \end{cases} \quad (3)$$

$$\begin{cases} y' = r \cos \beta = r \cos(\theta + \alpha) = r \cos \theta \cos \alpha - r \sin \theta \sin \alpha = y \cos \theta - z \sin \theta \\ z' = r \sin \beta = r \sin(\theta + \alpha) = r \sin \theta \cos \alpha + r \cos \theta \sin \alpha = y \sin \theta + z \cos \theta \\ z' = r \cos \beta = r \cos(\theta + \alpha) = r \cos \theta \cos \alpha - r \sin \theta \sin \alpha = z \cos \theta - x \sin \theta \\ x' = r \sin \beta = r \sin(\theta + \alpha) = r \sin \theta \cos \alpha + r \cos \theta \sin \alpha = z \sin \theta + x \cos \theta \\ x' = r \cos \beta = r \cos(\theta + \alpha) = r \cos \theta \cos \alpha - r \sin \theta \sin \alpha = x \cos \theta - y \sin \theta \\ y' = r \sin \beta = r \sin(\theta + \alpha) = r \sin \theta \cos \alpha + r \cos \theta \sin \alpha = x \sin \theta + y \cos \theta \end{cases} \quad (4)$$

$$\begin{bmatrix} y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} \quad \text{或} \quad \begin{bmatrix} z' \\ x' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} z \\ x \end{bmatrix} \quad \text{或} \quad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (5)$$

同理，軸自旋轉如圖四十四所示，三種座標點xyz之相互極座標關係如式(6)，其中 $\alpha$ 、 $\beta$ 各為旋轉前後與座標軸之夾角，則利用和差化積可得式(7)，再轉成二維矩陣表示式，可分別得到繞X、Y、Z軸之二維旋轉之轉換矩陣如式(8)。



圖四十四 座標點「軸自旋轉」示意圖

$$\begin{cases} r^2 = y'^2 + z'^2 \\ y' = r \cos \alpha \\ z' = r \sin \alpha \end{cases} \quad \begin{cases} r^2 = z'^2 + x'^2 \\ z' = r \cos \alpha \\ x' = r \sin \alpha \end{cases} \quad \begin{cases} r^2 = x'^2 + y'^2 \\ x' = r \cos \alpha \\ y' = r \sin \alpha \end{cases} \quad (6)$$

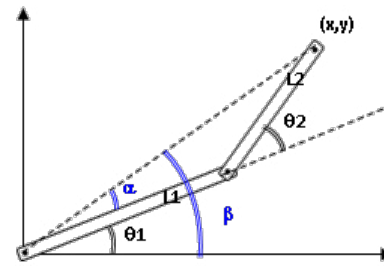
$$\begin{cases} y' = r \cos \beta = r \cos(\alpha - \theta) = r \cos \alpha \cos \theta + r \sin \alpha \sin \theta = y \cos \theta + z \sin \theta \\ z' = r \sin \beta = r \sin(\alpha - \theta) = r \sin \alpha \cos \theta - r \cos \alpha \sin \theta = z \cos \theta - y \sin \theta \\ z' = r \cos \beta = r \cos(\alpha - \theta) = r \cos \alpha \cos \theta + r \sin \alpha \sin \theta = z \cos \theta + x \sin \theta \\ x' = r \sin \beta = r \sin(\alpha - \theta) = r \sin \alpha \cos \theta - r \cos \alpha \sin \theta = z \sin \theta - x \cos \theta \\ x' = r \cos \beta = r \cos(\alpha - \theta) = r \cos \alpha \cos \theta + r \sin \alpha \sin \theta = x \cos \theta + y \sin \theta \\ y' = r \sin \beta = r \sin(\alpha - \theta) = r \sin \alpha \cos \theta - r \cos \alpha \sin \theta = y \cos \theta - x \sin \theta \end{cases} \quad (7)$$

$$\begin{bmatrix} y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} \quad \text{或} \quad \begin{bmatrix} z' \\ x' \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} z \\ x \end{bmatrix} \quad \text{或} \quad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (8)$$

因繞軸旋轉 $\theta$ 角及軸自旋轉 $\theta$ 角，座標點(x,y,z)往相反方向移動，需要把二者其中任一 $\theta$ 角變號，又 $\sin(-\theta)$ 為奇函數需變號、 $\cos(-\theta)$ 為偶函數不變號，最終二邊公式會一模一樣；因此：不管是繞軸旋轉或軸自旋轉，只要轉角 $\theta$ 方向相反，其公式相同座標一樣！

