

WEINTEK IIOT LTD.

# 定位控制 絕對/相對移動

FB

MC\_MoveAbsolute/MC\_MoveRelative

功能

工程檔案範例

## 目錄

1. 簡介與系統運作.....	1
2. 安裝 Weintek Library .....	3
3. CODESYS 專案加入 iR-PU01-P.....	4
4. iR-PU01-P 參數設定 .....	9
5. 功能塊介紹.....	12
6. 範例程序.....	16
7. 登入操作.....	19

## 1. 簡介與系統運作

建議使用 Weintek 官網的 CODESYS IDE 版本或以上版本開啟專案。

### 簡介

以下範例介紹如何使用 Weintek Library 功能塊及 iR-PU01-P，控制伺服或步進馬達進行定位控制操作。

使用 CODESYS 控制 iR-PU01-P 的脈波輸出信號，伺服或步進馬達的驅動器依據接收到 iR-PU01-P 送出的脈波信號，決定馬達轉動的距離與速度，達到定位或速度控制的應用。

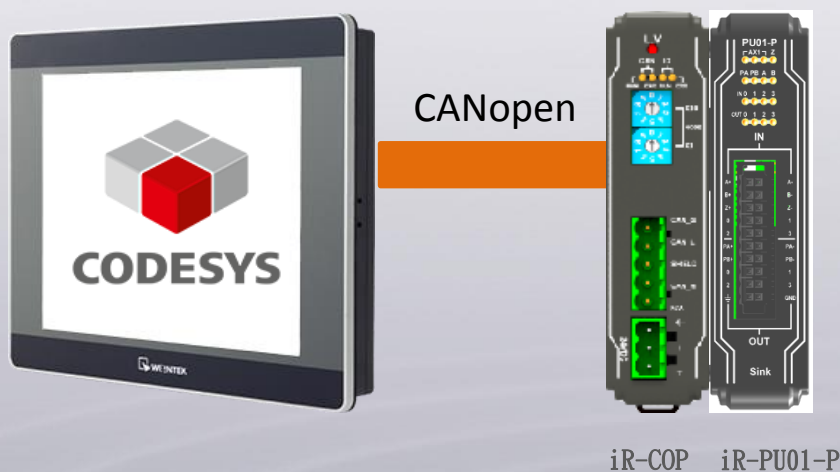
※ 使用 iR-PU01-P V1001 版本，參考檔案

iR\_Application\_Positioning\_Demo\_20190906

※ cMT-CTRL01 參考檔案 iR\_Application\_Positioning  
\_Demo\_CTRL\_20230523

※ cMT-HMI 參考檔案 iR\_Application\_Positioning \_Demo\_HMI\_20230523

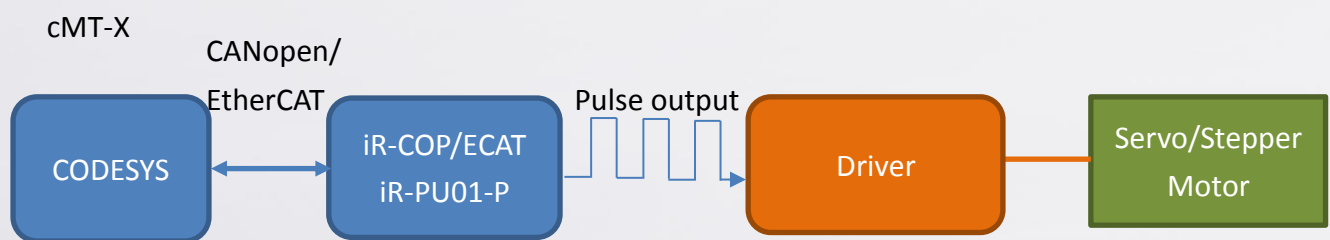
### 系統



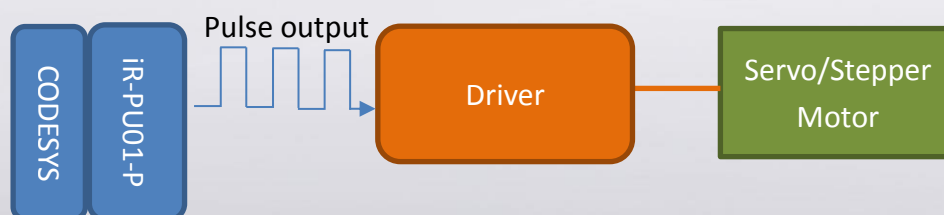


cMT-CTRL01 iR-PU01-P

定位功能信號流程圖：



cMT-CTRL01



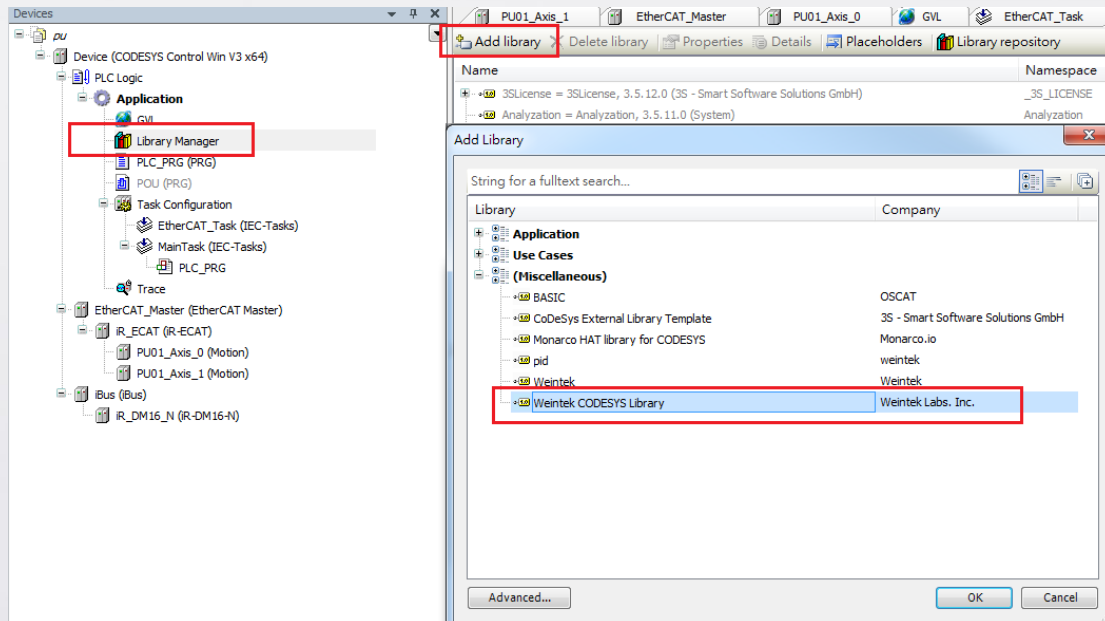
## 2. 安裝 Weintek Library

1. 開啟威綸官網下載頁面，搜尋 [cMT+CODESYS Package] 下載並安裝。

<https://www.weintek.com/globalw/Download/Download.aspx>

(此包括 iR-PU01-P 的裝置描述檔安裝)

