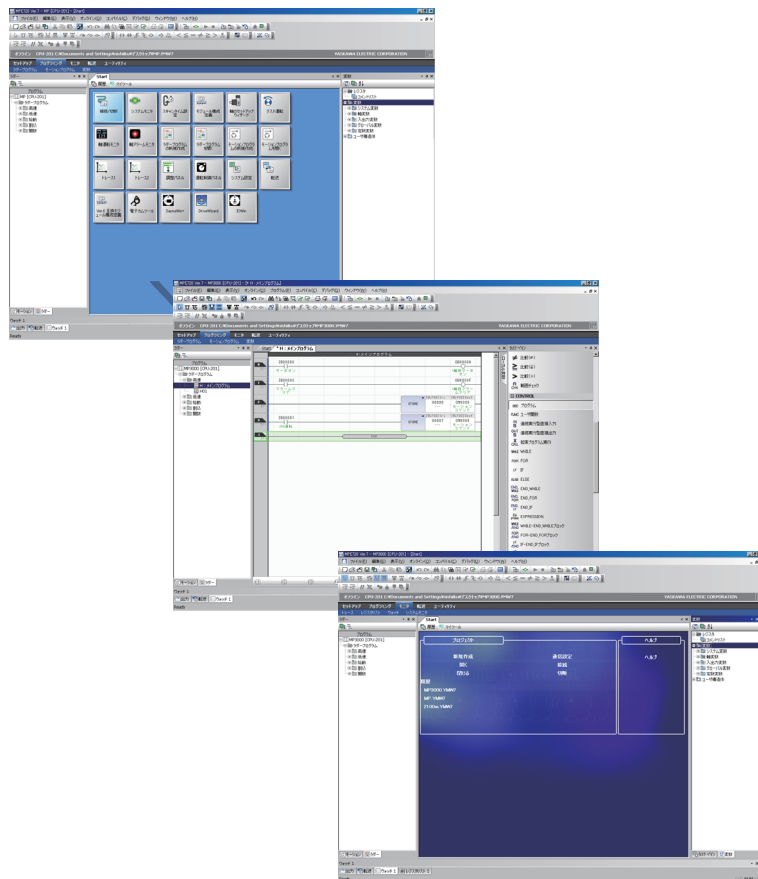


## 運動控制器 MP3000系列 階梯圖程式 程式設計手冊



階梯圖程式的特色及概要

1

階梯圖程式的開發流程

2

暫存器

3

階梯語言指令

4

工程工具(MPE720)  
功能介紹

5

系統服務暫存器一覽

附錄A

程式範例

附錄B

EXPRESSION  
指令的格式

附錄C

運動參數的相關注意事項

附錄D

運動控制器的規格一覽

附錄E





## 前言

本手冊將就運動控制器 MP3000 系列的階梯圖程式相關資訊進行說明。

請認真閱讀本手冊，以便您正確地使用運動控制器系統，並將運動控制器充分應用到貴公司製造系統的控制中。

並且，請妥善保管好本手冊，以便需要時隨時取閱和參考。

## 手冊的使用方法

### ◆ 縮略語及縮寫符號

本手冊使用如下所示縮略語和縮寫符號。

縮略語及縮寫符號	含義
運動控制器	MP3000 系列 運動控制器
MPE720	編程裝置用軟體或安裝好該軟體之編程裝置（個人電腦）
PLC	可編程控制器
MP3200	電源單元、CPU 單元、基本單元、固定架擴充 I/F 單元的總稱
MP3300	CPU 模組、基本單元之總稱

### ◆ 本手冊中使用的工程工具

本手冊中對使用 MPE720 Ver.7 的畫面進行說明。

### ◆ 反轉訊號名的標識

在本手冊的正文中，反轉訊號名（L Level 內有效的訊號）透過在訊號名前加（/）來表示。

書寫例

- $\overline{S-ON} = /S-ON$
- $\overline{P-CON} = /P-CON$

### ◆ 關於用語「轉矩」

旋轉型伺服馬達使用「轉矩」，線性伺服馬達使用「推力」，但本手冊中均統一使用「轉矩」表述（參數名稱除外）。

### ◆ 註冊商標等

- MECHATROLINK 為 MECHATROLINK 協會的商標。
- Ethernet 為 Xerox 公司的註冊商標。
- 手冊中提到的其他產品名稱、公司名稱等固有名詞是各公司的商標、註冊商標或商品名稱。本文中未顯示各公司的註冊商標或商標、TM、® 標誌。