## 8.2 二維機械手臂正轉換之三角解析

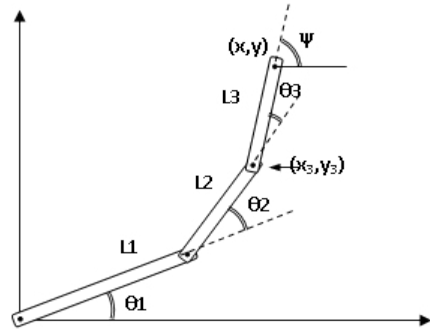
所謂機械手臂之「正轉換運動(Forward Kinematics)分析」，係指先輸入各馬達之轉角，再計算出手臂末端座標；若以二自由度機器手臂為例，如圖四十五所示，其中 $L_1$ 為各段手臂長度、 $\theta_1$ 為各軸馬達之轉角，若輸入 $(\theta_1, \theta_2)$ 欲求手臂末端座標(x,y)，則(x,y)如式(9)所示：



圖四十五 二自由度機器手臂示意圖

$$\begin{cases} \langle x, y \rangle = \langle L_1 \cos \theta_1, L_1 \sin \theta_1 \rangle + \langle L_2 \cos(\theta_1 + \theta_2), L_2 \sin(\theta_1 + \theta_2) \rangle \\ x = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) \\ y = L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2) \end{cases} \quad (9)$$

再以三自由度之機器手臂為例，如圖四十六所示，若輸入馬達各軸轉角 $(\theta_1, \theta_2, \theta_3)$ 欲求手臂末端座標(x,y, $\psi$ )，其中 $\psi$ 為最末段手臂相對水平之擺向，則其正轉換推導如式(10)所示：



圖四十六 三自由度機器手臂示意圖

$$\begin{cases} x = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ y = L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3) \\ \psi = \theta_1 + \theta_2 + \theta_3 \end{cases} \quad (10)$$

## 8.3 二維機械手臂逆轉換之三角解析

前述之正轉換運動座標分析比較直觀，且三角解析過程相當簡單；反之逆轉換運動分析在數學解析上相當困難，甚至會有多重解產生，亦即給定一個手臂末端座標，卻有可能產生多組馬達轉角之解，均皆可移動至該目的座標。

若以二自由度之機器手臂為例，若輸入手臂末端座標(x,y)欲求各軸馬達之轉角 $(\theta_1, \theta_2)$ ，如式(11)所示，將上式(1)與下式(2)先予以平方得式(12)，再將兩者相加合併以消除 $\theta_1$ ，得一簡單之式(13)，據此來解出其中一個轉角 $\theta_2$ 如式(14)所示；接著式(15)引入 $\alpha$ 與 $\beta$ 分別計算其各別tan表示式，式(16)再利用 $\theta_1 = \beta - \alpha$ 之關係求其 $\tan \theta_1$ ，依tan和差化積公式將 $\tan \alpha$ 及 $\tan \beta$ 置換成純粹 $\theta_2$ 之表示式，最終解得剩餘之另一轉角 $\theta_1$ 。

$$\begin{cases} x = L_1 \cos \theta_1 + L_2 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) \cdots \cdots (1) \\ y = L_1 \sin \theta_1 + L_2 (\sin \theta_1 \cos \theta_2 + \cos \theta_1 \sin \theta_2) \cdots \cdots (2) \end{cases} \quad (11)$$

$$\begin{cases} x^2 = L_1^2 \cos^2 \theta_1 + 2L_1L_2(\cos^2 \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_1 \sin \theta_2) \\ \quad + L_2^2(\cos^2 \theta_1 \cos^2 \theta_2 - 2\cos \theta_1 \cos \theta_2 \sin \theta_1 \sin \theta_2 + \sin^2 \theta_1 \sin^2 \theta_2) \dots \dots (1)^2 \\ y^2 = L_1^2 \sin^2 \theta_1 + 2L_1L_2(\sin^2 \theta_1 \cos \theta_2 + \sin \theta_1 \cos \theta_1 \sin \theta_2) \\ \quad + L_2^2(\sin^2 \theta_1 \cos^2 \theta_2 + 2\sin \theta_1 \cos \theta_2 \cos \theta_1 \sin \theta_2 + \cos^2 \theta_1 \sin^2 \theta_2) \dots \dots (2)^2 \end{cases} \quad (12)$$

$$x^2 + y^2 = L_1^2 + 2L_1L_2 \cos \theta_2 + L_2^2(\cos^2 \theta_2 + 0 + \sin^2 \theta_2) \quad (13)$$

$$\cos \theta_2 = \frac{x^2 + y^2 - L_1^2 - L_2^2}{2L_1L_2} \quad (14)$$

$$\begin{cases} \tan \alpha = \frac{\text{對邊}}{\text{臨邊}} = \frac{L_2 \sin \theta_2}{L_1 + L_2 \cos \theta_2} \\ \tan \beta = \frac{y}{x} \end{cases} \quad (15)$$

$$\begin{aligned} \tan \theta_1 &= \tan(\beta - \alpha) \\ &= \frac{\tan \alpha - \tan \beta}{1 + \tan \alpha \tan \beta} \\ &= \frac{\frac{L_2 \sin \theta_2}{L_1 + L_2 \cos \theta_2} - \frac{y}{x}}{1 + \frac{L_2 \sin \theta_2}{L_1 + L_2 \cos \theta_2} \cdot \frac{y}{x}} \\ &= \frac{xL_2 \sin \theta_2 - y(L_1 + L_2 \cos \theta_2)}{x(L_1 + L_2 \cos \theta_2) + yL_2 \sin \theta_2} \\ &= \frac{xL_2 \sin \theta_2 - y(L_1 + L_2 \cos \theta_2)}{x(L_1 + L_2 \cos \theta_2) + yL_2 \sin \theta_2} \end{aligned}$$

$$\text{亦即 } \tan \theta_1 = \frac{xL_2 \sin \theta_2 - y(L_1 + L_2 \cos \theta_2)}{x(L_1 + L_2 \cos \theta_2) + yL_2 \sin \theta_2} \quad (16)$$

