

**HIWIN® MIKROSYSTEM**



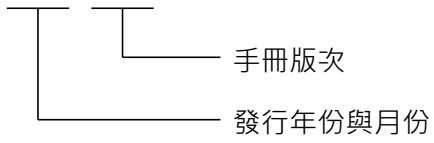
# Application Note

E 系列 EtherCAT 驅動器  
搭配 HIMC3 iA Studio

# 修訂紀錄

手冊版次資訊亦標記於手冊封面右下角。

MD55UC01-2604\_V1.0



發行日期	版次	適用產品	更新內容
2026/04/20	1.0	E 系列 EtherCAT 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](https://www.hiwinmikro.tw/Downloads/ManualOverview_TC.htm) )。

# 序言

本手冊說明 E 系列 EtherCAT 驅動器搭配 HIMC3 運動控制器時，軟體 iA Studio 的操作。欲瞭解 E 系列驅動器的詳細資訊，請參閱相關的使用者手冊。

## 軟硬體規格

名稱	軟體 / 韌體版本
E1 系列 EtherCAT 驅動器	軟體 ( Thunder ): 1.14.11.0 ( 含 ) 以上 韌體 : 2.14.9 ( 含 ) 以上 ESI 檔 : HIWIN_MIKROSYSTEM_EDxF_20250725 以上
E2 系列 EtherCAT 驅動器	軟體 ( Thunder ): 1.14.11.0 ( 含 ) 以上 韌體 : 3.14.9 / 4.14.9 ( 含 ) 以上 ESI 檔 : HIWIN_MIKROSYSTEM_EDxF_20250725 以上
HIMC3 運動控制器	軟體 ( iA Studio ): 3.3.0 ( 含 ) 以上 韌體 : 3.3.0 ( 含 ) 以上

# 目錄

1.	連線與模組設定.....	1-1
1.1	連線至控制器.....	1-2
1.1.1	連線設定.....	1-2
1.1.2	經由 Ethernet 連線至控制器.....	1-4
2.	參數設定.....	2-1
2.1	配置驅動器.....	2-2
2.2	安裝 ESI 檔.....	2-3
2.3	設定運動控制軸.....	2-5
2.4	配置 PDO.....	2-8
3.	試運轉.....	3-1
3.1	激磁.....	3-2
3.2	歸原點.....	3-3
3.3	吋動.....	3-5
3.4	相對移動.....	3-6
3.5	點對點運動.....	3-7
3.6	HMPL Editor.....	3-8
3.6.1	點對點運動.....	3-11
3.6.2	歸原點.....	3-13
4.	其他應用設定.....	4-1
4.1	誤差補償功能設定.....	4-2
4.1.1	1D 誤差補償.....	4-2
4.1.2	2D 誤差補償.....	4-3

# 1. 連線與模組設定

---

1.	連線與模組設定.....	1-1
1.1	連線至控制器.....	1-2
1.1.1	連線設定.....	1-2
1.1.2	經由 Ethernet 連線至控制器.....	1-4

## 1.1 連線至控制器

使用者可於 Connection Setting 視窗選擇通訊類型並連線至控制器。

### 1.1.1 連線設定

1. 點擊選單列的 **Controller**，再點擊 **Connection Setting**，開啟 Connection Setting 視窗。

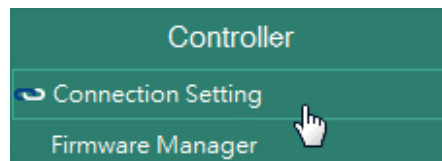


圖 1.1.1.1 Connection Setting

2. 選擇控制器與電腦連接的孔位 ( **CN3** 或 **CN4** )。

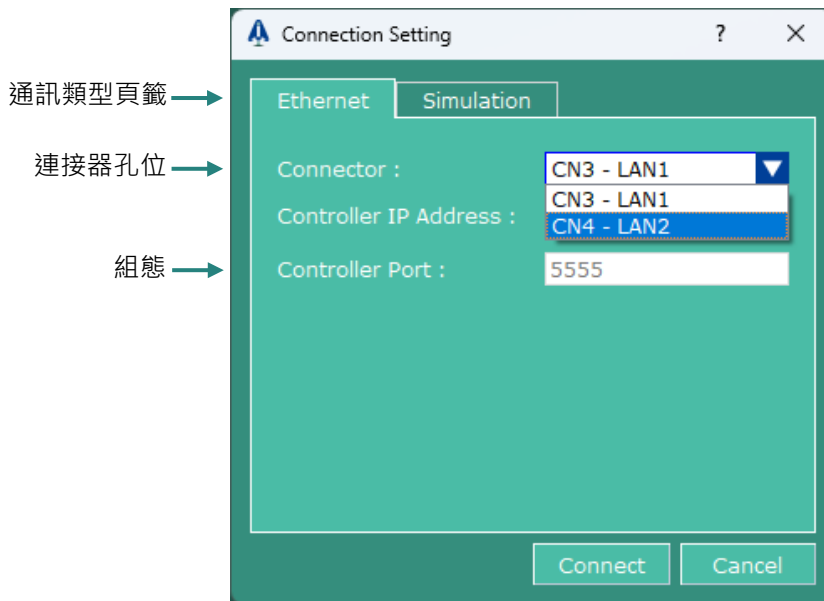


圖 1.1.1.2 Connection Setting 視窗：EtherCAT

3. 若使用 CN3，需將電腦網路卡與控制器設定於相同的 IP 網域下，例如：192.168.0.XXX。  
若使用 CN4，IP 固定為 169.254.188.20。

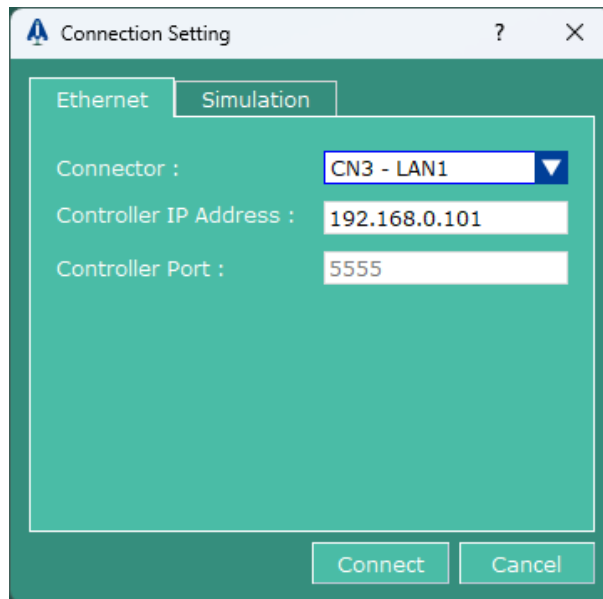


圖 1.1.1.3 CN3 - 自訂 IP 連線

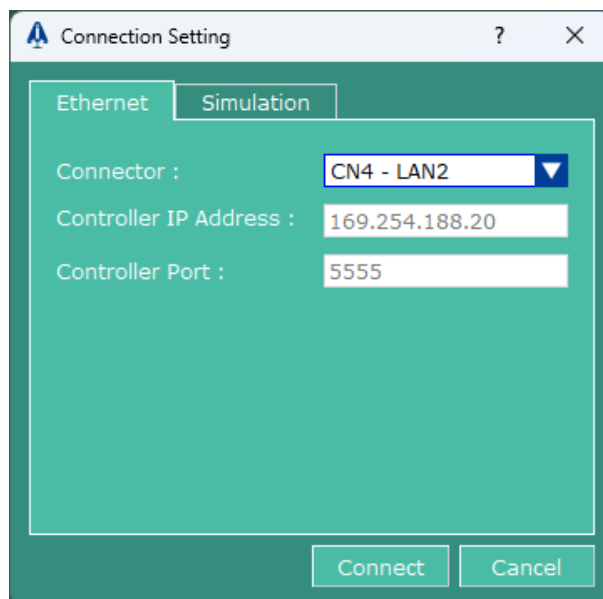


圖 1.1.1.4 CN4 - 固定 IP 連線

表 1.1.1.1 Connection Setting 視窗

通訊類型頁籤	說明
Ethernet	經由 TCP/IP 連線至控制器。
Simulation	連線至虛擬控制器。

## 1.1.2 經由 Ethernet 連線至控制器

可經由 Ethernet 連線至控制器，請參閱以下步驟建立連線。

1. 於 Connection Setting 視窗選擇 **Ethernet** 頁籤。
2. 輸入控制器 IP 位址及 IP 埠。
3. 點擊 **Connect** 按鈕與控制器建立連線，點擊後會出現進度條視窗。

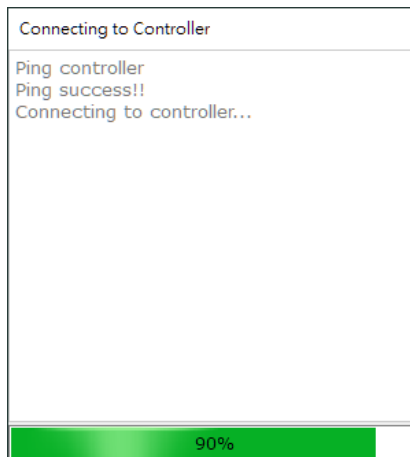


圖 1.1.2.1 進度條視窗

成功建立連線後，Connection Setting 視窗及進度條視窗會自動關閉。如無法建立連線，則會出現錯誤視窗，此時請檢查通訊線是否正確連接至控制器。

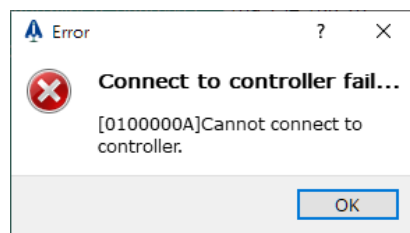


圖 1.1.2.2 無法連線至控制器

## 2. 參數設定

---

2.	參數設定.....	2-1
2.1	配置驅動器.....	2-2
2.2	安裝 ESI 檔.....	2-3
2.3	設定運動控制軸.....	2-5
2.4	配置 PDO.....	2-8

