



台達電子工業股份有限公司  
機電事業群  
33068 桃園縣桃園市興隆路 18 號  
TEL: 886-3-3626301  
FAX: 886-3-3716301

\* 本使用手冊內容若有變更，恕不另行通知

MULTIPROG PLC 操作手冊



## MULTIPROG PLC 操作手冊

[www.deltaww.com](http://www.deltaww.com)



# 序言

---

感謝您使用本產品，本使用手冊提供 MULTIPROG 的使用相關資訊。

本手冊內容

- 系統介紹及安裝
- MULTIPROG 操作介面
- IEC61131-3 介紹
- IEC61131-3 定義編程語言
- PLCopen 簡介及功能塊介紹
- 建立專案

MULTIPROG 產品特色

MULTIPROG 是基於 IEC61131-3 標準語言所建立的軟體編輯器，具備變數狀態監控、示波器、PLC 指令編寫等功能的控制器整合軟體。並支援台達依照 PLCOpen 所建立的 Function Block，讓使用者可依據自己的需求建立專案來實現不同的應用。

如何使用本操作手冊

您可視本手冊為學習使用 MULTIPROG 之參考資訊，手冊將告訴您如何安裝、設定、使用產品。

台達電子技術服務

如果您在使用上仍有問題，歡迎洽詢經銷商或本公司客服中心。

(此頁有意留為空白)

# 目錄

---

## 1

### 系統介紹及安裝

- 1.1 MULTIPROG 簡介 ..... 1-2
- 1.2 安裝流程 ..... 1-3

## 2

### MULTIPROG 操作介面

- 2.1 MULTIPROG 使用者介面簡介 ..... 2-2
- 2.2 功能區及常用工具列 ..... 2-3
- 2.3 專案樹 ..... 2-6
- 2.4 程式編輯區 ..... 2-7
- 2.5 功能塊函式庫 ..... 2-8
- 2.6 狀態顯示區 ..... 2-8

## 3

### IEC61131-3 介紹

- 3.1 IEC 61131-3 介紹 ..... 3-2
- 3.2 Configuration (配置) ..... 3-4
- 3.3 Resource (資源) ..... 3-5
- 3.4 Task (任務) ..... 3-8
- 3.5 POU (程序組織單元) ..... 3-13
- 3.6 Data Type (資料型態) ..... 3-19

## 4

### IEC61131-3 定義編程語言

- 4.1 指令表(IL)編程語言 ..... 4-2
- 4.2 梯形圖(LD)編程語言 ..... 4-4
- 4.3 功能塊圖(FBD)編程語言 ..... 4-11
- 4.4 結構化文字(ST)編程語言 ..... 4-16
- 4.5 順序功能圖(SFC)編程語言 ..... 4-18

# 5

## PLCopen 簡介及功能塊介紹

5.1 PLCopen 介紹 .....	5-3
5.1.1 PLCopen 組織簡介 .....	5-3
5.1.2 PLCopen Motion Control 簡介 .....	5-4
5.1.3 常用變數定義 .....	5-5
5.1.4 警報代碼 .....	5-9
5.2 功能塊介紹 .....	5-12
5.2.1 功能塊總覽 .....	5-12
5.2.2 單軸功能塊 .....	5-14
5.2.2.1 MC_Home .....	5-14
5.2.2.2 MC_Stop .....	5-18
5.2.2.3 MC_Jog .....	5-22
5.2.2.4 MC_MoveAbsolute .....	5-25
5.2.2.5 MC_MoveRelative .....	5-30
5.2.2.6 MC_MoveAdditive .....	5-33
5.2.2.7 MC_MoveVelocity .....	5-37
5.2.2.8 MC_Power .....	5-40
5.2.2.9 MC_Reset .....	5-42
5.2.2.10 MC_SetOverride .....	5-43
5.2.3 群組功能塊 .....	5-46
5.2.3.1 MC_GroupReset .....	5-46
5.2.3.2 MC_GroupSetOverride .....	5-48
5.2.3.3 MC_GroupStop .....	5-51
5.2.3.4 MC_MoveLinearAbsolute .....	5-55
5.2.3.5 MC_MoveLinearRelative .....	5-58
5.2.3.6 MC_MoveDirectAbsolute .....	5-61
5.2.3.7 MC_MoveDirectRelative .....	5-64
5.2.4 擴充功能塊 .....	5-67
5.2.4.1 MC_MoveDirectAbsolute_Ex .....	5-67
5.2.4.2 MC_MoveDirectRelative_Ex .....	5-70
5.2.4.3 MC_Home_Ex .....	5-73
5.2.4.4 MC_MoveLinearJog .....	5-75
5.2.5 系統功能塊 .....	5-78
5.2.5.1 SYS_AlarmSet .....	5-78
5.2.5.2 SYS_AlarmShow .....	5-80
5.2.5.3 SYS_ReadPara .....	5-82
5.2.5.4 SYS_WritePara .....	5-84
5.2.5.5 SYS_RemoteReadPara .....	5-86
5.2.5.6 SYS_RemoteWritePara .....	5-88
5.2.5.7 APP_ReadPara .....	5-91

5.2.5.8	APP_WritePara .....	5-93
5.2.5.9	DRL_CommandSet .....	5-95
5.2.5.10	DRL_GetStatus.....	5-97
5.2.5.11	DRL_GetMainCurrentLine .....	5-98
5.2.5.12	DRL_GetCommandReply.....	5-99
5.2.5.13	DRL_GetCurrentProgNo .....	5-100
5.2.5.14	MB_ReadHoldRegs .....	5-101
5.2.5.15	MB_WriteRegs .....	5-105
5.2.5.16	MB_WriteSingleRegs .....	5-108
5.2.5.17	MB_TCPReadHoldRegs .....	5-110
5.2.5.18	MB_TCPWriteRegs .....	5-114
5.2.5.19	MB_TCPWriteSingleRegs .....	5-117
5.2.5.20	CAN_SDORead.....	5-119
5.2.5.21	CAN_SDOWrite .....	5-122

# 6

## 建立專案

6.1	建立功能塊.....	6-2
6.2	建立函式庫.....	6-4
6.3	建立更新檔(bootfile.img).....	6-6

(此頁有意留為空白)

# 1

## 系統介紹及安裝

---

本章節將介紹 MULTIPROG 軟體及在 Windows 作業系統中安裝 MULTIPROG 的步驟。

1.1	MULTIPROG 簡介 .....	1-2
1.2	安裝流程 .....	1-3

## 1.1 MULTIPROG 簡介

MULTIPROG 是 Phoenix Contact 公司針對自動化應用開發的通用 PLC 程式設計軟體。主要針對中大型控制應用場合，可以廣泛的應用在機械製造、傳統產業、自動化等行業。

此軟體是基於 COM / DCOM 的技術架構，適用於 Windows 7、Windows 8 及以上的視窗操作系統，其結構符合 IEC61131-3 標準，支援標準定義的五種 PLC 程式語言，且可以讓使用者自行定義函式庫及資料結構。該程式編輯工具可以應用在現有的控制系統，也可以同時使用多個任務來對 PLC 控制器進行統一配置、程式編輯和操作。

MULTIPROG 提供了豐富的操作命令和人機互動介面，支持拖曳且所有功能皆可透過鍵盤來操作。亦提供變數即時監控，強制設定及覆寫功能，並有完整偵錯模式，可以在程式設置斷點和單步執行，同時有邏輯分析功能，可便利的紀錄輸入輸出波型。在特殊應用上，可以進行原始碼保護和不停機的線上下載功能。針對不同國家的程式編輯習慣，提供包含變數名稱在內的多國語言，目前支援英文、簡體中文、日文、德文。

## 1.2 安裝流程

### ■ 軟體需求

1. Window7 · Windows8 或 Windows10 作業系統。
2. Microsoft .NET Framework 4.0 (含) 以上版本。

### ■ 安裝步驟

步驟一：執行安裝檔 MULTIPROG 5.50 Express Setup.exe，選擇安裝語言。(請使用管理者權限來執行此安裝檔)



圖 1.2.1 安裝語言選擇

步驟二：按照安裝精靈步驟開始進行安裝，確定安裝請點選「下一步」。

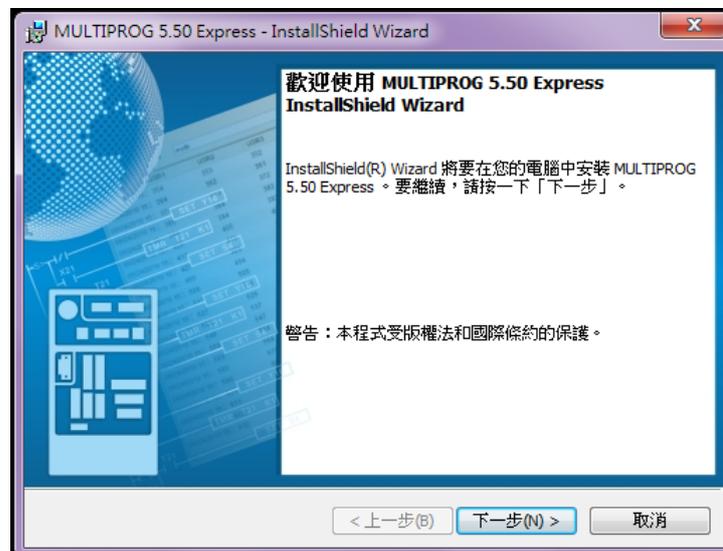


圖 1.2.2 MULTIPROG 安裝精靈

1

步驟三：在頁面中輸入使用者名稱及組織，輸入完成請點選「下一步」。

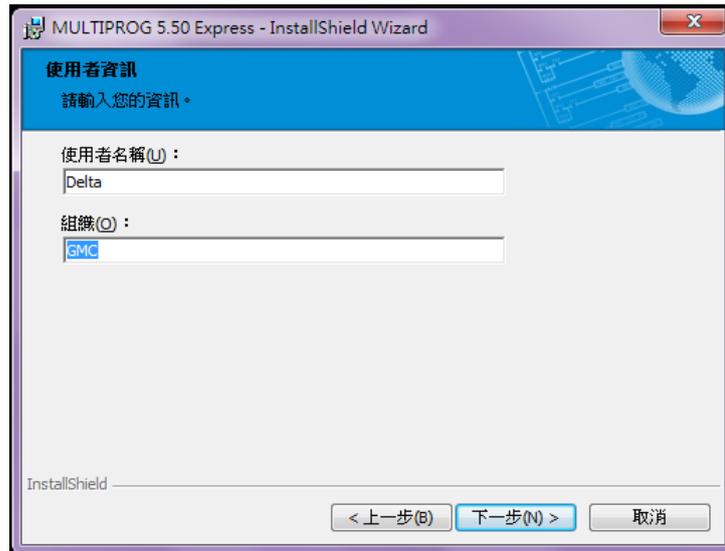


圖 1.2.3 使用者資訊輸入

步驟四：接著選擇安裝路徑，可以按照預設系統安裝路徑。使用者也可以針對自己需求，點選「變更」按鈕自訂安裝路徑。

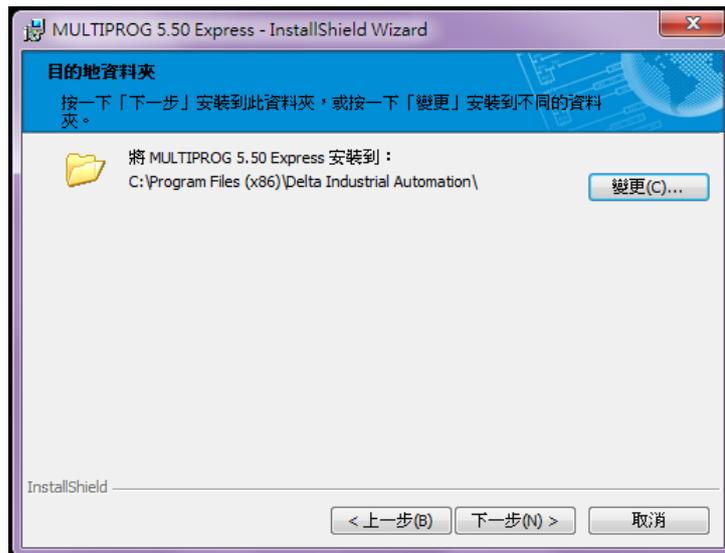


圖 1.2.4 安裝路徑選擇

步驟五：視窗會顯示安裝前確認，確定安裝請點選「安裝」，要更改其他安裝設定請點選「上一步」進行修改。放棄安裝則點選「取消」。點選安裝，精靈就會開始安裝軟體，並顯示安裝進度。

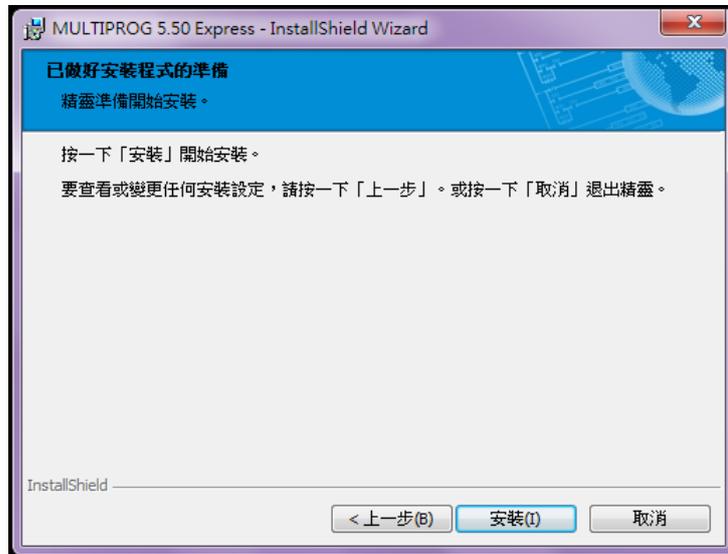


圖 1.2.5 安裝設定確認安裝頁面

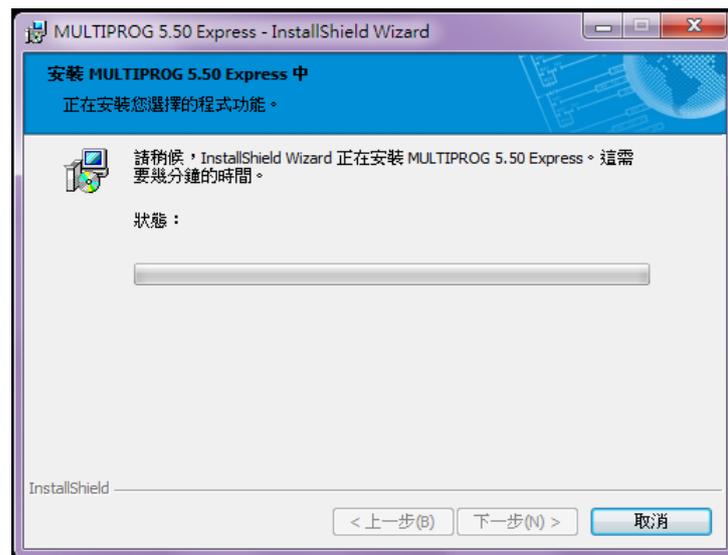


圖 1.2.6 安裝進度

步驟六：安裝結束後，視窗會顯示以下畫面，點選「安裝完成」結束精靈。

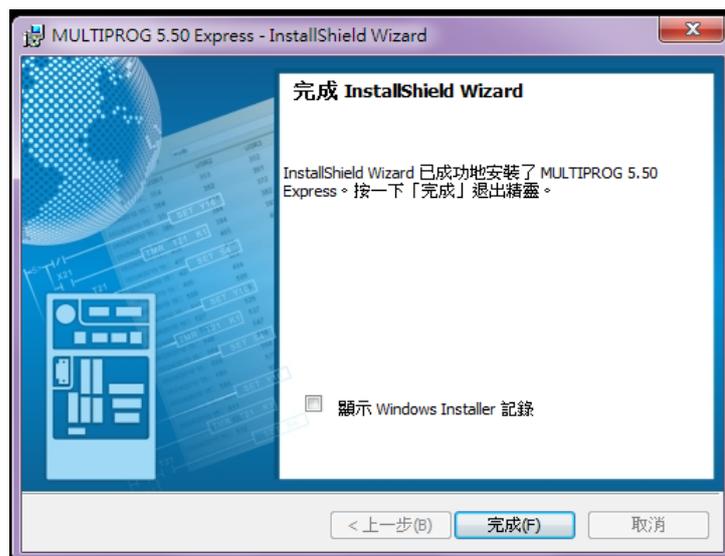


圖 1.2.7 安裝完成

# MULTIPROG 操作介面

# 2

本章節簡介 MULTIPROG 操作介面

2.1	MULTIPROG 使用者介面簡介.....	2-2
2.2	功能區及常用工具列.....	2-3
2.3	專案樹.....	2-6
2.4	程式編輯區.....	2-7
2.5	功能塊函式庫.....	2-8
2.6	狀態顯示區.....	2-8

## 2

## 2.1 MULTIPROG 使用者介面簡介

MULTIPROG 操作介面主要可分為五大部分，依序為(1)功能區及常用工具列、(2)專案樹、(3)程式編輯區、(4)功能塊函式庫以及(5)狀態顯示區。

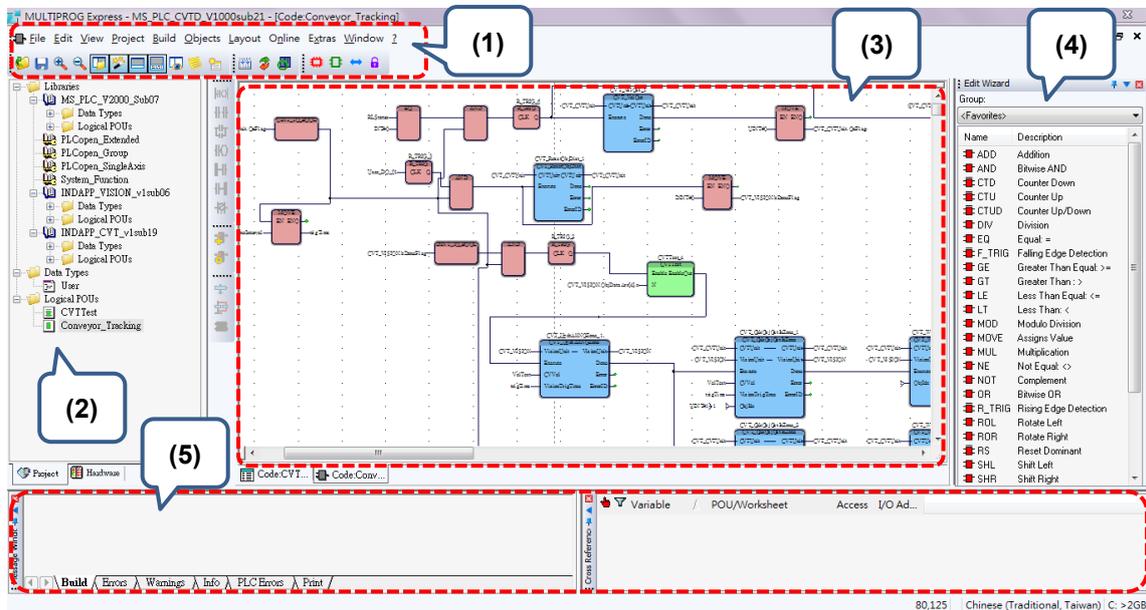


圖 2.1.1 MULTIPROG 使用者介面

功能區包括 11 個下拉式功能表，以及常用的編譯工具。

參數樹分為[Project]和[Hardware]頁籤，若是使用台達所開發的公版所建立的 Project，則 [Project]頁籤中內含預設的 MS PLC 公版和變數型態宣告，而[Hardware]則說明整個專案軟體配置。工作區是編輯程式的區塊，其中具有五種程式語言可選擇，針對各種不同需求進行開發。功能塊函式庫提供基礎以及符合 PLCopen 規範的功能塊，狀態顯示區則顯示編譯的錯誤、警報等資訊，同時也可以用來觀察變數的變化。

## 2.2 功能區及常用工具列

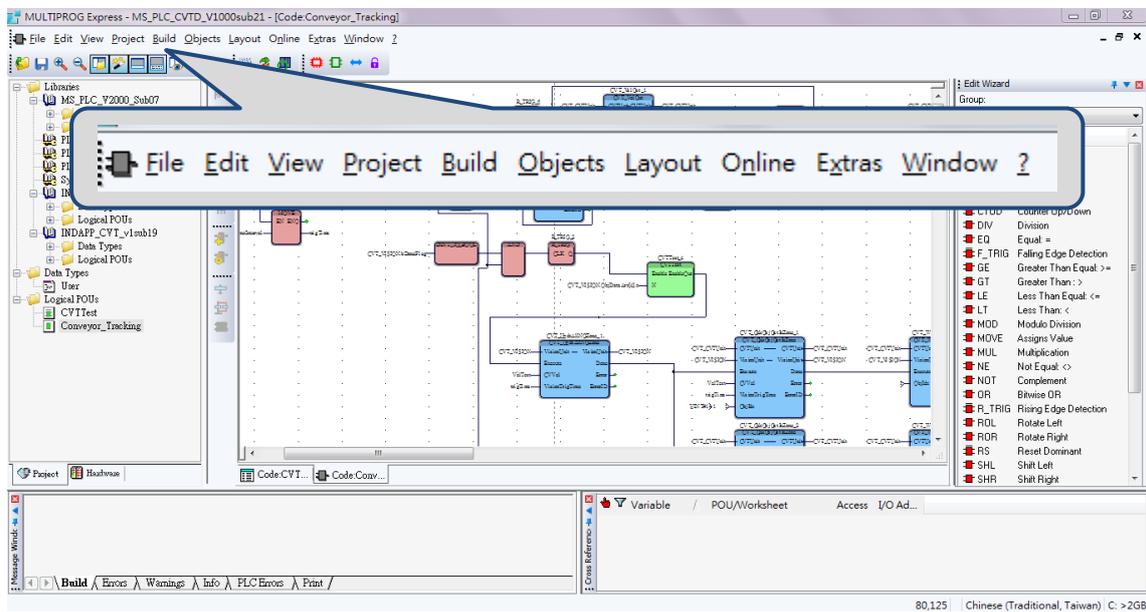
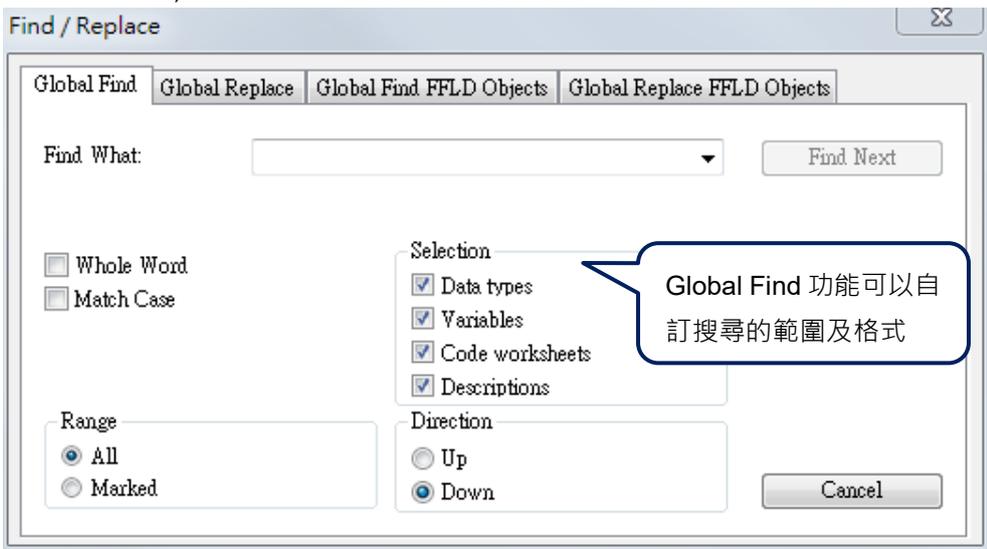
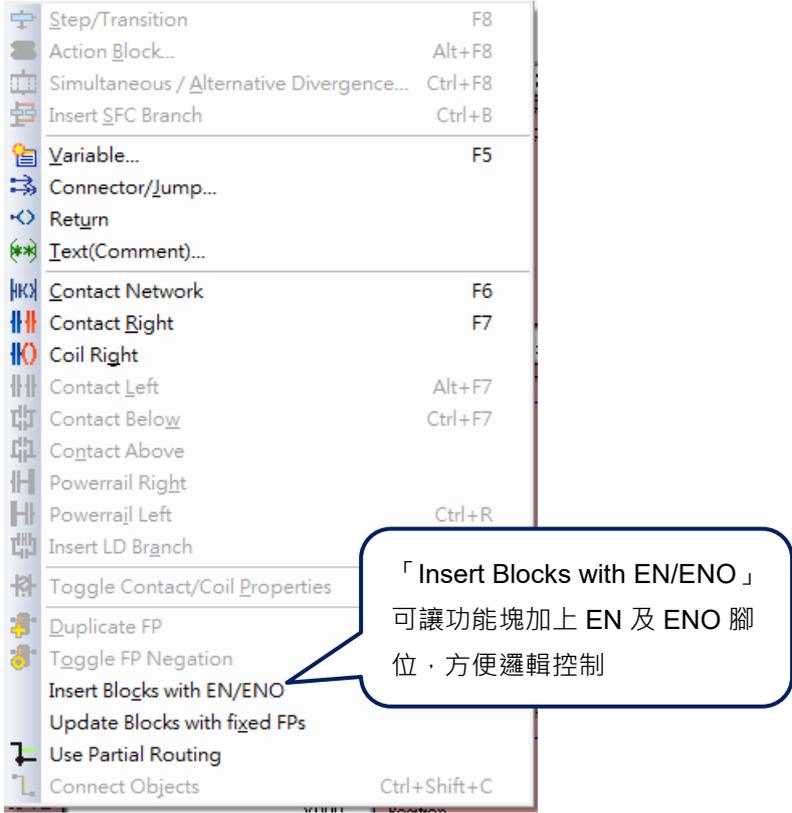


圖 2.2.1 功能區

選單	敘述
File	包含新建、開啟、儲存、關閉、刪除專案、儲存、刪除樣板、列印、列印設定以及預覽列印等功能。
Edit	包含常用的剪下、複製以及貼上等基本編輯功能，以及全域變數搜尋及取代選項 (Edit → Global Find)，可透過自訂搜尋限制範圍或吻合度，方便大範圍的搜尋。 
View	用於編輯操作介面。使用者可選擇區塊的顯示或隱藏並自行調整介面。
Project	可新增函式庫、變數型態和程式管理單元。
Build	提供編譯、協助除錯和移除未使用到的變數和功能塊等功能。

# 2

選單	敘述
Object	<p>只有在編輯程式會顯示。其中[Variable]功能用來插入新變數和編輯圖形化語言時，可以直接插入 Network、Coil 等梯形圖元件，其他不支援的情況下，這些元件則會顯示為灰色，表示無法點選。</p>
Layout	<p>提供頁面放大縮小功能。實際選項會根據當前編輯是圖形化還是文本化的編輯語言有些許不同。[Insert Block with EN/ENO] 功能如下圖：</p> 
Online	<p>提供專案偵錯指令。子選單中的[Logic Analyzer]可以記錄變數的數值，並將多組數值繪製成曲線供使用者比較觀察。</p>
Extras	<p>選單中的[PageLayoutEditor]可用來預覽及編輯列印的畫面佈局，其中 Option 功能則可以讓使用者自訂軟體的外觀。</p>
Windows	<p>可安排視窗及編輯按鈕。</p>

選單	敘述															
<p>?</p>	<p>提供所有指令和功能的協助。軟體的基本操作可以參考 PLC Help，基礎功能塊的定義則可以查詢 Help on Standard FB/FU，如下圖。</p> <p><b>功能查詢</b></p> <p><b>功能說明</b></p> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Data types</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>MN</td> <td>ELEMENTARY</td> <td>Minimum limit.</td> </tr> <tr> <td>IN</td> <td>ELEMENTARY</td> <td>Input value.</td> </tr> <tr> <td>MX</td> <td>ELEMENTARY</td> <td>Maximum limit.</td> </tr> <tr> <td>OUT</td> <td>ELEMENTARY</td> <td>Output value.</td> </tr> </tbody> </table>	Parameter	Data types	Description	MN	ELEMENTARY	Minimum limit.	IN	ELEMENTARY	Input value.	MX	ELEMENTARY	Maximum limit.	OUT	ELEMENTARY	Output value.
Parameter	Data types	Description														
MN	ELEMENTARY	Minimum limit.														
IN	ELEMENTARY	Input value.														
MX	ELEMENTARY	Maximum limit.														
OUT	ELEMENTARY	Output value.														

除了下拉選單外，還有常用按鈕列，提供檔案存取、介面設計、編輯除錯等常用功能。

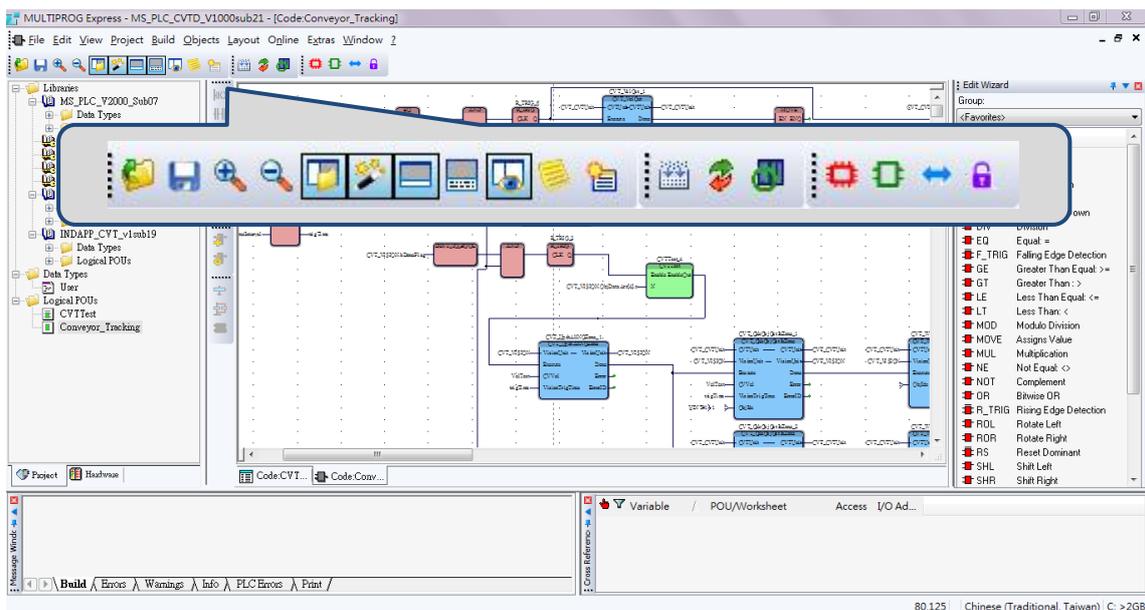


圖 2.2.2 常用工具列

# 2

## 2.3 專案樹

專案樹分為[Project]和[Hardware]兩個子頁面。[Project]頁面包含了公版提供的函式庫，而函式庫可再細分成變數型態(Data Type)與公版提供的基本程式單元 Logical POU。使用者可點選這兩個子項目並開查看內容，但無法編輯。有鑰匙圖案的项目，如 DMCNET\_RM 等六個項目，則無法查看。使用者可以自行在最下方的 Logical POUs 新增需要的功能，但要盡量避免和公版提供的程式有所衝突，以免發生錯誤。

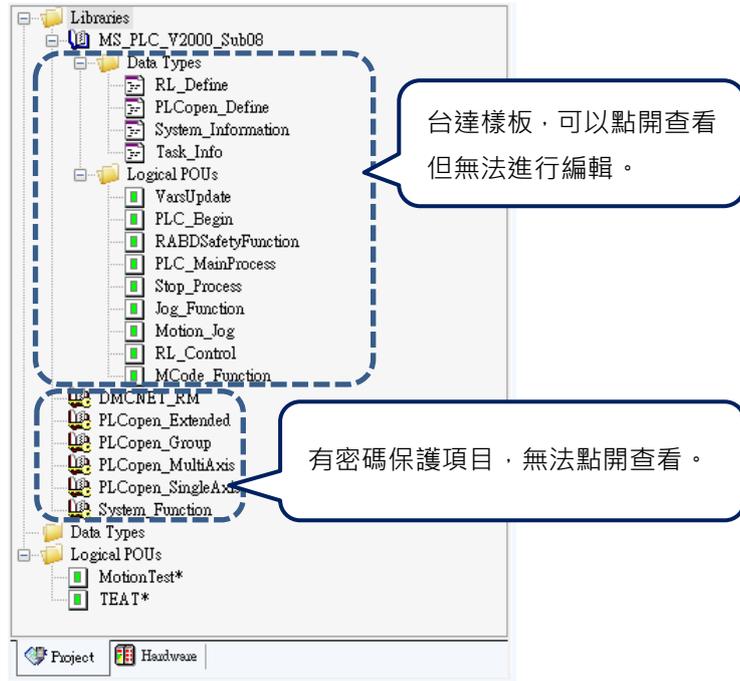


圖 2.3.1 專案樹 Project 頁面

[Hardware]頁面介紹整個專案的配置，可以針對需求將 POU 放置到不同的 Tasks 中。  
 [Global Variables]可以查看所有的變數。  
 [IO\_Configuration]可以查看目前選擇產品的 IO 配置。  
 關於 Hardware 詳細說明，請參考第三章。

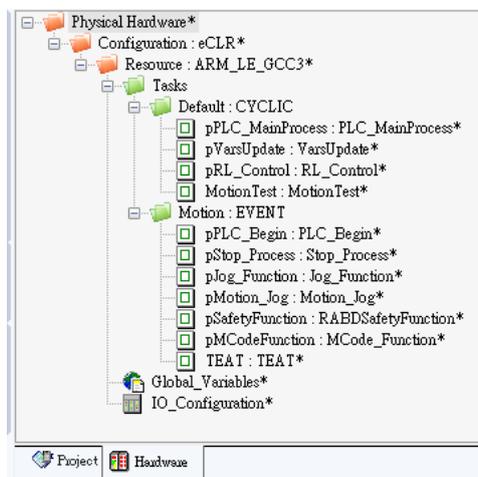


圖 2.3.2 專案樹 Hardware 頁面

## 2.4 程式編輯區

此視窗用來編寫 PLC 程式，下方的頁籤可以切換當前選擇的程式或是變數頁面，圖形化語言可以互相配合使用。不同的程式可以透過不同的 PLC 語言撰寫，透過不同語言的優勢來實現所需功能。

2

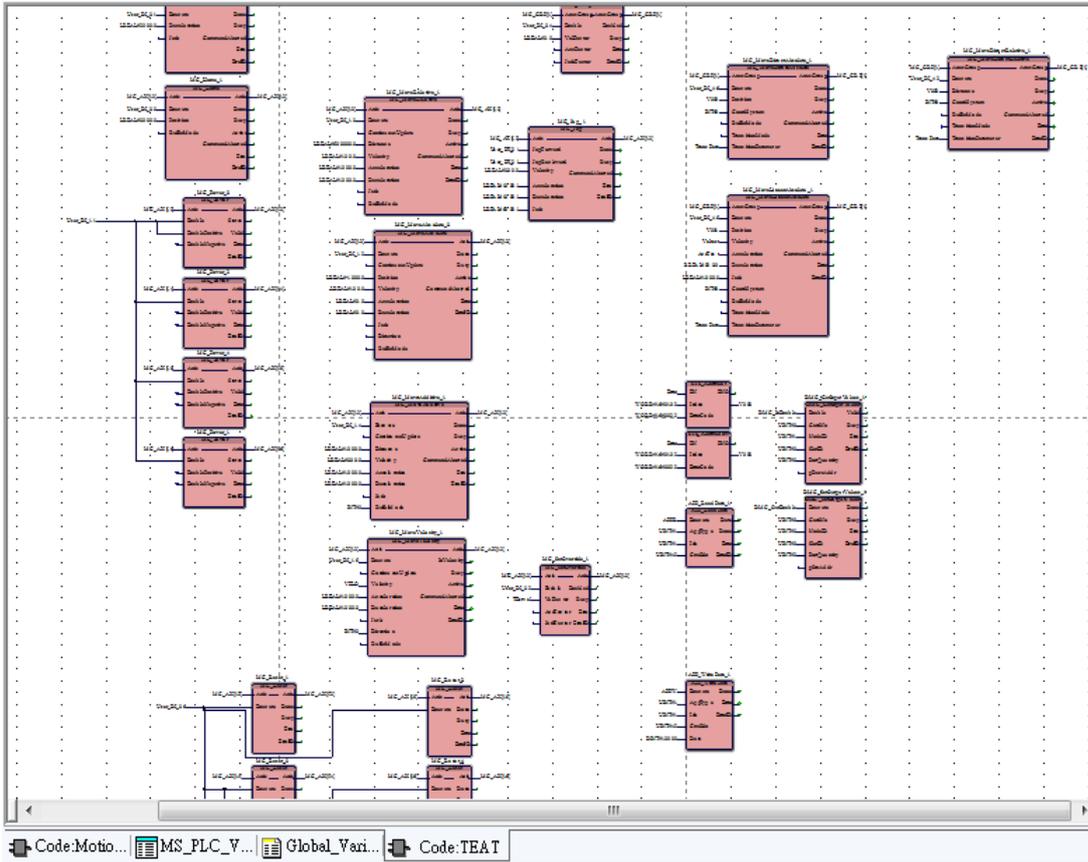


圖 2.4.1 程式編輯區

## 2

## 2.5 功能塊函式庫

此視窗顯示目前使用的函式庫中所有的功能塊。從函式庫直接拖曳功能塊進編輯區直接編輯使用。點選[Object]選單中的[Insert Blocks with EN / ENO]選項，可以讓功能塊增加 EN / ENO 腳位，方便邏輯控制。基礎功能塊定義可以參考 [?] 選單中的 [Help on Standard FB/FU]，符合 PLCOpen 定義的功能塊，詳細說明可參考本手冊第五章。

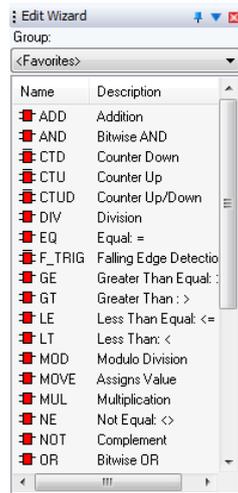


圖 2.5.1 功能塊函式庫

## 2.6 狀態顯示區

此區塊主要進行狀態的顯示，[Message Window]提供編譯的錯誤以及警告資訊，[Watch Window]監控變數的數值變化，[Logic Analyzer]可以同時觀察多個變數，並將變數繪製成曲線圖，方便觀察各變數變化狀況。最底下狀態列顯示目前的語言，及磁碟目前剩餘空間。

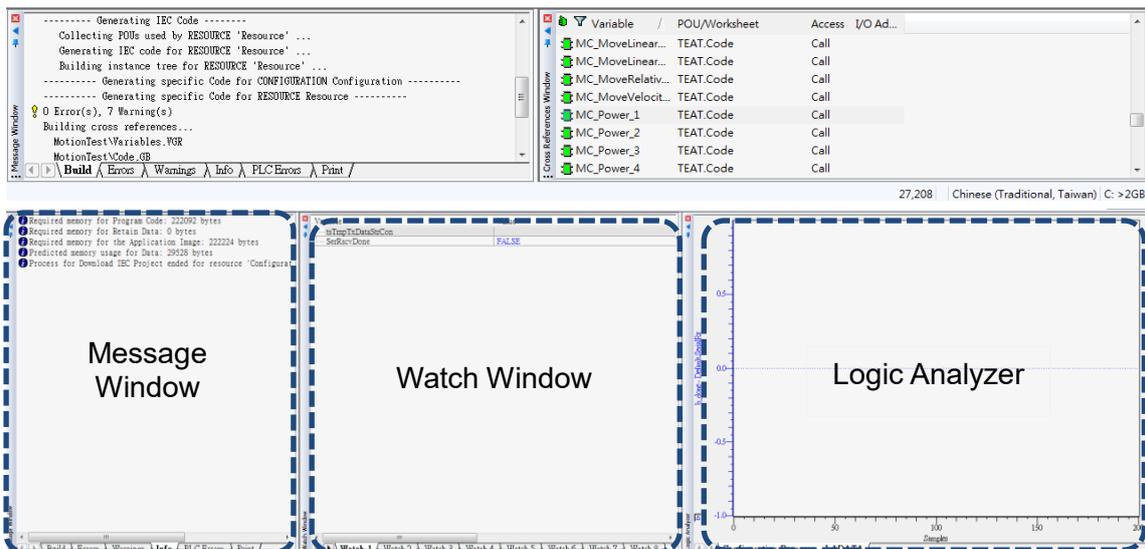


圖 2.6.1 狀態顯示區

# 3

## IEC61131-3 介紹

---

此章節介紹 IEC61131-3 此 PLC 規範，並介紹此規範定義的軟體模型階層。

3.1 IEC 61131-3 介紹 .....	3-2
3.2 Configuration (配置) .....	3-4
3.3 Resource (資源) .....	3-5
3.4 Task (任務) .....	3-8
3.5 POU (程序組織單元) .....	3-13
3.6 Data Type (資料型態) .....	3-19

# 3

## 3.1 IEC 61131-3 介紹

早期 PLC 控制程式軟體大多使用階梯圖程式(Ladder Diagram, LD)語言，其語法簡單易學，因此被廣泛使用於 PLC 應用程式之開發。

隨著眾多廠商紛紛投入可程式控制器的開發，使得可程式控制器的語法越來越多，造成使用者在不同廠牌之間程式轉換不便。因此，1993 年國際電工委員會(IEC)制定了可編程控制器的國際標準 IEC 61131，其中的第三部分關於編程語言的標準，規範了可編程控制器的編程語言及其基本元素。

IEC61131-3 標準的軟體模型採用層次結構來表示，各層的關係如圖所示。

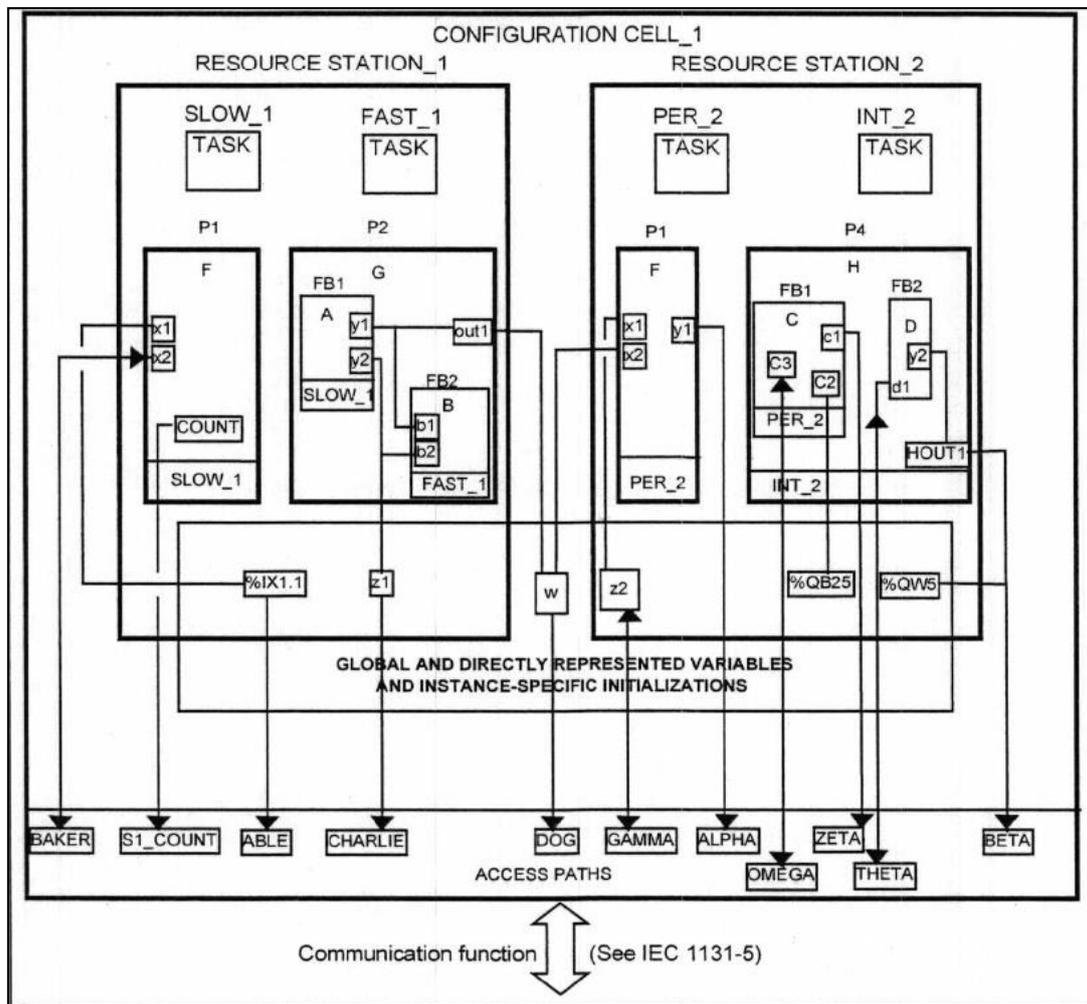


圖 3.1.1 IEC61131-3 軟體架構各層關係圖(出處: INTERNATIONAL STANDARD IEC 61131-3)

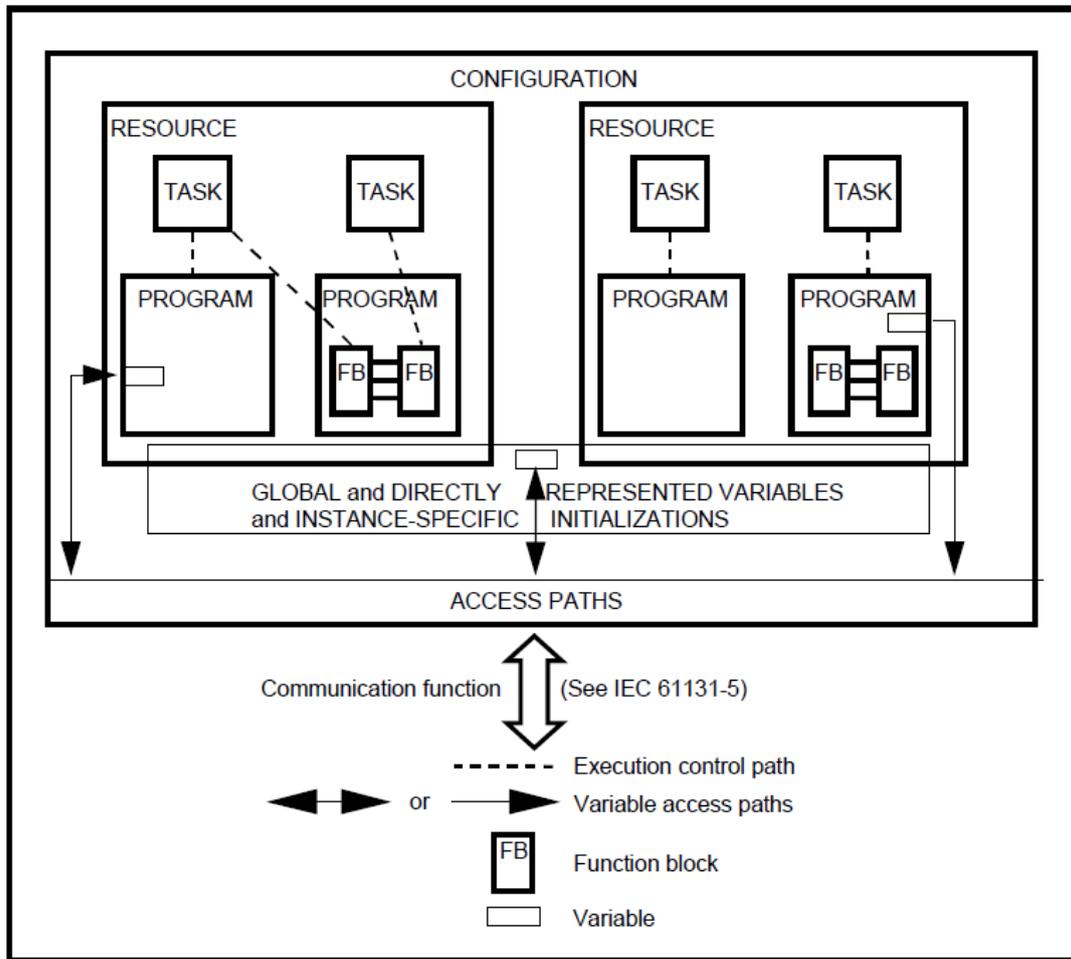


圖 3.1.2 IEC61131-3 軟體架構示意圖 (出處: INTERNATIONAL STANDARD IEC 61131-3)

MULTIPROG 遵循了 IEC61131-31 標準定義的層次結構。透過 MULTIPROG 開啟專案後，左側的專案樹窗口中會列出當前專案所用到的所有節點信息，如圖所示。

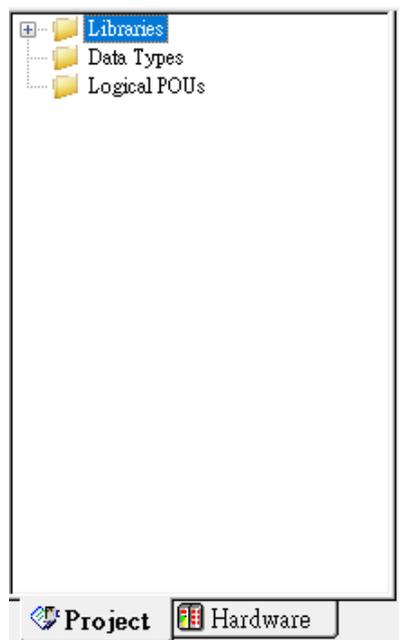


圖 3.1.3 專案樹窗口 Project 頁面

## 3

點擊專案樹窗口下方的 Hardware 選項，切換至[Hardware]頁面，並點擊 Physical Hardware 節點，即可展開 Physical Hardware 的配置(Configuration)。使用者可以看到 Configuration (配置)、Resource (資源)、Task (任務)、Global\_Variables 和 IO\_Configuration 這幾個節點，如圖所示。其中 Task 節點預設有兩個任務，分別為 Default 和 Motion。

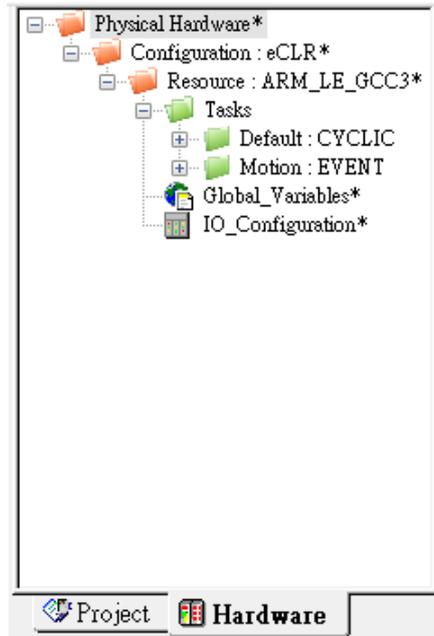


圖 3.1.4 專案樹窗口 Hardware 頁面

## 3.2 Configuration (配置)

配置是一種結構元素，對應於 IEC61131-3 標準中最外層的框架，用來定義控制系統的特性，包括硬體設備、中央處理器資源、I/O 通道、儲存和通信地址等。通常一個配置對應一台實際的控制器。當需要處理的控制問題較複雜時，則需要幾台控制器的相互配合，此時可以定義多個配置，每個配置都是相互獨立的個體，不過可以通過通信接口實現信息的交換。

目前 ASDA-MS 配合的 MULTIPROG 5.50 Express 版本僅提供一組配置 eCLR 設定，如果需要增加配置，需要使用 MULTIPROG 5.50 Pro 版本。

### 3.3 Resource (資源)

Resource(資源)節點位於 IEC61131-3 軟體架構的第二層，可支援多個運行的程序。資源反映了控制器的物理結構，為程序和 PLC 的物理輸入輸出通道提供一個接口。一個資源可以增加並執行多個相互獨立的程序。

資源可以通過資源名稱來加以區別，一個資源相當於一個 CPU，所以資源也可以簡單理解為控制器裡的中央處理器單元。

目前 ASDA-MS 配合的 MULTIPROG 5.50 Express 版本僅提供一組資源設定，如果需要增加資源，需要使用 MULTIPROG 5.50 Pro 版本。

#### ■ Settings

滑鼠右擊 Resource 節點，在彈出的選單中選擇 Settings...，如圖所示，其彈出的視窗如圖所示。

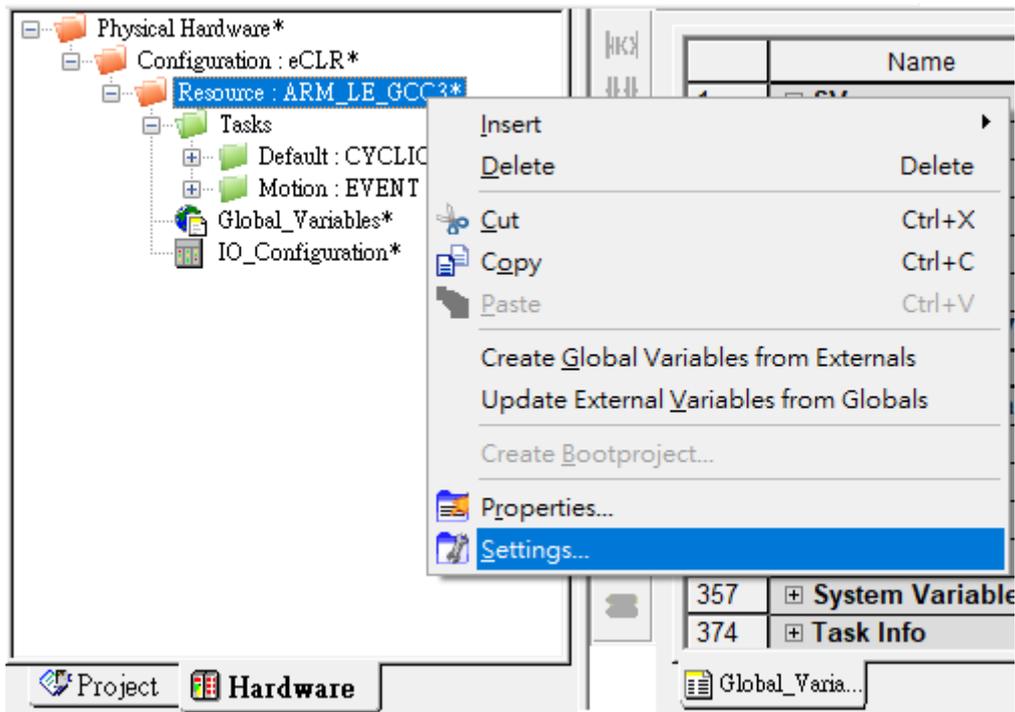


圖 3.3.1 Resource Setting 路徑

## 3

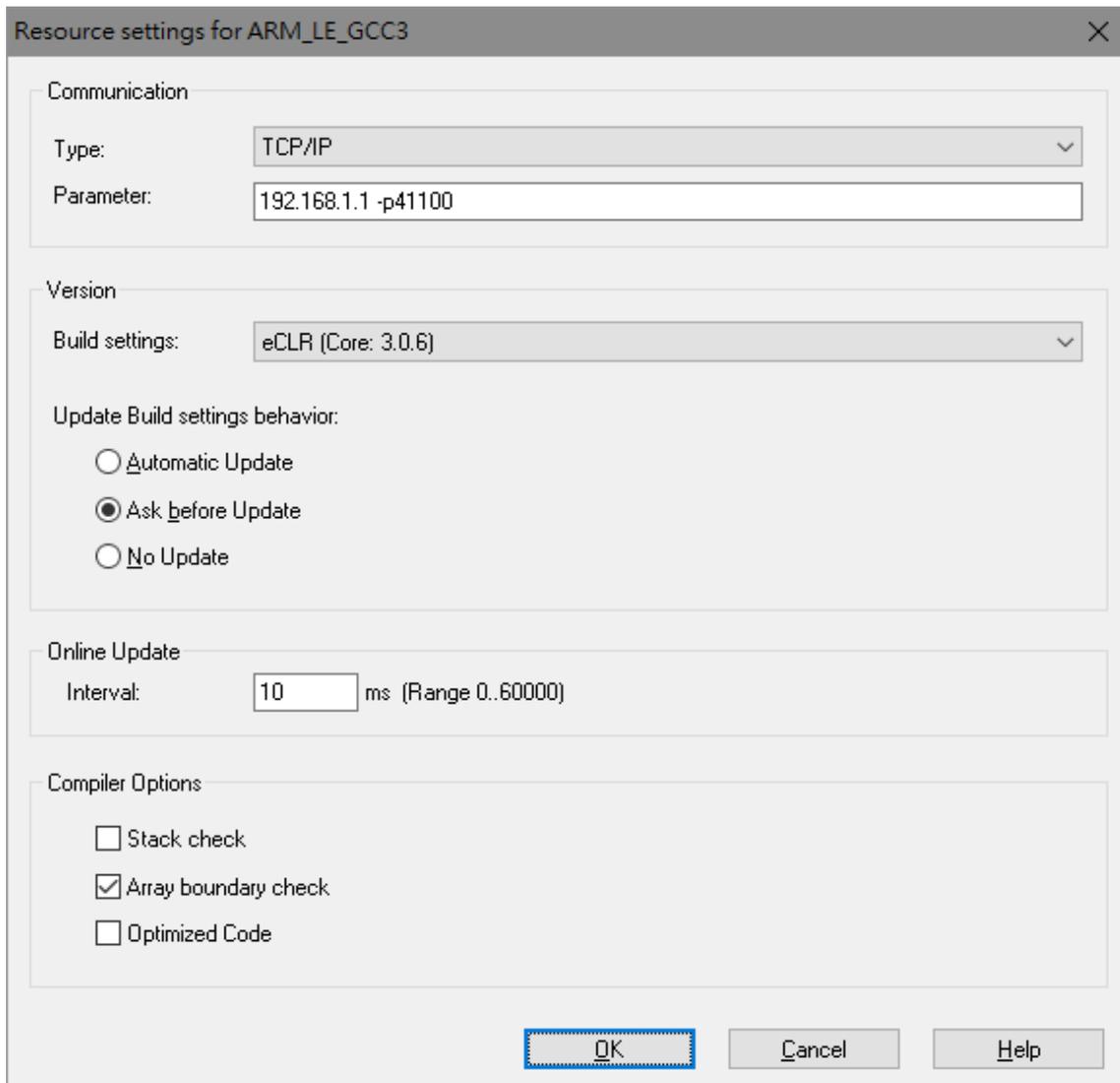


圖 3.3.2 Resource Setting 頁面

MULTIPROG 透過 TCP/IP 模式與 ASDA-MS 通訊，使用者可以透過更改 Parameter 的 IP 位址來連接不同的 ASDA-MS 設備。

Online Update 的時間間隔指的是在使用 MULTIPROG 的 Debug 模式時，控制器內部變數的資料在 MULTIPROG 界面上顯示的更新週期。

