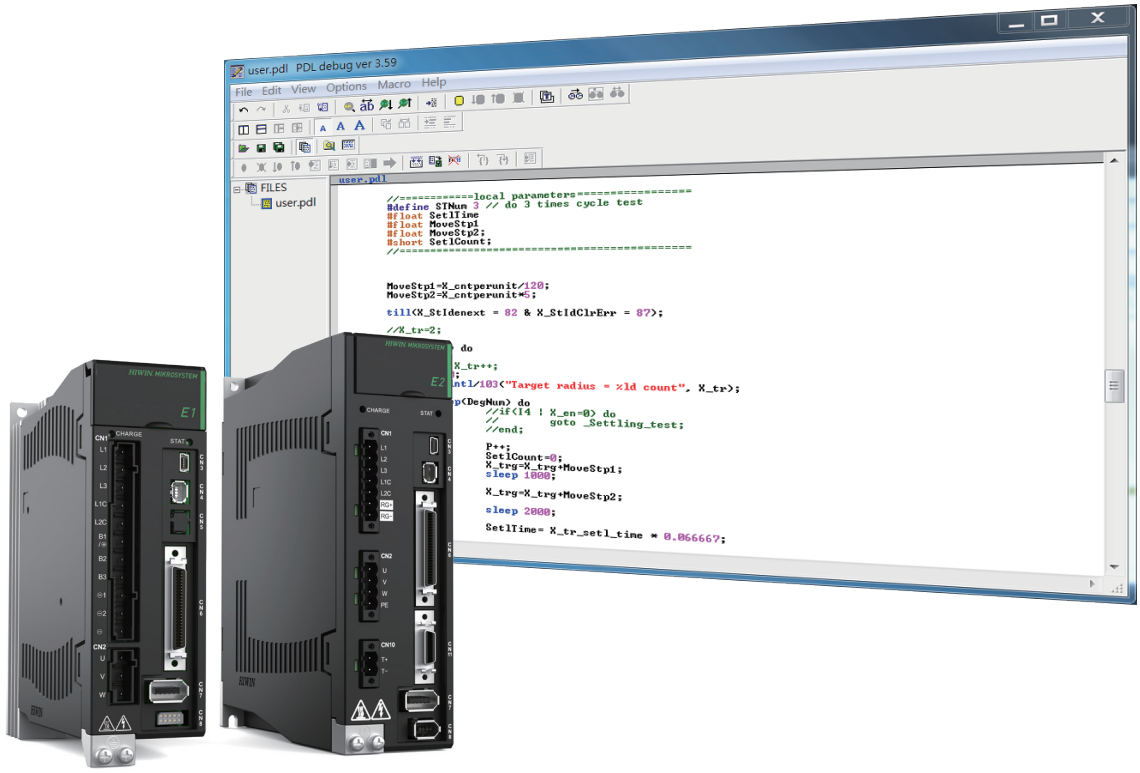
HIWIN MIKROSYSTEM



E 系列驅動器

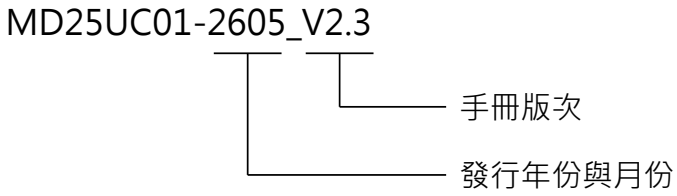
PDL 範例程式

www.hiwinmikro.tw

MD25UC01-2605_V2.3

修訂紀錄

手冊版次資訊亦標記於手冊封面右下角。



發行日期	版次	適用產品	更新內容
2026/05/25	2.3	E1 系列驅動器 E2 系列驅動器	1. 更新第一章所有**基本介紹**章節 2. 新增第二章**內建指令觸發功能** 3. 更新第三章**常用變數**
2024/03/29	2.2	E1 系列驅動器 E2 系列驅動器	更新 2.3 節**物理量監控變數 (讀)**。
2023/04/25	2.1	E1 系列驅動器 E2 系列驅動器	1. 更新手冊名稱。 2. 更新適用驅動器與軟韌體版次需求。
2021/01/06	2.0	E1 系列驅動器	1. 新增第 2 章**常用變數**。 2. 因應新增的常用變數，優化 PDL 與修改應用說明。
2020/12/02	1.1	E1 系列驅動器	優化 PDL 與修改應用說明。
2020/02/20	1.0	E1 系列驅動器	初版發行。

相關文件

透過相關文件，使用者可快速了解此手冊的定位，以及各手冊、產品之間的關聯性。詳細內容請至本公司官網→下載中心→手冊總覽閱覽（https://www.hiwinmikro.tw/Downloads/ManualOverview_TC.htm）。

目錄

1. 範例程式 1-1

1.1 直接搜尋 Index 訊號歸原點 1-3

1.1.1 基本介紹 1-3

1.1.2 程式內容 1-3

1.1.3 使用說明 1-5

1.1.4 應用說明 1-6

1.2 碰撞 Hard Stop 歸原點 1-7

1.2.1 基本介紹 1-7

1.2.2 程式內容 1-7

1.2.3 使用說明 1-10

1.2.4 應用說明 1-12

1.3 I/O 觸發馬達移動固定距離 1-13

1.3.1 基本介紹 1-13

1.3.2 程式內容 1-13

1.3.3 使用說明 1-15

1.3.4 應用說明 1-15

1.4 單位轉換 1-16

1.4.1 基本介紹 1-16

1.4.2 程式內容 1-16

1.4.3 使用說明 1-17

1.4.4 應用說明 1-18

1.5 教導功能 1-19

1.5.1 基本介紹 1-19

1.5.2 程式內容 1-19

1.5.3 使用說明 1-22

1.6 重新激磁延續未完成的絕對座標運動 1-23

1.6.1 基本介紹 1-23

1.6.2 程式內容 1-23

1.6.3 使用說明 1-25

1.6.4 應用說明 1-25

1.7 吋動 1-26

1.7.1 基本介紹 1-26

1.7.2 程式內容 1-26

1.7.3 使用說明 1-28

1.7.4 應用說明 1-28

1.8 類比輸入控制吋動 1-29

1.8.1 基本介紹 1-29

1.8.2 程式內容 1-29
1.8.3 使用說明 1-31
1.8.4 應用說明 1-32
2. 內建指令功能 2-33
2.1 雷射量測誤差補償表建立 2-34
2.1.1 基本介紹 2-34
2.1.2 使用說明 2-35
2.1.3 功能執行流程 2-37
2.2 龍門旋轉軸誤差補償表建立 2-38
2.2.1 基本介紹 2-38
2.2.2 使用說明 2-39
2.2.3 功能執行流程 2-42
3. 常用變數 3-1
3.1 運動控制變數 (讀 / 寫) 3-2
3.2 運動狀態變數 (讀) 3-8
3.3 物理量監控變數 (讀) 3-10
3.4 伺服訊號狀態變數 (讀) 3-14
3.5 伺服訊號 IO 變數 (讀) 3-16
3.6 總線通訊自定義物件變數 (讀/寫) 3-17

1. 範例程式

1. 範例程式 1-1
   1.1 直接搜尋 Index 訊號歸原點 1-3
      1.1.1 基本介紹 1-3
      1.1.2 程式內容 1-3
      1.1.3 使用說明 1-5
      1.1.4 應用說明 1-6
   1.2 碰撞 Hard Stop 歸原點 1-7
      1.2.1 基本介紹 1-7
      1.2.2 程式內容 1-7
      1.2.3 使用說明 1-10
      1.2.4 應用說明 1-12
   1.3 I/O 觸發馬達移動固定距離 1-13
      1.3.1 基本介紹 1-13
      1.3.2 程式內容 1-13
      1.3.3 使用說明 1-15
      1.3.4 應用說明 1-15
   1.4 單位轉換 1-16
      1.4.1 基本介紹 1-16
      1.4.2 程式內容 1-16
      1.4.3 使用說明 1-17
      1.4.4 應用說明 1-18
   1.5 教導功能 1-19
      1.5.1 基本介紹 1-19
      1.5.2 程式內容 1-19
      1.5.3 使用說明 1-22
   1.6 重新激磁延續未完成的絕對座標運動 1-23
      1.6.1 基本介紹 1-23
      1.6.2 程式內容 1-23
      1.6.3 使用說明 1-25
      1.6.4 應用說明 1-25
   1.7 吋動 1-26
      1.7.1 基本介紹 1-26
      1.7.2 程式內容 1-26
      1.7.3 使用說明 1-28
      1.7.4 應用說明 1-28

1.8 類比輸入控制吋動 1-29
1.8.1 基本介紹 1-29
1.8.2 程式內容 1-29
1.8.3 使用說明 1-31
1.8.4 應用說明 1-32

本章提供使用者 E 系列驅動器 PDL 範例程式應用，包含歸原點、相對移動、吋動、單位轉換等。
其相關控制、物理量、伺服變數請參閱第 3 章、《E1 系列驅動器使用者操作手冊》與《E2 系列驅動器使用者操作手冊》。

1.1 直接搜尋 Index 訊號歸原點

1.1.1 基本介紹

程式編號	1
檔名	1.pdl
適用驅動器	E 系列驅動器
適用馬達	AC 伺服馬達、線性馬達、直驅馬達、力矩馬達
軟韌體版次需求	E1 系列驅動器：Thunder 1.6.$.$　MDP 2.6.$（含）以上 E2 系列驅動器：Thunder 1.9.$.$　MDP 3.9.$（含）以上 E2-R 系列驅動器：Thunder 1.14.$.$　MDP 4.14.$（含）以上
適用模式	內部位置模式
功能	觸發 I2 數位輸入端，執行驅動器內建歸原點功能。且在歸原點成功後，由 O2 數位輸出端輸出一個脈波給上位控制器。
使用輸入	I2
使用輸出	O2

註：
請先至 Thunder 主畫面選擇書籤列中的**工具**、點選 **IO 設定**，將 I2 與 O2 設為 **Not configure**，以避免功能相衝突。

1.1.2 程式內容

```
// ============================ 巨　集 ============================
//  使用者可在此巨集內設定歸原點速度與輸出脈波寬度。
#define SEARCH_DIR   34                    //  往正方向搜尋 index
//  #define SEARCH_DIR   33                //  往負方向搜尋 index
#define R_SPEED_SEARCH_FAST   30           //  旋轉搜尋近原點開關速度 (單位：rpm)
#define R_SPEED_SEARCH_SLOW   10           //  旋轉搜尋原點速度 (單位：rpm)
#define L_SPEED_SEARCH_FAST   20           //  線性搜尋近原點開關速度 (單位：mm/s)
#define L_SPEED_SEARCH_SLOW   10           //  線性搜尋原點速度 (單位：mm/s)
#define HOMEING_TIMER   30                 //  歸原點程序的時間限制 (單位：秒)
#define OUTPUT_WIDTH   2000                //  輸出脈波寬度 (單位：毫秒)
```

```
proc pdl_en() do                                    //  激磁馬達
    mEnable = 1;
    till (S_RDY);
end;
proc pdl_dis() do                                   //  解激磁馬達
    mEnable = 0;
    till (~S_RDY);
end;
proc home_para() do                                 //  設定歸原點參數
    Pt700 = SEARCH_DIR;                             //  歸原點方法
    Pt701 = R_SPEED_SEARCH_FAST;
    Pt702 = R_SPEED_SEARCH_SLOW;
    Pt703 = HOMEING_TIMER;
    Pt705 = L_SPEED_SEARCH_FAST;
    Pt706 = L_SPEED_SEARCH_SLOW;
end;
proc homing() do
    home_para();
    pdl_en();
    till (S_RDY);
    mHome = 1;                                      //  呼叫驅動器內建歸原點命令
    sleep 100;                                      //  等待 100 毫秒
    till (dHomeState = 3);                          //  確認歸原點程序 (正在執行中)
    till (dHomeState = 2 | dHomeState = -1);        //  確認歸原點程序 (成功或失敗)
    if (dHomeState <> 2) do                         //  確認歸原點狀態是否成功
        printl/101("Homing process failed.");
        pdl_dis();
    end;
    sleep 1000;
end;
// ============================ 主程序 ============================
#task/1;                                            //  建立一編號為 1 的 task
till (~I2);                                         //  等待 I2 關閉
setoff O2;                                          //  關閉 O2
// ========================== 歸原點程序 ==========================
till (I2);                                          //  等待 I2 觸發
```

```
homing();                                   //  呼叫歸原點程式
//  歸原點成功後，輸出脈波給上位控制器。
if (dHomeState = 2) do                      //  判斷歸原點狀態是否成功
    seton O2;                               //  觸發 O2
    sleep OUTPUT_WIDTH;                     //  等待 2 秒
    setoff O2;                              //  關閉 O2
    sleep 1000;                             //  等待 1 秒
    pdl_dis();
end;
ret;
```

1.1.3 使用說明

本範例程式藉由觸發 I2 數位輸入端，執行驅動器內建的歸原點功能。且在歸原點程序完成後，由 O2 數位輸出端輸出一個脈波給上位控制器。程式流程圖如圖 1.1.3.1。