2. 在 CODESYS 軟體介面上加入 Weintek CODESYS Library。



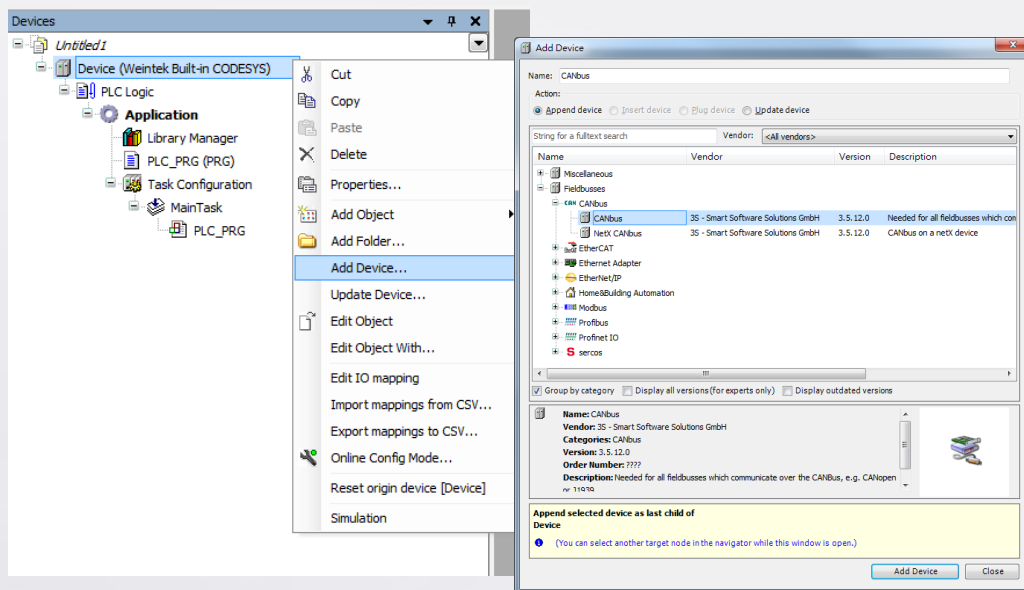
3. 完成安裝，即可使用 Motion Function Block。

### 3. CODESYS 專案加入 iR-PU01-P

#### 3.1. 使用 Weintek Built-in CODESYS 加入 iR-PU01-P 裝置

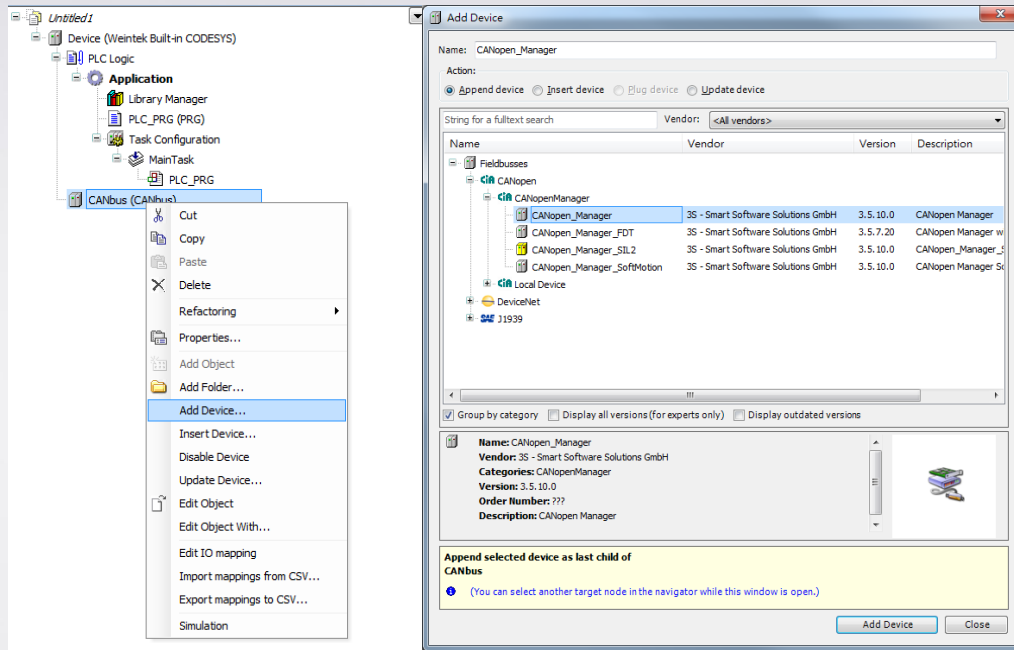
- 加入 CANbus 裝置：

[Device]->[Add Device]->[Fieldbusses]->[CANbus]