接著以三自由度之機器手臂為例，若輸入手臂末端座標及水平擺向(x,y,ψ)為已知，欲求各軸馬達之轉角(θ<sub>1</sub>,θ<sub>2</sub>,θ<sub>3</sub>)，亦即3已知項欲解3未知項，但位置表示式僅能提供2未知項線索，還需加上ψ=θ<sub>1</sub>+θ<sub>2</sub>+θ<sub>3</sub>之額外線索方可求解；故先將位置之表示式拆成上式(1)與下式(2)如式(17)，為了先解出θ項，式(18)把含有已知項ψ搬到等號左邊，再以式(19)將上下兩式分別平方，用式(20)將其相加及合併整理，式(21)將已知項ψ及未知項θ分別搬到等式左邊與右邊，令式(22)含已知項ψ之等號左邊假設為Δ，則式(23)可利用與已知Δ與等號右邊之關係來解得未知θ<sub>2</sub>。

$$\begin{cases} x = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\psi) \dots \dots (1) \\ y = L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\psi) \dots \dots (2) \end{cases} \quad (17)$$

$$\begin{cases} x - L_3 \cos(\psi) = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) \\ y - L_3 \sin(\psi) = L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2) \end{cases} \quad (18)$$

$$\begin{cases} [x - L_3 \cos(\psi)]^2 = L_1^2 \cos^2 \theta_1 \\ \quad + 2L_1L_2 \cos \theta_1 \cos(\theta_1 + \theta_2) \\ \quad + L_2^2 \cos^2(\theta_1 + \theta_2) \\ [y - L_3 \sin(\psi)]^2 = L_1^2 \sin^2 \theta_1 \\ \quad + 2L_1L_2 \sin \theta_1 \sin(\theta_1 + \theta_2) \\ \quad + L_2^2 \sin^2(\theta_1 + \theta_2) \end{cases} \quad (19)$$

$$[x - L_3 \cos(\psi)]^2 + [y - L_3 \sin(\psi)]^2 = L_1^2 + 2L_1L_2[\sin \theta_1 \sin(\theta_1 + \theta_2) + \cos \theta_1 \cos(\theta_1 + \theta_2)] + L_2^2 \quad (20)$$

$$\begin{aligned} & \frac{[x - L_3 \cos(\psi)]^2 + [y - L_3 \sin(\psi)]^2 - L_1^2 - L_2^2}{2L_1L_2} \\ &= \sin \theta_1 \sin(\theta_1 + \theta_2) + \cos \theta_1 \cos(\theta_1 + \theta_2) \end{aligned} \quad (21)$$

$$\Delta = \frac{[x - L_3 \cos(\psi)]^2 + [y - L_3 \sin(\psi)]^2 - L_1^2 - L_2^2}{2L_1L_2} \quad (22)$$

$$\begin{aligned} \Delta &= \sin \theta_1 \sin(\theta_1 + \theta_2) + \cos \theta_1 \cos(\theta_1 + \theta_2) \\ &= \cos(\theta_1 + \theta_2) \cos \theta_1 + \sin(\theta_1 + \theta_2) \sin \theta_1 \\ &= \cos(\theta_1 + \theta_2 - \theta_1) \\ &= \cos \theta_2 \\ \theta_2 &= \cos^{-1} \Delta \end{aligned} \quad (23)$$

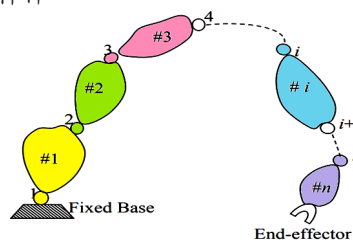
求出θ<sub>2</sub>後，利用ψ=θ<sub>1</sub>+θ<sub>2</sub>+θ<sub>3</sub>之關係，轉成θ<sub>1</sub>=ψ-θ<sub>2</sub>-θ<sub>3</sub>，再代入位置表示式，利用上述解θ<sub>2</sub>之類似手法，將θ<sub>1</sub>解出來，便可知θ<sub>3</sub>。