## 2.1 配置驅動器

1. 點擊選單列的 **Project**，再點擊 **Configuration Wizard**。

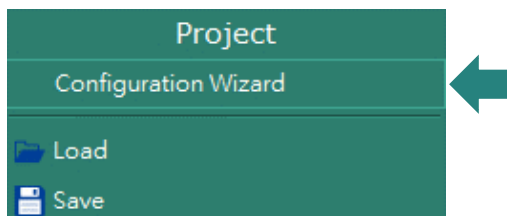


圖 2.1.1 Configuration Wizard

2. 點擊  掃描驅動器。

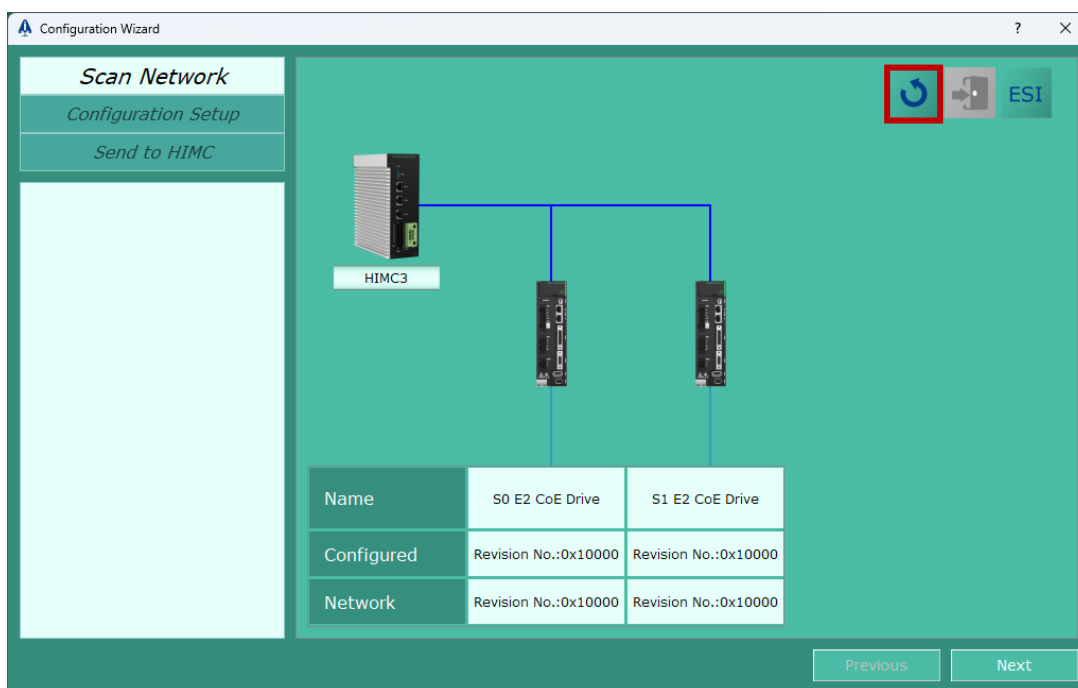


圖 2.1.2 掃描驅動器

## 2.2 安裝 ESI 檔



Important

- (1) E系列驅動器的ESI檔名：HIWIN\_MIKROSYSTEM\_EDxF\_yyyymmdd.xml
- (2) 可於驅動器Thunder軟體的安裝路徑（Thunder/doc/ESI Files）取得E系列驅動器的ESI檔。

1. 於 Configuration Wizard 視窗點擊 **ESI**。

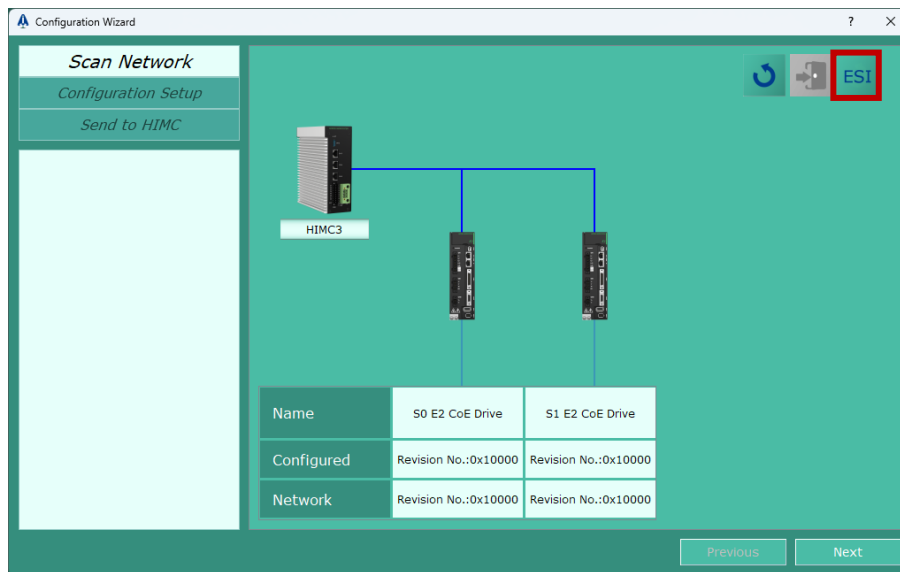


圖 2.2.1 設定 ESI 檔

2. 點擊 **Add** 新增對應韌體版本的 ESI 檔。

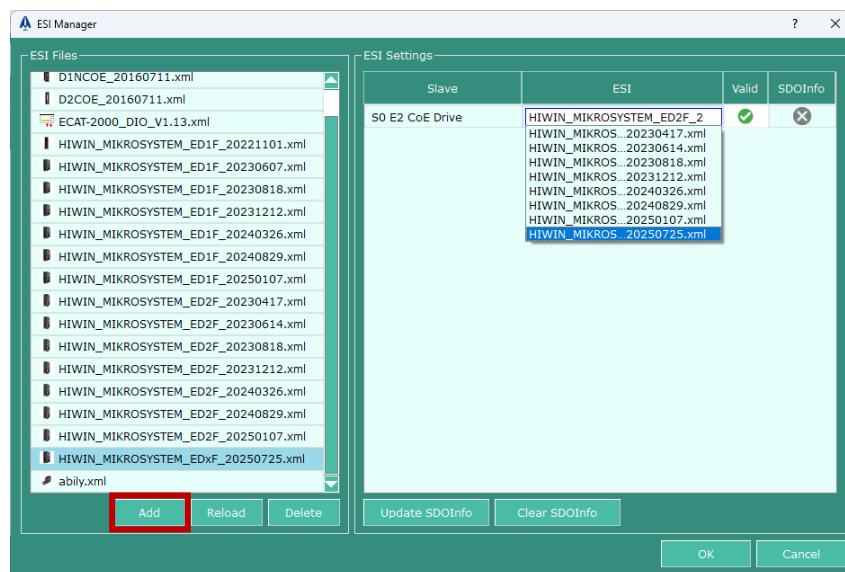


圖 2.2.2 新增 ESI 檔

3. 選擇欲載入的 ESI 檔 (.xml) · 再點擊 **Load** 。

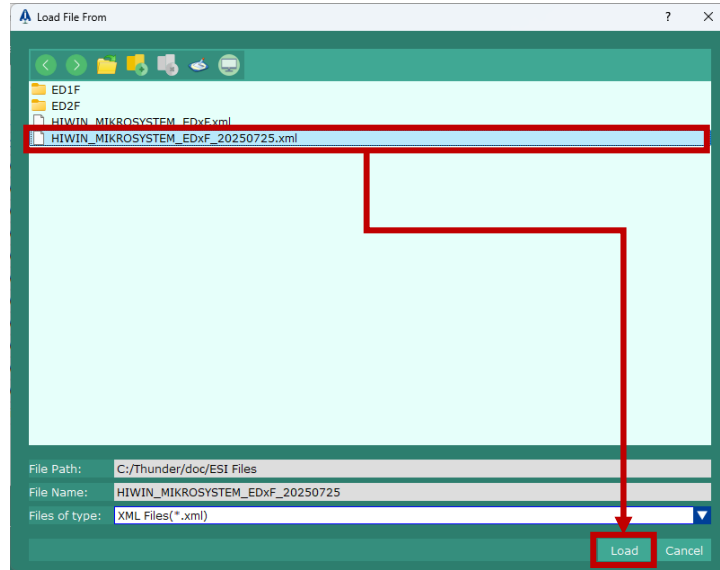


圖 2.2.3 載入 ESI 檔

4. 選擇所需的 ESI 檔 (.xml) 版本 · 再點擊 **OK** · 即完成。

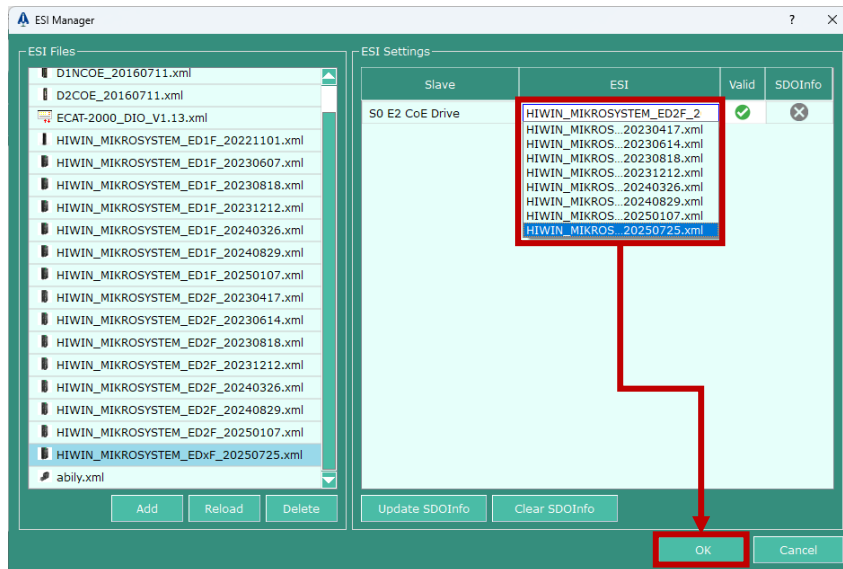


圖 2.2.4 選擇 ESI 檔

## 2.3 設定運動控制軸

1. 在 Configuration Setup 頁籤中點擊從軸前方的 。

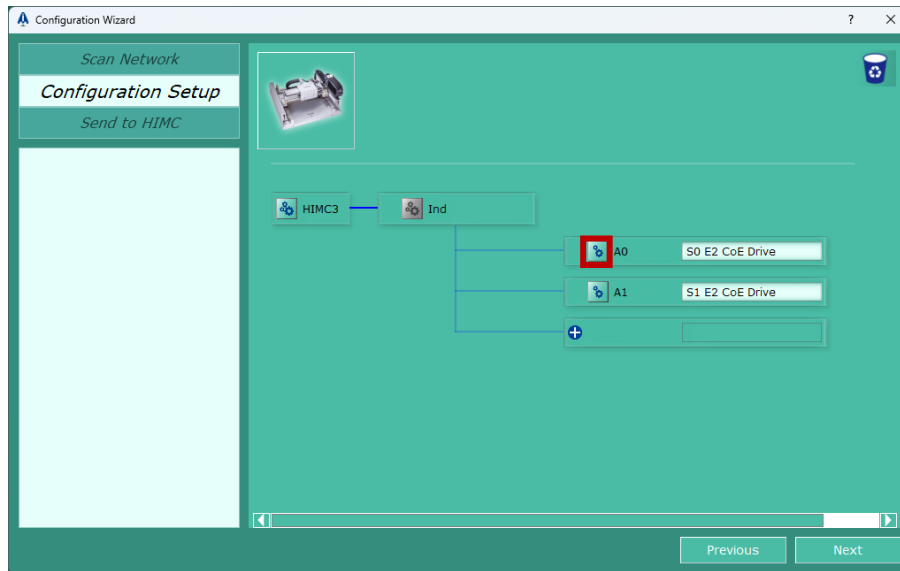


圖 2.3.1

2. 在 Axis 頁籤中設定軸相關參數，例如：操作模式、馬達類型、驅動器設定、安全設定等。

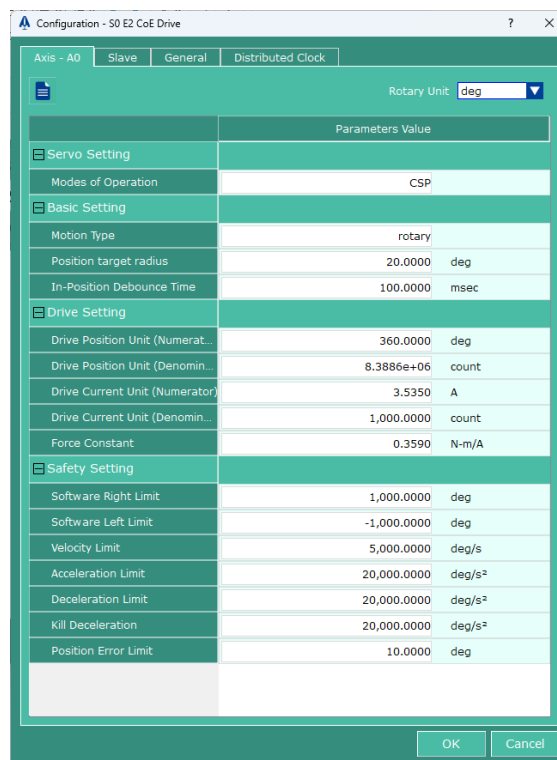


圖 2.3.2

### 3. 確認 Motion Type 是否正確。



Important

針對非旋轉直驅架構 ( 包含旋轉馬達與螺桿之組合應用 )，需透過驅動器中電子齒輪比設定螺桿進給常數，進而轉換成線性 ( Linear ) 控制單位。因此在控制器設定時，需要將Motion Type設定為線性 ( Linear )，以免發生控制異常。