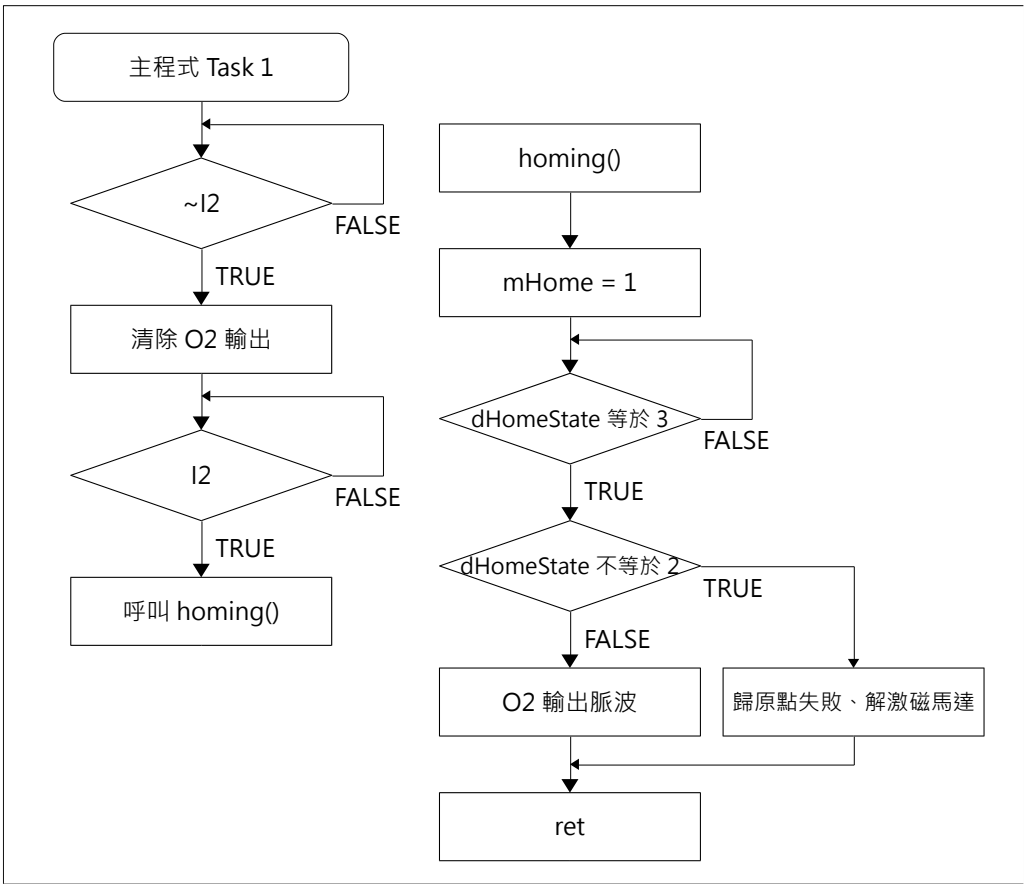


圖 1.1.3.1 程式流程圖

1.1.4 應用說明

進入程式主程序前，宣告一 task：#task/1。主要用途為指定使用者所編輯的 PDL 應用程式，在驅動器多工（Multitasking）功能程序中，所佔用的任務編碼。E 系列驅動器提供使用者自由編輯 task 編碼 1~3。

在驅動器作業系統中，含有一功能命令 mHome，此命令為驅動器內建歸原點功能。在程式中撰寫 mHome = 1 時，會開始執行歸原點動作，其歸原點方式為執行方法 34『直接搜尋 Index 訊號歸原點』。相關參數設定位於 Thunder 人機介面的**歸原點**視窗，如圖 1.1.4.1 所示。

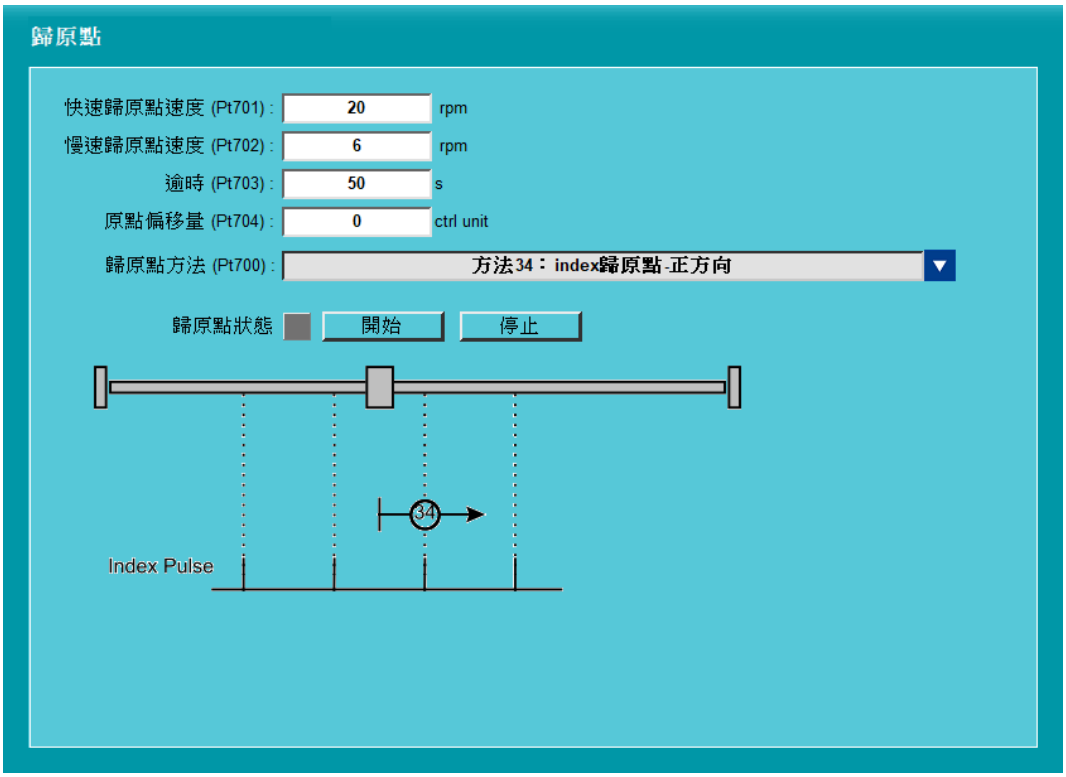


圖 1.1.4.1 歸原點功能相關參數設定

在本範例程式中，歸原點程序亦使用到一變數 dHomeState，此變數表示目前歸原點狀態，如表 1.1.4.1 所示。

表 1.1.4.1 dHomeState 變數與歸原點狀態對照表

dHomeState	歸原點狀態
-1	歸原點失敗
0	尚未執行歸原點
3	正在執行歸原點
4	歸原點暫停
2	歸原點成功

1.2 碰撞 Hard Stop 歸原點

1.2.1 基本介紹

程式編號	2
檔名	2.pdl
適用驅動器	E 系列驅動器
適用馬達	AC 伺服馬達（搭配螺桿）、線性馬達、直驅馬達、力矩馬達
軟韌體版次需求	E1 系列驅動器：Thunder 1.6.$.$　MDP 2.6.$（含）以上 E2 系列驅動器：Thunder 1.9.$.$　MDP 3.9.$（含）以上 E2-R 系列驅動器：Thunder 1.14.$.$　MDP 4.14.$（含）以上
適用模式	內部位置模式
功能	馬達先往負方向移動，碰撞 Hard Stop 且輸出電流達到設定門檻值時，馬達會開始往正方向搜尋 index，完成歸原點動作。
使用輸入	I2
使用輸出	無

註：

請先至 Thunder 主畫面選擇書籤列中的**工具**、點選 **IO 設定**，將 I2 設為 **Not configure**，以避免功能相衝突。

1.2.2 程式內容

```
// ============================ 巨　集 ============================
//  使用者可選擇歸原點開始方向，並依照馬達種類設定速度參數。
#define SEARCH_DIR   1                          //  往負方向開始
//  #define SEARCH_DIR   2                      //  往正方向開始
#define BackDistance   10000                    //  遠離 Hard Stop 的距離 (單位 : control unit)
#define P_Constant_Vel   1                      //  正方向等速度運動
#define N_Constant_Vel   -1                     //  負方向等速度運動
#define R_SPEED_SEARCH_FAST   30                //  旋轉搜尋近原點開關速度 (單位：rpm)
#define R_SPEED_SEARCH_SLOW   10                //  旋轉搜尋原點速度 (單位：rpm)
#define L_SPEED_SEARCH_FAST   20                //  線性搜尋近原點開關速度 (單位：mm/s)
#define L_SPEED_SEARCH_SLOW   10                //  線性搜尋原點速度 (單位：mm/s)
#define HOMEING_TIMER   30                      //  歸原點程序的時間限制 (單位：秒)
#define CONTINUOUS_CURRENT   3.5                //  連續電流 (單位：A-amp)
#long Start_dir;
```

```
proc pdl_en() do                                    //  激磁馬達
    mEnable = 1;
    till (S_RDY);
end;
proc pdl_dis() do                                   //  解激磁馬達
    mEnable = 0;
    till (~S_RDY);
end;
proc home_para() do                                 //  設定歸原點參數
    if (Start_dir = 1) do
        Pt700 = 34;                                 //  歸原點方法
    else do
        Pt700 = 33;                                 //  歸原點方法
    end;
    Pt701 = R_SPEED_SEARCH_FAST;
    Pt702 = R_SPEED_SEARCH_SLOW;
    Pt703 = HOMEING_TIMER;
    Pt705 = L_SPEED_SEARCH_FAST;
    Pt706 = L_SPEED_SEARCH_SLOW;
end;
proc homing() do
    home_para();
    pdl_en();
    till (S_RDY);
    mHome = 1;                                      //  呼叫驅動器內建歸原點命令
    sleep 100;                                      //  等待 100 毫秒
    till (dHomeState = 3);                          //  確認歸原點程序 (正在執行中)
    till (dHomeState = 2 | dHomeState = -1);        //  確認歸原點程序 (成功或失敗)
    if (dHomeState <> 2) do                         //  確認歸原點狀態是否成功
        printl/101("Homing process failed.");
        pdl_dis();
    end;
    sleep 1000;
end;
// ========================== 主程序 ==========================
#task/1;                                            //  建立一編號為 1 的 task
```

```
#float CurrThreshold;
mTestRun = 1;                                    //  強制切換至內部位置試運轉模式
Start_dir = SEARCH_DIR;
CurrThreshold = CONTINUOUS_CURRENT*1.5;          //  設定電流門檻值 (單位：A-amp)
CurrThreshold*= CurrThreshold;
Pt533 = 50;                                      //  旋轉搜尋 Hard Stop 的速度 (單位：rpm)
Pt585 = 50;                                      //  線性搜尋 Hard Stop 的速度 (單位：mm/s)
Pt534 = 100;                                     //  軟啟動加速時間 (單位：ms)
Pt537 = 100;                                     //  軟啟動減速時間 (單位：ms)
till (~I2);                                      //  等待 I2 關閉
till (I2);                                       //  等待 I2 觸發
pdl_en();
till (S_RDY);
printl/103("Start to search hard stop.");
if (Start_dir = 1) do
    mRun = P_Constant_Vel;                       //  往正方向碰撞 Hard Stop
else do
    mRun = N_Constant_Vel;                       //  往負方向碰撞 Hard Stop
end;
till (dFbCurr > CurrThreshold | ALM);            //  等待碰撞後馬達電流超越門檻值
                                                 //  dFbCurr 為馬達電流 (單位：A-amp^2)
mStopMove = 1;                                   //  運動停止
if (ALM) do
    printl/101("Search hard stop failed!");
    goto _end;
end;
till (~X_run);
printl/103("Hard stop found, going back.");
sleep 100;
if (Start_dir = 1) do
    mMovePos += BackDistance;                    //  離開 Hard Stop
else do
    mMovePos -= BackDistance;                    //  離開 Hard Stop
end;
sleep 100;
till (~X_run);
```

```
printl/103("Start to search index.");
homing();
printl/103("Homing process succeeded.");
_end:
pdl_dis();
ret;
```

1.2.3 使用說明

在歸原點程序中，為達到每次搜尋 index 訊號起始基準點與方向的一致性，使用者可選擇以「觸發 Limit Switch」或「碰撞 Hard Stop」的方式定義起始基準點。在沒有 Limit Switch 裝置的情況下，使用者可選擇以「碰撞 Hard Stop」的方式定義起始基準點。

碰撞 Hard Stop 的主要作動方式為碰撞固定方向的 Hard Stop，再往反方向搜尋 index 訊號，以獲得良好的原點重現性。其歸原點程序流程圖如圖 1.2.3.1。