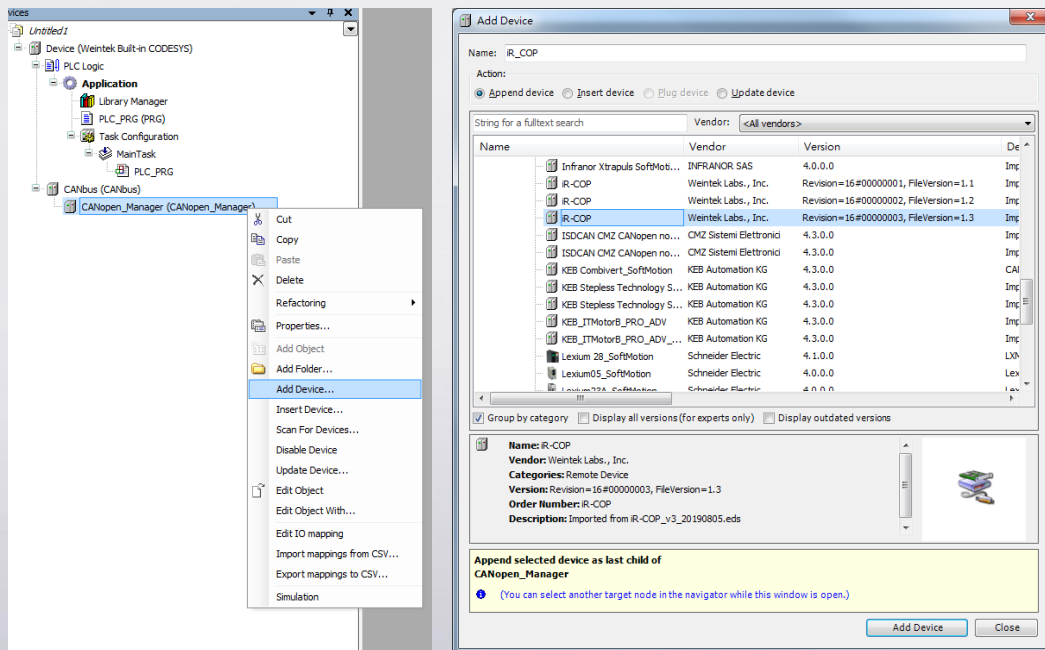
- 加入 CANopen\_Manager 裝置：

[CANbus]->[Add Device]->[CANopen\_Manager]



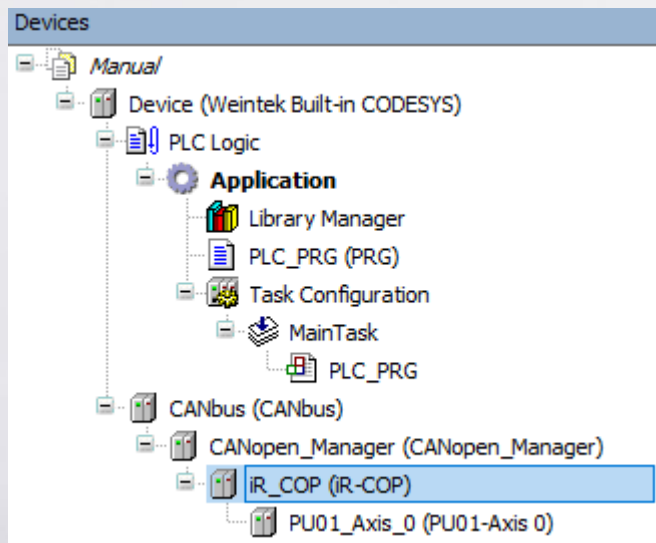
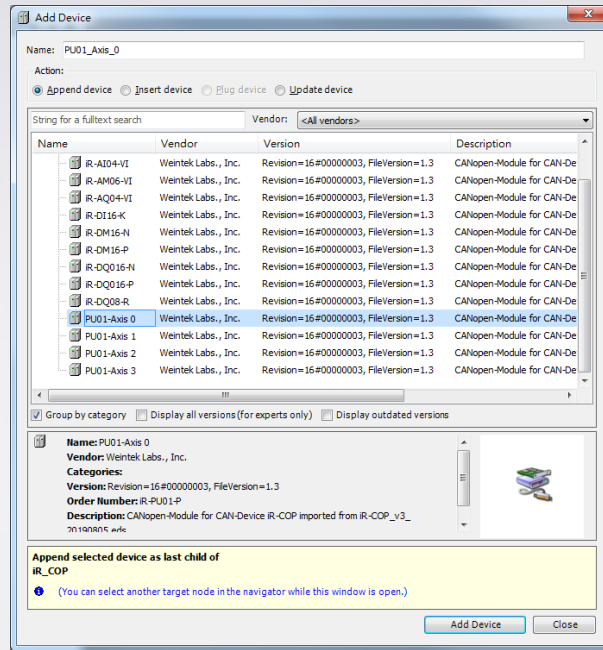
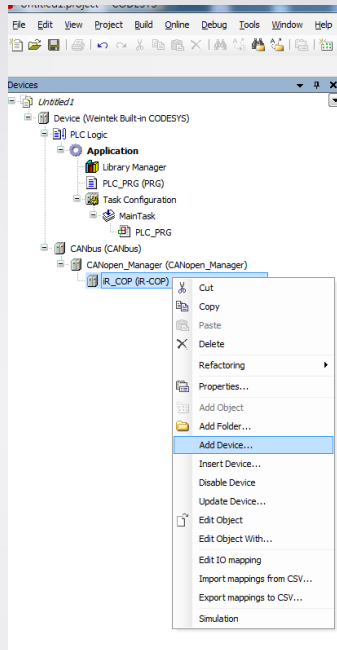
- 加入 iR-COP 模組：

[CANopen\_Manager] ->[Add Device]->[iR-COP] (版本 V1.3)



- 加入 iR-PU01-P 模組：

[iR-COP]->[Add Device]->[PU01-Axis 0]

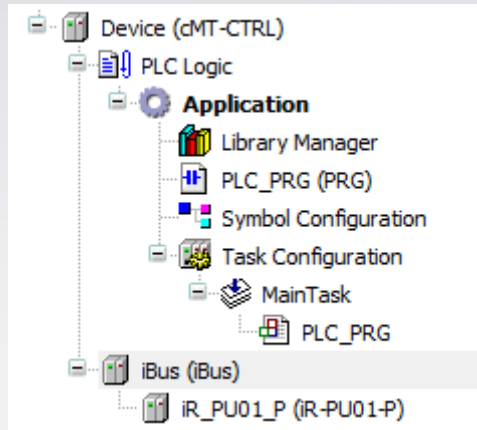


## 3.2. 使用 cMT-CTRL 加入 iR-PU01-P 裝置

- 加入 iR-PU01-P 裝置：

[iBus]->[Add Device]->[Miscellaneous]->[iR-PU01-P]