## ■ Global Variables

每個資源(Resource)中僅提供一組的「Global variables」節點。雙擊該節點即可開啟全域變數表，如圖所示。

	Name	Type	Usage	Description	Address	Init	Retain	DeviceMapping
1	SV							
2	Axis1ServoON	BOOL	VAR_GL...	Axis Servo On Co...	%MX3.0.0		<input type="checkbox"/>	<input type="checkbox"/>
3	Axis2ServoON	BOOL	VAR_GL...		%MX3.1.0		<input type="checkbox"/>	<input type="checkbox"/>
4	Axis3ServoON	BOOL	VAR_GL...		%MX3.2.0		<input type="checkbox"/>	<input type="checkbox"/>
5	Axis4ServoON	BOOL	VAR_GL...		%MX3.3.0		<input type="checkbox"/>	<input type="checkbox"/>
6	Axis5ServoON	BOOL	VAR_GL...		%MX3.4.0		<input type="checkbox"/>	<input type="checkbox"/>
7	Axis6ServoON	BOOL	VAR_GL...		%MX3.5.0		<input type="checkbox"/>	<input type="checkbox"/>
8	Axis13ServoON	BOOL	VAR_GL...		%MX3.12.0		<input type="checkbox"/>	<input type="checkbox"/>
9	Axis14ServoON	BOOL	VAR_GL...		%MX3.13.0		<input type="checkbox"/>	<input type="checkbox"/>
10	Axis15ServoON	BOOL	VAR_GL...		%MX3.14.0		<input type="checkbox"/>	<input type="checkbox"/>
11	Axis16ServoON	BOOL	VAR_GL...		%MX3.15.0		<input type="checkbox"/>	<input type="checkbox"/>
12	Axis1FaultRst	BOOL	VAR_GL...	Axis Fault Reset ...	%MX3.64.0		<input type="checkbox"/>	<input type="checkbox"/>
13	Axis2FaultRst	BOOL	VAR_GL...		%MX3.65.0		<input type="checkbox"/>	<input type="checkbox"/>
14	Axis3FaultRst	BOOL	VAR_GL...		%MX3.66.0		<input type="checkbox"/>	<input type="checkbox"/>
15	Axis4FaultRst	BOOL	VAR_GL...		%MX3.67.0		<input type="checkbox"/>	<input type="checkbox"/>

圖 3.3.3 Global Variables 頁面

每個資源中僅提供一組全域變數表，在其所屬的資源中存在的任何一個程序都可以存取全域變數表中的變數，使同一資源中的不同程序可以交換數據和共享資料。

全域變數工作單中比較常用的是 Name、Type、Address 和 Init4 個屬性。

Address 用來表示該變數是否和特定的物理地址相關聯，關聯有兩種類型：一種是和實體 I/O 模組相關聯，另外一種是和記憶體中的某一地址關聯。

以「%M」開頭表示該變數存在於控制器的共享記憶體中，後面緊跟著的一個字母表示數據的類型。第一位數字表示共享記憶體的第幾個區域，目前只支持 0、1、3 這三個數字，其中 M0 與 M1 為系統所使用，使用者只能使用 M3 區域。第一個小數點後的一串數字表示該變數在共享內存區域的位元組(Byte)偏移量。如果數據的類型是位元(對應字母為 X)，則有第二個小數點，其後的一個數字代表在前面一串數字表示的偏移量所定位的一個位元組中的第幾位，範圍為 0~7。

以「%I」開頭的表示該全域變數關聯到實體上的一個輸入，後面緊跟著的一個字母表示數據類型，其後的一串數字表示從輸入起始到該變數所在地址的位元組偏移量。如果數據類型為一個位元，則有一個小數點，其後的數字表示位元的偏移量。

關聯到實體上輸出的全域變數的地址命名規則和關聯到輸入的變數相同，只是以「%Q」開始。

## 3

**■ IO Configuration**

當有變數與實體輸入輸出地址相關時，就需要使用 I/O 設置。雙擊專案樹中 I/O\_Configuration 節點，就可以彈出 I/O 設置窗口，目前可以檢視設定結果，可對這些變數的屬性和描述進行更改。

設定配置主要是搭配 ASDA-MS 內部晶片參數設定，目前 ASDA-MS 僅提供兩組 Input 和兩組 Output 的定義。

**3.4 Task (任務)**

任務(Task) 位於 IEC61131-3 軟體架構的第三層，是程式規劃的最小單位，其用於管理載入程式組織單元(POU)的類型和運行週期(POU 的介紹將在下一章說明)。為了能更清楚地區分不同功能的任務，方便編程人員使用，MULTIPROG 定義了 3 種任務類型，分別是 Default、Cyclic、Event，這些任務的優先權由低到高分別為 Default、Cyclic、Event。

**■ Task 任務類型**

**Default 任務：**與傳統 PLC 中運行的程序相同，當控制器上電之後便一直重複循環運行，可以被其他任務打斷。

**Cyclic 任務：**此類型任務為周期性的任務，此類型任務具有一個重要參數「運行間隔」(Interval)，每隔這個間隔時間，任務即被呼叫運行，可以被 Event 任務插斷。

**Event 任務：**此類型任務是根據事件而觸發的，在控制器運行時該任務處於非運行狀態，一旦對應的事件被觸發，該任務就會開始運行。該類型的任務通常用於處理較緊急的事件。目前 ASDA-MS 僅提供一組 Event 任務用來實現運動控制，因此使用者無法自行增加 Event 任務。

### ■ 增加刪除任務

一個資源可以建立多個任務的運行，這些任務可以同時運行。使用者可以通過 **Tasks** 節點上點擊滑鼠右鍵，然後在選單中選擇 **[Insert] > [Task]** 來增加任務，如圖所示。

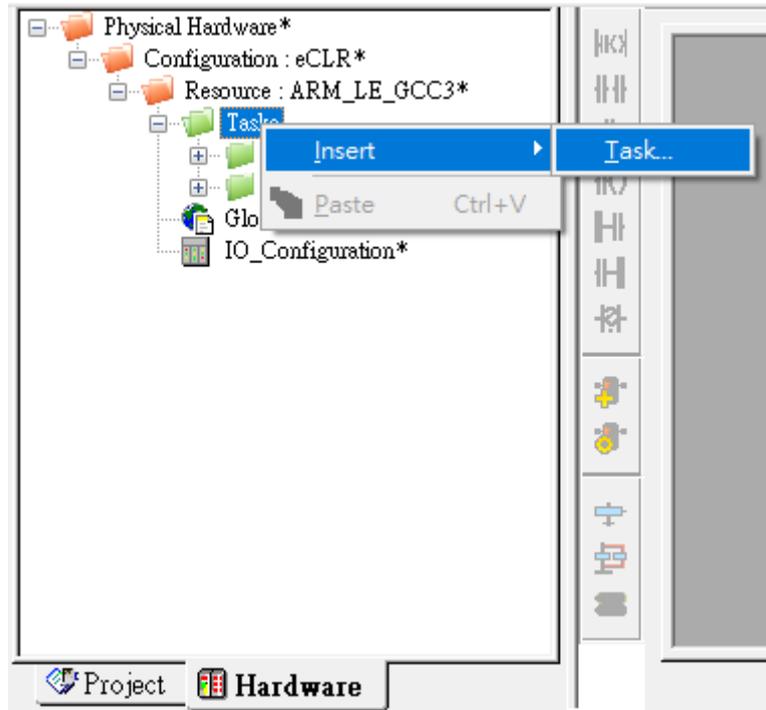


圖 3.4.1 添加任務

在彈出的視窗中寫入任務的名稱並選擇任務的類型，點擊 **OK**，如圖所示。

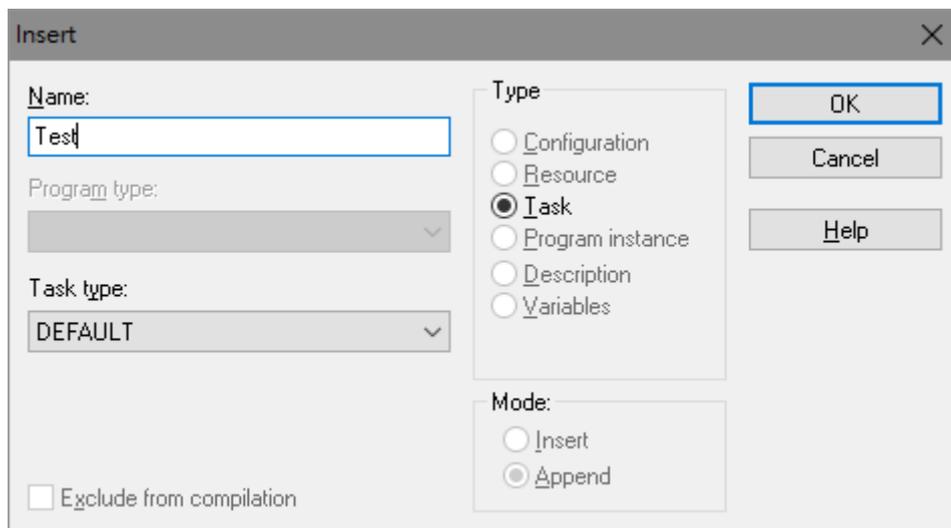


圖 3.4.2 任務屬性設置

當任務不再需要時，可以在專案樹節點上點擊右鍵，在彈出的選單中選擇 **Delete** 將任務刪除。

## 3

## ■ Task 屬性修改

滑鼠右鍵點擊專案樹中 Tasks 節點中的某個任務的節點，在選單中選擇 Properties...，如圖所示。

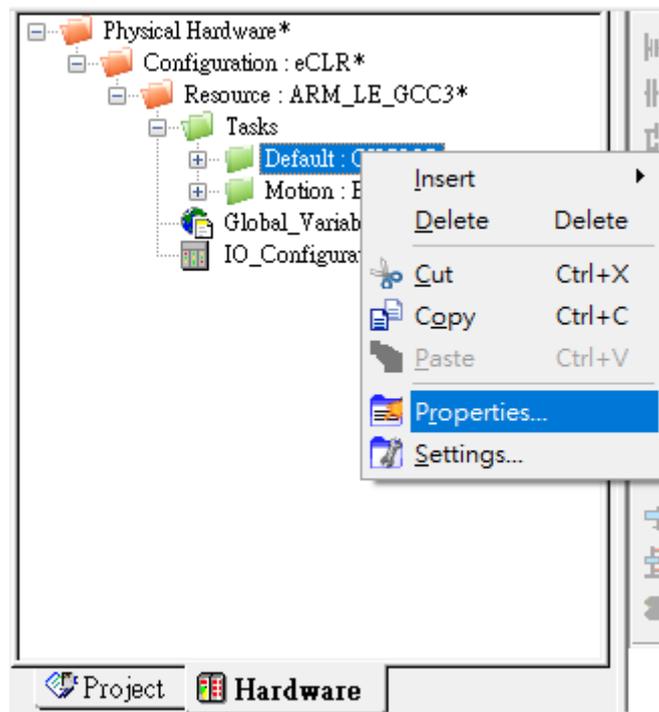


圖 3.4.3 修改任務屬性

在彈出視窗中選擇 Type 標籤頁，在 Task Type 下拉選單中即可選擇任務的類型，如圖所示。

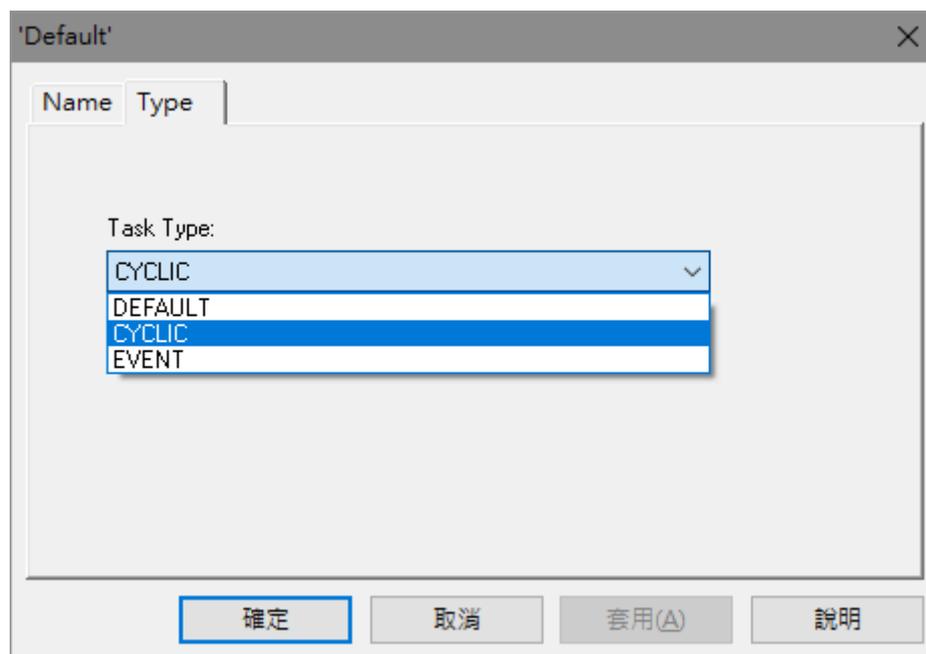


圖 3.4.4 任務屬性窗口

不同的任務類型，所需要設置的參數也不同，下面分別進行說明。在需要更改參數的任務節點上點擊滑鼠右鍵，在彈出選單中選擇 **Settings...**，如圖所示。

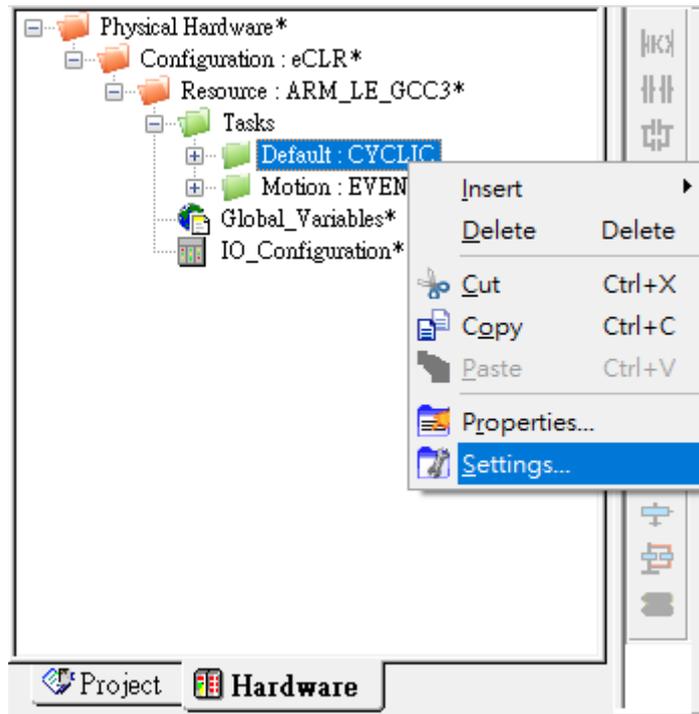


圖 3.4.5 任務屬性路徑

#### Default Task :

圖示為 Default 任務的參數設定視窗。Watchdog Time 為監視計時器的計時時間，當勾選了 Enable Watchdog 監視計時器後，監視計時器就會根據設定的 Watchdog Time 開始計時。當計時到達之後，監視計時器就會對運行中的任務發出信號，若運行的任務沒有回應，則監視計時器會認為任務運行發生異常，此時就會產生一個運行期異常，對應的 System Task 就會被觸發。(System Task 為處理緊急事件，不開放給使用者編輯，在此就不贅述相關設定。)

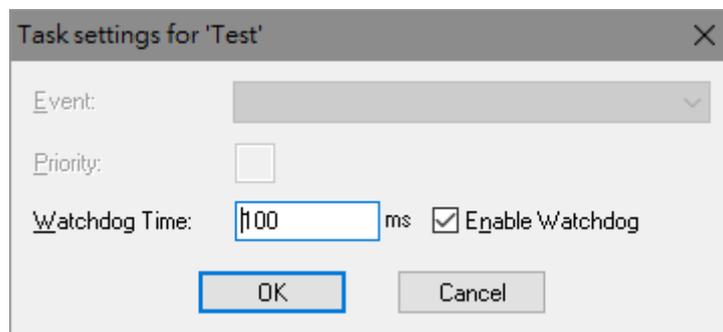


圖 3.4.6 Default 任務設定

## 3

## Cyclic Task :

圖示為 Cyclic Task 的參數設置視窗，其中需要設置的參數有 3 個：Interval、Priority 和 Watchdog Time。Interval 為任務的運行週期，Priority 為任務的優先順序，只是這個優先順序是指所有的 Cyclic Task。可以設定的優先權範圍是 0 ~ 15，其中優先序數值愈小表示優先權愈高。

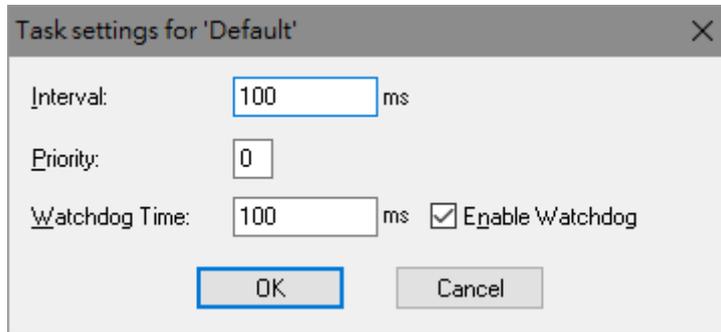


圖 3.4.7 Cyclic 任務設定

## Event 任務 :

圖示為 Event Task 的參數設定視窗，和 Cyclic Task 不一樣，Event Task 主要是設定該任務所對應的事件編號，ASDA-MS 預設 Motion Task 對應的事件編號為 0，由於 ASDA-MS 僅提供一組 Event Task，在此就不贅述相關設定。

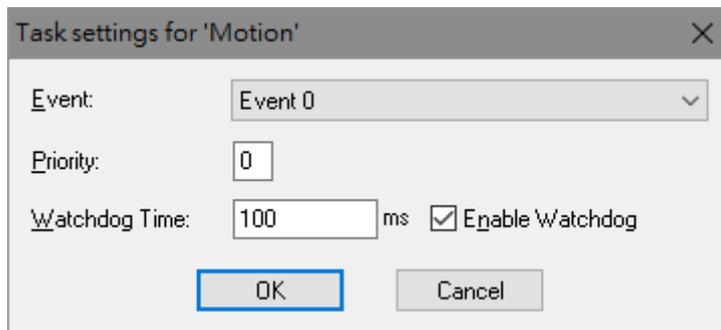


圖 3.4.8 Event 任務設定

### 3.5 POU (程序組織單元)

程序組織單元(Programming Organization Unit · POU)是 IEC61131-3 標準中最基本的程序單元，其中包括了程序(Program)、功能塊(Function Block)和功能(Function)三種，各有不同的特性和權限。

程序(Program)：屬於撰寫程序的主程式，內容可以包含 IO 的使用配置、全域變數和區域變數的定義，是程序組織單元的最大形式。同時程序也是三種 POU 中擁有最大的呼叫權限，可以呼叫功能塊(Function Block)與功能(Function)在程序中使用。

功能塊(Function Block)：功能塊是一種函數，一個功能塊擁有專屬於自己的記憶體空間，可以存放變數值。功能塊可以呼叫其他功能塊及功能來使用。

功能(Function)：功能是只有一個回傳值的函數，而且沒有自己的記憶體空間。只能計算某些經過設定的運算，功能只能呼叫功能，而不能呼叫功能塊。

#### ■ 增加程序、功能塊、功能

滑鼠右鍵點擊專案樹中 Logical POUs 節點，在選單中選擇[Insert] > [Function Block]或是其他想要增加的 POU 類型，如圖所示。

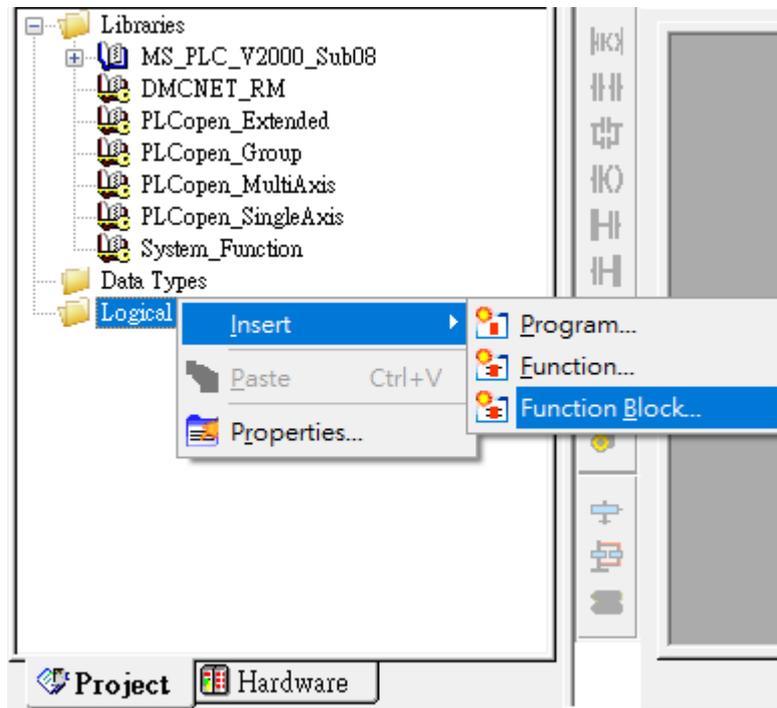


圖 3.5.1 建立 POU

在彈出的視窗中輸入功能塊的名稱，選擇需要的編程語言，點擊 **OK**，如下圖所示。

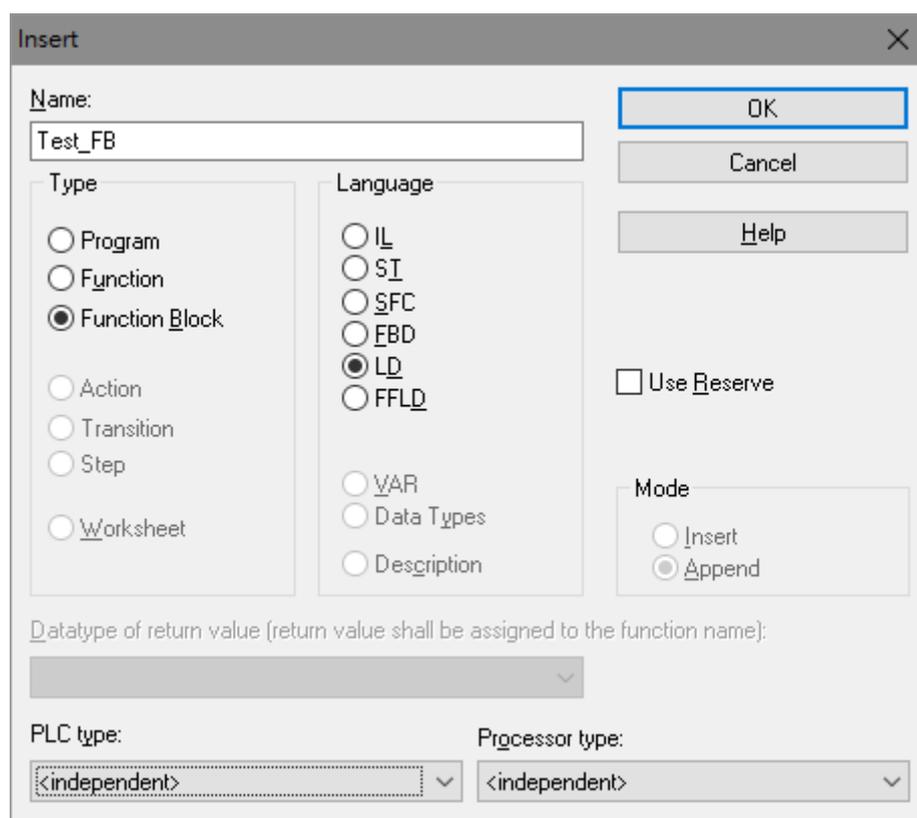


圖 3.5.2 POU 設定

刪除 POU 也是一樣，在不需要使用的 POU 節點上點擊滑鼠右鍵，在選單中選擇 Delete 即可。

### ■ 功能塊實例化

在 Logical POU 中使用功能塊的表單編寫完一個函數之後，僅代表宣告了此功能塊的定義，並不能直接在控制器中運行。如果要使其能真正運行，就必須要實例化(Instance)。功能塊實例化之後就產生這個功能塊所需要的專屬於記憶體空間。

也就是說功能塊的實例化是通過程序對功能塊的呼叫來實現的。在 MULTIPROG 中進行實例化功能塊的步驟很簡單，以 FBD 語言的程序說明，如圖 3.5.3 所示。

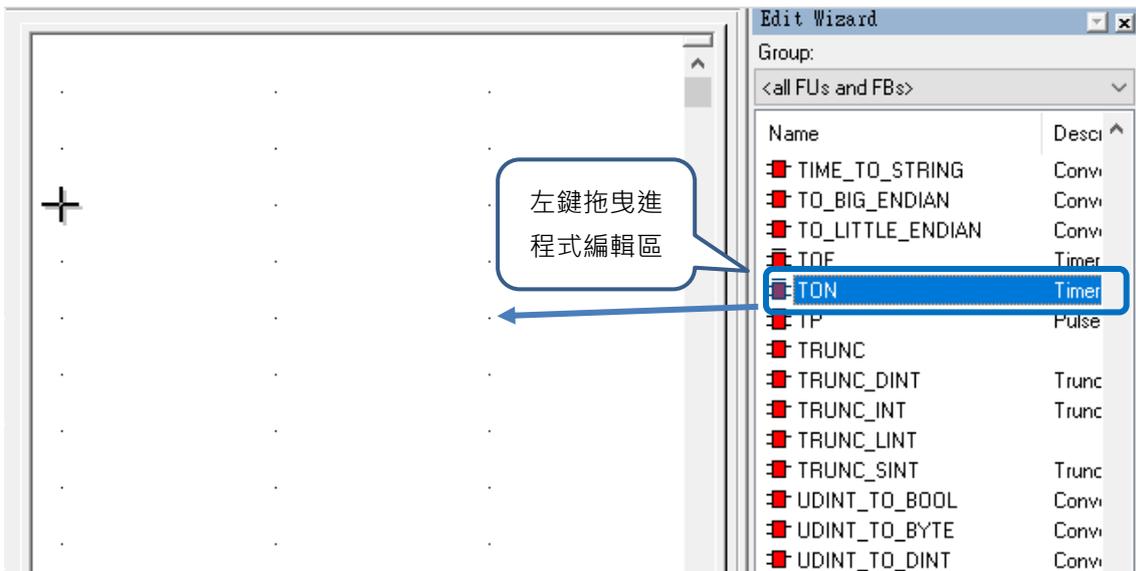


圖 3.5.3 功能塊應用案例

在 MULTIPROG 右側的 Edit Wizard 視窗中，通過下拉式選單選擇 all FUs and FBs，在任意一個功能塊圖示(以 TON 為例)上按住滑鼠左鍵不放，將圖示拖曳至程式編輯區內，然後鬆開滑鼠左鍵，會彈出屬性設定視窗，如下圖所示。

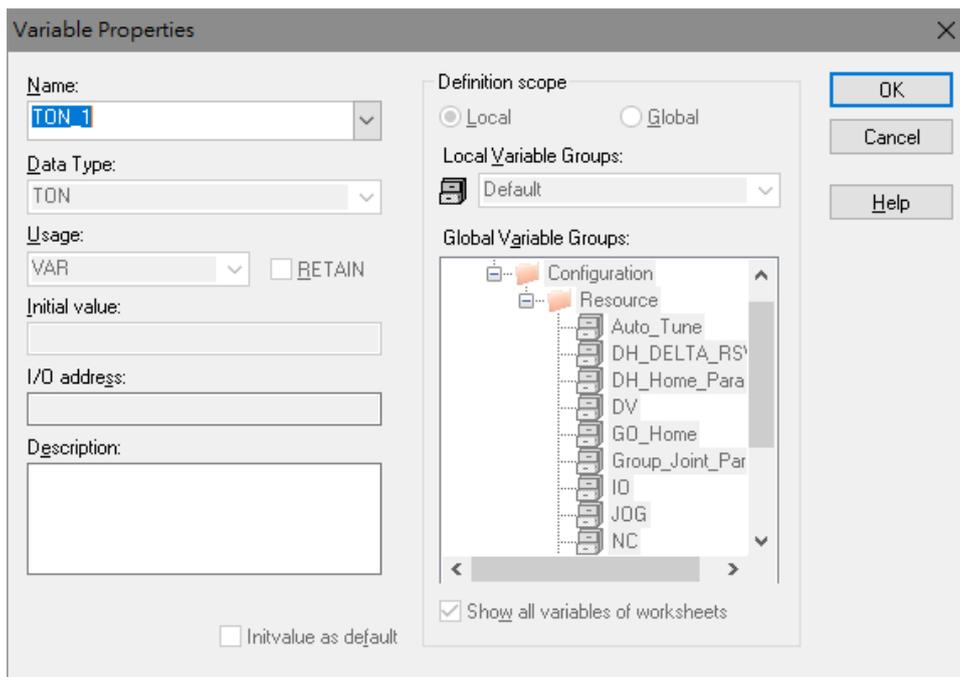


圖 3.5.4 功能塊設定

## 3

在 **Name** 預設名稱為功能塊名稱加上編號，如下圖中的 TON\_1，功能塊實例的名稱可以自行定義，不一定要包含有功能塊名稱，但不可與其他功能塊名稱相同。

點擊視窗中的 **OK** 按鈕，在程序的工作單中就出現了功能塊的一個實例，如下圖所示。在功能塊實例 TON\_1 圖示的頂部可以看到 TON 文字，表示此功能塊是通過哪個功能塊實例化產生的。

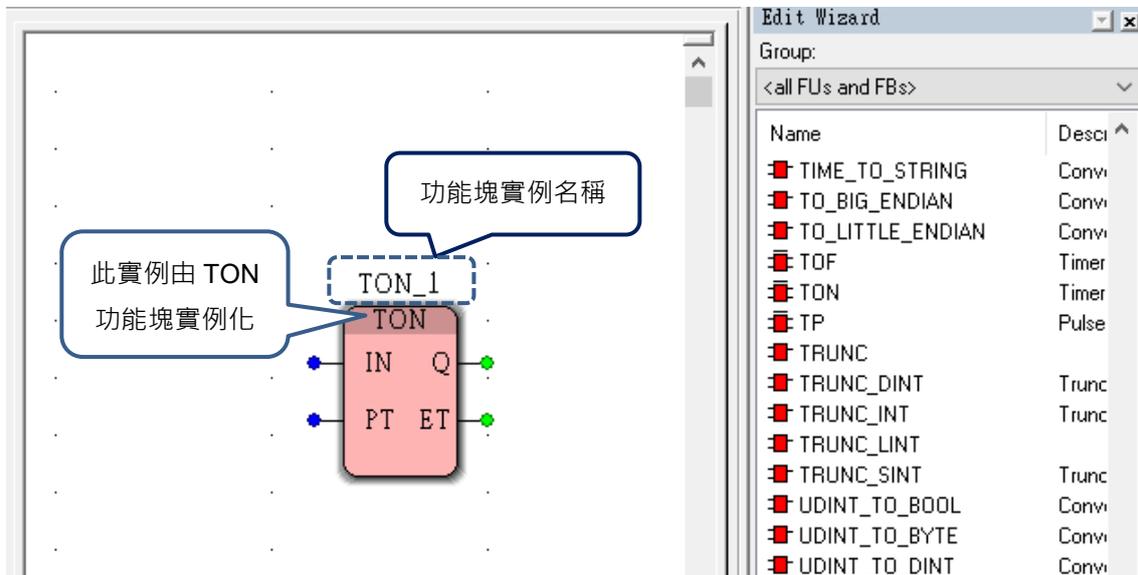


圖 3.5.5 功能塊實際範例

### ■ 程序實例化

和功能塊的情形一樣，程序僅是宣告了數據的結構和類型，不能在實際的控制器中運行。如果要使其能真正運行，同樣地要實例化(Instance)。要使程序能實際運行，同樣需要一個實例化的過程。程序的實例化是與任務相關的，MULTIPROG 會根據選擇的任務類型來分配其在控制器上所需要的記憶體空間。

在專案樹中右擊需要運行程序的任務節點，選擇選單中的[Insert] > [Program instance...]，如圖所示。

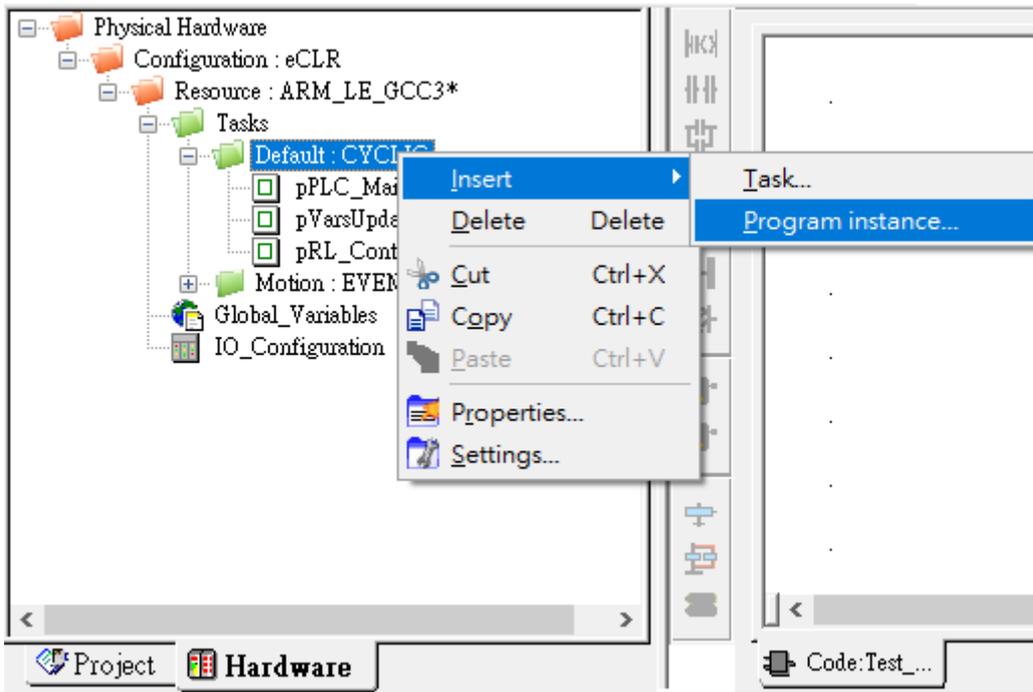


圖 3.5.6 插入程序實例路徑

在彈出視窗中輸入程序實例名稱(Program instance)，Program Type 選擇剛剛新增的程序 Test\_PROG，並在右方 Type 欄位選擇「Program instance」，如下圖所示。

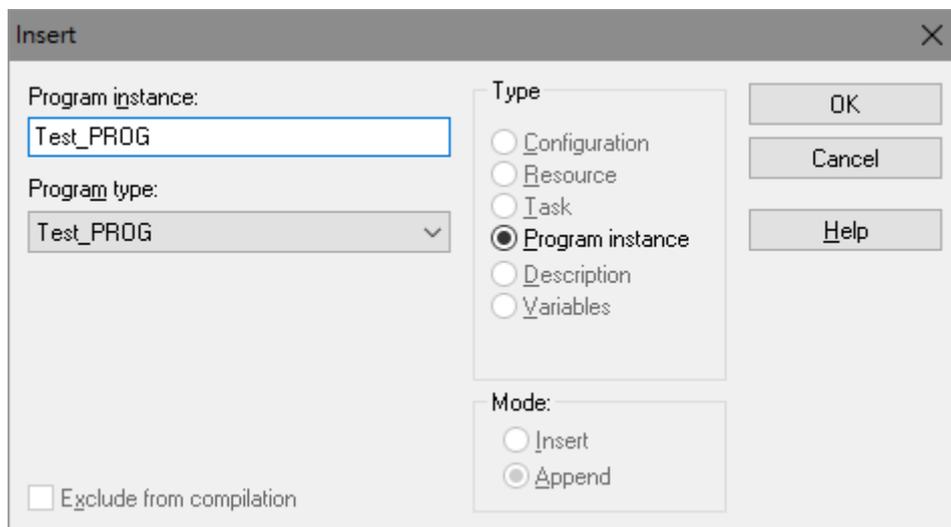


圖 3.5.7 程序實例設定

點擊 **OK** 按鈕後，就可以看到在專案樹任務節點「Default : CYCLIC」下出現節點「Test\_PROG : Test\_PROG」，如下圖所示。

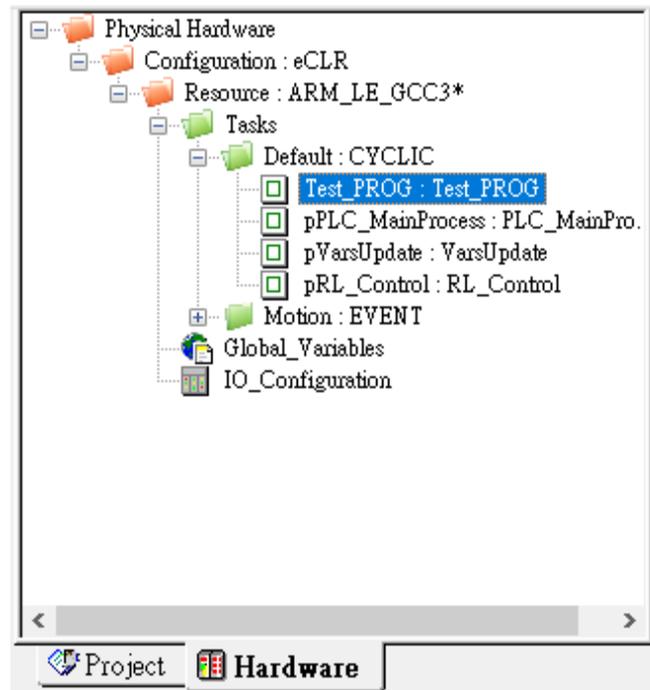


圖 3.5.8 程序實例範例

註：程序的實例和程序的名稱可以重複，但同一個任務中不能有重複的程序實例。

3

## 3.6 Data Type (資料型態)

### ■ 基本資料型態

基本資料型態是在 IEC61131-3 標準中定義好的標準化的數據類型，包括資料型態、資料長度、資料範圍和初始值，詳細如下表所示。

表 3.6.1 變數資料型態

資料型態	關鍵字	資料長度(bit)	範圍	初始值
布林	BOOL	1	0和1	0
位元組	BYTE	8	0 ~ 0xFF	
字組	WORD	16	0 ~ 0xFFFF	
雙字組	DWORD	32	0 ~ 0xFFFFFFFF	
長字組	LWORD	64	0 ~ 0xFFFFFFFFFFFFFFFF	
短整數	SINT	8	-128 ~ +127	0
整數	INT	16	-32768 ~ +32767	0
雙整數	DINT	32	$-2^{31} \sim +2^{31}-1$	0
長整數	LINT	64	$-2^{63} \sim +2^{63}-1$	0
無號短整數	USINT	8	0 ~ +255	0
無號整數	UINT	16	0 ~ +65535	0
無號雙整數	UDINT	32	0 ~ $+2^{32}-1$	0
無號長整數	ULINT	64	0 ~ $+2^{64}-1$	0
實數	REAL	32	$1.5e^{-45} \sim 3.4e^{38}$	0.0
長實數	LREAL	64	$5.0e^{-324} \sim 1.7e^{308}$	0.0
時間	TIME	-	-	T#0s
日期	DAY	-	-	D#0001-01-01
時刻	TOD	-	-	TOD#00:00:00
日期和時刻	DT	-	-	DT#0001-01-01-00:00:00
字串	STRING	80	-	空字串

## 3

## ■ 衍生資料型態

衍生資料型態是在基本資料型態的基礎上由使用者自行建立的資料型態，也就是使用者自定義的資料型態。

在 MULTIPROG 中插入使用者自定義的資料型態步驟如下：

專案樹中右擊 Data Types 資料型態，選擇選單的 [Insert] > [Datatypes]，如圖所示。

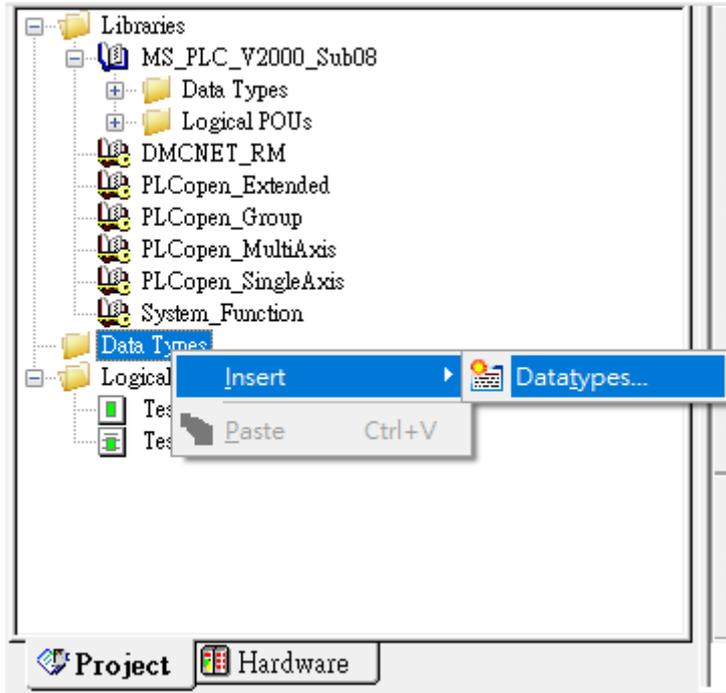


圖 3.6.1 資料型態插入路徑

在彈出的視窗中輸入名稱，點擊確定 **OK**，如圖所示。

The screenshot shows the 'Insert' dialog box with the following settings:

- Name:** User\_add
- Type:**  Program,  Function,  Function Block,  Action,  Transition,  Step,  Worksheet
- Language:**  IL,  SI,  SFC,  FBD,  LD,  FFLD,  VAR,  Data Types,  Description
- Use Reserve:**
- Mode:**  Insert,  Append
- Datatype of return value (return value shall be assigned to the function name):** [Empty dropdown]
- PLC type:** <independent>
- Processor type:** <independent>

圖 3.6.2 資料型態設定

註：此處的名稱(Name)只是資料型態工作單的名稱，與數據型態沒有關係。

## 3

在資料型態工作單中使用描述式語言進行編輯，依照 IEC61131-3 標準的規定，所有的衍生資料型態的定義都使用「TYPE」關鍵字開頭，以「END\_TYPE」關鍵字結束。

MULTIPROG 支援 4 種基本的使用者自定義資料型態，分別是列舉、範圍、陣列和結構，其對應的範例如圖 3.6.3 所示。

```
1  TYPE
2     TEST_Enum : (Value1, Value2 , Value3);
3  END_TYPE
4
5
6  TYPE
7     TEST_Range : INT(-18..100);
8  END_TYPE
9
10
11 TYPE
12     TEST_Array : ARRAY [1..4] OF LREAL;
13 END_TYPE
14
15 TYPE
16     ROBOT_POSTURE :
17     STRUCT
18         Shoulder      : BOOL;
19         Elbow         : BOOL;
20         Flip          : BOOL;
21         PS            : BOOL;
22         UF            : USINT;
23         TF            : USINT;
24         RSV1         : USINT;
25         RSV2         : USINT;
26     END_STRUCT;
27 END_TYPE
```

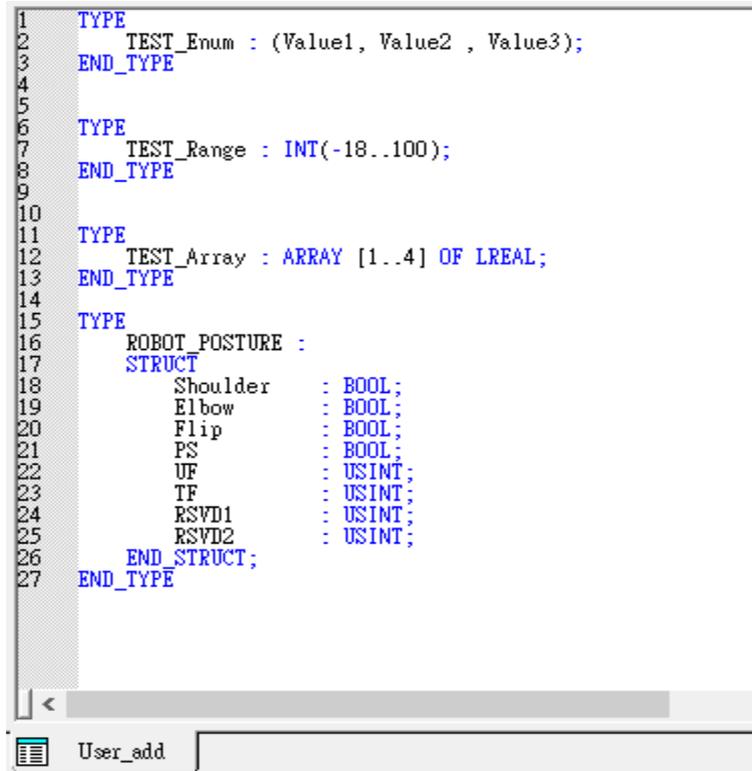


圖 3.6.3 資料型態範例

# IEC61131-3 定義編程語言

# 4

IEC61131-3 定義五種標準的 PLC 編程語言，此章節透過範例方式分別針對此五種語言進行說明。

4.1 指令表(IL)編程語言 .....	4-2
4.2 梯形圖(LD)編程語言 .....	4-4
4.3 功能塊圖(FBD)編程語言 .....	4-11
4.4 結構化文字(ST)編程語言 .....	4-16
4.5 順序功能圖(SFC)編程語言 .....	4-18

# 4

## 4.1 指令表(IL)編程語言

指令表編程語言是 PLC 程式設計最基本的語言；具有容易儲存、便於操作的特點，同時其佔用系統資源也比較少，但同時可讀性較差，比較適合在功能不複雜的小型控制系統中使用。

範例：實現四則運算  $A+B \times (C+D)$ 。

1. 在專案樹 POU 節點中插入功能塊(Function Block)，名稱為 IL\_Example，Language 選擇為 IL。

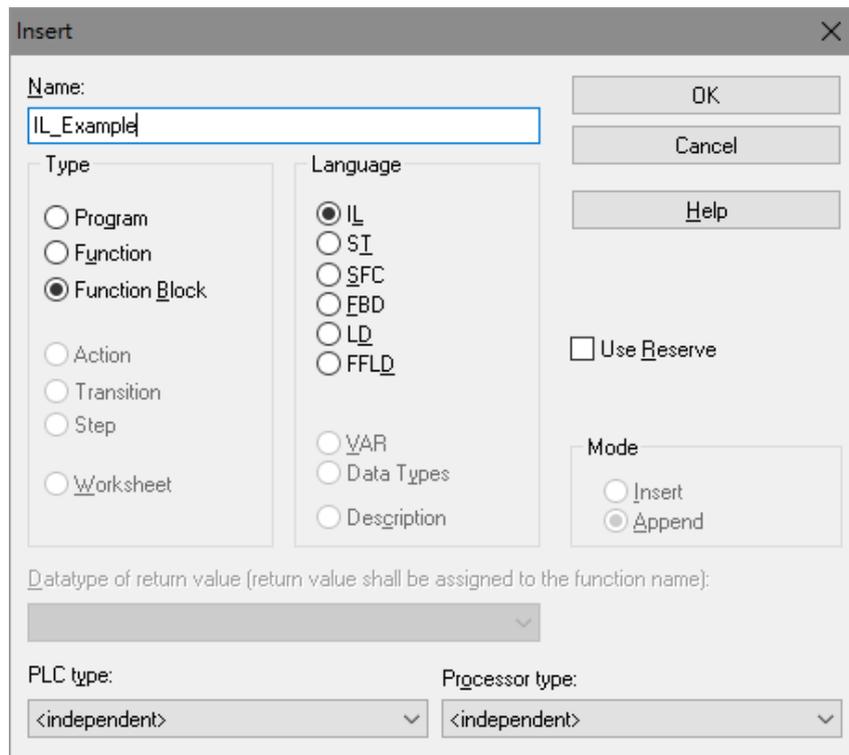


圖 4.1.1 新增 IL 範例

2. 按下 **OK** 後，在其中編寫如圖所示程式。

```

1  LD D
2  ADD C
3  MUL B
4  ADD A
5  ST X
    
```

圖 4.1.2 IL 程式範例

3. 在程式編輯區中，點擊滑鼠右鍵，在選單中選擇 **Open Variables Worksheets** 後，在其中定義如圖所示的變數。

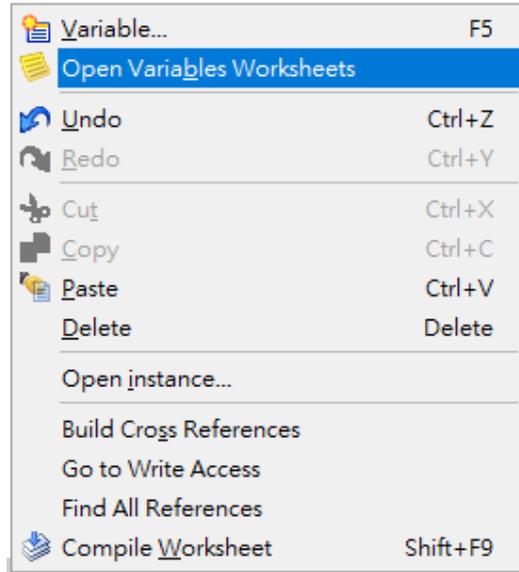


圖 4.1.3 新增變數路徑

	Name	Type	Usage	Description
1	Default			
2	A	INT	VAR_INPUT	
3	B	INT	VAR_INPUT	
4	C	INT	VAR_INPUT	
5	D	INT	VAR_INPUT	
6	X	INT	VAR_OUTPUT	

圖 4.1.4 變數宣告

4. 點擊常用工具列中的編譯(Make)按鈕圖示或是功能區 **Build** 選單中的 **Make** 功能。
5. 在程序(Program)中可以透過拖曳 **IL\_Example** 功能塊來實例化此功能塊，如下圖所示。