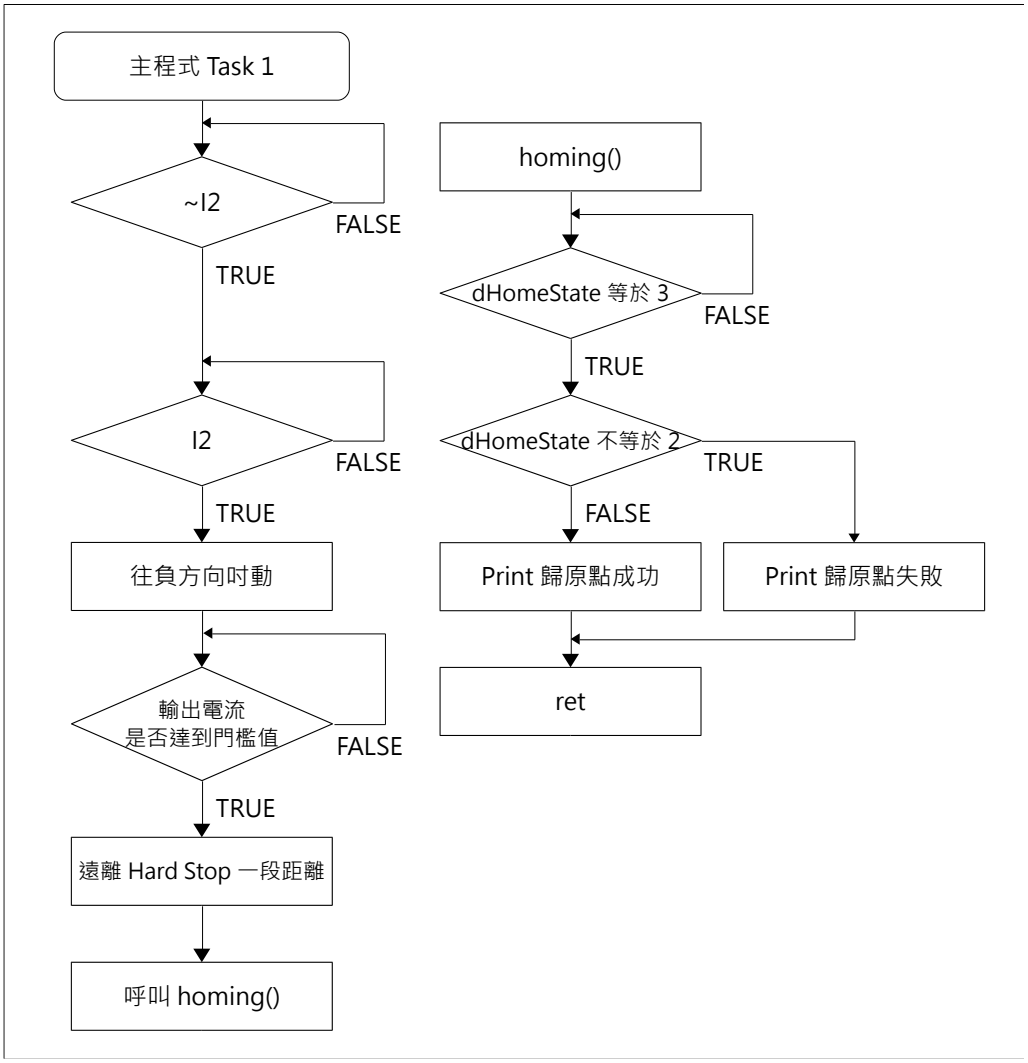

主程式 Task 1
~I2
FALSE
TRUE
I2
FALSE
TRUE
往負方向吋動
輸出電流
是否達到門檻值
FALSE
TRUE
遠離 Hard Stop 一段距離
呼叫 homing()
homing()
dHomeState 等於 3
FALSE
TRUE
dHomeState 不等於 2
TRUE
FALSE
Print 歸原點成功
Print 歸原點失敗
ret

圖 1.2.3.1 程序流程圖

在程式的最初，使用者須設定連續電流值 (CONTINUOUS_CURRENT) 與電流門檻值 (CurrThreshold)。設定流程如下：

Step 1. 開啟 Thunder 人機介面的**設定精靈**視窗。
Step 2. 於**馬達設定**頁籤查看『連續電流』的值 (單位：A-rms)。
Step 3. 將單位 A-rms 轉換成 A-amp (公式轉換：A-rms*1.414 = A-amp)，設定為連續電流值。
Step 4. 設定電流門檻值，稍微大於連續電流值即可。建議為連續電流值的 1.5 倍。
註：電流門檻值不可設定太大，否則在碰撞 Hard Stop 時，容易產生『過電流檢出』的警報。

1.2.4 應用說明

若使用者想使用「觸發 Limit Switch，再執行搜尋 Index」的方式，可直接至 Thunder 人機介面的**歸原點**視窗選擇歸原點方法 1。

1.3 I/O 觸發馬達移動固定距離

1.3.1 基本介紹

程式編號	3
檔名	3.pdl
適用驅動器	E 系列驅動器
適用馬達	AC 伺服馬達、線性馬達、直驅馬達、力矩馬達
軟韌體版次需求	E1 系列驅動器：Thunder 1.6.$.$　MDP 2.6.$（含）以上 E2 系列驅動器：Thunder 1.9.$.$　MDP 3.9.$（含）以上 E2-R 系列驅動器：Thunder 1.14.$.$　MDP 4.14.$（含）以上
適用模式	內部位置模式
功能	觸發 I2 與 I3 數位輸入端使馬達移動一固定距離；其中 I2 為正向運動，I3 為反向運動。
使用輸入	I2、I3
使用輸出	無

註：

(1) 若使用線性馬達，須注意行程極限。

(2) 請先至 Thunder 主畫面選擇書籤列中的**工具**、點選 **IO 設定**，將 I2 與 I3 設為 **Not configure**，以避免功能相衝突。

1.3.2 程式內容

```
// ============================ 巨　集 ============================
//  馬達移動速度取決於 Pt533 或 Pt585 的設定。
#define Distance   1000                    //  設定欲移動的距離 (單位：control unit)
proc pdl_en() do                           //  激磁馬達
    mEnable = 1;
    till (S_RDY);
end;
proc pdl_dis() do                          //  解激磁馬達
    mEnable = 0;
    till (~S_RDY);
end;
// ============================ 主程序 ============================
#long io_switch, io;
```

```
#task/1;                                    //  建立一編號為 1 的 task
mTestRun = 1;                               //  強制切換至內部位置試運轉模式
io_loop:
if (I2& ~I3) do                             //  判斷 I2 為 ON 且 I3 為 OFF
    io=1;
    io_switch=1;
    call _trigger_loop;
end;
if (I3 & ~I2) do                            //  判斷 I2 為 OFF 且 I3 為 ON
    io=2;
    io_switch=1;
    call _trigger_loop;
end;
goto io_loop;
ret;

_trigger_loop:
if (io_switch=1) do
    pdl_en();
    till (S_RDY);
    if (io=1) do                            //  判斷 I2 為 ON 且 I3 為 OFF，正向運動
        mMovePos += Distance;               //  正向運動並移動設定好的距離
        sleep 50;
        till (COIN);                        //  等待馬達停止運動
        printl/103("Positive motion ended.");
        sleep 1000;
    end;
    if (io=2) do                            //  判斷 I2 為 OFF 且 I3 為 ON，反向運動
        mMovePos -= Distance;               //  反向運動並移動設定好的距離
        sleep 50;
        till (COIN);                        //  等待馬達停止運動
        printl/103("Negative motion ended.");
        sleep 1000;
    end;
end;
io_switch=0;
```

```
io=0;
pdl_dis();
till (~I2 & ~I3);                    //  確定 I2 與 I3 恢復初始狀態
ret;
```

1.3.3 使用說明

使用者可先設定馬達移動的距離（Distance），再觸發 I2 或 I3 數位輸入端，使馬達往正或負方向移動。本範例程式中，重新執行_trigger_loop 前，程式都會先判斷 I2 與 I3 的狀態。其輸入端狀態與移動方向之對照如表 1.3.3.1 所示。

表 1.3.3.1 輸入端狀態與移動方向對照表

I2	I3	移動方向
ON	OFF	正
OFF	ON	負

1.3.4 應用說明

PDL 程式中，mMovePos 為執行絕對座標運動的命令。當 mMovePos 與當下編碼器的回授值不同時，馬達會移動到 mMovePos 的位置。此命令亦可接受運動過程中變更數值，也就是說，最終馬達會移動到最後一次修改 mMovePos 數值的位置上。

因此，使用 PDL 撰寫相對座標運動時，參考位置務必使用 Target position（目標位置），寫法如下：

```
mMovePos += Distance;
```

另外，可藉 printl/retprintl 將訊息顯示於 Message+command prompt 視窗中。

1.4 單位轉換

1.4.1 基本介紹

程式編號	4
檔名	4.pdl
適用驅動器	E 系列驅動器
適用馬達	AC 伺服馬達、線性馬達、直驅馬達、力矩馬達
軟韌體版次需求	E1 系列驅動器：Thunder 1.6.$.$　MDP 2.6.$（含）以上 E2 系列驅動器：Thunder 1.9.$.$　MDP 3.9.$（含）以上 E2-R 系列驅動器：Thunder 1.14.$.$　MDP 4.14.$（含）以上
適用模式	內部位置模式
功能	將單位由 mm 或 deg 轉換成 control unit，使後接之程式碼，不須由軟體開發工程師自行計算單位轉換。
使用輸入	無
使用輸出	無

註：
使用 AC 伺服馬達搭配滾珠螺桿時，須先設定外部編碼器進給長度 (Pt20A)。

1.4.2 程式內容

```
// ============================ 巨　集 ============================
#define Distance   45                                    //  設定移動的距離（mm 或 deg）
// ============================ 主程序 ============================
// =   敘述：使用線性馬達時由 mm 轉換成 control unit
// =           使用 AC 伺服馬達搭配滾珠螺桿時由 mm 轉換成 control unit（須先設定 Pt20A）
// =           使用直驅馬達 / 力矩馬達則由 deg 轉換成 control unit
// =   引數：無
// ================================================================
#task/1;                                                 //  建立一編號為 1 的 task
#float LMCntPerMM;
#float DisTemp;
#long mm2rev                                             //  Pt20A 外部編碼器進給長度
mm2rev= Pt20A/1000;                                      //  um -> mm
```

```
if (dMotorType=2) do                          //  馬達為 AC 伺服馬達
    DisTemp = Distance;
    DisTemp /= mm2rev;                        //  mm -> rev
    DisTemp *= dCntPerUnit;                   //  rev -> count
    DisTemp *= Pt210;                         //  電子齒輪比分母
    DisTemp /= Pt20E;                         //  電子齒輪比分子  count -> control unit
end;

if (dMotorType=1) do                          //  馬達為直驅馬達 / 力矩馬達
    DisTemp = Distance;
    DisTemp /= 360;                           //  deg -> rev
    DisTemp *= dCntPerUnit;                   //  rev -> count
    DisTemp *= Pt210;                         //  電子齒輪比分母
    DisTemp /= Pt20E;                         //  電子齒輪比分子  count -> control unit
end;

if (dMotorType=0) do                          //  馬達為線性馬達
    DisTemp = Distance;
    LMCntPerMM = dCntPerUnit / 100;           //  count / 100mm -> count / mm
    DisTemp *= LMCntPerMM;                    //  mm -> count
    DisTemp *= Pt210;                         //  電子齒輪比分母
    DisTemp /= Pt20E;                         //  電子齒輪比分子  count -> control unit
end;

ret;
```

1.4.3 使用說明

在 PDL 程式中，所有運動參數(如速度 Pt533、加速度 Pt534、減速度 Pt537 等)與位置參數(mMovePos) 都以 control unit 來計算。因此，須將程式輸入的物理量轉換成 count，再利用電子齒輪比轉換為 control unit。本範例程式可解決使用者繁雜的單位換算問題，在執行其他程式前，先行換算須單位轉換的變數。

以下舉例說明。若線性馬達想移動相對於目前位置 45mm 的距離 (如本範例中#define Distance 45)，因程式會將計算結果放入 DisTemp，所以可下達下列運動命令來移動 45mm。

```
mMovePos += DisTemp;
```

1.4.4 應用說明

在本範例程式中，有一驅動器內建變數 Pt20A，此變數表示 AC 伺服馬達所搭配之螺桿導程。另外，dMotorType 則表示目前選用的馬達類型，如表 1.4.4.1 所示。

表 1.4.4.1 dMotorType 變數值與馬達類型對照表

dMotorType	馬達類型
0	線性馬達
1	直驅馬達 / 力矩馬達
2	AC 伺服馬達

在本範例程式中，使用了驅動器內建變數 dCntPerUnit，該變數為編碼器解析度，此變數之定義會依馬達類型不同而有所差異，如表 1.4.4.2 所示。

表 1.4.4.2 馬達類型與 dCntPerUnit 定義對照表

馬達類型	dCntPerUnit 定義
線性馬達	count / 100mm
直驅馬達 / 力矩馬達	count / rev
AC 伺服馬達	count / rev

1.5 教導功能

1.5.1 基本介紹

程式編號	5
檔名	5.pdl
適用驅動器	E 系列驅動器
適用馬達	AC 伺服馬達、線性馬達、直驅馬達、力矩馬達
軟韌體版次需求	E1 系列驅動器：Thunder 1.6.$.$　MDP 2.6.$（含）以上 E2 系列驅動器：Thunder 1.9.$.$　MDP 3.9.$（含）以上 E2-R 系列驅動器：Thunder 1.14.$.$　MDP 4.14.$（含）以上
適用模式	內部位置模式
功能	使用者以數位輸入觸發教導數個位置後，再藉由數位輸入選擇欲移動的教導位置，以達到可定位至使用者教導的任意點之功能。
使用輸入	I2、I3、I4、I6
使用輸出	無

註：
請先至 Thunder 主畫面選擇書籤列中的**工具**、點選 **IO 設定**，將 I2、I3、I4 與 I6 設為 **Not configure**，以避免功能相衝突。