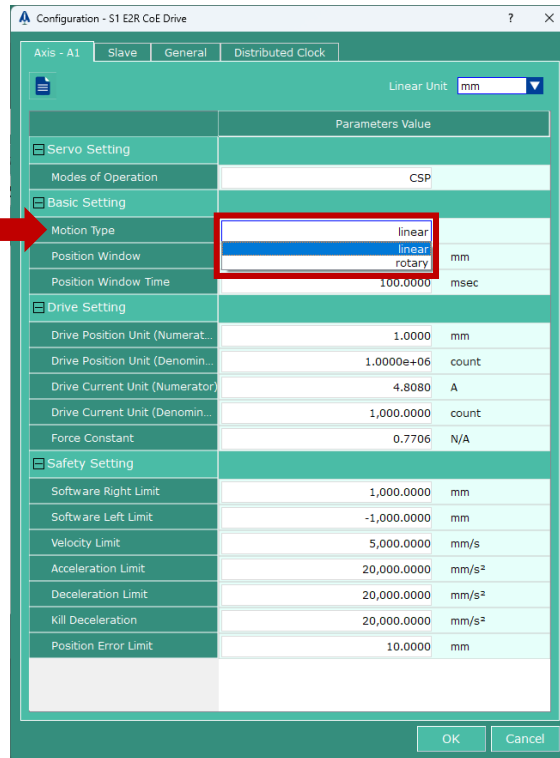


圖 2.3.3

### 4. 確認驅動器與控制器的控制單位是否相同。



Important

若設備機構的電子齒輪比已在驅動器中設定，僅需確認驅動器與控制器的控制單位是否相同。若設備機構在驅動器中的電子齒輪比設定為1:1，此時機構的電子齒輪比轉換需在控制器中設定。

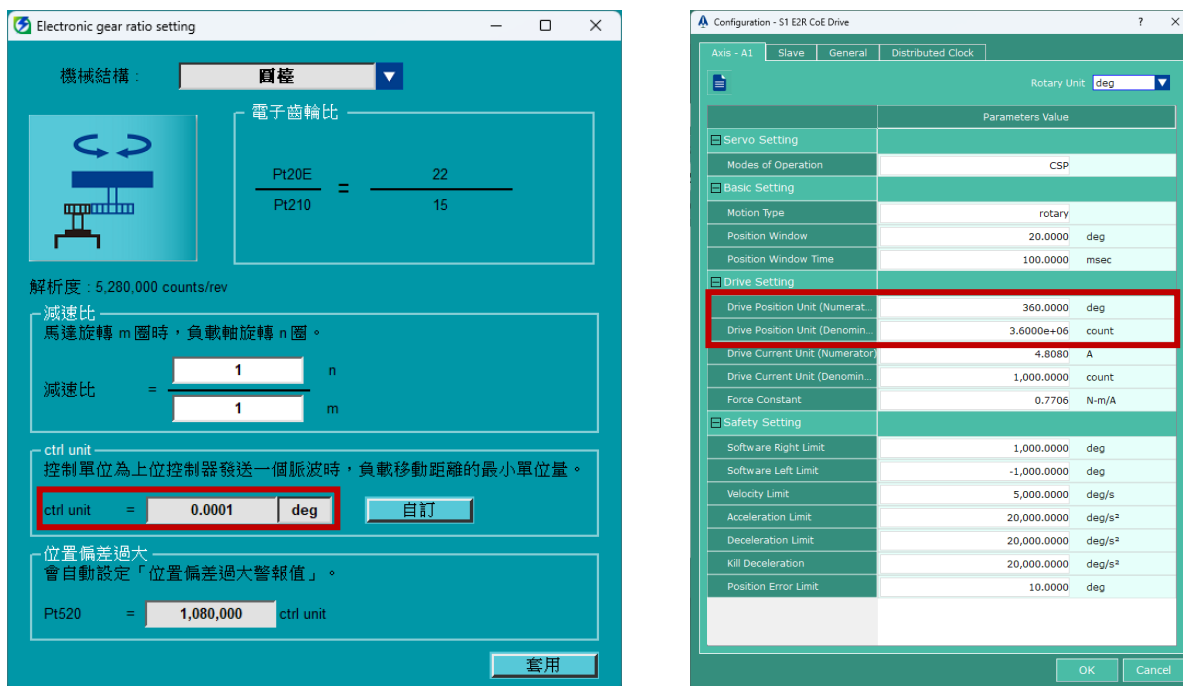


圖 2.3.4

5. 切換至 Send to HIMC 頁籤，確認修改的參數無誤後，點擊 **Send to HIMC** 完成設定。

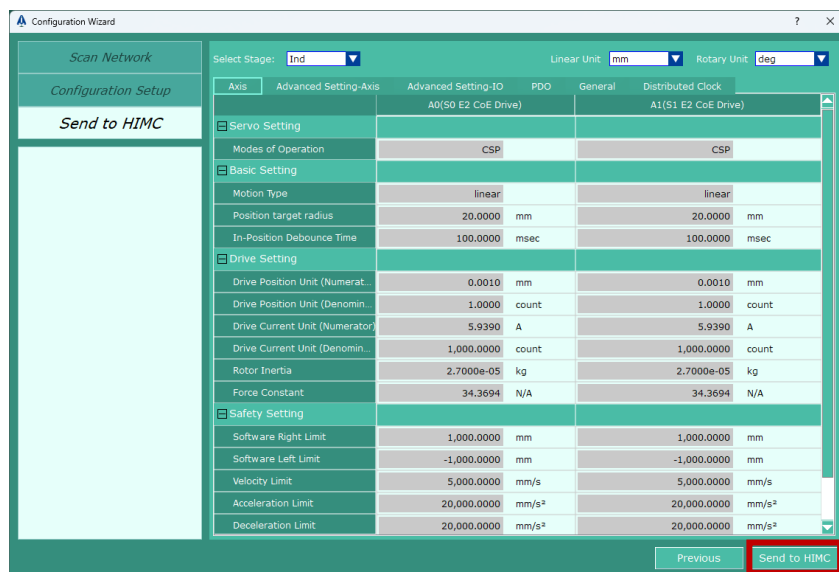



圖 2.3.5

## 2.4 配置 PDO



Important

- (1) 點擊任一PDO群組後，Edit PDO Map Settings視窗右側會顯示此群組預設的PDO物件。
- (2) 可點擊**Add PDO Entry**於群組加入其他的物件，或點擊**Delete PDO Entry**移除群組中已存在的物件。
- (3) E系列驅動器的RxPDO、TxPDO最大物件數目為各10個。

