

## 第4章 軟體規劃

### 4.1 函式庫架構

軟體設計是用來實現所設計電路的控制，因此各函式庫規劃是以 MCX314 為主要架構，利用其內部的命令、狀態、位置暫存器，撰寫各相關函式庫，最後整合成人機介面，利用此人性化之介面可讓操作和控制更簡單且有效率。

規劃的函式庫之分類與功能如表 4-1 所示：

表 4-1 函式庫之分類與功能表

函數分類	功能簡介
初始化設定函數 (Initial Functions)	設定系統的初始化或程式的初始化功能設定。
各軸狀態偵測函數 (Axis Status Functions)	偵測各軸的運動狀態，包括是否在極限開關上或是否正在移動中。
手動操作模式函數 (Manual Operation Mode Functions)	用以手動操作原點返回或是以手動方式移動。
運動控制命令函數 (Motion Control Command Functions)	包括基本運動命令函數，其中有定量驅動、連續驅動、減速停止、立即停止、直線加減速、S-Curve 加減速等基本命令；多軸直線插補函數；圓弧插補函數；位元補間函數和原點返回函數。
中斷控制函數 (Interrupt Control Functions)	用以中斷致能、中斷除能，取得中斷時的狀態。
錯誤訊息函數 (Error Message Box Functions)	可產生錯誤訊息、顯示錯誤的位置和狀態。
位置暫存器管理函數 (Register Management Functions)	包括設動與讀取邏輯位置計數器(LP)的值、設定與讀取實際位置計數器(EP)的值、讀取驅動速度和加減速度。
各軸 I/O 信號函數 (Axis I/O Signal Functions)	包括設定 Alarm(異常)信號的工作模式、設定伺服(Server)的輸出狀態。

軟體函式庫的開發流程，是先以 Turbo C 程式撰寫，然後在 DOS 的作業系統進行運動命令之實驗測試與程式除錯。之後配合人機介面的開發和軟體模組化的觀念，則重新以 Visual C++ 重新將函式庫編譯成動態連結函式庫(Dynamic Link Library, DLL)，並引入相關在視窗作業系統上(98、2000, XP) 關於底層 I/O 的 API 函數，以輸入或讀取運動控制卡的暫存器。視窗程式的部份則以 MFC 來開發實作，並引入所編譯完成後的 DLL 運動函式庫，最後在視窗環境下可以利用開發完成後的人機介面，直接下達運動命令給馬達運動控制卡以驅動馬達。

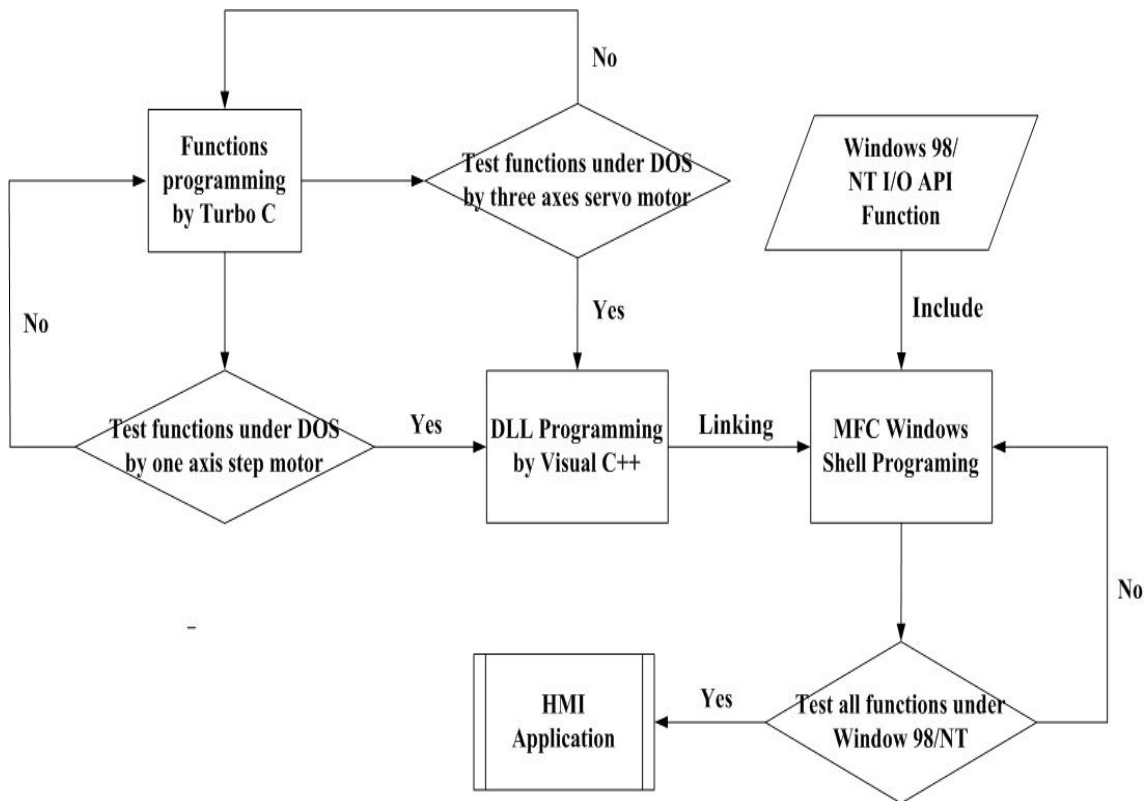


圖 4-1 軟體函式庫之開發流程

## 4.2 動態連結函式庫的介紹

動態連結函式庫(Dynamic Link Library, DLL)是一個包含可以被其他程式使用的模組的檔案，基本上一個 DLL 就是磁碟上的一個檔案，檔案的副檔名通常是.dll。DLL 可以由全域資料、函式、和資源組成，它們都會變成程式中處理程序的一部分。DLL 檔案中擁有可被匯出(export)的函式，當使用者端在載入 DLL 檔案時則會匯入(import)這些函式。

在以往程式中，當使用者建立一個函式，而在別的應用程式呼叫它，此時連結器會從函式庫複製一個副本至應用程式中，雖然使用靜態連結的方式簡單而方便，但由於 Windows 是一個多工的系統，它可以同時執行多個應用程式，如果靜態連結的函式庫同時被多個應用程式所使用，那麼記憶體中就有很多它們的副本，這樣一來程式便變得很沒有效率。然而使用動態連結的方式去連結應用程式，DLL 內的函式只當應用程式執行時才會被連結，因此可以有效率地管理記憶體。如圖 4-2 所示。利用動態連結所編譯的應用程式體積也不致於過大，而且即使更換動態連結函式庫時，也不需要重新編譯或連結使用該動態連結函式庫的應用程式。