1.5.2 程式內容

```
// ============================ 巨　集 ============================
#define BufferSize   4                          //  定義教導點位置暫存器大小
proc pdl_en() do                                //  激磁馬達
    mEnable = 1;
    till (S_RDY);
end;
proc pdl_dis() do                               //  解激磁馬達
    mEnable = 0;
    till (~S_RDY);
end;
// ============================ 全域變數 ============================
#short BufferNum;                               //  位置暫存器索引值
#long PosBuff[BufferSize];                      //  位置暫存器，用來儲存各個教導點的位置。
```

```
// ============================ 主程序 ============================
#task/1;                                        //  建立一編號為 1 的 task
BufferNum=0;                                    //  位置暫存器索引值初始
mTestRun = 1;                                   //  強制切換至內部位置試運轉模式

_trigger_loop:
if (I2) do                                      //  若 I2 為 ON 狀態，則呼叫教導位置程序。
    call _PosMem;                               //  呼叫教導位置程序
    till (~I2);
end;
if (I6) do                                  //  若 I6 為 ON 狀態，則呼叫移動到教導位置程序。
    pdl_en();
    till (S_RDY);
    if (mEnable=0) do                           //  若無激磁
        printl/101("Motor is not enabled!");    //  列印無激磁警告訊息
        till (~I6);
        goto _trigger_loop;                     //  等待 I6=0 時再回到迴圈開頭
    end;
    call _Move2Tar;                             //  呼叫移動到教導位置程序
    till (~I6);
end;
goto _trigger_loop;
ret;

// ====================  教導位置程序 PosMem ====================
//  =   敘述：依位置暫存器編號 BufferNum 將目前位置儲存到位置暫存器 PosBuff
//  =   引數：無
// ============================================================
_PosMem:
#long posTemp;
posTemp=dFbPos;
PosBuff[BufferNum]= posTemp;                    //  回授位置儲存到位置暫存器
printl/103("Teaching point %ld, Position=%ld", BufferNum, posTemp);
//  列印儲存的教導點編號與位置
BufferNum+=1;                                   //  位置暫存器移到下一個
    if (BufferNum=BufferSize) do                //  索引值超出位置暫存器容量
```

```
            BufferNum=0;                                    //  回到第一個暫存器
        end;
ret;
// ==================== 移動到教導位置 Move2Tar ====================
// =   敘述：依一般輸入的 I4、I3 來選擇移動目標
// =   引數：無
// ================================================================
_Move2Tar:
#short index;
if (~I4 & ~I3) do                                           //  (I4, I3) = (0, 0)
        index =0;
end;
if (~I4 & I3) do                                            //  (I4, I3) = (0, 1)
        index=1;
end;
if (I4 & ~I3) do                                            //  (I4, I3) = (1, 0)
        index=2;
end;
if (I4 & I3) do                                             //  (I4, I3) = (1, 1)
        index=3;
end;
mMovePos=PosBuff[index];                                    //  移動到目標位置
till (~X_run);                                              //  等待馬達停止運動
printl/103("Move to point %ld ended.", index);              //  列印到位訊息
printl/103("Position=%ld", PosBuff[index]);                 //  列印座標
sleep 100;
ret;
```

1.5.3 使用說明

本程式利用一般數位輸入執行教導的功能，數位輸入與功能之對照如表 1.5.3.1。

表 1.5.3.1 數位輸入與功能對照表

數位輸入	功能
I2	教導位置，記錄目前座標到教導點。
I3	選擇要移動到的教導點。
I4	
I6	正緣觸發移動到選擇的教導點。

I2 正緣觸發一次即教導一次，I2 觸發次數與教導位置關係如表 1.5.3.2。此程式可教導四個位置，若 I2 觸發超過 4 次，將從第 1 點依序覆蓋先前的教導點不斷循環。教導位置時若想重新開始，可直接重置驅動器或重開電源。然而，重置後會清除先前記錄的教導點，教導點位置初始皆為零。

表 1.5.3.2 觸發次數與教導位置對照表

索引值	教導位置
0	把所在位置座標記錄到教導點 0。
1	把所在位置座標記錄到教導點 1。
2	把所在位置座標記錄到教導點 2。
3	把所在位置座標記錄到教導點 3。

若要執行移動到教導點，請先到**參數設定**設定好速度、加速度等運動參數，再利用 I4、I3 選擇要移動到的教導點，利用 I6 正緣觸發開始運動。I4、I3 輸入狀態與相對應程序如表 1.5.3.3。

表 1.5.3.3 I3、I4 輸入狀態與對應程序對照表

輸入狀態			動作
I4	I3	I6	
OFF	OFF	正緣觸發	移動到教導點 0
OFF	ON		移動到教導點 1
ON	OFF		移動到教導點 2
ON	ON		移動到教導點 3

1.6 重新激磁延續未完成的絕對座標運動

1.6.1 基本介紹

程式編號	6
檔名	6.pdl
適用驅動器	E 系列驅動器
適用馬達	AC 伺服馬達、線性馬達、直驅馬達、力矩馬達
軟韌體版次需求	E1 系列驅動器：Thunder 1.6.$.$　MDP 2.6.$（含）以上 E2 系列驅動器：Thunder 1.9.$.$　MDP 3.9.$（含）以上 E2-R 系列驅動器：Thunder 1.14.$.$　MDP 4.14.$（含）以上
適用模式	內部位置模式
功能	馬達執行絕對座標運動時，中途若產生錯誤導致馬達解激磁，排除錯誤後再重新激磁時，馬達會移動到解激磁前的目標位置。
使用輸入	I2、I3
使用輸出	無

註：

請先至 Thunder 主畫面選擇書籤列中的**工具**、點選 **IO 設定**，將 I2 與 I3 設為 **Not configure**，以避免功能相衝突。

1.6.2 程式內容

```
// ============================ 巨　集 ============================
#define TargetPos1   Pt531                    //  目標位置 1 (單位：control unit)
#define TargetPos2   Pt532                    //  目標位置 2 (單位：control unit)
proc pdl_en() do                              //  激磁馬達
    mEnable = 1;
    till (S_RDY);
end;
proc pdl_dis() do                             //  解激磁馬達
    mEnable = 0;
    till (~S_RDY);
end;
// ============================ 主程序 ============================
#task/1;                                      //  建立一編號為 1 的 task
#long targetpos;
```

```
#long MotionCompleteFg;                    //  辨識馬達是否確實到達目標位置的旗標
mTestRun = 1;                              //  強制切換至內部位置試運轉模式
MotionCompleteFg = 1;                      //  MotionCompleteFlag = 1 表示已到位

_TargetMove:
till (~I2 & ~I3);
till (I2 | I3);
if (I2 & ~I3) do
    targetpos = TargetPos1;
    MotionCompleteFg = 0;                  //  MotionCompleteFlag = 0 未到位
end;
if (~I2 & I3) do
    targetpos = TargetPos2;
    MotionCompleteFg = 0;
end;
pdl_en();
while (MotionCompleteFg = 0) do
    till (S_RDY);                          //  若因錯誤而跳停，在此等候重新激磁。
    mMovePos = targetpos;                  //  開始移動到目標位置
    sleep 100;
    till (~X_run);                         //  等待運動正常結束或因錯誤而跳停
    if (COIN) do
        MotionCompleteFg = 1;              //  若正常完成運動，則準備離開 while 迴圈。
    end;
end;
goto _TargetMove;
ret;
```

1.6.3 使用說明

使用者設定絕對座標運動的目標位置(TargetPos1, TargetPos2)後，利用觸發 I2 與 I3 來選取目標位置（請參閱表 1.6.3.1），馬達將朝目標位置移動。若中途馬達因產生錯誤而停止運動與解激磁，排除錯誤後重新激磁，程式會判斷馬達是否確實到達目標位置。若未到達目標位置，程式會重複執行 mMovePos 命令使馬達往目標位置移動，直到到達目標位置後才停止運動。

表 1.6.3.1 輸入腳位狀態與目標位置對照表

I2	I3	變數名	目標位置
ON	OFF	TargetPos1	Pt531
OFF	ON	TargetPos2	Pt532

1.6.4 應用說明

以先前範例與本範例程式所使用的 mMovePos 命令執行絕對座標移動，當位置命令 mMovePos 座標值不等於位置回授 dFbPos 數值時，馬達便會往 mMovePos 座標值移動，直到 dFbPos 等於 mMovePos，馬達才結束運動。

若因馬達解激磁而結束運動，重新激磁時位置命令 mMovePos 會回歸當下位置回授 dFbPos 數值。故正常情況下，馬達在運動過程中發生錯誤解激磁，再重新激磁時，馬達不會繼續移動。

因此，本範例程式設計一 while loop，宣告變數 MotionCompleteFg 判斷馬達是否確實到達目標位置。當馬達結束運動後，再判斷馬達是否為激磁狀態。若為激磁狀態，馬達確實到達目標位置；若非激磁狀態，則表示馬達未到達目標位置。

1.7 吋動

1.7.1 基本介紹

程式編號	7
檔名	7.pdl
適用驅動器	E 系列驅動器
適用馬達	AC 伺服馬達、線性馬達、直驅馬達、力矩馬達
軟韌體版次需求	E1 系列驅動器：Thunder 1.6.$.$　MDP 2.6.$（含）以上 E2 系列驅動器：Thunder 1.9.$.$　MDP 3.9.$（含）以上 E2-R 系列驅動器：Thunder 1.14.$.$　MDP 4.14.$（含）以上
適用模式	內部位置模式
功能	透過驅動器的數位輸入觸發吋動功能。
使用輸入	I2、I3
使用輸出	無

註：

請先至 Thunder 主畫面選擇書籤列中的**工具**、點選 **IO 設定**，將 I2 與 I3 設為 **Not configure**，以避免功能相衝突。

1.7.2 程式內容

```
// ============================ 巨　集 ============================
#define CW_JOG   I2                          //  定義 I2 為正向吋動按鈕
#define CCW_JOG   I3                         //  定義 I3 為反向吋動按鈕
#define P_Constant_Vel   1                   //  正方向等速度運動
#define N_Constant_Vel   -1                  //  負方向等速度運動
proc pdl_en() do                             //  激磁馬達
    mEnable = 1;
    till (S_RDY);
end;
proc pdl_dis() do                            //  解激磁馬達
    mEnable = 0;
    till (~S_RDY);
end;
// ============================ 主程序 ============================
#task/1;                                     //  建立一編號為 1 的 task
```

```
sleep 1000;
mTestRun = 1;                                   //  強制切換至內部位置試運轉模式
pdl_en();

_loop:
if (S_RDY) do                                   //  若激磁時
    if (CW_JOG) do                              //  按正向吋動按鈕時
        mRun = P_Constant_Vel;                  //  以 Pt533 或 Pt585 的速度值執行正向吋動
        till (~CW_JOG);                         //  直到釋放正向吋動按鈕
        mStopMove = 1;                          //  運動停止
        till (mStopMove = 0);
    end;
    if (CCW_JOG) do                             //  反向吋動
        mRun = N_Constant_Vel;                  //  以 Pt533 或 Pt585 的速度值執行反向吋動
        till (~CCW_JOG);                        //  直到釋放反向吋動按鈕
        mStopMove = 1;                          //  運動停止
        till (mStopMove = 0);
    end;
else do
    printl/101("Motor is not enabled!");        //  未激磁警告訊息
    printl/103("Please release jog button before enabling motor.");    //  告知釋放吋動按鈕
    till (S_RDY & ~CW_JOG & ~CCW_JOG);          //  直到馬達激磁且已釋放吋動按鈕
end;
goto _loop;
ret;
```

1.7.3 使用說明

本程式利用一般數位輸入執行吋動功能，數位輸入與功能對照如表 1.7.3.1。

表 1.7.3.1 數位輸入與功能對照表

數位輸入	功能
I2	正向吋動
I3	反向吋動

此程式的運動參數是以 Thunder 人機上**試運轉**視窗中的參數來實施的，如表 1.7.3.2。

表 1.7.3.2 運動參數與**試運轉**視窗參數對照表

運動參數	對應的**試運轉**視窗參數
速度	Pt533（旋轉）、Pt585（線性）
加速度	Pt534
減速度	Pt537

註：欲執行吋動程式，須先設定好參數。

1.7.4 應用說明

本範例程式開頭使用到#define CW_JOG I2 這個巨集，使程式中的 CW_JOG 直接對應 I2。此為較具彈性的寫法，以本範例正向吋動(CW_JOG)為例，若使用者想改變輸入為 I8，只要將其修改為#define CW_JOG I8 即可。

本範例程式使用到兩個驅動器內建變數，如表 1.7.4.1 所示。

表 1.7.4.1 驅動器內建變數說明

驅動器內建變數	說明
mRun	用來設定馬達吋動時的方向
mStopMove	設為 1 時停止馬達吋動

1.8 類比輸入控制时動

1.8.1 基本介紹

程式編號	8
檔名	8.pdl
適用驅動器	E 系列驅動器
適用馬達	AC 伺服馬達、線性馬達、直驅馬達、力矩馬達
軟韌體版次需求	E1 系列驅動器：Thunder 1.6.$.$　MDP 2.6.$（含）以上 E2 系列驅動器：Thunder 1.9.$.$　MDP 3.9.$（含）以上 E2-R 系列驅動器：Thunder 1.14.$.$　MDP 4.14.$（含）以上
適用模式	內部位置模式
功能	以類比電壓輸入（0V~+10V）控制馬達速度，並利用數位輸入控制正反向时動。
使用輸入	I2、I3、Ref+、Ref-
使用輸出	無

註：

請先至 Thunder 主畫面選擇書籤列中的**工具**、點選 **IO 設定**，將 I2 與 I3 設為 **Not configure**，以避免功能相衝突。