1. 切換至 Slave 頁籤後，點擊 PDO Mapping 的 。

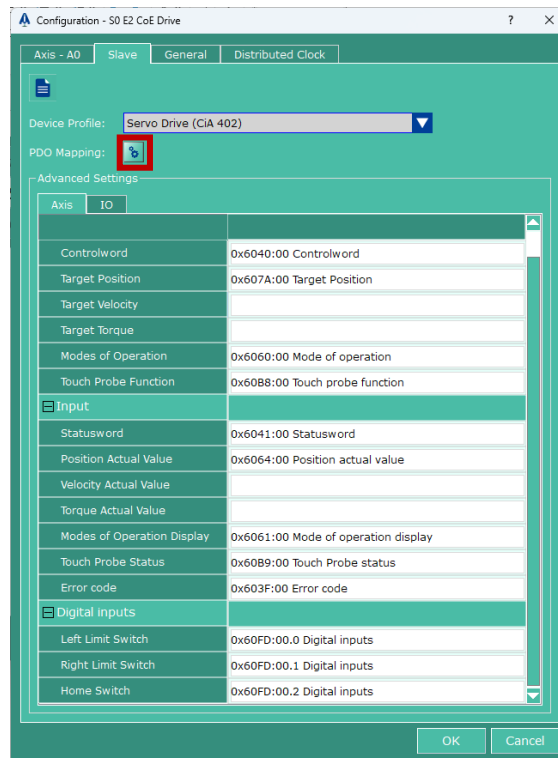


圖 2.4.1

2. 選擇欲使用的預設 PDO。若需調整，可點擊右上角的 。設定完成後點擊 **OK**。

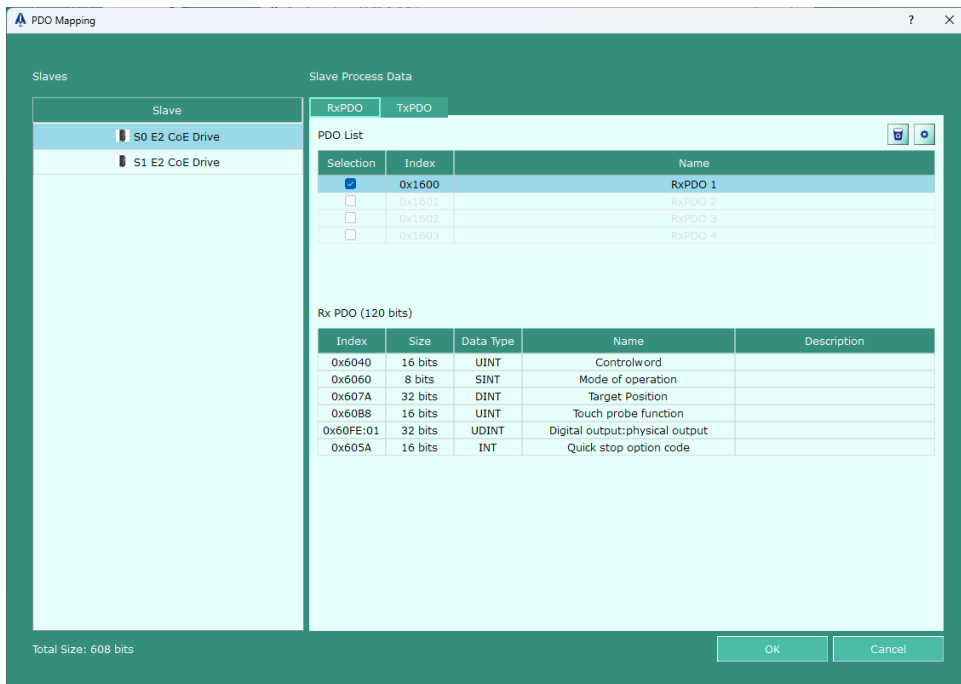


圖 2.4.2 選擇預設 PDO

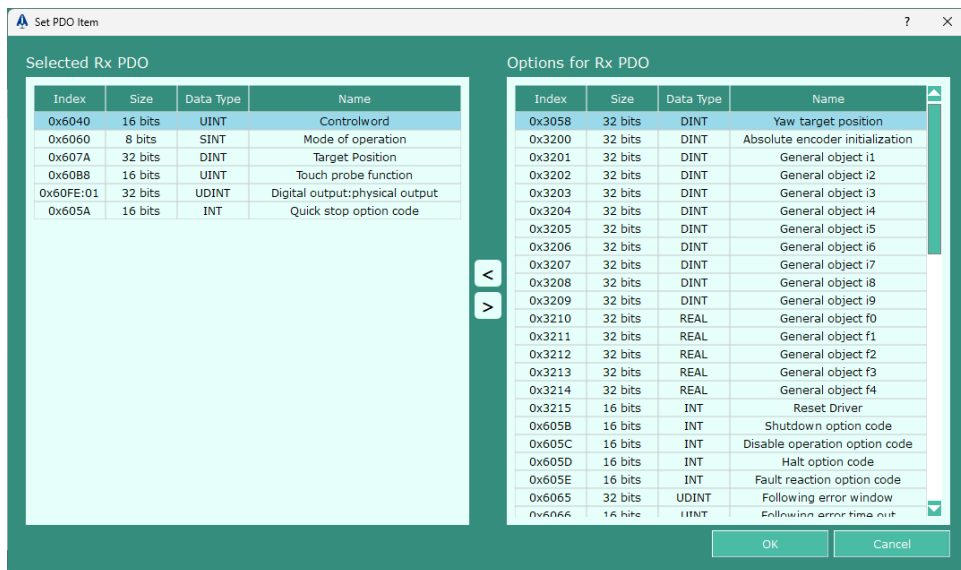


圖 2.4.3 選擇自訂 PDO

### 3. 最後在 Slave 頁籤中進行 PDO 配置。

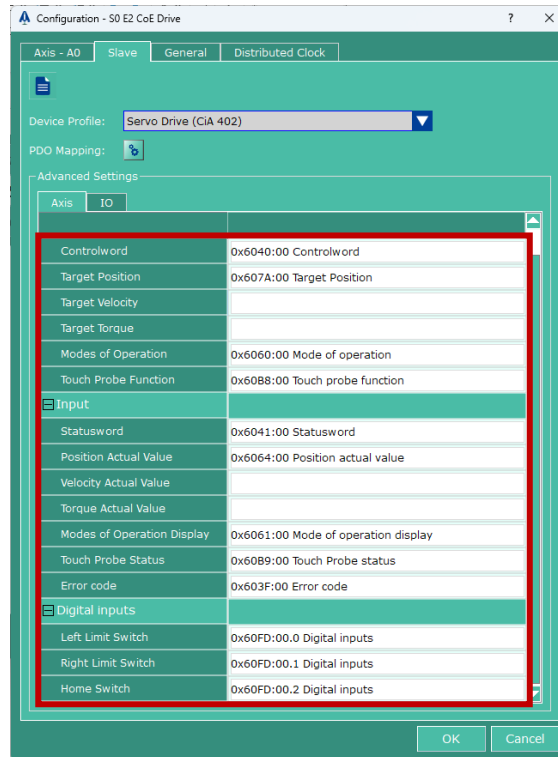


圖 2.4.4

## 3. 試運轉

3.	試運轉 .....	3-1
3.1	激磁 .....	3-2
3.2	歸原點 .....	3-3
3.3	吋動 .....	3-5
3.4	相對移動 .....	3-6
3.5	點對點運動 .....	3-7
3.6	HMPL Editor .....	3-8
3.6.1	點對點運動 .....	3-11
3.6.2	歸原點 .....	3-13

本章將介紹如何透過 iA Studio 的介面操作及 HMPL 簡易程式進行試運轉。



Important

本手冊僅介紹基本功能，其餘功能請參閱《[HIMC系列iA Studio軟體使用手冊](#)》與《[HIMC系列HMPL使用手冊](#)》。

## 3.1 激磁

1. 點擊選單列的 **Tools**，再點擊 **Motion Manager**。

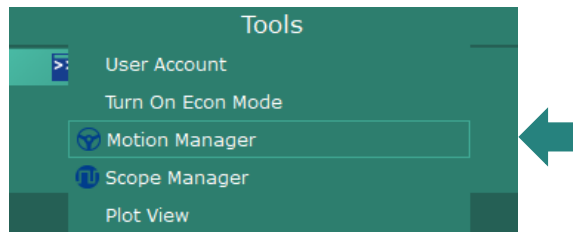


圖 3.1.1 Motion Manager

2. 於 Motion Manager 視窗點擊 **Enable/Disable**，即可激磁/解激磁馬達，同時可觀察 Enable 的燈號來確認是否執行成功。