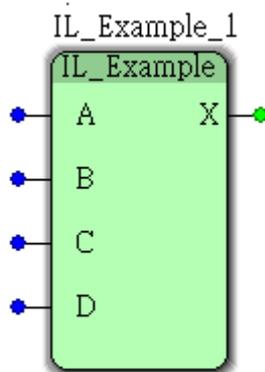


圖 4.1.5 IL 範例功能塊

# 4

## 4.2 梯形圖( LD )編程語言

梯形圖是 PLC 使用得最多的圖形編程語言，被稱為 PLC 的第一編程語言。根據梯形圖中各接點的狀態和邏輯關係，從左到右、自上而下的順序排列。

範例：伺服的啟動、保持和停止控制

1. 在專案樹 POU 節點中插入功能塊(Function Block)，名稱為 LD\_Example，Language 選擇為 LD。

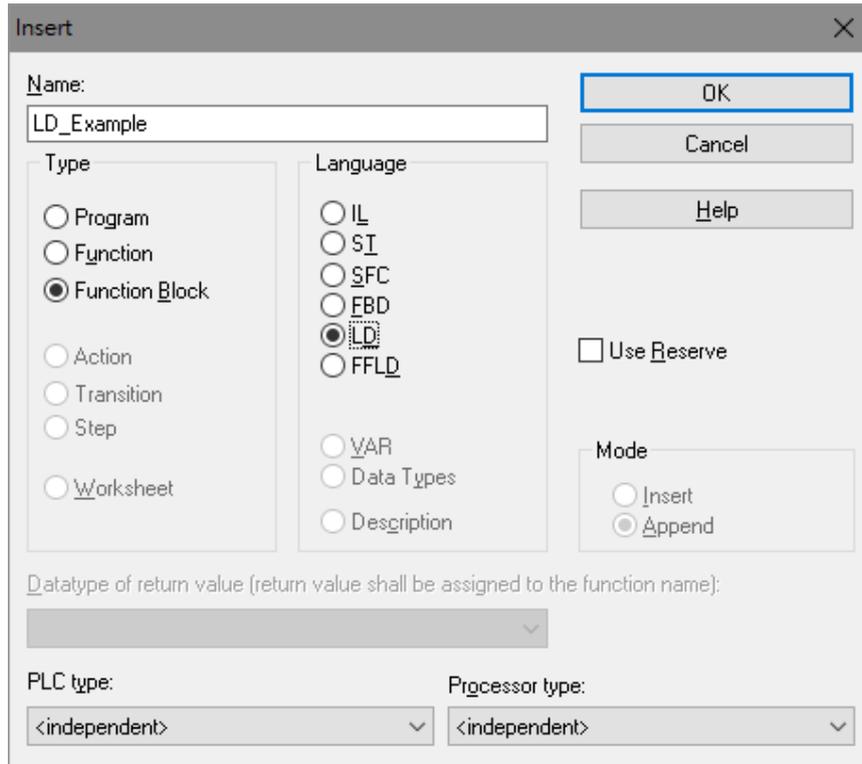


圖 4.2.1 新增 LD 範例

2. 按下 **OK** 後，在其中工作單中，點擊滑鼠右鍵，在選單中選擇 **Contact Network**，如圖所示程式。

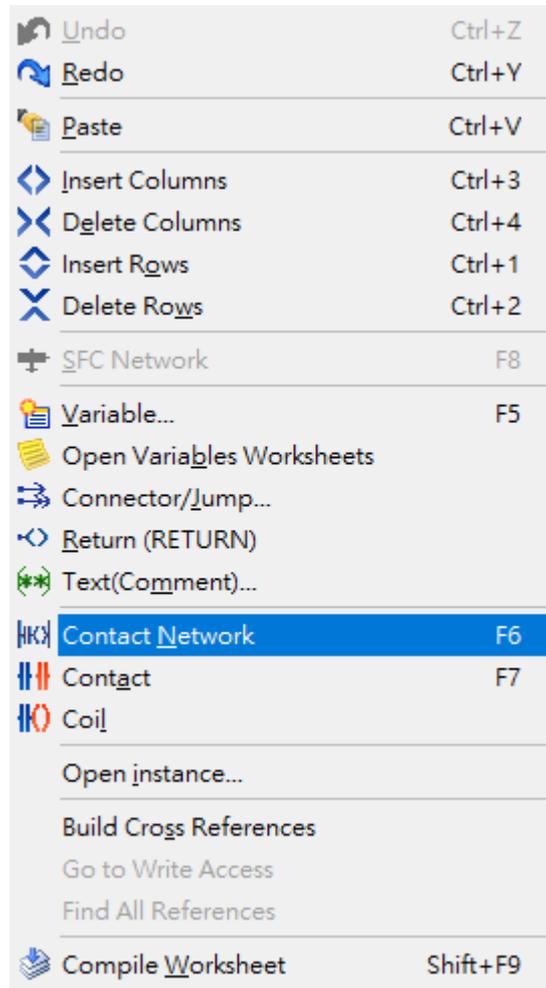


圖 4.2.2 新增 LD 元件路徑

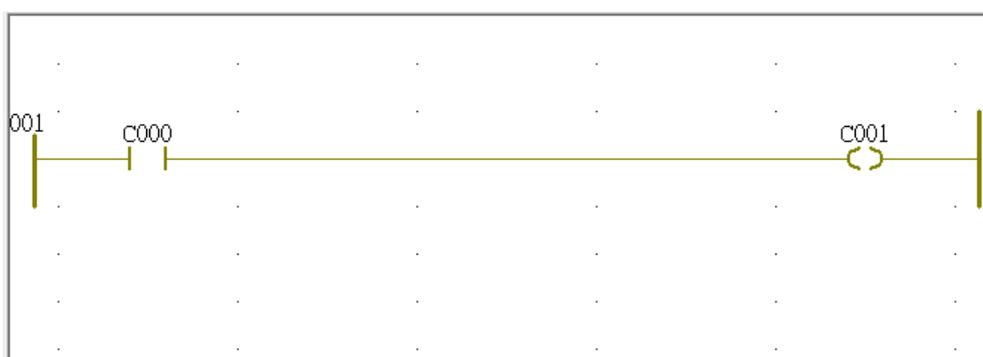


圖 4.2.3 LD 範例

3. 滑鼠雙擊 C000 接點，在彈出視窗中將其名稱修改為 Start，Usage 選擇為 VAR\_INPUT，如圖所示，然後點擊 OK

4

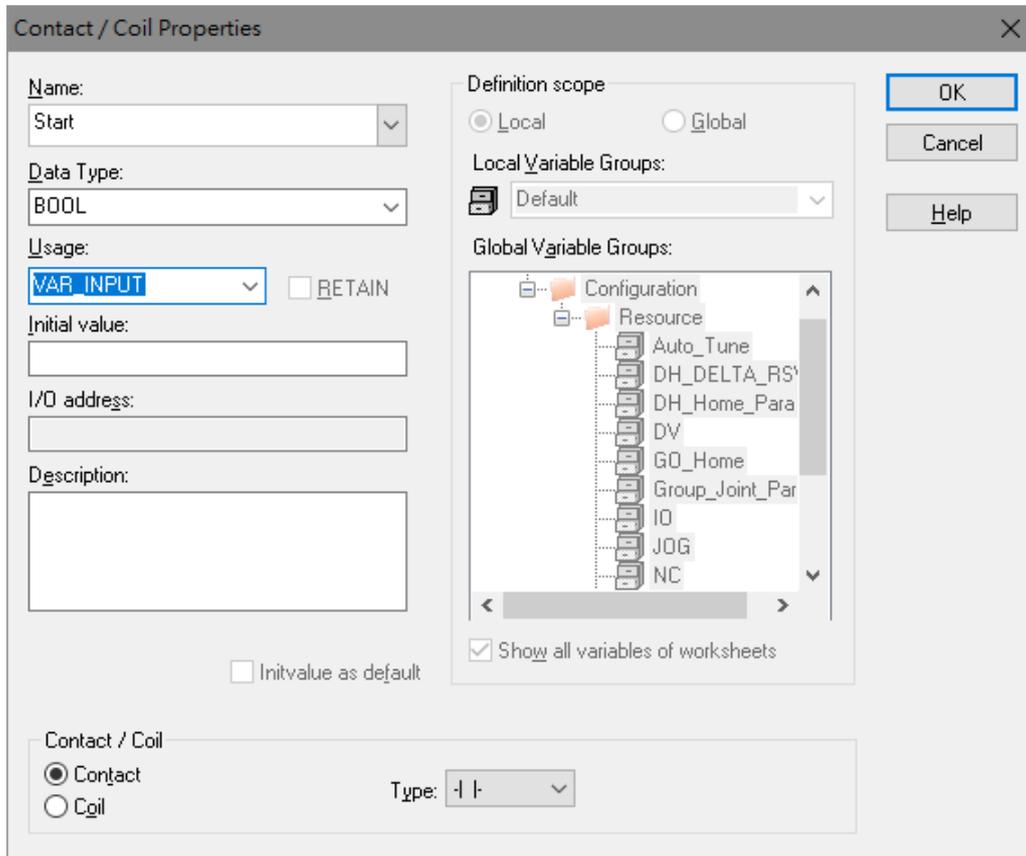


圖 4.2.4 設定接點變數

4. 滑鼠雙擊 C001 線圈，在彈出視窗中將其名稱修改為 Run，Usage 選擇為 VAR\_OUTPUT，如圖所示，然後點擊 OK。

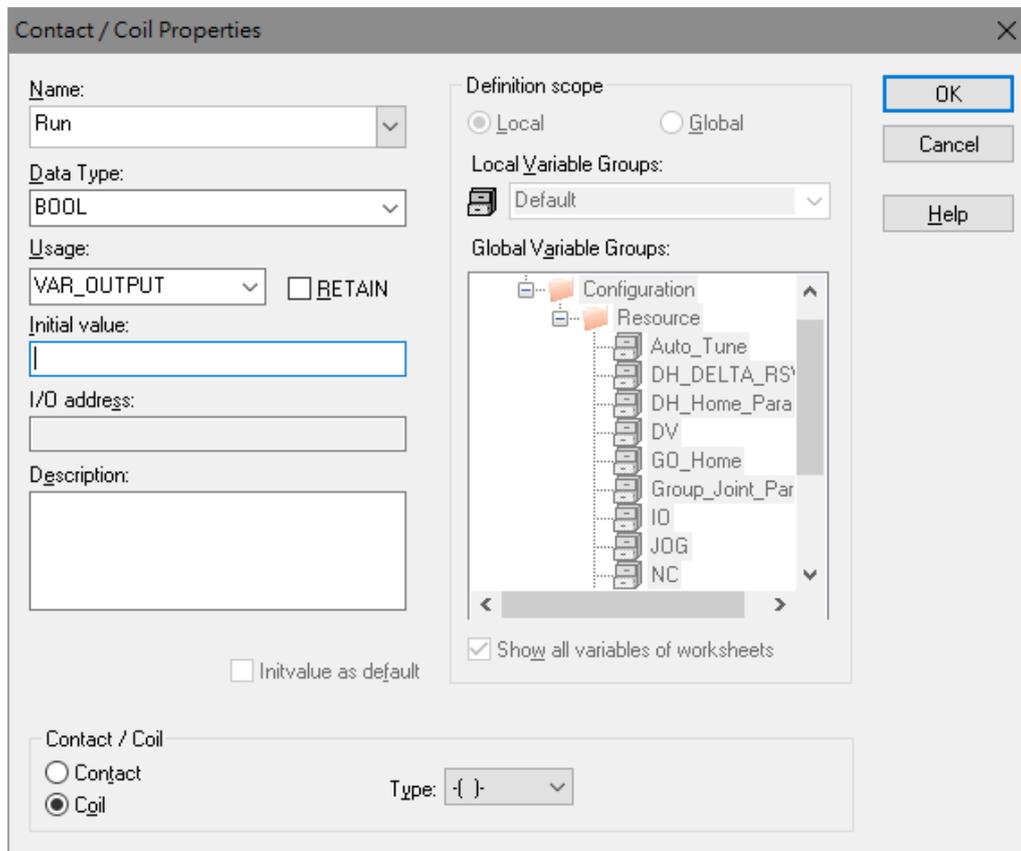


圖 4.2.5 設定接點變數

# 4

5. 點擊工作單中的 **Start** 接點，然後再點擊滑鼠右鍵選擇 **Contact Below**，在其下面插入一個平行的接點 **C002**，並將該接點改名為 **RUN**，如圖所示。

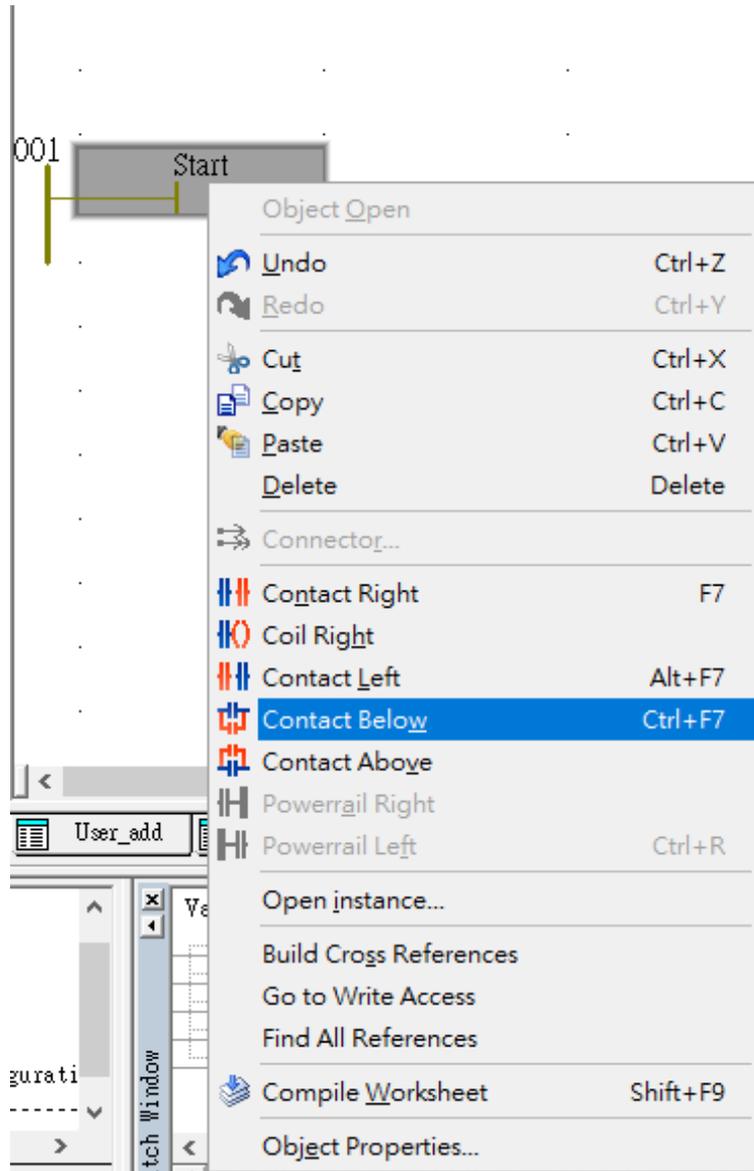


圖 4.2.6 新增 LD 元件路徑

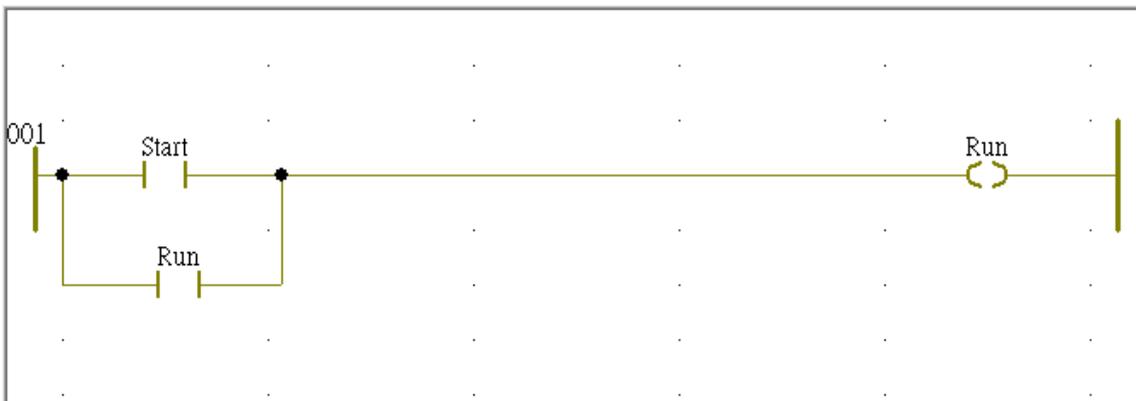


圖 4.2.7 LD 範例

6. 點擊 RUN 線圈，然後再點擊滑鼠右鍵選擇 Contact Left，在左邊插入一個接點 C003，如圖所示。

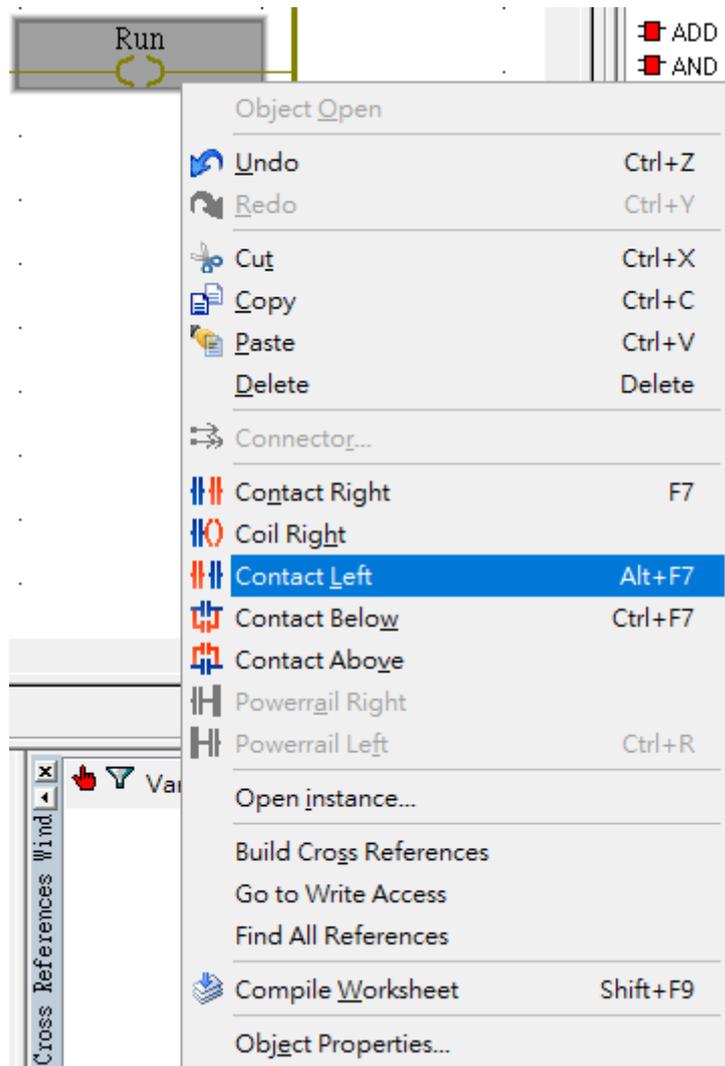


圖 4.2.8 新增 LD 元件路徑

4

7. 雙擊接點 C003，並在彈出的視窗中將其名稱改為 Stop，Usage 選擇為 VAR\_INPUT，並在視窗下方選擇 Type 為常閉接點，如圖所示。

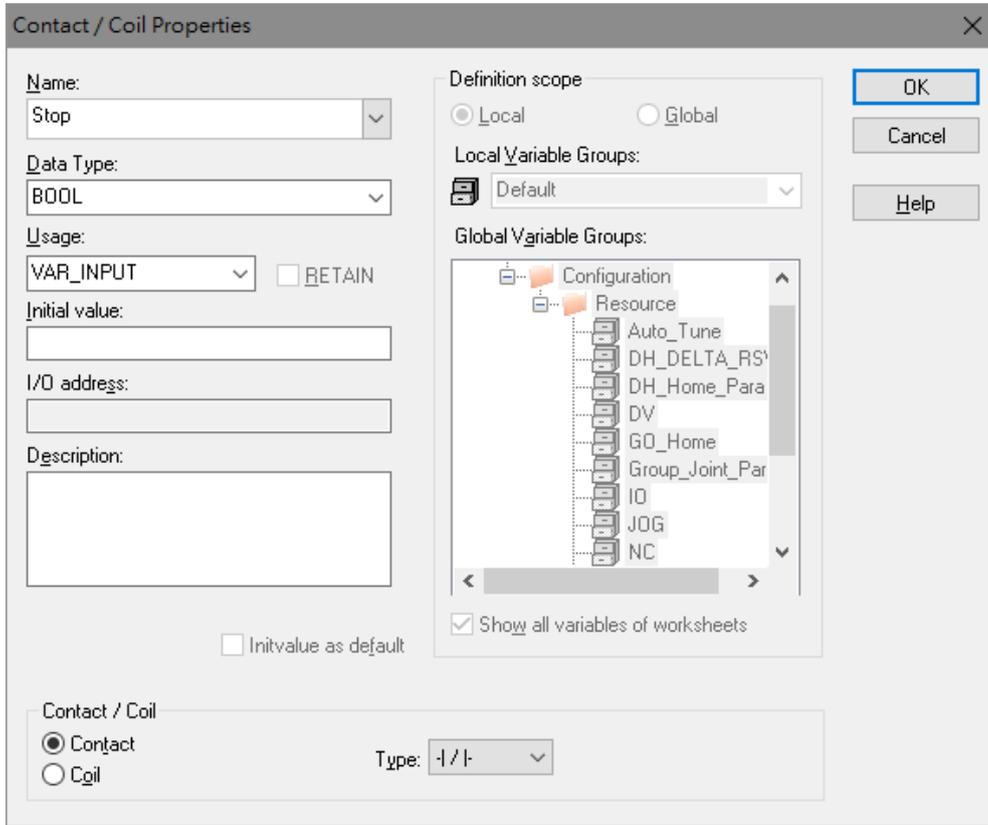


圖 4.2.9 設定接點變數

8. 點擊常用工具列中的編譯(Make)按鈕圖示或是功能區 Build 選單中的 Make 功能。
9. 在程序(Program)中可以透過拖曳 LD\_Example 功能塊來實例化此功能塊，如圖所示。

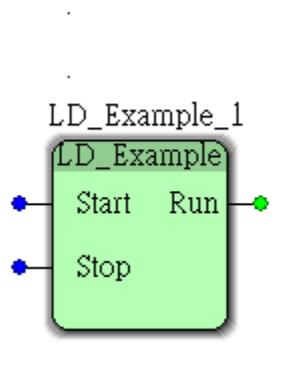


圖 4.2.10 LD 範例功能塊

### 4.3 功能塊圖( FBD )編程語言

功能塊圖(Function Block Diagram，簡稱 FBD)是一種圖形化的編程語言，實際上是用代表邏輯或算數符號的功能塊所組成，各方塊圖的輸入或輸出是由方塊圖之間的連接線來連接，類似繪製電路圖的方式來進行編程。

範例：方波產生器，負載週期(Duty Cycle)50%，週期可根據輸入參數調整。

1. 在專案樹 POU 節點中插入功能塊(Function Block)，名稱為 FBD\_Example，Language 選擇為 FBD。

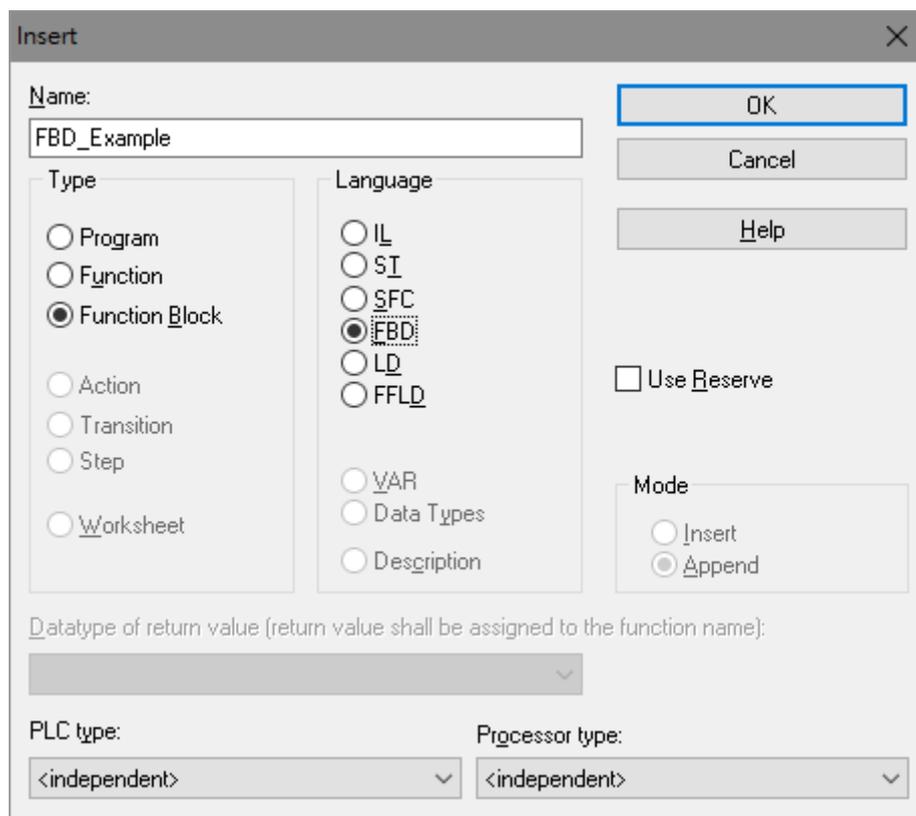


圖 4.3.1 新增 FBD 範例

# 4

2. 按下 **OK** 後，在其中工作單中，從右側的 **Edit Wizard** 視窗中選擇 **TON** 功能塊，並把 **TON** 功能塊拖曳至其工作單，在彈跳的視窗中將其命名為 **TON\_Front**，點擊 **OK**。使用同樣的方法實例化第二個 **TON** 功能塊，取名為 **TON\_Back**。

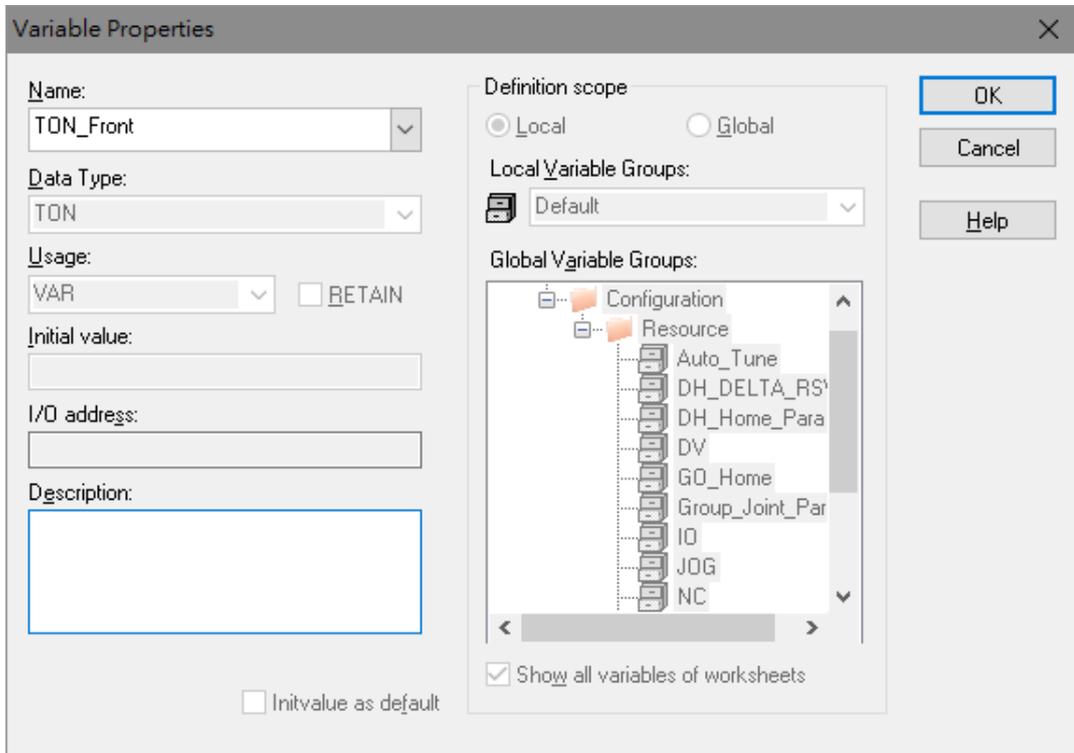


圖 4.3.2 FBD 接點設定

3. 將滑鼠移至 **TON\_Back** 的 **Q** 輸出端，如圖所示。此時按住滑鼠左鍵不放，將滑鼠移動到 **TON\_Front** 的 **IN** 輸入端，等待連接線變為綠色後，鬆開滑鼠左鍵。

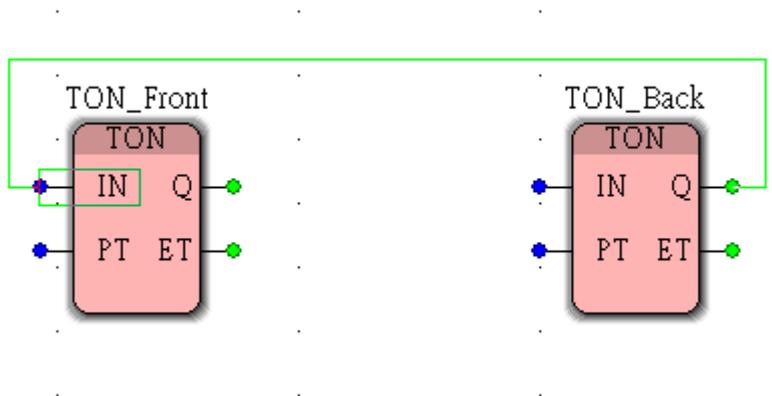


圖 4.3.3 FBD 範例

4. 同步驟 3，連接 **TON\_Front** 的 **Q** 和 **TON\_Back** 的 **IN**。

5. 雙擊 TON\_Front 功能塊，在彈出視窗中的 Formal Parameters 中 IN 選項勾選 Negated，如圖所示。點擊 **OK**，此時工作單中的 TON\_Front 的 IN 輸入端會增加了一個小圓圈，如圖所示。

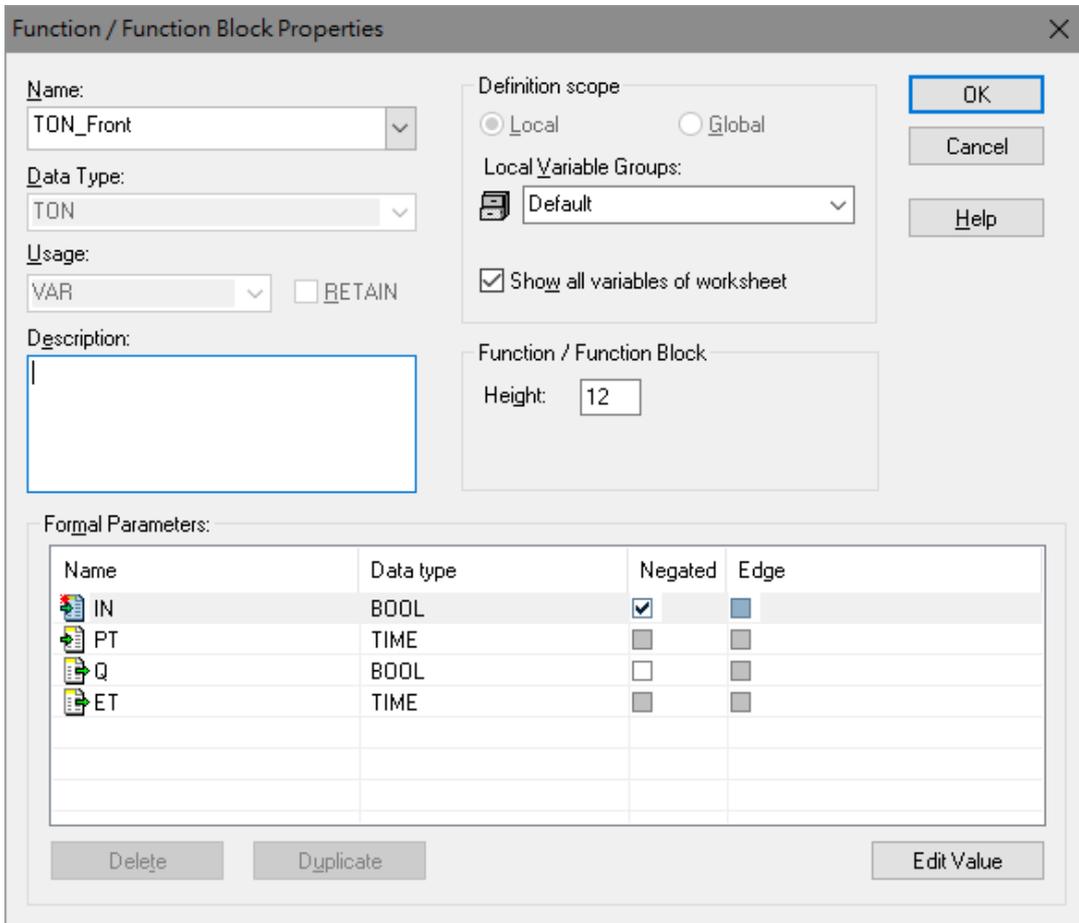


圖 4.3.4 FBD 接點設定

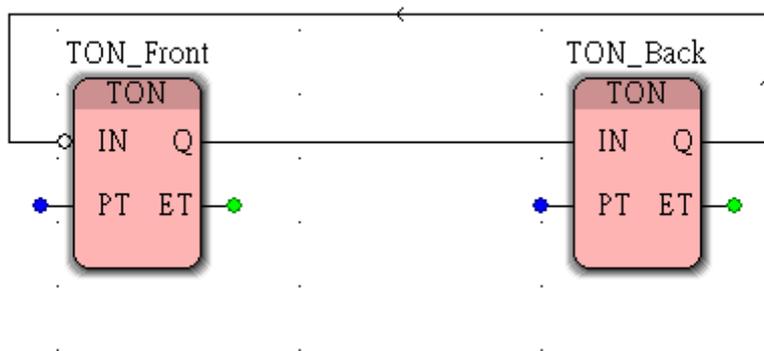


圖 4.3.5 FBD 範例

# 4

- 在 TON\_Front 的輸入端 PT 上點擊滑鼠右鍵，選擇選單中的 Variable，如圖所示。在彈出的視窗中輸入名稱 Cycle\_Time，Usage 選擇 VAR\_INPUT，並點擊 OK。

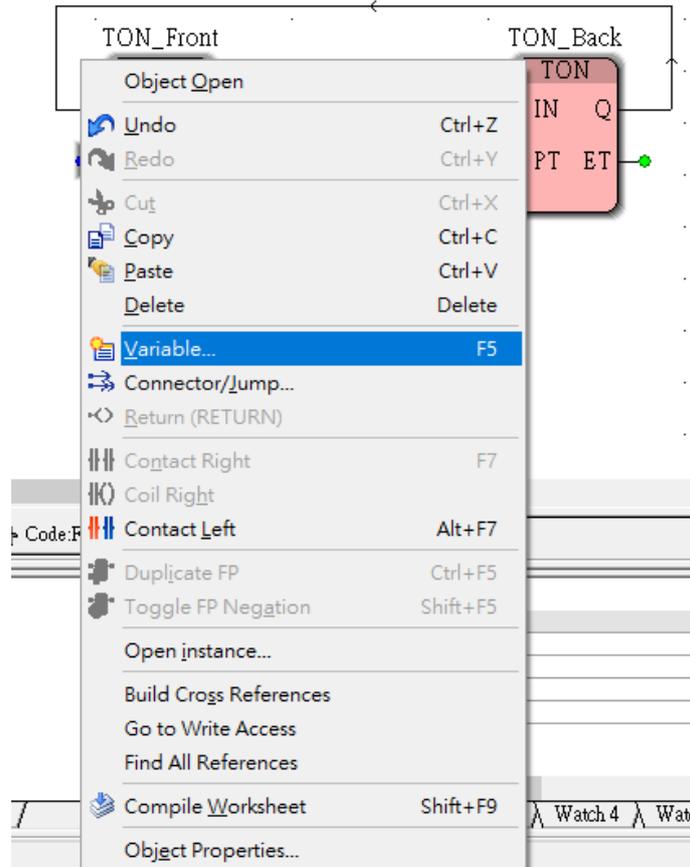


圖 4.3.6 變數設定路徑

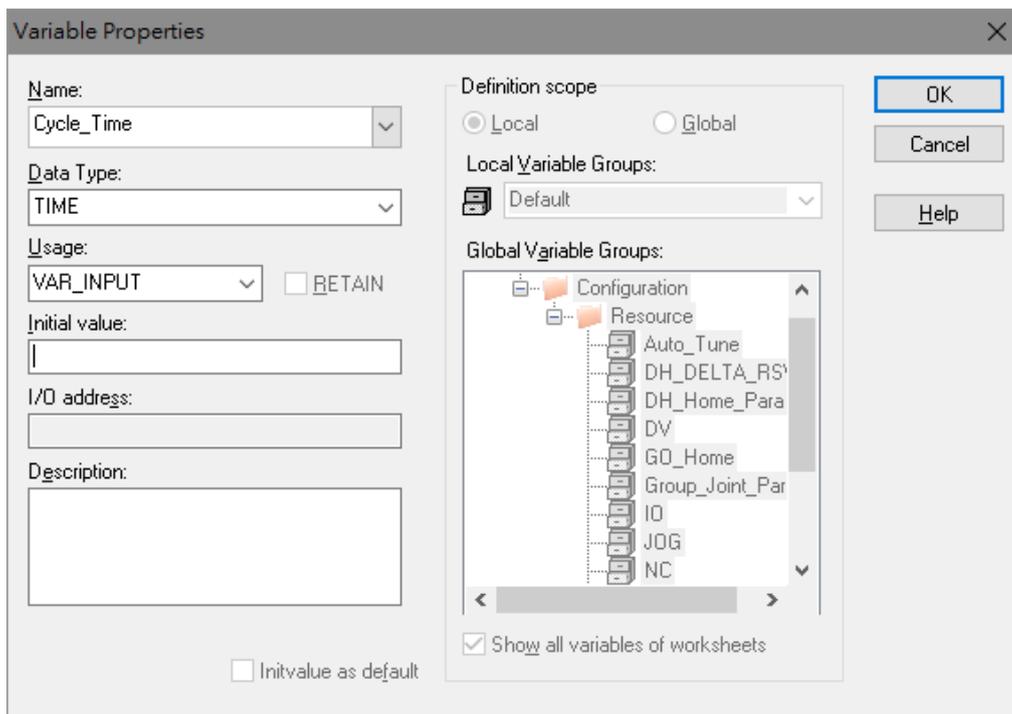


圖 4.3.7 變數設定

7. 連接 TON\_Back 的 PT 輸入端和變數 Cycle\_Time，如圖 4.3.8 所示。

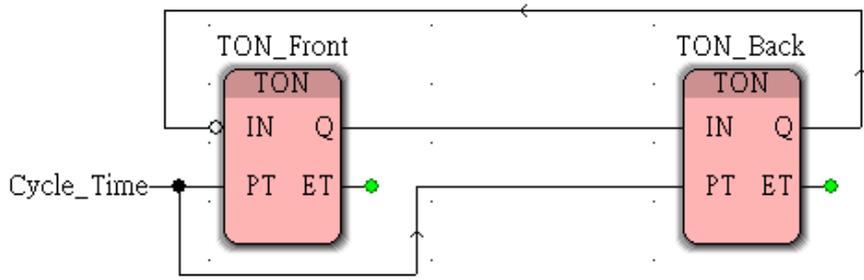


圖 4.3.8 FBD 範例

8. 在空白處點選滑鼠右鍵，在選單中選擇 Variable，然後插入一個變數 WaveOut，Data Type 選擇為 BOOL，Usage 為 VAR\_OUTPUT。
9. 連接 WaveOut 到 TON\_Front 的輸出端 Q 和 TON\_Back 的 IN 的連接線上，如圖所示。

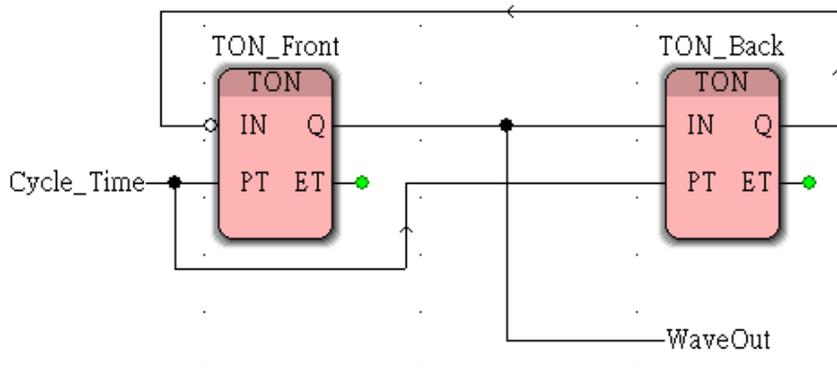


圖 4.3.9 FBD 範例

10. 點擊常用工具列中的編譯(Make)按鈕圖示，或是功能區 Build 選單中的 Make 功能。
11. 在程序(Program)中，可以透過拖曳 FBD\_Example 功能塊來實例化此功能塊，如圖所示。

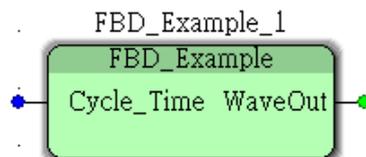


圖 4.3.10 FBD 範例功能塊

# 4

## 4.4 結構化文字( ST )編程語言

結構化文字(Structured Text·簡稱 ST)是一種支援塊狀結構(Block Structured)的高階語言·以 Pascal 為基礎·語法也類似 Pascal。具有和其他高階編程語言相同的程序流程控制能力·編寫的代碼結構清晰·適合用於解決複雜的控制問題·結構式文件編程語言支援複雜的敘述及巢狀指令。

範例：求自然數的累加和階乘。

1. 在專案樹 POU 節點中插入功能塊(Function Block)·名稱為 ST\_Example·Language 選擇為 ST。

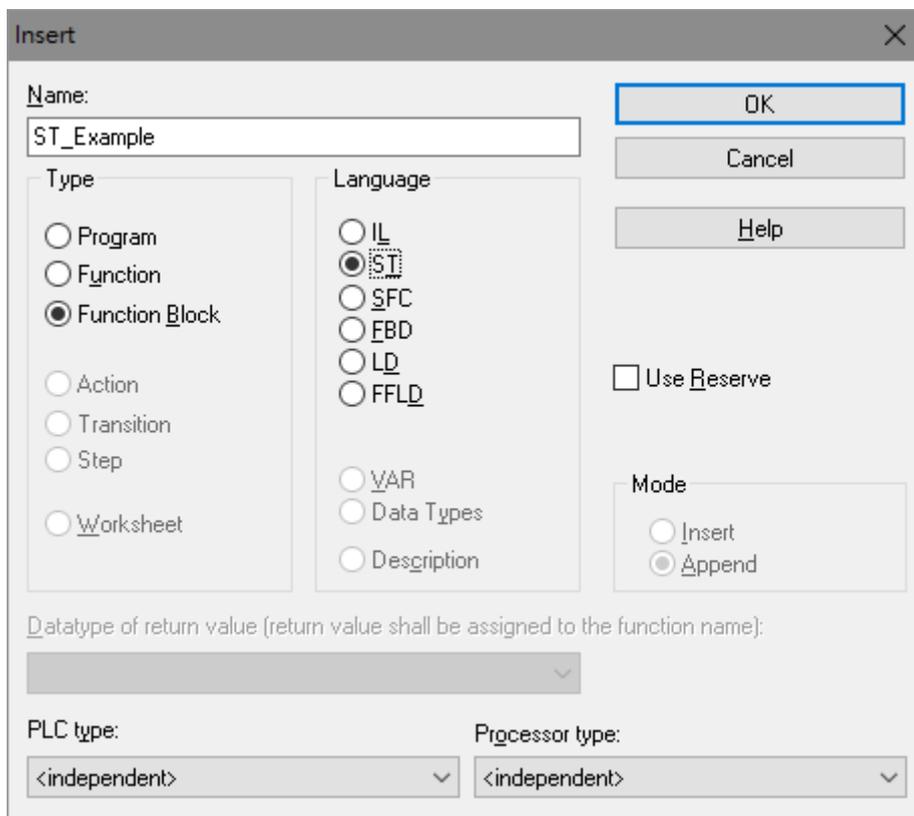


圖 4.4.1 新增 ST 範例

2. 按下 **OK** 後，在其中工作單中，點擊滑鼠右鍵，在選單中選擇 **Open Variables Worksheets**，並輸入如圖所示的變數。

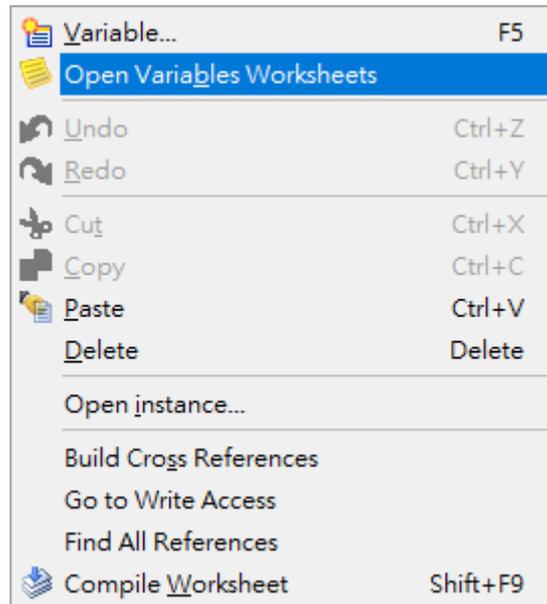


圖 4.4.2 變數設定路徑

	Name	Type	Usage	Description
1	Default			
2	Sum	INT	VAR_OUTPUT	
3	Factorial	INT	VAR_OUTPUT	
4	Begin	INT	VAR_INPUT	
5	End	INT	VAR_INPUT	

圖 4.4.3 變數設定

3. 在工作單輸入如圖所示之程式內容。

```

1 Sum := 0;
2 Factorial := 1;
3 FOR I:= Begin TO End BY 1 DO
4   Sum := Sum + I;
5   Factorial:= Factorial * I;
6 END_FOR;
7
8

```

圖 4.4.4 ST 範例

4. 點擊常用工具列中的編譯(Make)按鈕圖示或是功能區 **Build** 選單中的 **Make** 功能。

# 4

5. 在程序(Program)中可以透過拖曳 ST\_Example 功能塊來實例化此功能塊，如下圖所示。

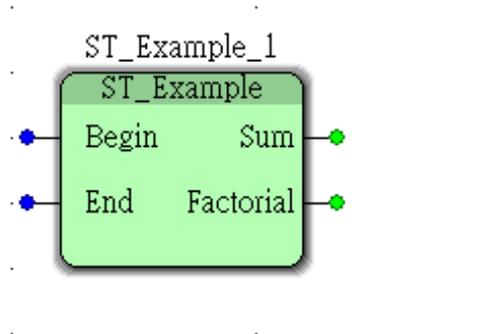


圖 4.4.5 ST 範例功能塊

## 4.5 順序功能圖( SFC )編程語言

順序功能圖(Sequence Function Chart，簡稱 SFC)，也叫狀態轉移圖，是一種描述順序控制系統控制條件和過程的方法。在順序功能圖中，將系統的工作過程分為多個階段，每個階段稱為一個「步」(Step)，每一步中可以完成一個或者多個特定的動作，同時也採用了文字敘述和圖形符號相結合的方式。SFC 程式代碼佔用的存儲空間較多，運行期間所需的資源也比較多，不適合解決簡單的問題。

範例：產線流程控制程序

產線上有一個控制開關，當開關啟動後，進入自動切換狀態。自動狀態下，首先工序 1，Y1 動作，接著自動切換，工序 2，Y2 動作，接著自動切換，工序 3，延時計數，在三個工序結束後，用重複的方式，直接返回工序 1，形成序列的循環。

1. 在專案樹 POU 節點中插入功能塊(Function Block) · 名稱為 SFC\_Example · Language 選擇為 SFC 。

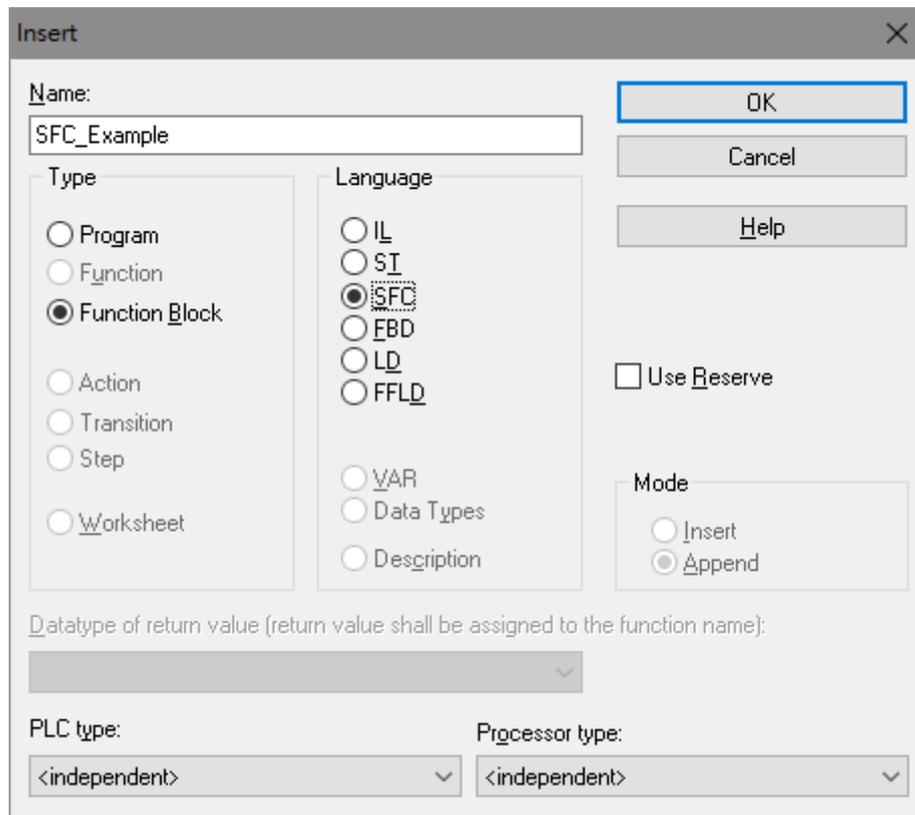


圖 4.5.1 新增 SFC 範例

2. 按下 **OK** 後，在其中工作單中，點擊滑鼠右鍵，在選單中選擇 **Open Variables Worksheets**，並輸入如圖所示的變數。

	Name	Type	Usage	Description
1	Default			
2	Start	BOOL	VAR_INPUT	
3	Process_1	BOOL	VAR_OUTPUT	
4	Process_2	BOOL	VAR_OUTPUT	
5	Y1_forward	BOOL	VAR_OUTPUT	
6	Y2_Back	BOOL	VAR_OUTPUT	
7	Delay_03	BOOL	VAR_OUTPUT	

圖 4.5.2 變數設定

# 4

3. 返回工作單，點擊滑鼠右鍵，在選單中選擇 SFC Network，如圖所示。

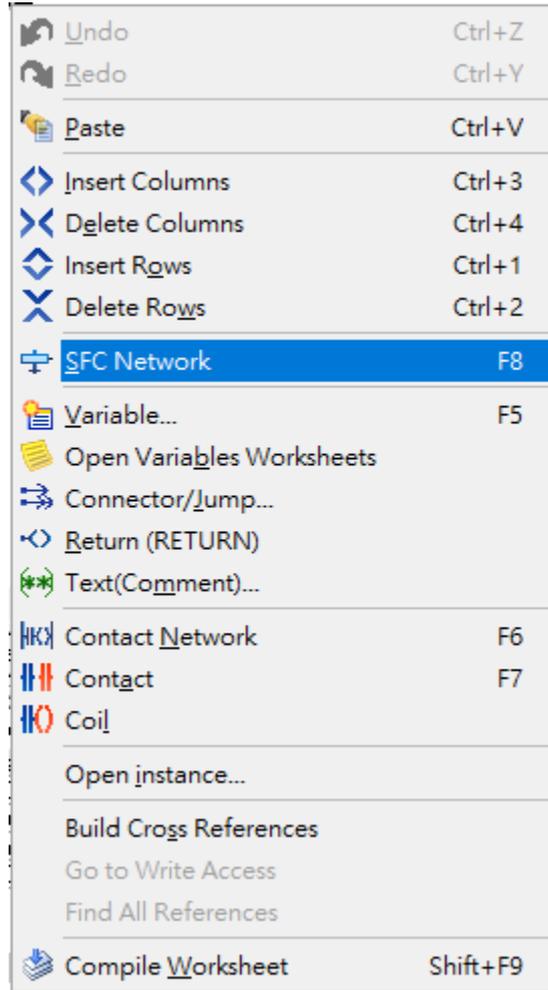


圖 4.5.3 SFC 新增元件路徑

4. 建立 SFC Network 之後，滑鼠左鍵點擊 S001 節點，然後點擊滑鼠右鍵，在選單中可以看到三個與 SFC 相關的指令。

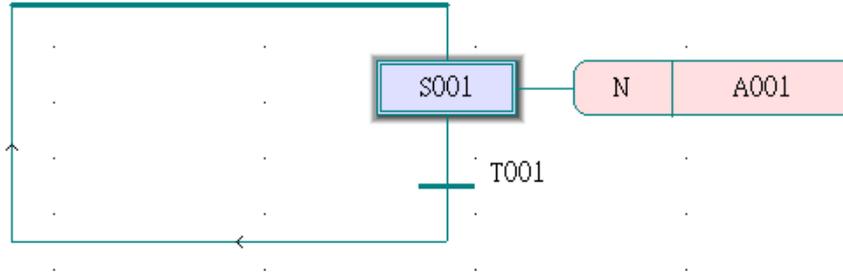


圖 4.5.4 SFC 範例

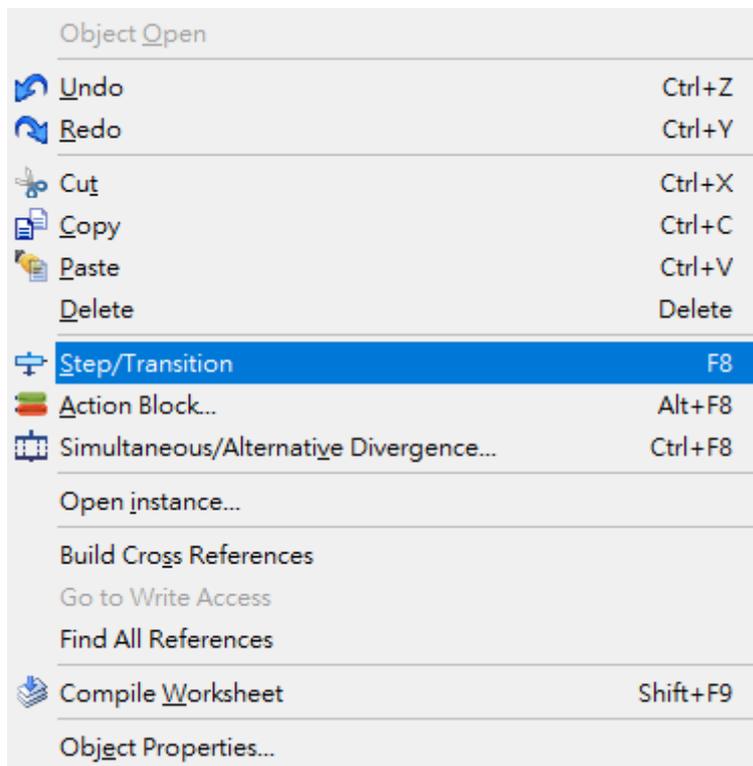


圖 4.5.5 SFC 新增元件路徑

**Step/Transition**：在選擇 Step 的情況下，可以增加一個工作 Step。

**Action Block**：在選擇 Action Block 或 Step 的情況下，可以增加一個動作。

**Simultaneous/Alternative Divergence**：在選轉換(T001)的情況下，則會在該轉換下增加一個 2 條支路的並行結構。如果在選擇 Step 的情況下點擊，則會在該步下增加 2 條支路的選擇結構。

# 4

5. 在工作單輸入如圖所示之程式內容。

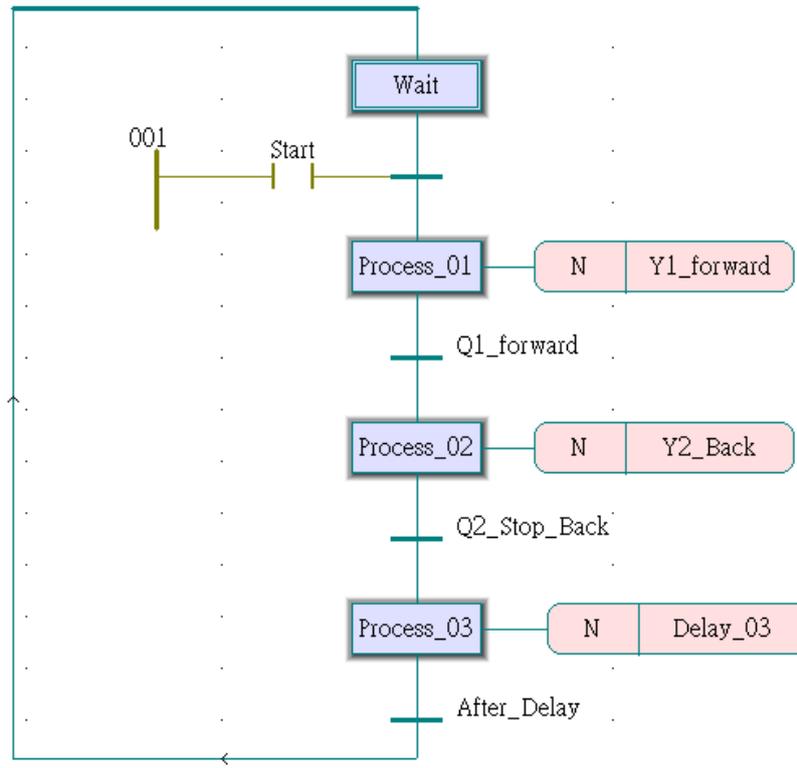


圖 4.5.6 SFC 範例

6. 雙擊 Q1\_forward 轉換，開啟工作單，在其中編寫 FBD 程序，如圖所示。

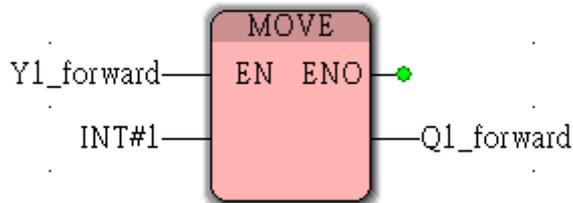


圖 4.5.7 SFC 範例功能塊

註：轉換程序可以採用 LD、ST、IL、FBD 四種編程語言中的任何一種實現。

7. 點擊常用工具列中的編譯(Make)按鈕圖示或是功能區 Build 選單中的 Make 功能。

8. 在程序(Program)中可以透過拖曳 ST\_Example 功能塊來實例化此功能塊，如圖所示。

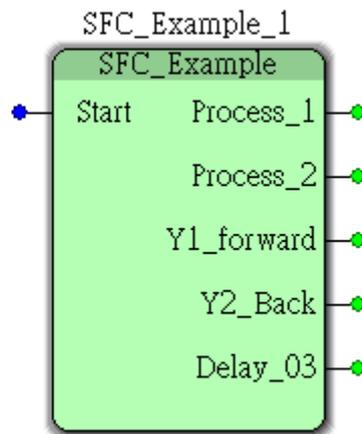


圖 4.5.8 SFC 範例功能塊

(此頁有意留為空白)

# 4

# PLCopen 簡介及功能塊介紹

# 5

簡介 PLCopen 組織，並提供常用的功能塊說明。

5.1	PLCopen 介紹	5-3
5.1.1	PLCopen 組織簡介	5-3
5.1.2	PLCopen Motion Control 簡介	5-4
5.1.3	常用變數定義	5-5
5.1.4	警報代碼	5-9
5.2	功能塊介紹	5-12
5.2.1	功能塊總覽	5-12
5.2.2	單軸功能塊	5-14
5.2.2.1	MC_Home	5-14
5.2.2.2	MC_Stop	5-18
5.2.2.3	MC_Jog	5-22
5.2.2.4	MC_MoveAbsolute	5-25
5.2.2.5	MC_MoveRelative	5-30
5.2.2.6	MC_MoveAdditive	5-33
5.2.2.7	MC_MoveVelocity	5-37
5.2.2.8	MC_Power	5-40
5.2.2.9	MC_Reset	5-42
5.2.2.10	MC_SetOverride	5-43
5.2.3	群組功能塊	5-46
5.2.3.1	MC_GroupReset	5-46
5.2.3.2	MC_GroupSetOverride	5-48
5.2.3.3	MC_GroupStop	5-51
5.2.3.4	MC_MoveLinearAbsolute	5-55
5.2.3.5	MC_MoveLinearRelative	5-58
5.2.3.6	MC_MoveDirectAbsolute	5-61
5.2.3.7	MC_MoveDirectRelative	5-64
5.2.4	擴充功能塊	5-67
5.2.4.1	MC_MoveDirectAbsolute_Ex	5-67
5.2.4.2	MC_MoveDirectRelative_Ex	5-70
5.2.4.3	MC_Home_Ex	5-73
5.2.4.4	MC_MoveLinearJog	5-75
5.2.5	系統功能塊	5-78

## 5

5.2.5.1	SYS_AlarmSet	5-78
5.2.5.2	SYS_AlarmShow	5-80
5.2.5.3	SYS_ReadPara	5-82
5.2.5.4	SYS_WritePara	5-84
5.2.5.5	SYS_RemoteReadPara	5-86
5.2.5.6	SYS_RemoteWritePara	5-88
5.2.5.7	APP_ReadPara	5-91
5.2.5.8	APP_WritePara	5-93
5.2.5.9	DRL_CommandSet	5-95
5.2.5.10	DRL_GetStatus	5-97
5.2.5.11	DRL_GetMainCurrentLine	5-98
5.2.5.12	DRL_GetCommandReply	5-99
5.2.5.13	DRL_GetCurrentProgNo	5-100
5.2.5.14	MB_ReadHoldRegs	5-101
5.2.5.15	MB_WriteRegs	5-105
5.2.5.16	MB_WriteSingleRegs	5-108
5.2.5.17	MB_TCPReadHoldRegs	5-110
5.2.5.18	MB_TCPWriteRegs	5-114
5.2.5.19	MB_TCPWriteSingleRegs	5-117
5.2.5.20	CAN_SDORead	5-119
5.2.5.21	CAN_SDOWrite	5-122