1.8.2 程式內容

```
// ============================ 巨　集 ============================
#define LINEAR_RATE_SPEED   1500          //  設定線性馬達的額定速度 (單位：mm/s)
#define AC_RATE_SPEED   3000              //  設定 AC 伺服馬達的額定速度 (單位：rpm)
#define DD_RATE_SPEED   600               //  設定直驅馬達 / 力矩馬達的額定速度
                                          //  (單位：rpm)
#float value_temp, rate_speed, speedPervolt;
proc pdl_en() do                          //  激磁馬達
    mEnable = 1;
    till (S_RDY);
end;
proc pdl_dis() do                         //  解激磁馬達
    mEnable = 0;
    till (~S_RDY);
end;
// ========================= 單位轉換 proc =========================
```

```
proc Unit_Transform_Vel(float *velTemp) do
    *velTemp /= 100;                          //  0.01V -> 1V
    value_temp = *velTemp;
    if (dMotorType=0) do                      //  馬達為線性馬達
        rate_speed = LINEAR_RATE_SPEED;
    end;
    if (dMotorType=1) do                      //  馬達為直驅馬達 / 力矩馬達
        rate_speed = DD_RATE_SPEED;
    end;
    if (dMotorType=2) do                      //  馬達為 AC 伺服馬達
        rate_speed = AC_RATE_SPEED;
    end;
    speedPervolt = rate_speed / value_temp;
    *velTemp = speedPervolt;
end;
// ============================ 主程序 ============================
#task/1;
#float JOG_VEL_VOL, JogVel_Scale;
mTestRun = 1;                                 //  強制切換至內部位置試運轉模式
mAdjCmd |= 0x1001;                            //  修正類比偏壓
sleep 600;
if (dAdjState=1) do
    sleep 300;
    printl/101("Modify analog offset failed!");
    goto _task_end;
else do
    sleep 300;
    printl/103("Modify analog offset succeeded.");
end;
JogVel_Scale = Pt300;                         //  Pt300 預設為 6V/額定速度
Unit_Transform_Vel(&JogVel_Scale);            //  執行單位換算
sleep 1000;
pdl_en();
_loop:
till (S_RDY);                                 //  等待伺服就緒
JOG_VEL_VOL = dVcmd*JogVel_Scale;             //  由 Ref+/-讀入電壓，並換算為命令速度。
```

```
if (dMotorType<>0) do
    Pt533 = JOG_VEL_VOL;
else do
    Pt585 = JOG_VEL_VOL;
end;
if ( I2 ) do
    if ( I3 ) do                    //  (I2, I3) = (1, 1)
        mRun = 0;                   //  停止運動
    else do                         //  (I2, I3) = (1, 0)
        mRun = 1;                   //  執行正向吋動
    end;
else do
    if ( I3 ) do                    //  (I2, I3) = (0, 1)
        mRun = -1;                  //  執行反向吋動
    else do                         //  (I2, I3) = (0, 0)
        mRun = 0;                   //  停止運動
    end;
end;
goto _loop;
_task_end:
ret;
```

1.8.3 使用說明

本程式利用類比電壓輸入的大小來控制運動速度，以數位輸入（I2、I3）控制正向吋動、反向吋動或停止運動。數位輸入與功能對照如表 1.8.3.1。

表 1.8.3.1 數位輸入與功能對照表

I2	I3	動作
0	0	停止運動
0	1	反向吋動
1	0	正向吋動
1	1	停止運動

程式內預設每 6V 對應額定速度。欲修改該比例，可在 Thunder 人機介面中修改 Pt300 的值。

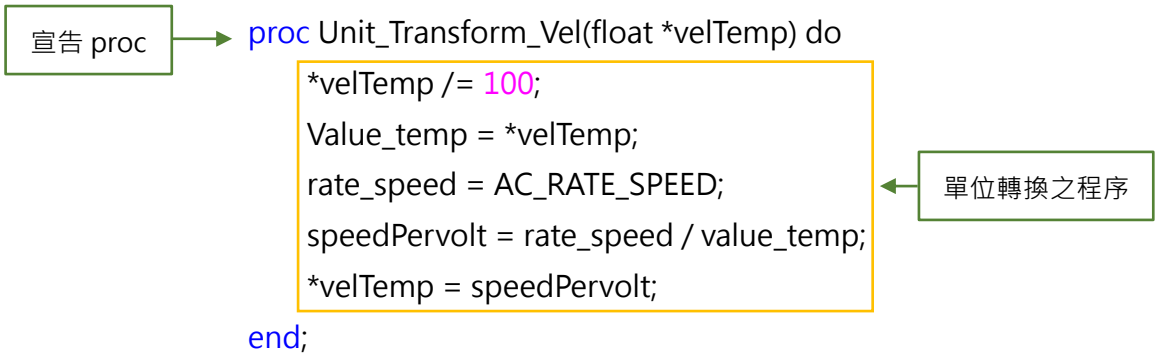
註：至 Thunder 人機介面的**設定精靈**視窗，點選**馬達設定**頁籤，即可查看『額定速度』的值。

1.8.4 應用說明

本範例程式利用驅動器內建變數 dVcmd，讀取由驅動器外部 (Ref+、Ref-) 所輸入之類比電壓，再乘上比例，以得到欲輸出之速度。因線性馬達預設的額定速度為 1500mm/s，以本範例程式的比例 6V=1500mm/s 為例，若使用者輸入類比電壓為 0~6V，則可輸出 0~1500mm/s 的速度。

本範例使用了 PDL 的 proc 機制來執行單位轉換 (請參閱 1.4 節)。在執行主程序前先宣告一個 proc，並在 proc 內設定好執行單位轉換的程序，使用者即可在執行主程序時，將 proc 呼叫出來執行單位轉換。可重覆呼叫執行單位轉換，以減少主程序的複雜度。

Step1. 以本範例為例，在主程序之前先宣告 proc，並將單位轉換的程序建立在 proc 之內。



Step2. 在主程序內若須使用此單位轉換之程序，呼叫此 proc 即可。



2. 內建指令功能

2. 內建指令功能 2-33
2.1 雷射量測誤差補償表建立 2-34
2.1.1 基本介紹 2-34
2.1.2 使用說明 2-35
2.1.3 功能執行流程 2-37
2.2 龍門旋轉軸誤差補償表建立 2-38
2.2.1 基本介紹 2-38
2.2.2 使用說明 2-39
2.2.3 功能執行流程 2-42

本章節說明 E 系列驅動器可使用的內建指令及其功能。使用者可以透過訊息與命令視窗啟動系統內建的各項自動化流程。這些功能由驅動器內部邏輯執行，涵蓋運動控制、量測流程、補償機制與其他自動化功能，無需額外撰寫程式，讓使用者能有效運用驅動器能力並順利完成各項作業流程。

2.1 雷射量測誤差補償表建立

為協助使用者有效率的建立誤差補償表，使用者可以呼叫 E 系列驅動器內建的雷射量測指令後。系統會依據馬達種類及參數自動執行動作，使用者可搭配雷射量測儀器取得誤差數據。

2.1.1 基本介紹

適用驅動器	E 系列驅動器
適用馬達	AC 伺服馬達、線性馬達、直驅馬達、力矩馬達
軟韌體版次需求	E1 系列驅動器：Thunder 1.15.$.$　MDP 2.15.$ (含) 以上 E2 系列驅動器：Thunder 1.15.$.$　MDP 3.15.$ (含) 以上 E2-R 系列驅動器：Thunder 1.15.$.$　MDP 4.15.$ (含) 以上
適用模式	內部位置模式
功能	使用**訊息與命令視窗**啟動系統內建的雷射量測流程，系統控制馬達執行動作時配合外部雷射量測儀器取得誤差數據，並由使用者將量測儀器取得的誤差補償值填入驅動器的誤差補償表。
使用輸入	無
使用輸出	O2

註：請先至 Thunder 主畫面選擇書籤列中的**工具**、點選 **IO 設定**，將 O2 設為 **Not configure**，以避免功能相衝突。

2.1.2 使用說明

Step 1. 依據《E 系列驅動器使用者操作手冊》建立平台，並且完成歸原點作業。

Step 2. 在參數設定頁面中的 Others 分頁中新增表 2.1.2.1 所示之變數，並依需求填入變數數值。

表 2.1.2.1 雷射量測變數

變數名稱	說明	單位
mCompensPoints	量測點數	-
mDwellTime	停止時間	ms
mFullDistance	全行程距離	deg(旋轉) or mm(線性)
mPreTravel	預行程	deg (旋轉) or mm(線性)
mRunCycleCount	量測循環次數	-
mStartPos	量測起始點	deg(旋轉) or mm(線性)
dErrCode	錯誤代碼	-

如圖 2.1.2.1 所示，量測平台為「旋轉馬達搭配滾珠螺桿」之機構，其變數設定如下：

- 量測點數：5 點
- 停止時間：500 毫秒
- 全行程：4000 mm
- 預行程：50 mm
- 量測循環次數：1 次
- 量測起始點：300 mm



圖 2.1.2.1 變數設定

Step 3. 在試運轉頁面中的運動參數欄位設定速度、加減速時間、減速度時間及緊急減速時間。

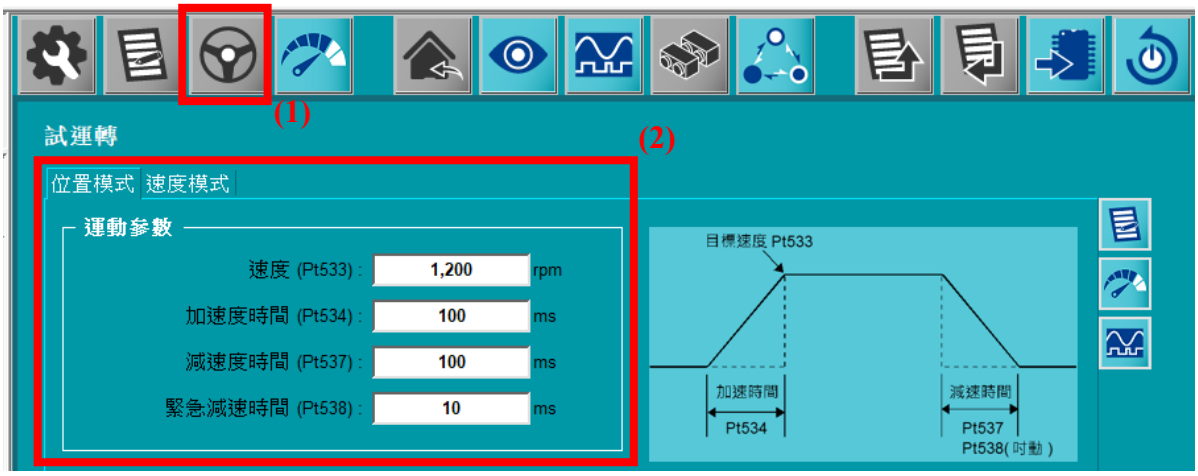


圖 2.1.2.2 運動參數設定

Step 4. 開啟 Thunder →Tools→訊息與命令視窗，執行 run AutoLaserCompensation 指令，馬達開始移動且可透過訊息與命令視窗觀測狀態，如圖 2.1.2.3。錯誤代碼可參考表 2.1.2.2。