圖 4-3 為建立 DLL 專案時的畫面。欲實作單純以 C 或 C++ 的函數，則可以使用 Win32 Dynamic Link Library；欲實作與視窗程式相關或需引入 MFC 類別的函數則使用 MFC AppWizard。當視窗使用 DLL 所需要的檔案必須有 h 檔、lib 檔、dll 檔。其中 h 檔定義了從 DLL 出口入口的項目，讓編譯器可以適當的處理其他程式中用到出口項目的參照。lib 檔則是讓連結器在連結使用 DLL 的程式時，可以處理出口項目的參照。dll 檔則含有函數的可執行碼，當使用 DLL 的程式執行時，dll 檔則會被 Windows 載入。

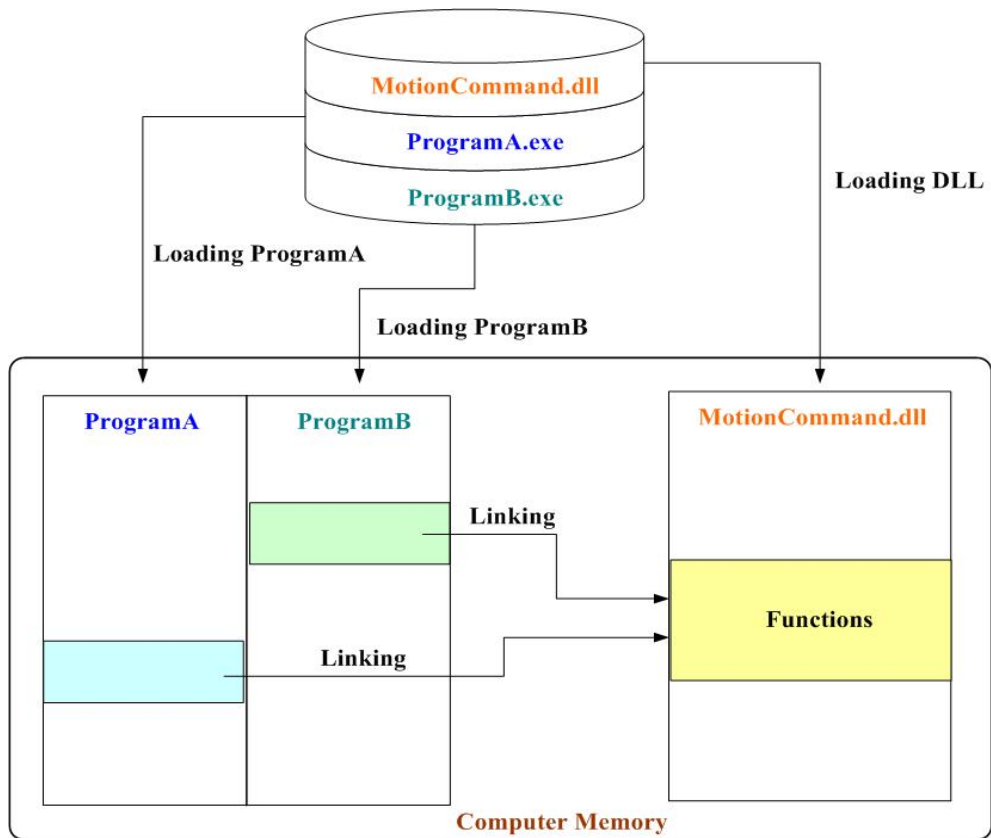


圖 4-2 動態連結函式庫示意圖[8]

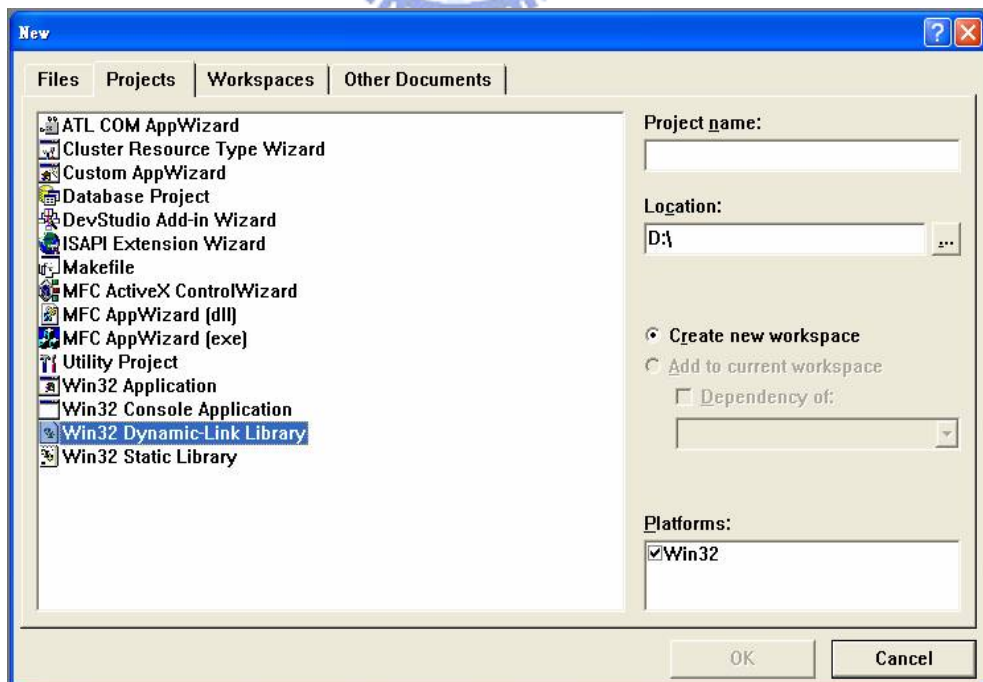


圖 4-3 建立 DLL 的專案

### 4.3 初始化設定函數

初始化設定函數為下達運動命令前，所必須選擇執行的函數，否則運動命令將會失效。包括有選擇輸出脈波模式、設定倍率、設定範圍、規劃運動的最大驅動速度。

表 4-2 初始化定函數

函式	功能
SetPulseMode	選擇輸出脈波模式
SetMultiple	設定倍率
SetRange	設定範圍
SetMaxSpeed	規劃運動的最大驅動速度

- void SetPulseMode(int axis, int mode)

功能說明：此函數可設置脈波輸出模式，承接 3-1 節，可分為單脈波輸出模式與雙脈波輸出模式。相關圖例說明如圖 3-1~圖 3-4。

參數 axis 為選擇之運動軸。

參數 mode 為欲設定之脈波輸出模式。



- void SetMultiple(int axis, int data)