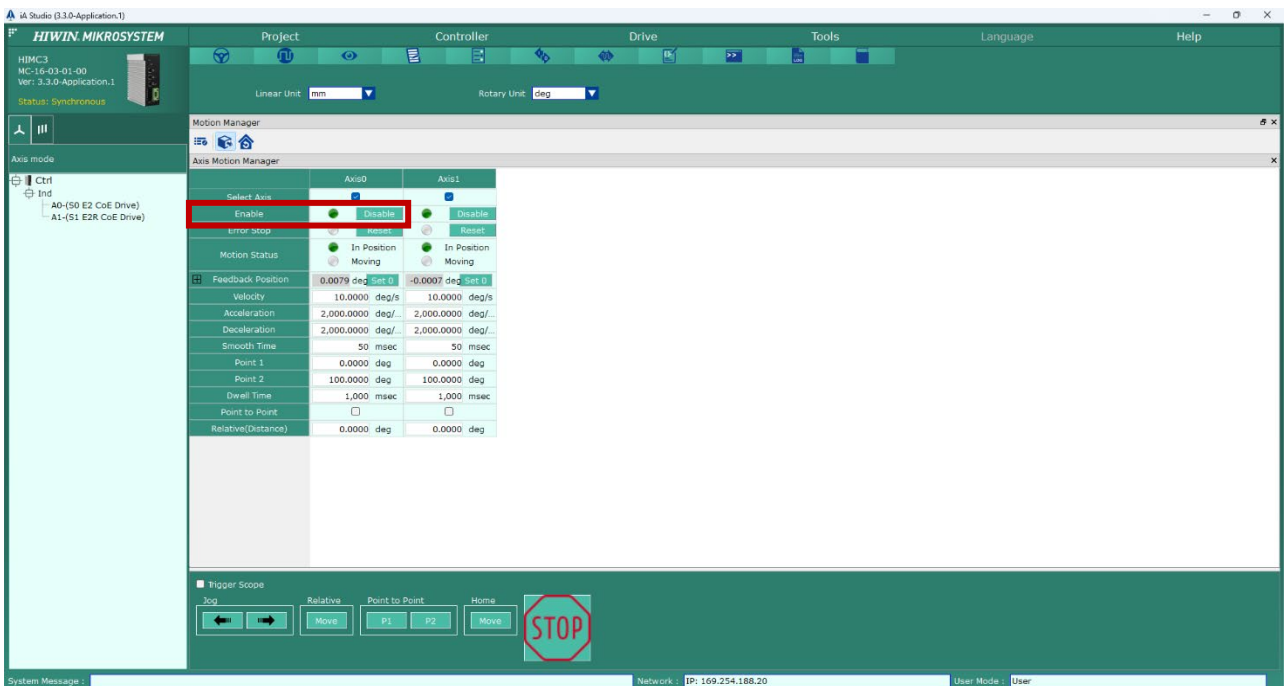



圖 3.1.2 Motion Manager 視窗 - 激磁/解激磁

### 3.2 歸原點

1. 於 Motion Manager 視窗點擊 ，切換至歸原點功能視窗。

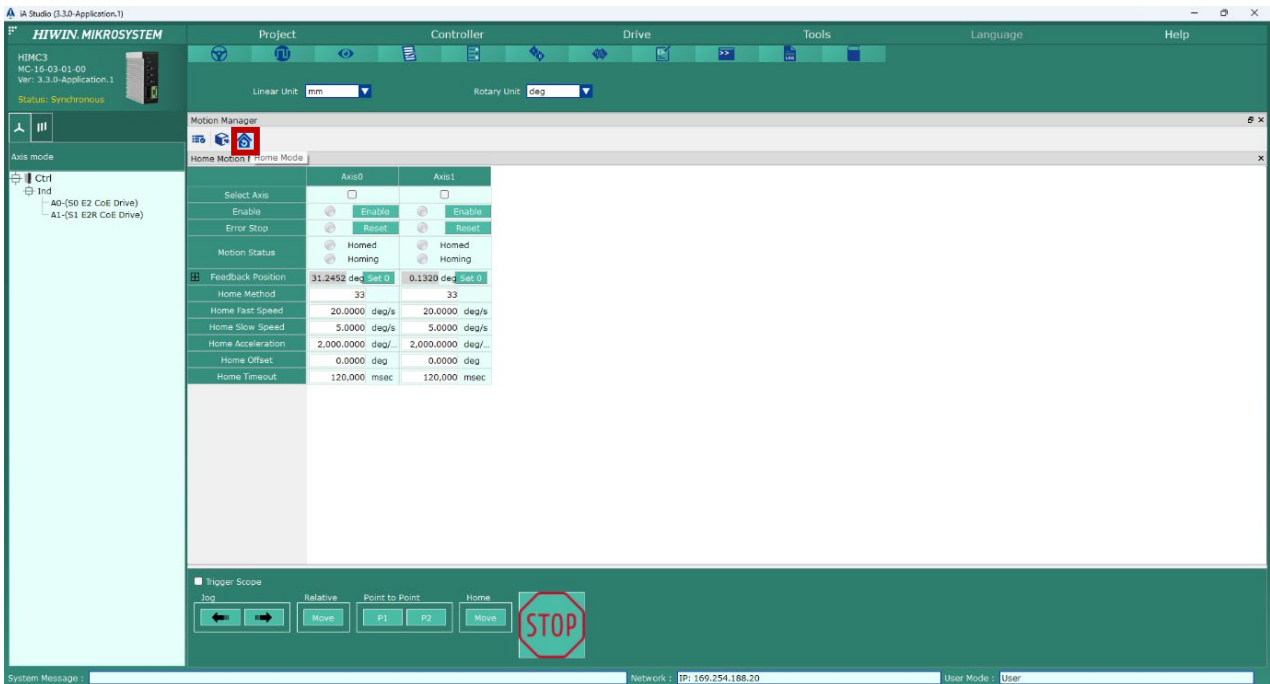


圖 3.2.1 Motion Manager 視窗 - 歸原點

2. 根據應用所需的歸原點條件進行歸原點方法、速度、加速度、偏移量等參數設定。

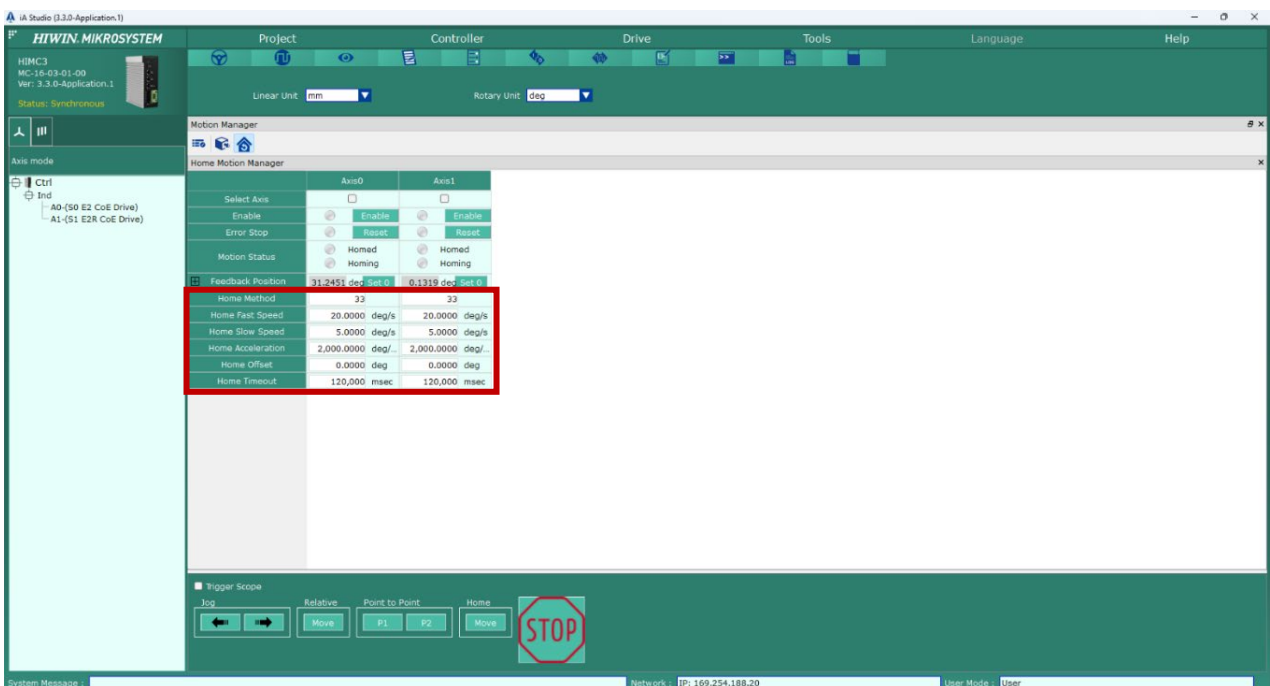


圖 3.2.2 歸原點條件設定

3. 勾選欲執行歸原點的軸，點擊 **Enable** 後，再點擊 Home 的 **Move**。

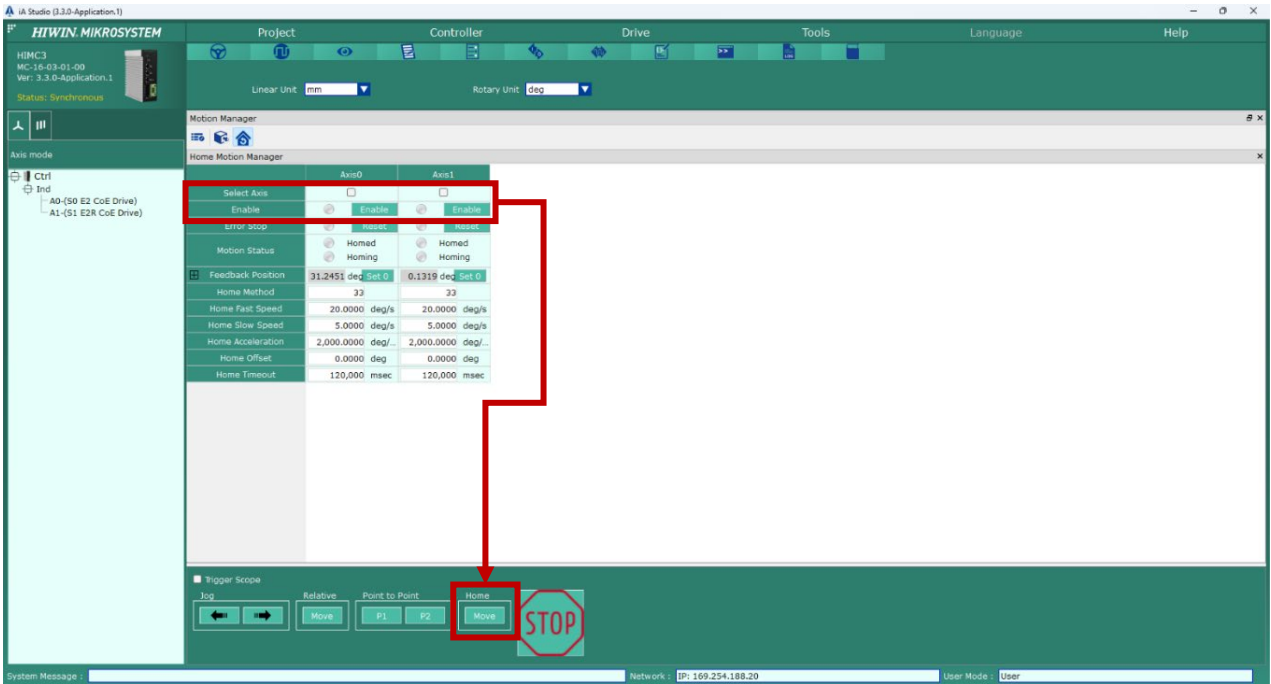


圖 3.2.3 啟動歸原點

4. Homed 亮綠燈，即完成歸原點。

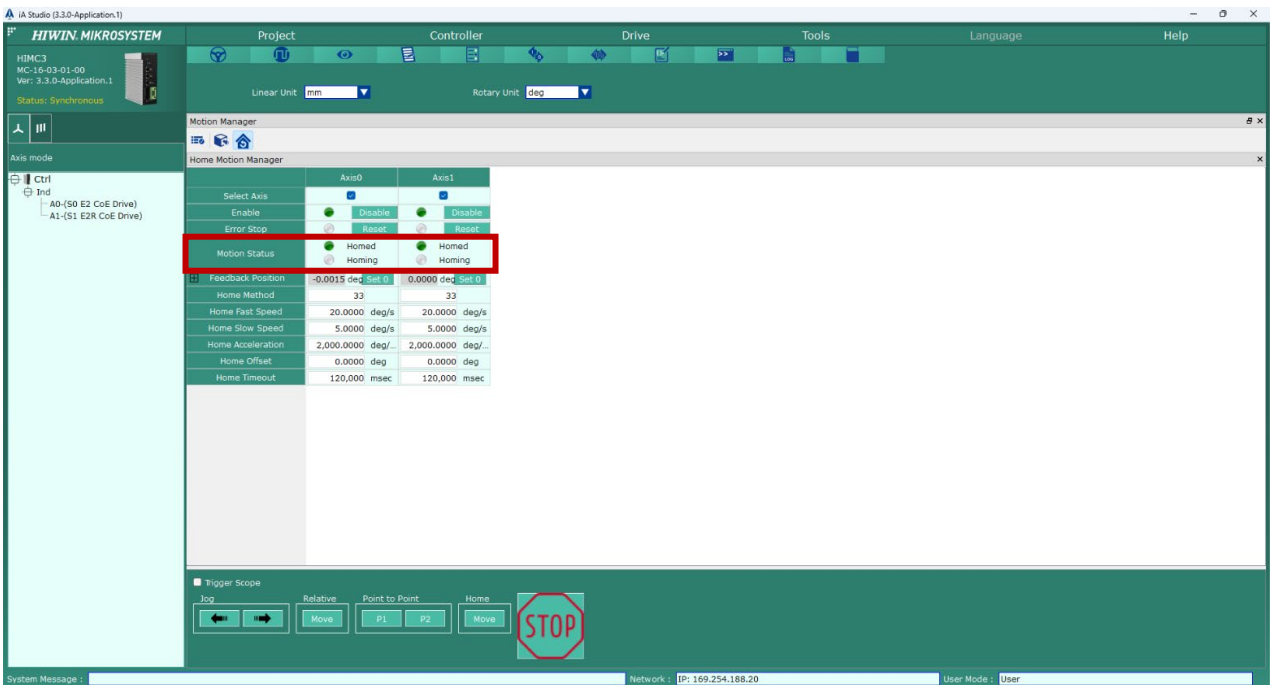


圖 3.2.4 歸原點完成

### 3.3 吋動

於 Motion Manager 視窗點擊 Jog 的箭頭，即可進行正向或反向吋動。

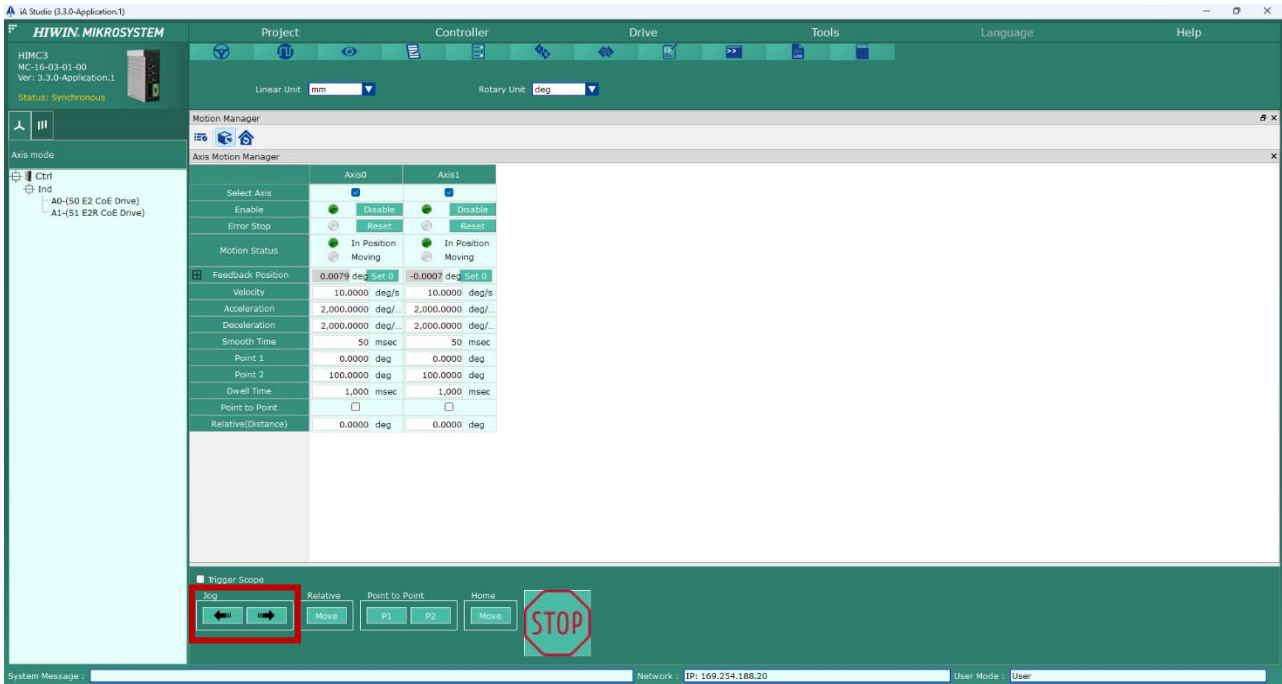


圖 3.3.1 Motion Manager 視窗 - 吋動

## 3.4 相對移動

於 Motion Manager 視窗設定 **Relative(Distance)**，再點擊 Relative 的 **Move**，即可進行相對移動。

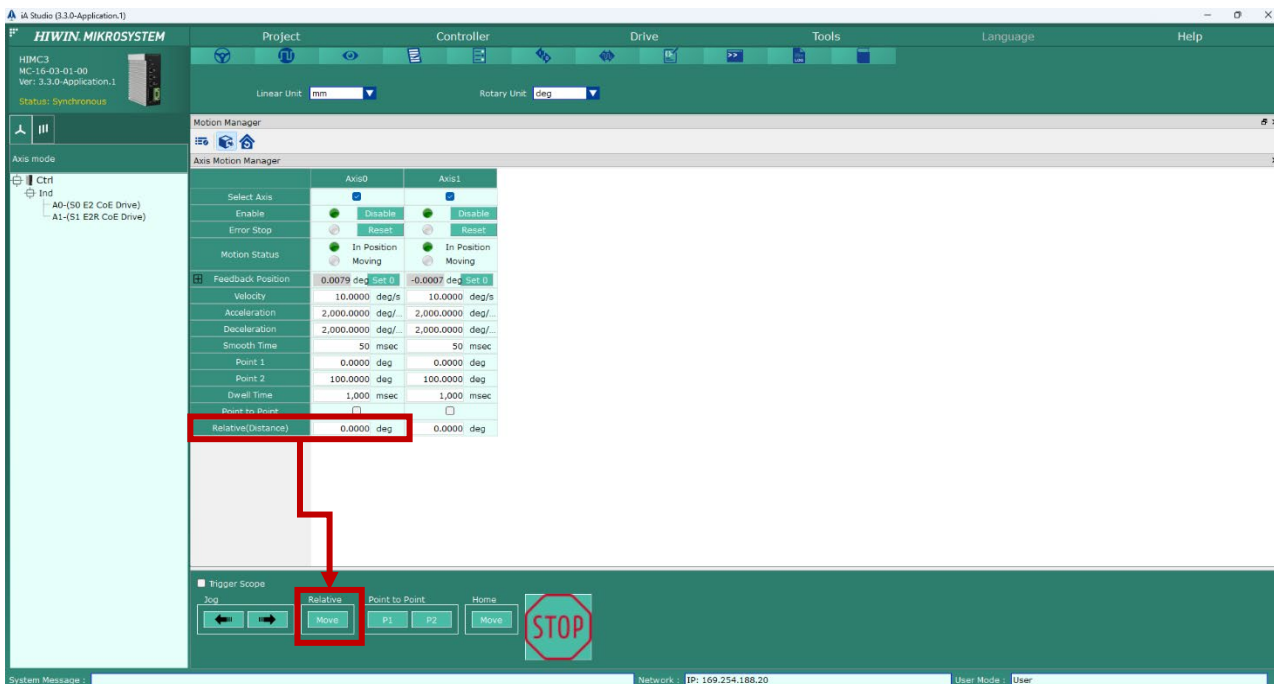


圖 3.4.1 Motion Manager 視窗 - 相對移動

### 3.5 點對點運動

1. 於 Motion Manager 視窗設定 Point 1、Point 2、Dwell Time。
2. 點擊 Point to Point 的 P1 或 P2，即可將馬達移動到 Point 1 或 Point 2 的位置。
3. 勾選 Point to Point，再點擊 P1 或 P2，即可使馬達在 Point 1 及 Point 2 的位置間重複運動。