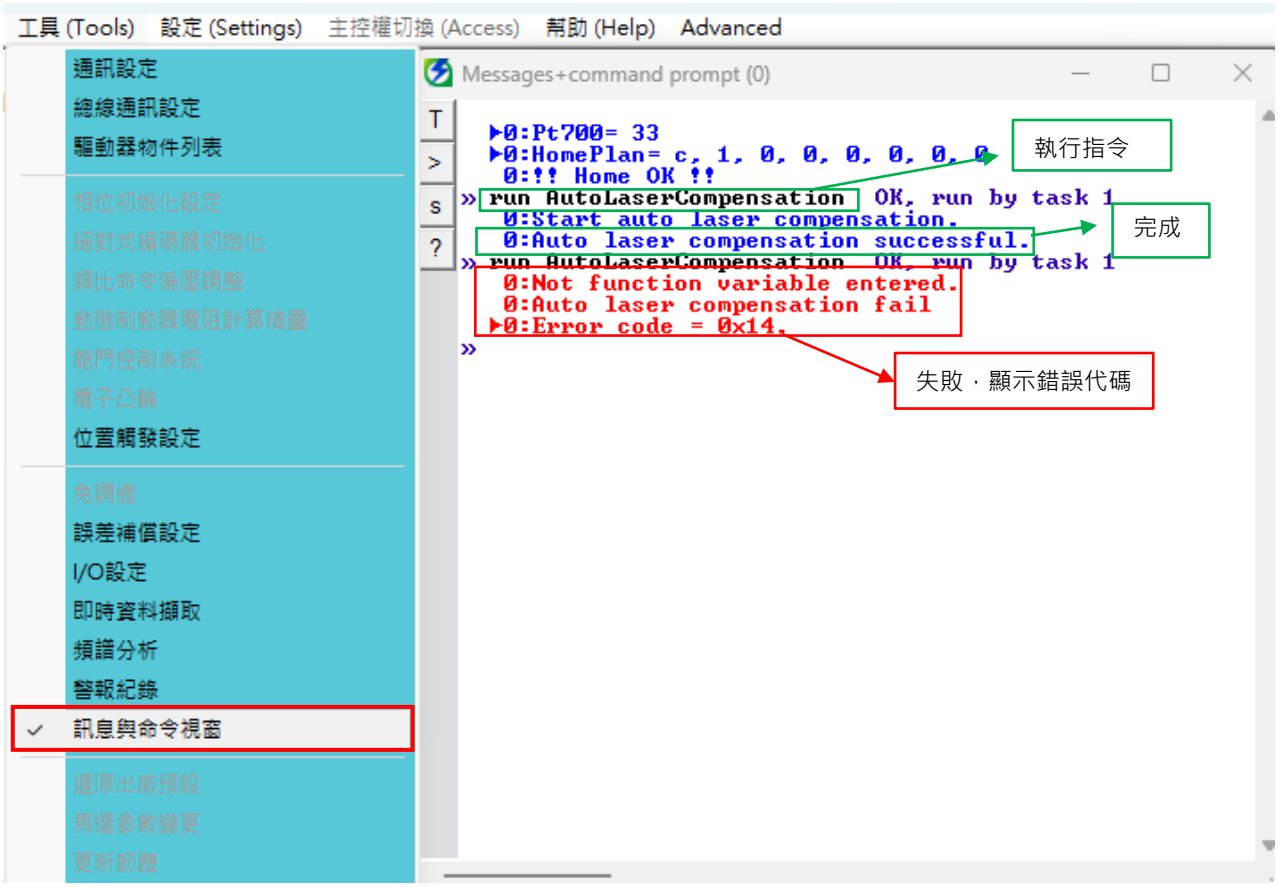


圖 2.1.2.3 訊息與指令視窗

表 2.1.2.2 錯誤代碼說明

變數名稱	值	說明
dErrCode	0x01	移動時發生解激磁
	0x02	模式無法切換至內部位置模式
	0x04	主控權未切換至 Thunder

2.1.3 功能執行流程

圖 2.1.2.2 馬達的實際速度如圖 2.1.3.1 所示。

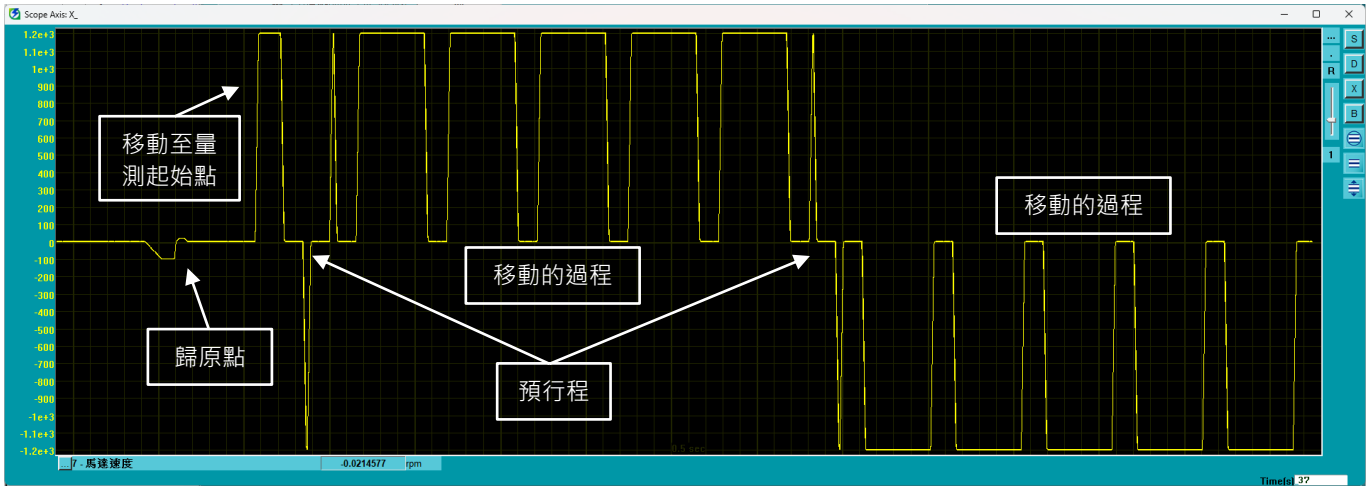


圖 2.1.3.1 流程示意圖

註：

(1) 在輸入觸發指令之前，請先手動完成歸原點作業，並於參數設定中正確填入量測所需的參數值。

(2) 到位訊號觸發時，O2 數位輸出端輸出一個脈波，雷射量測儀器可依據 O2 訊號做出反應。

2.2 龍門旋轉軸誤差補償表建立

為協助使用者有效率的建立龍門旋轉軸誤差補償表，使用者可以呼叫 E 系列驅動器內建的龍門旋轉軸偏移量測指令。系統會啟動龍門線性軸的往復運動並且同時量測龍門旋轉軸於不同位置時的偏移量，使用者可將量測結果填入龍門旋轉軸的誤差補償表，藉以提升龍門平台高速運行時的效率與同步精度。

2.2.1 基本介紹

適用驅動器	E 系列驅動器
適用情境	龍門通訊系統
軟韌體版次需求	E1 系列驅動器：Thunder 1.15.$.$　MDP 2.15.$（含）以上 E2 系列驅動器：Thunder 1.15.$.$　MDP 3.15.$（含）以上 E2-R 系列驅動器：Thunder 1.15.$.$　MDP 4.15.$（含）以上
適用模式	內部位置模式
功能	使用訊息與命令視窗啟動系統內建的龍門旋轉軸偏移量流程。系統會啟動龍門線性軸的往復運動並且同時量測龍門旋轉軸於不同位置時的偏移量，使用者可將量測結果填入龍門旋轉軸的誤差補償表。在龍門平台高速運行時，可即時修正龍門平台結構造成的反向電流，使兩側馬達保持更佳平衡與同步精度。
使用輸入	無
使用輸出	無

註：執行本功能會將 Pt712 清除為零與寫入 Pt711 原點偏移值，並且開啟旋轉軸鎖定功能 Pt710 = t.　　1。

2.2.2 使用說明

Step 1. 依據《E 系列驅動器使用者操作手冊》建立平台，並且完成歸原點。

Step 2. 在參數設定頁面中的 Others 分頁中新增表 2.2.2.1 所示之變數，並依需求填入變數數值。

表 2.2.2.1 龍門旋轉軸量測變數

變數名稱	說明	單位
mCompensPoints	量測點數	-
mFullDistance	全行程距離	Control unit
mRunCycleCount	量測循環次數	-
mStartPos	補償起始點	Control unit
dErrCode	錯誤代碼	-

如圖 2.2.2.1 所示，量測範例變數設定如下：

- 量測點數： 10 點
- 全行程： 5000000 um
- 量測循環次數： 1 次
- 量測起始點： 400000 um



圖 2.2.2.1 變數設定

Step 3. 在試運轉頁面中的運動參數欄位設定速度、加減速時間、減速度時間及緊急減速時間。

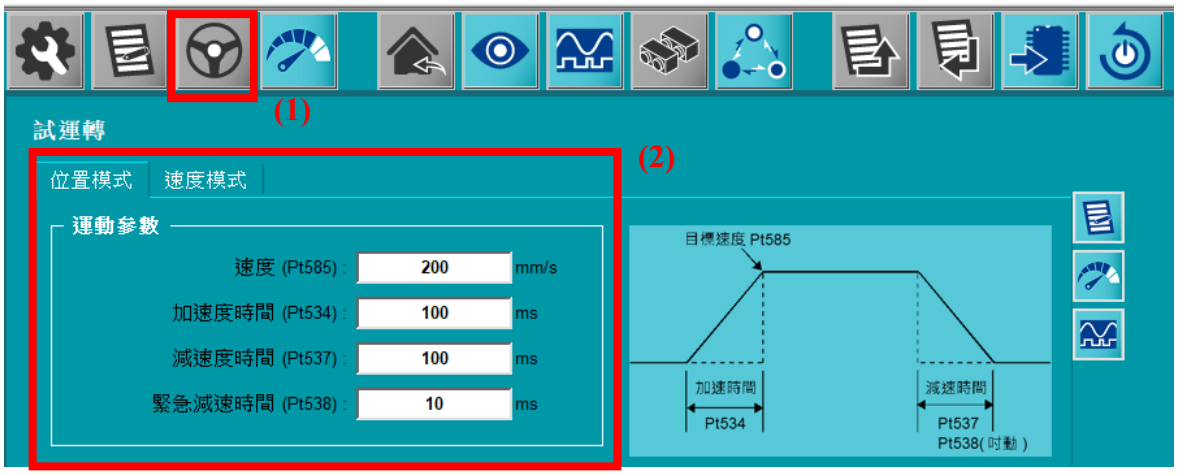


圖 2.2.2.2 運動參數設定

Step 4. 開啟 Thunder →Tools→訊息與命令視窗，執行 run AutoYawErrEst 指令，馬達開始移動且可透過訊息與命令視窗觀測狀態，如下圖 2.2.2.3。錯誤代碼可參考表 2.2.2.2。

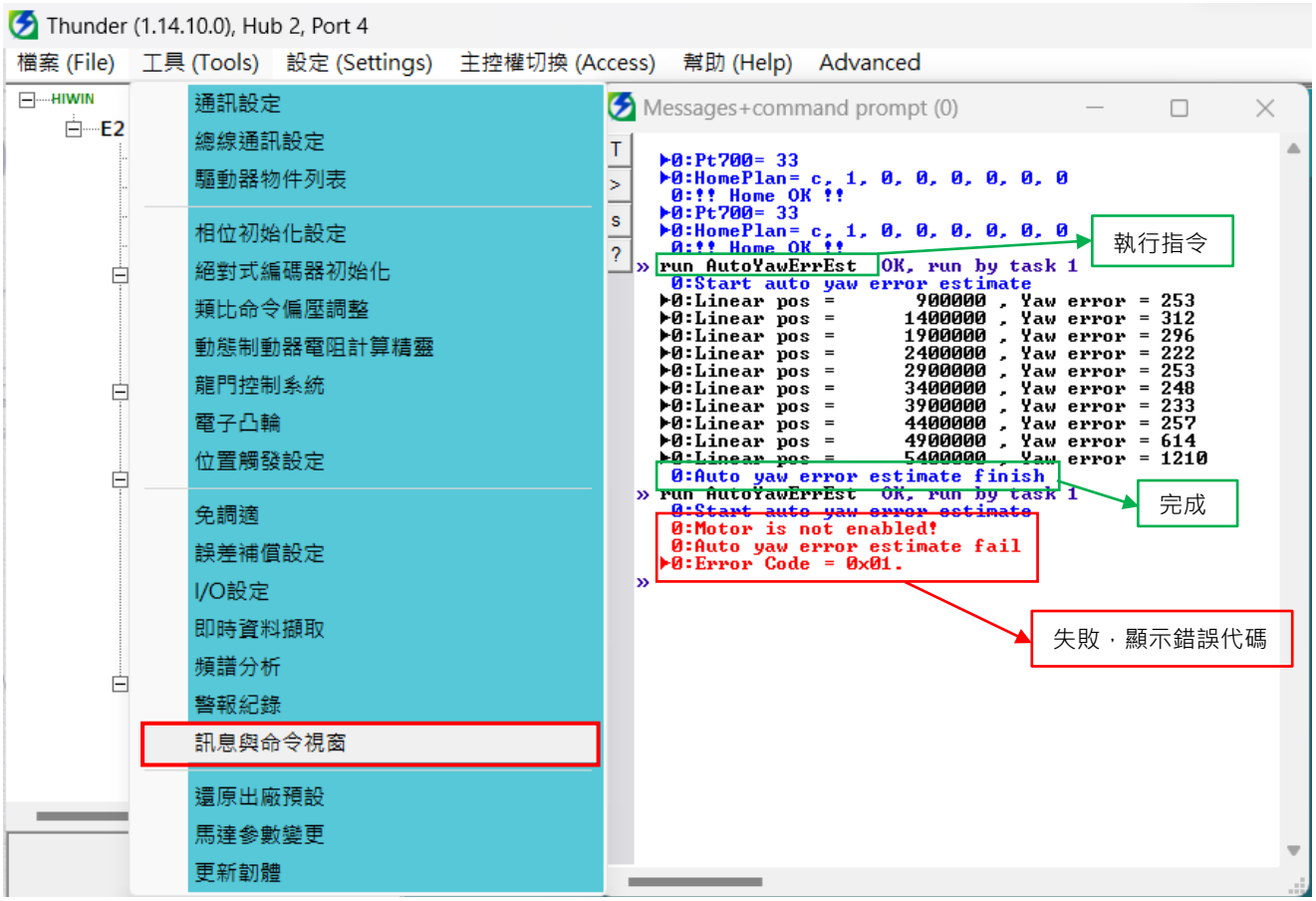


圖 2.2.2.3 訊息與指令視窗

表 2.2.2.2 錯誤代碼說明表

變數名稱	編號	說明
dErrCode	0x01	移動時發生解激磁
	0x04	主控權未切換至 Thunder
	0x08	未完成歸原點
	0x10	龍門主軸設定參數至龍門旋轉軸錯誤
	0x20	主軸讀取旋轉軸參數錯誤
	0x40	未開啟龍門模式