功能說明：此函數可決定各個參數的倍率，如初速度、速度等，這些參數乘以倍率才是實際馬達驅動的實際初速度、實際速度。

參數 data 為所設置之倍率，參數資料範圍為 1~500。

- void SetRange(int data)

功能說明：此函數可設置一範圍參考值，以換算成一適當的倍率。(倍率=8000000/範圍)

參數 data 為欲設定之範圍值，資料範圍為 8000000~16000。

- void SetMaxSpeed( int data)

功能說明：此函數可設置一最大驅動速度作為參考值，以提供一適當的倍率。

參數 data 為欲設定之最大驅動速度值，資料範圍為 1~4000000。

## 4.4 基本運動命令函數

MCX314 產生基本運動命令的條件包括：範圍(R)、倍率(M)、加速度變化率(K)、加速度(A)、減速度(D)、初始速度(SV)、驅動速度(V)和輸出脈衝數(P)的設置。各參數計算公式如下：倍率  $M = \frac{8000000}{R}$ ；實際加速度 =  $A \times 125 \times M$ ；實際減速度 =  $D \times 125 \times C$ ；實際加速度變化率 =  $\frac{62500000}{K} \times M$ ；實際初始速度 =  $SV \times M$ ；實際驅動速度 =  $V \times C$ 。整個基本運動命令處理流程，須先設定相關之初始化的命令暫存器，即可設定各個運動參數，之後可由 CPU 送入命令或資料經由動作管理部(Acting Managing Section)再與相關運動參數相聯繫計算後，再進入理論位置計數器(Logic Position Counter)，最後再將命令送出至驅動器以驅動馬達。

表 4-3 基本運動函式

函式	功能
SetSV	設定驅動初速度
SetV	設定驅動速度
SetA	設定驅動加速度
SetD	設定驅動減速度
SetAA	設定驅動加速度變化率
SetAO	設定減速脈波長度
Pulse	設定輸出脈動數
AutoDec	設定為自動減速
ManDec	設定為手動減速
TCurve_F	設定 T-Curve 加減速為定量驅動
TCurve_C	設定 T-Curve 加減速為連續驅動
SCurve_F	設定 S-Curve 加減速為定量驅動
SCurve_C	設定 S-Curve 加減速為連續驅動

基本運動函數說明如下：

- void SetSV(int axis, int data)

功能說明：此函數可決定在加減速驅動中加速開始和減速結束時的速度。若假設初速度之設定值 SV，則實際驅動初速度如下列公式：

$$\text{驅動初速度(PPS)} = \text{SV} \times \text{倍率(M)}$$

參數 axis 為選擇之運動軸。

參數 data 為欲設定之初速度值，資料範圍為 1~8000。

- void SetV(int axis, int data)

功能說明：此函數可決定在加減速驅動中到達等速區段時的速度。若假設速度之設定值 V，則實際驅動速度如下列公式：

$$\text{驅動速度(PPS)} = \text{V} \times \text{倍率(M)}$$

參數 axis 為選擇之運動軸。

參數 data 為欲設定之速度值，資料範圍為 1~8000。

- void SetA(int axis, int data)

功能說明：此函數可決定在線性加減速驅動時的加速度，在 S-Curve 加減速驅動中的加速度，則從 0 至設定值為止呈線性的增加。若假設加速度之設定值為 A，則實際驅動加速度如下列公式：

$$\text{驅動加速度(PPS/Sec)} = \text{A} \times 125 \times \text{倍率(M)}$$

參數 axis 為選擇之運動軸。

參數 data 為欲設定之初速度值，資料範圍為 1~8000。

- void SetD(int axis, int data)

功能說明：此函數可決定在線性加減速驅動時的減速度，在 S-Curve 加減速驅動中的減速度，則從 0 至設定值為止呈線性的增加。若假設減速度之設定值為 D，則實際驅動減速度如下列公式：

$$\text{驅動減速度(PPS/Sec)} = \text{D} \times 125 \times \text{倍率(M)}$$

參數 axis 為選擇之運動軸。

參數 data 為欲設定之初速度值，資料範圍為 1~8000。



- void SetAA(int axis, int data)

功能說明：此函數可決定加速度變化率的設定值，是設定在 S-Curve 加減速中之加減速度的單位時間加減變化率的參數。若假設加速度變化率之設定值為 K，則實際驅動加速度變化率如下列公式：

$$\text{驅動加速度變化率(PPS/Sec}^2\text{)} = 62.5 \times 10^6 / K \times \text{倍率(M)}$$

參數 axis 為選擇之運動軸。

參數 data 為欲設定之初速度值，資料範圍為 1~65535。

- void Pulse(int axis, int data)

功能說明：此函數可設定在定量驅動時的輸出脈波總數。

參數 axis 為選擇之運動軸。

參數 data 為欲設定之初速度值，資料範圍為 0~268435455。

- void SetAO(int axis, int data)

功能說明：此函數可設定驅動脈波在減速段結束後到達初速度時的減速停止偏移值。當使用者所使用之機台馬達為步進馬達時，宜善用此函數，以讓速度曲線能先降至初始速度(低速)再停止，以避免讓機械平台由高速直接停止而產生過衝的現象。

參數 axis 為選擇之運動軸。

參數 data 為欲設定之初速度值，資料範圍為 0~65535。

- void AutoDec(void)

功能說明：此函數可設定全軸在加減速定量驅動之減速為自動減速。

註：圓弧補間於 T-Curve 加減速無法使用此函數。

- void ManualDec(int axis, int DecPulse)

功能說明：此函數可設定各軸在加減速定量驅動之減速為人工減速。

註：圓弧補間於 T-Curve 加減速需搭配使用此函數。

參數 axis 為選擇之運動軸。

參數 DecPulse 即為使用者設定的人工減速點。

- void TCurve\_F(int axis, int SV, int m\_V, int A, int Pulse)

功能說明：此函數可設定單軸或多軸以 T-Curve 加減速定量驅動，各運動軸會自動計算加減速段的時間，而消耗完輸入脈波。T-Curve 加減速之定量驅動通常用於固定距離的快速移動，只需設置相關運動參數，MCX314 會自動計算加減速時間和等速段的時間，適當地消耗完輸出脈波。



參數 axis 為選擇之運動軸。  
參數 SV 為設置之初速度。  
參數 V 為設置之速度。  
參數 A 為設置之加速度。  
參數 Pulse 為設置之終點，為相對座標。

- void TCurve\_C(int axis, int SV, int V, int A)

功能說明：此函數可設定單軸或多軸以 T-Curve 加減速連續驅動，各運動軸會自動計算加減速段的時間，而消耗完輸入脈波。T-Curve 加減速之連續驅動通常用於大範圍距離的快速移動，只需設置相關運動參數，MCX314 會自動計算加速時間，到達等速段後，會維持此定速移動，直至使用者下達停止命令或接受到外部驅動停止命令才會停止輸出驅動脈波。