## ◆ 圖示的標注

為使讀者瞭解說明內容的區分，本書中設計了如下圖示。並在必要的地方使用這些圖示。



表示必須遵守的注意事項及限制事項。  
同時也表示發出警報，但還不至於造成裝置損壞的注意事項。



表示提醒注意，記載了防止錯誤操作的注釋。

例

表示操作或設定範例等。

補充

表示補充事項或瞭解後有助於使用的資訊。



對難於理解的用語進行解釋，並對事先沒有說明而後出現的用語進行說明。

## 相關手冊

下表展示相關之手冊。請根據需要參照。

請於充分了解產品之規格、使用限制等條件後，再使用產品。

分類	資料名稱	資料編號	內容
基本功能	運動控制器 MP2000/MP3000 系列 運動控制器系統 安裝手冊	SIJP C880725 00	對 MP2000/MP3000 系列運動控制器的安裝與連接、設定、試運轉、程式設計與調試及各功能進行說明。
	運動控制器 MP3000 系列 MP3200/MP3300 故障排除手冊	SIJP C880725 01	說明 MP3000 系列 運動控制器的故障排除。
	運動控制器 MP3000 系列 MP3200 用戶手冊	SIJP C880725 10	說明有關 MP3000 系列 MP3200 之規格、系統架構及 CPU 單元之功能。
	運動控制器 MP3000 系列 MP3300 產品手冊	SIJP C880725 21	說明有關 MP3000 系列 MP3300 之規格、系統架構及 CPU 單元之功能。
通訊功能	運動控制器 MP3000 系列 通訊功能 用戶手冊	SIJP C880725 12	對 MP3000 系列的乙太網路通訊規格、系統構成及通訊連接方法進行詳細說明。
運動控制功能	運動控制器 MP3000 系列 運動控制功能 用戶手冊	SIJP C880725 11	說明 MP3000 系列之運動控制功能 (SVC32/SVR32) 之規格、系統架構與使用方法。
程式	運動控制器 MP3000 系列 運動程式 程式設計手冊	SIJP C880725 14	對 MP3000 系列的運動程式、序列程式的規格和指令進行說明。
	運動控制器 MP2000 系列 用戶手冊 階梯圖程式篇	SI-C887-1.2	對 MP2000 系列的階梯圖程式中使用的運算指令進行詳細說明。
工程工具	運動控制器 MP2000/MP3000 系列 系統整合工程工具 MPE720 Ver.7 用戶手冊	SIJP C880761 03	對 MPE720 Ver.7 的操作方法進行說明。

## 安全注意事項


本資料中有關安全的內容，使用如下標記進行說明。

有關安全標記的說明，表示重要內容，請務必遵守。




表示錯誤使用時，將會引發危險情況，導致人身傷亡。




表示錯誤使用時，將會引發危險情況，導致輕度或中度人身傷害，損壞設備。  
另外，即使是  中記載的事項，根據具體情況，有時也可能導致重大事故。



表示禁止（絕對不能做）。例如嚴禁煙火時，則表示為 。



表示強制（必須做）。例如接地時，則表示為 。

本節記敘了保管、搬運、安裝、配線、運轉、維護、檢查、廢棄時必須遵守的重要事項。

### ◆ 常規



- 請由操作熟練的技術人員正確安裝。  
否則會導致觸電或受傷。
- 與機械連接後開始運轉時，請使設備處於可隨時緊急停止的狀態。  
否則會導致受傷。
- 如果在運轉過程中發生瞬時停電後又恢復供電，機械可能會突然再起動，因此切勿靠近機器。請採取措施以確保重啟時不會危及到人身安全。  
否則會導致受傷。
- 請絕對不要觸摸產品的內部。  
否則會導致觸電。
- 因此，請勿在通電狀態下拆下外罩、電纜、連接器以及選購配件。  
否則會導致觸電、故障、損壞。
- 請勿損傷或用力拖拉電纜，勿使電纜過度受力，勿在電纜上吊掛重物，或被箱蓋、櫃門等夾住。  
否則會導致觸電、產品停止動作或燒壞。
- 請絕對不要對本產品進行改造。  
否則會導致受傷、機械損壞。

## ◆ 保管、搬運



- 請在如下環境中進行保管。
  - 無陽光直射的場所
  - 環境溫度不超過保管溫度條件的場所
  - 相對濕度不超過保管濕度條件的場所
  - 溫差小、不結露的場所
  - 無腐蝕性氣體、可燃性氣體的場所
  - 塵土、灰塵、鹽分及金屬粉末較少的場所
  - 不易濺上水、油及藥品等的場所
  - 震動或衝擊不會波及產品的場所否則會導致火災、觸電或機器損壞。
- 搬運時，請務必握持產品主體。  
僅握持電纜或連接器搬運時，可能會導致連接器損壞或電纜斷線、人員受傷。
- 請勿過多地將本產品堆積在一起（請根據指示）。  
否則會導致受傷或故障。
- 運輸的所有場合，都不得使產品暴露在含鹵素（氟、氯、溴、碘等）的環境中。  
否則會導致故障或損壞。
- 包裝用木質材料（含木框、三合板、貨架等）需要進行消毒、除蟲處理時，請務必採用薰蒸以外的方法。  
例：熱處理（材芯溫度 56°C 以上，處理時間 30 分鐘以上）  
此外，必須在包裝前的材料階段進行處理，而不是在包裝後進行整體處理。  
使用經過薰蒸處理的木質材料包裝電氣產品（單機或裝載在機械等上的產品）時，該木質材料產生的氣體和蒸汽會對電子零組件造成致命的損傷。特別是鹵素類消毒劑（氟、氯、溴、碘等）可能會導致電容器內部腐蝕。