### 九、Denavit-Hartenberg法則應用探討

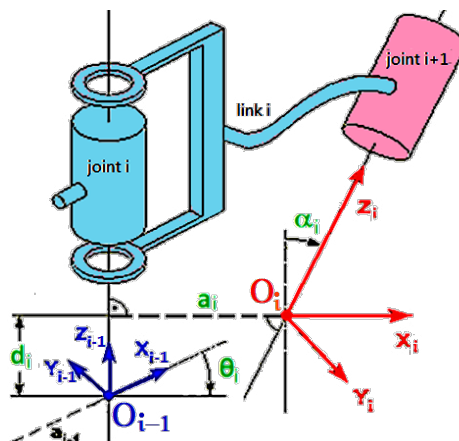
由前述範例可知，二維機械手臂之正轉換僅需簡單地解析角度來寫出座標式，但逆轉換之三角解析卻是相當複雜，解二維位置聯立式將會產生很大心力及痛苦，倘若擴展至三維機械手臂，三角解析法幾乎在實務上變得不可行；有鑑於此，三維之D-H矩陣法便應運而生了。

#### 9.1 D-H法則之四類參數介紹

為導入D-H座標轉換矩陣法[7]，首先須先規定各段手臂桿件各自擁有前端之關節，如圖四十七所示，若#0表示基座，則共有#0-#n合計n+1之桿件，介於各段桿件間之關節則合計有n個，每個關節均隸屬於接續其後之桿件。



圖四十七 關節與各段手臂桿件之歸屬示意圖



圖四十八 關節與各段手臂桿件之歸屬示意圖

在圖四十八中，因本法採用平移或旋轉之座標



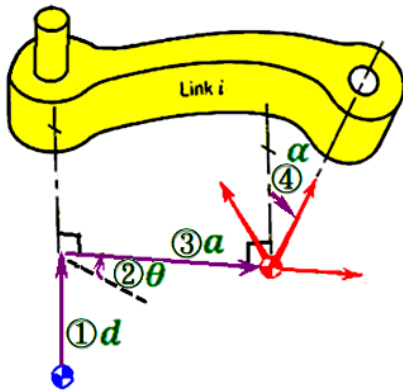
轉換矩陣，加上旋轉矩陣係讓座標點之旋轉半徑與關節軸正交之故，使得轉換過度座標點係由前關節軸之最近直交點移動到後關節軸之最近直交點。如表六之參數表所示，D-H法依轉換順序定義四類基本參數  $d$ 、 $\theta$ 、 $a$ 、 $\alpha$ ，分別為「層差」、「旋轉」、「軸移」以及「扭擺」；若以轉換過度點當成各關節軸之原點，則層差  $d_i$  就是由本關節原點沿著本關節 Z 軸爬升到對應次關節最近點的高度(使用平移矩陣)；旋轉  $\theta_i$  就是將以本關節 Z 軸旋轉，以讓 X 軸朝向次關節 X 軸所需之角度(使用旋轉矩陣)；軸移  $a_i$  就是過度點沿最近直交線從本關節過渡點 X 軸移到次關節原點的最短距離(使用平移矩陣)；扭擺  $\alpha_i$  就是讓已移到次關節之本關節 Z 軸重新轉到次關節 Z 軸所需角度(使用旋轉矩陣)。

表六 D-H法之四類基本參數

D-H Parameter Table				
序號	層差	旋轉	軸移	扭擺
$i$	$d_i$	$\theta_i$	$a_i$	$\alpha_i$
方向	Z移	Z轉	X移	X轉

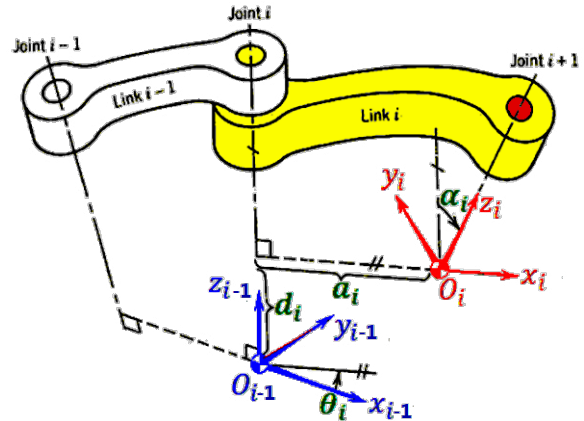
### 9.2 D-H法則之公式推導

推導公式前需再次強調，如圖四十九所示，各關節旋轉軸為該關節 Z 軸，原點為該關節之最近過渡點，轉換係平移-旋轉-平移-旋轉四個過程，先在本關節進行 Z 旋轉來對正 X 軸，後在次關節進行 X 旋轉來對正 Z 軸。



圖四十九 平移-旋轉-平移-旋轉四個轉換順序圖

D-H法則係根據此四個過程，連續進行矩陣座標轉換，如圖五十及式(24)所示， $A$  為第  $i-1$  關節轉換至第  $i$  關節之合成轉換矩陣， $T$  為平移轉換矩陣， $R$  為旋轉轉換矩陣；若  $s$  表  $\sin$ 、 $c$  表  $\cos$ ，則合成轉換矩陣即為平移  $T(z,d)$ -旋轉  $R(z,\theta)$ -平移  $T(x,a)$ -旋轉  $R(x,\alpha)$  之連續乘積，括號內之前索引為平移方向或旋轉軸，後索引為距離或角度。