參數 axis 為選擇之運動軸。  
參數 SV 為設置之初速度。  
參數 V 為設置之速度。  
參數 A 為設置之加速度。

- void SCurve\_F(int axis, int m\_SV, int m\_V, int m\_A, int, int m\_K, int m\_Pulse)

功能說明：此函數可設定單軸或多軸以 S-Curve 加減速定量驅動，各運動軸會自動計算加減速段的時間，而消耗完輸入脈波。S-Curve 加減速之定量驅動通常用於固定範圍距離的精密移動，只需設置相關運動參數，MCX314 會自動計算加減速時間和等速段的時間，適當地消耗完輸出脈波。

參數 axis 為選擇之運動軸。  
參數 SV 為設置之初速度。  
參數 V 為設置之速度。  
參數 A 為設置之加速度。  
參數 K 為設置之加速度變化率。

- void SCurve\_C(int axis, int m\_SV, int m\_V, int m\_A, int m\_K)

功能說明：此函數可設定各軸為 S-Curve 加減速連續驅動，各運動軸會加速至等速段，而依停止命令下達後才停止輸出脈波。S-Curve 加減速之連續驅動通常用於需要於加速段時的精密移動，只需設置相關運動參數，MCX314 會自動計算加速時間，到達等速段後，會維持此定速移動，直至使用者下達停止命令或接受到外部驅動停止命令才會停止輸出驅動脈波。

參數 axis 為選擇之運動軸。  
參數 SV 為設置之初速度。

參數 V 為設置之速度。  
 參數 A 為設置之加速度。

## 4.5 多軸補間運動命令函數

MCX314 在內部具有直線、圓弧和位元三個補間計算部。補間驅動是由指定主軸而依照其驅動脈波所產生的時序，而進行補間計算。補間訊號也和中斷控制部相連接，所以如果補間驅動的過程中，如遇到設定的硬體極限、軟體極限或極限開關，都能立即停止，以防止機械裝置損壞。圖 4-4 為 MCX314 的機能方塊圖。關於補間運動命令函數整理如表 4-4 所示。

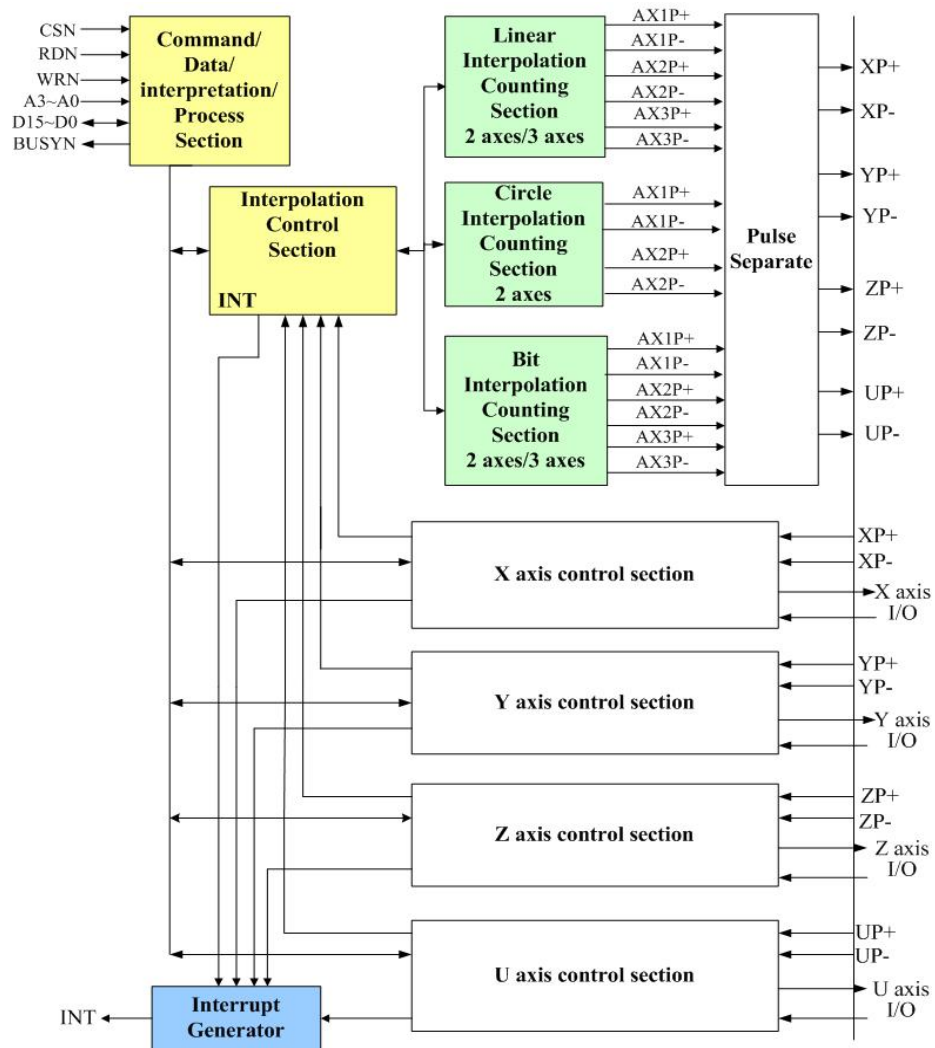


圖 4-4 MCX314 機能方塊圖[1]

表 4-4 補間運動命令函式

函式	功能
Line2D_C	2 軸直線等速補間
Line3D_C	3 軸直線等速補間
Line2D_T	2 軸直線補間以 T-Curve 加減速驅動
Line3D_T	3 軸直線補間以 T-Curve 加減速驅動
Line2D_S	2 軸直線補間以 S-Curve 加減速驅動
Line3D_S	3 軸直線補間以 S-Curve 加減速驅動
CircleCW_C	順時針方向圓弧補間以等速驅動
CircleCCW_C	逆時針方向圓弧補間以等速驅動
CircleCW_T	順時針方向圓弧補間以 T-Curve 加減速驅動
CircleCCW_T	逆時針方向圓弧以 T-Curve 加減速補間驅動
SetCP	設定圓弧補間時圓心座標
SetFP	設定補間運動時的各軸終點



#### 4.5.1 二軸三軸直線補間函數

直線補間函數的規劃，讓使用者可選擇任意二軸或三軸來進行直線補間，可以在各軸決定其相對座標的終點(即各軸輸出脈衝的大小值)，並使各軸補間運動為等速、T-Curve 加減速、S-Curve 加減速，隨後設置初始速度、速度、加速度等參數，最後呼叫函數即可進行直線補間運動。二軸三軸直線補間程式流程圖，如圖 4-5 所示。