## ◆ 安裝



### 注意

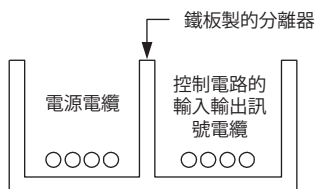
- 請在如下環境中進行安裝。
  - 無陽光直射的場所
  - 環境溫度不超過安裝溫度條件的場所
  - 相對濕度不超過安裝濕度條件的場所
  - 溫差小、不結露的場所
  - 無腐蝕性氣體、可燃性氣體的場所
  - 塵土、灰塵、鹽分及金屬粉末較少的場所
  - 不易濺上水、油及藥品等的場所
  - 震動或衝擊不會波及產品的場所否則會導致火災、觸電或機器損壞。
- 所有安裝方式，都不得使產品暴露在含鹵素（氟、氯、溴、碘等）的環境中。否則會導致故障或損壞。
- 請勿坐在本產品上或者在其上面放置重物。否則會導致受傷或故障。
- 請勿堵塞吸氣口與排氣口。也不要使產品內部進入異物。否則會因內部元件老化而導致故障或火災。
- 請務必遵守安裝方向的要求。否則會導致故障。
- 安裝時，請確保產品與控制盤內表面以及其他機器之間具有規定的間隔。否則會導致火災或故障。
- 請勿施加過大衝擊。否則會導致故障。
- 請由操作熟練的技術人員正確安裝電池。否則會導致觸電、受傷、機械損壞。
- 請勿接觸電池的電極部分。否則可能破壞靜電。

## ◆ 配線

### ⚠ 注意

- 請正確、可靠地進行配線。  
否則可能導致馬達暴衝、人員受傷或機器故障。
- 請使用指定的電源電壓。  
否則會導致火災或故障。
- 在電源狀況不良的情況下使用時，請確保可在指定的電壓變動範圍內供給輸入電源。  
否則可能導致機器損壞。
- 請安裝斷路器等安全裝置以防止外部配線短路。  
否則會引發火災。
- 在以下場所使用時，請採取適當的遮罩措施。
  - 因靜電而產生雜訊時
  - 產生強電場或強磁場的場所
  - 可能遭放射線輻射的場所
  - 附近有電源線的場所否則可能導致機器損壞。
- 設計電路時，請確保 CPU 單元 / CPU 模組先於輸入輸出用 24V 電源通電。關於電路的詳情，請參照以下手冊。
  - 📖 MP3000 系列 CPU 單元 使用說明書（資料編號：TOBP C880725 16）
  - MP3000 系列 MP3300 CPU 模組 使用說明書（資料編號：TOBP C880725 23）輸入輸出用 24V 電源等外部電源通電後，CPU 單元 / CPU 模組再通電時，CPU 單元 / CPU 模組的輸出可能會瞬間 ON，從而會導致非預期動作引起的人員受傷或設備損壞。
- 請務必在產品外部的控制電路中設定安全保護相關的緊急停止電路、互鎖電路及限位電路。  
否則會導致受傷、機械損壞。
- 使用 MECHATROLINK 輸入輸出模組時，請將已建立 MECHATROLINK 通訊作為互鎖輸出的條件。  
否則可能導致機器損壞。
- 請按照正確的極性連接電池。  
否則會導致電池損壞或爆炸。
- 請考慮下列因素，選擇連接產品與週邊設備的輸入輸出訊號線（外部配線）。
  - 機械強度
  - 雜訊的影響
  - 配線距離
  - 訊號電壓
- 為了抑制來自電源電纜的雜訊影響，無論在控制盤的內部或外部，控制電路輸入輸出訊號電纜的配線和走線均應與電源電纜分離。  
若分離不充分，可能會導致錯誤動作。

配線分離例



## ◆ 運轉

### 注意

- 請按與產品相應的用戶手冊中說明的步驟、指示進行運轉及試運轉。  
在伺服馬達和機械連接的狀態下，如果發生操作錯誤，則不僅會造成機械損壞，有時還可能導致人員傷亡。
- 請在產品外部設定互鎖訊號等安全電路，以便即使發生如下狀況，也能確保系統整體的安全。
  - 出現因產品故障或外部原因導致異常的狀態
  - 產品透過診斷功能檢測到異常，停止運轉，關閉（或保持）輸出訊號的狀態
  - 因輸出繼電器熔敷、燒損或輸出電晶體破損導致產品輸出一直保持 ON 或 OFF 的狀態
  - 產品的 DC24 V 輸出因超載狀態或短路導致電壓過低，無法輸出訊號的狀態
  - 產品自診斷功能無法檢測的電源部、輸入輸出部或記憶體等異常導致非預期輸出的狀態  
可能會導致受傷、設備損壞、燒損。