## 5.1 PLCopen 介紹

### 5.1.1 PLCopen 組織簡介

PLCopen 組織是獨立於製造商和產品的國際組織，此組織的宗旨是促進 PLC 兼容軟件的開發和使用，提倡使用 PLC 國際標準(IEC 61131-3)作為使用規範，其目的以「解決與控制編程相關的問題」和「支持該領域內國際標準的使用」為使命。用戶可通過在眾多程序開發環境中應用此標準，於不同品牌產品和不同類型的控制器之間移植控制程序，實現互換功能，因此 PLCopen 為此設下了技術與推廣組織，如下圖。

PLCopen 組織的一項主要工作是使用 IEC-61131-3 工業控制編程(Programming)國際標準，將各 Function Block 接腳統一化，使設計者可依照此標準設計，也讓使用方式與邏輯更加一致。PLCopen 組織利用 IEC-61131-3 標準共制定了五項的 PLC 程式語言；Ladder Diagram(LD)階梯圖、Sequential Function Chart (SFC) 順序功能流程圖、Function Block Diagram (FBD) 功能區塊圖(具有結構性，屬物件導向)、Structured Text (ST)似 Pascal 語言、Instruction List (IL)簡單文本 PLC 語言。而每一個程式則由邏輯元件、模組化以及軟體技術組成，從而提高使用率、減少錯誤並有效的編輯程式。

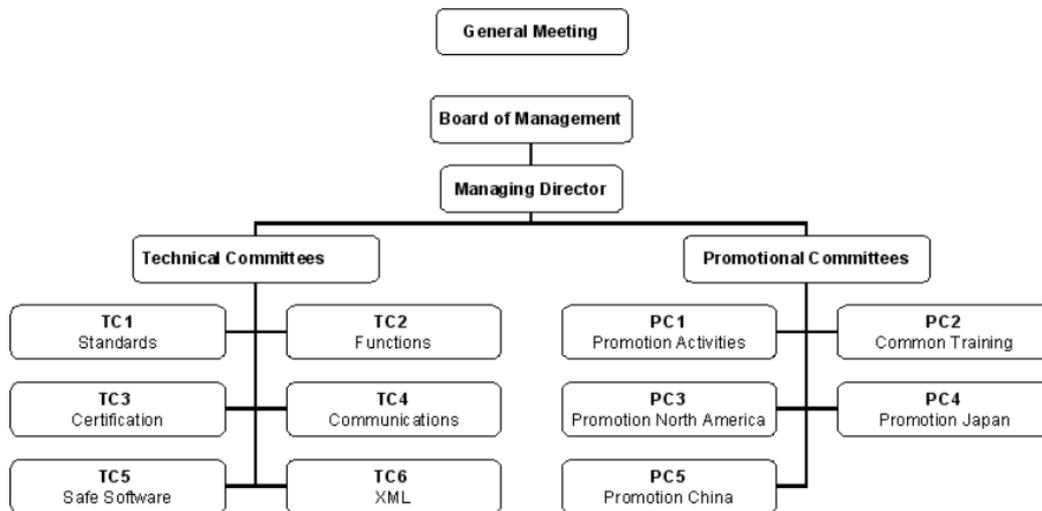


圖 5.1.1.1 PLCopen 組織架構 (出處: <http://www.plcopen.org/index.html>)

## 5

## 5.1.2 PLCopen Motion Control 簡介

PLCopen 運動控制主要由 PLCopen 組織底下的 TC2 負責規劃，TC2 基於應用需求和工程專案規格，設計工程師需要使用或選擇範圍廣泛的運動控制器來完成。過去這些應用開發需要使用個別應用軟體來創建，即使功能相同，也會因為軟體的不同而無法共用。

PLCopen 運動標準提供了一種標準的應用程序庫，可重複使用在多種硬體平台上，這樣的方式可大幅降低開發和維護成本。除此之外，編輯程式變得更加容易簡單，培訓成本降低、並且統一設計的方式，使得功能可做跨平台使用。實際上，其標準化是通過定義可重複使用的 **Block** 來完成的。透過此方式，使用者無須依賴硬體的編程，應用軟件亦可重複增加，以減少人員培訓和功能維護的成本，使各種控制解決方案變得更有彈性。由於數據隱藏和封裝可使用不同的體系，例如集中或分散式的控制，不僅能運用在單一的應用功能上，更能作為一個基本層，廣泛的用於不同領域中。

PLCopen 為可開放式，可應用在現有或未來技術上，故 PLCopen 所定義的運動控制功能主要是以 **Function Block** 為主，PLCopen 運動控制功能所開發的標準文件如下：

- Part 1 - Function Blocks for Motion Control
- Part 2 - Extensions(in the new release 2.0 merged with Part 1)
- Part 3 - User Guidelines
- Part 4 - Coordinated Motion
- Part 5 - Homing Procedures
- Part 6 - Fluid Power Extensions

經 PLCopen 認證以後符合標準的標示如下，由圖示便可得知其組織定義的 **Motion** 功能是以 **Function Block** 的概念為設計基礎。



圖 5.1.2.1 PLCopen motion control 認證標記圖

### 5.1.3 常用變數定義

下表為功能塊常見的輸入及輸出變數列表，依照不同功能塊的特性而定，通常會包含至少一個或是一組變數，其變數如下：

輸入變數			
名稱	資料型態	設定值 (預設值)	功能
Enable	BOOL	True / False (False)	啟用功能塊
Execute	BOOL	True / False (False)	啟動功能塊
輸出變數			
名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	功能塊動作完成
Valid	BOOL	True / False (False)	輸出值有效
Busy	BOOL	True / False (False)	功能塊動作已被觸發
Active	BOOL	True / False (False)	功能塊動作執行中
CommandAborted	BOOL	True / False (False)	功能塊被其他命令中斷
Error	BOOL	True / False (False)	功能塊產生錯誤

一個運動功能塊中必定會包含 **Enable** 或是 **Execute** 其中一個輸入變數，用於執行此功能塊。而顯示運動功能塊執行狀態時，則會包含 **Busy** 和 **Done** 兩種輸出變數。**Active** 變數則是有 **Buffer** 輸入變數的功能塊才提供。對於同一個運動軸，可同時有多個 **Busy** 輸出，但只會有一個功能塊 **Active** 輸出。如果功能塊在執行過程可被其他功能塊中斷，則會包含 **CommandAborted** 變數。**Error** 變數則用來顯示此運動功能塊在啟動過程中發生錯誤。單一運動功能塊的輸入變數除了 **Enable** 和 **Execute** 以外，會包含其他運動資料輸入端，這些資料 / 狀態會有以下特性：

■ 輸入資料取用時機

1. 具有 **Enable** 輸入時：參數會在 **Enable** 上緣觸發時被取用。與 **Execute** 相比，**Enable** 較常被設計為執行中持續更新的形式，例如讀取參數功能塊。
2. 具有 **Execute** 輸入時：參數會在 **Execute** 上緣觸發時被取用。若要讓變更資料再次生效，必須在修改輸入參數後，再次讓 **Execute** 上緣觸發。

■ 輸入資料超出範圍

若於運動功能塊輸入超出允許範圍的數值且本功能塊被啟動時，會造成輸入的資料被限制或是產生錯誤。此時產生的軸錯誤結果，是運動功能塊的應用錯誤所造成的，因此使用者必須確保輸入的資料正確。

## 5

- 輸入時缺少參數  
根據 IEC-61131-3 定義，假使功能塊輸入參數缺少時，則輸入參數維持上次輸入值，第一次使用則是使用系統預設值。
- 輸出狀態互斥
  1. 功能塊有 **Execute** 輸入變數時，**Busy**、**Done**、**Error** 和 **CommandAborted** 會彼此互斥。運動功能塊中同時只有一個能為 **True**，且其中一個必定為 **True**。
  2. 功能塊有 **Buffer** 輸入變數時，**Active**、**Error**、**Done** 和 **CommandAborted** 同時只能有一個被設置。
  3. 功能塊有 **Enable** 輸入變數時，輸出 **Valid** 與 **Error** 彼此互斥，同時只能有一個被設置。
- 輸出資料 / 狀態有效時機
  1. 功能塊有 **Execute** 變數時，**Done**、**Error**、**ErrorID**、**CommandAborted** 以及資料輸出會在 **Execute** 下緣時被重置，然而 **Execute** 下緣不會停止，也不影響功能塊實際的執行，即使在功能塊完成前 **Execute** 就被重置，相對應的輸出狀態仍會產生，並保持一個週期。如果功能塊在完成之前就收到新的 **Execute**，功能塊不會對之前動作的 **Done** 與 **CommandAborted** 有任何反饋，且可能產生功能塊錯誤。
  2. 功能塊有 **Enable** 變數時，**Valid**、**Enable**、**Busy**、**Error** 和 **ErrorID** 輸出將跟著 **Enable** 下緣被重置。
- 各輸出端特性
  1. **Done** 輸出特性：**Done** 輸出會在被命令的運動成功完成時被設置。
  2. **Busy** 輸出特性：
    - 2.1 功能塊有 **Execute** 變數時，每個運動功能塊會有一個 **Busy** 輸出，用來反映運動功能塊尚未完成，並且輸出狀態值將會被更新。**Busy** 在 **Execute** 上緣被設置時，在 **Done**、**CommandAborted** 和 **Error** 被設置時將被重置。
    - 2.2 功能塊有 **Enable** 變數時，每個運動功能塊會有一個 **Busy** 輸出，用來反應運動功能塊尚未完成，並且輸出狀態值將會被更新，**Busy** 在 **Enable** 上緣被設置時，只要運動功能還在執行，動作就會保持住，同時，對應的輸出仍會有變化。
  3. **Active** 輸出特性：
    - 3.1 有 **Buffer** 緩衝輸入的功能塊需要 **Active** 輸出變數，當功能塊實際控制對應的運動軸時，此輸出會被設置，沒有 **Buffer** 緩衝輸入的功能塊，其 **Active** 與 **Busy** 輸出行為相同。
    - 3.2 對於同一個運動軸，可以有多個功能塊的 **Busy** 變數同時輸出，但在一個時刻只能有一個功能塊的 **Active** 變數輸出，表示正在執行此功能塊。

CommandAborted 輸出特性：CommandAborted 被啟動時，會即時停止所有尚未完成的運動。

4. Enable 與 Valid 的關係：搭配 Enable 時，若運動功能塊包含輸出狀態或是數值的變數，則會由 Valid 輸出變數來表示這些輸出是否有效，Valid 輸出只在 Enable 為真及輸出有效時為真，若運動功能塊有錯誤會讓輸出無效，Valid 會為 False，直到錯誤狀況消失，輸出值重新生效時，Valid 才會再次被設置。
5. 正負號規則：Acceleration、Deceleration 和 Jerk 永遠為正。Velocity、Position 或 Distance 則可能包含正負號。

■ 輸入/輸出端的變化時序

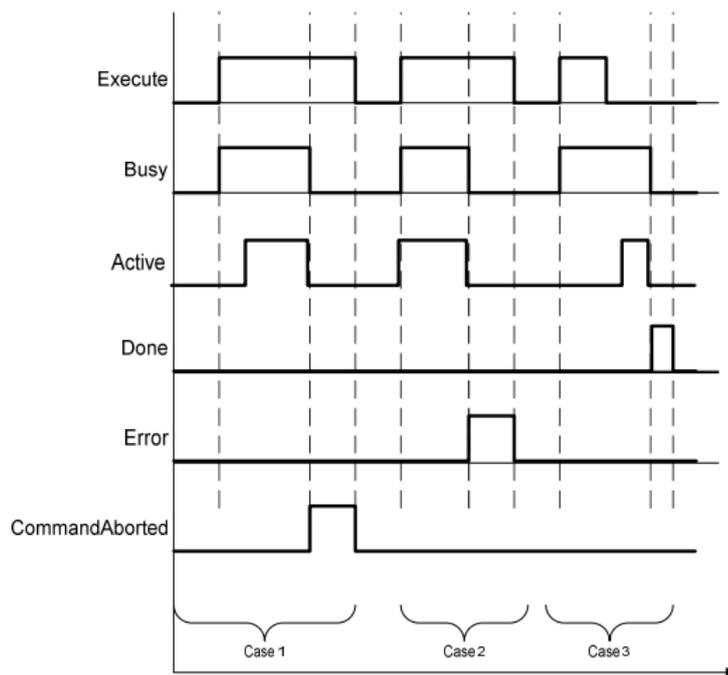


圖 5.1.3.2 Execute 功能塊輸入輸出端的變化時序範例 (出處: MULTIPROG 快速入門指南)

5

Case 1 : 含 Buffer 輸入的運動功能塊執行時，被其他功能塊中斷。

Case 2 : 運動功能塊執行時發生異常。

Case 3 : 含 Buffer 輸入的運動功能塊完成正常動作。(即使 Execute 已為 False，功能塊仍要將動作完成，完成時 Done 訊號輸出至少要維持一個 PLC 掃描週期。

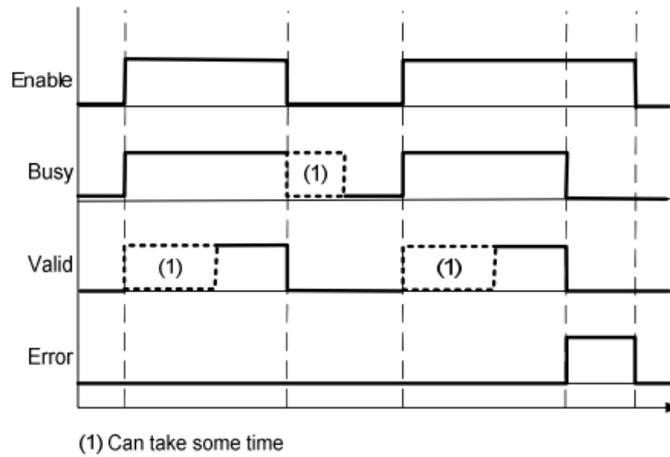


圖 5.1.3.1 Enable 功能塊輸入輸出端的變化時序範例 (出處: MULTIPROG 快速入門指南)

註：(1)表示需要一段時間。

■ 插斷(Aborting)及緩存(Buffer)模式

若運動功能塊包含 BufferMode 輸入接腳，可選擇 Non-buffered mode(預設模式，即插斷)或 Buffered mode 兩種方式來觸發運動功能塊。

1. 當 Non-buffered mode 運動功能塊執行時，立即插斷前一個正在執行的運動功能塊，並清除緩存命令。
2. 當 Buffer mode 運動功能塊執行時，會等待前一個正在執行的運動功能塊完成，例如「Done」(或「InPosition」、「InVelocity」...)，再接著執行此命令。

其中 BufferMode 有以下幾種選擇：

- Aborting：預設模式，立即插斷前一個正在執行的運動功能塊，並清除緩存命令。
- Buffered：下一個運動功能塊，接著上一個運動功能塊「Done」輸出而執行，且速度沒有疊合(Blending)。

### 5.1.4 警報代碼

類別	錯誤代碼	說明
一般錯誤 (General)	0x1000	Axis 號碼超出範圍
	0x1001	Node 號碼超出範圍
	0x1002	Group 號碼超出範圍
	0x1003	Index 號碼超出範圍
	0x1004	Command Index 號碼超出範圍
	0x1005	Type 號碼超出範圍
	0x1006	CardN 號碼超出範圍
	0x1007	SlotID 號碼超出範圍
	0x1008	Port 號碼超出範圍
	0x1009	用來儲存回傳或寫入資料所配置的記憶體不足
	0x10F0	逾時
	0x10F1	命令失效
軸錯誤 (Axis)	0x1200	Axis 編號錯誤
	0x1201	Axis 伺服未備妥
	0x1202	Axis 伺服異常
	0x1203	Axis 停止
	0x1204	Axis Master 類型錯誤
	0x1205	Axis Master 編號錯誤
	0x1206	Axis Probe 編號錯誤
	0x1220	Axis 命令緩衝區錯誤
	0x1221	Axis 運動方塊編號錯誤
	0x1222	Axis 運動命令解譯錯誤
	0x1223	Axis 運動命令速度錯誤
	0x1224	Axis 運動命令加速度錯誤
	0x1225	Axis 運動命令減速度錯誤
	0x1226	Axis 運動命令加加速度錯誤
	0x1227	Axis 運動命令 Buffer Mode 錯誤
	0x1228	Axis 運動命令 — Master Value Source 錯誤
	0x1229	Axis 運動命令 — Gear Ration 錯誤
	0x122A	Axis 運動命令 — Velocity Factor 錯誤
	0x122B	Axis 運動命令 — Direction 錯誤
	0x122C	Axis 運動命令 — Scale 錯誤
	0x122D	Axis 運動命令 — Table Index 錯誤
	0x122E	Axis 運動命令 — Table 錯誤
	0x122F	Axis 運動命令 — Start Mode 錯誤
0x1230	Axis 原點復歸錯誤	
0x1231	Axis 原點復歸動作中	

## 5

類別	錯誤代碼	說明
	0x1240	Axis Admin — 命令緩衝區錯誤
	0x1241	Axis Admin — 方塊編號錯誤
	0x1242	Axis Admin — 命令解釋錯誤
	0x1243	Axis Admin — 輸入參數錯誤
群組錯誤 (Group)	0x1300	Group 編號錯誤
	0x1301	Group 群組未生效
	0x1302	Group 群組未備妥
	0x1303	Group 群組異常
	0x1320	Group 命令緩衝區錯誤
	0x1321	Group 運動方塊編號錯誤
	0x1322	Group 運動命令—解釋錯誤
	0x1323	Group 運動命令—速度錯誤
	0x1324	Group 運動命令—加速度錯誤
	0x1325	Group 運動命令—檢速度錯誤
	0x1326	Group 運動命令—加加速度錯誤
	0x1327	Group 運動命令— Buffer Mode 錯誤
	0x1328	Group 運動命令— Trans Mode 錯誤
	0x1329	Group 運動命令— Coordinate System 錯誤
	0x132A	Group 運動命令— Velocity Factor 錯誤
	0x1340	Group Admin — 命令緩衝區錯誤
	0x1341	Group Admin — 方塊編號錯誤
	0x1342	Group Admin — 命令解釋錯誤
	0x1343	Group Admin — 輸入參數錯誤
總線錯誤(Fieldbus)	0x1800	總線未備妥
	0x1801	總線斷線
	0x1802	總線該 Node 站號未能啟動
	0x1803	總線該 Node 站號連線以後斷線
	0x1804	總線該 Node 站號不允許針對此類型的操作
通訊錯誤 (Communication)	0x1900	通訊忙碌無法處理此筆通訊命令要求
	0x1901	通訊 open()設定後超時
	0x1902	通訊 write()操作後超時
	0x1903	通訊 read()操作後超時
	0x1904	通訊處理完成後超時
	0x1905	RS232 / RS485 接口沒有備妥
	0x1906	RS232 / RS485open()失敗
	0x1907	ModbusTCP socket 無法建立
	0x1908	網路裝置無法綁定(PortNo 有誤)
	0x1909	網路連線被拒絕
	0x1980	PortNo 超出範圍

類別	錯誤代碼	說明
	0x1981	IP Address 錯誤
	0x1982	Slave 站號超出範圍
	0x1983	用來儲存 Modbus 回傳或寫入資料所配置的記憶體不足
	0x1984	讀取或寫入的暫存器數目超出範圍
	0x1985	Modbus Function Code 沒有支援
	0x19FF	底層回覆 Unknown Exception

## 5

## 5.2 功能塊介紹

## 5.2.1 功能塊總覽

分類		名稱	說明
單軸 運動控制	定位控制	MC_Home	啟動指定軸的原點復歸流程。
		MC_Stop	控制指定軸減速到停止。
		MC_Jog	啟動指令軸的連續吋動指令。
		MC_MoveAbsolute	控制指定軸依照指定的運動方式，移動到使用者設定的絕對目標位置。
		MC_MoveRelative	控制指定軸依照指定的運動方式，移動到使用者設定的相對目標位置。
		MC_MoveAdditive	控制指定軸依照指定的運動方式，再附加一段移動距離。
	速度控制	MC_Velocity	控制指定軸依照指定運動方式執行使用者指定的速度均速運動。
	管理型	MC_Power	啟動或關閉單軸伺服。
		MC_Reset	清除指定軸相關錯誤。
MC_SetOverride		透過超馳控制(override control)係數，改變目標軸的速度。	
多軸 運動控制	同步控制	MC_GearIn	依照輸入的齒輪比，使從軸速度命令跟隨乘以齒輪比後的主軸速度命令。
群組 運動控制	管理型	MC_GroupReset	清除群組軸及其成員軸的相關錯誤。
		MC_GroupSetOverride	透過超馳控制(override control)係數，改變目標群組軸的速度。
	定位控制	MC_GroupStop	控制指定群組軸減速到停止。
		MC_MoveDirectAbsolute	控制指定群組軸執行無規畫路徑運動，移動到使用者設定的絕對目標位置。
		MC_MoveDirectRelative	控制指定群組軸執行無規畫路徑運動，移動到使用者設定的相對目標位置。
		MC_MoveLinearAbsolute	控制指定群組軸執行直線插補運動，移動到使用者設定的絕對目標位置。
		MC_MoveLinearRelative	控制指定群組軸執行直線插補運動，移動到使用者設定的相對目標位置。

分類		名稱	說明
擴充 運動控制	定位控制	MC_Home_Ex	啟動指定軸按照設定模式及速度進行原點復歸。
		MC_MoveDirectAbsolute_Ex	控制群組軸按照設定的速度參數執行無規畫路徑運動至指定的絕對位置。
		MC_MoveDirectRelative_Ex	控制群組軸按照設定速度參數執行無規畫路徑運動至指定的相對位置。
		MC_MoveLinearJog	啟動群組軸按照設定方向及速度參數進行直線插補吋動。
系統控制	警報設定	SYS_AlarmSet	將警報錯誤碼儲存至歷史錯誤紀錄區域。
		SYS_AlarmShow	將警報錯誤碼顯示至七段顯示器。
	參數設定	SYS_ReaPara	MS本體軸參數讀取
		SYS_WritePara	MS本體軸參數寫入
		SYS_RemoteReadPara	MS擴充軸參數讀取
		SYS_RemoteWritePara	MS擴充軸參數寫入
		APP_ReadPara	產業應用參數讀取
		APP_WritePara	產業應用參數寫入
		CAN_SDORead	擴充模組參數讀取
		CAN_SDOWrite	擴充模組參數寫入
	RL狀態	DRL_CommandSet	RL的執行狀態設定
		DRL_GetStatus	讀取目前RL的執行狀態
		DRL_GetMainCurrentLine	讀取目前RL執行行數
		DRL_GetCommandReply	讀取RL的回覆狀態
		DRL_GetCurrentProgNo	讀取目前RL的專案編號
	記憶體 控制	MB_ReadHoldRegs	透過串列傳輸讀取外部Modbus資料，並傳回MS記憶體。
		MB_WriteRegs	透過串列傳輸將MS記憶體資料，寫入至外部Modbus裝置。
		MB_WriteSingleRegs	透過串列傳輸將一筆MS記憶體資料，寫入至外部Modbus裝置。
		MB_TCPReadHoldRegs	透過Ethernet傳輸讀取外部Modbus資料，並傳回MS記憶體。
		MB_TCPWriteRegs	透過 Ethernet 傳輸將MS記憶體資料，寫入至外部 Modbus 裝置。
MB_TCPWriteSingleRegs		透過Ethernet傳輸將一筆MS記憶體資料，寫入至外部Modbus裝置。	

# 5

## 5.2.2 單軸功能塊

### 5.2.2.1 MC\_Home

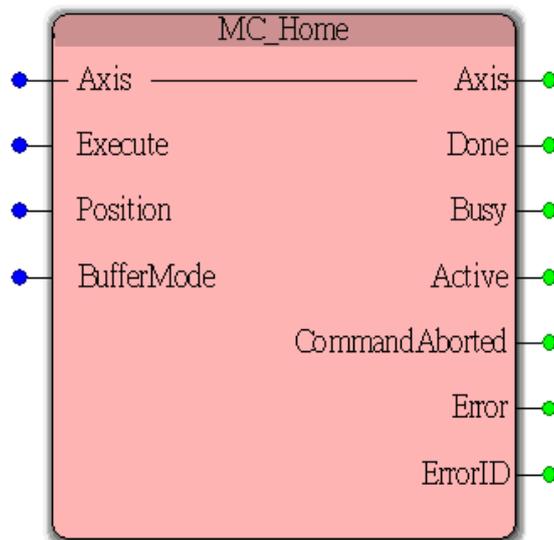
■ 類型

Function Block

■ 功能描述

啟動單軸的「尋找原點」流程，將依據使用者選擇的回原點模式執行。例如：尋找極限、感測器和扭力等模式來執行回原流程。

■ 功能塊



■ 輸入輸出共用變數

名稱	資料型態	設定值 (預設值)	功能
Axis	AXIS_REF	1 ~ 16 (-)	指定運動軸編號

■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	當Execute為上升緣時，此功能即開始執行
Position	LREAL	負數、正數或0 (0)	原點距離目前位置的偏差值
BufferMode	MC_BUFFER_MODE	0x00 ~ 0x0A (0)	Buffer Mode選擇，請參考第5.1.3章節

■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	當絕對定位完成時為True
Busy	BOOL	True / False (False)	當指令被觸發執行時為True
Active	BOOL	True / False (False)	當指定軸受控制時為True
Command Aborted	BOOL	True / False (False)	當指令被中斷時為True
Error	BOOL	True / False (False)	當指令錯誤發生時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	當指令錯誤發生時所記錄的錯誤碼；錯誤碼詳細說明請參考第5.1.4章節

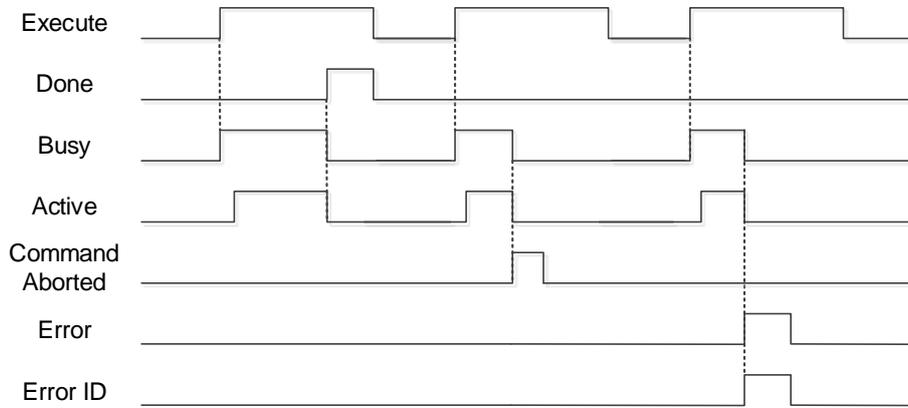
■ 輸出變數的變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Done	當絕對定位完成時	<ul style="list-style-type: none"> <li>■ 當Execute轉為False時</li> <li>■ 若Execute為False，而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Busy	當 Execute 上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
Active	當軸運動開始時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
Command Aborted	<ul style="list-style-type: none"> <li>■ 當此功能塊指令被其他緩衝模式設為mcAborting的指令中斷時</li> <li>■ 當此功能塊指令被 MC_Stop中斷時</li> </ul>	<ul style="list-style-type: none"> <li>■ 在Execute下緣時</li> <li>■ 當Execute為False，而CommandAborted轉為True，此時CommandAborted維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在ErrorID)	在Execute下緣時 (清除ErrorID紀錄之錯誤碼)

# 5

■ 輸出端的變化時序

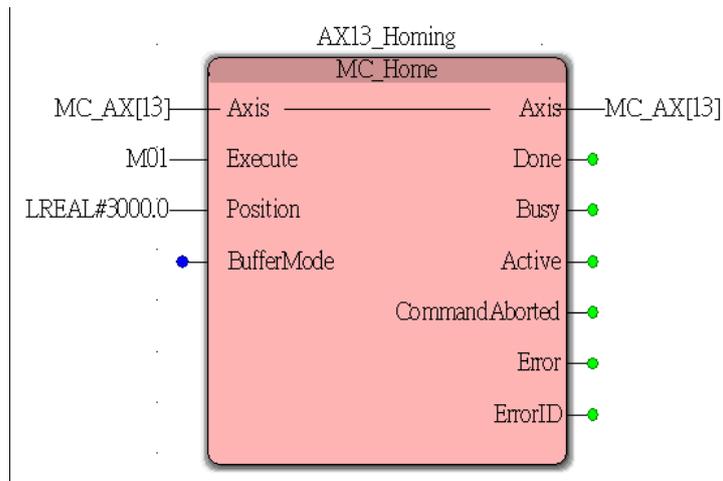
下方將介紹三種不同情況的回原狀態時序圖，其中將包含「完整執行回原」、「因中斷命令而停止回原」以及「因錯誤產生時停止回原的三種情況。



■ 回原模式

回原模式的流程將依據使用者所設定之回原模式進行。(模式種類請參考 ASDA-A2 手冊。)

■ 參考範例



此範例將依下列條件來設計：

- I. M01 設置觸發變數，可以為某個數位輸入訊號。
- II. 目前位置 10000 PUU。
- III. 功能塊輸入變數 Position 設計為 3000.0。

預計獲得動作與結果：

- I. 將目前位置設為 0 PUU(原點)。
- II. 將相對目前位置 3000.0 PUU 位置設為 0 PUU(原點)。
- III. 目前點位將變更為-3000.0 PUU。

■ 圖形表示

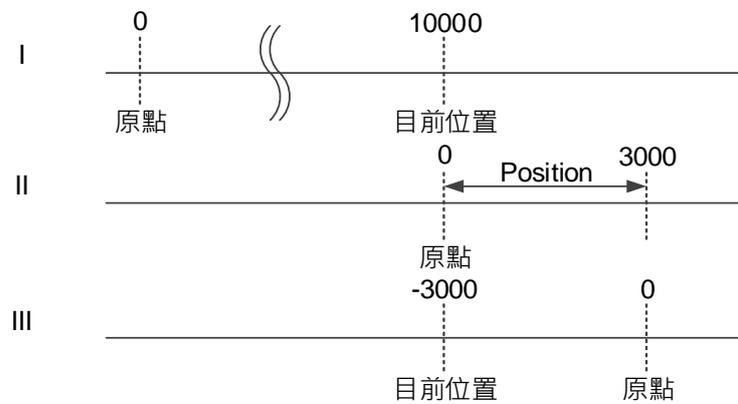
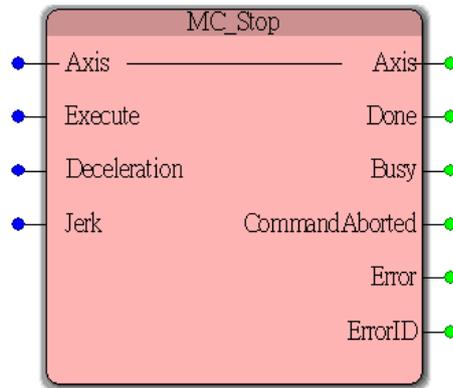


圖 5.2.2.1.1 MC\_Home 範例說明

# 5

## 5.2.2.2 MC\_Stop

- 類型  
Function Block
- 功能描述  
依照使用者輸入的減速度值來執行一個受控的單軸運動停止指令。
- 功能塊



■ 輸入輸出共用變數

名稱	資料型態	設定值(預設值)	功能
Axis	AXIS_REF	1 ~ 16 (-)	指定運動軸編號

■ 輸入變數

名稱	資料型態	設定值(預設值)	功能
Execute	BOOL	True / False (False)	當Execute為上升緣時，此功能即開始執行
Deceleration	LREAL	負數、正數或0 (0)	運動停止命令之減速度數值
Jerk	LREAL	負數、正數或0 (0)	運動停止命令之減減速度數值，將影響運動的平滑程度 (單位：使用者單位/秒 <sup>3</sup> )

■ 輸出變數

名稱	資料型態	設定值(預設值)	功能
Done	BOOL	True / False (False)	絕對定位完成時為True
Busy	BOOL	True / False (False)	指令被觸發執行時為True
Command Aborted	BOOL	True / False (False)	指令被中斷時為True
Error	BOOL	True / False (False)	指令錯誤發生時為True
ErrorID	WORD	0x0000~0xFFFF (0)	指令錯誤發生時所記錄的錯誤碼；錯誤碼詳細說明請參考第5.1.4章節

■ 輸出變數的變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Done	當絕對定位完成時	<ul style="list-style-type: none"> <li>■ 當Execute轉為False時</li> <li>■ 若Execute為False而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Busy	當Execute上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
CommandAborted	<ul style="list-style-type: none"> <li>■ 當此功能塊指令被其他緩衝模式設為mcAborting的指令中斷時</li> <li>■ 當此功能塊指令被MC_Stop中斷時</li> </ul>	<ul style="list-style-type: none"> <li>■ 在Execute下緣時</li> <li>■ 當Execute為False而CommandAborted轉為True，此時CommandAborted維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在ErrorID)	在Execute下緣時 (清除ErrorID紀錄之錯誤碼)

■ 輸出變數的變化時序

下方將表示功能塊正確完成程序。

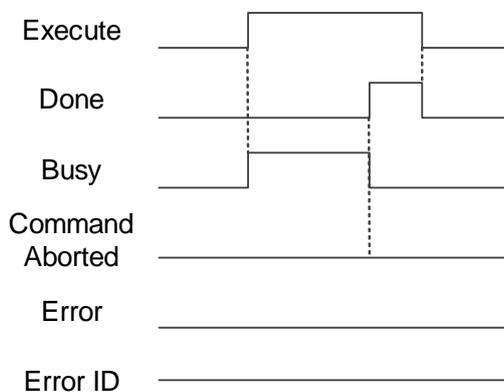


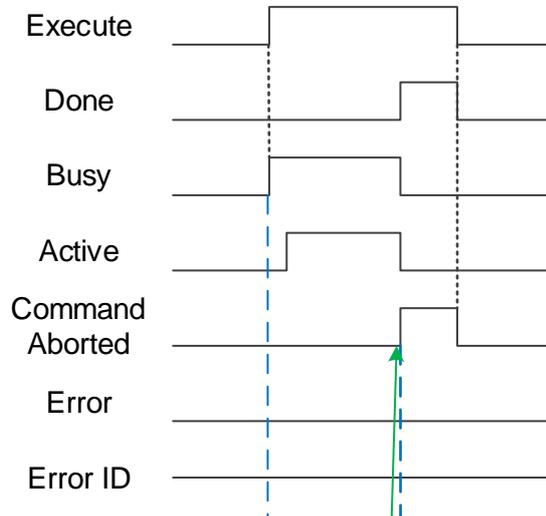
圖 5.2.2.2.1 MC\_Stop 變數時序圖(時序詳細說明請參考 5.1.3 節)

# 5

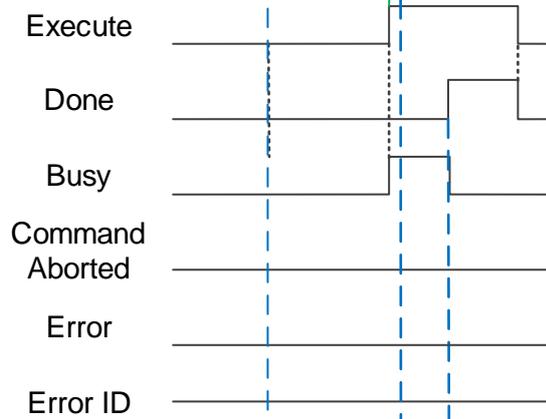
■ 速度時間圖

下圖為一單軸運動中，MC\_Stop 功能被啟動而中斷此單軸運動命令，使用者可觀察到其速度圖的變化。

軸上運動時序



MC\_Stop執行時序



運動速度圖

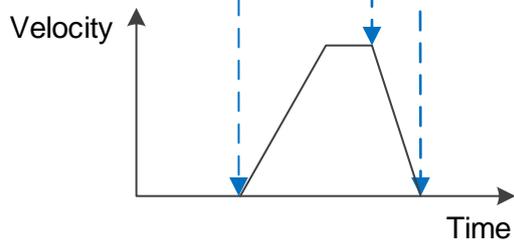


圖 5.2.2.2.2 MC\_Stop 速度時間圖

■ 參考範例

下例提供 13 號軸被啟動 Stop 的運動命令，並解釋了兩種 Execute 的執行方式，一是保持住執行，直到完成後才釋放，另一個是觸發後即釋放的兩種啟動方式。

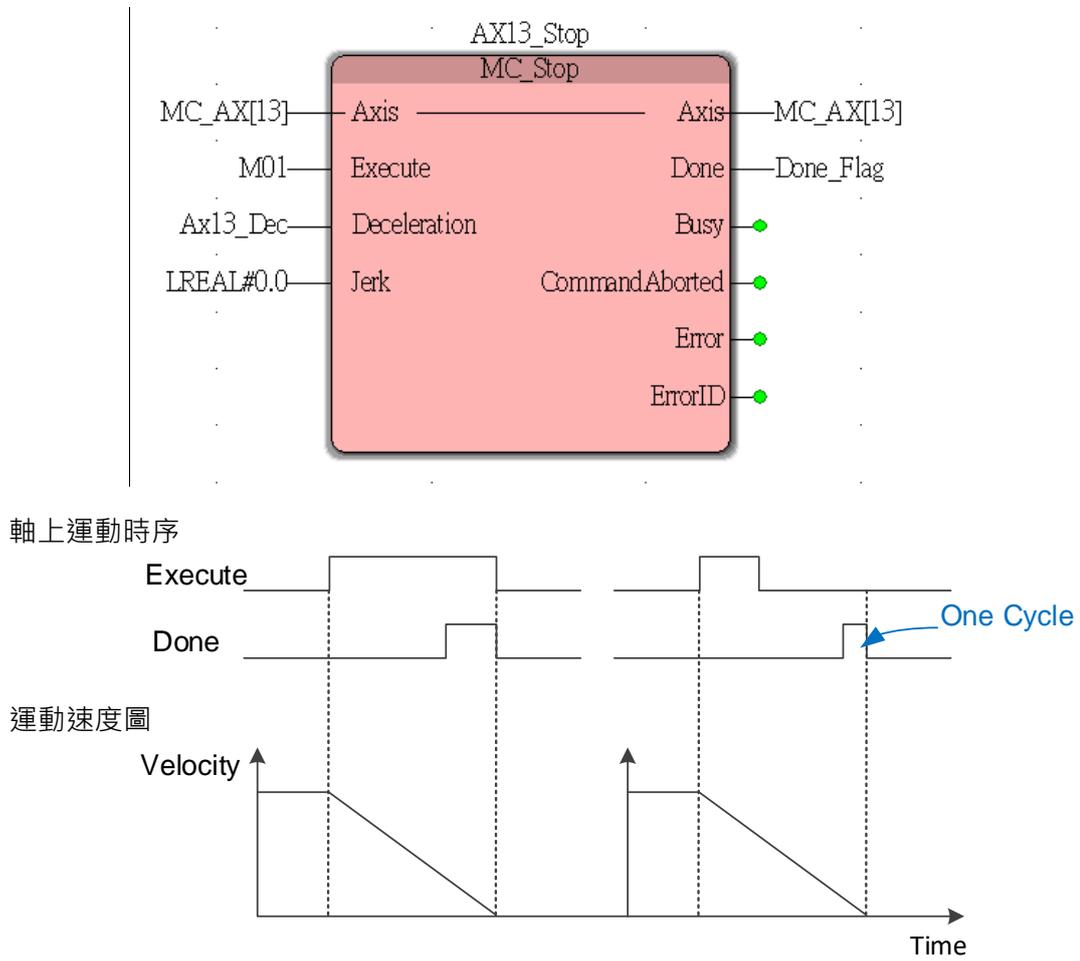
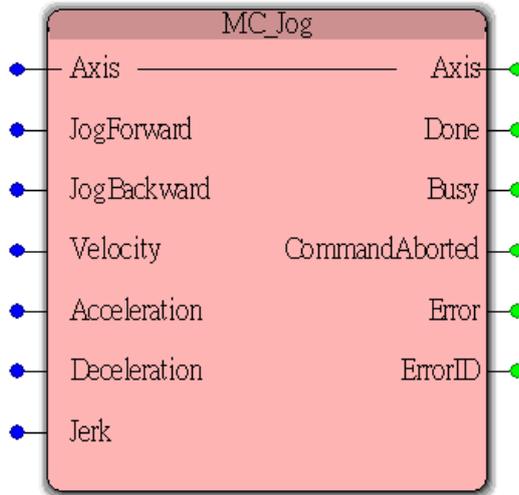


圖 5.2.2.2.3 MC\_Stop 範例說明

# 5

## 5.2.2.3 MC\_Jog

- 類型  
Function Block
- 功能描述  
對指定軸執行連續型寸動指令。
- 功能塊



■ 輸入輸出共用變數

名稱	資料型態	設定值 (預設值)	功能
Axis	AXIS_REF	1 ~ 16 (-)	指定運動軸編號

■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
JogForward	BOOL	True / False (False)	當JogForward為True時·開始執行連續寸動指令；反之則停止。
JogBackward	BOOL	True / False (False)	當JogBackward為True時·開始執行連續寸動指令；反之則停止。
Velocity	LREAL	正數或0 (0)	最大速度數值 (使用者單位/秒)
Acceleration	LREAL	正數或0 (0)	加速度數值 (使用者單位/秒 <sup>2</sup> )
Deceleration	LREAL	正數或0 (0)	減速度數值 (使用者單位/秒 <sup>2</sup> )
Jerk	LREAL	正數或0 (0)	加加速度數值 (使用者單位/秒 <sup>3</sup> )

■ 輸出變數

名稱	資料型態	設定值(預設值)	功能
Done	BOOL	True / False (False)	絕對定位完成時為True
Busy	BOOL	True / False (False)	指令被觸發執行時為True
Command Aborted	BOOL	True / False (False)	指令被中斷時為True
Error	BOOL	True / False (False)	指令錯誤發生時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	指令錯誤發生時所記錄的錯誤碼；錯誤碼詳細說明請參考第5.1.4章節

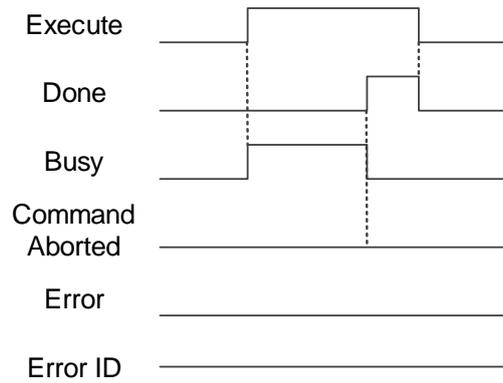
■ 輸出變數的變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Done	絕對定位完成時	<ul style="list-style-type: none"> <li>■ 在Execute轉為False時</li> <li>■ 若Execute為False而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Busy	在Execute上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
Command Aborted	<ul style="list-style-type: none"> <li>■ 當此功能塊指令被其他緩衝模式設為mcAborting的指令中斷時</li> <li>■ 當此功能塊指令被MC_Stop中斷時</li> </ul>	<ul style="list-style-type: none"> <li>■ 在Execute下緣時</li> <li>■ 當Execute為False，而CommandAborted轉為True，此時CommandAborted維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在ErrorID)	在Execute下緣時 (清除ErrorID紀錄之錯誤碼)

# 5

■ 輸出變數的變化時序

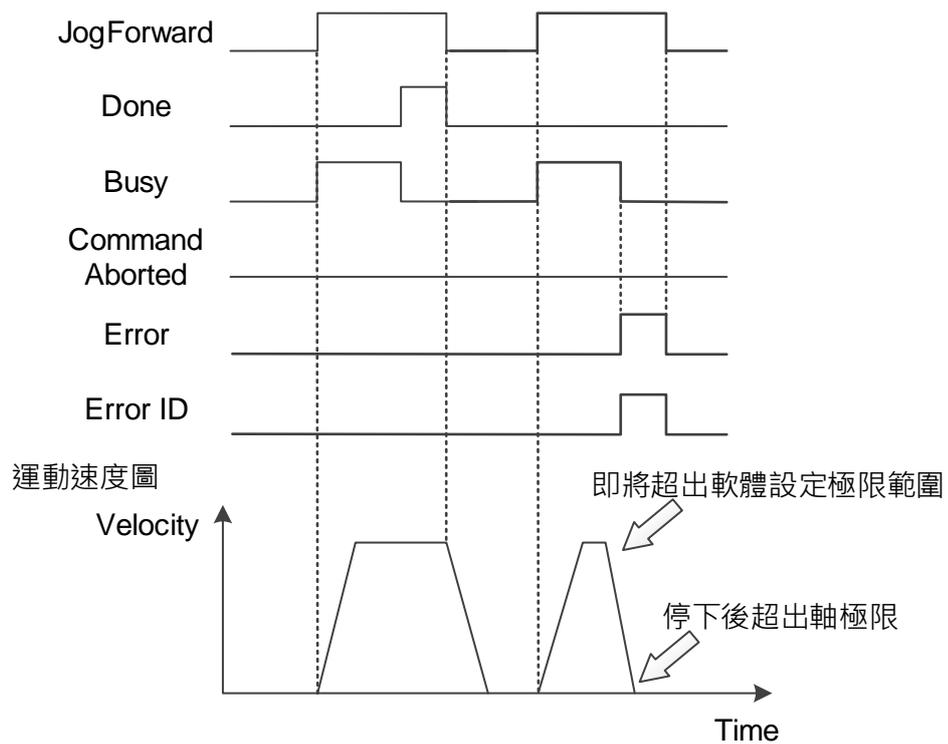
此時序圖表現了一個完整的 Jog 運動命令啟動至停止。



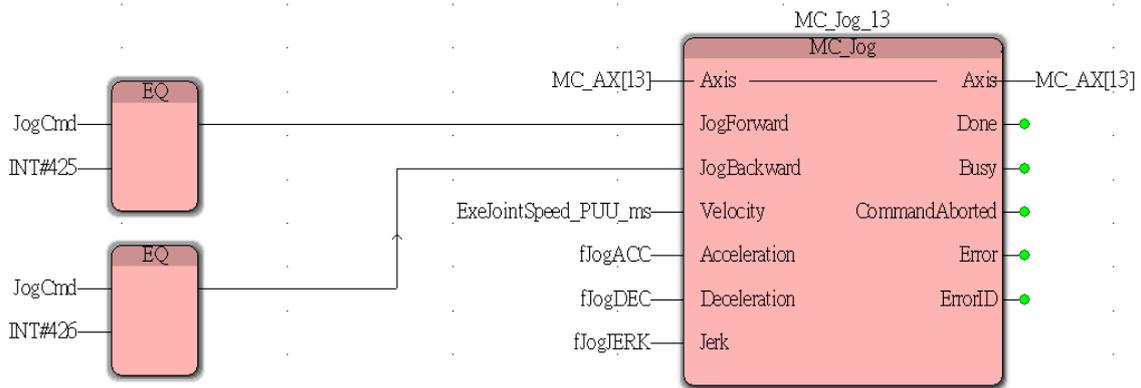
■ 速度時間圖

以下時間圖提供了兩種時間圖，一是完整完成 Jog(寸動)運動命令，另一個是在 Jog 運動命令執行時，因為錯誤產生而中斷。

MC\_Jog執行時序



■ 參考範例



此範例為連結外部命令執行向前或向後連續寸動指令

1. JogCmd 影響軸寸動方向
2. ExeJointSpeed\_PUU\_ms 影響寸動軸最大速度
3. fJogACC 影響寸動軸的加速度
4. fJogDEC 影響寸動軸的減速度
5. fJogJERK 影響寸動軸的加加速度

5.2.2.4 MC\_MoveAbsolute

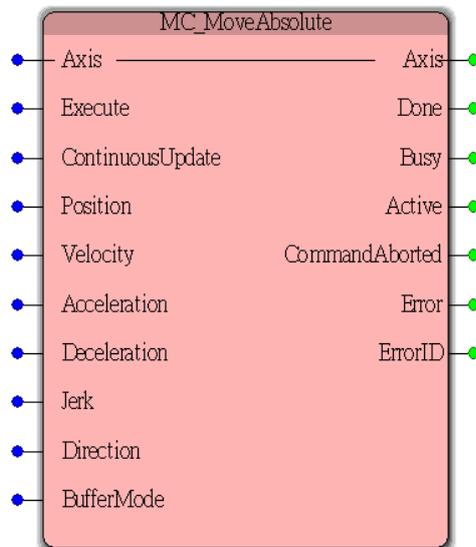
■ 類型

Function Block

■ 功能描述

絕對位置的運動命令，將依照使用者所規劃之速度、加速度、減速度等運動參數進行軸運動。

■ 功能塊



## 5

## ■ 輸入輸出共用變數

名稱	資料型態	設定值(預設值)	功能
Axis	AXIS_REF	1~16 (-)	指定運動軸編號

## ■ 輸入變數

名稱	資料型態	設定值(預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時，執行此指令
Continuous Update	BOOL	True / False (False)	位置、速度和加速度等資訊即時更新。(*無支援)
Position	LREAL	負數，正數或0 (0)	絕對位置 (使用者單位)
Velocity	LREAL	正數 (0)	目標最大速度數值 (使用者單位/秒)
Acceleration	LREAL	正數或0 (0)	加速度數值 (使用者單位/秒 <sup>2</sup> )
Deceleration	LREAL	正數或0 (0)	減速度數值 (使用者單位/秒 <sup>2</sup> )
Jerk	LREAL	正數或0 (0)	加加速度數值 (使用者單位/秒 <sup>3</sup> )
Direction	MC_DIRECTION	0: mcPositiveDirection 1: mcShortestWay 2: mcNegativeDirection 3: mcCurrentDirection	指定軸伺服馬達運轉方向 PositiveDirection：正向 ShortesetWay：最短距離 NegativeDirection：負向 CurrentDirection：目前方向 (*無支援)
BufferMode	MC_BUFFER_MODE	0: mcAborting 1: mcBufered	指定此功能塊的緩衝行為

## ■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	絕對定位完成時為True
Busy	BOOL	True / False (False)	指令被觸發執行時為True
Active	BOOL	True / False (False)	指定軸受控制時為True
Command Aborted	BOOL	True / False (False)	指令被中斷時為True
Error	BOOL	True / False (False)	指令錯誤發生時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	指令錯誤發生時所記錄的錯誤碼；錯誤碼詳細說明請參考第5.1.4章節

■ 輸出變數的變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Done	絕對定位完成時	<ul style="list-style-type: none"> <li>■ 當Execute轉為False時</li> <li>■ 若Execute為False，而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Busy	Execute上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
Active	軸運動開始時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
CommandAborted	<ul style="list-style-type: none"> <li>■ 此功能塊指令被其他緩衝模式設為mcAborting的指令中斷時</li> <li>■ 此功能塊指令被MC_Stop中斷時</li> </ul>	<ul style="list-style-type: none"> <li>■ 在Execute下緣時</li> <li>■ 當Execute為False而CommandAborted轉為True，此時CommandAborted維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Error / ErrorID	指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在ErrorID)	在Execute下緣時 (清除ErrorID紀錄之錯誤碼)

■ 輸出變數的變化時序

此時序圖提供了三種不同情況的時序，包含了「完整執行」、「因中斷命令而停止」以及「產生錯誤而停止」的情況時序。

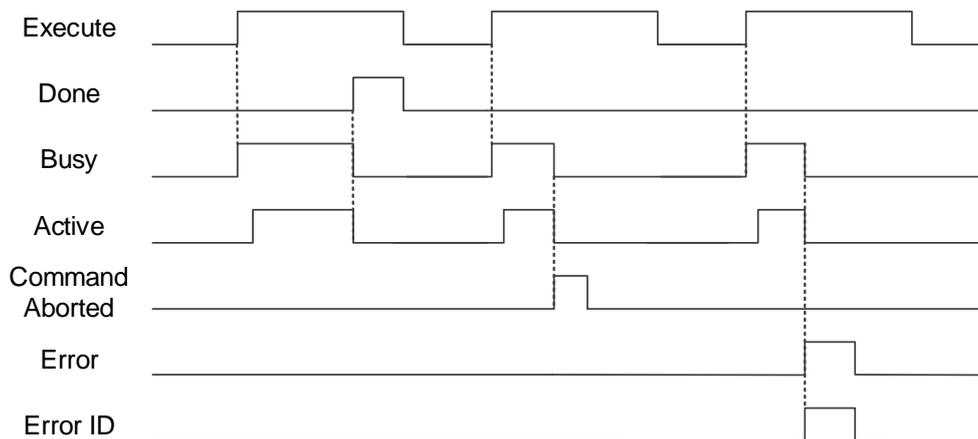


圖 5.2.2.4.1 (時序詳細說明請參考 5.1.3 節)

# 5

## ■ 速度時間圖

此速度時間圖解釋了有無加加速度(Jerk)對於速度規劃的影響，無加加速度的速度規劃則無平滑曲線的效果，有加加速度的速度規劃則有平滑曲線的效果，但會增加運動時間。

MC\_MoveAbsolute執行時序

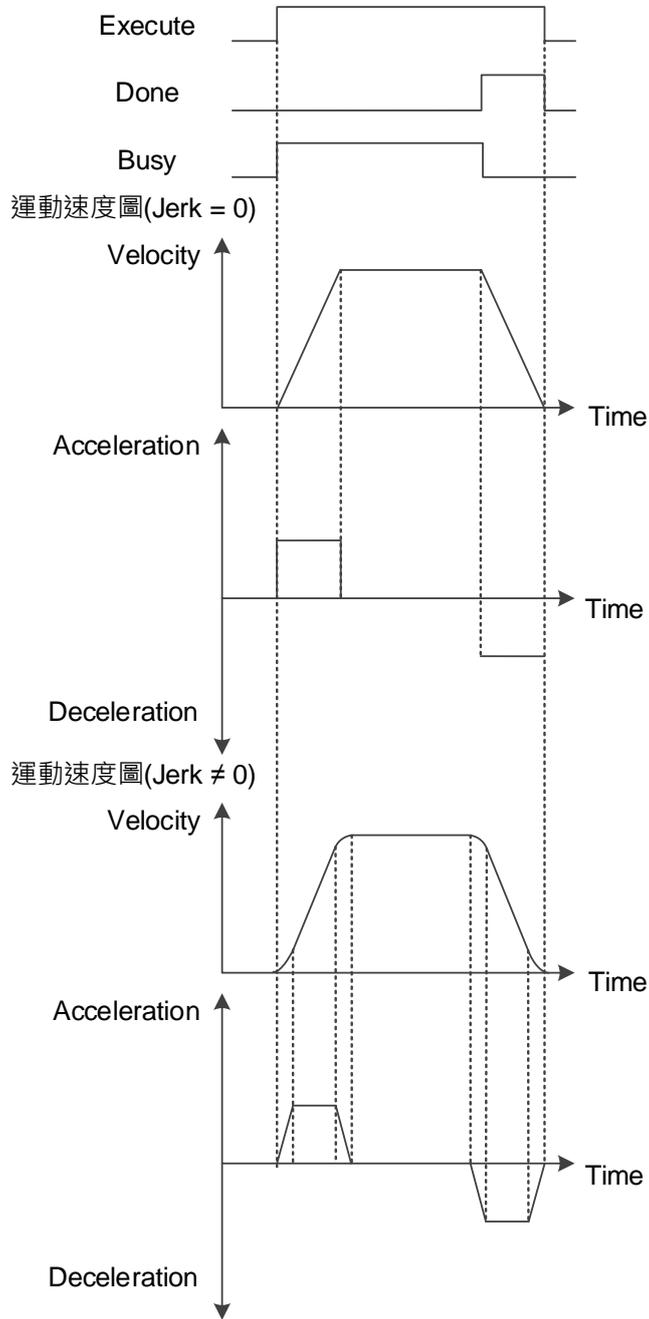
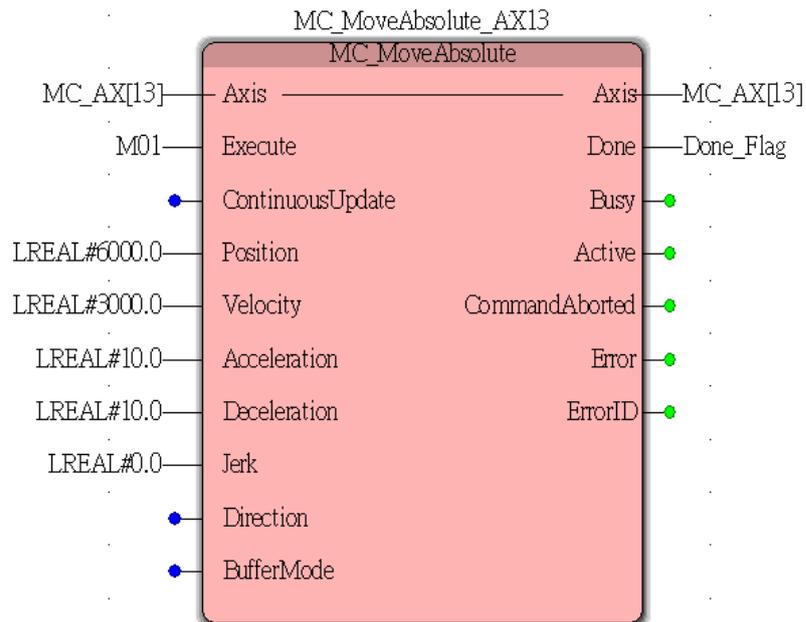