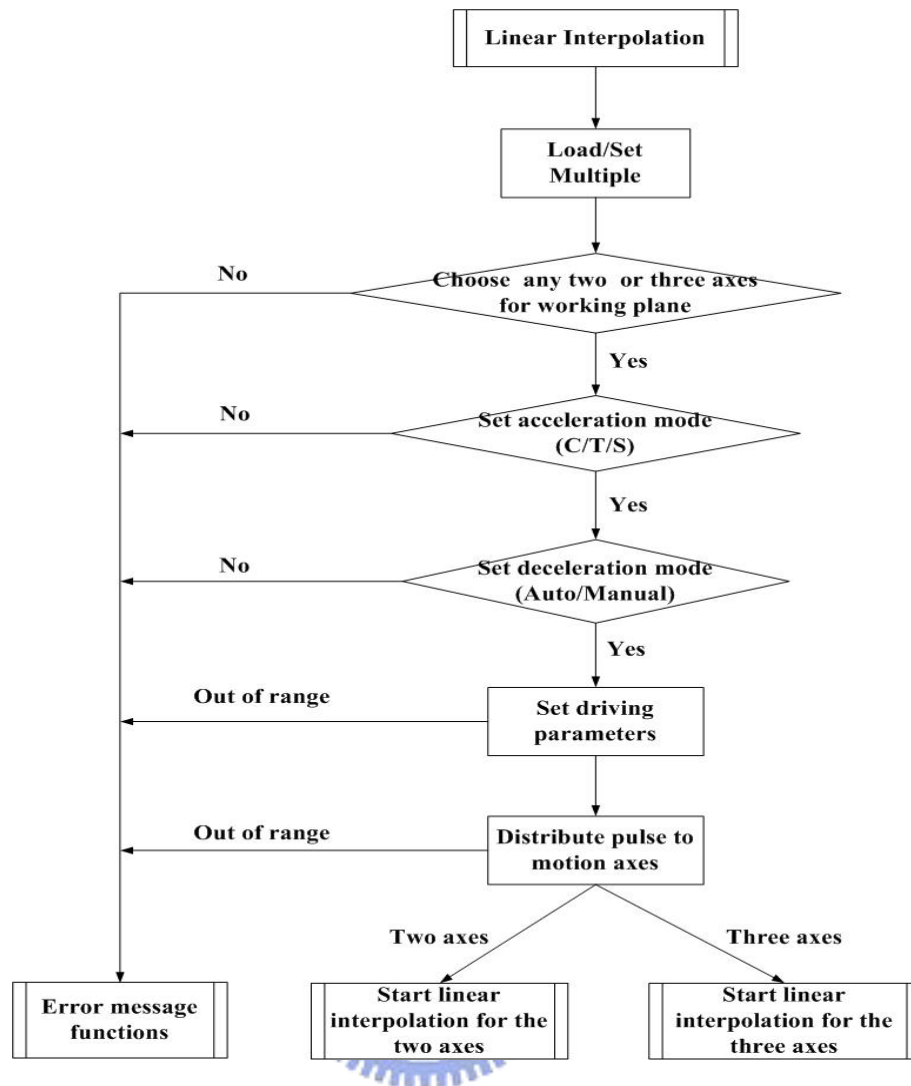


圖 4-5 二軸三軸直線補間程式流程圖

關於直線補間相關函式說明如下：

- void Line2D\_C(LINEDATA \*pLD, int axis)

功能說明：此函數為設置 2 軸直線補間以等速方式運動。

pLD 為結構 LINEDATA 的指標，可指向或儲存其結構內的參數，裡面的參數包括了初速度、速度、加速度等。

參數 axis 為選擇之運動軸，xy 軸為 0x3、xz 軸為 0x5、xu 軸為 0x9、yz 軸為 0x6、yu 軸為 0xa、zu 軸為 0xc。

- void Line3D\_C(LINEDATA \*pLD, int axis)

功能說明：此函數為設置 3 軸直線補間以等速方式運動。

pLD 為結構 LINEDATA 的指標，可指向或儲存其結構內的參數，裡面的參數包括了初速度、速度、加速度等。

參數 axis 為選擇之運動軸，xyz 軸為 0x7、xyu 軸為 0xb、xzu 軸為 0xd、yzu 軸為 0xe。

- void Line2D\_T(LINEDATA \*pLD, int axis)

功能說明：此函數為設置 2 軸直線補間以 T-Curve 方式加減速運動。

pLD 為結構 LINEDATA 的指標，可指向或儲存其結構內的參數，裡面的參數包括了初速度、速度、加速度等。

參數 axis 為選擇之運動軸，xy 軸為 0x3、xz 軸為 0x5、xu 軸為 0x9、yz 軸為 0x6、yu 軸為 0xa、zu 軸為 0xc。

- void Line3D\_T(LINEDATA \*pLD, int axis)

功能說明：此函數為設置 3 軸直線補間以 T-Curve 方式加減速運動。

pLD 為結構 LINEDATA 的指標，可指向或儲存其結構內的參數，裡面的參數包括了初速度、速度、加速度等。

參數 axis 為選擇之運動軸，xyz 軸為 0x7、xyu 軸為 0xb、xzu 軸為 0xd、yzu 軸為 0xe。

- void Line2D\_S(LINEDATA \*pLD, int axis)

功能說明：此函數為設置 2 軸直線補間以 S-Curve 方式加減速運動。

pLD 為結構 LINEDATA 的指標，可指向或儲存其結構內的參數，裡面的參數包括了初速度、速度、加速度等。

參數 axis 為選擇之運動軸，xy 軸為 0x3、xz 軸為 0x5、xu 軸為 0x9、yz 軸為 0x6、yu 軸為 0xa、zu 軸為 0xc。

- void Line3D\_S(LINEDATA \*pLD, int axis)

功能說明：此函數為設置 3 軸直線補間以 S-Curve 方式加減速運動。

pLD 為結構 LINEDATA 的指標，可指向或儲存其結構內的參數，裡面的參數包括了初速度、速度、加速度等。

參數 axis 為選擇之運動軸，xyz 軸為 0x7、xyu 軸為 0xb、xzu 軸為 0xd、yzu 軸為 0xe。

## 4.5.2 圓弧補間函數

圓弧補間函數的規劃，可以讓使用者選擇順時針方向或逆時針方向的補間運動，之後選擇補間軸，然後設置起點座標、終點座標、圓心座標，並使所選擇之二軸補間為等速運動或 T-Curve 加減速驅動，之後設置驅動速度，最後呼叫函數即可進行圓弧補間運動。圓弧補間程式流程圖如圖 4-6 所示。