## ◆ 維護·檢查

### 注意

- 請勿拆卸、修理產品。  
會導致觸電、受傷、機械破損。
- 通電過程中請勿變更配線。  
會導致觸電、受傷、機械破損。
- 請由操作熟練的技術人員正確更換電池。  
否則會導致觸電、受傷、機械損壞。
- 請務必在 MP3□00 通電的狀態下更換電池。  
在 MP3□00 斷電的狀態下更換電池時，MP3□00 的記憶體中儲存的資料可能會丟失。
- 更換電池時，請勿觸摸電極部分。  
否則可能破壞靜電。
- 更換 CPU 單元 / CPU 模組時，請勿遺漏以下操作。
  - 請對需更換的 CPU 單元 / CPU 模組的程式及參數進行備份。
  - 請將已儲存的程式及參數傳輸至新 CPU 單元 / CPU 模組中。  
未傳輸資料的狀態下即開始運轉新 CPU 單元 / CPU 模組時，可能會因非預期動作導致受傷、設備損壞。
- 請勿於通電中或中斷電源後一段時間內，觸摸 CPU 單元 / CPU 模組的散熱器。  
當散熱器溫度過高時，可能會導致燒傷。

## ◆ 廢棄

### 注意

- 本產品請按一般工業廢棄物處置。
- 用過的電池請遵照當地的規定進行處理。



---

## ◆ 一般注意事項

### 請在使用時予以注意。

- 為了說明產品的細節部分，本資料中的部分插圖在描繪時去掉了外罩或安全保護體。在實際運轉時，請務必按規定將外罩或安全保護體安裝到原來的位置，再根據用戶手冊的說明進行運轉。
- 本資料中的插圖為代表性圖例，可能會與您收到的產品有所不同。
- 因破損或遺失而需訂購本資料時，請與本公司代理店或本資料封底記載的最近的分公司聯繫。聯繫時請告知資料編號。

## 關於保固

### ◆ 保固內容

#### ■ 保固期限

購買產品（以下稱為交付產品）的保固期限為下列 2 個條件中先滿足的條件：向指定場所交付產品後滿 1 年，或是產品自本公司出廠後滿 18 個月。

#### ■ 保證範圍

上述保固期限內發生基於本公司責任的故障時，本公司將無償提供替換品或維修服務。因交付產品到達壽命而造成的故障以及消耗零組件、壽命零組件的更換不屬於保固物件。

此外，當故障原因符合下列情形之一時，不屬於保固物件範圍。

- 因非產品型錄、手冊或另行交付的規格書等資料中記載的不恰當條件、環境、操作及使用而造成故障時
- 因交付產品以外的原因而造成故障時
- 因非本公司的改造或維修而造成故障時
- 因產品使用方法不當而造成故障時
- 因本公司出廠當時的科學、技術水準無法預計的事由而造成故障時
- 因天災、災害等其他不屬於本公司責任的原因而造成故障時

### ◆ 免責事項

- 對於因交付產品故障引發的損害及使用者的機會損失，本公司概不負責。
- 對於可程式設計的本公司產品，由本公司以外之第三方進行的程式設計（包含各種參數設定）及由此造成的結果，本公司概不負責。
- 產品型錄或手冊中記載的資訊是為了讓客戶根據用途購買合適的產品。並不保證或承諾使用這些資訊不會對本公司及第三方的智慧財產權或其他權利產生權利侵害。
- 因使用產品型錄或手冊中刊載之資訊而對第三方之智慧財產權及其他權利造成之侵害，本公司概不負責。

---

## ◆ 適用用途、條件等的確認

- 將本公司產品與其他產品配套使用時，請由使用者確認應當符合的標準、應當遵守的法規或限制條款。
- 請由使用者確認其使用的系統、機械、裝置是否適用於本公司產品。
- 用於以下用途時，請向本公司諮詢後再決定是否使用。如果可行時，則應採用賦予額定值、性能餘量的使用方法，或者採取萬一發生故障時將風險降至最低的安全措施。
  - 用於室外用途及受到潛在的化學污染、電氣干擾的用途，或者在產品型錄、手冊中未記載的條件和環境下使用
  - 原子能控制設備、焚燒設備、鐵路／航空／車輛設備、醫療機械、娛樂器材及符合行政機構和各行業限制規定的設備
  - 可能危及人身、財產安全的系統、機械、裝置
  - 燃氣、自來水、電氣供應系統或 24 小時連續運轉系統等需要高度可靠性的系統
  - 其他以上述各項為準的需要高度安全性的系統
- 將本公司產品用於可能嚴重危及人身、財產安全的用途時，請務必透過危險警告或冗餘設計，事先確認設計可確保必要的安全性以及本公司產品已進行了適當的配電和安裝。
- 產品型錄或手冊中記載的電路實例及其他應用實例僅供參考。請在確認所用設備、裝置的功能和安全性後再採用。
- 請在正確理解所有使用禁止事項和注意事項的基礎上正確使用本公司產品，以免給第三方造成意外損害。

## ◆ 規格的變更

產品型錄或手冊中記載的品名、規格、外觀及附件等可能會因品質改進或其他事由而變更，恕不事先告知。變更後，產品型錄或手冊的資料編號將進行更新，並作為改訂版發行。考慮使用或訂購資料中記載的產品時，請事先諮詢銷售通路。