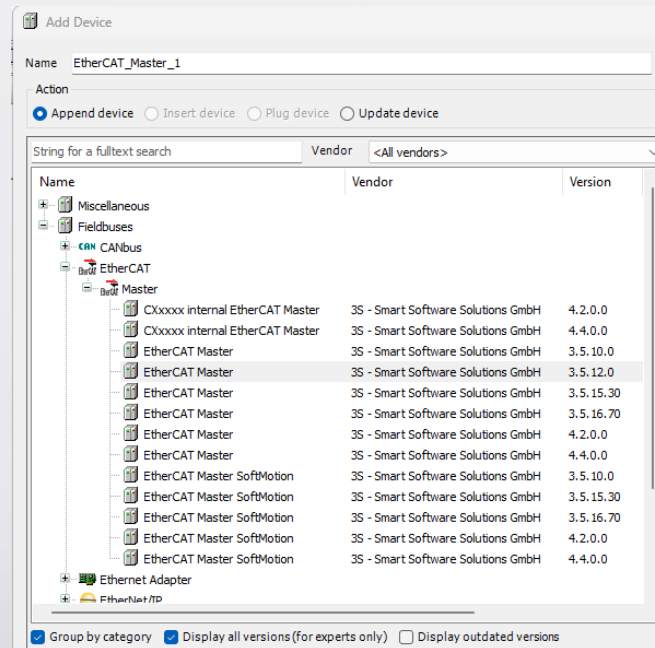




### 3.3. 使用 Weintek Built-in CODESYS 加入 iR-PU01-P 裝置

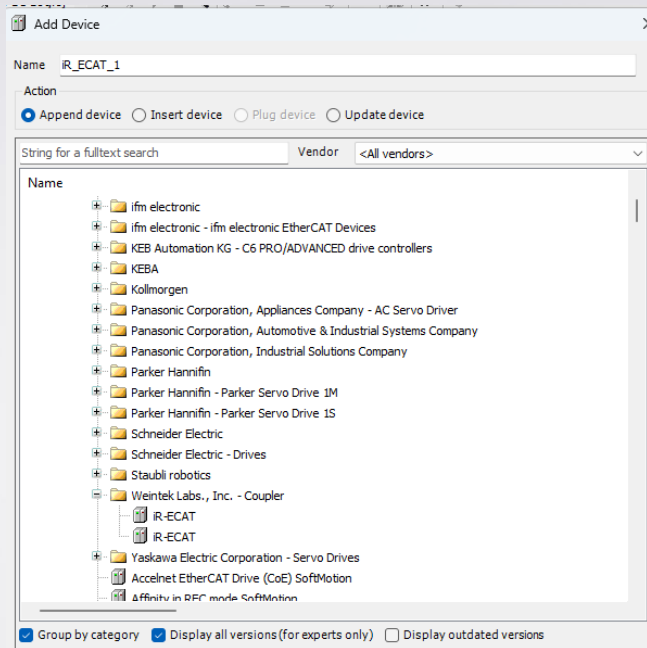
- 加入 EtherCAT\_Master 裝置：

[Device]->[Add Device]->[Fieldbuses]->[EtherCAT]->[Master]



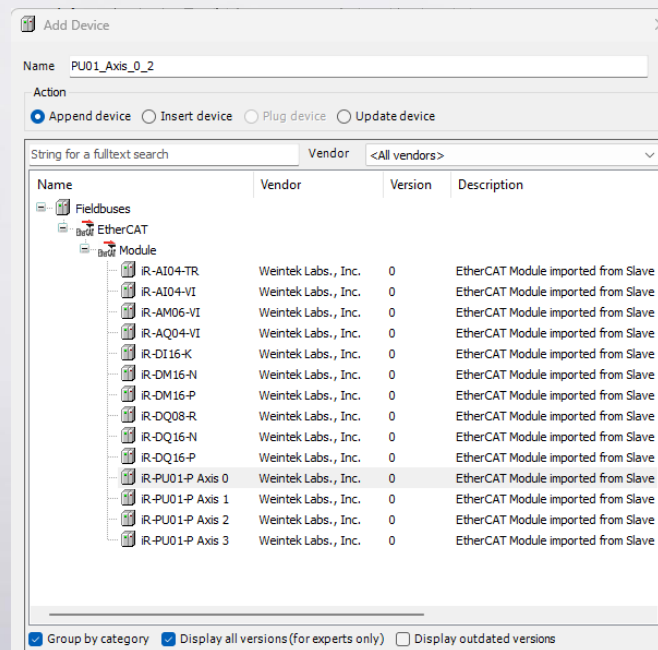
- 加入 iR-ECAT 裝置：

[EtherCAT\_Master]->[Add Device]->[iR-ECAT]



- 加入 iR-PU01-P 模組：

[iR-COP]->[Add Device]->[PU01-Axis 0]



## 4. iR-PU01-P 參數設定

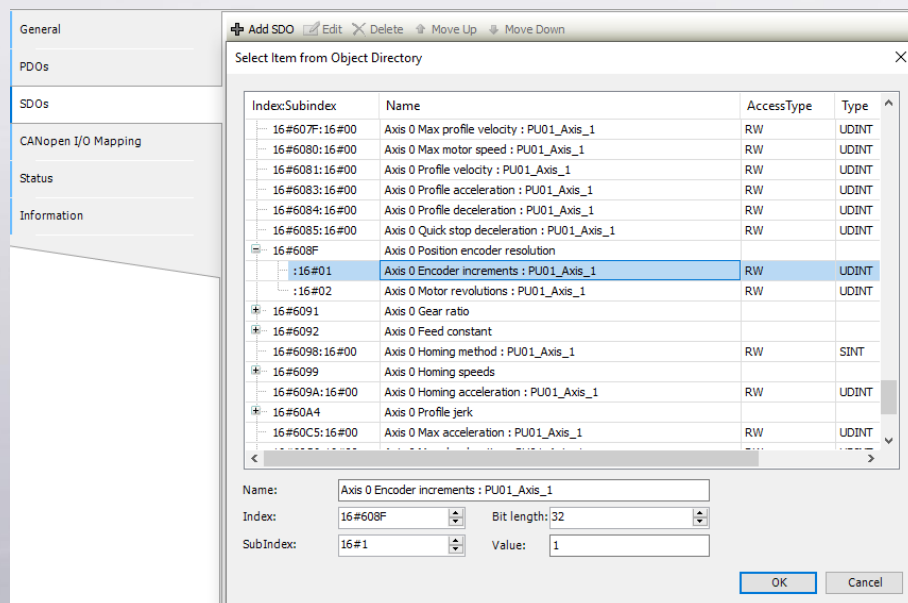
範例使用設定項：

Line	Index:Subindex	Name	Value	Bit length
1	16#608F:16#01	Encoder increments : AX1_PU01	16#1	32
2	16#608F:16#02	Axis 1 Motor revolutions : PU01_Axis_1	1	32
3	16#5511:16#00	Axis 1 Pulse Output Method : PU01_Axis_1	4	8
4	16#6080:16#00	Axis 1 Max motor speed : PU01_Axis_1	200000	32
5	16#607F:16#00	Axis 1 Max profile velocity : PU01_Axis_1	200000	32
6	16#60C5:16#00	Axis 1 Max acceleration : PU01_Axis_1	100000	32
7	16#60C6:16#00	Axis 1 Max deceleration : PU01_Axis_1	100000	32
8	16#6085:16#00	Axis 1 Quick stop deceleration : PU01_Axis_1	100000	32
9	16#6098:16#00	Axis 1 Homing method : PU01_Axis_1	27	8
10	16#607C:16#00	AX1 Home offset	0	32
11	16#6099:16#01	Speed during search for switch : AX1_PU01	10000	32
12	16#6099:16#02	Speed during search for zero : AX1_PU01	2000	32
13	16#609A:16#00	AX1 Homing acceleration : AX1_PU01	10000	32