圖 5.2.2.4.2

■ 參考範例

此範例為 MC\_MoveAbsolute 的各項參數輸入，移動至絕對位置 6000 PUU 上。

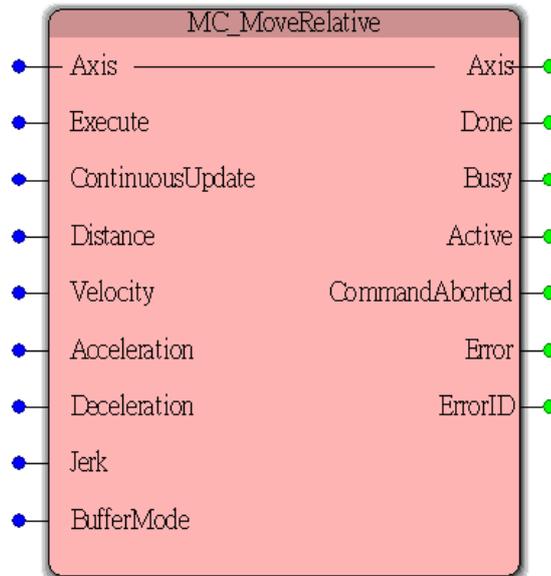


1. M01 觸發此功能
2. 設定運動參數位置為 6000 PUU
3. 設定運動參數速度為 3000 PUU / S
4. 設定運動參數加速度為 10 PUU / S<sup>2</sup>
5. 設定運動參數減速度為 10 PUU / S<sup>2</sup>
6. 設定運動參數加加速度為 0 PUU / S<sup>3</sup>

# 5

## 5.2.2.5 MC\_MoveRelative

- 類型  
Function Block
- 功能描述  
相對距離的運動命令。依照使用者輸入的目標速度、加速度、減速度與加加速度執行一個移動相對距離的運動命令。
- 功能塊



■ 輸入輸出共用變數

名稱	資料型態	設定值 (預設值)	功能
Axis	AXIS_REF	1 ~ 16 (-)	指定運動軸編號

■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時，執行此指令
Continuous Update	BOOL	True / False (False)	位置、速度和加速度等資訊即時更新。(*無支援)
Position	LREAL	負數、正數或0 (0)	相對位置(使用者單位)
Velocity	LREAL	正數 (0)	目標最大速度數值 (使用者單位/秒)
Acceleration	LREAL	正數或0 (0)	加速度數值(使用者單位/秒 <sup>2</sup> )
Deceleration	LREAL	正數或0 (0)	減速度數值(使用者單位/秒 <sup>2</sup> )
Jerk	LREAL	正數或0 (0)	加加速度數值 (使用者單位/秒 <sup>3</sup> )
Direction	MC_DIRECTION	0: mcPositiveDirection 1: mcShortestWay 2: mcNegativeDirection 3: mcCurrentDirection	指定軸伺服馬達運轉方向 PositiveDirection：正向 ShortesetWay：最短距離

名稱	資料型態	設定值 (預設值)	功能
			NegativeDirection : 負向 CurrentDirection : 目前方向 (*無支援)
BufferMode	MC_BUFFER_MODE	0: mcAborting 1: mcBufered	指定此功能塊的緩衝行為

■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	絕對定位完成時為True
Busy	BOOL	True / False (False)	指令被觸發執行時為True
Active	BOOL	True / False (False)	指定軸受控制時為True
Command Aborted	BOOL	True / False (False)	指令被中斷時為True
Error	BOOL	True / False (False)	指令錯誤發生時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	指令錯誤發生時所記錄的錯誤碼；錯誤碼詳細說明請參考第5.1.4章節

■ 輸出變數的變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Done	絕對定位完成時	<ul style="list-style-type: none"> <li>■ Execute轉為False時</li> <li>■ 若Execute為False而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Busy	Execute上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
Active	軸運動開始時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
CommandAborted	<ul style="list-style-type: none"> <li>■ 當此功能塊指令被其他緩衝模式設為mcAborting的指令中斷時</li> <li>■ 當此功能塊指令被MC_Stop中斷時</li> </ul>	<ul style="list-style-type: none"> <li>■ 在Execute下緣時</li> <li>■ 當Execute為False而CommandAborted轉為True，此時CommandAborted維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在ErrorID)	在Execute下緣時 (清除ErrorID紀錄之錯誤碼)

# 5

■ 輸出變數的變化時序

此時序圖提供了三種不同情況的時序，包含了「完整執行」、「因中斷命令產生而停止」以及「因錯誤產生而停止」的情況時序。

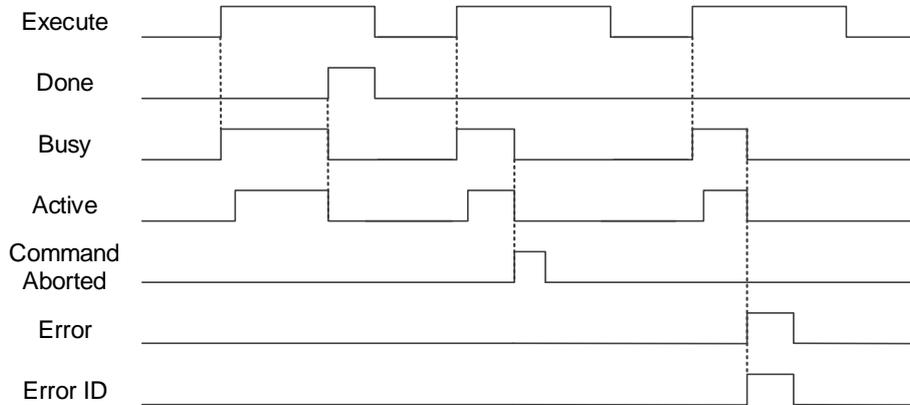


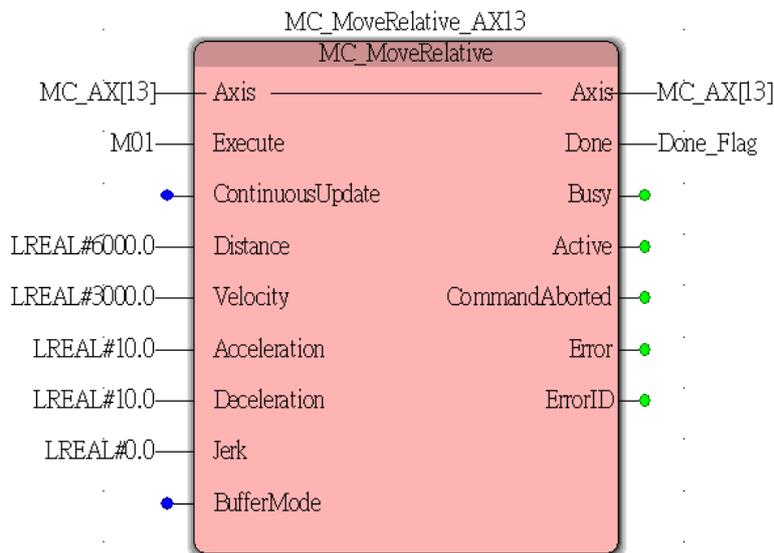
圖 5.2.2.19 MC\_MoveRelative 變數時序圖(時序詳細說明請參考 5.1.3 節)

■ 速度時間圖

請參考圖 5.2.2.4.2 MC\_MoveAbsolute 速度時間圖

■ 參考範例

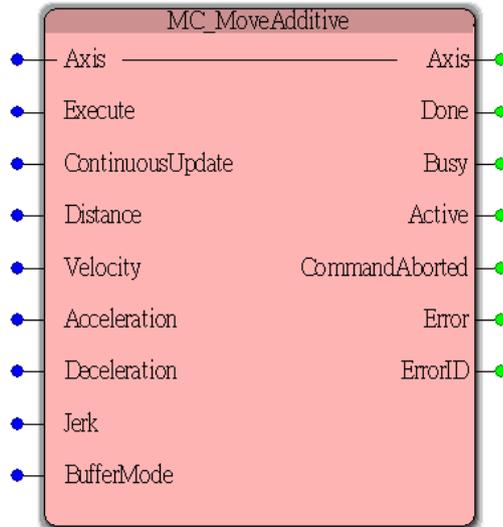
此範例為 MC\_MoveRelative 的各項參數輸入，移動至相對於當前位置 6000 PUU 上。



1. M01 觸發此功能
2. 設定運動參數相對位置為 6000 PUU
3. 設定運動參數速度為 3000 PUU / S
4. 設定運動參數加速度為 10 PUU / S<sup>2</sup>
5. 設定運動參數減速度為 10 PUU / S<sup>2</sup>
6. 設定運動參數加加速度為 0 PUU / S<sup>3</sup>

### 5.2.2.6 MC\_MoveAdditive

- 類型  
Function Block
- 功能描述  
依照使用者輸入的目標速度、加速度、減速度與加加速度，來執行一個移動相對距離的運動命令，此命令會附加到最新指定的相對移動運動。
- 功能塊



■ 輸入輸出共用變數

名稱	資料型態	設定值 (預設值)	功能
Axis	AXIS_REF	1 ~ 16 (-)	指定運動軸編號

■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時，執行此指令
Continuous Update	BOOL	True / False (False)	位置、速度和加速度等資訊即時更新。(*無支援)
Position	LREAL	負數、正數或0 (0)	相對位置(使用者單位)
Velocity	LREAL	正數 (0)	目標最大速度數值 (使用者單位 / 秒)
Acceleration	LREAL	正數或0 (0)	加速度數值 (使用者單位 / 秒 <sup>2</sup> )
Deceleration	LREAL	正數或0 (0)	減速度數值 (使用者單位 / 秒 <sup>2</sup> )
Jerk	LREAL	正數或0 (0)	加加速度數值 (使用者單位 / 秒 <sup>3</sup> )
Direction	MC_DIRECTION	0: mcPositiveDirection 1: mcShortestWay 2: mcNegativeDirection 3: mcCurrentDirection	指定軸伺服馬達運轉方向 PositiveDirection：正向

## 5

名稱	資料型態	設定值 (預設值)	功能
			ShortesetWay : 最短距離 NegativeDirection : 負向 CurrentDirection : 目前方向 (*無支援)
BufferMode	MC_BUFFER_MODE	0: mcAborting 1: mcBufered	指定此功能塊的緩衝行為

### ■ 輸出變數

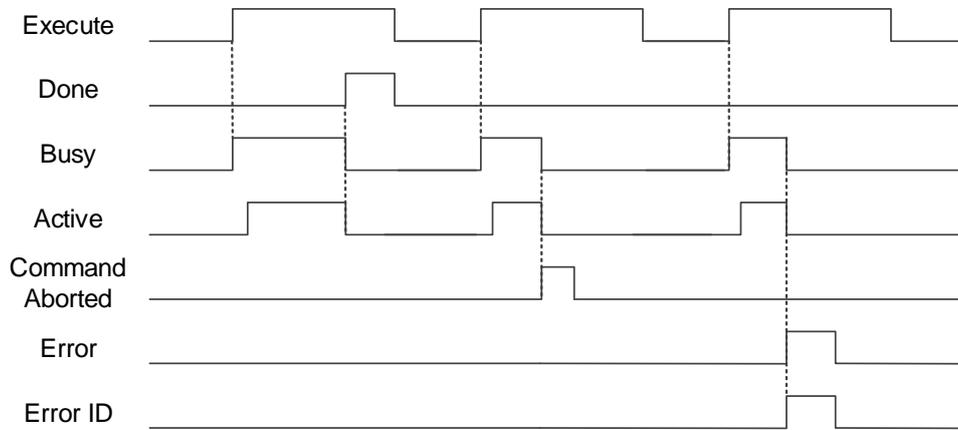
名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	絕對定位完成時為True
Busy	BOOL	True / False (False)	指令被觸發執行時為True
Active	BOOL	True / False (False)	指定軸受控制時為True
Command Aborted	BOOL	True / False (False)	指令被中斷時為True
Error	BOOL	True / False (False)	指令錯誤發生時為True
ErrorID	WORD	0x0000~0xFFFF(0)	指令錯誤發生時所記錄的錯誤碼；錯誤碼詳細說明請參考第5.1.4章節

### ■ 輸出變數的變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Done	絕對定位完成時	<ul style="list-style-type: none"> <li>Execute轉為False時</li> <li>若Execute為False而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Busy	Execute上緣觸發時	<ul style="list-style-type: none"> <li>在Done上緣時</li> <li>在Error上緣時</li> <li>在CommandAborted上緣時</li> </ul>
Active	軸運動開始時	<ul style="list-style-type: none"> <li>在Done上緣時</li> <li>在Error上緣時</li> <li>在CommandAborted上緣時</li> </ul>
CommandAborted	<ul style="list-style-type: none"> <li>當此功能塊指令被其他緩衝模式設為mcAborting的指令中斷時</li> <li>當此功能塊指令被MC_Stop中斷時</li> </ul>	<ul style="list-style-type: none"> <li>在Execute下緣時</li> <li>當Execute為False而CommandAborted轉為True，此時CommandAborted維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在ErrorID)	在Execute下緣時 (清除ErrorID紀錄之錯誤碼)

■ 輸出變數的變化時序

此時序圖提供了三種不同情況的時序，包含了完整執行、因中斷命令產生而停止以及錯誤產生而停止的情況時序。



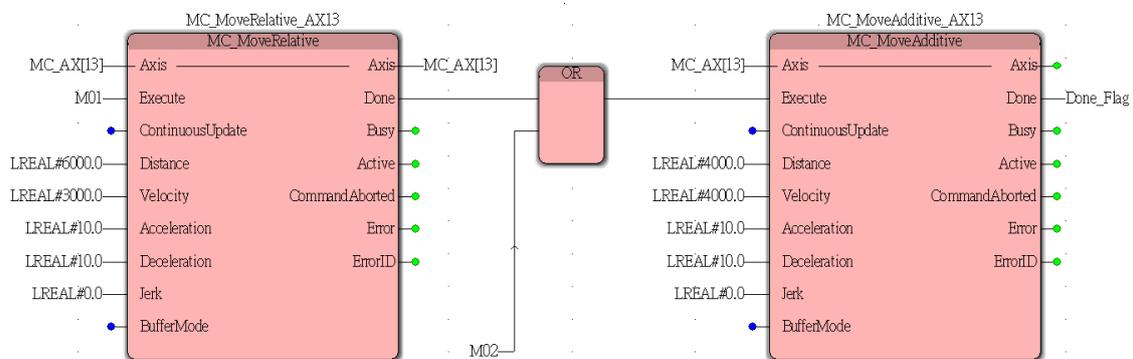
(時序詳細說明請參考 5.1.3 節)

■ 速度時間圖

請參考圖 5.2.2.4.2 MC\_MoveAbsolute 速度時間圖

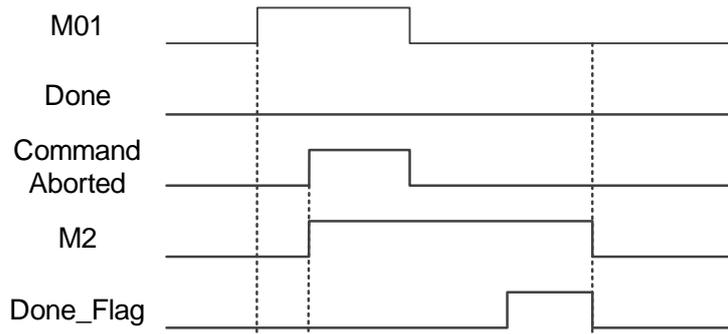
■ 參考範例

此範例提供了 MC\_MoveAdditive 功能塊的運動所執行的距離。在完成了第一個功能塊 MC\_MoveAbsolute 後，第二個 MC\_MoveAdditive 會執行相對運動命令，並且移動至所指定的位置上。

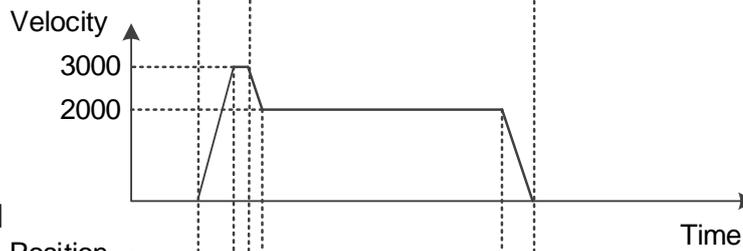


# 5

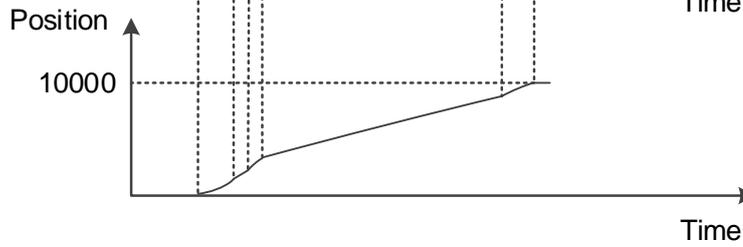
軸上運動時序



運動速度圖

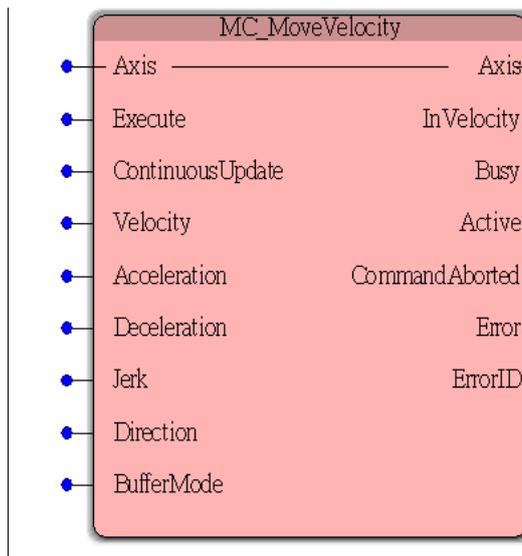


運動位置圖



### 5.2.2.7 MC\_MoveVelocity

- 類型  
Function Block
- 功能描述  
將以指定速度進行軸運動。
- 功能塊



■ 輸入輸出共用變數

名稱	資料型態	設定值 (預設值)	功能
Axis	AXIS_REF	1 ~ 16 (-)	指定運動軸編號

■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時·執行此指令
Continuous Update	BOOL	True / False (False)	位置、速度和加速度等資訊即時更新。(*無支援)
Velocity	LREAL	正數(0)	目標最大速度數值 (使用者單位 / 秒)
Acceleration	LREAL	正數或 0 (0)	加速度數值(使用者單位 / 秒 <sup>2</sup> )
Deceleration	LREAL	正數或 0 (0)	減速度數值(使用者單位 / 秒 <sup>2</sup> )
Jerk	LREAL	正數或 0 (0)	加加速度數值 (使用者單位 / 秒 <sup>3</sup> )
Direction	MC_DIRECTION	0: mcPositiveDirection 1: mcShortestWay 2: mcNegativeDirection 3: mcCurrentDirection	指定軸伺服馬達運轉方向 PositiveDirection : 正向 ShortesetWay : 最短距離 NegativeDirection : 負向 CurrentDirection : 目前方向 (*無支援)

## 5

名稱	資料型態	設定值 (預設值)	功能
BufferMode	MC_BUFFER_MODE	0: mcAborting 1: mcBufered	指定此功能塊的緩衝行為

### ■ 輸出變數

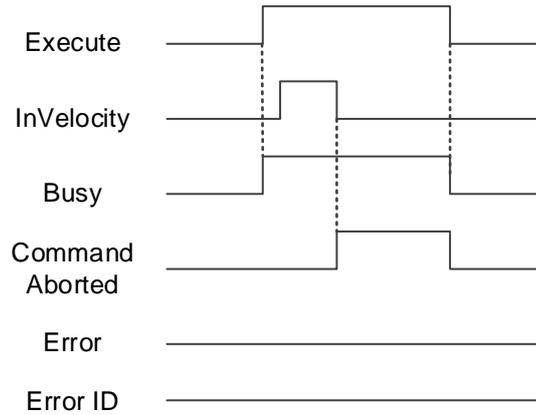
名稱	資料型態	設定值 (預設值)	功能
InVelocity	BOOL	True / False (False)	達到目標速度時為True
Busy	BOOL	True / False (False)	指令被觸發執行時為True
Active	BOOL	True / False (False)	指定軸受控制時為True
Command Aborted	BOOL	True / False (False)	指令被中斷時為True
Error	BOOL	True / False (False)	指令錯誤發生時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	指令錯誤發生時所記錄的錯誤碼；錯誤碼詳細說明請參考第 5.1.4 章節

### ■ 輸出變數的變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
InVelocity	當速度達到設定值	<ul style="list-style-type: none"> <li>■ 當Execute轉為False時</li> <li>■ 若Execute為False而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為 False</li> </ul>
Busy	Execute上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
Active	軸運動開始時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
Command Aborted	<ul style="list-style-type: none"> <li>■ 當此功能塊指令被其他緩衝模式設為 mcAborting 的指令中斷時</li> <li>■ 當此功能塊指令被 MC_Stop中斷時</li> </ul>	<ul style="list-style-type: none"> <li>■ 在Execute下緣時</li> <li>■ 當Execute為False而CommandAborted 轉為 True，此時 CommandAborted 維持一個掃描週期的 True 狀態後，立刻轉為 False</li> </ul>
Error / ErrorID	指令的執行條件或輸入值發生錯誤時 (錯誤碼紀錄在 ErrorID)	在Execute下緣時 (清除 ErrorID 紀錄之錯誤碼)

■ 輸出變數的變化時序

此時序圖提供完整運動過程的時序變化。



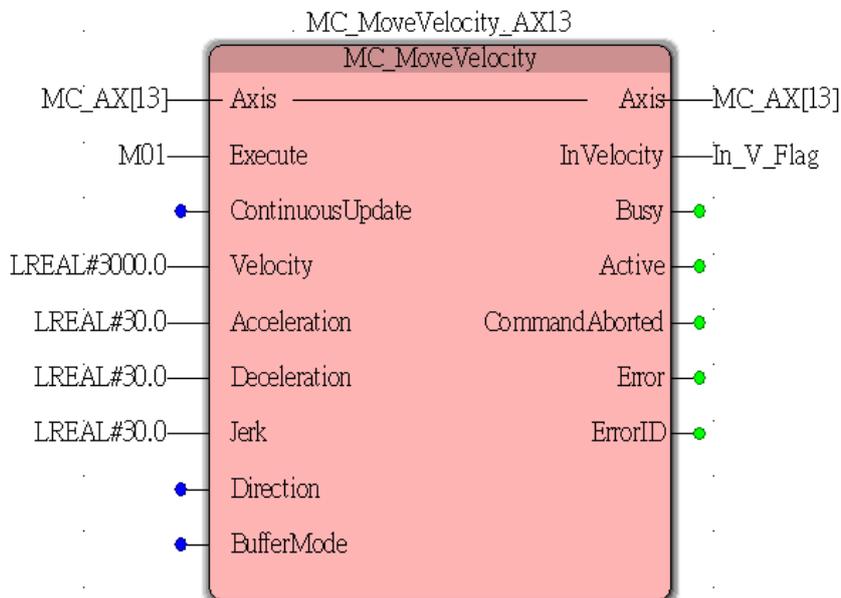
(時序詳細說明請參考 5.1.3 節)

■ 速度時間圖

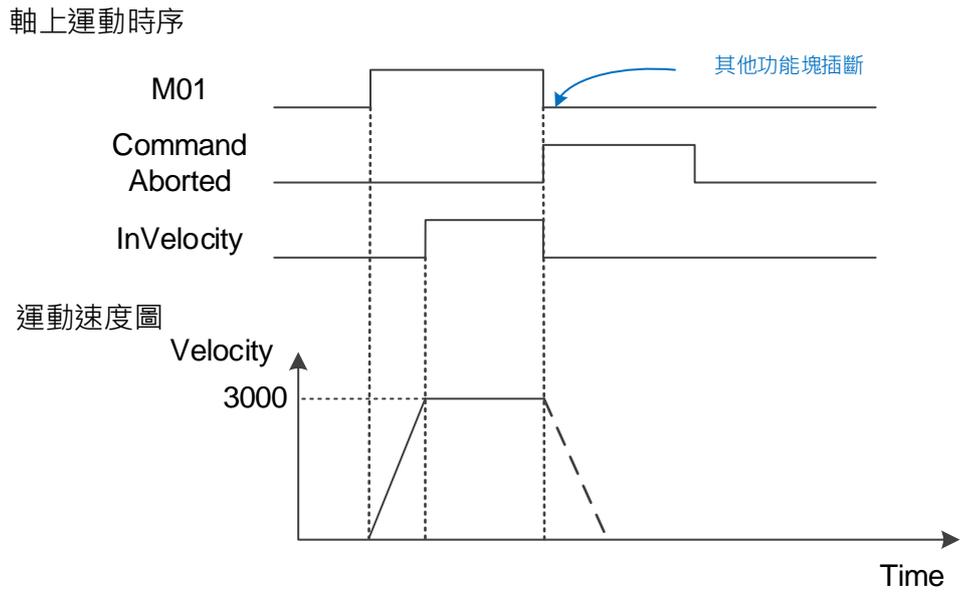
請參考圖 5.2.2.4.2 MC\_MoveAbsolute 速度時間圖

■ 參考範例

此範例提供了軸在 MC\_MoveVelocity 的命令下可以達到目標速度，並展現了其他功能塊的插斷，導致軸速度開始減速停止。

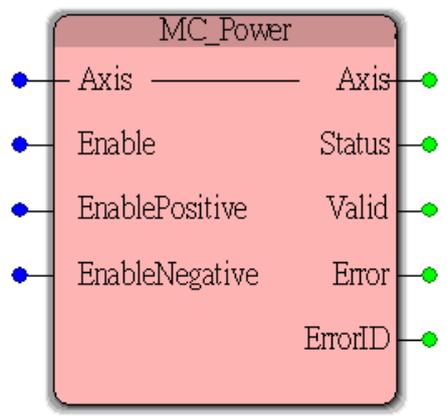


# 5



### 5.2.2.8 MC\_Power

- 類型  
Function Block
- 功能描述  
控制伺服啟動開關。
- 功能塊



■ 輸入輸出共用變數

名稱	資料型態	設定值 (預設值)	功能
Axis	AXIS_REF	1 ~ 16 (-)	指定運動軸編號

■ 輸入變數

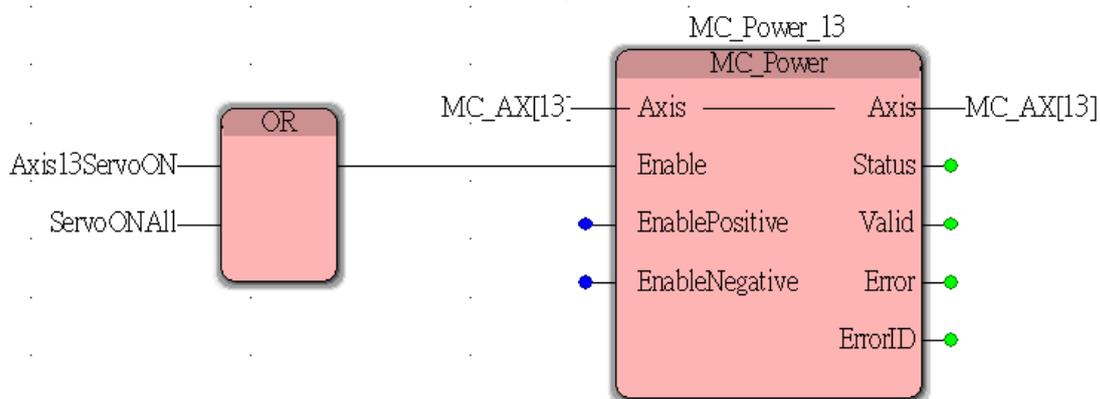
名稱	資料型態	設定值 (預設值)	功能
Enable	BOOL	True / False (False)	Enable為True時，啟用此功能
EnablePositive	BOOL	True / False (False)	位置、速度和加速度等資訊即時更新。(*無支援)
EbableNegative	BOOL	True / False (False)	相對位置(*無支援)

■ 輸出變數

名稱	資料型態	設定值(預設值)	功能
Status	BOOL	True / False (False)	伺服狀態，伺服啟動時為 True; 反之則為False
Valid	BOOL	True / False (False)	成立時表示此Function Block 有效
Error	BOOL	True / False (False)	錯誤發生時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	錯誤發生時所記錄的錯誤碼；錯誤碼詳細說明請參考第5.1.4 章節

■ 參考範例

此範例提供了透過外部命令來控制伺服啟動功能。

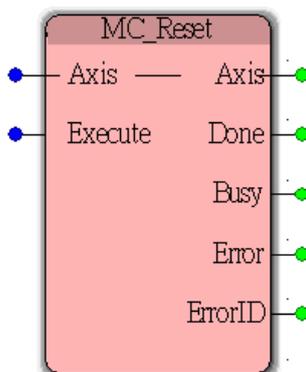


1. Axis13ServoON 連接外部命令，如數位輸入變數或記憶體位置。
2. ServoONAll 連接外部命令，如數位輸入變數或記憶體位置。

# 5

## 5.2.2.9 MC\_Reset

- 類型  
Function Block
- 功能描述  
清除或重置所有與對應軸相關的錯誤。
- 功能塊



### ■ 輸入輸出共用變數

名稱	資料型態	設定值 (預設值)	功能
Axis	AXIS_REF	1 ~ 16 (-)	指定運動軸編號

### ■ 輸入變數

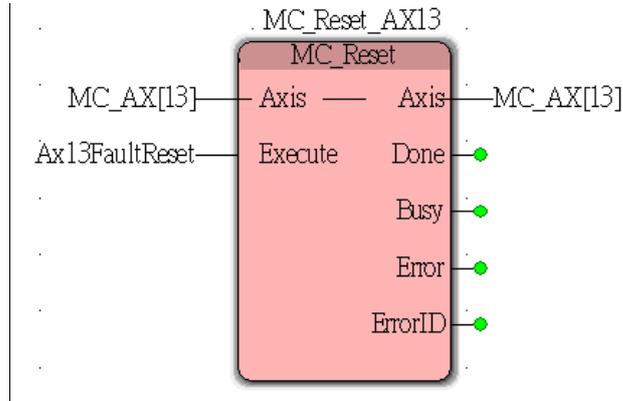
名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute 上緣時 · 執行此指令

### ■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	絕對定位完成時為True
Busy	BOOL	True / False (False)	指令被觸發執行時為True
Error	BOOL	True / False (False)	指令錯誤發生時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	指令錯誤發生時所記錄的錯誤碼；錯誤碼詳細說明請參考第5.1.4章節

■ 參考範例

此範例提供了透過外部訊號清除軸錯誤。



Ax13FaultReset 連接外部命令，如數位輸入變數或記憶體位置。

5.2.2.10 MC\_SetOverride

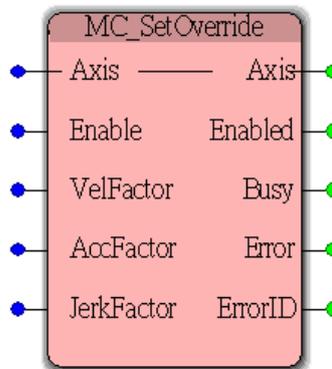
■ 類型

Function Block

■ 功能描述

將依照輸入的速度、加速度、減速度以及加加速度因子來計算運動參數。

■ 功能塊



■ 輸入輸出共用變數

名稱	資料型態	設定值 (預設值)	功能
Axis	AXIS_REF	1 ~ 16 (-)	指定運動軸編號

■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Enable	BOOL	True / False (False)	Enable為True時，啟用此功能
VelFactor	REAL	正數或0 (1.0)	超馳控制速度係數
AccFactor	REAL	正數、負數或0 (1.0)	超馳控制係數，正數為加速度係數，負數為減速度係數 (*無支援)
JerkFactor	REAL	正數或0 (1.0)	超馳控制加加速度係數

# 5

■ 輸出變數

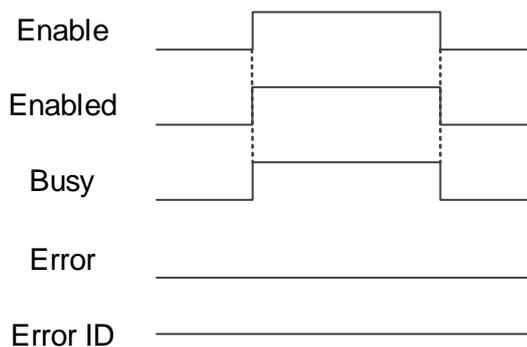
名稱	資料型態	設定值 (預設值)	功能
Enabled	BOOL	True / False (False)	True時表示超馳控制被設立成功
Busy	BOOL	True / False (False)	指令執行時為True
Error	BOOL	True / False (False)	指令錯誤發生時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	指令錯誤發生時所記錄的錯誤碼；錯誤碼詳細說明請參考第 5.1.4 章節

■ 輸出變數的變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Enabled	當設定值成立後	<ul style="list-style-type: none"> <li>■ 當Execute轉為False時</li> <li>■ 若Execute為False而Done 轉為 True，此時Done維持一個掃描週期的True狀態後，立刻轉為 False</li> </ul>
Busy	當Execute上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在 ErrorID)	在Execute下緣時 (清除ErrorID紀錄之錯誤碼)

■ 輸出變數的變化時序

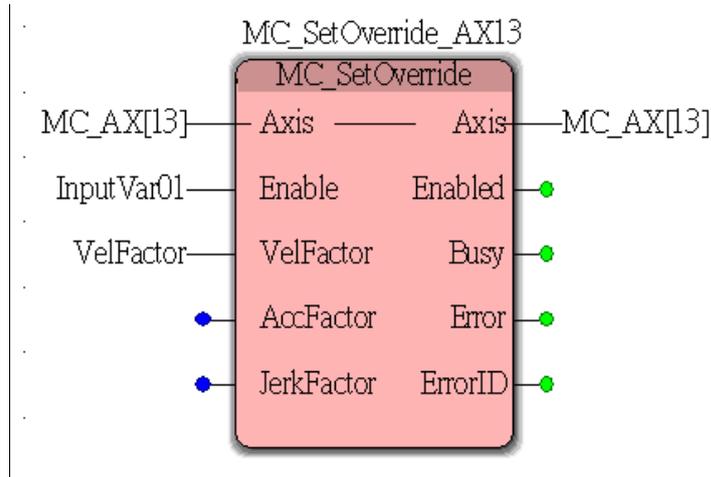
此時序圖表示了完整的運動時序。



(時序詳細說明請參考 5.1.3 節)

■ 參考範例

此範例透過外部訊號控制，來啟動運動速度的比例值。



軸上運動時序

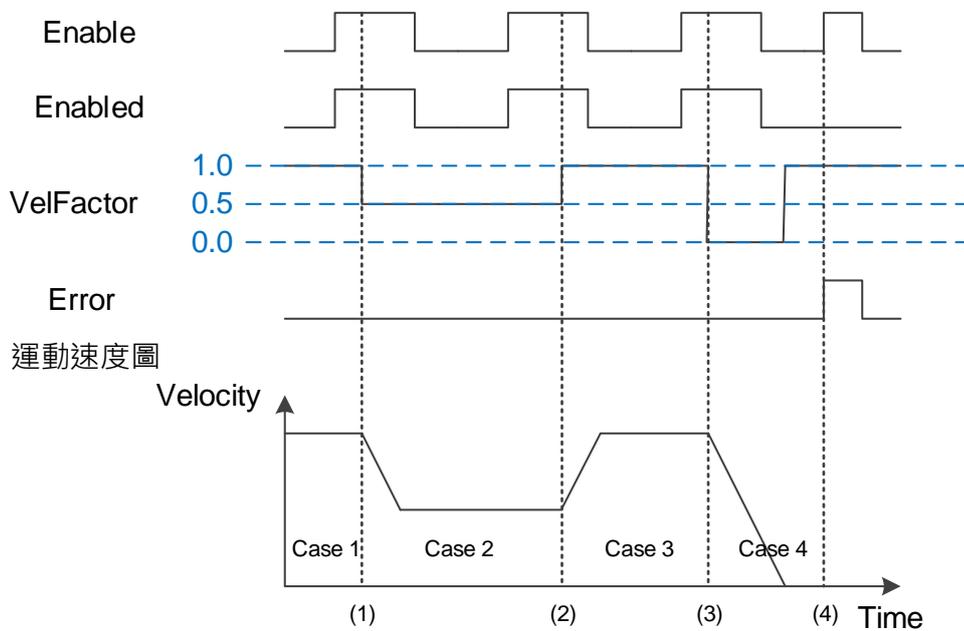


圖 5.2.2.36 MC\_SetOverride 範例說明

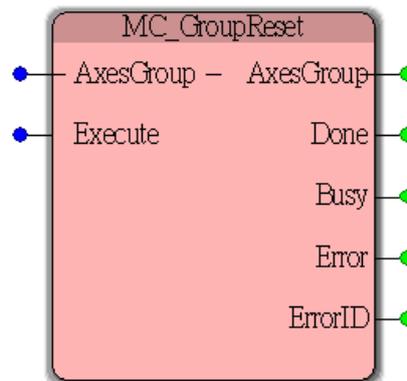
1. InputVar01 連接外部命令，如數位輸入變數或記憶體位置。
2. VelFactor 連接外部命令，如記憶體位置。
3. VelFactor 變動影響軸運動速度
  - Case1 : VelFactor = 1，軸速度維持 100%
  - Case2 : VelFactor = 0.5，軸速度降速為 50%
  - Case3 : VelFactor = 1，軸速度升速至 100%
  - Case4 : VelFactor = 0，軸速度降速至 0%
4. 軸速度因為錯誤產生，導致不會變更至 100%而保持在 0%

# 5

## 5.2.3 群組功能塊

### 5.2.3.1 MC\_GroupReset

- 類型  
Function Block
- 功能描述  
清除或重置所有和群組以及群組成員軸的錯誤。
- 功能塊



■ 輸入輸出共用變數

名稱	資料型態	設定值 (預設值)	功能
AxisGroup	GROUP_REF	1 (-)	指定群組編號

■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時，執行此指令

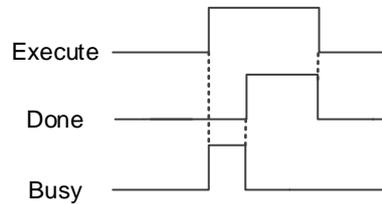
■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	異常清除完成時為True
Busy	BOOL	True / False (False)	指令被觸發執行時為True
Error	BOOL	True / False (False)	指令錯誤發生時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	指令錯誤發生時所記錄的錯誤碼；錯誤碼詳細說明請參考第5.1.4章節

■ 輸出變數的變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Done	當異警清除完成時	<ul style="list-style-type: none"> <li>■ 當Execute轉False時</li> <li>■ 若Execute為False 而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Busy	當 Execute 上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在ErrorID)	在Execute下緣時 (清除ErrorID紀錄之錯誤碼)

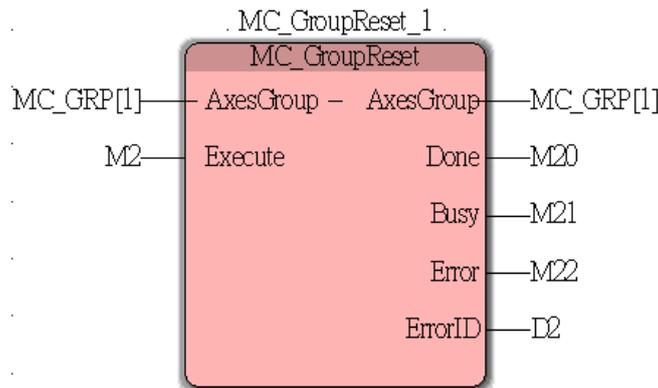
■ 輸出變數的變化時序



(時序詳細說明請參考 5.1.3 節)

■ 參考範例

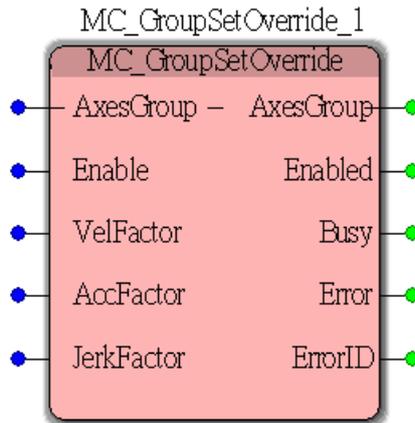
此範例說明 MC\_GroupReset 使用方式，M2 為被啟動後，功能塊會進行清除編號一號群組及群組中成員軸的異警。



# 5

## 5.2.3.2 MC\_GroupSetOverride

- 類型  
Function Block
- 功能描述  
透過超馳控制(Override)係數改變軸群組的速度。
- 功能塊



■ 輸入輸出共用變數

名稱	資料型態	設定值 (預設值)	功能
AxisGroup	GROUP_REF	1 (-)	指定群組編號

■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Enable	BOOL	True / False (False)	Enable上緣時，啟用此指令。當Enable為True時，超馳係數可以持續被更新。
VelFactor	LREAL	正數或 0 (0)	超馳控制係數，用來控制速度倍率。
AccFactor	-	-	保留
JerkFactor	-	-	保留

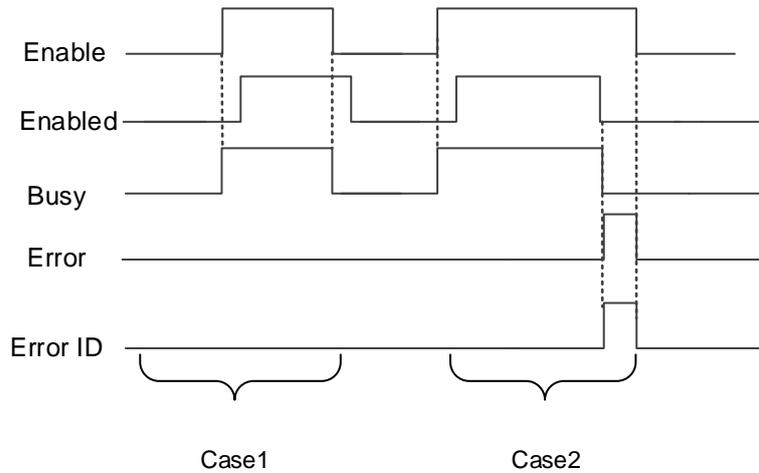
■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
Enabled	BOOL	True / False (False)	群組軸被控制時為True
Busy	BOOL	True / False (False)	指令被啟用執行時為True
Error	BOOL	True / False (False)	指令錯誤發生時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	指令錯誤發生時所記錄的錯誤碼；錯誤碼詳細說明請參考第5.1.4章節

■ 輸出變數的變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Enabled	當群組軸被控制時	<ul style="list-style-type: none"> <li>■ 在Enable下緣一個週期後</li> <li>■ 若Error上緣時</li> </ul>
Busy	當Enable上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Enable下緣時</li> <li>■ 在Error上緣時</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在 ErrorID)	在Execute下緣時 (清除ErrorID紀錄之錯誤碼)

■ 輸出變數的變化時序



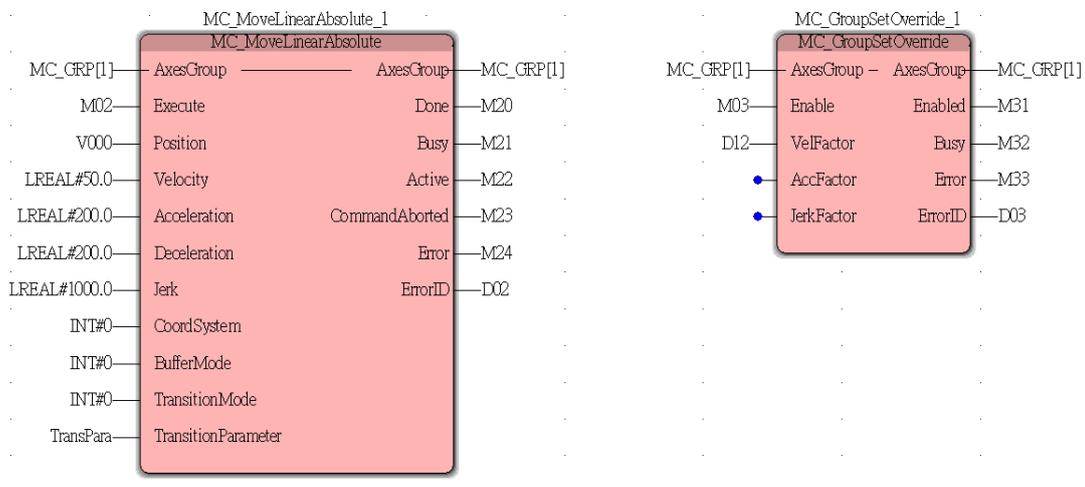
(時序詳細說明請參考 5.1.3 節)

Case 1：功能塊正常執行。

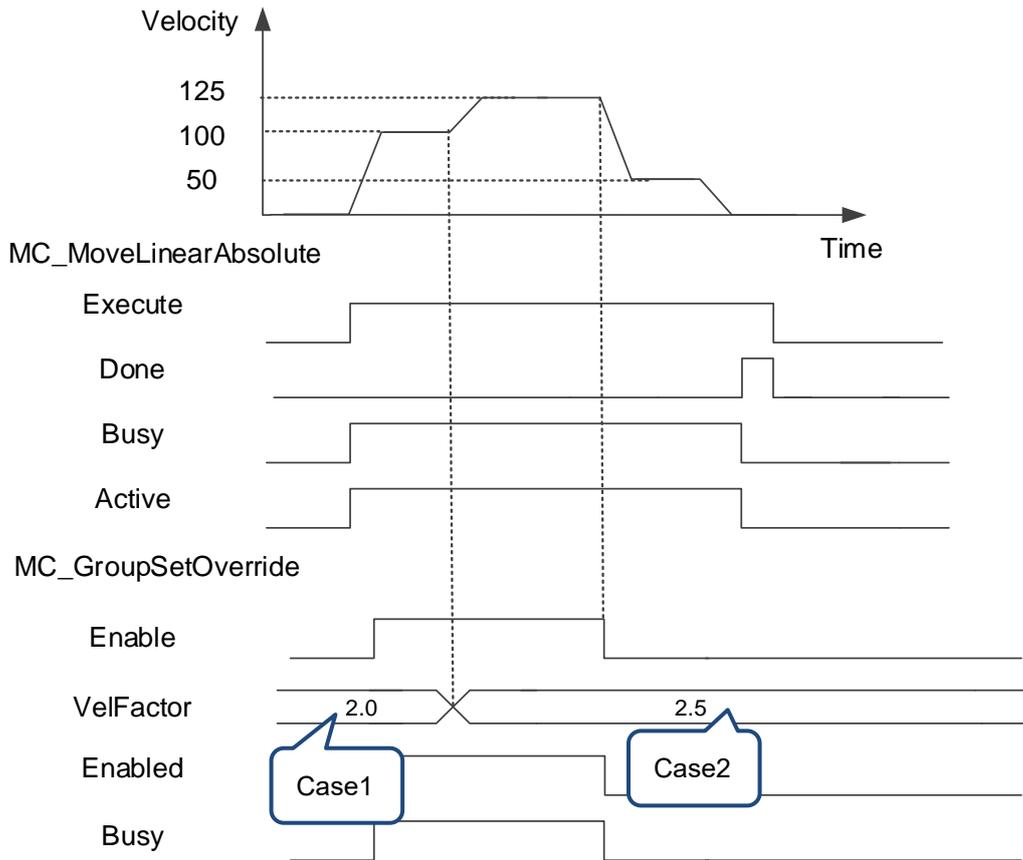
Case 2：功能塊執行中有錯誤產生。

■ 參考範例

此範例說明 MC\_GroupSetOverride 的使用方式以及執行時的運動軌跡。



# 5



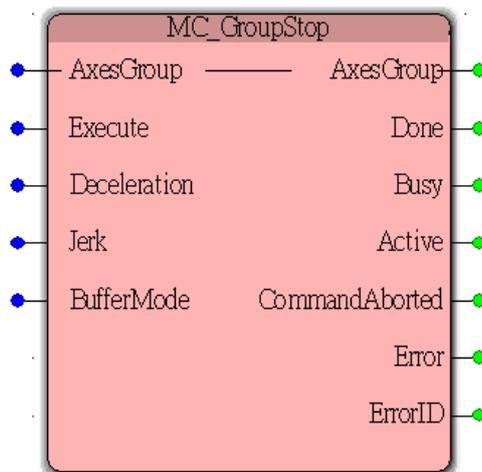
圖

### 5.2.3.2.1 MC\_GroupSetOverride 範例說明

1. M02 轉為 True 時啟動 MC\_MoveLinearAbsolute，群組軸以 50 的速度往絕對位置目標移動。
2. M03 轉為 True 時啟動 MC\_GroupSetOverride，
  - Case 1：D12 數值為 2，MC\_GroupSetOverride 功能塊將目標運動速度提高為兩倍，目標運動速度將從 50 升速到 100。
  - Case 2：在 MC\_GroupSetOverride 功能塊的 Enable 變數為 True 時，速度係數 (VelFator) 的數值修改都會持續影響目標運動，D12 由 2 改為 2.5，目標運動速度將再升速至 125。
3. 當 M03 轉為 False 時，群組的目標速度 Override 係數回復到 1，目標速度會減速至原本 MC\_MoveLinearAbsolute 指定的 50。

### 5.2.3.3 MC\_GroupStop

- 類型  
Function Block
- 功能描述  
使所有群組成員軸減速至停止。
- 功能塊



■ 輸入輸出共用變數

名稱	資料型態	設定值 (預設值)	功能
AxisGroup	GROUP_REF	1 (-)	指定軸群組編號

■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時 · 執行此指令
Deceleration	LREAL	正數或0 (0)	減速度( $\mu\text{m} / \text{s}^2$ )
Jerk	LREAL	正數或0 (0)	加加速度( $\mu\text{m} / \text{s}^2$ )
BufferMode	MC_BUFFER_MODE	0: mcAborting 1: mcBuffered	指定此功能塊的緩衝行為

## 5

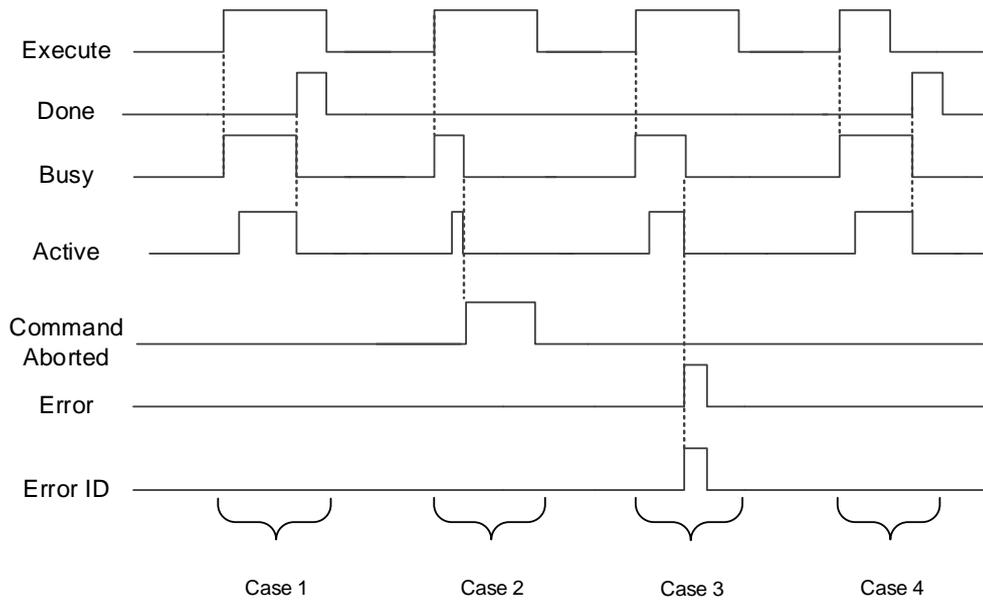
## ■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	當群組成員軸停止運動，且群組軸狀態進入Stopping時為True
Busy	BOOL	True / False (False)	指令被觸發執行時為True
Active	BOOL	True / False (False)	群組軸受控制時為True
CommandAborted	BOOL	True / False (False)	指令被中斷時為True
Error	BOOL	True / False (False)	指令錯誤發生時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	指令錯誤發生時所記錄的錯誤碼；錯誤碼詳細說明請參考第5.1.4章節

## ■ 輸出變數的變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Done	群組軸減速至0時	<ul style="list-style-type: none"> <li>■ 當Execute由True轉為False時</li> <li>■ 若Execute為False，而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Busy	Execute上緣觸發時	<ul style="list-style-type: none"> <li>■ 在 Done 上緣時</li> <li>■ 在 Error 上緣時</li> <li>■ 在 CommandAborted 上緣時</li> </ul>
Active	軸開始減速時	<ul style="list-style-type: none"> <li>■ 在Execute下緣且Done為True</li> <li>■ 在Done上緣且Execute為False</li> <li>■ 在 Error 上緣時</li> <li>■ 在 CommandAborted 上緣時</li> </ul>
CommandAborted	<ul style="list-style-type: none"> <li>■ 當此功能塊指令被其他緩衝模式設為 mcAborting 的指令中斷時</li> <li>■ 當此功能塊指令被 MC_Stop中斷時</li> </ul>	<ul style="list-style-type: none"> <li>■ 在 Execute下緣時</li> <li>■ 當 Execute為False而 CommandAborted轉為True，此時CommandAborted 維持一個掃描週期的True 狀態後，立刻轉為False</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在ErrorID)	在Execute下緣時 (清除ErrorID紀錄之錯誤碼)

■ 輸出變數的變化時序



(時序詳細說明請參考 5.1.3 節)

Case 1：運動功能塊正常執行。

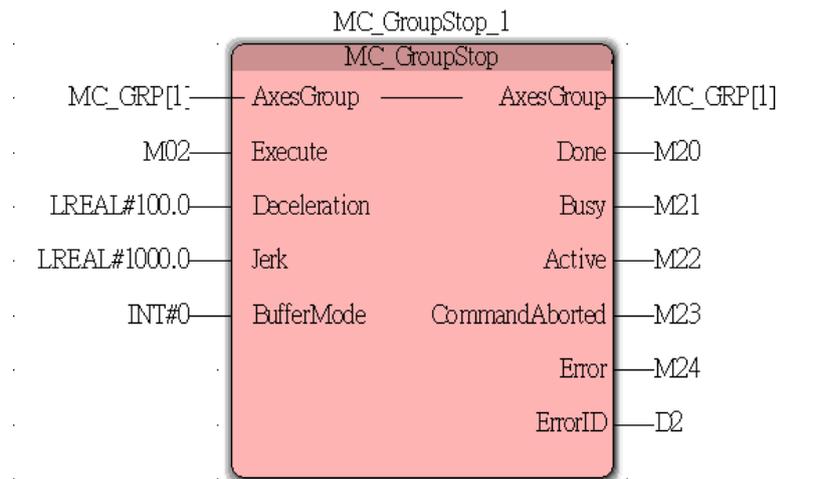
Case 2：運動功能塊執行過程中被其他運動功能塊插斷。

Case 3：運動功能塊執行過程中，有錯誤產生。

Case 4：有 Buffer 輸入的運動功能塊正常執行。(即使 Execute 已經是 False，功能塊仍要執行動作完成，完成時 Done 訊號輸出至少要維持一個 PLC 掃描週期。)