# 目錄

前言	iii
手冊的使用方法	iii
相關手冊	v
安全注意事項	vi
關於保固	xii

## 1

### 階梯圖程式的特色及概要

<b>1.1</b>	<b>何謂階梯圖程式</b>	<b>1-2</b>
<b>1.2</b>	<b>特色</b>	<b>1-3</b>
	在各種時間點動作的階梯圖	1-3
	程式的模組化	1-4
	複雜數值運算的簡單記述	1-4
	與外部裝置的通訊控制	1-5
	與運動控制的完全同步	1-5
<b>1.3</b>	<b>概要</b>	<b>1-6</b>
	階梯圖程式編輯器	1-6
	階梯圖	1-7
	使用者函數	1-13
	表資料	1-19

## 2

### 階梯圖程式的開發流程

<b>2.1</b>	<b>概要</b>	<b>2-2</b>
<b>2.2</b>	<b>準備連接裝置</b>	<b>2-3</b>
	硬體連接	2-3
	安裝 MPE720 Ver.7	2-4
<b>2.3</b>	<b>建立專案</b>	<b>2-5</b>
<b>2.4</b>	<b>自動配置</b>	<b>2-6</b>
<b>2.5</b>	<b>線上連接</b>	<b>2-7</b>
<b>2.6</b>	<b>階梯圖程式製作</b>	<b>2-8</b>
<b>2.7</b>	<b>寫入階梯圖程式</b>	<b>2-12</b>

**2.8 階梯圖程式動作確認..... 2-14**

動作確認的準備.....2-14  
第 0000 行 (AND 迴路) 的動作確認.....2-16  
第 0001 行 (計時器迴路) 的動作確認.....2-17

**2.9 快閃儲存階梯圖程式..... 2-18**

**3**

**暫存器**

暫存器.....3-2

**4**

**梯形圖語言指令**

**4.1 概要..... 4-5**

階梯語言指令一覽.....4-5  
梯形圖語言指令的解讀.....4-8

**4.2 繼電器電路指令..... 4-9**

A 接點 (NOC).....4-9  
上緣觸發 A 接點 (ONP-NOC).....4-10  
下降沿 A 接點 (OFFP-NOC).....4-11  
B 接點 (NCC).....4-12  
上緣觸發 B 接點 (ONP-NCC).....4-13  
下降沿 B 接點 (OFFP-NCC).....4-14  
通電延遲計時器 (TON (1 ms)).....4-15  
斷電延遲計時器 (TOFF (1 ms)).....4-17  
通電延遲計時器 (TON (10 ms)).....4-19  
斷電延遲計時器 (TOFF (10 ms)).....4-21  
通電延遲計時器 (TON (1 s)).....4-23  
斷電延遲計時器 (TOFF (1 s)).....4-25  
上升脈衝 (ON-PLS).....4-27  
下降脈衝 (OFF-PLS).....4-29  
線圈 (COIL).....4-31  
反轉線圈 (REV-COIL).....4-32  
上緣觸發變化偵測線圈 (ONP-COIL).....4-33  
下降變化偵測線圈 (OFFP-COIL).....4-34  
設定線圈 (S-COIL).....4-35  
重置線圈 (R-COIL).....4-36

**4.3 數值運算指令..... 4-37**

儲存 (STORE).....4-37  
加法 (ADD (+)).....4-39  
擴展加法 (ADDX (++) ).....4-41  
減法 (SUB (-)).....4-43  
擴充減法 (SUBX (--)).....4-45  
乘法 (MUL (x)).....4-47  
除法 (DIV (/)).....4-49

整數餘數 (MOD)	4-51
實數餘數 (REM)	4-53
增量 (INC)	4-55
減量 (DEC)	4-57
增加時間 (TMADD)	4-59
減少時間 (TMSUB)	4-61
經過時間 (SPEND)	4-63
符號反轉 (INV)	4-66
1 的補數 (COM)	4-67
絕對值轉換 (ABS)	4-68
2 進制轉換 (BIN)	4-69
BCD 轉換 (BCD)	4-70
同位轉換 (PARITY)	4-71
ASCII 轉換 1 (ASCII)	4-72
ASCII 轉換 2 (BINASC)	4-74
ASCII 轉換 3 (ASCBIN)	4-76

#### 4.4 邏輯運算／比較指令 4-78

邏輯乘積 (AND)	4-78
邏輯或 (OR)	4-80
互斥或 (XOR)	4-82
比較 (<)	4-84
比較 (≤)	4-85
比較 (=)	4-86
比較 (≠)	4-87
比較 (≥)	4-88
比較 (>)	4-89
範圍檢查 (RCHK)	4-90