在使用運動控制模組控制馬達之前，必須先對模組加入保護措施以及單位設定，若未設定直接使用功能塊進行控制，iR-PU01-P 模組則顯示錯誤狀態燈號。

參數設定方式如下：

- [iR-COP]->[SDOs]->[Add SDO]
- [iR-ECAT]->[Startup Parameters]->[Add]



加入後的 SDO 參數在登入啟動時，寫入 iR-PU01-P 模組內。

- 馬達解析度設定:608Fh

馬達解析度為馬達轉一圈需要的脈波數。在試運轉中數值皆設定為 1。

$$\text{馬達解析度} \text{Position encoder resolution} = \frac{\text{編碼器增量} \text{encoder increments}}{\text{馬達公轉} \text{motor revolution}}$$

## ● 脈波輸出方式:5511h

脈波輸出方式必須依照馬達驅動器支援的脈波形式作設定。驅動器與 iR-PU01-P 模組兩邊的設定脈波形式相同才能正確控制馬達的方向與距離。

子索引 00h：脈波輸出型式

Bit7-	保留			
Bit 4				
Bit3-	設定值	PA	PB	
Bit 0	0	Disable	Disable	
	1	CW	CCW	
	2	Pulse	NC	
	3	Pulse	Direction	
	4	A	B	
	5	A(2 倍頻)	B(2 倍頻)	

	6	A(4 倍頻)	B(4 倍頻)	
--	---	---------	---------	--

## ● 速度最大值:6080h, 607Fh, 60C5h, 60C6h

馬達最大轉速：6080h

馬達最大轉速依照馬達規格填入最大轉速，一般馬達最大轉速的單位為

RPM(Round per minute)，此項參數需填入的單位為 PPS(Pulse per second)，

填入前須先計算上列兩者單位轉換。

檔案最大速度：607Fh

此最大速度為限制專案中的最大速度，若與馬達最大轉速 6080h 相衝的話，則選擇數值較小的為最大速度。

最大加速度/減速度：60C5h, 60C6h

限制馬達最大加速度及減速度，在程式中若設定值超過最大值，則以最大值運行。

## ● 急停減速度:6085h

iR-PU01-P 模組在運動中發生錯誤或者碰觸到極限訊號時，會自動依照急停所設定的減速度停止運動。

## ● 歸原點方式:6098h

iR-PU01-P 模組提供 37 種歸原點方式，此範例使用歸原點方式 27。啟動後快速向負極限移動找尋原點，碰觸原點感測器改慢速反向離開原點感測器。

## ● 原點偏移量:607Ch

做完歸原點運動後可藉由原點偏移量給定起始位置。

## ● 歸原點速度:6099h

啟動歸原點後代入 Speed during search for switch 的速度。

找到第一個參考點後帶入 Speed during search for zero 的速度。

## ● 歸原點加速度:609Ah

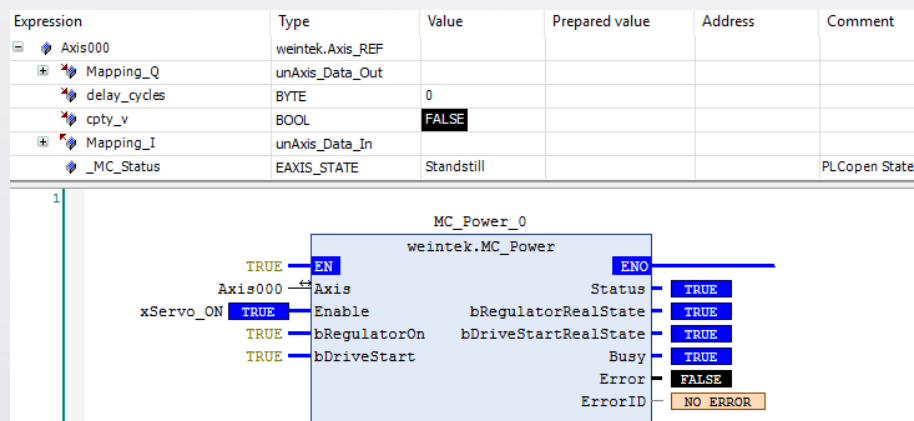
歸原點運動的加速度設定。

## 5. 功能塊介紹

關於如何加入 Weintek Library 請參考文件《[威綸函式庫說明](#)》的第二章。

### 軸控功能啟動 MC\_Power

MC\_Power 功能塊如同伺服控制的 Servo 啟動功能，所有運動功能塊使用前都必須先啟動 Power 功能塊，Power 功能塊啟動後若沒有發生錯誤，則軸進入待命狀態〈Standstill〉。



如上圖範例所示，觸發 MC\_Power.Enable，MC\_Status 處在〈Standstill〉狀態，代表軸待命可隨時執行任何移動的功能塊指令。

### 歸原點運動 MC\_Home

Motion Function Block 提供 37 種歸原點方式，當觸發 MC\_Home 功能塊會依照設定執行對應的歸原點方式。原點設定參數如下：

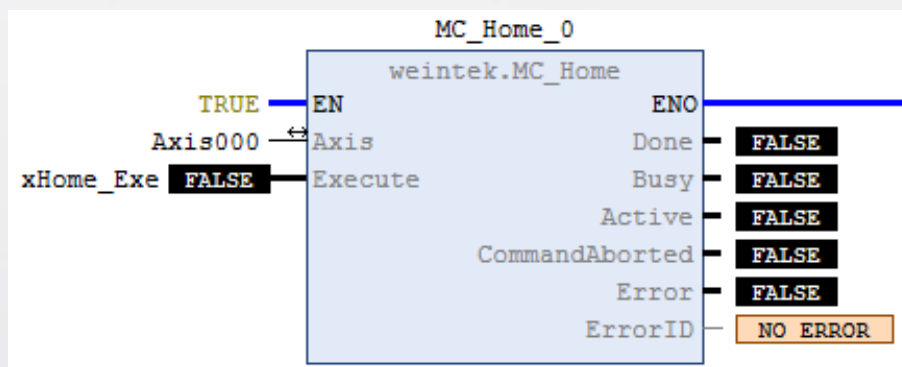
- 歸原點方式 6098(根據 CiA402 規範的 37 種歸原點方式)
- 歸原點速度-慢速 6099#01
- 歸原點速度-快速 6099#02
- 歸原點加速度 609A