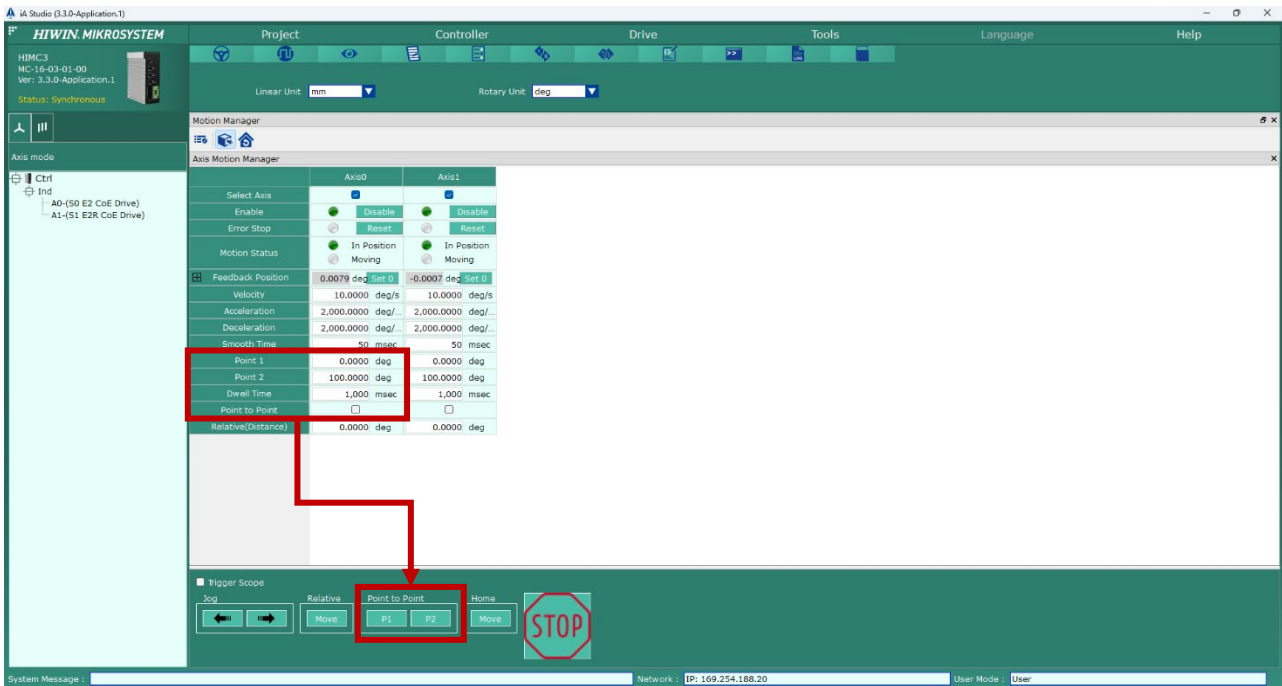


圖 3.5.1 Motion Manager 視窗 - 點對點運動

## 3.6 HMPL Editor

1. 點擊選單列的 **Tools**，再點擊 **HMPL Editor**，開啟 HMPL 編輯視窗。

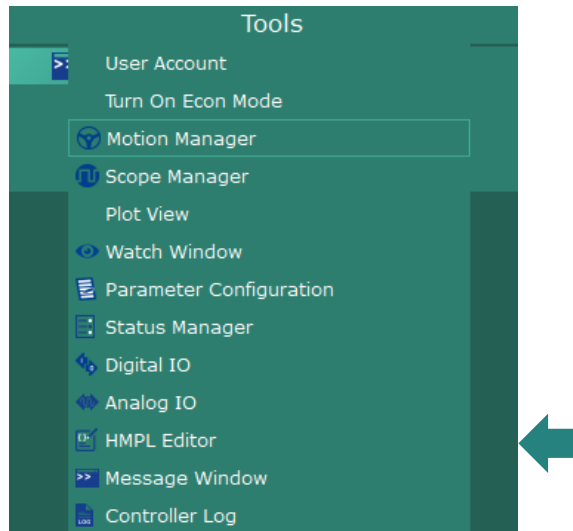


圖 3.6.1 HMPL Editor

2. 雙擊 task 即可開啟程式編輯頁面。

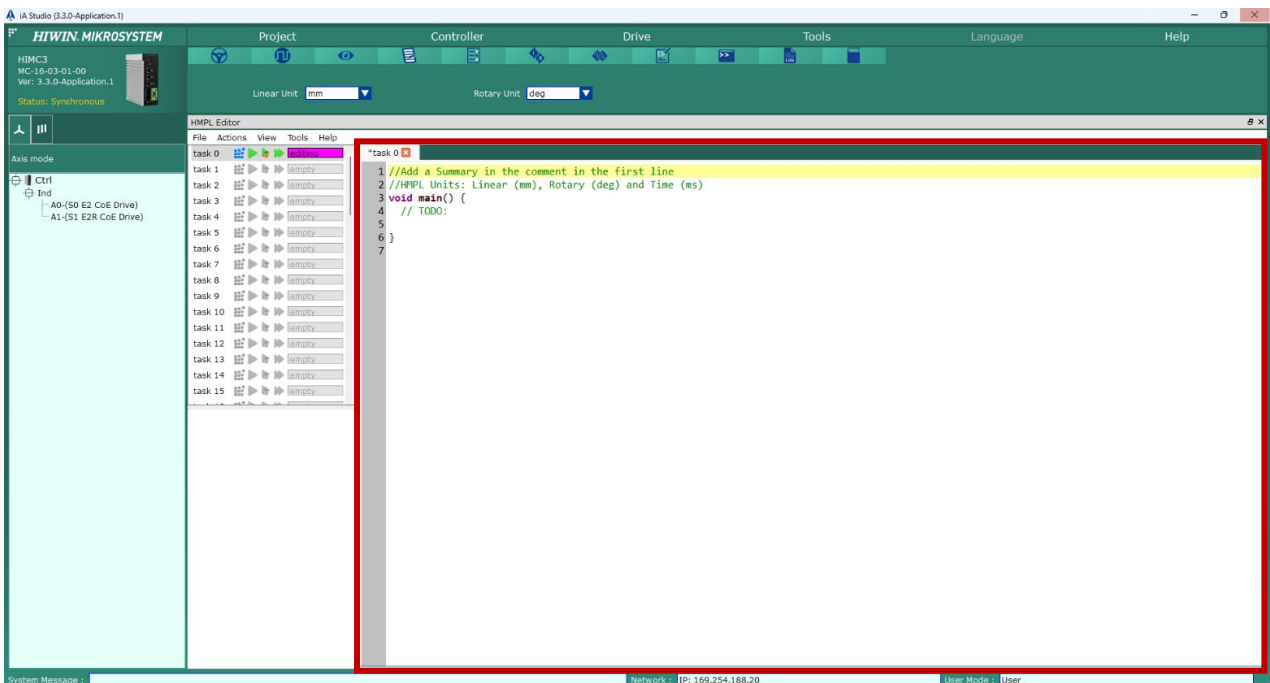



圖 3.6.2 程式編輯

3. 點擊  (F7) 將編寫完的程式進行編譯。

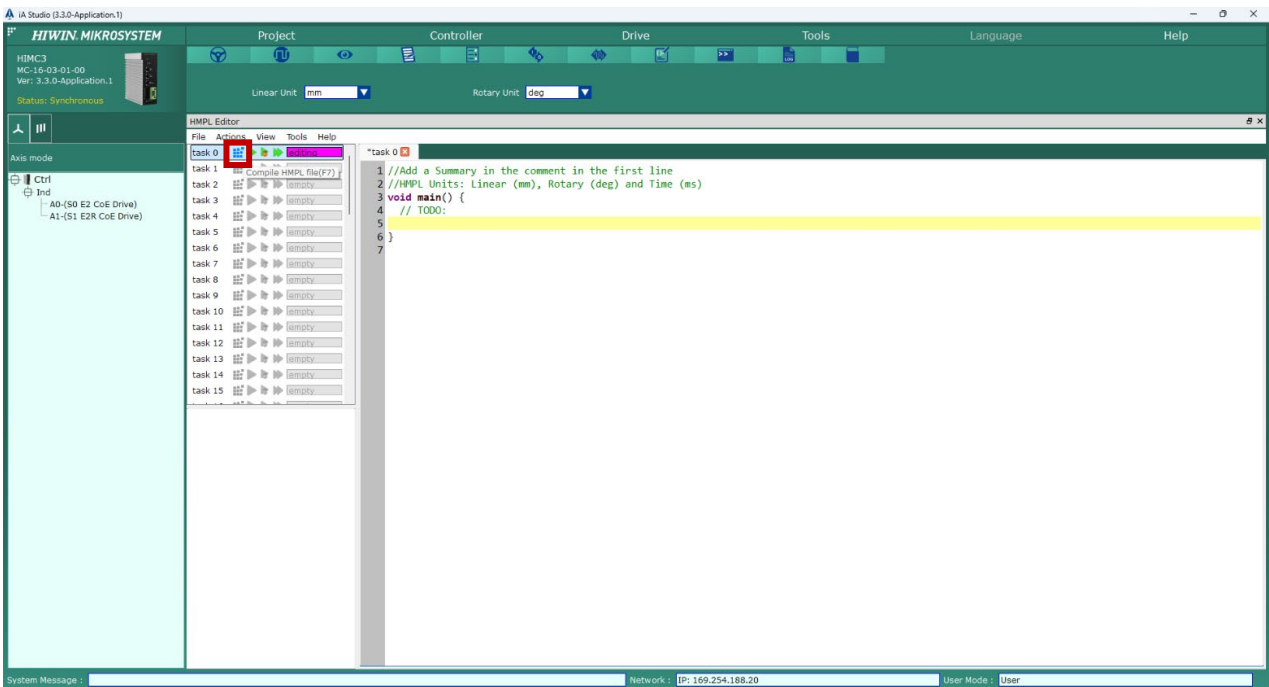



圖 3.6.3 程式編譯

4. 點擊  將 task 儲存至控制器。

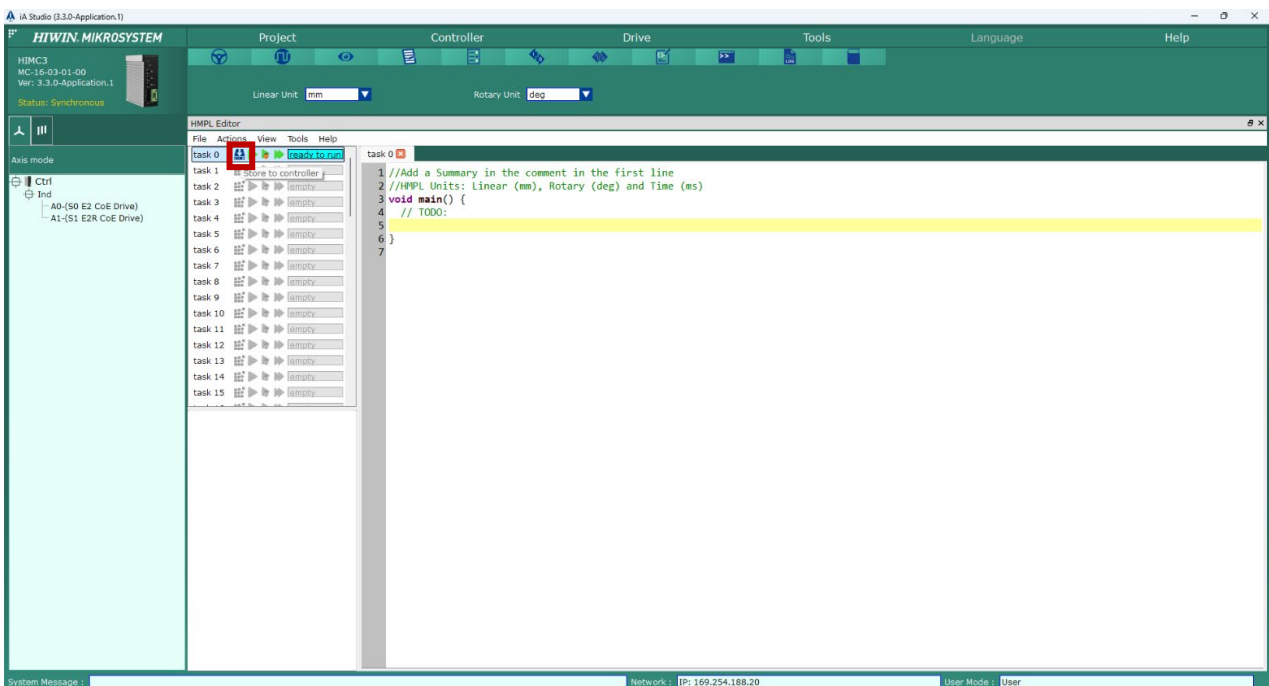



圖 3.6.4 程式寫入控制器

5. 點擊  (Ctrl+F5) 執行 task 中的程式。

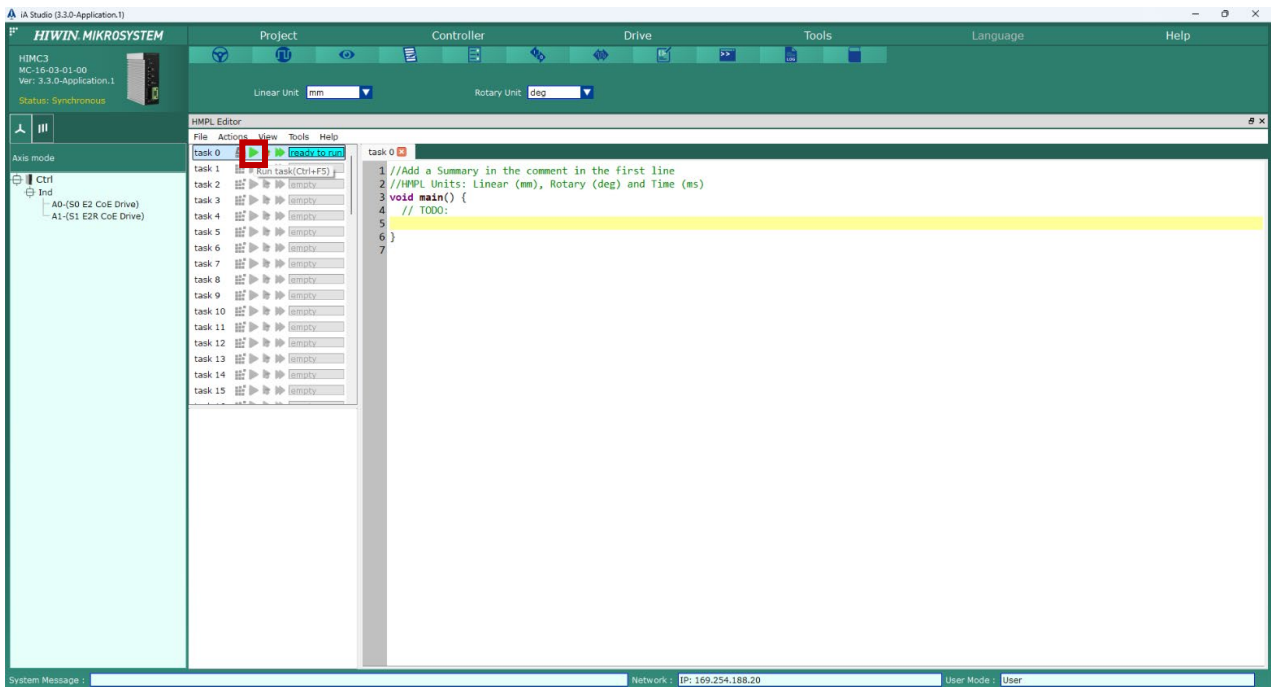


圖 3.6.5 程式執行

### 3.6.1 點對點運動

1. 開啟新的 task。
2. 在 task 中寫入點到點程式 ( 設定 P1/P2 位置、速度、加減速度等 )。
3. 將程式編譯後並寫入控制器中。
4. 執行程式。

```
int axis_x = 0; // 軸編號
int P1 = 0; // P1位置 單位：線性(mm)、旋轉(deg)
int P2 = 30; // P2位置 單位：線性(mm)、旋轉(deg)
int Dwell_Time = 1000; // 休息時間(ms)

void main()
{
    SetVel(axis_x, 50); // 設定速度 單位：線性(mm/s)、旋轉(deg/s)
    SetAcc(axis_x, 2000); // 設定加速度 單位：線性(mm/s^2)、旋轉(deg/s^2)
    SetDec(axis_x, 2000); // 設定減速度 單位：線性(mm/s^2)、旋轉(deg/s^2)

    Enable(axis_x); // 激磁軸
    Till(IsEnabled(axis_x)); // 等待激磁完成
    Sleep(500);

    while(IsEnabled(axis_x)){
        MoveAbs(axis_x, P1); // 移動至 P1
        Till(IsInPos(axis_x)); // 等待到位
        Sleep(Dwell_Time); // 休息時間
        MoveAbs(axis_x, P2); // 移動至 P2
        Till(IsInPos(axis_x)); // 等待到位