Step 5. 完成時，將 Thunder 連線至龍門旋轉軸。

Step 6. 將 Yaw error 填入龍門旋轉軸誤差補償表，啟動誤差補償表。如圖 2.2.2.4。

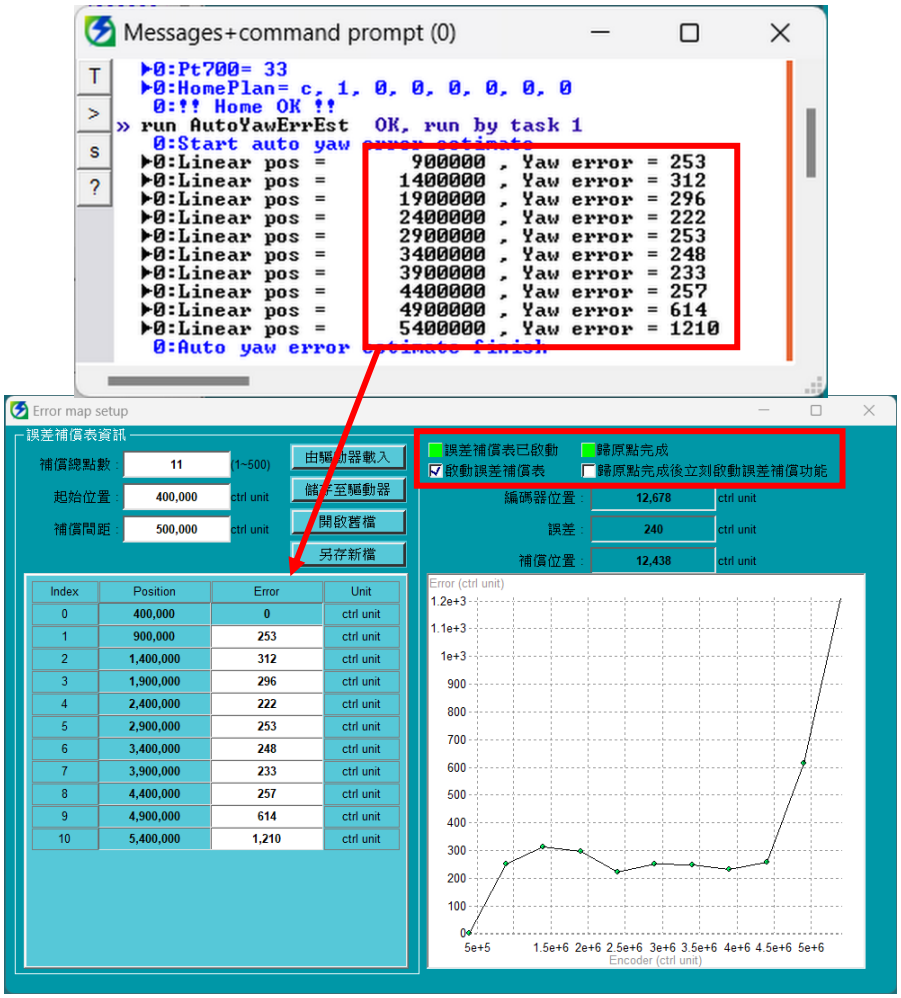


圖 2.2.2.4 誤差補償表設定

註：

(1) 誤差補償表是建立在龍門旋轉軸，需要勾選啟動誤差補償表並且 Save to flash。

(2) 補償表總點數 = 設定的量測點數 + 1 (補償起始點)

(3) 補償表間距 = 全行程距離 / 量測點數。

2.2.3 功能執行流程

馬達實際速度如下圖 2.2.3.1 所示。

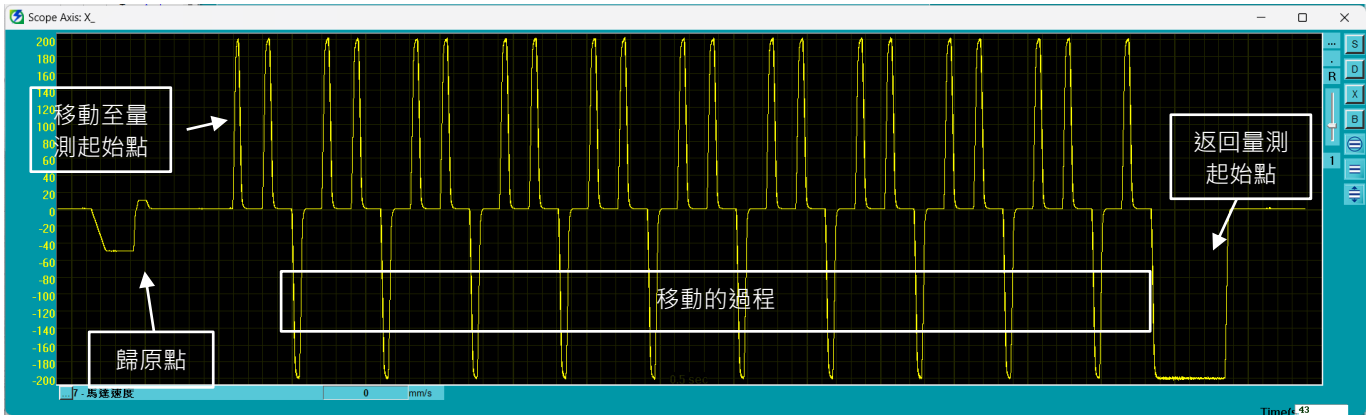


圖 2.2.3.1 流程示意圖

註：在執行 run AutoYawErrEst 指令之前，請先完成歸原點作業，並於變數設定中正確填入量測所需的參數值。

3. 常用變數

3. 常用變數 3-1
3.1 運動控制變數 (讀 / 寫) 3-2
3.2 運動狀態變數 (讀) 3-8
3.3 物理量監控變數 (讀) 3-10
3.4 伺服訊號狀態變數 (讀) 3-14
3.5 伺服訊號 IO 變數 (讀) 3-16

本章提供使用者 E 系列驅動器相關變數及說明以利撰寫編譯 PDL。主要分為五大類：運動控制變數 (讀 / 寫)、運動狀態變數 (讀)、物理量監控變數 (讀)、伺服訊號狀態變數 (讀) 與伺服訊號 IO 變數 (讀)。

3.1 運動控制變數 (讀 / 寫)

變數名稱	mEnable	操作模式	所有	變數單位	-
型態	Int16	變數範圍	0 ~ 1		
使用說明	激磁 / 解激磁馬達				
變數定義	0 - 解激磁馬達 1 - 激磁馬達				

範例說明	mEnable = 1; // 激磁馬達
	till (S_RDY); // 等待馬達激磁
	mEnable = 0; // 解激磁馬達
	till (~S_RDY); // 等待馬達解激磁				

變數名稱	mTestRun	操作模式	所有	變數單位	-
型態	Unit16	變數範圍	0 ~ 2		
使用說明	試運轉模式切換				
變數定義	0 - 無切換 1 - 切換至內部位置試運轉模式 2 - 切換至內部速度試運轉模式				

範例說明	mTestRun = 1; // 強制切換至內部位置試運轉模式
	till (dOperMode = 0); // 確認切換至內部位置模式
	mMovePos = 100;
	sleep 1; // 等待狀態轉態延遲時間
	till (COIN); // 等待到位
	mTestRun = 2; // 強制切換至內部速度試運轉模式
	till (dOperMode = 4); // 確認切換至內部速度模式
	mJog = 1; // 正方向等速度運動
	sleep 1; // 等待狀態轉態延遲時間
	till (V_CMP); // 等待達到預定速度				

變數名稱	mMovePos	操作模式	內部位置	變數單位	控制單位
型態	Int32	變數範圍	-2147483648 ~ 2147483647		
使用說明	移動馬達至指定位置				
變數定義	數值 - 指定位置				

範例說明	Pt533 = 100; // 設定旋轉馬達運動速度為 100 rpm
	mTestRun = 1; // 強制切換至內部位置試運轉模式
	mMovePos = 100; // 移動馬達至 100 控制單位位置
	// 旋轉馬達運動速度取決於 Pt533 的設定
	// 線性馬達運動速度取決於 Pt585 的設定
	// 馬達運動加速度時間取決於 Pt534 的設定
	// 馬達運動減速度時間取決於 Pt537 的設定
	sleep 1; // 等待狀態轉態延遲時間
	till (COIN); // 等待到位
	// 請注意：不可同時下達 mMovePos 與 mRun 及 mP2P 命令				

變數名稱	mP2P	操作模式	內部位置	變數單位	-
型態	Uint16	變數範圍	0 ~ 1		
使用說明	內部位置試運轉模式下點對點(P2P)運動				
變數定義	0 - 停止 P2P 運動 1 - 開始 P2P 運動				

範例說明	mTestRun = 1; // 強制切換至內部位置試運轉模式
	mP2P = 1; // 開始 P2P 運動
	// 旋轉馬達運動速度取決於 Pt533 的設定
	// 線性馬達運動速度取決於 Pt585 的設定
	// 馬達運動加速度時間取決於 Pt534 的設定
	// 馬達運動減速度時間取決於 Pt537 的設定
	// 馬達運動點位 P1 取決於 Pt531 的設定
	// 馬達運動點位 P2 取決於 Pt532 的設定
	// 馬達運動等待時間取決於 Pt535 的設定
	// 請注意：不可同時下達 mMovePos 與 mRun 及 mP2P 命令
	mP2P = 0; // 停止 P2P 運動
	// 馬達運動減速度時間取決於 Pt538 的設定				

變數名稱	mRun	操作模式	內部位置	變數單位	-
型態	Int16	變數範圍	-1 ~ 1		
使用說明	內部位置試運轉模式下等速度運動				
變數定義	0 - 停止運動 1 - 正方向等速度運動 -1 - 負方向等速度運動				

範例說明	mTestRun = 1; // 強制切換至內部位置試運轉模式
	mRun = 1; // 正方向等速度運動
	mRun = -1; // 負方向等速度運動
	mRun = 0; // 停止運動
	// 旋轉馬達運動速度取決於 Pt533 的設定
	// 線性馬達運動速度取決於 Pt585 的設定
	// 馬達運動加速度時間取決於 Pt534 的設定
	// 馬達運動減速度時間取決於 Pt537 的設定
	// 請注意：不可同時下達 mMovePos 與 mRun 及 mP2P 命令				

變數名稱	mStopMove	操作模式	內部位置	變數單位	-
型態	Uint16	變數範圍	0 ~ 2		
使用說明	停止馬達				
變數定義	0 - 無作用或已停止 1 - 使用緊急減速度停止馬達 (Pt538) 2 - 使用減速度停止馬達 (Pt537)				

範例說明	mTestRun = 1; // 強制切換至內部位置試運轉模式
	mMovePos = 10000; // 移動馬達至 10000 控制單位位置
	till (I1); // 確認 I1 觸發（假設 I1 為使用者所設定的停止動作輸入）
	mStopMove = 1;
	// 馬達停止移動
	// 馬達運動減速度時間取決於 Pt538 的設定				

變數名稱	mJog	操作模式	內部速度	變數單位	-
型態	Int16	變數範圍	-1 ~ 1		
使用說明	內部速度試運轉模式下等速度運動				
變數定義	0 - 停止運動 1 - 正方向等速度運動 -1 - 負方向等速度運動				

範例說明	mTestRun = 2; // 強制切換至內部速度試運轉模式
	mJog = 1; // 正方向等速度運動
	mJog = -1; // 負方向等速度運動
	mJog = 0; // 停止運動
	// 旋轉馬達運動速度取決於 Pt304 的設定
	// 線性馬達運動速度取決於 Pt383 的設定
	// 馬達運動加速度時間取決於 Pt305 的設定
	// 馬達運動減速度時間取決於 Pt306 的設定				

變數名稱	mHome	操作模式	所有	變數單位	-
型態	Int16	變數範圍	0 ~ 2		
使用說明	執行歸原點				
變數定義	0 - 不執行歸原點 1 - 執行歸原點 2 - 暫停執行歸原點				

範例說明	mHome = 0; // 不執行歸原點
	mHome = 1; // 執行歸原點
	till (dHomeState = 2);				

變數名稱	mAdjCmd	操作模式	所有	變數單位	-
型態	Uint16	變數範圍	0 ~ 65535		
使用說明	調整型動作命令				
變數定義	0 - 無作用 0x1001 - 啟動偏壓校正動作 0x1008 - 啟動清除多圈圈數				

範例說明	mAdjCmd = 0x1008;
	sleep 600; // 動作執行時間約 600 毫秒
	if (dAdjState = 1) do
	  printl/101("Adjusted action failed!");
	else do
	  printl/103("Adjusted action succeeded.");
	end;				

變數名稱	mInInvert	操作模式	所有	變數單位	-
型態	Uint16	變數範圍	0 ~ 65535		
使用說明	數位輸入訊號反向				
變數定義	每個 bit 分別代表對應的 Input 邏輯反向 0 - 無作用 0x0001 - I1 反向 0x0002 - I2 反向 0x0004 - I3 反向 0x0005 - I1、I3 反向 以此類推				

範例說明	mInInvert = 1; // I1 反向
	till (I1); // 確認 I1 觸發				

變數名稱	mOutInvert	操作模式	所有	變數單位	-
型態	Uint16	變數範圍	0 ~ 65535		
使用說明	數位輸出訊號反向				
變數定義	每個 bit 分別代表對應的 Output 邏輯反向 0 - 無作用 0x0001 - O1 反向 0x0002 - O2 反向 0x0004 - O3 反向 0x0005 - O1、O3 反向 以此類推				
範例說明	mOutInvert = 1; // O1 反向				

變數名稱	mCompensPoints	操作模式	內部位置	變數單位	-
型態	Uint16	變數範圍	0 ~ 65535		
使用說明	量測點數				
變數定義	參照第 2 章：內建指令應用功能的使用說明				
範例說明	-				

變數名稱	mDwellTime	操作模式	內部位置	變數單位	ms
型態	Uint16	變數範圍	0 ~ 65535		
使用說明	停止時間				
變數定義	參照第 2 章：內建指令應用功能的使用說明				
範例說明	-				