圖五十 相鄰兩關節之D-H法則公式推導

$$A_i^{i-1} = T(z,d) \cdot R(z,\theta) \cdot T(x,a) \cdot R(x,\alpha)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta & -s\theta & 0 & 0 \\ s\theta & c\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha & -s\alpha & 0 \\ 0 & s\alpha & c\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (24)$$

最後將式(24)整理合併後，得出式(25)之最後由  $i-1$  關節轉換至  $i$  關節之正轉換矩陣，為了未來計算座標方便，式(26)亦將其逆轉換之反矩陣一併算出來。

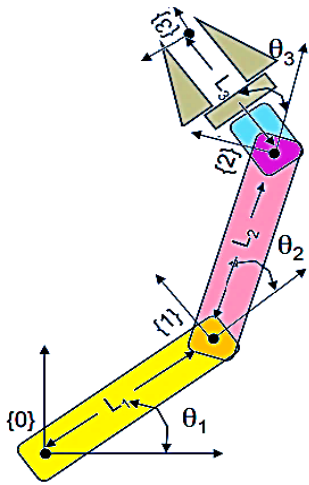
$$\text{正轉換} \quad A_i^{i-1} = \begin{bmatrix} c\theta_i & -c\alpha_i s\theta_i & s\alpha_i s\theta_i & a_i c\theta_i \\ s\theta_i & c\alpha_i c\theta_i & -s\alpha_i c\theta_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (25)$$

$$\text{反矩陣} \quad A_{i-1}^i = (A_i^{i-1})^{-1} = \begin{bmatrix} c\theta_i & s\theta_i & 0 & -a_i \\ -c\alpha_i s\theta_i & c\alpha_i c\theta_i & s\alpha_i & -d_i s\alpha_i \\ s\alpha_i s\theta_i & -s\alpha_i c\theta_i & c\alpha_i & -d_i c\alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (26)$$

### 9.3 D-H法則正轉換之應用範例

茲先將D-H法則應用於二維機械手臂(如圖五十一)之正轉換運動學分析，其中標號0為基座、標號1為上臂、標號2為下臂、標號3為夾爪；因二維機構各轉軸均垂直紙面，故各關節均無層差  $d$  及扭擺  $\alpha$ ，則其D-H法則參數表詳見表七，須注意表中  $\theta_1$ 、 $\theta_2$  及  $\theta_3$  均為待輸入之未知參數， $L_1$ 、 $L_2$  及  $L_3$  均為已知參數；接著將此表參數數值各別套入式(25)之正轉換基本公式中，可得出三個關節之正轉換矩陣  ${}^0A_1$ 、 ${}^1A_2$  與  ${}^2A_3$ ，如式(27)所示，接著作連續矩陣相乘積，最終式(28)可得到描述夾爪末端座標點之正轉換矩陣  ${}^0T_3$  了。





圖五十一 二維機械手臂之正轉換運動學分析

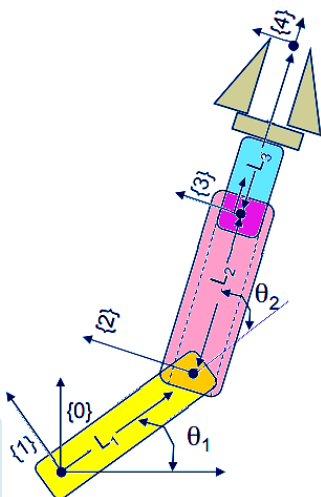
表七 二維機械手臂之D-H法則參數表

Link	$d_i$	$\theta_i$	$a_i$	$\alpha_i$
1	0	$\theta_1$	$L_1$	0
2	0	$\theta_2$	$L_2$	0
3	0	$\theta_3$	$L_3$	0

$${}^0A_1 = \begin{bmatrix} c_{\theta_1} & -s_{\theta_1} & 0 & L_1 c_{\theta_1} \\ s_{\theta_1} & c_{\theta_1} & 0 & L_1 s_{\theta_1} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^1A_2 = \begin{bmatrix} c_{\theta_2} & -s_{\theta_2} & 0 & L_2 c_{\theta_2} \\ s_{\theta_2} & c_{\theta_2} & 0 & L_2 s_{\theta_2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^2A_3 = \begin{bmatrix} c_{\theta_3} & -s_{\theta_3} & 0 & L_3 c_{\theta_3} \\ s_{\theta_3} & c_{\theta_3} & 0 & L_3 s_{\theta_3} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (27)$$

$${}^0T_3 = {}^0A_1 {}^1A_2 {}^2A_3 = \begin{bmatrix} c_{a_{23}} & -s_{a_{23}} & 0 & L_1 c_{\theta_1} + L_2 c_{a_{23}} + L_3 c_{a_{23}} \\ s_{a_{23}} & c_{a_{23}} & 0 & L_1 s_{\theta_1} + L_2 s_{a_{23}} + L_3 s_{a_{23}} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (28)$$