- 原點偏移量 607C

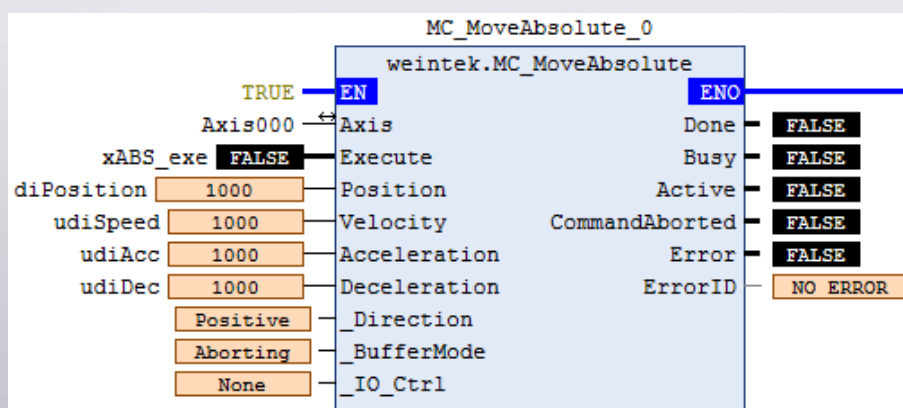
+ Add SDO Edit Delete Move Up Move Down				
Line	Index:Subindex	Name	Value	Bit length
1	16#6098:16#00	Axis 1 Homing method : PU01_Axis_1	27	8
2	16#6099:16#01	Axis 1 Speed during search for switch : PU01_Axis_1	2000	32
3	16#6099:16#02	Axis 1 Speed during search for zero : PU01_Axis_1	10000	32
4	16#609A:16#00	Axis 1 Homing acceleration : PU01_Axis_1	10000	32
5	16#607C:16#00	Axis 1 Home offset : PU01_Axis_1	1000	32

軸在〈Standstill〉狀態下觸發 MC\_Home.Execute 即根據上述參數進行歸原點運動，軸狀態轉移為〈Homing〉待歸原點完成回復成〈Standstill〉。



## 絕對位置模式移動 MC\_MoveAbsolute

指定絕對位置。依照功能塊的參數進行移動。觸發 MC\_MoveAbsolute.Execute 即根據功能塊參數進行定位控制。



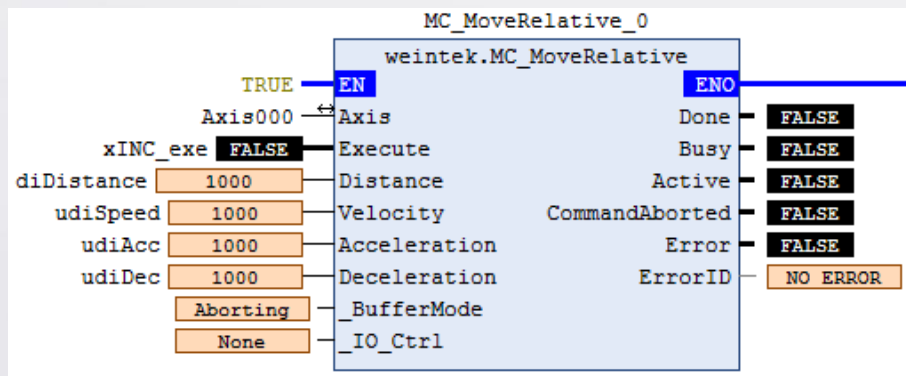
- Position：絕對位置輸入。目標位置 = 絕對位置輸入。
- Velocity：移動速度，數值不可為 0。
- Acceleration/Deceleration：移動加速度/減速度，數值不可為 0。

- `_Direction`：若為旋轉軸，此參數可指定移動方向或最短路徑。
- `_BufferMode`：可設定此參數接續前一個移動命令達到連續移動的功能。
- `_IO_Ctrl`：此功能塊可藉由數位輸入觸發移動命令，功能完成時輸出數位信號。
- 軸狀態在觸發後轉為〈Discrete Motion〉待移動到位轉成〈Standstill〉。

### 相對位置模式移動 MC\_MoveRelative

指定移動距離由當前位置加上移動距離等於移動完成位置。觸發

`MC_MoveRelative.Execute` 即根據功能塊參數進行定位控制。



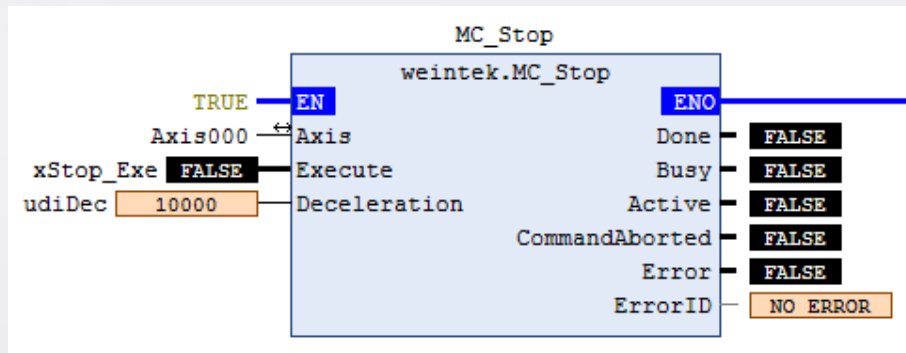
- `Position`：相對位置輸入。目標位置 = 現在位置 + 相對位置輸入。
- `Velocity`：移動速度，數值不可為 0。
- `Acceleration/Deceleration`：移動加速度/減速度，數值不可為 0。
- `_BufferMode`：可設定此參數接續前一個移動命令達到連續移動的功能。
- `_IO_Ctrl`：此功能塊可藉由數位輸入觸發移動命令，功能完成時輸出數位信號。
- 軸狀態在觸發後轉為〈Discrete Motion〉待移動到位轉成〈Standstill〉。