#### 4.5 程式控制指令 4-92

呼叫圖面 (SEE)	4-92
呼叫運動程式 (MSEE)	4-93
呼叫使用者函數 (FUNC)	4-95
直接輸入連續執行形 (INS)	4-96
直接輸出連續執行形 (OUTS)	4-98
執行擴展程式 (XCALL)	4-100
WHILE 文 (WHILE, END_WHILE)	4-101
FOR 文 (FOR, END_FOR)	4-103
IF 文 (IF, END_IF)	4-105
IF-ELSE 文 (IF, ELSE, END_IF)	4-107
公式敘述 (EXPRESSION)	4-109

#### 4.6 基本函數指令 4-111

平方根 (SQRT)	4-111
正弦 (SIN)	4-113
餘弦 (COS)	4-115
正切 (TAN)	4-117
反正弦 (ASIN)	4-118
反餘弦 (ACOS)	4-119
反正切 (ATAN)	4-120
指數 (EXP)	4-121

自然對數 (LN) .....	4-122
常用對數 (LOG) .....	4-123

#### **4.7 資料移動指令 ..... 4-124**

位元向左旋轉 (ROTL) .....	4-124
位元向右旋轉 (ROTR) .....	4-126
位元傳送 (MOVB) .....	4-128
文字傳送 (MOVW) .....	4-130
替換傳送 (XCHG) .....	4-132
表初始化 (SETW) .....	4-134
位元組→文字展開 (BEXTD) .....	4-136
文字→位元組壓縮 (BPRESS) .....	4-138
搜尋資料 (BSRCH) .....	4-140
排序 (SORT) .....	4-142
位元向左位移 (SHFTL) .....	4-144
位元向右位移 (SHFTR) .....	4-146
文字複製 (COPYW) .....	4-148
位元組交換 (BSWAP) .....	4-150

#### **4.8 DDC 指令 ..... 4-151**

不感帶 A (DZA) .....	4-151
不感帶 B (DZB) .....	4-153
上下限值 (LIMIT) .....	4-155
PI 控制 (PI) .....	4-157
PD 控制 (PD) .....	4-162
PID 控制 (PID) .....	4-168
一次延遲 (LAG) .....	4-173
相位前進延遲 (LLAG) .....	4-176
函數產生器 (FGN) .....	4-179
反函數產生器 (IFGN) .....	4-184
直線加減速器 1 (LAU) .....	4-189
直線加減速器 2 (SLAU) .....	4-196
脈衝寬變調 (PWM) .....	4-206

#### **4.9 表操作指令 ..... 4-209**

區段讀取 (TBLBR) .....	4-209
區段寫入 (TBLBW) .....	4-212
行搜尋 (縱向) (TBL SRL) .....	4-215
列搜尋 (橫向) (TBL SRC) .....	4-218
刪除區段 (TBLCL) .....	4-221
傳送表間區段 (TBLMV) .....	4-224
讀取佇列表 (QTBLR, QTBLRI) .....	4-228
寫入佇列表 (QTBLW, QTBLWI) .....	4-232
佇列表標清除 (QTBLCL) .....	4-236

#### **4.10 系統函數指令 ..... 4-238**

計數器 (COUNTER) .....	4-238
先進先出 (FINFOUT) .....	4-241
追蹤 (TRACE) .....	4-245
讀取追蹤資料 (DTRC-RD) .....	4-247
資訊傳送 (MSG-SND) .....	4-251

資訊傳送 (擴展) (MSG-SNDE) .....	4-253
訊息接收 (MSG-RCV) .....	4-255
訊息接收 (擴展) (MSG-RCVE) .....	4-257
伺服單元參數寫入 (MLNK-SVW) .....	4-259
伺服單元參數載入 (MLNK-SVR) .....	4-264
快閃操作 (FLASH-OP) .....	4-269
寫入運動暫存器 (MOTREG-W) .....	4-273
載入運動暫存器 (MOTREG-R) .....	4-276
匯入 (IMPORT/IMPORTL) .....	4-279
匯出 (EXPORT/EXPORTL) .....	4-287

# 5

## 工程工具 (MPE720) 功能介紹

<b>5.1</b>	<b>階梯圖程式的執行中監視 .....</b>	<b>5-4</b>
<b>5.2</b>	<b>搜尋／取代 .....</b>	<b>5-5</b>
	程式內的搜尋／取代 .....	5-5
	專案檔案中的搜尋／取代 .....	5-8
<b>5.3</b>	<b>相互參照 .....</b>	<b>5-10</b>
<b>5.4</b>	<b>檢查雙重線圈 .....</b>	<b>5-13</b>
<b>5.5</b>	<b>線圈強制 ON / OFF .....</b>	<b>5-14</b>
	從階梯圖程式設定強制 ON / OFF .....	5-14
	從強制線圈清單子視窗變更為強制 ON / OFF .....	5-14
<b>5.6</b>	<b>引用呼叫程式 .....</b>	<b>5-17</b>
<b>5.7</b>	<b>暫存器清單 .....</b>	<b>5-18</b>
	顯示暫存器對照表 .....	5-18
	切換暫存器對照表的顯示內容 .....	5-19
	編輯資料 .....	5-20
<b>5.8</b>	<b>調整面板 .....</b>	<b>5-21</b>
<b>5.9</b>	<b>階梯圖程式的有效／無效 .....</b>	<b>5-22</b>
<b>5.10</b>	<b>監看 .....</b>	<b>5-23</b>
	顯示監看資料 .....	5-23
	編輯 [ 數值 ] 欄位 .....	5-24
<b>5.11</b>	<b>安全功能 .....</b>	<b>5-25</b>
<b>5.12</b>	<b>追蹤功能 .....</b>	<b>5-26</b>
<b>5.13</b>	<b>進階使用方式 .....</b>	<b>5-27</b>
	運動程式 .....	5-27