■ 參考範例

此範例說明 MC\_GroupStop 的運行方式，及執行時的運動軌跡。



# 5

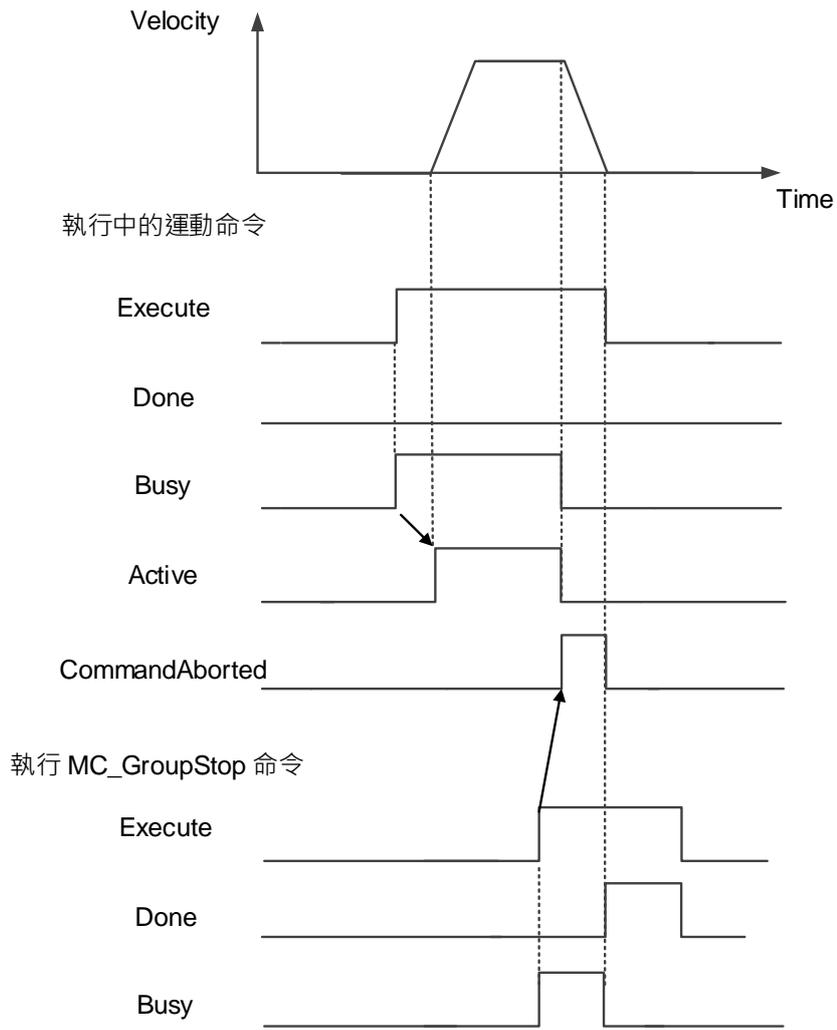
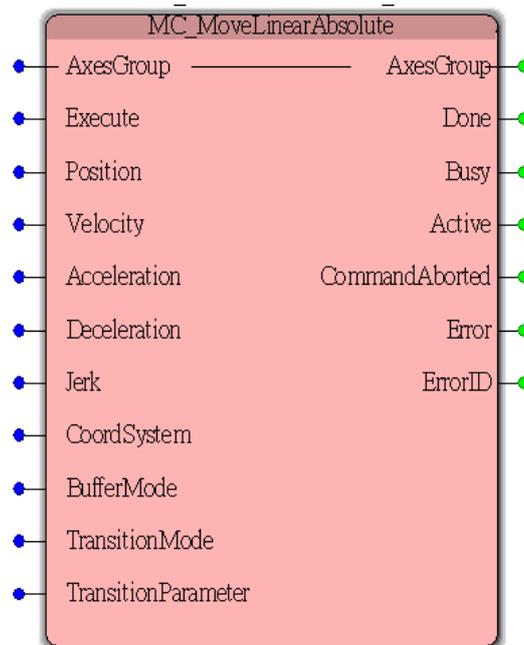


圖 5.2.3.3.1 MC\_GroupSetOverride 範例說明

### 5.2.3.4 MC\_MoveLinearAbsolute

- 類型  
Function Block
- 功能描述  
群組軸依照設定運動參數，執行直線運動至指定的絕對位置。
- 功能塊



■ 輸入輸出共用變數

名稱	資料型態	設定值 (預設值)	功能
AxisGroup	GROUP_REF	1 (-)	指定軸群組編號

■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時，執行此指令
Position	S_WC_POS	整數 (0)	絕對位置(um)。型態為一陣列，設定目標空間座標 X、Y、Z、A、B、C。
Velocity	LREAL	正整數或0 (0)	目標速度(um / s)
Acceleration	LREAL	正整數或 (0)	加速度(um / s <sup>2</sup> )
Deceleration	LREAL	正整數或0 (0)	減速度(um / s <sup>2</sup> )
Jerk	LREAL	正整數或0 (0)	加加速度(um / s <sup>3</sup> )
CoordSystem	INT	0: MCS 1: PCS 2: TCS 3: ACS	指定坐標系 MCS：大地坐標系 PCS：使用者座標系 TCS：工具座標系 ACS：關節座標系

# 5

名稱	資料型態	設定值 (預設值)	功能
BufferMode	MC_BUFFER_MODE	0: mcAborting 1: mcBuffered	指定此功能塊的緩衝行為
TransitionMode	INT	0: TMNone 3: TMCornerDistance A: TMDecSegment	指定重疊路徑模式
TransitionParameter	S_MC_TRANSITION_PARAMETER_REF	正整數或0 (0)	重疊路徑參數重疊路徑參數 (um或%)

■ 輸出變數

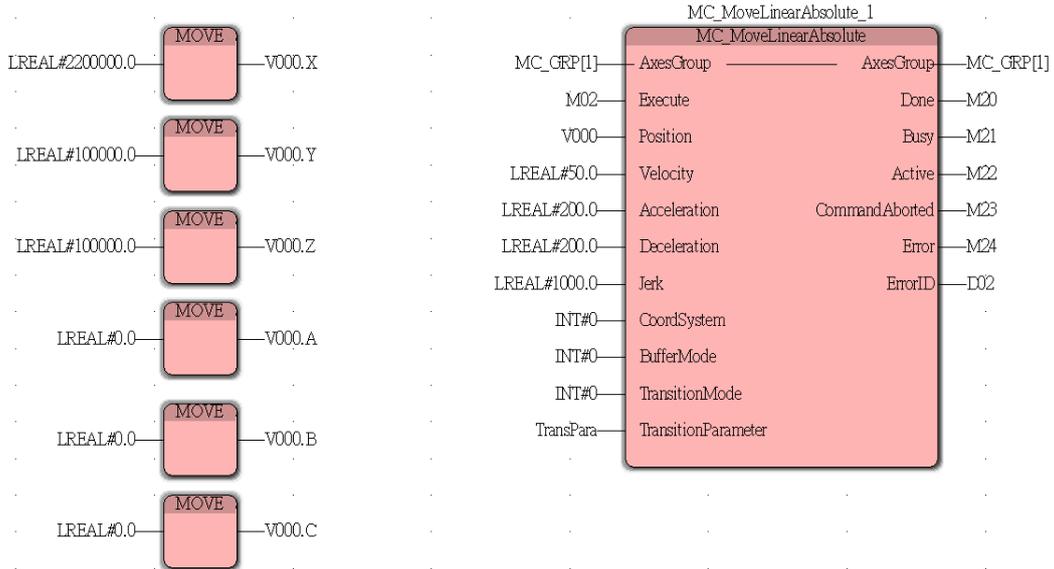
名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	運動指令完成時為True
Busy	BOOL	True / False (False)	指令被觸發執行時為True
Active	BOOL	True / False (False)	指定軸受控制時為True
CommandAborted	BOOL	True / False (False)	指令被中斷時為True
Error	BOOL	True / False (False)	指令錯誤發生時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	指令錯誤發生時所記錄的錯誤碼；錯誤碼詳細說明請參考第5.1.4章節

■ 輸出變數的變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Done	當運動指令到位完成時	<ul style="list-style-type: none"> <li>■ 當Execute轉為False時</li> <li>■ 若Execute為False·而Done轉為True·此時Done維持一個掃描週期的True狀態後·立刻轉為False</li> </ul>
Busy	當Execute上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
Active	當軸運動開始時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted 上緣時</li> </ul>
CommandAborted	<ul style="list-style-type: none"> <li>■ 當此功能塊指令被其他緩衝模式設為mcAborting的指令中斷時</li> <li>■ 當此功能塊指令被MC_Stop中斷時</li> </ul>	<ul style="list-style-type: none"> <li>■ 在Execute下緣時</li> <li>■ 在Execute為False而CommandAborted 轉為True·此時CommandAborted 維持一個掃描週期的True狀態後·立刻轉為False</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在 ErrorID)	在Execute下緣時 (清除ErrorID紀錄之錯誤碼)

- 輸出變數的變化時序  
請參考圖 5.2.2.4.1 MC\_MoveAbsolute 時序圖

- 參考範例  
此範例說明 MC\_MoveLinearAbsolute 的使用方式，及執行時輸出變數狀態。

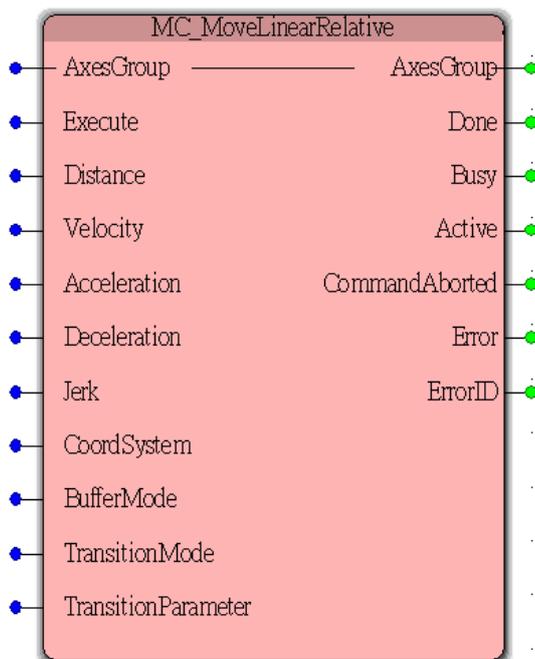


1. 位置(Position)型態 S\_WC\_POS 是一個陣列，必須先將欲移動的目標絕對位置以及姿態填入此陣列中，範例將空間中絕對位置點位設在(220000, 10000, -10000, 0, 0, 0)。
2. 當 M2 轉為 True 時，MC\_MoveLinearAbsolute 即控制群組軸往設定的絕對位置做直線移動。
3. 當群組軸已抵達指定的絕對位置後，若重複執行此運動指令，群組軸將不做任何移動。

# 5

## 5.2.3.5 MC\_MoveLinearRelative

- 類型  
Function Block
- 功能描述  
群組軸依照設定運動參數，執行直線運動至指定的相對位置。
- 功能塊



■ 輸入輸出共用變數

名稱	資料型態	設定值 (預設值)	功能
AxisGroup	GROUP_REF	1 (-)	指定軸群組編號

■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時，執行此指令
Distance	S_WC_POS	整數 (0)	相對位置，距離基準點當前群組軸所在位置。
Velocity	LREAL	正整數或0 (0)	目標速度(um / s)
Acceleration	LREAL	正整數或0 (0)	加速度(um / s <sup>2</sup> )
Deceleration	LREAL	正整數或0 (0)	減速度(um / s <sup>2</sup> )
Jerk	LREAL	正整數或0 (0)	加加速度(um / s <sup>3</sup> )
CoordSystem	INT	1: MCS 2: PCS 3: TCS 4: ACS	指定坐標系 MCS：大地坐標系 PCS：使用者座標系 TCS：工具座標系 ACS：關節座標系

名稱	資料型態	設定值 (預設值)	功能
BufferMode	MC_BUFFER_MODE	1: mcAborting 2: mcBuffered	指定此功能塊的緩衝行為
TransitionMode	INT	1: TMNone 3: TMCornerDistanceATMDecSegment	指定重疊路徑模式
TransitionParameter	S_MC_TRANSITION_PARAMETER_REF	正整數或0 (0)	重疊路徑參數重疊路徑參數 (um或%)

■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	運動命令完成時為True
Busy	BOOL	True / False (False)	指令被觸發執行時為True
Active	BOOL	True / False (False)	指定軸受控制時為True
CommandAborted	BOOL	True / False (False)	指令被中斷時為True
Error	BOOL	True / False (False)	指令錯誤發生時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	指令錯誤發生時所記錄的錯誤碼；錯誤碼詳細說明請參考第5.1.4章節

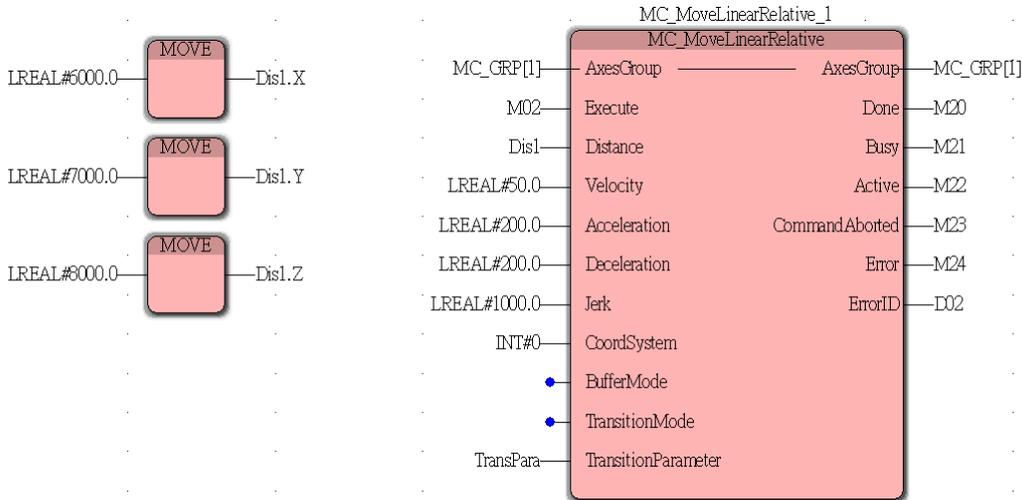
■ 輸出變數的變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Done	當絕對定位完成時	<ul style="list-style-type: none"> <li>■ 當Execute由True轉為False時</li> <li>■ 若Execute為False而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Busy	當Execute上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
Active	當軸運動開始時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
CommandAborted	<ul style="list-style-type: none"> <li>■ 當此功能塊指令被其他緩衝模式設為mcAborting的指令中斷時</li> <li>■ 當此功能塊指令被MC_Stop中斷時</li> </ul>	<ul style="list-style-type: none"> <li>■ 在Execute下緣時</li> <li>■ 當Execute為False而CommandAborted轉為True，此時CommandAborted維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在ErrorID)	在Execute下緣時 (清除ErrorID紀錄之錯誤碼)

# 5

- 輸出變數的變化時序  
請參考圖 5.2.2.4.1 MC\_MoveAbsolute 時序圖

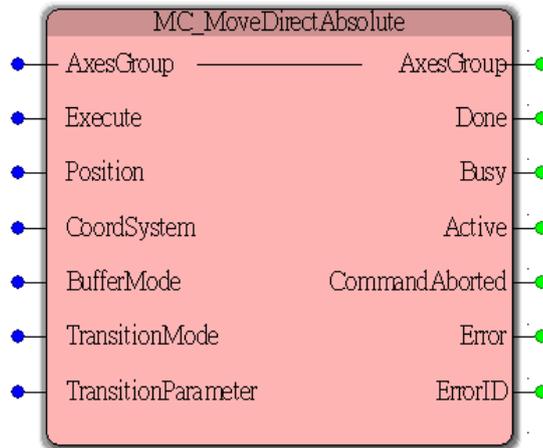
- 參考範例  
此範例說明 MC\_MoveLinearRelative 的運行方式，及執行時的運動描述。



1. 位置(Position)型態 S\_WC\_POS 是一個陣列，必須先將欲移動的目標絕對位置以及姿態填入此陣列中，範例將空間中相對位置點位設在(6000.0, 7000.0, -8000.0, 0, 0, 0)。
2. 當 M2 轉為 True 時，MC\_MoveLinearRelative 即控制群組軸往設定的相對位置做直線移動。
3. 當群組軸移動了相對距離(6000, 7000, -8000)後，再次重複觸發 M2，運動指令將會再次執行一次相對距離移動。

### 5.2.3.6 MC\_MoveDirectAbsolute

- 類型  
Function Block
- 功能描述  
控制群組軸執行無規畫路徑運動至指定的絕對位置。
- 功能塊



■ 輸入輸出共用變數

名稱	資料型態	設定值 (預設值)	功能
AxisGroup	GROUP_REF	1 (-)	指定軸群組編號

■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時，執行此指令
Position	S_WC_POS	整數 (0)	絕對位置(um)。型態為一陣列，設定目標空間座標 X、Y、Z、A、B、C。
CoordSystem	INT	1: MCS 2: PCS 3: TCS 4: ACS	指定坐標系 MCS：大地坐標系 PCS：使用者座標系 TCS：工具座標系 ACS：關節座標系
BufferMode	MC_BUFFER_MODE	0: mcAborting 1: mcBufered	指定此功能塊的緩衝行為
TransitionMode	INT	0: TMNone 3: TMCornerDistance A: TMDecSegment	指定重疊路徑模式
TransitionParameter	S_MC_TRANSITION_PARAMETER_REF	正整數或 0(0)	重疊路徑參數(um或%)

## 5

## ■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	運動指令完成時為True
Busy	BOOL	True / False (False)	指令被觸發執行時為True
Active	BOOL	True / False (False)	指定軸受控制時為True
CommandAborted	BOOL	True / False (False)	指令被中斷時為True
Error	BOOL	True / False (False)	指令錯誤發生時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	指令錯誤發生時所記錄的錯誤碼；錯誤碼詳細說明請參考第5.1.4章節

## ■ 輸出變數的變化時序

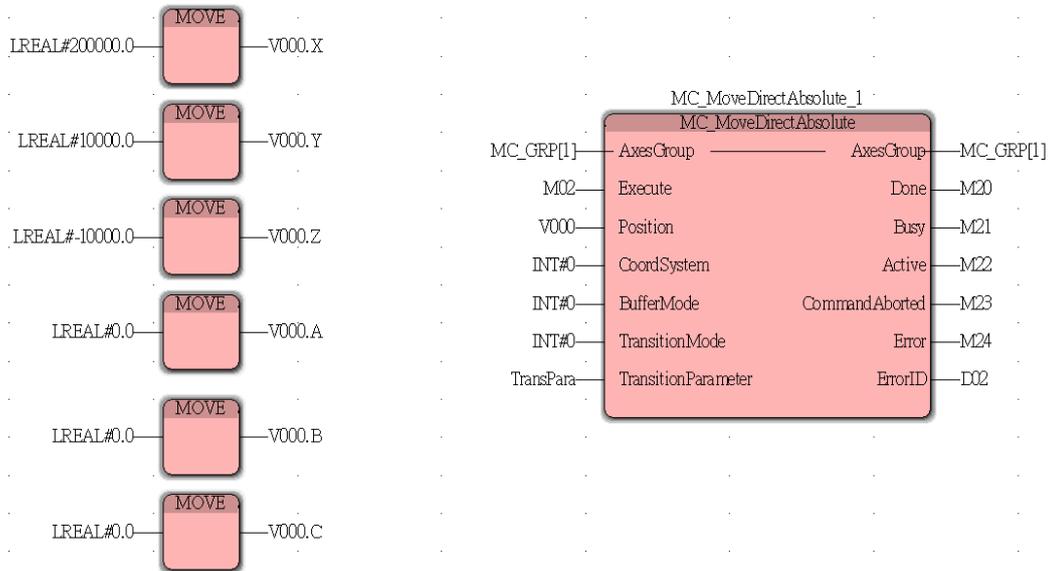
名稱	輸出變數上緣時機	輸出變數下緣時機
Done	運動指令到位完成時	<ul style="list-style-type: none"> <li>■ 當Execute轉為False時</li> <li>■ 若Execute為False而Done 轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Busy	Execute上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
Active	軸運動開始時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
CommandAborted	<ul style="list-style-type: none"> <li>■ 當此功能塊指令被其他緩衝模式設為mcAborting的指令中斷時</li> <li>■ 當此功能塊指令被MC_Stop中斷時</li> </ul>	<ul style="list-style-type: none"> <li>■ 在Execute下緣時</li> <li>■ 當Execute為False而CommandAborted轉為True，此時CommandAborted維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在ErrorID)	在Execute下緣時 (清除ErrorID紀錄之錯誤碼)

## ■ 輸出變數的變化時序

請參考圖 5.2.2.4.1 MC\_MoveAbsolute 時序圖

■ 參考範例

此範例說明 MC\_MoveDirectAbsolute 的運行方式，及執行時的運動描述。

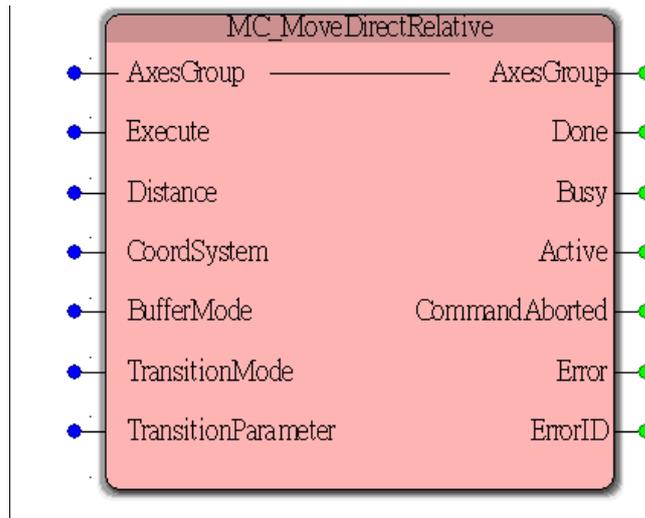


1. 位置(Position)型態 S\_WC\_POS 是一個陣列，必須先將欲移動的目標絕對位置以及姿態填入此陣列中，範例將空間中絕對位置點位設在(200000, 10000, -10000, 0, 0, 0)。
2. 當 M2 轉為 True 時，MC\_MoveDirectAbsolute 即控制群組軸往設定的絕對位置做直線移動。
3. 當群組軸已抵達指定的絕對位置後，若重複執行此運動指令，群組軸將不做任何移動。

# 5

## 5.2.3.7 MC\_MoveDirectRelative

- 類型  
Function Block
- 功能描述  
控制群組軸執行無規畫路徑運動至指定的相對位置。
- 功能塊



■ 輸入輸出共用變數

名稱	資料型態	設定值 (預設值)	功能
AxisGroup	GROUP_REF	1 (-)	指定軸群組

■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時，執行此指令
Distance	S_WC_POS	整數 (0)	相對位置，距離基準點當前群組軸所在位置。
CoordSystem	INT	1: MCS 2: PCS 3: TCS 4: ACS	指定坐標系 MCS：大地坐標系 PCS：使用者座標系 TCS：工具座標系 ACS：關節座標系
BufferMode	MC_BUFFER_MODE	0: mcAborting 1: mcBuffered	指定此功能塊的緩衝行為
TransitionMode	INT	0: TMNone 3: TMCornerDistance A: TMDecSegment	指定重疊路徑模式
TransitionParameter	S_MC_TRANSITION_PARAMETER_REF	正整數或0 (0)	重疊路徑參數(um或%)

■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	絕對定位完成時為True
Busy	BOOL	True / False (False)	指令被觸發執行時為True
Active	BOOL	True / False (False)	指定軸受控制時為True
CommandAborted	BOOL	True / False (False)	指令被中斷時為True
Error	BOOL	True / False (False)	指令錯誤發生時為True
ErrorID	DWORD	0x0000 ~ 0xFFFF (0)	指令錯誤發生時所記錄的錯誤碼；錯誤碼詳細說明請參考第5.1.4章節

■ 輸出變數的變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Done	絕對定位完成時	<ul style="list-style-type: none"> <li>■ 當Execute轉為False時</li> <li>■ 若Execute為False，而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Busy	Execute上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
Active	軸運動開始時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
CommandAborted	<ul style="list-style-type: none"> <li>■ 當此功能塊指令被其他緩衝模式設為 mcAborting 的指令中斷時</li> <li>■ 當此功能塊指令被 MC_Stop中斷時</li> </ul>	<ul style="list-style-type: none"> <li>■ 在Execute下緣時</li> <li>■ 當Execute為False而CommandAborted轉為True，此時CommandAborted維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在ErrorID)	在Execute下緣時 (清除ErrorID紀錄之錯誤碼)

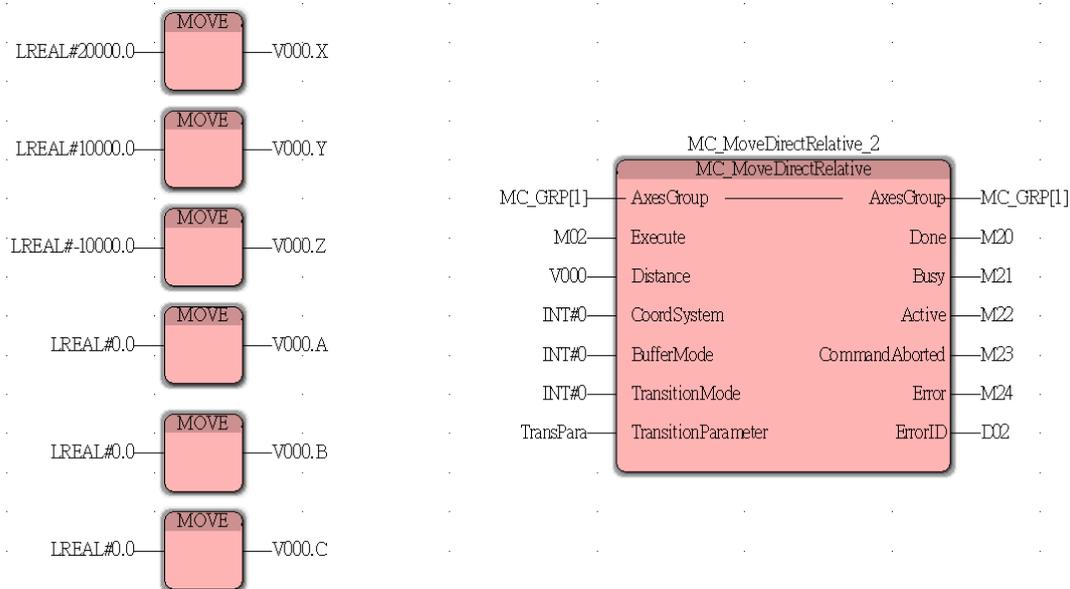
■ 輸出變數的變化時序

請參考圖 5.2.2.4.1 MC\_MoveAbsolute 時序圖

# 5

■ 參考範例

此範例說明 MC\_MoveDirectRelative 的運行方式，及執行時的運動描述。

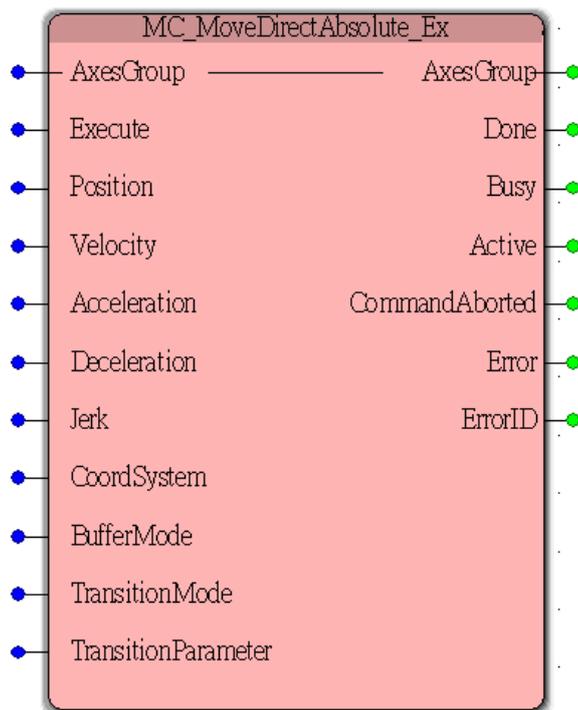


1. Position 型態為 S\_WC\_POS，為一陣列。必須設定將空間座標值及姿態填入陣列相對位置，範例設定點為空間點位(20000.0, 10000.0, -10000.0, 0, 0, 0)。
2. 當 M2 轉為 True，MC\_MoveLinearRelative 開始控制群組軸往設定的相對位置移動
3. 當群組軸移動了相對距離(20000,10000,-10000)後，再次重複觸發 M2，運動指令會再次執行一次相對距離。

## 5.2.4 擴充功能塊

### 5.2.4.1 MC\_MoveDirectAbsolute\_Ex

- 類型  
Function Block
- 功能描述  
控制指定群組軸按照所設定的速度參數，執行無規畫路徑運動至指定的絕對位置。
- 功能塊



■ 輸入輸出共用變數

名稱	資料型態	設定值 (預設值)	功能
AxisGroup	GROUP_REF	1 (-)	指定軸群組

■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時，執行此指令
Position	S_WC_POS	整數 (0)	絕對位置(um)。型態為一陣列，設定目標空間座標 X、Y、Z、A、B、C。
Velocity	LREAL	正整數或0 (0)	目標速度(um / s)
Acceleration	LREAL	正整數或0 (0)	加速度(um / s <sup>2</sup> )
Deceleration	LREAL	正整數或0 (0)	減速度(um / s <sup>2</sup> )

# 5

名稱	資料型態	設定值 (預設值)	功能
Jerk	LREAL	正整數或0 (0)	加加速度(um / s <sup>3</sup> )
CoordSystem	INT	1: MCS 2: PCS 3: TCS 4: ACS	指定坐標系 MCS：大地坐標系 PCS：使用者座標系 TCS：工具座標系 ACS：關節座標系
BufferMode	MC_BUFFER_MODE	0: mcAborting 1: mcBufered	指定此功能塊的緩衝行為
TransitionMode	INT	0: TMNone 3: TMCornerDistance A: TMDecSegment	指定重疊路徑模式
TransitionParameter	S_MC_TRANSITION_PARAMETER_REF	正整數或0 (0)	重疊路徑參數(um或%)

■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	運動指令完成時為True
Busy	BOOL	True / False (False)	指令被觸發執行時為True
Active	BOOL	True / False (False)	指定軸受控制時為True
CommandAborted	BOOL	True / False (False)	指令被中斷時為True
Error	BOOL	True / False (False)	指令錯誤發生時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	指令錯誤發生時所記錄的錯誤碼； 錯誤碼詳細說明請參考第5.1.4章節

■ 輸出變數的變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Done	絕對定位完成時	<ul style="list-style-type: none"> <li>■ 當Execute轉為False時</li> <li>■ 若Execute為False，而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Busy	Execute上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
Active	軸運動開始時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
CommandAborted	<ul style="list-style-type: none"> <li>■ 當此功能塊指令被其他緩衝模式設為mcAborting的指令中</li> </ul>	<ul style="list-style-type: none"> <li>■ 在Execute下緣時</li> <li>■ 當Execute為False而CommandAborted轉為True，此時</li> </ul>

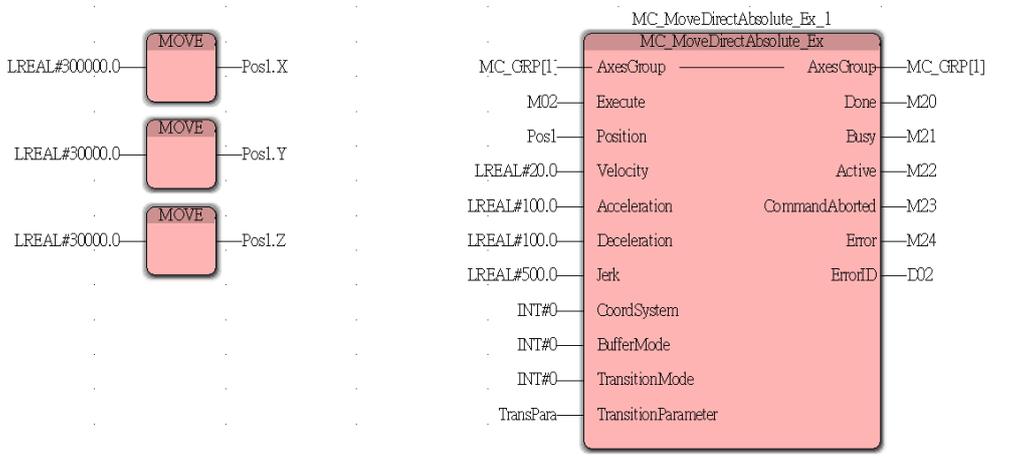
名稱	輸出變數上緣時機	輸出變數下緣時機
	斷時 ■ 當此功能塊指令被 MC_Stop 中斷時	CommandAborted 維持一個掃描週期的 True 狀態後，立刻轉為 False
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在 ErrorID)	在 Execute 下緣時 (清除 ErrorID 紀錄之錯誤碼)

■ 輸出變數的變化時序

請參考圖 5.2.2.4.1 MC\_MoveAbsolute 時序圖

■ 參考範例

此範例說明 MC\_MoveDirectAbsolute\_Ex 的運行方式，及執行時的運動描述。

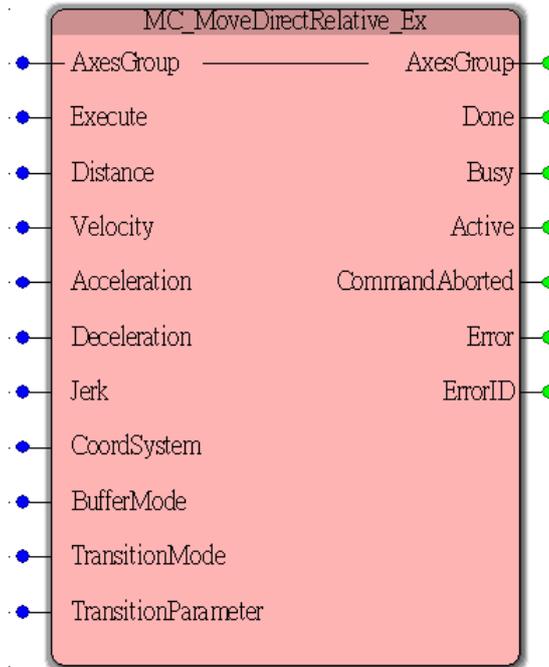


1. Position 型態為 S\_WC\_POS，為一陣列。必須設定將空間座標值及姿態填入陣列相對位置，範例設定點為空間點位(300000, -30000, -30000, 0, 0, 0)。
2. 當 M2 轉為 True，MC\_MoveDirectAbsolute 開始控制群組軸往設定的絕對位置移動。
3. 當軸抵達指定的絕對位置，重複出發此運動指令不會使運動軸移動。

# 5

## 5.2.4.2 MC\_MoveDirectRelative\_Ex

- 類型  
Function Block
- 功能描述  
控制指定群組軸按照設定速度參數，執行無規畫路徑運動至指定的相對位置。
- 功能塊



■ 輸入輸出共用變數

名稱	資料型態	設定值 (預設值)	功能
AxisGroup	GROUP_REF	1 (-)	指定軸群組

■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時，執行此指令
Distance	S_WC_POS	整數 (0)	相對位置，距離基準點當前群組軸所在位置。
Velocity	LREAL	正整數或0 (0)	目標速度(um / s)
Acceleration	LREAL	正整數或0 (0)	加速度(um / s <sup>2</sup> )
Deceleration	LREAL	正整數或0 (0)	減速度(um / s <sup>2</sup> )
Jerk	LREAL	正整數或0 (0)	加加速度(um / s <sup>3</sup> )
CoordSystem	INT	1: MCS 2: PCS 3: TCS 4: ACS	指定坐標系 MCS：大地坐標系 PCS：使用者座標系 TCS：工具座標系

名稱	資料型態	設定值 (預設值)	功能
			ACS：關節座標系
BufferMode	MC_BUFFER_MODE	0: mcAborting 1: mcBuffered	指定此功能塊的緩衝行為
TransitionMode	INT	0: TMNone 3: TMCornerDistance A: TMDecSegment	指定重疊路徑模式
TransitionParameter	S_MC_TRANSITION_PARAMETER_REF	正整數或0 (0)	重疊路徑參數 (um或%)

■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	運動指令完成時為True
Busy	BOOL	True / False (False)	指令被觸發執行時為True
Active	BOOL	True / False (False)	指定軸受控制時為True
CommandAborted	BOOL	True / False (False)	指令被中斷時為True
Error	BOOL	True / False (False)	指令錯誤發生時為 True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	指令錯誤發生時所記錄的錯誤碼；錯誤碼詳細說明請參考第5.1.4章節

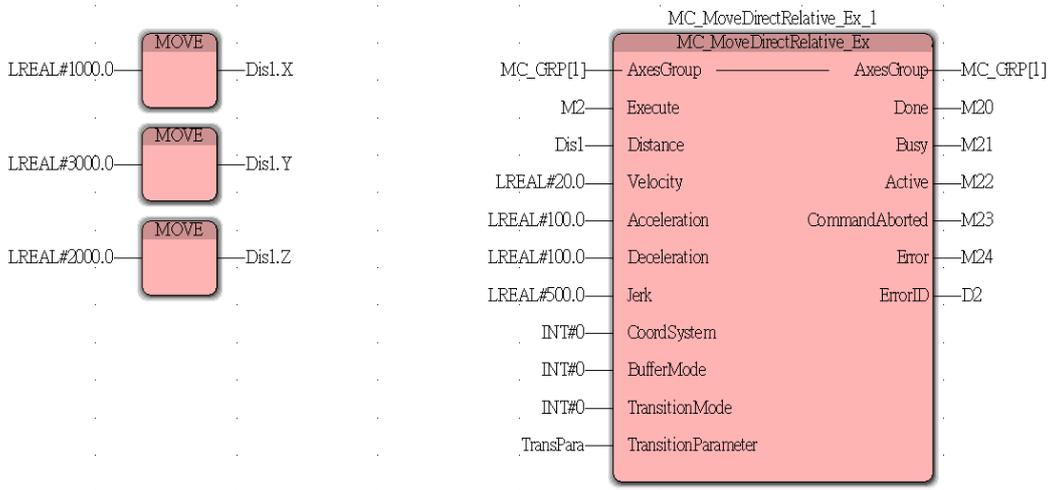
■ 輸出變數的變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Done	絕對定位完成時	<ul style="list-style-type: none"> <li>■ 當Execute轉為False時</li> <li>■ 若Execute為False而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Busy	Execute上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
Active	軸運動開始時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 當 CommandAborted上緣時</li> </ul>
CommandAborted	<ul style="list-style-type: none"> <li>■ 當此功能塊指令被其他緩衝模式設為 mcAborting 的指令中斷時</li> <li>■ 當此功能塊指令被 MC_Stop中斷時</li> </ul>	<ul style="list-style-type: none"> <li>■ 在Execute下緣時</li> <li>■ 當Execute為False而CommandAborted轉為True，此時CommandAborted維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在 ErrorID)	在Execute下緣時 (清除ErrorID紀錄之錯誤碼)

# 5

- 輸出變數的變化時序  
請參考圖 5.2.2.4.1 MC\_MoveAbsolute 時序圖

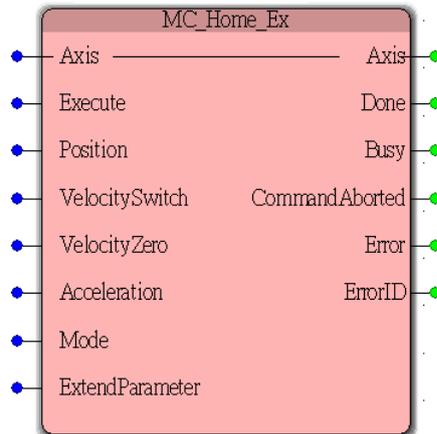
- 參考範例  
此範例說明 MC\_MoveDirectRelative\_Ex 的運行方式，及執行時的運動描述。



1. Position 型態為 S\_WC\_POS，為一陣列。必須設定將空間座標值及姿態填入陣列相對位置，範例設定點為空間點位(1000.0, 3000.0, -2000.0 0, 0, 0)。
2. 當 M2 轉為 True，MC\_MoveLinearRelative 開始控制群組軸往設定的相對位置移動 (+X 方向移動 1000 um+Y 方向移動 3000 um，-Z 方向移動 2000 um)。
3. 當群組軸移動了相對距離(1000、3000、-2000)後，再次重複觸發 M2，運動指令會再次執行一次相對距離。

### 5.2.4.3 MC\_Home\_Ex

- 類型  
Function Block
- 功能描述  
啟動指定軸按照設定模式及速度進行原點復歸。
- 功能塊



■ 輸入輸出共用變數

名稱	資料型態	設定值 (預設值)	功能
Axis	AXIS_REF	1 (-)	指定軸群組

■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時·執行此指令
Position	DINT	正整數·負整數或0 (0)	目標原點與當前原點的偏差值
VelocitySwitch <sup>*1</sup>	DINT	正整數或0 (0)	第一段高速原點復歸速度 (rpm)。找尋碰撞點·正負極限時的速度。*1
VelocityZero <sup>*1</sup>	DINT	正整數或0 (0)	第二段低速原點復歸速度 (rpm)。用於找尋 Z 脈波速度。*1
Acceleration	DINT	正整數或0 (0)	原點復歸過程加速度(使用者單位 / 秒 <sup>2</sup> )
Mode	INT	-4 ~ 35 (0)	原點復歸模式*2
ExtendParameter	S_MC_HOM E_EXTEND PARA	-	特殊原點復歸模式參數設定·目前未有模式使用。

註：

1. VelocitySwitch 表示在尋找第一個原點賦歸的速度·VelocityZero 則為第二段尋找 Z 脈波的速度設定·詳細說明請參考 ASDA-A2 手冊。
2. 原點復歸模式請參考 ASDA-A2 手冊。

## 5

## ■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	原點復歸完成時為True
Busy	BOOL	True / False (False)	指令被觸發執行時為True
CommandAborted	BOOL	True / False (False)	指令被中斷時為True
Error	BOOL	True / False (False)	指令錯誤發生時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	當指令錯誤發生時所記錄的錯誤碼；錯誤碼詳細說明請參考第5.1.4章節

## ■ 輸出變數的變化時序

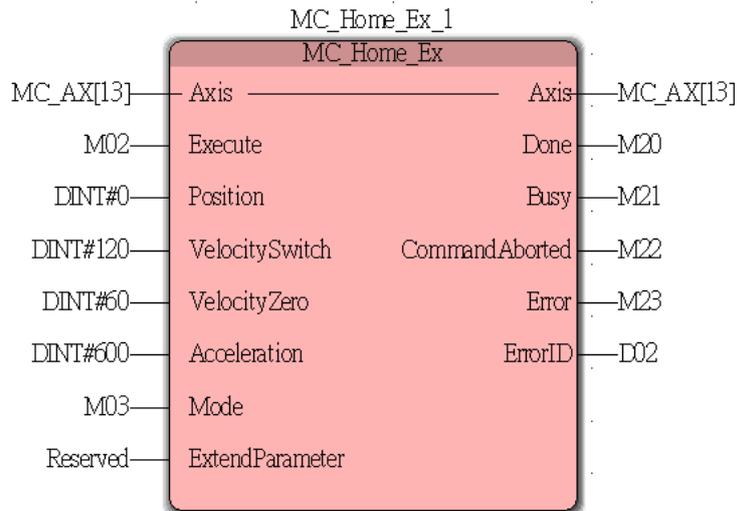
名稱	輸出變數上緣時機	輸出變數下緣時機
Done	絕對定位完成時	<ul style="list-style-type: none"> <li>■ 當Execute轉為False時</li> <li>■ 若Execute為False，而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Busy	當Execute上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
CommandAborted	<ul style="list-style-type: none"> <li>■ 當此功能塊指令被其他緩衝模式設為 mcAborting 的指令中斷時</li> <li>■ 當此功能塊指令被 MC_Stop 中斷時</li> </ul>	<ul style="list-style-type: none"> <li>■ 在Execute下緣時</li> <li>■ 當 Execute為False而 CommandAborted 轉為 True，此時CommandAborted 維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Error / ErrorID	<ul style="list-style-type: none"> <li>■ 當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在 ErrorID)</li> </ul>	在Execute下緣時 (清除ErrorID紀錄之錯誤碼)

## ■ 輸出變數的變化時序

請參考圖 5.2.2.4.1 MC\_MoveAbsolute 時序圖

■ 參考範例

此範例說明 MC\_Home\_Ex 的運行方式，及執行時的運動描述。



1. M02 轉為 True，MC\_Home\_Ex 控制指定軸按照指定的模式及速度進行原點復歸。
2. 再次將 M02 轉為 True，會再次進行原點復歸。

5.2.4.4 MC\_MoveLinearJog

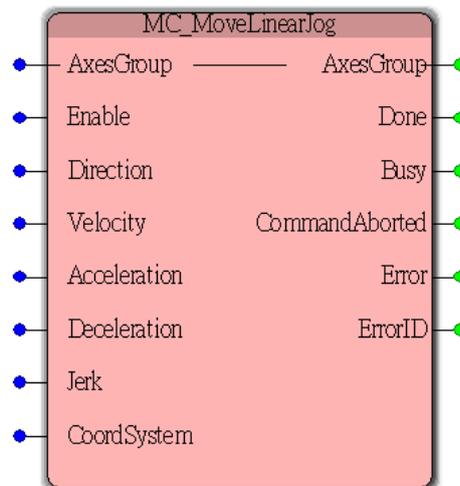
■ 類型

Function Block

■ 功能描述

啟動指定群組軸按照設定方向及速度參數進行直線吋動。

■ 功能塊



## 5

## ■ 輸入輸出共用變數

名稱	資料型態	設定值 (預設值)	功能
AxesGroup	GRP_REF	1 (-)	指定軸群組

## ■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Enble	BOOL	True / False (False)	Enable為True時，執行此指令。
Direction	S_WC_POS	整數 (0)	吋動方向，陣列數值只參考正負號，作為移動方向依據。
Velocity	LREAL	正整數或0 (0)	目標速度( $\mu\text{m} / \text{s}$ )
Acceleration	LREAL	正整數或0 (0)	加速度( $\mu\text{m} / \text{s}^2$ )
Deceleration	LREAL	正整數或0 (0)	減速度( $\mu\text{m} / \text{s}^2$ )
Jerk	LREAL	正整數或0 (0)	加加速度( $\mu\text{m} / \text{s}^3$ )
CoordSystem	INT	1: MCS 2: PCS 3: TCS 4: ACS	指定坐標系 MCS：大地坐標系 PCS：使用者座標系 TCS：工具座標系 ACS：關節座標系

## ■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	原點復歸完成時為True
Busy	BOOL	True / False (False)	指令被觸發執行時為True
CommandAborted	BOOL	True / False (False)	指令被中斷時為True
Error	BOOL	True / False (False)	指令錯誤發生時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	指令錯誤發生時所記錄的錯誤碼；錯誤碼詳細說明請參考第5.1.4章節

## ■ 輸出變數的變化時序

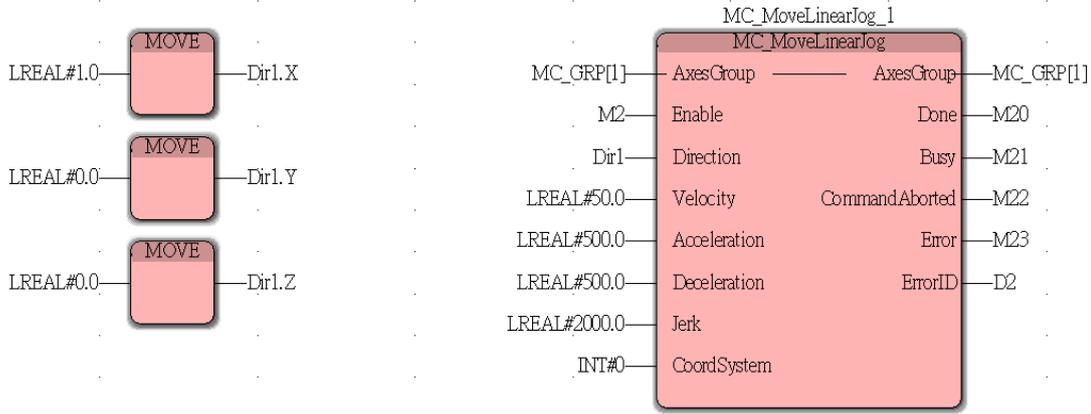
名稱	輸出變數上緣時機	輸出變數下緣時機
Done	當絕對定位完成時	<ul style="list-style-type: none"> <li>■ 當Execute轉為False時</li> <li>■ 若Execute為False，而Done 轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為 False</li> </ul>
Busy	當Execute上緣觸發時	<ul style="list-style-type: none"> <li>■ 當 Done 上緣時</li> <li>■ 當 Error 上緣時</li> <li>■ 當 CommandAborted 上緣時</li> </ul>

名稱	輸出變數上緣時機	輸出變數下緣時機
CommandAborted	<ul style="list-style-type: none"> <li>當此功能塊指令被其他緩衝模式設為mcAborting的指令中斷時</li> <li>當此功能塊指令被MC_Stop中斷時</li> </ul>	<ul style="list-style-type: none"> <li>當Execute下緣時</li> <li>當Execute為False，而CommandAborted轉為True，此時CommandAborted維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在ErrorID)	當Execute下緣時 (清除ErrorID紀錄之錯誤碼)

- 輸出變數的變化時序  
請參考圖 5.2.2.4.1 MC\_MoveAbsolute 時序圖

參考範例

此範例說明 MC\_MoveLinearJog 的運行方式，及執行時的運動軌跡。



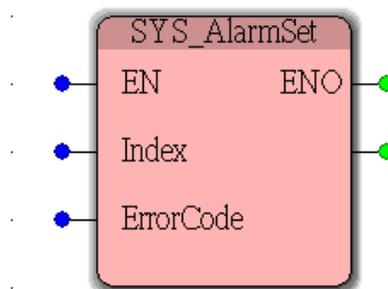
- Dir1 型態為 S\_WC\_POS，為一陣列。陣列中填寫的數值只看正負號而不看數值，範例中 Dir1 代表向 +X、方向進行吋動。
- 當 M2 轉為 True，MC\_MoveLinearJog 開始控制群組軸往設定方向(+X、+Y、-Z)移動。

# 5

## 5.2.5 系統功能塊

### 5.2.5.1 SYS\_AlarmSet

- 類型  
Function
- 功能描述  
將警報錯誤碼儲存至歷史錯誤記錄區域。
- 功能塊



註：使用 SYS\_AlarmSet 功能塊，建議開啟 EN/ENO 功能(Object→Insert Blocks with EN/ENO)。

#### ■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
EN	BOOL	True / False (False)	EN 上緣時執行指令，將指定的參數寫入歷史錯誤記錄區域。EN 上緣觸發時，若有其他警報已經存在，則此功能塊失效。
Index	WORD	0x0000 ~ 0xFFFF (0)	索引資料
ErrorCode	WORD	0x0000 ~ 0xFFFF (0)	錯誤碼資料

#### ■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
None	INT	-32768 / 32767 (0)	

#### ■ 功能說明- Index

索引資料 (16-Bits)			
U	Z	Y	X
編號		保留 (0)	類型

索引資料為 WORD 型態，使用者可自根據不同的情況來輸入對應的代碼，代碼定義如下：

編號：Group 或 Axis No.

■ 類型

0x0：控制類(Controller)

0x1：群組類(Group)

0x2：軸別類(Axis)

0x3：自訂類(User)

0x4 ~ 0xF：系統保留

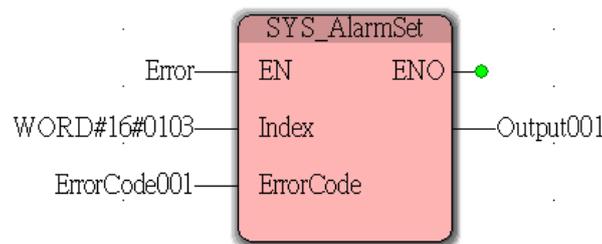
■ 功能說明- ErrorCode

錯誤碼資料為 WORD 型態，使用者可自行定義錯誤碼。

自行定義之錯誤碼請勿與 ASDA-MS 系統錯誤碼重複，以免混淆。關於 ASDA-MS 系統錯誤碼，詳細說明請參考 ASDA-MS 系列使用手冊第 11 章—異警排除。

■ 參考範例

此範例使用 SYS\_AlarmSet 功能塊，設定自定義之錯誤碼並儲存至歷史錯誤記錄區域。



1. 當 Error(EN)轉為 True 時，SYS\_AlarmSet 功能塊開始將索引資料(Index)0x0103 以及錯誤碼資料(ErrorCode)0x0900 儲存至歷史錯誤記錄區域。
2. 歷史錯誤記錄區域資料，可透過 DRAS 軟體則可切換至 [錯誤紀錄]，並點選更新按鈕來觀察。

時間	編號	類型	錯誤碼
2017/8/8 下午 08:31:15	1	User	0x0900
2017/8/8 下午 08:31:15	1	User	0x0900
2017/8/8 下午 08:22:00	1	User	0x009B DSP AD2異常
2017/8/8 下午 08:21:59	1	User	0x009B DSP AD2異常
2017/8/8 下午 08:21:55	1	User	0x009B DSP AD2異常
2017/8/8 下午 08:21:55	1	User	0x009B DSP AD2異常
2017/8/8 下午 08:19:49	1	User	0x009B DSP AD2異常

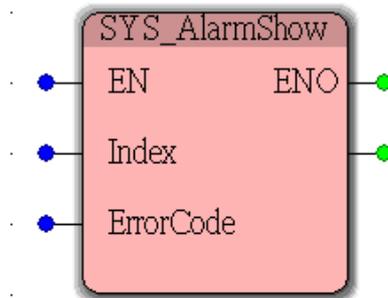
圖 5.2.5.1.1 MC\_AlarmSet 錯誤紀錄

3. 當使用 Modbus 通訊時，可透過(P3-00)+2 來讀取歷史錯誤記錄區域資料。

# 5

## 5.2.5.2 SYS\_AlarmShow

- 類型  
Function
- 功能描述  
將警報錯誤碼顯示至七段顯示器。
- 功能塊



註：使用 SYS\_AlarmShow 功能塊，建議開啟 EN/ENO 功能(Object→Insert Blocks with EN/ENO)。

### ■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
EN	BOOL	True / False(False)	EN 上緣時執行指令，將指定的參數寫入歷史錯誤記錄區域。 <sup>註</sup>
Index	WORD	0x0000 ~ 0xFFFF(0)	索引資料
ErrorCode	WORD	0x0000 ~ 0xFFFF(0)	錯誤碼資料，詳細錯誤碼資訊，請參考第5.1.4章節

註：EN 上緣觸發時，若有其他警報已經存在，則此功能塊失效。

### ■ 輸出變數

名稱	資料型態	設定值(預設值)	功能
None	INT	True / False(False)	-

### ■ 功能說明- Index

索引資料(16-Bits)			
U	Z	Y	X
編號		保留(0)	類型

索引資料為 WORD 型態，使用者可自根據不同的情況來輸入對應的代碼，代碼定義如下：

編號：Group 或 Axis No.