倘若將剛才二維機械手臂之下臂改成伸縮電動缸，如圖五十二所示，則該參數表要隨之修正如表八，須注意 $\theta_1$ 及 $\theta_2$ 為未知， $L_1$ 、 $L_2$ 及 $L_3$ 為已知。

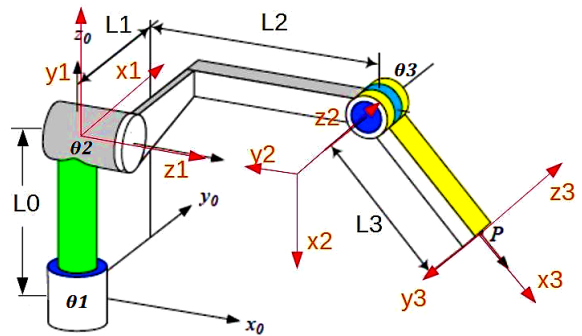


圖五十二 下臂改為電動缸之二維機械手臂

表八 修正之D-H法則參數表

Link	$d_i$	$\theta_i$	$a_i$	$\alpha_i$
1	0	$\theta_1$	0	0
2	0	$\theta_2$	$L_1$	0
3	0	0	$L_2$	0
4	0	0	$L_3$	0

為了增進對D-H法則之了解，進一步研究如何針對三維機械手臂提取其D-H法則之參數表，圖五十三為三維機械手臂之外觀，此時因為各軸方向不一致，層差 $d_i$ 及扭擺 $\alpha_i$ 不再為0，須注意 $\theta_1$ 、 $\theta_2$ 及 $\theta_3$ 為未知以外，其餘參數均為已知之幾何長度及角度數據。



圖五十三 三維機械手臂之正轉換運動學分析

表九 三維機械手臂之D-H法則參數表

Link	$d_i$	$\theta_i$	$a_i$	$\alpha_i$
1	$L_0$	$\theta_1$	0	$\pi/2$
2	$L_2$	$\theta_2$	0	$-\pi/2$
3	$L_1$	$\theta_3$	$L_3$	0

### 9.4 D-H法則逆轉換之應用範例

接著介紹D-H矩陣分析法之逆轉換，也就是由夾爪座標點來求各馬達角度之反向分析；假設某三維機械手臂之D-H參數表如表十所示，將每個關節之參數帶入公式，可得式(29)之各關節之轉換矩陣，將其這三個關節矩陣由左至右相乘積，可得整體之轉換矩陣如式(30)。

表十 三維機械手臂之逆轉換參數表

Joint	$d_i$	$\theta_i$	$a_i$	$\alpha_i$
1	$d_1$	$\theta_1$	0	$90^\circ$
2	0	$\theta_2$	$a_2$	$0^\circ$
3	0	$\theta_3$	$a_3$	$0^\circ$

$$A_1^0 = \begin{bmatrix} c\theta_1 & 0 & s\theta_1 & 0 \\ s\theta_1 & 0 & -c\theta_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, A_2^1 = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & a_2 c\theta_2 \\ s\theta_2 & c\theta_2 & 0 & a_2 s\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, A_3^2 = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & a_3 c\theta_3 \\ s\theta_3 & c\theta_3 & 0 & a_3 s\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (29)$$

$$A_3^0 = \begin{bmatrix} c\theta_1 c(\theta_2 + \theta_3) & -c\theta_1 s(\theta_2 + \theta_3) & s\theta_1 & c\theta_1 [a_3 c(\theta_2 + \theta_3) + a_2 c\theta_2] \\ s\theta_1 c(\theta_2 + \theta_3) & -s\theta_1 s(\theta_2 + \theta_3) & -c\theta_1 & s\theta_1 [a_3 c(\theta_2 + \theta_3) + a_2 c\theta_2] \\ s(\theta_2 + \theta_3) & c(\theta_2 + \theta_3) & 0 & a_3 s(\theta_2 + \theta_3) + a_2 s\theta_2 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (30)$$



假設夾爪原點相對於自身座標系為 $p^3=[0 \ 0 \ 0 \ 1]^T$ ，經逆轉換至基座座標系 $0$ 之座標值為輸入已知 $p^0=[p_x \ p_y \ p_z \ 1]^T$ ，則式(31)為其帶入座標轉換方程式，得到 $p_x$ 、 $p_y$ 、 $p_z$ 與馬達 $\theta_1$ 、 $\theta_2$ 、 $\theta_3$ 之三個關係式。前二式互相相比，便可消去 $\theta_2$ 及 $\theta_3$ 得到 $\theta_1$ ，如式(32)所示：

$$\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = {}^0A_3 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} c\theta_1[a_3c(\theta_2+\theta_3)+a_2c\theta_2] \\ s\theta_1[a_3c(\theta_2+\theta_3)+a_2c\theta_2] \\ a_3s(\theta_2+\theta_3)+a_2s\theta_2+d_1 \\ 1 \end{bmatrix} \quad (31)$$

$$\begin{cases} p_x = c\theta_1[a_3c(\theta_2+\theta_3)+a_2c\theta_2] \\ p_y = s\theta_1[a_3c(\theta_2+\theta_3)+a_2c\theta_2] \\ \frac{p_y}{p_x} = \frac{s\theta_1}{c\theta_1} = \tan\theta_1 \\ \theta_1 = \tan^{-1}\frac{p_y}{p_x} \end{cases} \quad (32)$$