---

## 附錄 A 系統服務暫存器一覽

---

全圖面共通 .....	A-2
DWG.H (高速掃描處理圖面) 專用 .....	A-3
DWG.L (低速掃描處理圖面) 專用 .....	A-4
掃描執行狀態和日曆 .....	A-5
系統程式軟體編號和程式記憶內容剩餘容量 .....	A-6

## 附錄 B 程式範例

---

<b>B.1</b> 由操作盤進行 JOG 運轉 .....	<b>B-2</b>
<b>B.2</b> 運動程式控制 .....	<b>B-3</b>
<b>B.3</b> 使用虛擬軸的簡易 2 軸同步運轉 .....	<b>B-4</b>

## 附錄 C EXPRESSION 指令的格式

---

<b>C.1</b> 可用公式與構成要素 .....	<b>C-2</b>
運算子 .....	C-2
操作數 .....	C-3
EXPRESSION 指令可以使用的指令 .....	C-4
<b>C.2</b> 敘述方式限制 .....	<b>C-5</b>
算術運算子 .....	C-5
比較運算子 .....	C-5
邏輯運算子 .....	C-5
指定運算子 .....	C-6
函數 .....	C-6
括弧 .....	C-6

## 附錄 D 運動參數的相關注意事項

---

## 附錄 E 運動控制器的規格一覽

---

索引

改版履歷



# 階梯圖程式的特色及概要

# 1

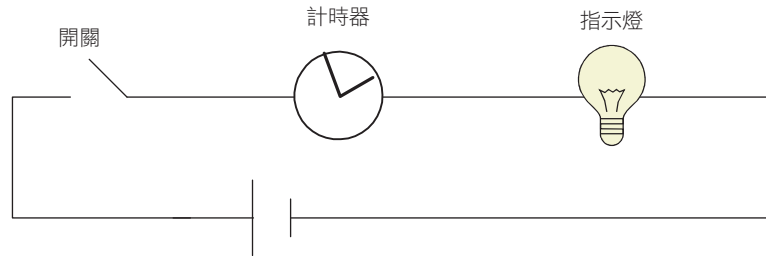
本章將就階梯圖程式之特色與及概要進行說明。

<b>1.1</b>	<b>何謂階梯圖程式</b> .....	<b>1-2</b>
<b>1.2</b>	<b>特色</b> .....	<b>1-3</b>
	在各種時間點動作的階梯圖 .....	1-3
	程式的模組化 .....	1-4
	複雜數值運算的簡單記述 .....	1-4
	與外部裝置的通訊控制 .....	1-5
	與運動控制的完全同步 .....	1-5
<b>1.3</b>	<b>概要</b> .....	<b>1-6</b>
	階梯圖程式編輯器 .....	1-6
	階梯圖 .....	1-7
	使用者函數 .....	1-13
	表資料 .....	1-19