類型：

0x0：控制類(Controller)，七段顯示器以英文字母「C」顯示。

0x1：群組類(Group)，七段顯示器以數字「1. ~ 2.」顯示。

0x2：軸別類(Axis)，七段顯示器以 1 至 6 軸：以數字「1」~「6」顯示；7 至 12 軸：

保留；13 至 18 軸：以英文字母「D」~「I」顯示；0x3：自訂類(User)，七段顯示器以英文字母「U」顯示；0x4 ~ 0xF：系統保留

關於 ASDA-MS 七段顯示器顯示異警代碼方式，詳細說明請參考 ASDA-MS 系列使用手冊第 11 章—異警排除。

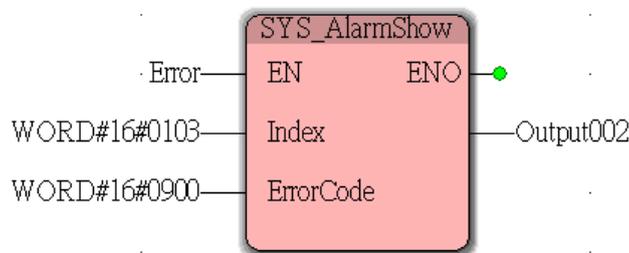
■ 功能說明- ErrorCode

錯誤碼資料為 WORD 型態，使用者可自行定義所需要的錯誤碼。

自行定義之錯誤碼請勿與 ASDA-MS 系統錯誤碼重複，以免混淆，關於 ASDA-MS 系統錯誤碼，詳細說明請參考 ASDA-MS 系列使用手冊第 11 章—異警排除。

■ 參考範例

此範例使用 SYS\_AlarmShow 功能塊，來顯示自定義之警報錯誤碼至七段顯示器

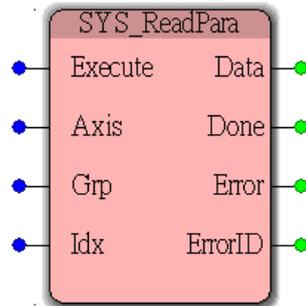


1. 當 Error(EN)轉為 True 時，SYS\_AlarmSet 開始將索引資料(Index)0x0103 以及錯誤碼資料(ErrorCode)0x0900 顯示至七段顯示器。
2. 觀察 ASDA-MS 本體之七段顯示器，顯示資料為 EU900。

# 5

## 5.2.5.3 SYS\_ReadPara

- 類型  
Function Block
- 功能描述  
ASDA-MS 本體軸參數讀取。
- 功能塊



註：使用 SYS\_ReadPara 功能塊，因為讀取參數的時間有可能會超過 2 ms (系統執行 PLC Event Task 之週期時間)，若將此功能塊建置於 Event Task 可能會發生逾時情況，建議將此功能塊建置於 Cyclic Task。關於 Cyclic Task 詳細說明請參考 3.4 節

### ■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False(False)	Execute上緣時執行指令，將指定的參數輸出至Data*1
Axis	INT	0 ~ 4 (0)	軸編號
Grp	INT	0 ~ 6 (0)	參數群組編號
Idx	INT	0 ~ 107 (0)	參數索引編號

註：

1. 當 Execute 上緣時執行指令，將指定的參數輸出至 Data。
2. 當 Execute 下緣時執行指令，將輸出 Data 資料清除為 0。

### ■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
Data	UDINT	0 ~ 4294967295	讀取參數資料內容
Done	BOOL	True / False (False)	讀取參數完成時為True
Error	BOOL	True / False (False)	讀取參數不存在時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	當讀取到不存在的參數值時所記錄的錯誤碼，錯誤碼詳細說明請參考第 5.1.4 章節

■ 輸出變數變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Data	當Execute轉為True / False時更新	當Execute轉為True / False時更新
Done	當讀取參數完成時	<ul style="list-style-type: none"> <li>■ 當Execute轉為False時</li> <li>■ 若Execute為False，而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在ErrorID)	當Execute下緣時(清除ErrorID紀錄之錯誤碼)

■ 輸出變數的變化時序

此時序將描述功能塊輸出變數在執行時的時序。

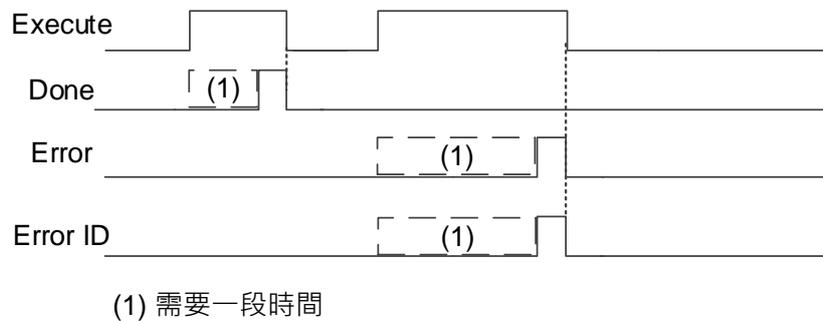
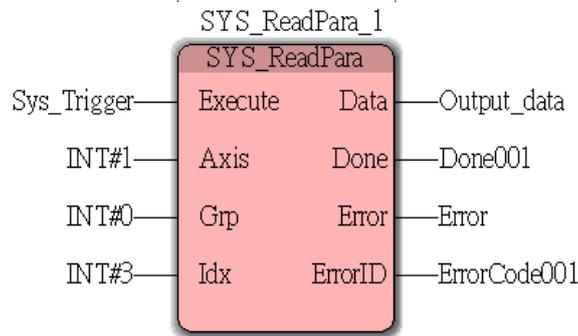


圖 5.2.5.3.1 SYS\_ReadPara 變數時序圖 (時序詳細說明請參考 5.1.3 節)

# 5

■ 參考範例

這個範例使用 SYS\_ReadPara 功能塊，來讀取 ASDA-MS 本體軸參數內容。



1. 當 Sys\_Trigger(Execute)轉為 TRUE 時，SYS\_ReadPara 將 ASDA-MS 軸 1 驅動器參數 P0-03 資料輸出至 Output\_data(Data)。
2. 完成輸出動作之後，Done001(Done)內容輸出為 True。

### 5.2.5.4 SYS\_WritePara

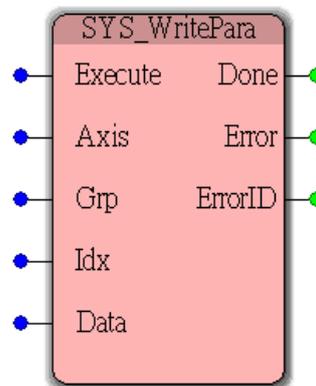
■ 類型

Function Block

■ 功能描述

ASDA-MS 本體軸參數寫入。

■ 功能塊



註：使用 SYS\_WritePara 功能塊，因為寫入參數的時間有可能會超過 2 ms(系統執行 PLC Event Task 之週期時間)，若將此功能塊建置於 Event Task 可能會發生逾時情況，建議將此功能塊建置於 Cyclic Task。關於 Cyclic Task 詳細說明請參考第 3.4 節。

■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時執行指令，將指定的參數輸出至Data
Axis	INT	0 ~ 4 (0)	軸編號
Grp	INT	0 ~ 6 (0)	參數群組編號
Idx	INT	0 ~ 107 (0)	參數索引編號
Data	UDINT	0 ~ 4294967295 (0)	寫入參數資料內容，每個參數之詳細範圍，請參考ASDA-MS系列使用手冊第8章-參數與功能。

■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	讀取參數完成時為True
Error	BOOL	True / False (False)	讀取參數不存在時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	當讀取到不存在的參數值時所記錄的錯誤碼，錯誤碼詳細說明請參考第5.1.4章節

■ 輸出變數變化時序

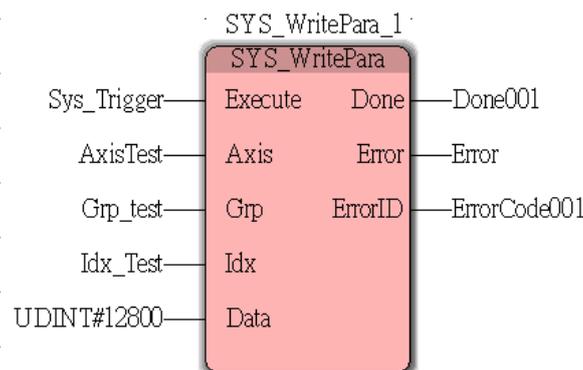
名稱	輸出變數上緣時機	輸出變數下緣時機
Done	讀取參數完成時	<ul style="list-style-type: none"> <li>■ 當Execute轉為False時</li> <li>■ 若Execute為False，而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在ErrorID)	在Execute下緣時(清除ErrorID紀錄之錯誤碼)

■ 輸出變數的變化時序

請參考圖 5.2.3.1 Sys\_ReadPara 時序圖

■ 參考範例

這個範例使用 SYS\_WritePara 功能塊去將資料寫入至 ASDA-MS 本體軸

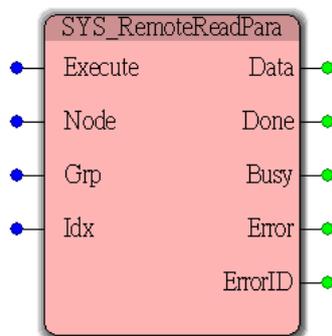


# 5

1. 當 Sys\_Trigger(Execute)轉為 True 時，SYS\_WritePara 將資料 12800(Data)寫入 ASDA-MS 軸 1 驅動器參數 P1-44。
2. 完成輸出動作之後，Done001(Done)內容輸出為 True。

### 5.2.5.5 SYS\_RemoteReadPara

- 類型  
Function Block
- 功能描述  
ASDA-MS 擴充軸參數讀取。
- 功能塊



註:使用 SYS\_RemoteReadPara 功能塊,因為讀取參數的時間有可能會超過 2 ms(系統執行 PLC Event Task 之週期時間),若將此功能塊建置於 Event Task 可能會發生逾時情況,建議將此功能塊建置於 Cyclic Task。關於 Cyclic Task 詳細說明請參考第 3.4 節。

#### ■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時執行指令，將指定的參數輸出至Data*1
Node	INT	1 ~ 12 (0)	擴充軸站號*2
Grp	INT	0 ~ 6 (0)	參數群組編號
Idx	INT	0 ~ 107 (0)	參數索引編號

註：

1. 當 Execute 上緣時執行指令，將指定的參數輸出至 Data。當 Execute 下緣時執行指令，將輸出 Data 資料清除為 0。
2. 當 ASDA-MS 外接擴充軸時，站號 1 必須存在，若站號 1 不存在則 ASDA-MS 掃描從站將會失敗。

■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
Data	UDINT	0 ~ 4294967295	讀取參數資料內容
Done	BOOL	True / False (False)	讀取參數完成時為True
Busy	BOOL	True / False (False)	指令被觸發執行時為True
Error	BOOL	True / False (False)	讀取參數不存在時為True
ErrorID	WORD	0x0000 ~ 0xFFFF(0)	當讀取參數不存在時記錄錯誤碼，錯誤碼詳細說明請參考第5.1.4章節

■ 輸出變數變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Data	當Execute 轉為True / False時更新	當Execute轉為True / False時更新
Done	讀取參數完成時	<ul style="list-style-type: none"> <li>■ 當Execute轉為False時</li> <li>■ 若Execute為False，而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Busy	當Execute上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在ErrorID)	在Execute下緣時(清除ErrorID紀錄之錯誤碼)

■ 輸出變數的變化時序

此時序將提供了三種不同狀況的輸出變數狀態:第一部分提供功能塊完成的時序圖；第二部分則是發生錯誤時被中斷的時序圖；而第三部分則說明 **Execute** 在啟動後釋放，功能塊依舊會完成整個命令。

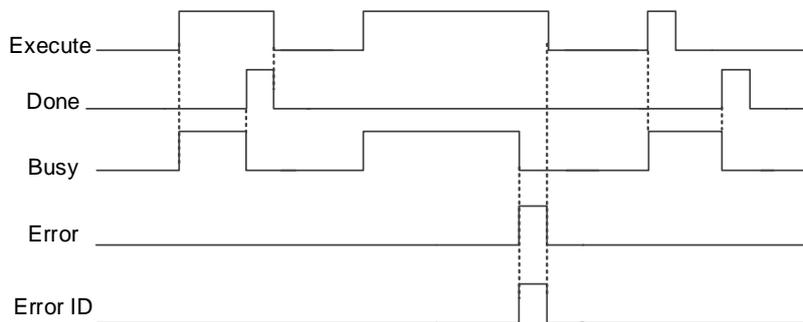
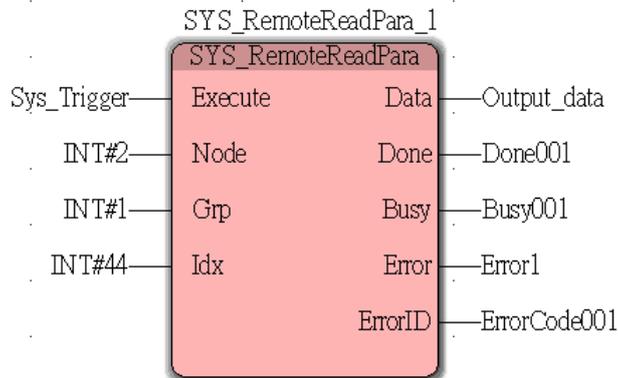


圖 5.2.5.5.1 SYS\_RemoteReadPara 變數時序圖(時序詳細說明請參考 5.1.3 節)

# 5

■ 參考範例

這個範例使用 SYS\_RemoteReadPara 功能塊去讀取 ASDA-MS 擴充軸參數內容



1. 當 Sys\_Trigger(Execute)轉為 TRUE 時，SYS\_RemoteReadPara 將 ASDA-MS 站號為 2 之擴充軸驅動器參數 P1-44 資料輸出至 Output\_data(Data)。
2. 完成輸出動作之後，Done001(Done)內容輸出為 True。

### 5.2.5.6 SYS\_RemoteWritePara

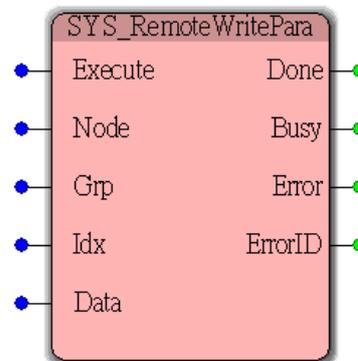
■ 類型

Function Block

■ 功能描述

ASDA-MS 擴充軸參數寫入。

■ 功能塊



註：使用 SYS\_RemoteWritePara 功能塊，因為寫入參數的時間有可能會超過 2 ms(系統執行 PLC Event Task 之週期時間)，若將此功能塊建置於 Event Task 可能會發生逾時情況，建議將此功能塊建置於 Cyclic Task。關於 Cyclic Task 詳細說明請參考第 3.4 節。

■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時執行指令，將指定的參數輸出至Data
Axis	INT	1 ~12 (0)	軸編號
Grp	INT	0 ~ 6 (0)	參數群組編號
Idx	INT	0 ~ 107 (0)	參數索引編號
Data	UDINT	0 ~ 4294967295 (0)	寫入參數資料內容，每個參數之詳細範圍，請參考ASDA-MS系列使用手冊第8章-參數與功能。

■ 輸出變數

名稱	資料型態	設定值(預設值)	功能
Done	BOOL	True / False (False)	讀取參數完成時為True
Busy	BOOL	True / False (False)	指令被觸發執行時為True
Error	BOOL	True / False (False)	讀取參數不存在時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	當讀取到不存在的參數值時所記錄的錯誤碼，錯誤碼詳細說明請參考第5.1.4章節

■ 輸出變數變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Done	讀取參數完成時	<ul style="list-style-type: none"> <li>■ 當Execute轉為False時</li> <li>■ 若Execute為False，而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Busy	當Execute上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在ErrorID)	在Execute下緣時 (清除ErrorID紀錄之錯誤碼)

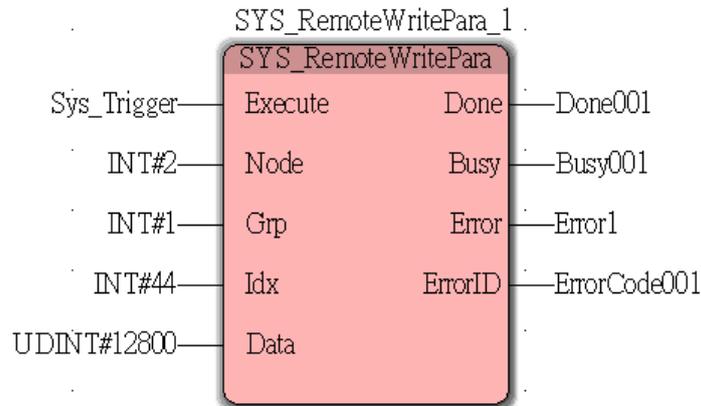
■ 輸出變數的變化時序

請參考圖 5.2.5.4.1。

# 5

■ 參考範例

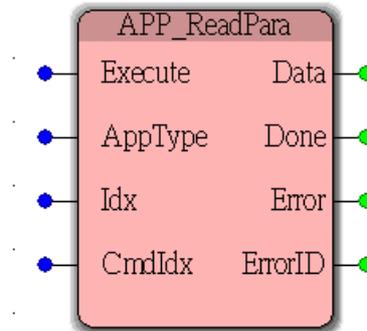
此範例使用 SYS\_RemoteWritePara 功能塊去將資料寫入至 ASDA-MS 本體軸。



1. 當 Sys\_Trigger(Execute)轉為 True 時，SYS\_RemoteWritePara 將資料 12800(Data) 寫入 ASDA-MS 站號為 2 之擴充軸驅動器參數 P1-44。
2. 完成輸出動作之後，Done001(Done)內容輸出為 True。

### 5.2.5.7 APP\_ReadPara

- 類型  
Function Block
- 功能描述  
產業應用參數讀取。
- 功能塊



註：使用 APP\_ReadPara 功能塊，因為讀取參數的時間有可能會超過 2 ms (系統執行 PLC Event Task 之週期時間)，若將此功能塊建置於 Event Task 可能會發生 Time out 情況，建議將此功能塊建置於 Cyclic Task。關於 Cyclic Task 詳細說明請參考第 3.4 節。

#### ■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時執行指令，將指定的參數輸出至Data
AppType	UINT	0 ~ 4294967295 (0)	產業應用類型
Idx	UINT	0 ~ 4294967295 (0)	產業應用編號
CmdIdx	UINT	0 ~ 4294967295 (0)	產業應用索引碼

#### ■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
Data	UDINT	0 ~ 4294967295	讀取參數資料內容
Done	BOOL	True / False (False)	讀取參數完成時為True
Error	BOOL	True / False (False)	讀取參數不存在時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	當讀取到不存在的參數值時所記錄的錯誤碼，錯誤碼詳細說明請參考第 5.1.4 章節

# 5

■ 輸出變數變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Data	當Execute轉為True / False時更新	當Execute轉為True / False時更新
Done	讀取參數完成時	<ul style="list-style-type: none"> <li>■ 當Execute轉為False時</li> <li>■ 若Execute 為False而Done轉為True，此時Done維持一個掃描週期的 True狀態後，立刻轉 False</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在ErrorID)	在Execute下緣時 (清除ErrorID紀錄之錯誤碼)

■ 輸出變數的變化時序

請參考圖 5.2.3.1 Sys\_ReadPara 時序圖

■ 功能說明- AppType

產業應用類型：

0：輸送帶追隨

1：手輪

2：TP

■ 功能說明- Idx

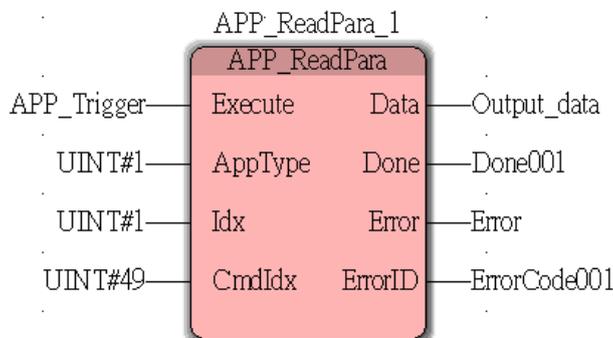
產業應用編號：操作的元件編號，ASDA-MS 目前支援兩組輸送帶，故可填入 1 或 2。

■ 功能說明- CmdIdx

產業應用索引碼包含輸送帶追隨產業，手輪操作索引碼，TP 操作索引碼，詳細說明請向原廠工程師索取。

■ 參考範例

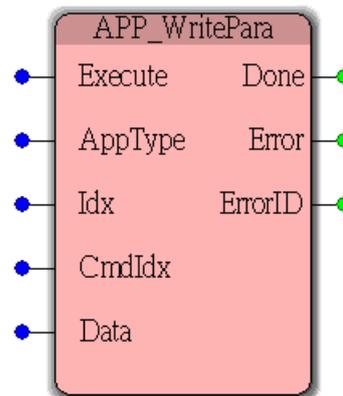
這個範例使用 APP\_ReadPara 功能塊去讀取輸送帶追隨相關參數



1. 當 APP\_Trigger(Execute)轉為 True 時，APP\_ReadPara 將讀取輸送帶追隨應用的 1 號輸送帶(Idx)的視覺坐標系參數 X 方向位移量(CmdIdx)，輸出至 Output\_data(Data)。
2. 完成輸出動作之後，Done001(Done)內容輸出為 True。

### 5.2.5.8 APP\_WritePara

- 類型  
Function Block
- 功能描述  
產業應用參數寫入。
- 功能塊



註：使用 APP\_WritePara 功能塊，因為寫入參數的時間有可能會超過 2 ms(系統執行 PLC Event Task 之週期時間)，若將此功能塊建置於 Event Task 可能會發生逾時情況，建議將此功能塊建置於 Cyclic Task。關於 Cyclic Task 詳細說明請參考第 3.4 節。

#### ■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時執行指令，將指定的參數輸出至Data
AppType	UINT	0 ~ 4294967295 (0)	產業應用類型
Idx	UINT	0 ~ 4294967295 (0)	產業應用編號
CmdIdx	UINT	0 ~ 4294967295 (0)	產業應用索引碼
Data	UDINT	0 ~ 4294967295 (0)	寫入參數資料內容，每個參數之詳細範圍，請參考ASDA-MS系列使用手冊第8章-參數與功能。

#### ■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	讀取參數完成時為True
Error	BOOL	True / False (False)	讀取參數不存在時為True
ErrorID	WORD	0x0000~0xFFFF (0)	當讀取到不存在的參數值時所記錄的錯誤碼，錯誤碼詳細說明請參考第 5.1.4 章節

# 5

■ 輸出變數變化時序

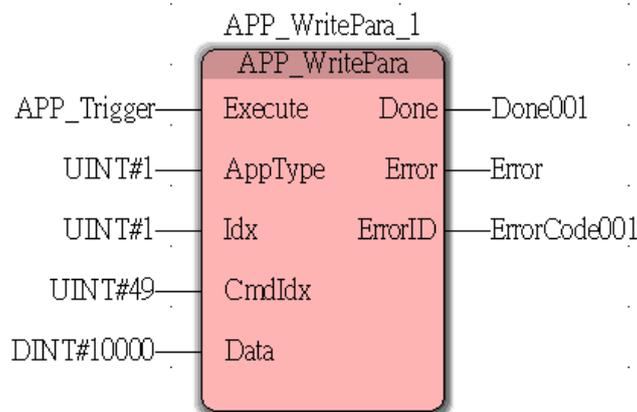
名稱	輸出變數上緣時機	輸出變數下緣時機
Done	讀取參數完成時	<ul style="list-style-type: none"> <li>■ 當Execute轉為False時</li> <li>■ 若Execute為False，而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在ErrorID)	在Execute下緣時 (清除ErrorID紀錄之錯誤碼)

■ 輸出變數的變化時序

請參考圖 5.2.3.1 Sys\_ReadPara 時序圖請參考圖 5.2.5.3.1

■ 參考範例

此範例使用 APP\_WritePara 功能塊寫入輸送帶追隨相關參數



1. 當 APP\_Trigger(Execute)轉為 True 時，APP\_WritePara 將輸送帶 1 號(Idx)的 X 方向位移量(CmdIdx)設為 10000 um(Data)。
2. 完成輸出動作之後，Done001(Done)內容輸出為 True。

### 5.2.5.9 DRL\_CommandSet

- 類型  
Function Block
- 功能描述  
機械程序(Robot Language)的執行狀態設定。
- 功能塊



#### ■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時執行指令，將設定專案的機械程序的執行狀態
CommandType	DINT	1 ~ 23 (0)	機械程序執行控制類型
ProgramNo	DINT	0 ~ 9999 (0)	機械程序專案編號

#### ■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	讀取參數完成時為True
Error	BOOL	True / False (False)	讀取參數不存在時為True
ErrorID	WORD	0x0000 ~ 0xFFFF(0)	當讀取到不存在的參數值時所記錄的錯誤碼，錯誤碼詳細說明請參考第5.1.4章節

#### ■ 輸出變數變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Done	讀取參數完成時	<ul style="list-style-type: none"> <li>■ 當Execute轉為False時</li> <li>■ 若Execute為False，而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在 ErrorID)	在Execute下緣時(清除ErrorID紀錄之錯誤碼)

## 5

- 輸出變數的變化時序

變數時序圖請參考圖 5.2.5.3.1

- 功能說明- CommandType

索引資料為 DINT 型態，使用者可根據不同的情況來輸入對應的類型，類型定義如下：

1 : Open ; 2 : Run ; 3 : Stop ; 4 : Pause ; 6 : Open & Run

11 : Step ; 12 : Step Over ; 13 : Step Out

21 : Block Run ; 22 : Open ; 23 : Block Run

- 參考範例

這個範例使用 DRL\_CommandSet 功能塊來設定機械程序(Robot Language)的執行狀態。



1. 當 RL\_Trigger(Execute)轉為 True 時，DRL\_CommandSet 將開啟並執行 (CommandType)專案編號 1(ProgramNo)的程序。
2. 完成輸出動作之後，Done001(Done)內容輸出為 True。

### 5.2.5.10 DRL\_GetStatus

- 類型  
Function
- 功能描述  
讀取目前機械程序(Robot Language)的執行狀態。
- 功能塊



■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
EN_IN	BOOL	True / False (False)	EN_IN為True時執行指令，將讀取機械程序的執行狀態。

■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
None	DINT	- 2147483648 ~ 2147483647(0)	顯示機械程序執行狀態

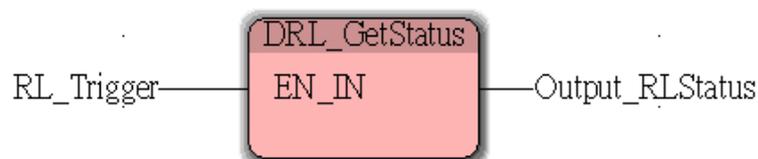
■ 功能說明

索引資料為 DINT 型態，使用者可自根據讀取結果的代碼來得知目前機械程序的執行狀態，代碼定義如下：

- 0：未啟動(Close)
- 1：執行中(Running)
- 2：斷點(Break Point)
- 3：暫停(Pause)
- 4：預執行(Pre-Run)

■ 參考範例

此範例使用 DRL\_GetStatus 功能塊來讀取目前機械程序的執行狀態。

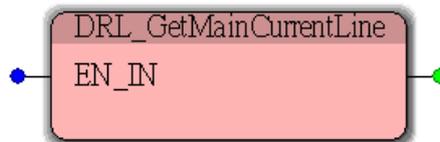


假設機械程序目前為運行狀態，當 RL\_Trigger(EN\_IN)轉為 True 時，DRL\_GetStatus 功能塊將讀取目前機器人腳本的執行狀態並將狀態輸出至輸出變數，因此 Output\_RLStatus 內容為 1。

# 5

## 5.2.5.11 DRL\_GetMainCurrentLine

- 類型  
Function
- 功能描述  
讀取目前機械程序(Robot Language)在 main.lua(機械程序執行檔)的執行行數。
- 功能塊



■ 輸入變數

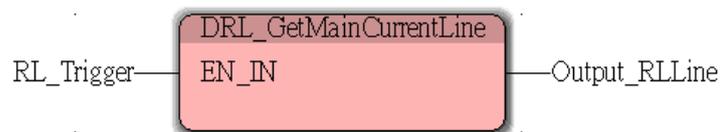
名稱	資料型態	設定值 (預設值)	功能
EN_IN	BOOL	True / False (False)	EN_IN為True時執行指令·將讀取目前機械程序的執行行數。

■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
None	UDINT	0 ~ 4294967295 (0)	顯示機械程序執行行數

■ 參考範例

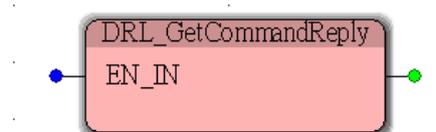
此範例使用 DRL\_GetMainCurrentLine 功能塊來讀取目前 RL 的執行狀態。



假設機械程序目前為運行狀態且正在執行第 9 行，當 RL\_Trigger(EN\_IN)轉為 True 時，DRL\_GetMainCurrentLine 功能塊將讀取目前機械程序中，所執行到的程序行號並輸出至輸出腳位上，因此 Output\_RLLine 內容為 9。

### 5.2.5.12 DRL\_GetCommandReply

- 類型  
Function
- 功能變數描述  
讀取機械程序(Robot Language)的回覆狀態。
- 功能塊



■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
EN_IN	BOOL	True / False (False)	EN_IN為True時執行指令，將讀取RL的回覆狀態。此功能塊與DRL_Status功能塊類似，但是此功能塊代碼較為詳細；通常與DRL_CommandSet搭配使用。

輸出腳位

名稱	資料型態	設定值 (預設值)	功能
None	DINT	- 2147483648 ~ 2147483647 (-1)	顯示機械程序執行狀態

■ 功能說明

索引資料為 DINT 型態，使用者可根據讀取結果的代碼，來得知目前機械程序的回覆狀態，代碼定義如下：

- 0 : BUSY
- 1 : CHECKING
- 10 : CANNOT\_OPEN ; 11 : OPENING ; 12 : OPENED
- 20 : CANNOT\_RUN ; 21 : RUNNING
- 30 : CANNOT\_STOP ; 31 : STOPPING ; 32 : STOPPED
- 40 : CANNOT\_PAUSE ; 41 : PAUSED
- 50 : CANNOT\_STEP ; 51 : STEPPED

# 5

■ 參考範例

此範例使用 DRL\_GetCommandReply 功能塊來讀取目前機械程序的回覆狀態。



假設機械程序目前為運行狀態；當 RL\_Trigger(EN\_IN)轉為 True 時，DRL\_GetCommandReply 將讀取目前機械程序的回覆狀態至輸出，因此 Output\_RLCommandReply 內容為 21。

### 5.2.5.13 DRL\_GetCurrentProgNo

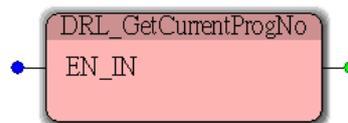
■ 類型

Function

■ 功能描述

讀取目前機械程序(Robot Language)的專案編號。

■ 功能塊



■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
EN_IN	BOOL	True / False (False)	EN_IN為True時執行指令，將讀取目前機械程序的專案編號。

■ 輸出變數

名稱	資料型態	設定值(預設值)	功能
None	DINT	- 2147483648 ~ 2147483647 (-1)	顯示機械程序執行狀態

■ 參考範例

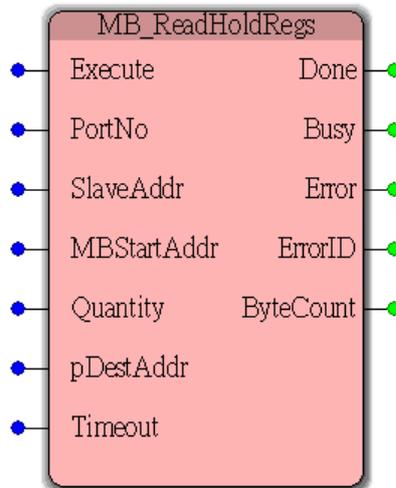
此範例使用 DRL\_GetCurrentProgNo 功能塊，來讀取目前機械程序的專案編號。



當 RL\_Trigger(EN\_IN)轉為 True 時，DRL\_GetCurrentProgNo 將讀取目前機械程序的回覆狀態至輸出 Output\_RLCurrentNo。

### 5.2.5.14 MB\_ReadHoldRegs

- 類型  
Function Block
- 功能描述  
透過串列傳輸讀取外部 Modbus 傳輸資料，並傳回 ASDA-MS 記憶體。
- 功能塊



註：使用 MB\_ReadHoldRegs 功能塊，因為寫入參數的時間有可能會超過 2 ms(系統執行 PLC Event Task 之週期時間)，若將此功能塊建置於 Event Task 可能會發生逾時情況，建議將此功能塊建置於 Cyclic Task。關於 Cyclic Task 詳細說明請參考第 3.4 節。

#### ■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時執行指令，將執行讀取外部Modbus動作
PortNo	USINT	0 ~ 255 (0)	通訊埠編號*1
SlaveAddr	USINT	1 ~ 247 (0)	指定從站站號
MBStartAddr	WORD	0x0000 ~ 0xFFFF(0)	外部Modbus起始通訊位址
Quantity	UINT	1 ~ 123 (0)	讀取暫存器個數，單位：WORD
pDestAddr	ANY*2	0 ~ 65535 (0)	儲存回傳資料
Timeout	TIME*3	正數、T#0s (T#0s)	逾時設定

註：

1. ASDA-MS 目前僅有一個通訊埠，因此通訊埠編號固定為 0；使用者可由參數 P3-05 來選擇 RS-232 或 RS-485 通訊界面。
2. pDestAddr 資料型態為 ANY，可以根據需求來選擇 WORD、DWORD、LWORD 或是其他型態，亦可搭配自定義型態，自定義型態詳細說明請參考 3.6 節。
3. 輸入格式為 IEC-61131-3 的 TIME 格式。

# 5

■ 輸出變數

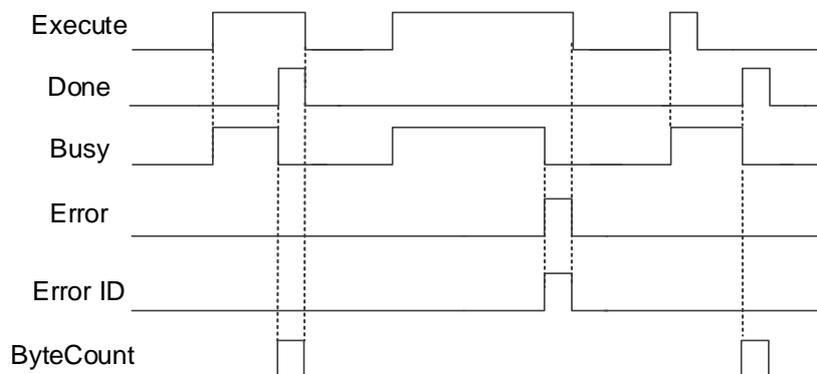
名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	讀取參數完成時為True
Busy	BOOL	True / False (False)	指令被觸發執行時為True
Error	BOOL	True / False (False)	讀取參數不存在時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	當讀取到不存在的參數值時所記錄的錯誤碼，錯誤碼詳細說明請參考第5.1.4章節
ByteCount	USINT	0 ~ 255 (0)	讀取參數完成時，顯示讀取成功的數量(單位：Byte)

■ 輸出變數變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Done	當讀取參數完成時	<ul style="list-style-type: none"> <li>■ 當Execute轉為False時</li> <li>■ 若Execute為False，而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Busy	當 Execute 上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在ErrorID)	在Execute下緣時 (清除ErrorID紀錄之錯誤碼)
ByteCount	當讀取參數完成時	在Execute下緣時

■ 輸出變數的變化時序

此輸出變數說明了三種情況：第一部分是完成功能塊並且獲得輸出數值；第二部分說明有錯誤產生，因此功能塊中斷並且無獲得正確的數值；第三部分則說明了Execute 啟動後並無等待獲得數值後才釋放，而功能塊依舊完成整個程序才截止。



## ■ 參考範例

這個範例使用 MB\_ReadHoldRegs 功能塊來讀取外部 Modbus 資料。

自定義型態 USER\_ARRAY 為大小 5 個 WORD 之陣列。

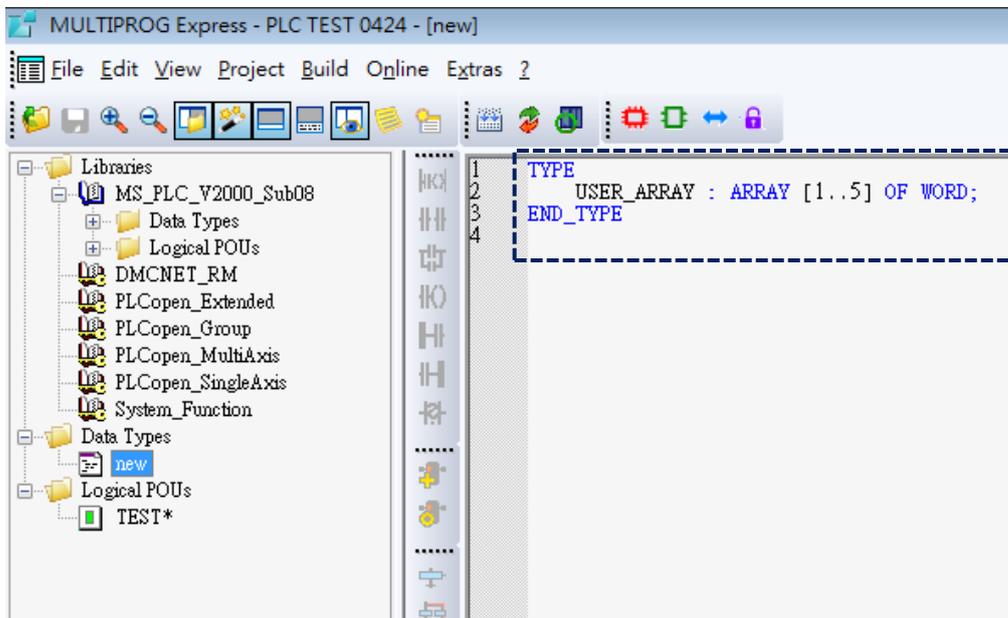


圖 5.2.5.14.1 自定義陣列

5

設定 pDestAddr 變數之資料型態為 USER\_ARRAY。

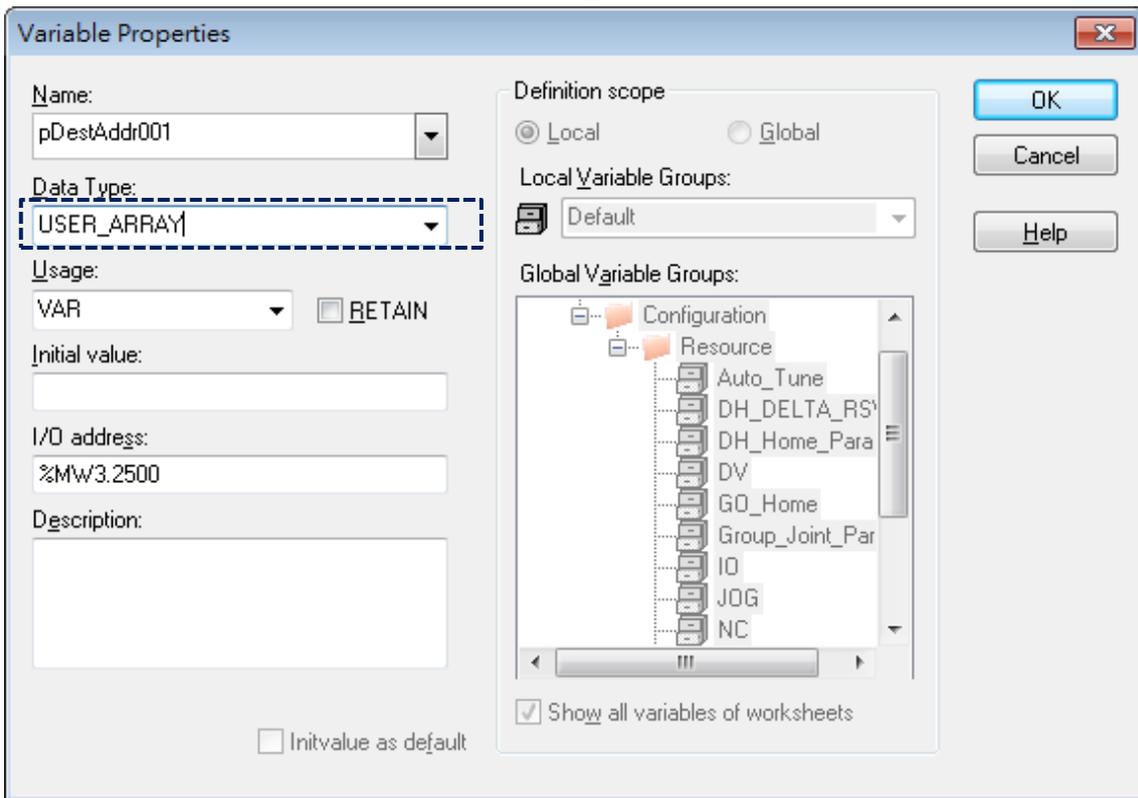


圖 5.2.5.14.2 自定義陣列設定

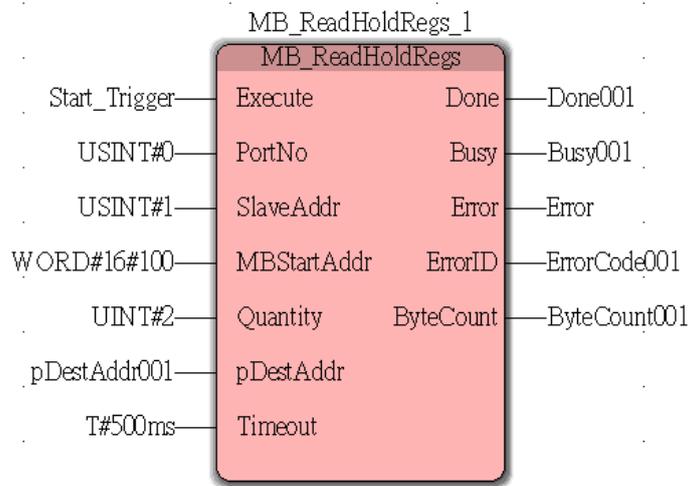
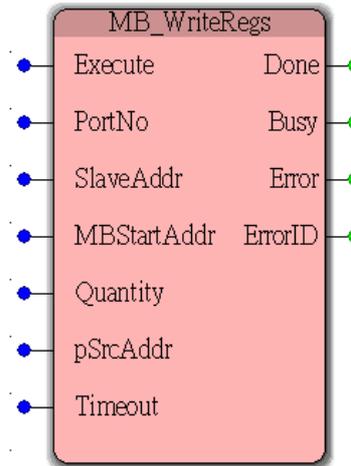


圖 5.2.5.14.3 MB\_ReadHoldRegs 使用參考範例

1. 當 Start\_Trigger(Execute)轉為 True 時，MB\_ReadHoldRegs 功能塊透過 Modbus 將外部站號 1(SlaveAddr)裝置的 0x100(MBStartAddr)位址資料讀取 2 筆 (Quantity)WORD 資料，並回傳至 USER\_ARRAY(pDestAddr)，並設定逾時時間為 (Timeout)為 500 ms。
2. 完成輸出動作之後，Done001(Done)內容輸出為 True。

### 5.2.5.15 MB\_WriteRegs

- 類型  
Function Block
- 功能描述  
透過串列傳輸將 ASDA-MS 記憶體資料，寫入至外部 Modbus 傳輸裝置。
- 功能塊



註：使用 MB\_WriteRegs 功能塊，因為寫入參數的時間有可能會超過 2 ms (系統執行 PLC Event Task 之週期時間)，若將此功能塊建置於 Event Task 可能會發生逾時情況，建議將此功能塊建置於 CyclicTask。關於 Cyclic Task 詳細說明請參考第 3.4 節。

#### ■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時執行指令，將執行讀取外部Modbus動作
PortNo	USINT	0 ~ 255 (0)	通訊埠編號*1
SlaveAddr	USINT	1 ~ 247 (0)	指定從站站號
MBStartAddr	WORD	0x0000 ~ 0xFFFF (0)	外部Modbus起始通訊位址
Quantity	UINT	1 ~123 (0)	讀取暫存器個數，單位：WORD
pDestAddr	ANY*2	0 ~ 65535 (0)	寫入外部裝置的資料
Timeout	TIME*3	正數、T#0s (T#0s)	逾時設定

註：

1. ASDA-MS 目前僅有一個通訊埠，因此通訊埠編號固定為 0；使用者可由控制器參數 P3-05 來選擇 RS-232 或 RS-485 通訊界面。
2. pDestAddr 資料型態為 ANY，可以根據需求來選擇 WORD、DWORD、LWORD 或是其他型態，亦可搭配自定義型態，自定義型態詳細說明請參考第 3.6 節。
3. 輸入格式為 IEC-61131-3 的 TIME 格式。

# 5

■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	當讀取參數完成時為True
Busy	BOOL	True / False (False)	當指令被觸發執行時為True
Error	BOOL	True / False (False)	當讀取參數不存在時為True
ErrorID	DWORD	0x0000 ~ 0xFFFF (0)	當讀取參數不存在時記錄錯誤碼，錯誤碼詳細說明請參考5.1.4節

■ 輸出變數變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Done	讀取參數完成時	<ul style="list-style-type: none"> <li>■ 當 Execute轉為False時</li> <li>■ 若 Execute為False，而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Busy	當Execute上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在ErrorID)	在Execute下緣時 (清除ErrorID紀錄之錯誤碼)

■ 輸出變數的變化時序

請參考圖 5.2.3.1 Sys\_ReadPara 時序圖

■ 參考範例

這個範例使用 MB\_WriteRegs 功能塊來寫入資料至外部 Modbus 裝置。  
自定義型態 USER\_ARRAY 為大小 5 個 WORD 之陣列。

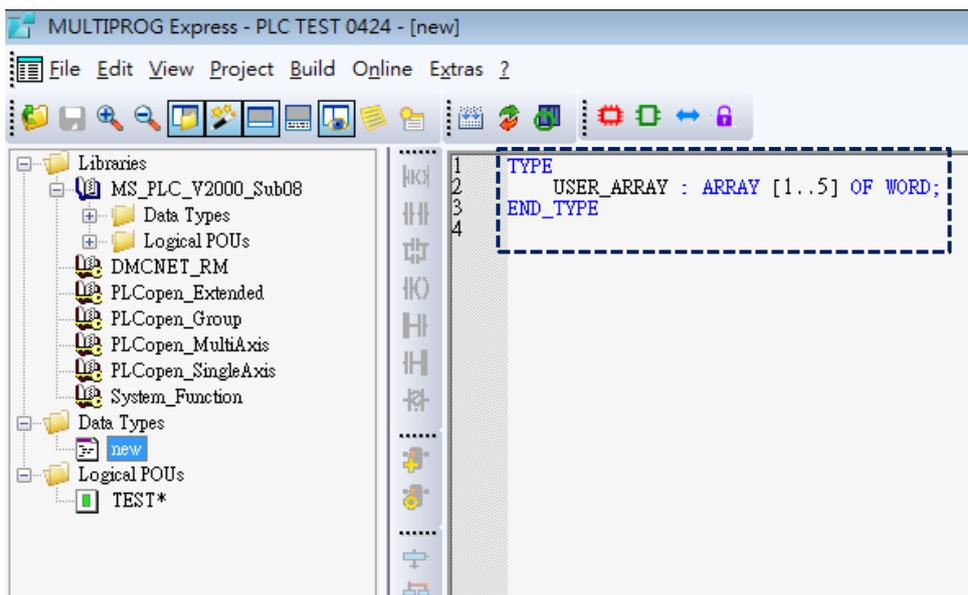


圖 5.2.5.15.1 自定義陣列

設定 pDestAddr 變數之資料型態為 USER\_ARRAY 。

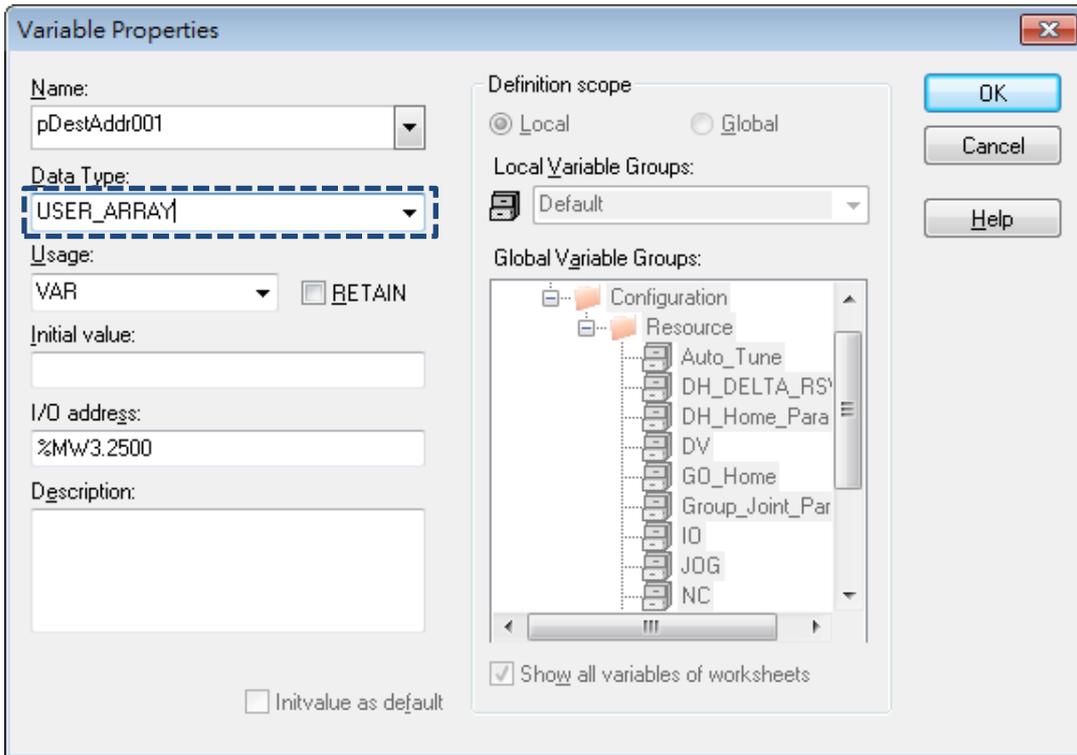


圖 5.2.5.15.2 自定義陣列設定

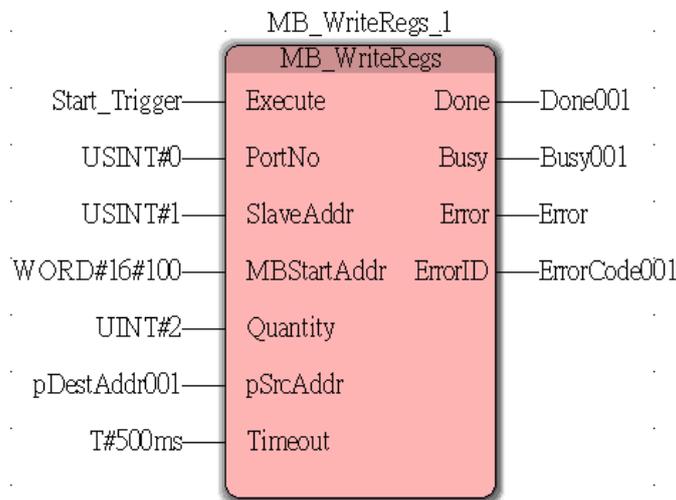


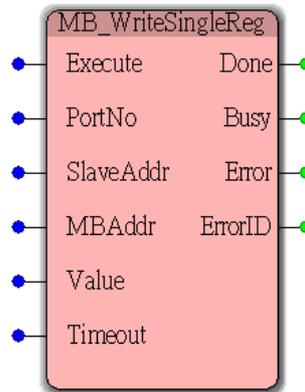
圖 5.2.5.15.3 MB\_WriteRegs 使用參考範例

1. 當 Start\_Trigger(Execute)轉為 True 時，MB\_WriteRegs 功能塊透過 Modbus 將 2 筆 (Quantity)資料 USER\_ARRAY(pDestAddr)寫入外部站號 1 (SlaveAddr)裝置的 0x100 (MBStartAddr)位址，並設定逾時時間為(Timeout)為 500 ms。
2. 完成輸出動作之後，Done001(Done)內容輸出為 True。

# 5

## 5.2.5.16 MB\_WriteSingleRegs

- 類型  
Function Block
- 功能描述  
透過串列傳輸將一筆 ASDA-MS 記憶體資料，寫入至外部 Modbus 裝置。
- 功能塊



註：使用 MB\_WriteSingleRegs 功能塊，因為寫入參數的時間有可能會超過 2 ms(系統執行 PLC Event Task 之週期時間)，若將此功能塊建置於 Event Task 可能會發生逾時情況，建議將此功能塊建置於 Cyclic Task。關於 Cyclic Task 詳細說明請參考第 3.4 節。

### ■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時執行指令，將執行讀取外部Modbus動作。
PortNo	USINT	0 ~ 255 (0)	通訊埠編號*1
SlaveAddr	USINT	1 ~ 247 (0)	指定從站站號
MBAAddr	WORD	0x0000 ~ 0xFFFF (0)	外部Modbus起始通訊位址
Value	WORD	0x0000~0xFFFF (0)	寫入外部裝置資料
Timeout	TIME*2	正數、T#0s (T#0s)	逾時設定

註：

1. ASDA-MS 目前僅有一個通訊埠，因此通訊埠編號固定為 0；使用者可由控制器參數 P3-05 來選擇 RS-232 或 RS-485 通訊界面。
2. 輸入格式為 IEC-61131-3 的 TIME 格式。

■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	讀取參數完成時為True
Busy	BOOL	True / False (False)	指令被觸發執行時為True
Error	BOOL	True / False (False)	讀取參數不存在時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	當讀取到不存在的參數值時所記錄的錯誤碼，錯誤碼詳細說明請參考第5.1.4章節

■ 輸出變數變化時序

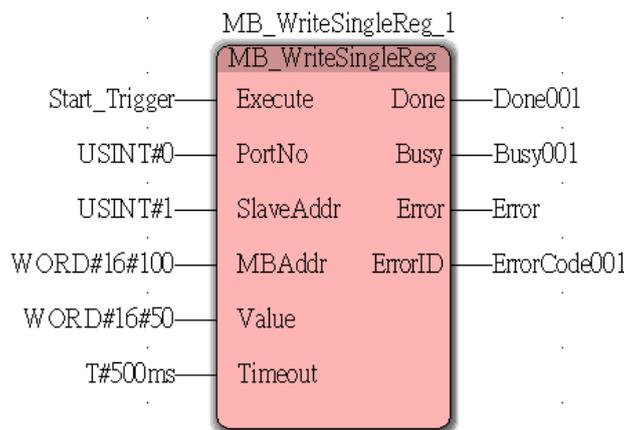
名稱	輸出變數上緣時機	輸出變數下緣時機
Done	讀取參數完成時	<ul style="list-style-type: none"> <li>■ 當Execute轉為False時</li> <li>■ 若Execute為False，而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Busy	當Execute上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在ErrorID)	在Execute下緣時 (清除ErrorID紀錄之錯誤碼)

■ 輸出變數的變化時序

請參考圖 5.2.3.1 Sys\_ReadPara 時序圖

■ 參考範例

此範例使用 MB\_WriteSingleRegs 功能塊，以寫入單筆資料至外部 Modbus 裝置。

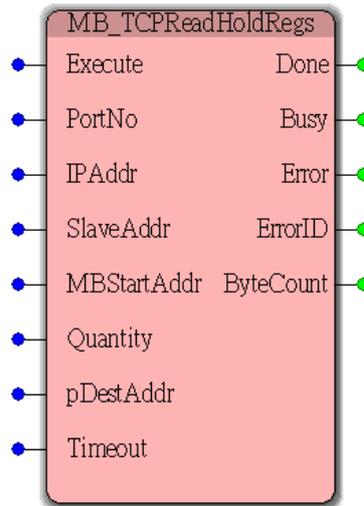


1. 當 Start\_Trigger(Execute)轉為 True 時，MB\_WriteSingleRegs 功能塊透過 Modbus 將一筆資料 0x50(Value)寫入外部站號 1(SlaveAddr)裝置的 0x100(MBStartAddr)位址，並設定逾時時間為(Timeout)為 500 ms。
2. 完成輸出動作之後，Done001(Done)內容輸出為 True。

5

5.2.5.17 MB\_TCPReadHoldRegs

- 類型  
Function Block
- 功能描述  
透過 EtherNet 傳輸讀取外部 Modbus 資料，並傳回 ASDA-MS 記憶體。
- 功能塊



註：使用 MB\_TCPReadHoldRegs 功能塊，因為寫入參數的時間有可能會超過 2 ms(系統執行 PLC Event Task 之週期時間)，若將此功能塊建置於 Event Task 可能會發生逾時情況，建議將此功能塊建置於 Cyclic Task。關於 Cyclic Task 詳細說明請參考 3.4 節。

■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時執行指令，將執行讀取外部Modbus動作。
PortNo	USINT	0 ~ 255 (0)	通訊埠編號*1
IP Addr	STRING	空字串	指定外部裝置IP位址
SlaveAddr	USINT	1 ~ 247 (0)	指定從站站號
MBStartAddr	WORD	0x0000 ~ 0xFFFF (0)	外部Modbus起始通訊位址
Quantity	UINT	1 ~ 123 (0)	讀取暫存器個數，單位：WORD
pDestAddr	ANY*1	0 ~ 65535 (0)	儲存回傳資料
Timeout	TIME*2	正數、T#0s (T#0s)	逾時設定

註：

1. ASDA-MS 目前僅有一個通訊埠，因此通訊埠編號固定為 0；使用者可由控制器參數 P3-24 (EtherNet 網路設定)、P3-25(EtherNet IP 位址設定)、P3-26(EtherNet 子網路遮罩設定)、與 P3-27 (EtherNet 預設開道設定)來設定相關 EtherNet 參數設定。且使用 TCP 模式時，P3-02(通訊協定)目前支援 RTU 格式。
2. pDestAddr 資料型態為 ANY，可以根據需求來選擇 WORD、DWORD、LWORD 或是其他型態，亦可搭配自定義型態，自定義型態詳細說明請參考第 3.6 節。
3. 輸入格式為 IEC-61131-3 的 TIME 格式。

■ 輸出變數

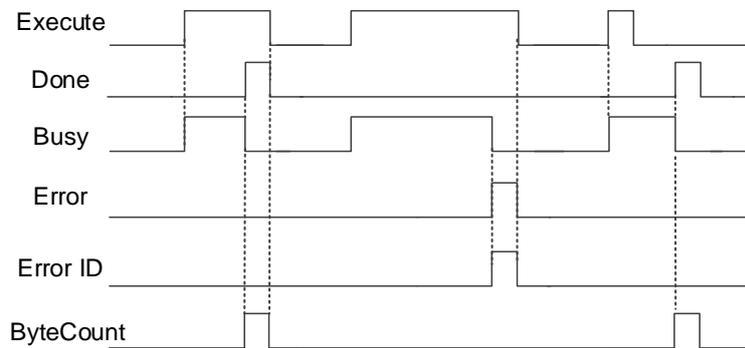
名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	讀取參數完成時為True
Busy	BOOL	True / False (False)	指令被觸發執行時為True
Error	BOOL	True / False (False)	讀取參數不存在時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	當讀取參數不存在時記錄錯誤碼·錯誤碼詳細說明請參考第5.1.4章節
ByteCount	USINT	0 ~ 255 (0)	讀取參數完成時·顯示讀取成功的數量 (單位: Byte)

■ 輸出變數變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Done	當讀取參數完成時	<ul style="list-style-type: none"> <li>■ 當Execute轉為False 時</li> <li>■ 若Execute為False·而Done轉為 True·此時Done維持一個掃描週期的True狀態後·立刻轉為False</li> </ul>
Busy	當 Execute 上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在ErrorID)	在Execute下緣時 (清除ErrorID紀錄之錯誤碼)
ByteCount	當讀取參數完成時	在Execute下緣時

■ 輸出變數的變化時序

此輸出變數說明了三種情況：第一部分是完成功能塊並且獲得輸出數值；第二部分情況為發生錯誤·因此功能塊中斷並且無獲得正確的數值；第三部分則說明了Execute 啟動後並無等待獲得數值後才釋放·而功能塊依舊完成整個程序才停止。