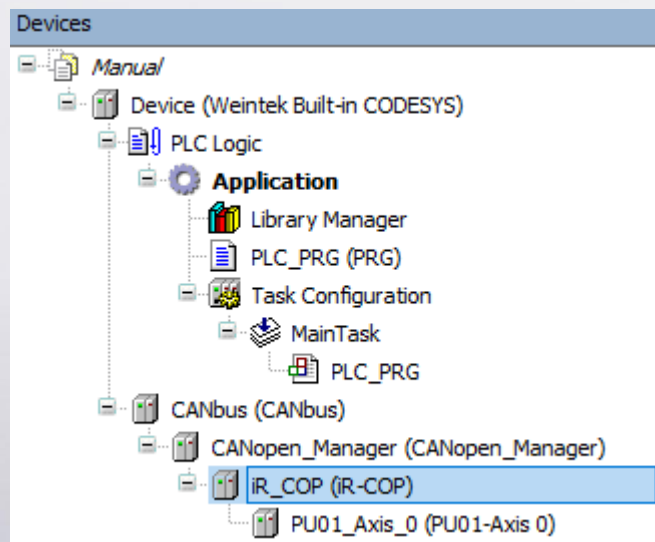
## 停止 MC\_Stop

停止屬於中斷移動軸的命令，使用停止必須等到停止完成後才能繼續下其他移動命令。

觸發 MC\_Stop.Execute 即停止運動控制。



- Deceleration：減速度，不可為 0。軸狀態在功能塊完成後轉為〈Standstill〉。



## 6. 範例程序

### 絕對/相對位置移動功能

xEnable\_Power：啟動控制系統按鈕。

xServe\_ON：啟動伺服激磁。

xHome\_exe：TRUE=執行歸原點運動。

xABS\_exe：TRUE=執行絕對位置移動。

xINC\_exe：TRUE=執行相對位置移動。

xTrig\_Stop：TRUE=運動中停止。

xTrig\_Reset：TRUE=復歸 iR-PU01-P 錯誤狀態。

diPosition：移動絕對位置。

diDistance：移動距離。

udiSpeed：移動速度設定。

udiAcc：移動加速度設定。

udiDec：移動減速度設定。

eBuffer\_Mode：可將下個移動命令存於 Buffer 中，運動執行結束接續執行。

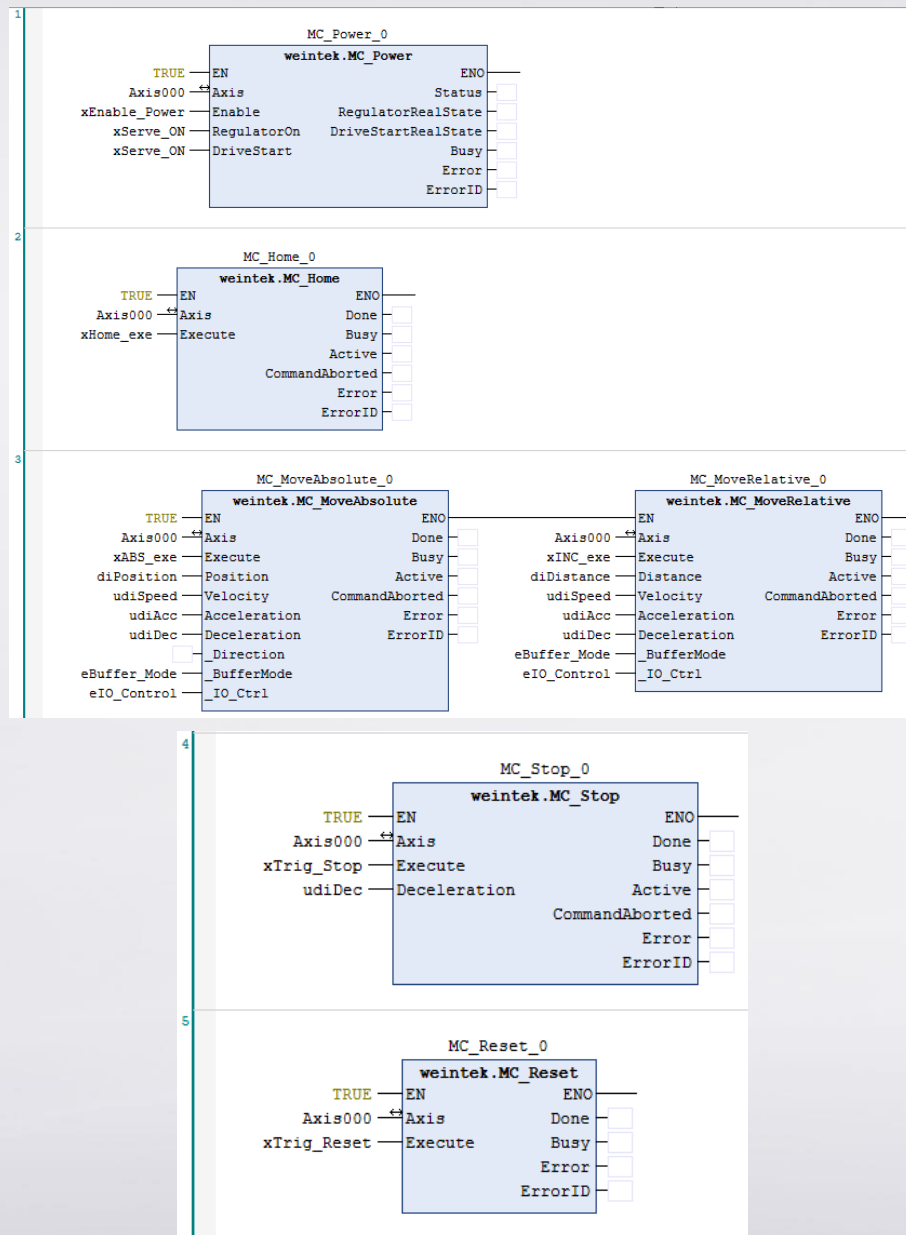
eIO\_Control：可藉由 iR-PU01-P 模組內建數位輸入觸發移動。

### 宣告

```
VAR
    // Axis reference
    Axis000 : Weintek.Axis_REF_Lite ;
    // Motion Control Function Block
    MC_Power_0: weintek.MC_Power ;
    MC_MoveAbsolute_0 : weintek.MC_MoveAbsolute ;
    MC_MoveRelative_0 : weintek.MC_MoveRelative ;
    MC_Home_0 : weintek.MC_Home ;
    MC_Stop_0: weintek.MC_Stop;
    MC_Reset_0: weintek.MC_Reset;
    // Positioning control Button
    xEnable_Power, xServe_ON, xABS_exe, xINC_exe, xHome_exe : BOOL ;
    xTrig_Stop, xTrig_Reset : BOOL;
    // Positioning parameter
    diPosition : DINT := 10000 ;
    diDistance : DINT := 10000 ;
    udiSpeed : UDINT := 1000 ;
    udiAcc : UDINT := 1000 ;
    udiDec : UDINT := 1000 ;
    eIO_Control : weintek.eMC_IO_CTRL ;
    eBuffer_Mode : weintek.eMC_BUFF_MODE ;
    // Variable
END_VAR
```