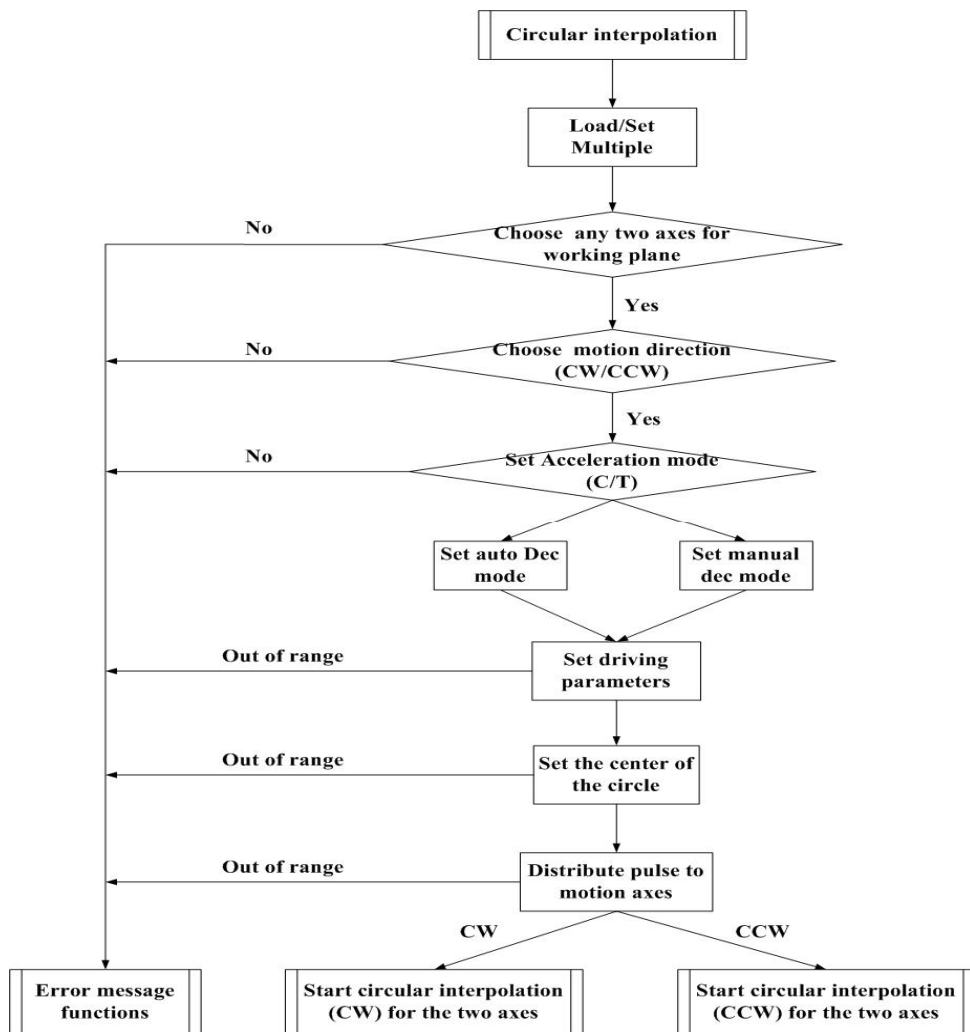


圖 4-6 圓弧補間程式流程圖

關於圓弧補間相關函式說明如下：

- void SetC(int axis, int data)

功能說明：此函數可設置圓弧補間運動時的圓心座標，圓心座標係為相對於起點(0, 0)之座標。



參數 axis 為選擇之運動軸，xy 軸為 0x3、xz 軸為 0x5、xu 軸為 0x9、yz 軸為 0x6、yu 軸為 0xa、zu 軸為 0xc。

- void SetP(int axis, int data)

功能說明：此函數可設置圓弧補間運動時的終點座標，終點座標需考慮和實際之運動半徑之間的關係，終點規劃不良將會使運動路徑不完整，或者無法驅動。

參數 axis 為選擇之運動軸，xy 軸為 0x3、xz 軸為 0x5、xu 軸為 0x9、yz 軸為 0x6、yu 軸為 0xa、zu 軸為 0xc。

- void CircleCW\_C(CIRCLEDATA \*pCD, int axis)

功能說明：此函數為設置任意 2 軸圓弧補間以等速順時針方向運動。

pCD 為結構 CIRCLEDATA 的指標，可指向或儲存其結構內的參數，裡面的參數包括了初速度、速度、加速度等。

參數 axis 為選擇之運動軸，xy 軸為 0x3、xz 軸為 0x5、xu 軸為 0x9、yz 軸為 0x6、yu 軸為 0xa、zu 軸為 0xc。

- void CircleCCW\_C(CIRCLEDATA \*pCD, int axis)

功能說明：此函數為設置任意 2 軸圓弧補間以等速逆時針方向運動。

pCD 為結構 CIRCLEDATA 的指標，可指向或儲存其結構內的參數，裡面的參數包括了初速度、速度、加速度等。

參數 axis 為選擇之運動軸，xy 軸為 0x3、xz 軸為 0x5、xu 軸為 0x9、yz 軸為 0x6、yu 軸為 0xa、zu 軸為 0xc。

- void CircleCW\_T(CIRCLEDATA \*pCD, int axis)

功能說明：此函數為設置任意 2 軸圓弧補間以 T-Curve 加減速，順時針方向運動。

pCD 為結構 CIRCLEDATA 的指標，可指向或儲存其結構內的參數，裡面的參數包括了初速度、速度、加速度等。

參數 axis 為選擇之運動軸，xy 軸為 0x3、xz 軸為 0x5、xu 軸為 0x9、yz 軸為 0x6、yu 軸為 0xa、zu 軸為 0xc。

- void CircleCCW\_T(CIRCLEDATA \*pCD, int axis)

功能說明：此函數為設置任意 2 軸圓弧補間以 T-Curve 加減速，逆時針方向運動。

pCD 為結構 CIRCLEDATA 的指標，可指向或儲存其結構內的參數，裡面的參數包括了初速度、速度、加速度等。

參數 axis 為選擇之運動軸，xy 軸為 0x3、xz 軸為 0x5、xu 軸為 0x9、yz 軸為 0x6、yu 軸為 0xa、zu 軸為 0xc。

## 4.6 I/O 信號管理函數

關於 I/O 信號管理函式整理如表 4-5 所示。

表 4-5 I/O 信號管理函式

函式	功能
HLMTP	設置硬體正方向極限訊號
HLMTM	設置硬體負方向極限訊號
HLMTSig	設置硬體極限停止訊號種類
SLMTP	設置軟體正方向極限訊號
SLMTM	設置軟體負方向極限訊號
SLMTSig	設置軟體極限停止訊號種類
EncoderMode	設置編碼器訊號模式
EncoderCWCCW	設置編碼器訊號模式
EnableALM	異常訊號致能
EnableInPos	定位完成訊號致能
ALMSig	選擇異常訊號為高/低位準為有效
InPosSig	選擇定位完成訊號為高/低位準為有效