        Sleep(Dwell_Time); // 休息時間
    }
}
```

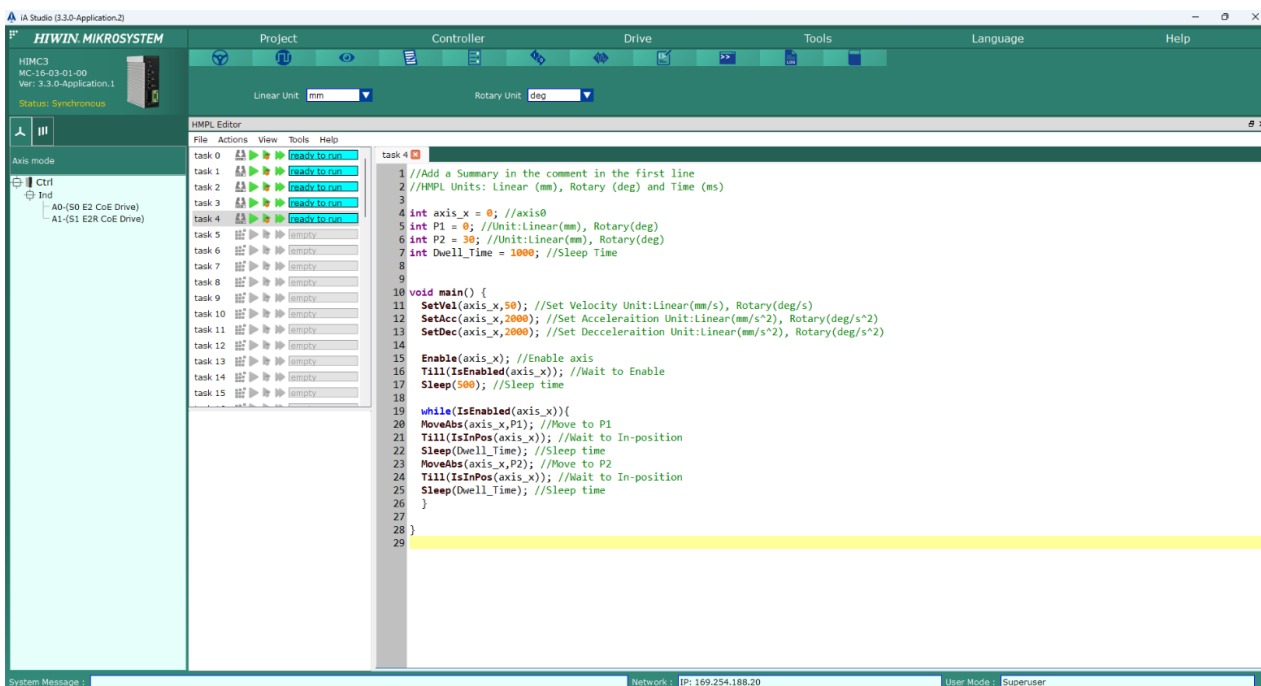


圖 3.6.1.1 點對點程式寫入

## 3.6.2 歸原點



Important

可至iA Studio的安裝路徑載入範例程式：

C:\Program Files (x86)\HIWIN MIKROSYSTEM\iA Studio (版本)\examples\HMPL

1. 開啟新的 task。
2. 在 task 中寫入歸原點程式 ( 設定歸原點方法、速度等 )。
3. 將程式編譯後並寫入控制器中。
4. 執行程式。

```
void main()
{
    int axis_id = 0; // 軸編號
    int home_method = 33; // 歸原點方法
    double fast_vel = 20; // 歸原點速度 (搜尋 Limit Switch) 單位：線性(mm/s)、旋轉(deg/s)
    double slow_vel = 2; // 歸原點速度 (搜尋 Index) 單位：線性(mm/s)、旋轉(deg/s)
    double acc = 2000; // 歸原點加速度 單位：線性(mm/s^2)、旋轉(deg/s^2)
    double home_offsets = 0; // 原點偏移量 單位：線性(mm)、旋轉(deg)
    int time_out = 10000; // 逾時時間 (ms)

    Enable(axis_id);
    Till(IsEnabled(axis_id));

    SetHomedStatus(axis_id, false); // 設定歸原點狀態
    SetHomeMethod(axis_id, home_method); // 設定歸原點方法
    SetHomeSwitchVel(axis_id, fast_vel); // 設定歸原點速度 (搜尋Limit Switch)
    SetHomeZeroVel(axis_id, slow_vel); // 設定歸原點速度 (搜尋Index)
    SetHomeAcc(axis_id, acc); // 設定歸原點加速度
    SetHomeOffset(axis_id, home_offsets); // 設定原點偏移量
    SetHomeTimeout(axis_id, time_out); // 設定逾時時間
    int result = MoveHome(axis_id);
    Till(!IsHoming(axis_id)); // 等待歸原點完成

    if (!IsHoming(axis_id)) {
        Print("Home success.");
    } else {
```

```
Print("Home fail:%d.", result);  
}  
}
```