5

■ 參考範例

這個範例使用 MB\_TCPReadHoldRegs 功能塊來讀取外部 Modbus 資料。

自定義型態 USER\_ARRAY 為大小 5 個 WORD 之陣列。

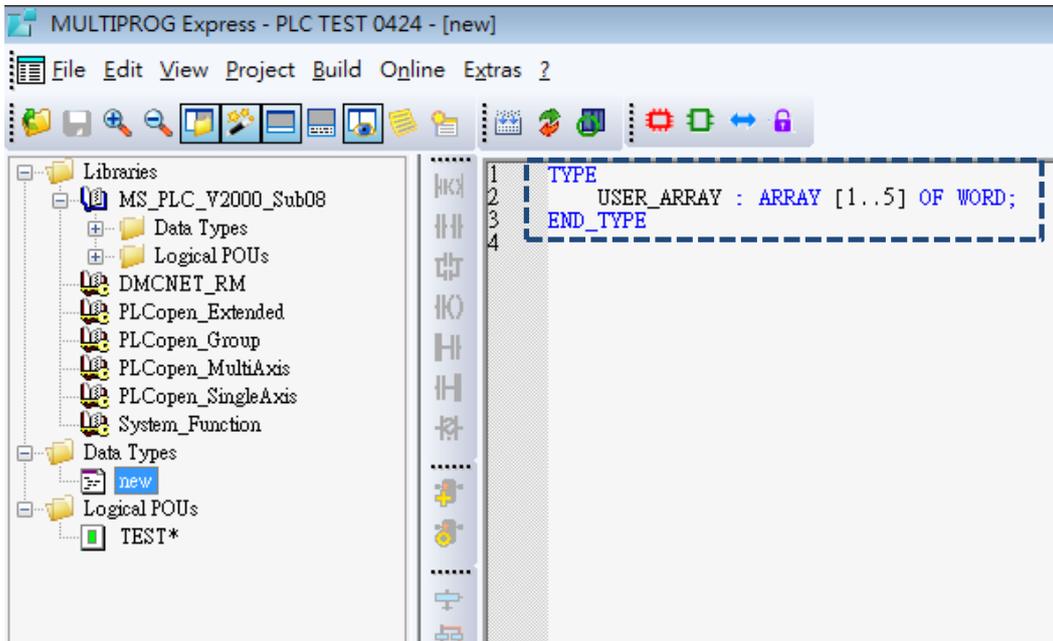


圖 5.2.5.17.1 自定義陣列

設定 pDestAddr 變數之資料型態為 USER\_ARRAY。

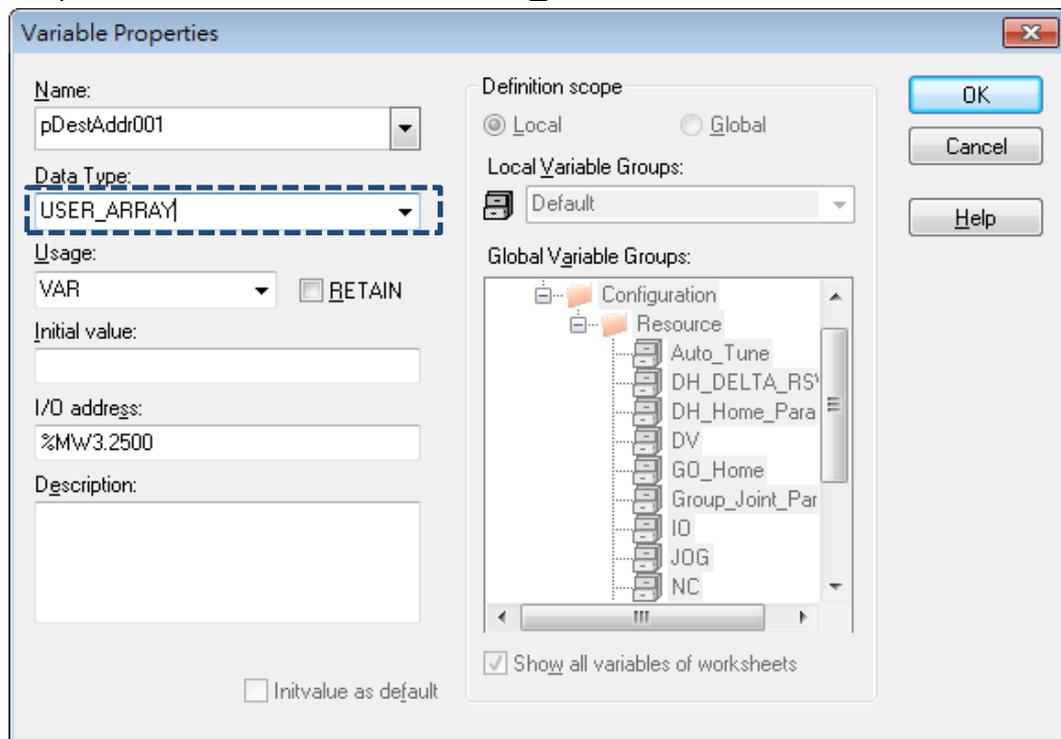


圖 5.2.5.17.2 自定義陣列設定

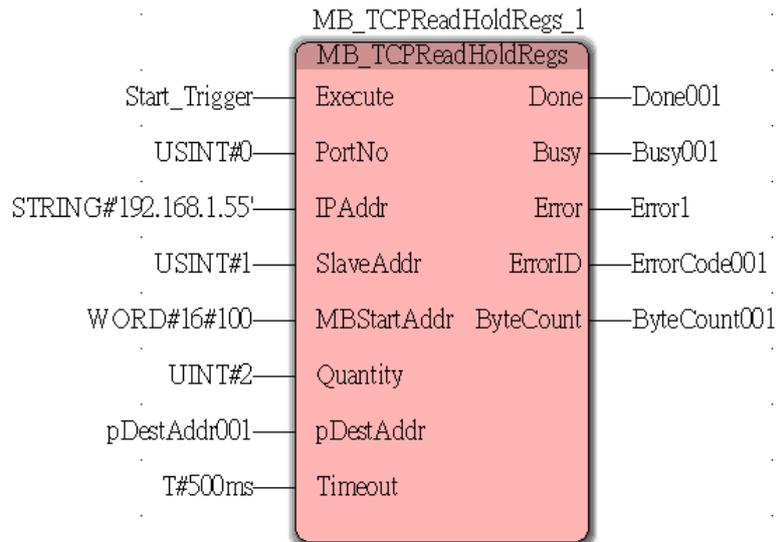


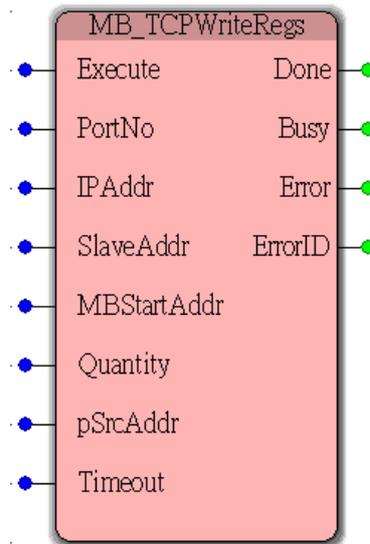
圖 5.2.5.17.3 MB\_TCPReadHoldRegs 參考範例說明

1. 當 Start\_Trigger(Execute)轉為 True 時·MB\_TCPReadHoldRegs 功能塊透過 Modbus 將外部 IP 位址為 192.168.1.55(IP Addr)且站號 1(SlaveAddr)裝置的 0x100(MBStartAddr)位址資料讀取 2 筆(Quantity)WORD 資料回傳至 USER\_ARRAY(pDestAddr)·並設定逾時時間為(Timeout)為 500 ms。
2. 完成輸出動作之後·Done001(Done)內容輸出為 True·ByteCount 變數讀取之 Byte 數量為 4。

# 5

## 5.2.5.18 MB\_TCPWriteRegs

- 類型  
Function Block
- 功能描述  
透過 EtherNet 傳輸將 ASDA-MS 記憶體資料，寫入至外部 Modbus 裝置
- 功能塊



註：使用 MB\_TCPWriteRegs 功能塊，因為寫入參數的時間有可能會超過 2 ms(系統執行 PLC Event Task 之週期時間)·若將此功能塊建置於 Event Task 可能會發生逾時情況·建議將此功能塊建置於 Cyclic Task。關於 Cyclic Task 詳細說明請參考 3.4 節。

### ■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時執行指令·將執行讀取外部Modbus動作。
PortNo	USINT	0 ~ 255 (0)	通訊埠編號*1
IP Addr	STRING	空字串	指定外部裝置IP位址
SlaveAddr	USINT	1 ~ 247 (0)	指定從站站號
MBStartAddr	WORD	0x0000 ~ 0xFFFF (0)	外部Modbus起始通訊位址
Quantity	UINT	1 ~ 123 (0)	讀取暫存器個數·單位：WORD
pDestAddr	ANY*2	0 ~ 65535 (0)	寫入外部裝置資料
Timeout	TIME*3	正數、T#0s (T#0s)	逾時設定

註：

1. ASDA-MS 目前僅有一個通訊埠·因此通訊埠編號固定為 0；使用者可由參數 P3-24 (EtherNet 網路設定)、P3-25 (EtherNet IP 位址設定)、P3-26 (EtherNet 子網路遮罩設定)、與 P3-27(EtherNet 預設閘道設定)來設定相關 EtherNet 參數設定。且使用 TCP 模式時·P3-02(通訊協定)目前支援 RTU 格式。
2. pDestAddr 資料型態為 ANY·可以根據需求來選擇 WORD、DWORD、LWORD 或是其他型態·亦可搭配自定義型態·自定義型態詳細說明請參考 3.6 節。
3. 輸入格式為 IEC-61131-3 的 TIME 格式。

■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	讀取參數完成時為True
Busy	BOOL	True / False (False)	指令被觸發執行時為True
Error	BOOL	True / False (False)	讀取參數不存在時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	當讀取到不存在的參數值時所記錄的錯誤碼，錯誤碼詳細說明請參考第 5.1.4 章節

■ 輸出變數變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Done	當讀取參數完成時	<ul style="list-style-type: none"> <li>■ 當Execute轉為False時</li> <li>■ 若Execute為False而Done轉為 True，此時Done維持一個掃描週期的 True 狀態後，立刻轉為False</li> </ul>
Busy	當 Execute 上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在ErrorID)	在Execute下緣時 (清除ErrorID紀錄之錯誤碼)

■ 輸出變數的變化時序

請參考圖 5.2.3.1 Sys\_ReadPara 時序圖

■ 參考範例

這個範例使用 MB\_TCPWriteRegs 功能塊來寫入資料至外部 Modbus 裝置。自定義型態 USER\_ARRAY 為大小 5 個 WORD 之陣列。

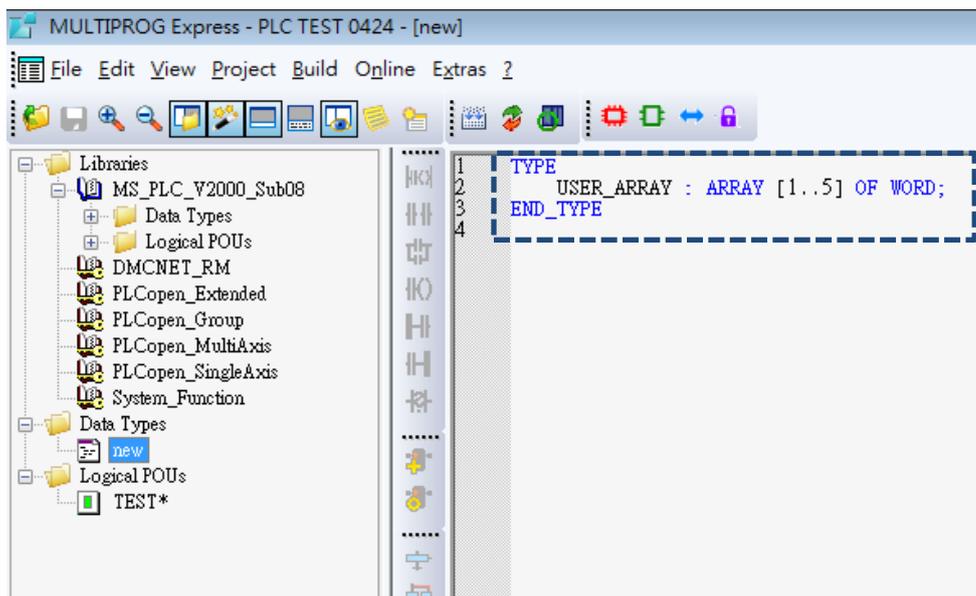


圖 5.2.5.18.1 自定義陣列

5

設定 pDestAddr 變數之資料型態為 USER\_ARRAY。

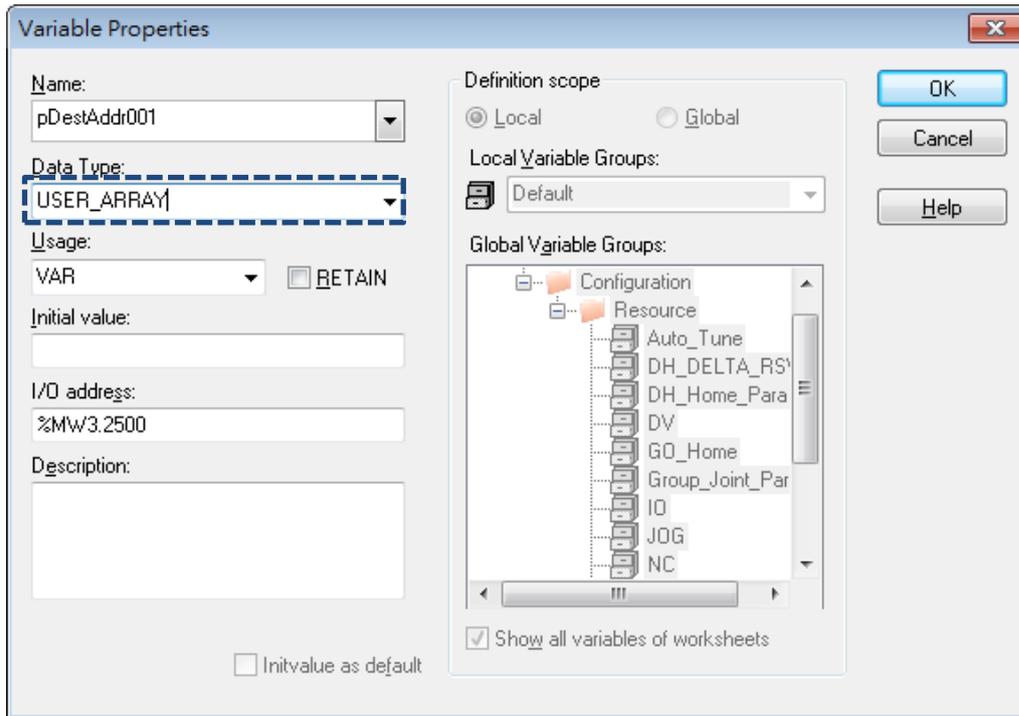


圖 5.2.5.18.2 自定義陣列設定

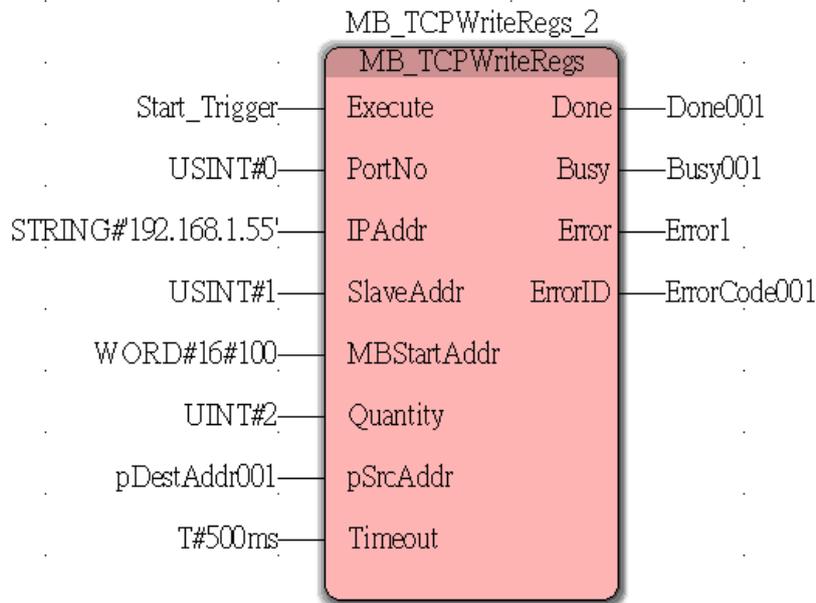
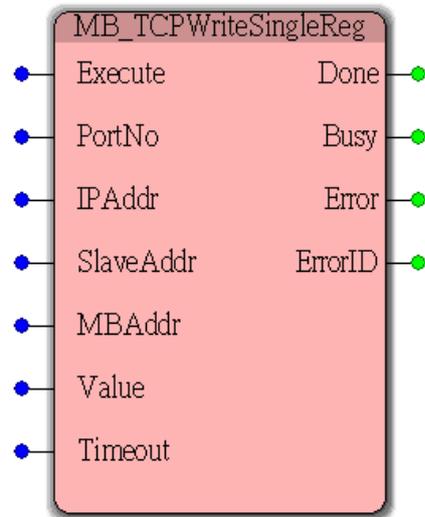


圖 5.2.5.18.3 MB\_TCPWriteRegs 使用參考範例

1. 當 Start\_Trigger(Execute)轉為 True 時，MB\_TCPWriteRegs 功能塊將 2 筆(Quantity)資料 USER\_ARRAY(pDestAddr)寫入 IP 位址為 192.168.1.55 (IP Adress)且站號為 1(SlaveAddr)的 0x100(MBStartAddr)位址，並設定逾時時間為(Timeout)為 500 ms。
2. 完成輸出動作之後，Done001(Done)內容輸出為 True。

### 5.2.5.19 MB\_TCPWriteSingleRegs

- 類型  
Function Block
- 功能描述  
透過 EtherNet 傳輸將一筆 ASDA-MS 記憶體資料，寫入至外部 Modbus 裝置
- 功能塊



註：使用 MB\_TCPWriteSingleRegs 功能塊，因為寫入參數的時間有可能會超過 2ms(系統執行 PLC Event Task 之週期時間)，若將此功能塊建置於 Event Task 可能會發生逾時情況，建議將此功能塊建置於 Cyclic Task。關於 Cyclic Task 詳細說明請參考第 3.4 節。

■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時執行指令，將執行讀取外部Modbus動作。
PortNo	USINT	0 ~ 255 (0)	通訊埠編號*1
IP Addr	STRING	空字串	指定外部裝置IP位址
SlaveAddr	USINT	1 ~ 247 (0)	指定從站站號
MAddr	WORD	0x0000 ~ 0xFFFF (0)	外部Modbus起始通訊位址
Value	WORD	0x0000 ~ 0xFFFF (0)	寫入外部裝置資料
Timeout	TIME*2	正數、T#0s (T#0s)	逾時設定

註：

1. ASDA-MS 目前僅有一個通訊埠，因此通訊埠編號固定為 0；使用者可由控制器參數 P3-24(EtherNet 網路設定)、P3-25(EtherNet IP 位址設定)、P3-26(EtherNet 子網路遮罩設定)、與 P3-27(EtherNet 預設閘道設定)來設定相關 EtherNet 參數設定。且使用 TCP 模式時，P3-02(通訊協定)目前支援 RTU 格式。
2. 輸入格式為 IEC-61131-3 的 TIME 格式。

# 5

■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	讀取參數完成時為True
Busy	BOOL	True / False (False)	指令被觸發執行時為True
Error	BOOL	True / False (False)	讀取參數不存在時為True
ErrorID	WORD	0x0000 ~ 0xFFFF (0)	當讀取到不存在的參數值時所記錄的錯誤碼，錯誤碼詳細說明請參考第 5.1.4 章節

■ 輸出變數變化時序

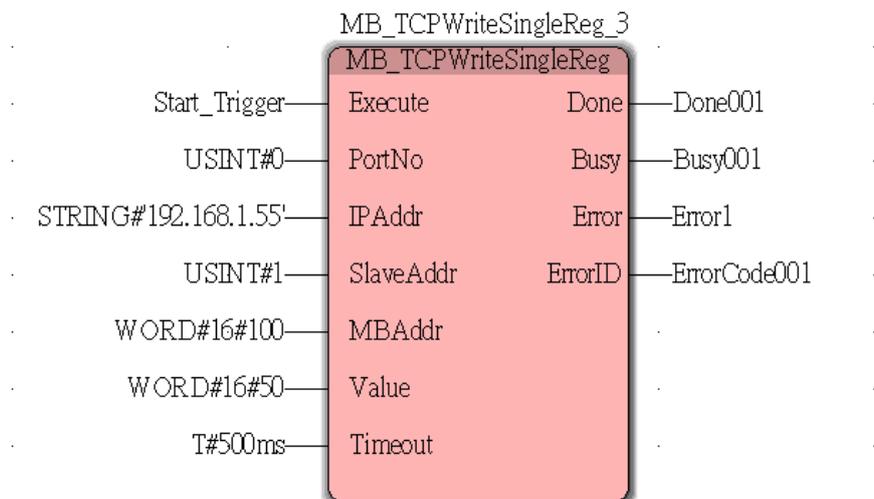
名稱	輸出變數上緣時機	輸出變數下緣時機
Done	讀取參數完成時	<ul style="list-style-type: none"> <li>■ 當Execute轉為False時</li> <li>■ 若Execute為False，而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Busy	當Execute上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
Error / ErrorID	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在ErrorID)	在Execute下緣時 (清除ErrorID紀錄之錯誤碼)

■ 輸出變數的變化時序

請參考圖 5.2.3.1 Sys\_ReadPara 時序圖

■ 參考範例

此範例使用 MB\_TCPWriteSingleRegs 功能塊，以寫入單筆資料至外部 Modbus 裝置。



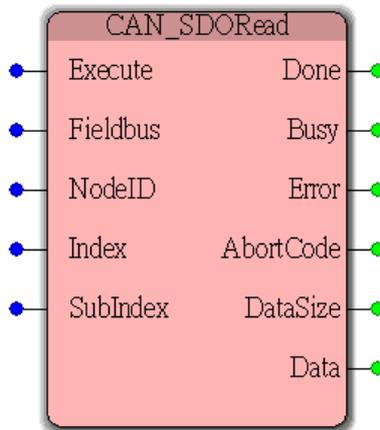
1. 當 Start\_Trigger(Execute)轉為 True 時，MB\_TCPWriteSingleRegs 功能塊將一筆資料 0x50(Value)寫入外部 IP 位址為 192.168.1.55(IP Addr)且站號 1(SlaveAddr)裝置的

0x100(MBStartAddr)位址並設定逾時時間為(Timeout)為 500 ms。

2. 完成輸出動作之後，Done001(Done)內容輸出為 True。

### 5.2.5.20 CAN\_SDORead

- 類型  
Function Block
- 功能描述  
讀取 CANopen 擴充模組參數。
- 功能塊



註：使用 CAN\_SDORead 功能塊，因為讀取參數的時間有可能會超過 2 ms (系統執行 PLC Event Task 之週期時間)，若將此功能塊建置於 Event Task 可能會發生逾時情況，建議將此功能塊建置於 Cyclic Task。關於 Cyclic Task 詳細說明請參考第 3.4 節。

#### ■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時執行指令，將指定的參數輸出至Data*1
Fieldbus	UINT	0 ~ 65535 (0)	通訊類型*2
NodeID	UINT	0 ~ 65535 (0)	節點編號
Index	UINT	0 ~ 65535 (0)	參數群組編號*3
SubIndex	USINT	0 ~ 255 (0)	參數索引編號*3

註：

1. 當 Execute 上緣時執行指令，將指定的參數輸出至 Data。當 Execute 下緣時執行指令，將輸出 Data 資料清除為 0。
2. 擴充模組的通訊類型目前僅支 DMCNe 模式，請設為 0。
3. 擴充模組其 SD 命令定義及說明，請參考各模組操作手冊。

## 5

## ■ 輸出變數

名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	當讀取參數完成時為 True
Busy	BOOL	True / False (False)	當指令被觸發執行時為 True
Error	BOOL	True / False (False)	當讀取參數不存在時為 True
AbortCode	DWORD	0x0000~0xFFFF (0)	當讀取參數不存在時記錄錯誤碼*1
DataSize	INT	-32768 ~ 32767	讀取的資料長度*2
Data	UDINT	0 ~ 4294967295	讀取及寫入的資料

註：

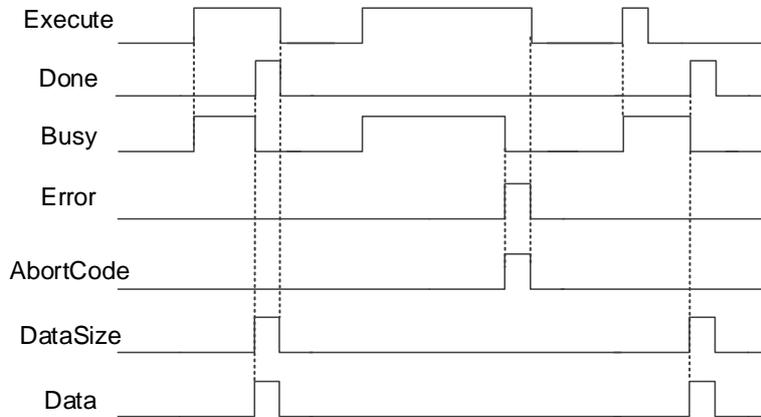
1. 當所下達的 SDO OD 碼不支援或是資料格式錯誤的時候，AbortCode 變數將回傳 0x0602 0000 數值。
2. DataSize 參數代碼定義如下：
  - 1：8-bit
  - 2：16-bit
  - 4：32-bit

## ■ 輸出變數變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Done	當讀取參數完成時	<ul style="list-style-type: none"> <li>■ 當Execute轉為False時</li> <li>■ 若Execute為False，而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Busy	當 Execute 上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 在CommandAborted上緣時</li> </ul>
Error / AbortCode	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在AbortCode)	當Execute下緣時 (清除AbortCode 紀錄之錯誤碼)
DataSize	當讀取參數完成時	在Execute下緣時
Data	當讀取參數完成時	在Execute下緣時

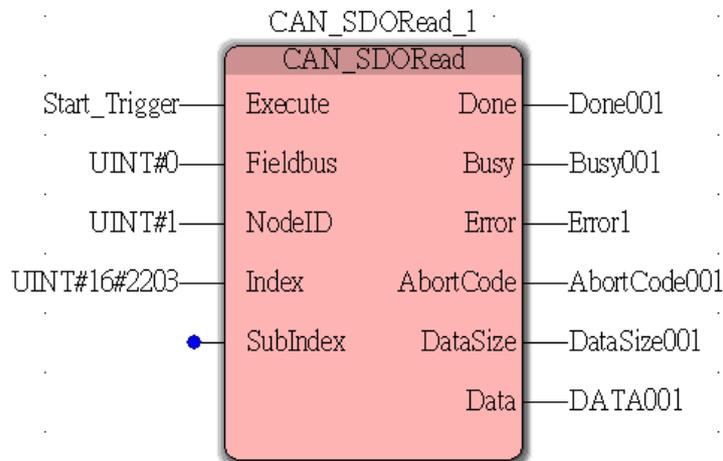
■ 輸出變數的變化時序

此輸出變數說明了三種情況：第一部分是完成功能塊並且獲得輸出數值；第二部分情況為發生錯誤，因此功能塊中斷並且無獲得正確的數值；第三部分則說明了 Execute 啟動後並無等待獲得數值後才釋放，而功能塊依舊完成整個程序才停止。



■ 參考範例

此範例使用 CAN\_SDORead 功能塊，來讀取擴充模組 ASDA-A2-F 參數。

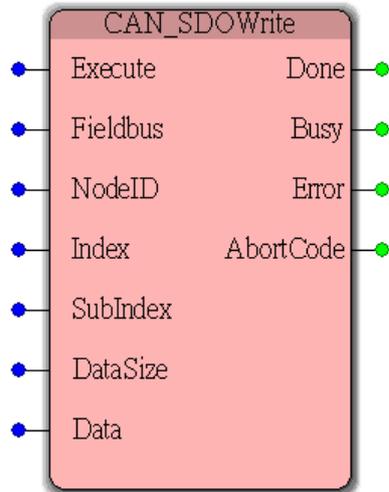


1. 當 Start\_Trigger(Execute)轉為 True 時，CAN\_SDORead 將透過 DMCNet(Fieldbus) 將從站站號 1(NodeID)參數 P2-03(Index)資料讀取至 DATA001(Data)，並將讀取的資料長度儲存於 DataSize001(DataSize)。
2. 完成輸出動作之後，Done001(Done)內容輸出為 True。

# 5

## 5.2.5.21 CAN\_SDOWrite

- 類型  
Function Block
- 功能描述  
擴充模組參數寫入。
- 功能塊



註：使用 CAN\_SDOWrite 功能塊，因為寫入參數的時間有可能會超過 2 ms(系統執行 PLC Event Task 之週期時間)，若將此功能塊建置於 Event Task 可能會發生逾時情況，建議將此功能塊建置於 Cyclic Task。關於 Cyclic Task 詳細說明請參考第 3.4 節

### ■ 輸入變數

名稱	資料型態	設定值 (預設值)	功能
Execute	BOOL	True / False (False)	Execute上緣時執行指令，將指定的參數輸出至Data
Filebus	UINT	0 ~ 65535 (0)	通訊類型*1
NotelD	UINT	0 ~ 65535 (0)	節點編號
Index	UINT	0 ~ 65535 (0)	參數群組編號*2
SubIndex	USINT	0 ~ 255 (0)	參數索引編號*2
DataSize	INT	-32768 ~ 32767	寫入的資料長度*3
Data	UDINT	0 ~ 4294967295	寫入的資料

註：

1. 擴充模組的通訊類型目前僅支援 DMCNet 模式，請設為 0。
2. 擴充模組其 SDO 命令定義及說明，請參考各模組操作手冊。
3. DataSize 參數代碼定義如下：
  - 1：8-bit
  - 2：16-bit
  - 4：32-bit

■ 輸出變數

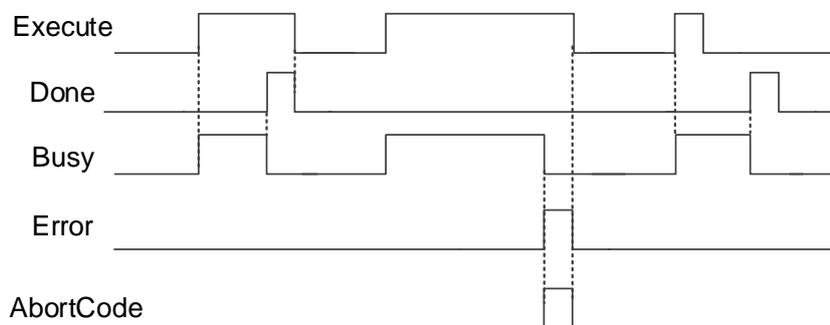
名稱	資料型態	設定值 (預設值)	功能
Done	BOOL	True / False (False)	存取的參數完成時為True
Busy	BOOL	True / False (False)	指令被觸發執行時為True
Error	BOOL	True / False (False)	存取的參數不存在時為True
AbortCode	DWORD	0x0000 ~ 0xFFFF (0)	當存取到不存在的參數值時所記錄的錯誤碼；當所下達的 SDO OD 碼不支援或是資料格式錯誤的時候，AbortCode變數將回傳 0x0602 0000 數值。

■ 輸出變數變化時序

名稱	輸出變數上緣時機	輸出變數下緣時機
Done	讀取參數完成時	<ul style="list-style-type: none"> <li>■ 當Execute轉為False時</li> <li>■ 若Execute為False，而Done轉為True，此時Done維持一個掃描週期的True狀態後，立刻轉為False</li> </ul>
Busy	Execute上緣觸發時	<ul style="list-style-type: none"> <li>■ 在Done上緣時</li> <li>■ 在Error上緣時</li> <li>■ 當CommandAborted上緣時</li> </ul>
Error / AbortCode	當指令的執行條件或輸入值發生錯誤時(錯誤碼紀錄在AbortCode)	當Execute下緣時(清除AbortCode 紀錄之錯誤碼)

■ 輸出變數時序

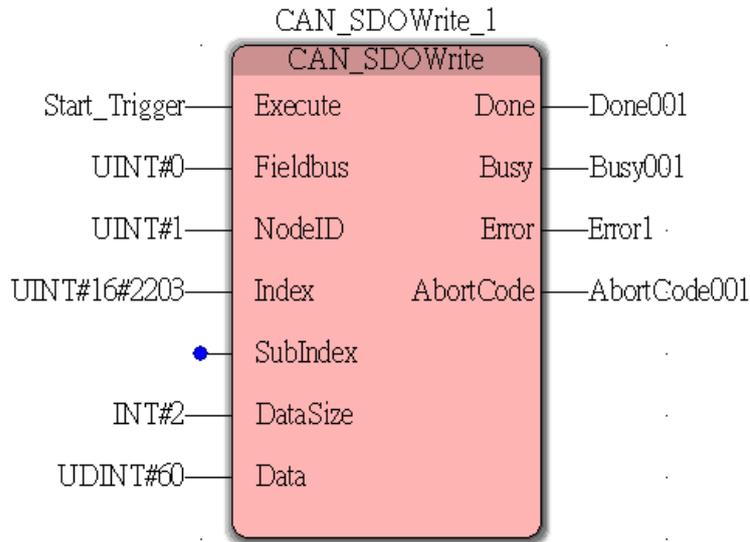
此輸出變數說明了三種情況：第一部分是完成功能塊並且獲得輸出數值；第二部分的情況為發生錯誤，因此功能塊中斷並且無獲得正確的數值；第三部分則說明了Execute 啟動後並無等待獲得數值後才釋放，而功能塊依舊完成整個程序才停止。



# 5

■ 參考範例

此範例使用 CAN\_SDOWrite 功能塊將資料寫入至擴充模組。



1. 當 Start\_Trigger(Execute)轉為 True 時，CAN\_SDOWrite 透過 Modbus(Fieldbus)16 bits(DataSize)資料 60(Data)寫入從站站號 1 的參數 P2-03。
2. 完成輸出動作之後，Done001(Done)內容輸出為 True

# 6

## 建立專案

---

此章節將介紹如何建立專案、開發功能塊和建立資料庫等作法，並且將會介紹如何產生更新檔。

6.1	建立功能塊.....	6-2
6.2	建立函式庫.....	6-4
6.3	建立更新檔( <code>bootfile.img</code> ).....	6-6

# 6

## 6.1 建立功能塊

在第二章節已介紹過 MULTIPROG 的畫面與操作介面，在功能塊函式庫區間可以查詢到所有可以使用的功能塊，這些功能塊可分為兩大部分：第一部分為 MULTIPROG 所提供的基本功能塊，第二部分是本公司依照 PLCopen 組織所規劃的設計架構所開發的功能塊。而 MULTIPROG 提供了二次開發系統，可供使用者製作功能塊並重複使用。

在第四章節中已有介紹如何透過 IEC61131-3 所規範的五種語言建立功能塊，而本文將介紹新建立的功能塊建立後，可以在何處找到並且使用。

(1) 依照 4.1 章節 指令表(IL)建立一簡易公式

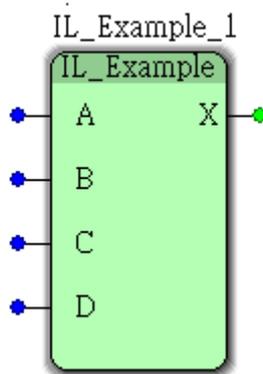


圖 6.1.1 IL 範例功能塊

(2) 透過工作視窗可在右方找到 Edit Wizard 資料庫，選擇<all FUs and FBs>。

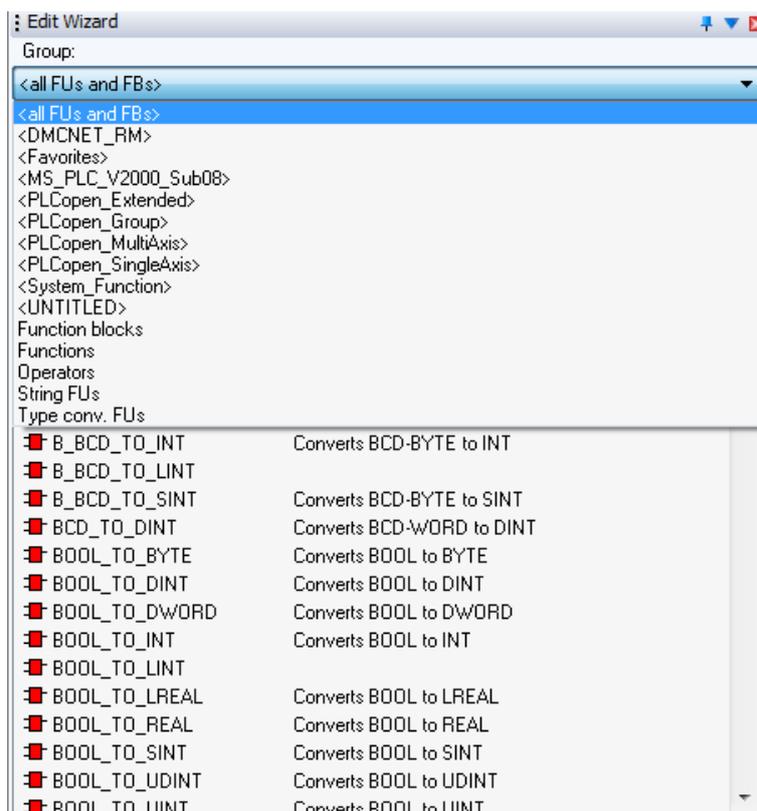


圖 6.1.2 功能塊函式庫

(3) 在<all FUs and FBs>資料庫中，可找尋到所創立的 IL\_Example 功能塊。

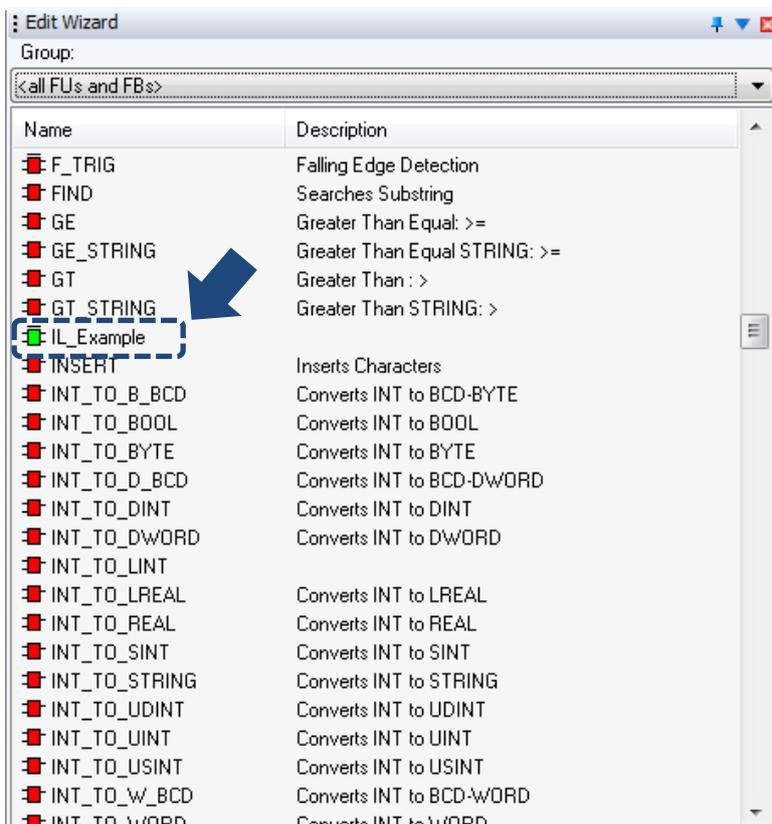


圖 6.1.3 功能塊函式庫

# 6

## 6.2 建立函式庫

在功能開發期間，MULTIPROG 可以建立一函式庫提供給其餘開發者使用，以下將介紹如何透過 MULTIPROG 來建立一函式庫。

1. 建立一專案並且完成編譯，將可獲得一專案.mwe 檔案以及內容檔案資料夾。

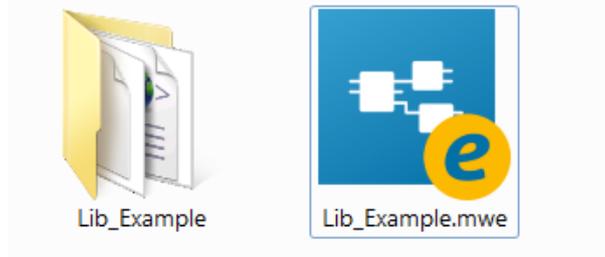


圖 6.2.1 專案資料夾及執行檔

2. 在欲被加入的專案中，在左方專案樹中的 Libraries 點擊滑鼠右鍵，即會彈出可加入 Library 的選項。

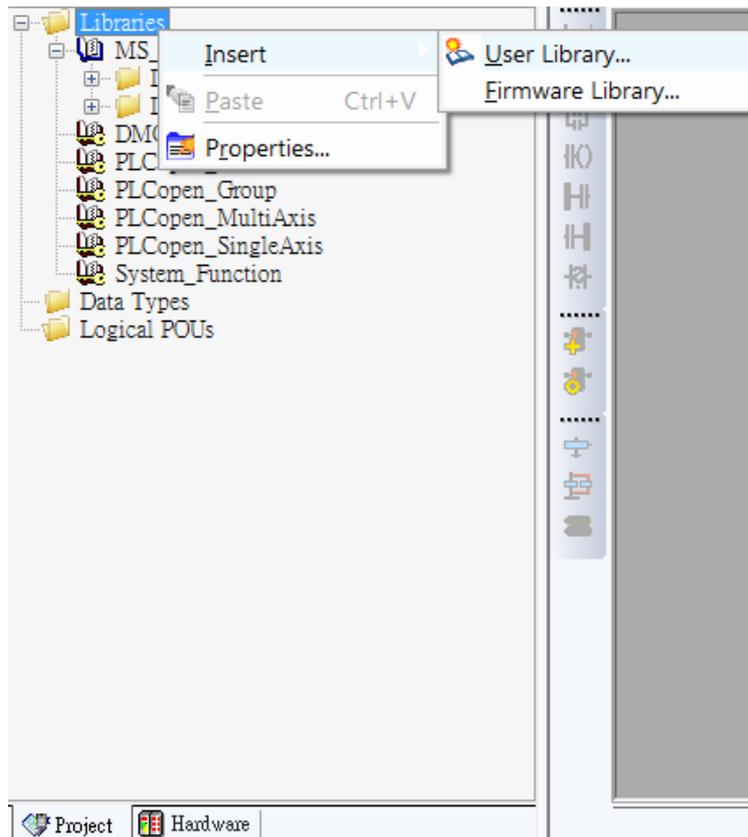


圖 6.2.2 新增 User Library

3. 點擊 User Library，MULTIPROG 會跳出 Include library 視窗，供使用者選擇函式庫，並且尋找到步驟一所建立好的專案。

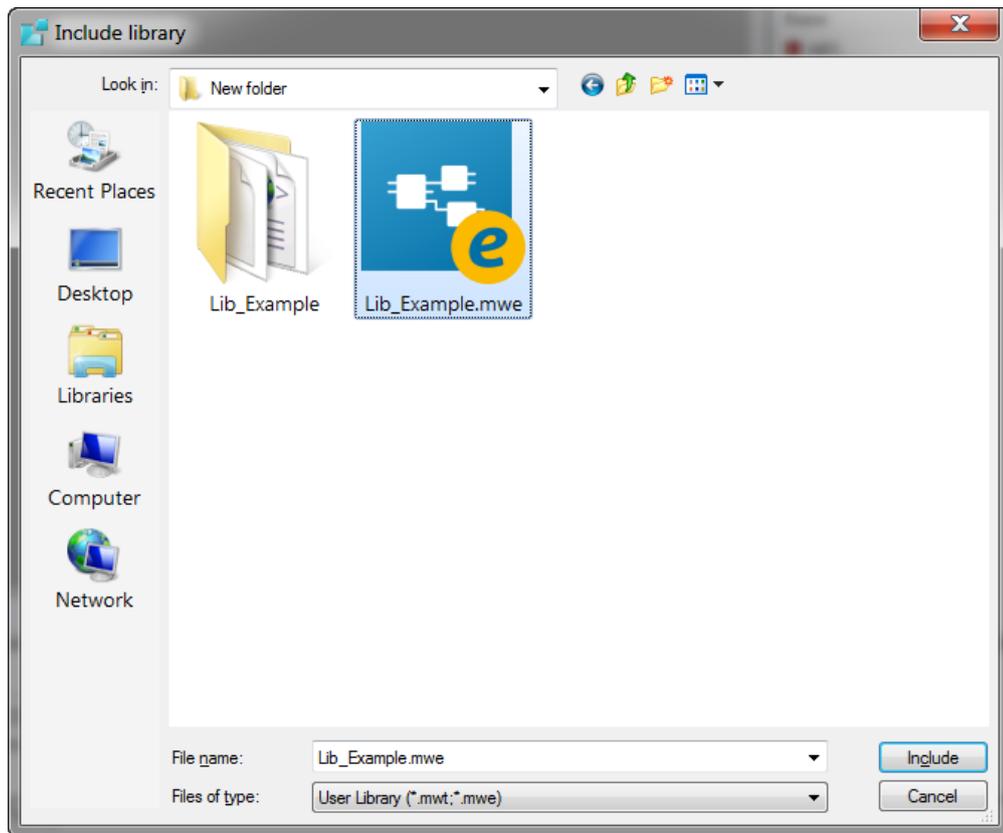


圖 6.2.3 選擇執行檔

4. 選擇 Lib\_Example.mwe。
5. 在專案樹中的 Libraries 的細項，可看到已加入的 Lib\_Example 資料庫已被納入。

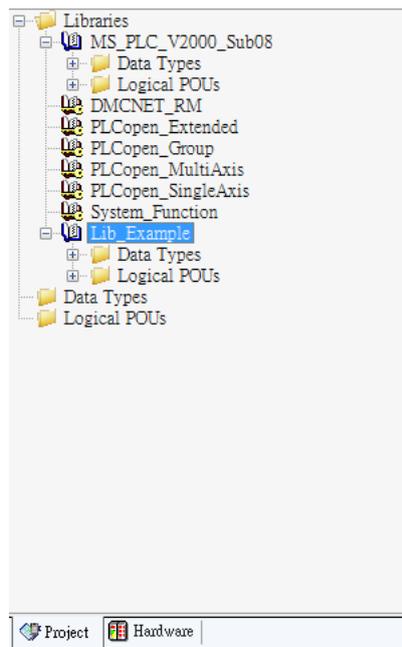


圖 6.2.2 User Library 已新增到 Libraries

# 6

## 6.3 建立更新檔(bootfile.img)

當使用者完成一專案時，須將此專案轉成一更新檔提供給終端使用者，您可透過以下步驟來完成。

- (1) 切換專案樹至[Hardware]書籤頁上。

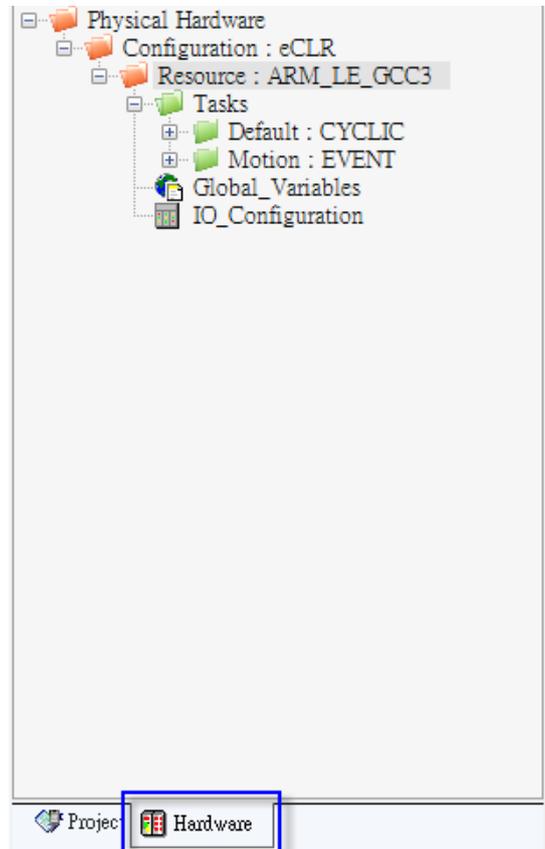


圖 6.3.1 專案樹 Resource 路徑

- (2) 在[Resource : ARM\_LE\_GCC3]的圖示上按下滑鼠右鍵，並在彈出的選單中點選 [Create Bootproject...]

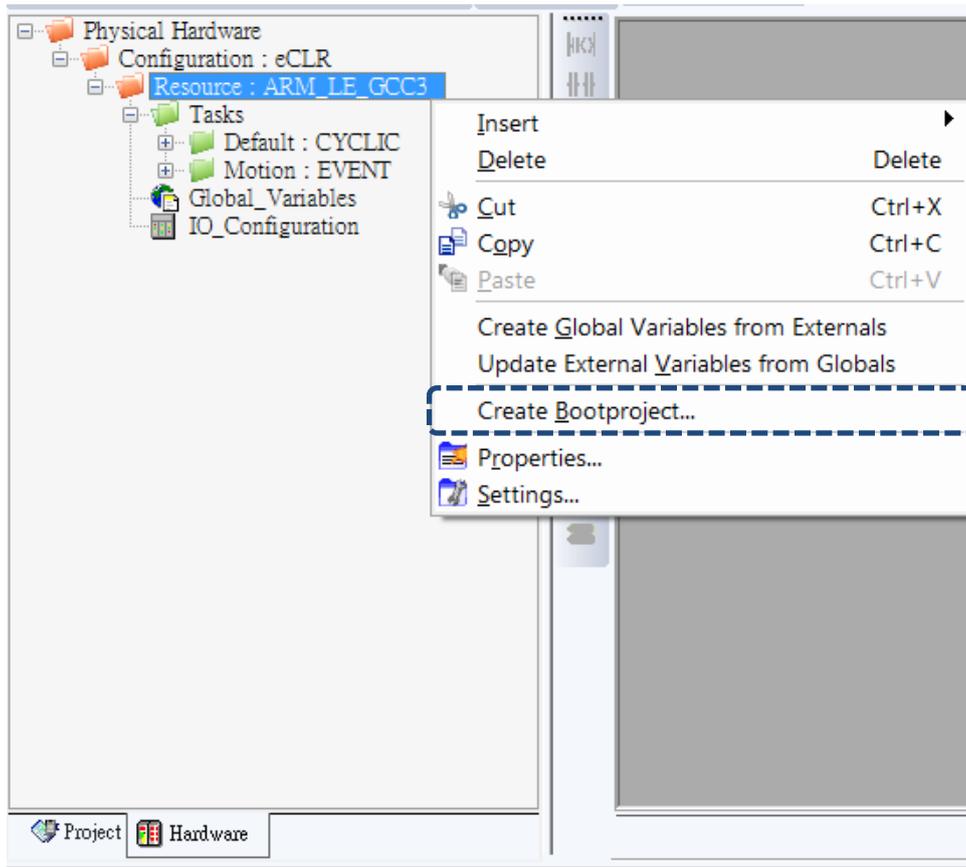


圖 6.3.2 建立 Bootfile.img 路徑

- (3) MULTIPROG 將會彈出 Bootproject Options 選項視窗，此視窗將詢問您是否需要在製作 bootfile.img 時包含原始程式。(一般不需要包含原始程式，若選擇有包含原始程式，製作出的 bootfile.img 檔案將會較大。) 選擇完成後，請按 **OK** 鍵。

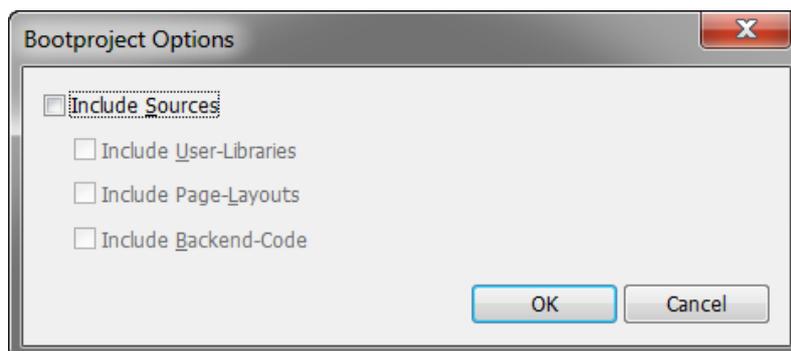


圖 6.3.3 Bootproject Options

6

- (4) 完成步驟四時，MULTIPROG 將產生一檔案在下圖路徑上。(若無法找到相對應資料夾，可透過作業系統資料夾管理來開啟隱藏資料夾。)

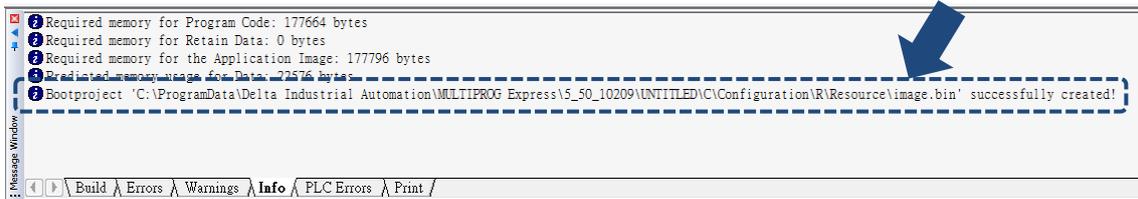


圖 6.3.4 顯示區 Bootproject 路徑

- (5) 在步驟五所產生的路徑上，尋找 image.bin 檔案。

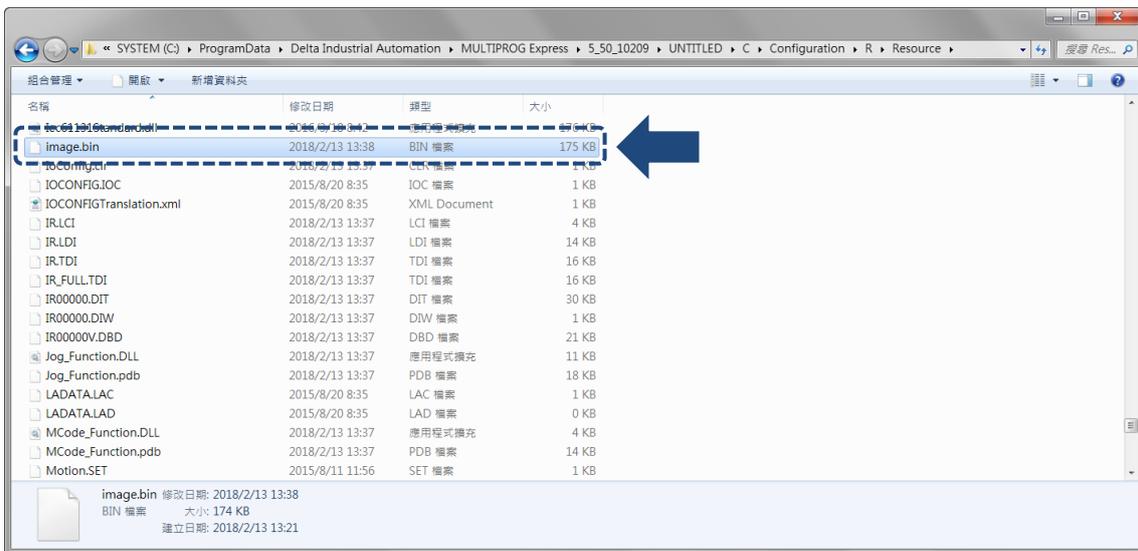


圖 6.3.5 Bootproject img.bin 資料夾

- (6) 將所尋找到的 image.bin 檔案名稱修改成 bootfile.img，即可完成更新檔製作。(由於 MS 驅控器在更新 PLC 檔案時，只支援 bootfile.img 檔案名稱，因此務必將此檔案更改至正確名稱。)

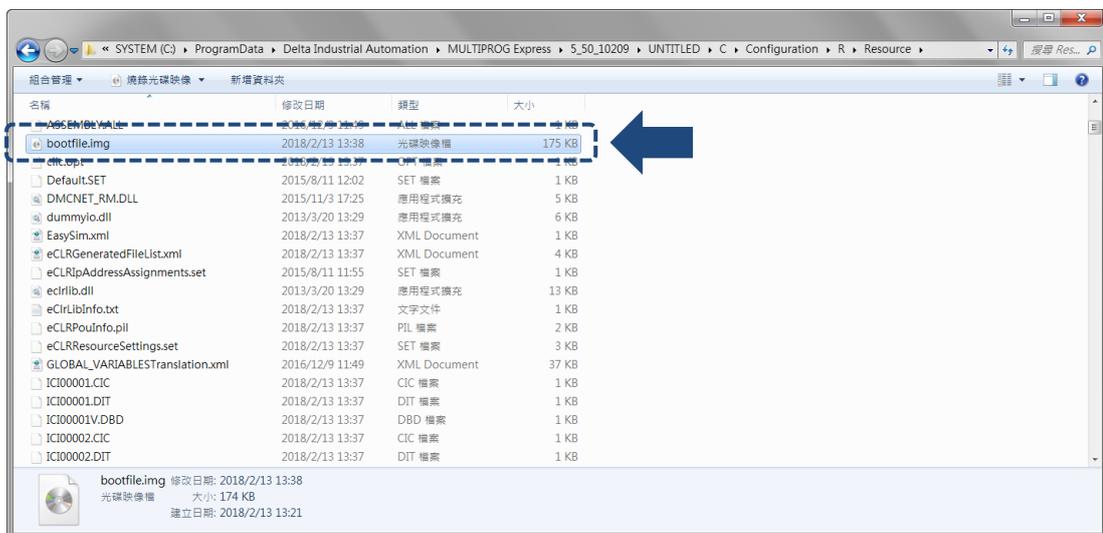


圖 6.3.6 將 image.bin 改名為 bootfile.img