關於 I/O 信號管理函數說明如下：

- void HLMTP(int n, int axis)

功能說明：此函數可設置硬體正方向極限訊號。

參數 n 為 0，代表設置為 Low 位準有效；參數 n 為 1，代表設置為 Hi 位準。

參數 axis 為選擇設置之運動軸，可多軸設置。

- void HLMTM(int n, int axis)

功能說明：此函數可設置硬體負方向極限訊號。

參數 n 為 0，代表設置為 Low 位準有效；參數 n 為 1，代表設置為 Hi 位準有效。

參數 axis 為選擇設置之運動軸，可多軸設置。

- void HLMTSig(int n, int axis)

功能說明：此函數可設置硬體極限停止訊號種類。

參數 n 為 0，代表設置為立即停止；參數 n 為 1，代表設置為減速停止。

參數 axis 為選擇設置之運動軸，可多軸設置。

- void SLMTP(int n)

功能說明：此函數可設置全軸之軟體正方向極限訊號。

其中參數 n 為 0，代表設置為 Low 位準有效；參數 n 為 1，代表設置為 Hi 位準有效。

- void SLMTM(int n)

功能說明：此函數可設置全軸之軟體正方向極限訊號。

參數 n 為 0，代表設置為 Low 位準有效；參數 n 為 1，代表設置為 Hi 位準有效。

- void SLMTSig(int n)

功能說明：此函數可設置全軸之 COMP+/-暫存器的比較對象是理論位置計數器(Logic Position Register)或者是實際位置計數器(Encoder Position Register)。

參數 n 為 0，代表設置為理論位置計數器；參數 n 為 1，代表設置為實際位置計數器。

- void EncoderMode(int n, int axis)

功能說明：此函數可設定編碼器輸入訊號為 2 相雙脈波輸入或單脈波輸入。編碼器輸入訊號會將實際位置計數器予以增加或減少計數。

參數 n 為 0，代表設置為 2 相雙脈波輸入；參數 n 為 1，代表設置為單脈波輸入。

參數 axis 為選擇設置之運動軸，可多軸設置。

- void EncoderCWCCW(int n, int axis)

功能說明：此函數可設定編碼器輸入訊號為 2 相雙脈波輸入之分頻比率。

參數 n 為 0，代表 2 相脈波輸入之分頻比率為 1/1；參數 n 為 1，代表 2 相脈波輸入之分頻比率為 1/2；參數 n 為 2，代表 2 相脈波輸入之分頻比率為 1/4。

參數 axis 為選擇設置之運動軸，可多軸設置。

- void EnableALM(int n)

功能說明：此函數使全軸之伺服馬達異常輸入訊號致能(Enable)或除能(Disable)。

參數 n 為 0，代表訊號除能；參數 n 為 1，代表訊號致能。

- void EnableInPos(int n)

功能說明：此函數使全軸之伺服馬達位置定位完畢輸入訊號致能(Enable)或除能(Disable)。

參數 n 為 0，代表訊號除能；參數 n 為 1，代表訊號致能。

- void ALMSig(int n, int axis)

功能說明：此函數設定伺服馬達異常輸入訊號的理論位準。

參數 n 為 0，代表 Low 位準為有效；參數 n 為 1，代表 Hi 位準為有效。

參數 axis 為選擇設置之運動軸，可多軸設置。

- void InPosSig(int n, int axis)

功能說明：此函數設定伺服馬達位置定位完畢輸入訊號的理論位準。

參數 n 為 0，代表 Low 位準為有效；參數 n 為 1，代表 Hi 位準為有效。

參數 axis 為選擇設置之運動軸，可多軸設置。

## 4.7 原點返回函數

承接 3-4 節，關於原點返回函式整理如表 4-6 所示。

表 4-6 原點返回函式

函式	功能
HomeType1	原點返回模式 1
HomeType2	原點返回模式 2
HomeType3	原點返回模式 3
HomeType4	原點返回模式 4

關於原點返回函式說明如下：

- void HomeType1(int WR1Reg, int SV, int V, int A)

運動模式：當移動至原點 ORG 的訊號位置時，即減速停止。

在此運動模式中主要應用於快速地回到原點，而不重視較精準的定位。原點返回運動模式一圖例說明如圖 3-28 所示。

參數 WR1Reg 為驅動停止輸入訊號的暫存器值，依照所選擇各軸 IN0~IN3 作為原點的訊號而有所不同。

參數 SV 為設置之初速度。

參數 V 為設置之速度。

參數 A 為設置之加速度。

- void HomeType2(int WR1Reg, int SV, int V, int A)