變數名稱	mRunCycleCount	操作模式	內部位置	變數單位	-
型態	Uint16	變數範圍	0 ~ 65535		
使用說明	量測循環次數				
變數定義	參照第 2 章：內建指令應用功能的使用說明				
範例說明	-				

變數名稱	mStartPos	操作模式	內部位置	變數單位	依據應用功能定義
型態	int32	變數範圍	-2147483648 ~ 2147483647		
使用說明	量測起始點				
變數定義	參照第 2 章：內建指令應用功能的使用說明				
範例說明	-				

變數名稱	mFullDistance	操作模式	內部位置	變數單位	依據應用功能定義
型態	Uint32	變數範圍	0 ~ 4294967295		
使用說明	全行程距離				
變數定義	參照第 2 章：內建指令應用功能的使用說明				
範例說明	-				

變數名稱	mPreTravel	操作模式	內部位置	變數單位	依據應用功能定義
型態	Uint32	變數範圍	0 ~ 4294967295		
使用說明	預行程				
變數定義	參照第 2 章：內建指令應用功能的使用說明				
範例說明	-				

3.2 運動狀態變數 (讀)

變數名稱	dMotorType	操作模式	所有	變數單位	-
型態	Int16	變數範圍	0 ~ 2		
使用說明	馬達類型顯示				
變數定義	0 - 線性馬達 1 - 直驅馬達 / 力矩馬達 2 - AC 伺服馬達				

變數名稱	dCntPerUnit	操作模式	所有	變數單位	參閱下方說明
型態	Int32	變數範圍	0 ~ 2147483648		
使用說明	編碼器解析度單位顯示				
變數單位說明	線性馬達：count / 100mm 直驅馬達 / 力矩馬達：count / rev AC 伺服馬達：count / rev				

變數名稱	dVcmd	操作模式	所有	變數單位	伏特
型態	Float32	變數範圍	-10 ~ 10		
使用說明	速度命令電壓顯示				

變數名稱	dTcmd	操作模式	所有	變數單位	伏特
型態	Float32	變數範圍	-10 ~ 10		
使用說明	轉矩命令電壓顯示				

變數名稱	dHomeState	操作模式	所有	變數單位	-
型態	Int16	變數範圍	-1 ~ 4		
使用說明	歸原點狀態顯示				
變數定義	0 - 尚未執行歸原點 2 - 歸原點成功 3 - 正在執行歸原點 4 - 歸原點暫停 -1 - 歸原點失敗				

變數名稱	dOperMode	操作模式	所有	變數單位	-
型態	Int16	變數範圍	0 ~ 4		
使用說明	操作模式顯示				
變數定義	0 - 內部位置模式 1 - 位置模式 2 - 速度模式 3 - 轉矩模式 4 - 內部速度模式				

變數名稱	X_run	操作模式	位置、內部位置	變數單位	-
型態	State	變數範圍	-		
使用說明	馬達運動輸出訊號				

變數名稱	dAdjState	操作模式	所有	變數單位	-
型態	Uint16	變數範圍	0 ~ 1		
使用說明	調整型動作顯示				
變數定義	此為 mAdjCmd 的設定結果 0 - 成功 1 - 失敗 225 - 正在執行				

變數名稱	dErrCode	操作模式	內部位置	變數單位	-
型態	Uint16	變數範圍	0x0000~0x0040		
使用說明	參照第 2 章：內建指令應用功能的使用說明				

3.3 物理量監控變數 (讀)

物理量監控變數名稱說明請參閱《E1 系列驅動器使用者操作手冊》圖 11.3.1.1、《E2 系列驅動器使用者操作手冊》圖 11.3.1.1，其表現與《E1 系列驅動器使用者操作手冊》、《E2 系列驅動器使用者操作手冊》中的『Scope 監控項目-物理量』相同。

物理量(1)	dPosErr	型態	Int32	變數單位	控制單位
說明	位置偏差				

物理量(2)	dFbPos	型態	Int32	變數單位	控制單位
說明	位置回授				

物理量(3)	dPosRefVel	型態	Float32	變數單位	rpm 或 mm/s
說明	位置命令速度 註：此變數不適用於 E1 / E2 韌體版本 2.10.6 / 3.10.6 (含) 以下。				

物理量(4)	dDLPosErr	型態	Int32	變數單位	控制單位
說明	馬達-負載位置間的偏差				

物理量(5)	dVelFFCmd	型態	Float32	變數單位	rpm 或 mm/s
說明	速度前饋				

物理量(6)	dVelCmd	型態	Float32	變數單位	rpm 或 mm/s
說明	速度命令				

物理量(7)	dFbVel	型態	Float32	變數單位	rpm 或 mm/s
說明	馬達速度				

物理量(8)	dToqFFCmd	型態	Float32	變數單位	A-amp
說明	轉矩前饋 註：當 E1 / E2 韌體版本為 2.10.6 / 3.10.6 (含) 以下，此變數單位為 0.1% drive peak current。				

物理量(9)	dToqCmd	型態	Float32	變數單位	A-amp
說明	轉矩命令 註：當 E1 / E2 韌體版本為 2.10.6 / 3.10.6 (含) 以下，此變數單位為 0.1% drive peak current。				

物理量(10)	dCurrCmd	型態	Float32	變數單位	A-amp
說明	電流命令 註：當 E1 / E2 韌體版本為 2.10.6 / 3.10.6 (含) 以下，此變數單位為 0.1% drive peak current。				

物理量(11)	dFbCurr	型態	Float32	變數單位	A-amp
說明	馬達電流 註：當 E1 / E2 韌體版本為 2.10.6 / 3.10.6 (含) 以下，此變數代表馬達電流的平方值，單位為 A-amp^2。				

物理量(12)	dServoVoltPercent	型態	Float32	變數單位	%
說明	伺服電壓百分比 註：此變數不適用於 E1 / E2 韌體版本 2.10.6 / 3.10.6 (含) 以下。				

物理量(13)	dHallSignal	型態	Int16	變數單位	-
說明	數位霍爾訊號				

物理量(14)	dMotLoadPercent	型態	Float32	變數單位	%
說明	馬達過載百分比 註：此變數不適用於 E1 / E2 韌體版本 2.10.6 / 3.10.6 (含) 以下。				

物理量(15)	dAmpPosErr	型態	Int32	變數單位	count
說明	位置放大誤差				

物理量(16)	dVelErr	型態	Float32	變數單位	rad/s 或 mm/s
說明	速度誤差				

物理量(17)	dMasterPos	型態	Int32	變數單位	控制單位
說明	主軸位置回授 註：此變數僅適用於主軸驅動器。				

物理量(18)	dSlavePos	型態	Int32	變數單位	控制單位
說明	從軸位置回授 註：此變數僅適用於從軸驅動器。				

物理量(19)	dYawPos	型態	Int32	變數單位	控制單位
說明	旋轉位置				

物理量(20)	X_run_pcmd	型態	State	變數單位	-
說明	位置命令傳送結束狀態				

物理量(21)	dEffectGain	型態	Float32	變數單位	-
說明	有效增益				
定義	0 - 未使用增益切換 1 - 使用第一組有效增益 2 - 使用第二組有效增益				

物理量(22)	dFbPosInt	型態	Int32	變數單位	控制單位
說明	內部位置回授				

物理量(23)	dLinearCmd	型態	Int32	變數單位	A-amp
說明	龍門線性軸電流命令				

物理量(24)	dYawCmd	型態	Int32	變數單位	A-amp
說明	龍門旋轉軸電流命令				

物理量(25)	dYawPosErr	型態	Int32	變數單位	控制單位
說明	龍門旋轉軸位置誤差				

物理量(26)	dLoadPos_MultiMotion	型態	Int32	變數單位	控制單位
說明	負載端單圈位置（多工位專用） 註：此變數不適用於 E1 / E2 韌體版本 2.10.6 / 3.10.6（含）以下。				

物理量(27)	dLoadPos	型態	Int32	變數單位	控制單位
說明	負載端位置 註：此變數不適用於 E1 / E2 韌體版本 2.10.6 / 3.10.6（含）以下。				

物理量(28)	dMotPeakLoadPercent	型態	Float32	變數單位	%
說明	峰值負載率 註：此變數不適用於 E1 / E2 韌體版本 2.10.6 / 3.10.6（含）以下。				

物理量(29)	dRegenLoadPercent	型態	Float32	變數單位	%
說明	回生負載率 註：此變數不適用於 E1 / E2 韌體版本 2.10.6 / 3.10.6（含）以下。				

物理量(30)	dDrvLoadPercent	型態	Float32	變數單位	%
說明	驅動器過載百分比 註：此變數不適用於 E1 / E2 韌體版本 2.10.6 / 3.10.6（含）以下。				

物理量(31)	dBusVolt	型態	Float32	變數單位	volt
說明	主電源電壓 註：此變數不適用於 E1 / E2 韌體版本 2.10.6 / 3.10.6 (含) 以下。				

物理量(32)	dExtPosCmd	型態	Int32	變數單位	控制單位
說明	外部位置命令 註：此變數不適用於 E1 / E2 韌體版本 2.10.6 / 3.10.6 (含) 以下。				

物理量(33)	dInrPosCmd	型態	Int32	變數單位	控制單位
說明	內部位置命令 註：此變數不適用於 E1 / E2 韌體版本 2.10.6 / 3.10.6 (含) 以下。				

3.4 伺服訊號狀態變數 (讀)

伺服訊號狀態變數表現與《E1 系列驅動器使用者操作手冊》、《E2 系列驅動器使用者操作手冊》中的『Scope 監控項目-伺服訊號狀態』相同。

監控編號	變數名稱	說明
51	S_ON	伺服 ON 輸入訊號
52	P_CON	P 動作命令輸入訊號
53	P_OT	禁止正轉側驅動輸入訊號
54	N_OT	禁止反轉側驅動輸入訊號
55	ALM_RST	警報重置輸入訊號
56	P_CL	正轉側外部轉矩限制輸入訊號
57	N_CL	反轉側外部轉矩限制輸入訊號
58	C_SEL	控制方式切換輸入訊號
59	SPD_D	馬達旋轉方向切換輸入訊號
60	SPD_A	內部設定速度切換 1 輸入訊號
61	SPD_B	內部設定速度切換 2 輸入訊號
62	ZCLAMP	零位固定輸入訊號
63	INHIBIT	命令脈波禁止輸入訊號
64	G_SEL	增益切換輸入訊號
65	PSEL	命令脈波輸入倍率切換輸入訊號
66	RST	驅動器重置輸入訊號
67	DOG	近原點開關輸入訊號
68	HOM	驅動器內建的歸原點程序輸入訊號
69	MAP	驅動器誤差補償啟動輸入訊號
70	FSTP	強制停止輸入訊號
71	CLR	位置偏差清除輸入訊號
72	ALM	錯誤警報輸出訊號
73	COIN	定位完成輸出訊號
74	V_CMP	速度一致輸出訊號
75	TGON	旋轉檢出/移動檢出輸出訊號
76	D_RDY	驅動器就緒輸出訊號
77	S_RDY	伺服就緒輸出訊號
78	CLT	轉矩限制檢出輸出訊號
79	VLT	速度限制檢出輸出訊號
80	BK	制動器控制輸出訊號
81	WARN	警告輸出訊號

監控編號	變數名稱	說明
82	NEAR	定位接近輸出訊號
83	PSELA	命令脈波輸入倍率切換輸出訊號
84	PT	位置觸發數位輸出訊號
85	DBK	外接動態制動器輸出訊號
86	HOMED	驅動器歸原點完成輸出
87	PAO	編碼器分頻脈波輸出 A 相訊號
88	PBO	編碼器分頻脈波輸出 B 相訊號
89	PZO	編碼器分頻脈波輸出 Z 相訊號
90	X_ind	原點訊號

3.5 伺服訊號 IO 變數 (讀)

變數名稱	說明
I1	數位輸入訊號 1
I2	數位輸入訊號 2
I3	數位輸入訊號 3
I4	數位輸入訊號 4
I5	數位輸入訊號 5
I6	數位輸入訊號 6
I7	數位輸入訊號 7
I8	數位輸入訊號 8
I9	數位輸入訊號 9
I10	數位輸入訊號 10
O1	數位輸出訊號 1
O2	數位輸出訊號 2
O3	數位輸出訊號 3
O4	數位輸出訊號 4
O5	數位輸出訊號 5

範例說明

till (I1); // 等待 I1 觸發，其他輸入訊號以此類推。
seton O1;
// 當 O1 未設定輸出條件時，可透過 PDL 改變輸出狀態，其他輸出訊號以此類推。

3.6 總線通訊自定義物件變數 (讀/寫)

變數名稱	說明
OD_0x3201	數據類型為 DINT 的自定義物件(1)
OD_0x3202	數據類型為 DINT 的自定義物件(2)
OD_0x3203	數據類型為 DINT 的自定義物件(3)
OD_0x3204	數據類型為 DINT 的自定義物件(4)
OD_0x3205	數據類型為 DINT 的自定義物件(5)
OD_0x3206	數據類型為 DINT 的自定義物件(6)
OD_0x3207	數據類型為 DINT 的自定義物件(7)
OD_0x3208	數據類型為 DINT 的自定義物件(8)
OD_0x3209	數據類型為 DINT 的自定義物件(9)
OD_0x3210	數據類型為 REAL 的自定義物件(0)
OD_0x3211	數據類型為 REAL 的自定義物件(1)
OD_0x3212	數據類型為 REAL 的自定義物件(2)
OD_0x3213	數據類型為 REAL 的自定義物件(3)
OD_0x3214	數據類型為 REAL 的自定義物件(4)