宣告程式中必要的變數並給定初值。

## FBD 程式



1：開啟軸控制系統，xEnable\_Power & xServe\_ON 皆必須為 TRUE，若無錯誤則可開始進行運動控制。

2：歸原點功能塊。

3：絕對位置移動與相對位置移動功能塊。

4：停止功能塊。運動中觸發 xTrig\_Stop 可停止當前運動。

5：若在移動中有發生錯誤，可觸發 xTrig\_Reset 復歸 iR-PU01-P。

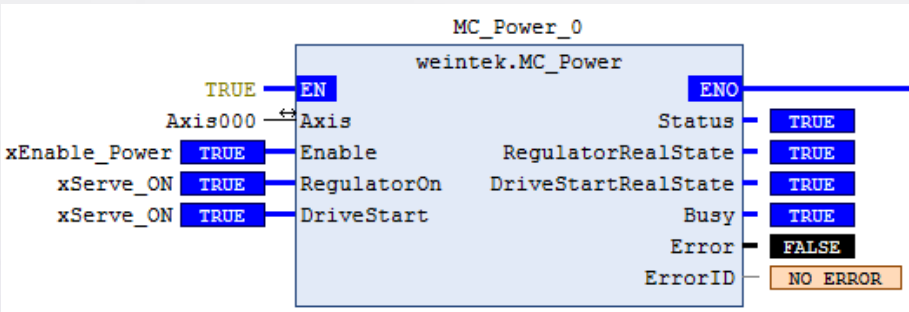
7. 登入操作

範例實際操作試運轉功能步驟。

啟動運動控制系統

xEnable\_Power & xServe\_ON 按鈕按下(TRUE)，啟動軸控系統。

無發生錯誤 Status、RegulatorRealState、DriveStartRealState、Busy 為 TRUE，  
發生錯誤則 Error 等於 TRUE。



歸原點運動

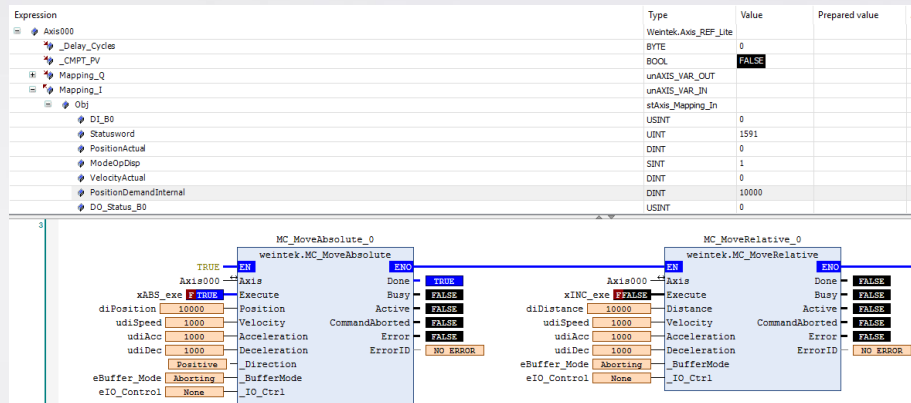
觸發 xHome\_exe 啟動歸原點，歸原點運動自動參照下圖配置執行。

9	16#6098:16#00	Axis 1 Homing method : PU01_Axis_1	27	8
10	16#607C:16#00	AX1 Home offset	0	32
11	16#6099:16#01	Speed during search for switch : AX1_PU01	10000	32
12	16#6099:16#02	Speed during search for zero : AX1_PU01	2000	32
13	16#609A:16#00	AX1 Homing aceleration : AX1_PU01	10000	32

以 Speed during search for switch 的速度往負向搜尋原點感測器，搜尋到原點感測器後以 Speed during search for zero 的速度往正向離開原點感測器，離開後帶入 Home offset 數值表示歸原點完成。

## 絕對位置移動

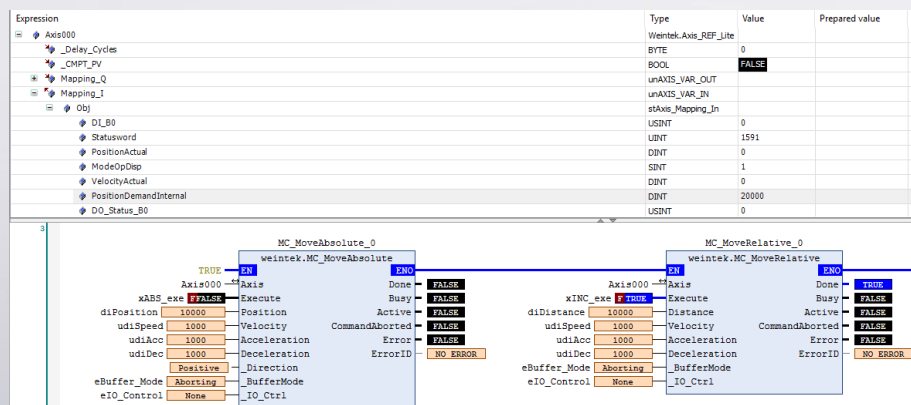
觸發 xABS\_exe 及啟動絕對位置移動。移動至脈波等於 10000 的位置。



運動完成後(MC\_MoveAbsolute\_0.Done=TRUE)，馬達定位在脈波等於 10000 的位置。可在軸參數 Axis000.Mapping\_I » Obj » PositionDemandInternal 暫存器查看當前位置。

## 相對位置移動

觸發 xINC\_exe 及啟動相對位置移動。移動至當前位置(10000)加上 diDistance(10000)的位置。



運動完成後(MC\_MoveRelative\_0.Done=TRUE)，馬達定位在脈波等於 20000 的位置。可在軸參數 Axis000.Mapping\_I » Obj » PositionDemandInternal 暫存器查看當前位置。

CODESYS® is a trademark of CODESYS GmbH.

本文中出現的其他公司名、產品名或商標均為各公司的商標或註冊商標。

本文件中的資訊可能隨時變更，本公司將不另行通知。

Copyright© 2023 WEINTEK IIOT LTD. All rights reserved.