運動模式：當移動至原點 ORG 的訊號位置時，即減速停止，然以等速後退，再以慢速前進到達原點停止。

在此運動模式中可以利用第二次的慢速回到原點，而有較精確的定位效果。原點返回運動模式二圖例說明如圖 3-29 所示。

參數 WR1Reg 為驅動停止輸入訊號的暫存器值，依照所選擇各軸 IN0~IN3 作為原點的訊號而有所不同。

參數 SV 為設置之初速度。

參數 V 為設置之速度。

參數 A 為設置之加速度。

#### ● void HomeType3(int WR1Reg, int SV, int V, int A)

運動模式：當移動至原點 ORG 的訊號位置時，即減速至初始速度然後偵測至 EZ 信號(馬達編碼器 Z 相訊號)後停止。

在此運動模式中主要應用於快速地回到原點，並利用馬達編碼器 Z 相訊號為停止命令，不重視較精準的定位。原點返回運動模式三圖例說明如圖 3-30 所示。

參數 WR1Reg 為驅動停止輸入訊號的暫存器值，依照所選擇各軸 IN0~IN3 作為原點的訊號而有所不同。

參數 SV 為設置之初速度。

參數 V 為設置之速度。

參數 A 為設置之加速度。



#### ● void HomeType4(int WR1Reg, int SV, int V, int A)

運動模式：當移動至原點 ORG 的訊號位置時，即減速停止，待偵測至 EZ 信號(馬達編碼器 Z 相訊號)後停止，以等速後退，再以慢速前進到達原點停止。

在此運動模式中可以利用第二次的慢速回到原點，並利用馬達編碼器 Z 相訊號為停止命令，而有較精確的定位效果。原點返回運動模式四圖例說明如圖 3-31 所示。

參數 WR1Reg 為驅動停止輸入訊號的暫存器值，依照所選擇各軸 IN0~IN3 作為原點的訊號而有所不同。

參數 SV 為設置之初速度。

參數 V 為設置之速度。

參數 A 為設置之加速度。

## 4.8 位置暫存器管理函數

X、Y、Z、U 各軸分別擁有兩個用管理現在位置的 32 位元加減計數器和兩個用來比較現在位置大小的比較暫存器。理論計數器(Logic Position Counter, LP)會再內部計算其正負方向的輸出脈波數，當正向脈波數輸出時計數器會增加 1；反之，負向時則會減少 1。實際位置計數器(Encoder Position Counter, EP)則負責計算從編碼器或其他等外部的輸入脈波數理論和實際計數器，皆可由使用者將資料寫入或讀取。而比較暫存器 COMP+和 COMP-則主要應用於正負方向的軟體極限。軟體極限可以彈性的設置，可利用其命令暫存器來決定是否使用。若正向驅動中理論或實際暫存器的值大於 COMP+時，即減速停止；同理，若反向驅動中理論或實際暫存器的值大於 COMP-時，即減速停止。如圖 4-7 為位置暫存器與軟體極限管理流程[1]。關於位置暫存器管理函數整理如表 4-7 所示。

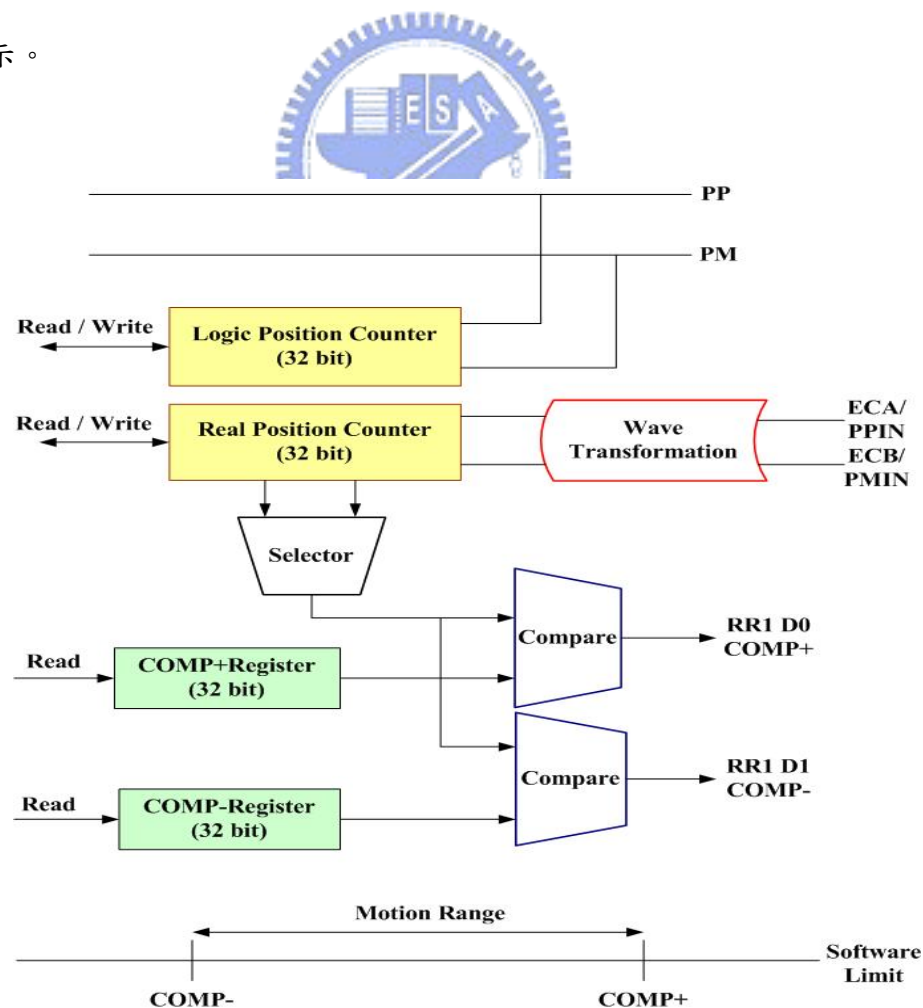


圖 4-7 位置暫存器與軟體極限管理流程[1]



表 4-7 位置暫存器管理函數

函式	功能
LPS	設置邏輯位置狀態暫存器
EPS	設置編碼器位置狀態暫存器
ReadLPS	讀取邏輯位置狀態暫存器
ReadEPS	讀取編碼器位置狀態暫存器
ReadCV	讀取目前速度
ReadCA	讀取目前加速度
SetCompP	設定正方向軟體極限
SetCompM	設定負方向軟體極限
RESET	清理重置暫存器

關於位置暫存器管理函數說明如下：

- void LPS(int axis, unsigned data)

功能說明：此函數可設置各軸理論位置計數器的值，可設置為 0，用以清除計數器的值。

參數 axis 為選擇設置之運動軸，為各軸分別設置。

參數 data 為任意輸入至計數器的值，資料範圍為-2147483648~+2147483648。

- void EPS(int axis, unsigned data)

功能說明：此函數可設置個軸實際位置計數器的值，可設置為 0，用以清除計數器的值。

參數 axis 為選擇設置之運動軸，為各軸分別設置。

參數 data 為任意輸入至計數器的值，資料範圍為-2147483648~+2147483648。

- unsigned ReadLPS(int axis)

功能說明：此函數可讀取各軸理論位置計數器的數值(即 MCX314 所規劃的位置命令)，在視窗程式中配合 Timer 使用可即時顯示計數器的數值。視窗程式內定為 0.01 秒為一個取樣時間。

參數 axis 為選擇設置之運動軸，為各軸分別讀取。

- unsigned ReadEPS(int axis)

功能說明：此函數可讀取各軸實際位置計數器的數值(即由馬達編碼器所讀取的位置響應)，在視窗程式中配合 Timer 使用可即時顯示計數器的數值。視窗程式內定為 0.01 秒為一個取樣時間。

參數 axis 為選擇設置之運動軸，為各軸分別讀取。

- unsigned ReadCV(int axis)

功能說明：此函數可讀取現在驅動速度的值，驅動停止時則為 0。視窗程式內定為 0.01 秒為一個取樣時間。

參數 axis 為選擇設置之運動軸，為各軸分別讀取。

- unsigned ReadCA(int axis)

功能說明：此函數可讀取現在驅動加速度、減速度的值，驅動停止或等速時則為一個 default 值或是最後一次所讀取的加速度值。視窗程式內定為 0.01 秒為一個取樣時間。

參數 axis 為選擇設置之運動軸，為各軸分別讀取。