此時增設 $k_x$ 與 $k_y$ 以簡化計算，若 $\cos\theta_1 > 0.01$ ，則令 $k_x = p_x / \cos\theta_1$ ，否則令 $k_x = p_y / \sin\theta_1$ ，同時讓 $\theta_2 + \theta_3$ 以 $\theta_{23}$ 取代如式(33)所示，接著式(34)將 $k_x$ 及 $k_y$ 平方後相加，得到類似三角形餘弦公式，藉由此獲得 $\theta_3$ 之解，但請注意若 $\cos$ 之內分數值 $\geq 1.0$ ，則請設 $\theta_3 = 0.0$ 。

$$\begin{aligned} k_x &\leftarrow \frac{p_x}{c\theta_1} = a_3c(\theta_2+\theta_3)+a_2c\theta_2 = a_2c\theta_2+a_3c\theta_{23} \\ \frac{p_y}{s\theta_1} &= a_3c(\theta_2+\theta_3)+a_2c\theta_2 \\ k_y &\leftarrow p_z - d_1 = a_3s(\theta_2+\theta_3)+a_2s\theta_2 = a_2s\theta_2+a_3s\theta_{23} \end{aligned} \quad (33)$$

$$\begin{aligned} k_x^2 + k_y^2 &= (a_2^2c^2\theta_2 + 2a_2a_3c\theta_2c\theta_{23} + a_3^2c^2\theta_{23}) \\ &\quad + (a_2^2s^2\theta_2 + 2a_2a_3s\theta_2s\theta_{23} + a_3^2s^2\theta_{23}) \\ &= a_2^2 + 2a_2a_3(c\theta_2c\theta_{23} + s\theta_2s\theta_{23}) + a_3^2 \\ &= a_2^2 + a_3^2 + 2a_2a_3c[\theta_2 - (\theta_2 + \theta_3)] \\ &= a_2^2 + a_3^2 + 2a_2a_3c\theta_3 \\ \theta_3 &= \cos^{-1} \frac{k_x^2 + k_y^2 - a_2^2 - a_3^2}{2a_2a_3} \end{aligned} \quad (34)$$

既然 $\theta_1$ 及 $\theta_3$ 均已解出，僅剩下 $\theta_2$ 欲未解，式(35)用和差化積拆開後，再以 $\theta_2$ 之 $\sin$ 及 $\cos$ 項作為分類原則整理之，同時以 $S$ 代替 $\sin$ 項、以 $C$ 代替 $\cos$ 項，以使後續式子簡潔。式(36)更進一步用 $L$ 及 $M$ 取代已解出 $\theta_3$ 之相關項，得到一簡潔之聯立方程式，此時 $\theta_2$ 便藏在 $S$ 與 $C$ 當中。

$$\begin{cases} k_x = a_2c\theta_2 + a_3(c\theta_2c\theta_3 - s\theta_2s\theta_3) = c\theta_2(a_2 + a_3c\theta_3) - a_3s\theta_2s\theta_3 \\ k_y = a_2s\theta_2 + a_3(s\theta_2c\theta_3 + c\theta_2s\theta_3) = s\theta_2(a_2 + a_3c\theta_3) + a_3c\theta_2s\theta_3 \\ \begin{cases} k_x = c\theta_2(a_2 + a_3c\theta_3) - a_3s\theta_2s\theta_3 \\ k_y = s\theta_2(a_2 + a_3c\theta_3) + a_3c\theta_2s\theta_3 \\ k_x = (a_2 + a_3c\theta_3)C - a_3s\theta_2S \\ k_y = (a_2 + a_3c\theta_3)S + a_3s\theta_2C \end{cases} \end{cases} \quad (35)$$

$$\begin{cases} k_x = (a_2 + a_3c\theta_3)C - a_3s\theta_2S \\ k_y = (a_2 + a_3c\theta_3)S + a_3s\theta_2C \\ \begin{cases} k_x = LC - MS \\ k_y = LS + MC \end{cases} \end{cases} \quad (36)$$