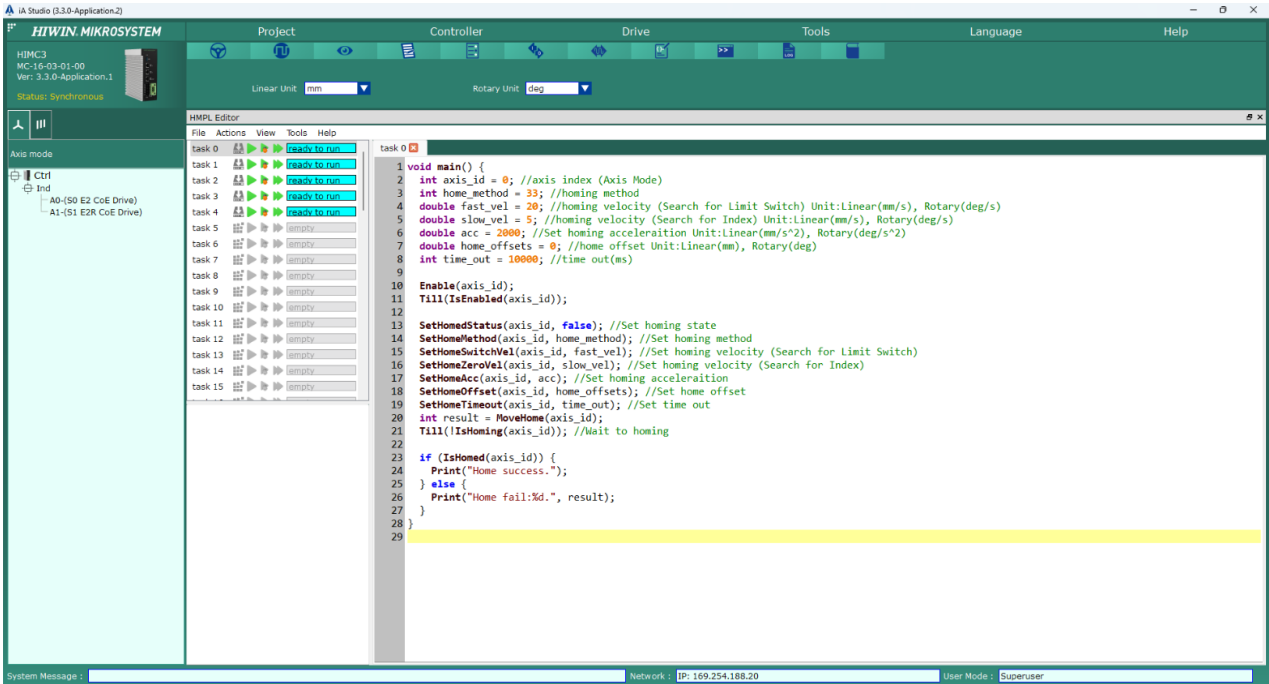


圖 3.6.2.1 歸原點程式寫入

## 4. 其他應用設定

---

4.	其他應用設定 .....	4-1
4.1	誤差補償功能設定 .....	4-2
4.1.1	1D 誤差補償 .....	4-2
4.1.2	2D 誤差補償 .....	4-3

## 4.1 誤差補償功能設定

因機構在安裝及移動時會產生公差，使用者若想提升精度，可透過雷射量測儀、千分表、校正片等量測工具取得誤差值並進行誤差補償。關於誤差補償功能、誤差補償函式、範例程式等詳細說明，請參閱《[HIMC 系列 HMPL 使用手冊](#)》第 14 章動態誤差補償函式。

### 4.1.1 1D 誤差補償

1D 誤差補償為參考一個軸進行單軸的補償，以下範例為參考 X 軸對 Y 軸進行補償，起始補償位置為-100、間距為 200、補償數量 5 個。依序的補償值為-0.2、0.1、-0.3、0.4、-0.1。詳細程式範例請參閱《[HIMC 系列 HMPL 使用手冊](#)》第 14 章動態誤差補償函式的範例 1：一維動態誤差補償。

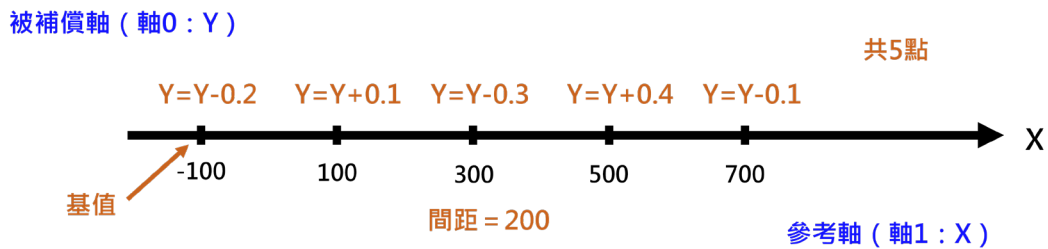


圖 4.1.1.1 1D 誤差補償位置示意圖

以下使用 HMPL 設定並啟用補償：

```
void main() {
    // 設定補償表
    double data[5] = {-0.2, 0.1, -0.3, 0.4, -0.1};
    SetUserTable(1, 5, data); // 將數據寫入 User Table

    SetupComp(
        0, // 被補償軸
        1, // User Table 中，補償點的起始編號
        -100, // 起始位置
        200, // 間距
        5, // 補償點的數量 (含起始位置)
        1 // 參考軸 (輸入)
    );
}
```

```

EnableComp(0); // 啟用軸 0 的補償
Enable(0); // 激磁軸 0 以啟用補償
}
    
```

以下使用 HMPL 關閉補償功能：

```

void main() {
    Disable(0); // 解激磁軸 0
    Till(!IsEnabled(0));
    DisableComp(0); // 取消對軸 0 的補償
}
    
```

### 4.1.2 2D 誤差補償

2D 誤差補償為同時參考兩個軸進行補償，常見於 XYZ 平面中，同時參考 X 軸和 Y 軸進行 X 軸、Y 軸或 Z 軸的補償。以下範例為同時參考 X 軸和 Y 軸進行 Y 軸的補償，詳細程式範例請參閱《[HIMC 系列 HMPL 使用手冊](#)》第 14 章動態誤差補償函式的範例 2：二維動態誤差補償。



Important

控制器內部僅有一個表格，因此兩軸在補償時會共用同一個表格。在建立補償值時需注意各軸寫入表格的起始位置及數量，避免數值被相互覆蓋。

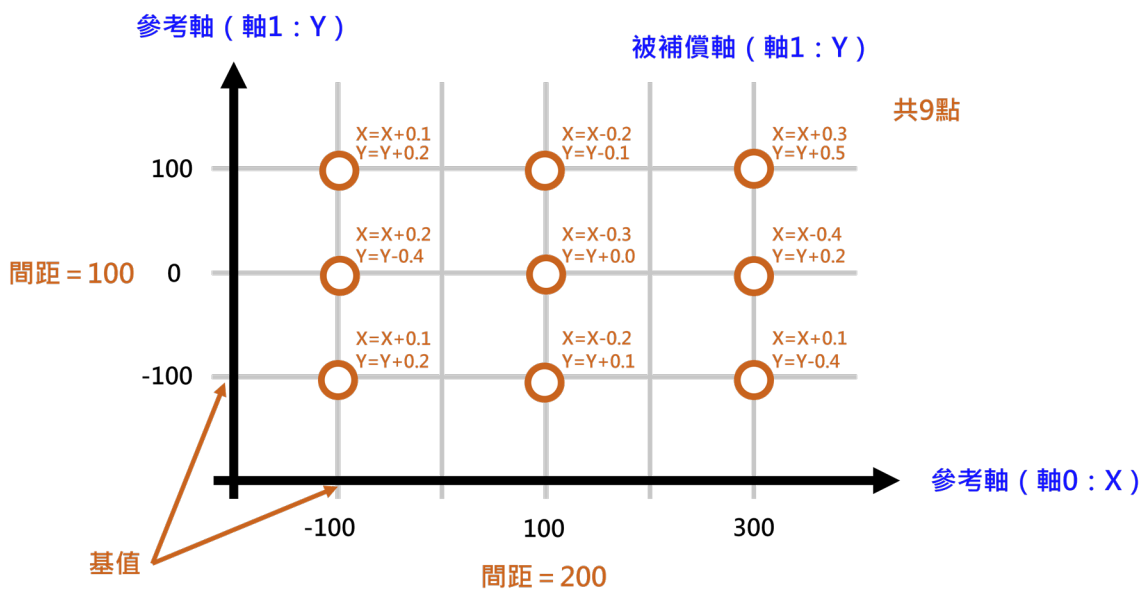


圖 4.1.2.1 2D 誤差補償位置示意圖

以下使用 HMPL 設定並啟用補償：

```
void main() {
    // 設定補償表
    double datay[9] = {0.2, 0.1, -0.4, -0.4, 0.0, 0.2, 0.2, -0.1, 0.5}; // 補償值 (左到
    右·下到上)
    double datax[9] = {0.1, -0.2, 0.1, 0.2, -0.3, -0.4, 0.1, -0.2, 0.3}; // 補償值 (左到
    右·下到上)
    SetUserTable(1, 9, datay); // 將數據寫入 User Table
    SetUserTable(10, 9, datax); // 將數據寫入 User Table

    double base[2] = {-100, -100};
    double interval[2] = {200, 100};
    int num_pt[2] = {3, 3};
    int ref_axis[2] = {0, 1};
    SetupComp2D(
        1, // 被補償軸 (軸 1 : Y)
        1, // User Table 中·補償點的起始編號
        base, // 起始位置
        interval, // 間距
        num_pt, // 補償點的數量 (含起始位置)
        ref_axis // 參考軸 (輸入)
    );

    SetupComp2D(
        0, // 被補償軸 (軸 0 : X)
        10, // User Table 中·補償點的起始編號
        base, // 起始位置
        interval, // 間距
        num_pt, // 補償點的數量 (含起始位置)
        ref_axis // 參考軸 (輸入)
    );

    EnableComp(0); // 啟用 (軸 0 : X) 的補償
    Enable(0); // 激磁 (軸 0 : X) 以啟用補償
    EnableComp(1); // 啟用 (軸 1 : Y) 的補償
    Enable(1); // 激磁 (軸 1 : Y) 以啟用補償
}
```

以下使用 HMPL 關閉補償功能：

```
void main() {
    Disable(1); // 解激磁軸 1
    Till(!IsEnabled(1));
    DisableComp(1); // 取消對軸 1 的補償
    Disable(0); // 解激磁軸 0
    Till(!IsEnabled(0));
    DisableComp(0); // 取消對軸 0 的補償
}
```

欲確認補償的數值是否正確，可在 Scope Manager 中監測變數 hcv.axis[0].pos\_fb\_comp ( 補償後的位置 ) 與 hcv.axis[0].pos\_fb ( 原始編碼器位置 )。



Important

- (1) 補償後的位置 = 原始編碼器位置 + 補償值
- (2) 修改[]內的數字即可監測不同軸。



圖 4.1.2.1 編碼器回授及補償後數值監測