針對聯立方程式，使用平衡係數兩兩消去法，得到 $S$ 如式(37)、得到 $C$ 如式(38)。式(39)再利用反正

切函數求得 $S$ 跟 $C$ 之比值，從而解出 $\theta_2$ ；須注意 $\theta_2$ 跑數值分析時，若 $C$ 與 $|S-1|$ 皆 $\leq 0.01$ ，則此時奇異點須讓 $\theta_2 = 90$ 度，否則令 $\theta_2 = \text{atan2}(S, C)$ 。

$$\begin{cases} Mk_x = LMC - M^2S \\ Lk_y = L^2S + LMC \\ Lk_y - Mk_x = L^2S + M^2S = (L^2 + M^2)S \\ S = \frac{Lk_y - Mk_x}{L^2 + M^2} \\ S = \frac{(a_2 + a_3c\theta_3)k_y - a_3s\theta_3k_x}{(a_2 + a_3c\theta_3)^2 + a_3^2s^2\theta_3} = \frac{(a_2 + a_3c\theta_3)k_y - a_3s\theta_3k_x}{a_2^2 + a_3^2 + 2a_2a_3c\theta_3} \end{cases} \quad (37)$$

$$\begin{cases} Lk_x = L^2C - LMS \\ Mk_y = LMS + M^2C \\ Lk_x + Mk_y = L^2C + M^2C = (L^2 + M^2)C \\ C = \frac{Lk_x + Mk_y}{L^2 + M^2} \\ C = \frac{(a_2 + a_3c\theta_3)k_x - a_3s\theta_3k_y}{(a_2 + a_3c\theta_3)^2 + a_3^2s^2\theta_3} = \frac{(a_2 + a_3c\theta_3)k_x + a_3s\theta_3k_y}{a_2^2 + a_3^2 + 2a_2a_3c\theta_3} \end{cases} \quad (38)$$

$$\begin{aligned} \tan\theta_2 &= \frac{\sin\theta_2}{\cos\theta_2} = \frac{S}{C} \\ \rightarrow \theta_2 &= \tan^{-1} \frac{S}{C} \\ &= \tan^{-1} \frac{(a_2 + a_3c\theta_3)k_y - a_3s\theta_3k_x}{(a_2 + a_3c\theta_3)k_x - a_3s\theta_3k_y} \end{aligned} \quad (39)$$

## 十、結論與展望

本研究已完成雷雕機硬體組裝、雷雕機軟韌體控制、機械手臂概念設計、組件雷雕路徑繪製、機械手臂 Arduino 電路配線、機械手臂 Arduino 程式設計...等預期手動控制須完成之成果。

同時本研究也針對機械手臂正/逆轉換運動學分析以及 Denavit-Hartenberg 法則座標正/逆轉換矩陣之計算，進行了基礎理論研究，未來將可繼續朝向機械手臂自動定位功能邁進，並加以結合影像處理，將可達成智慧取物、寫字機或書法機之進階功能。

## 誌謝

本研究案為威綸科技股份有限公司與黎明技術學院產學合作成果，承蒙威綸科技股份有限公司提供研究經費、材料、設備使用，特致上最深謝忱。

## 參考文獻

1. WOODPECKER CNC 裝配指南，河南倫業進出口貿易有限公司，2016年5月。
2. GRBL Atmega328P-MU-V11 Controller 設置說明書，河南倫業進出口貿易有限公司，2011年。
3. Woodpecker CNC 使用指南，河南倫業進出口貿易有限公司，2011年3月。
4. 微雕管家使用說明，河南倫業進出口貿易有限公司，2011年。

5. Configuring Grbl, [https:// github.com/grbl/grbl/wiki](https://github.com/grbl/grbl/wiki) · 2019 .
6. Industrial Robots and Robot System Safety, Occupational Safety and Health Administration; Section IV:Chapter 4, <https://www.osha.gov>.
7. Introduction to Robotics, S.K. Saha, Tata McGraw-Hill, New Delhi, 2008.

## **The implementation and development of the fashionable and technological micro-controller based wooden robot arm as the cultural and creative product**

Yang, Chi-Shih<sup>1\*</sup>

<sup>1</sup> Lee-Ming Institute of Technology,  
Department of Fashion Design

\*Email: [yjs@mail.lit.edu.tw](mailto:yjs@mail.lit.edu.tw)

### **Abstract**

In recent years, the concepts of "internet of things (IoT)", "smart manufacturing" and "cultural and creative goods" have swept the industry and commerce. The former two need to have a lot of highly specialized knowledge and technology. Among them, the robot arm is the most mainstream representative. This research combines the robot arm and the fashionable technological product to achieve the hardware development and technical research. The results will be based on the creative and scientific sense to demonstrate the microcomputer controlled wooden robot arm for its entertainment and healing function. According to the research and development process, this technical report is introduced in order by the sections of hardware assembly of laser cutting machine, firmware and software control of laser cutting machine, concept design of robot arm, cutting path drawing of components, Arduino circuit wiring, Arduino program designing, robot arm kinematics analysis and the applied discussion of the Denavit-Hartenberg rule.

*Keywords:* Internet of things, Smart manufacturing, Robot arm, Arduino microcomputer control, Laser cutting machine, Denavit-Hartenberg rule, Coordinate transformed matrix, Fashionable technology, Healing function