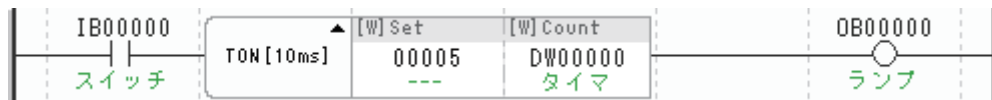
# 1.1 何謂階梯圖程式

階梯圖程式是將開關、計時器、指示燈等所架構的電路，以各種階梯語言指令與暫存器加以程式化。

<電路示意圖>



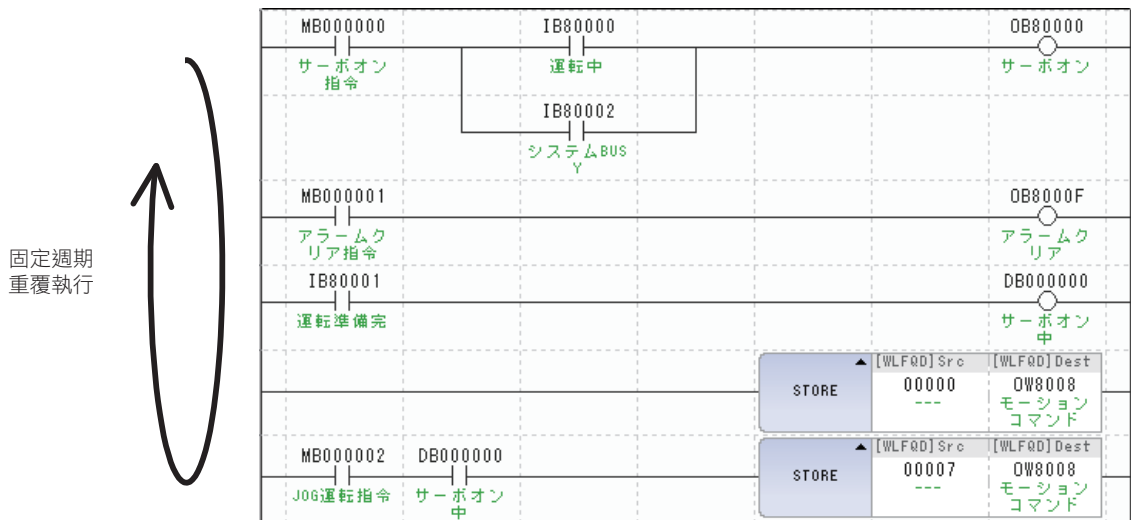
<階梯圖程式>



即使是複雜、大規模的電路，也整夠以階梯圖程式簡易敘述。

建立後的階梯圖程式以 1 次掃描執行，並以固定週期重覆執行。

<階梯圖程式範例>

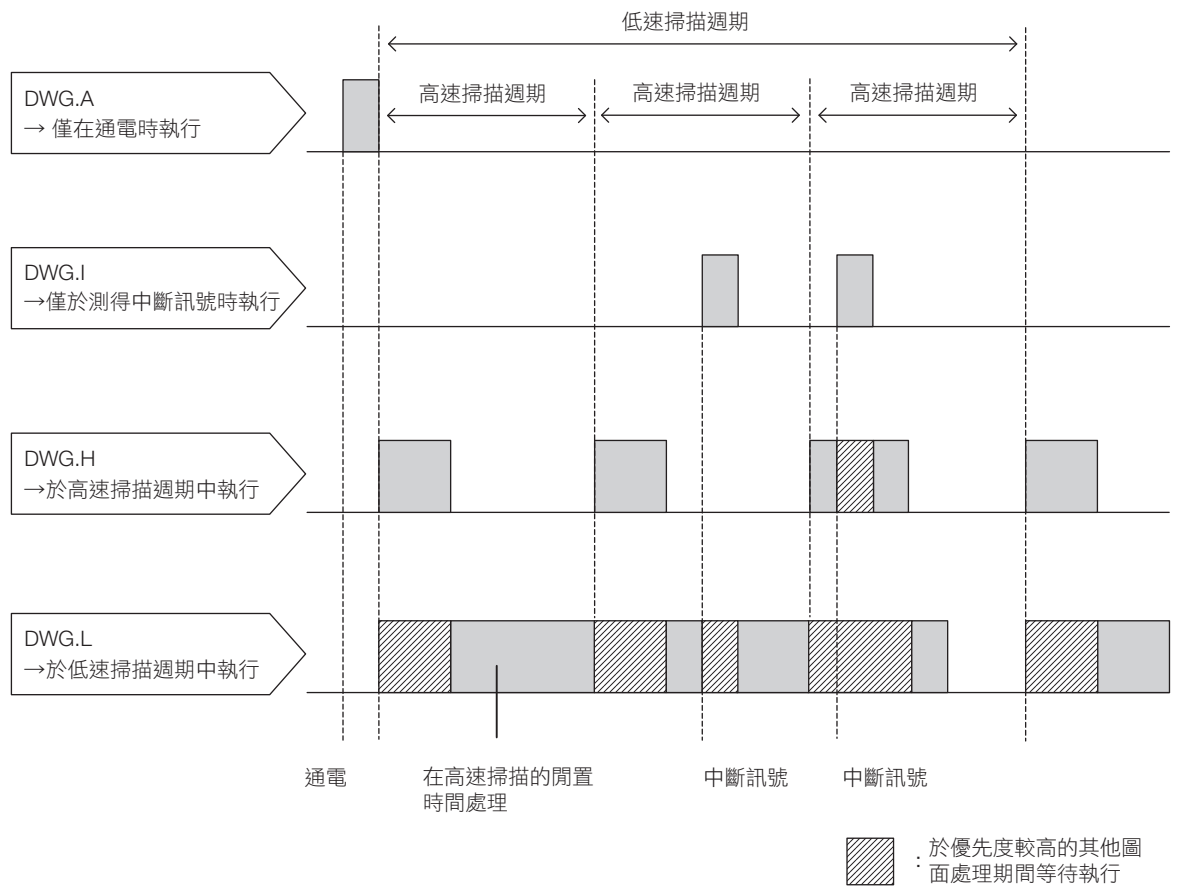


# 1.2 特色

本處將就階梯圖程式之特色進行說明。

## 在各種時間點動作的階梯圖

階梯圖程式以圖面（DWG 編號）為單位進行管理。這被稱為階梯圖面。  
MP3000 系列擁有如下張圖表所示，在各種時間點展開動作的階梯圖面。  
能夠配合各階梯圖面執行時間點的處理敘述，進行需要時間點的處理。



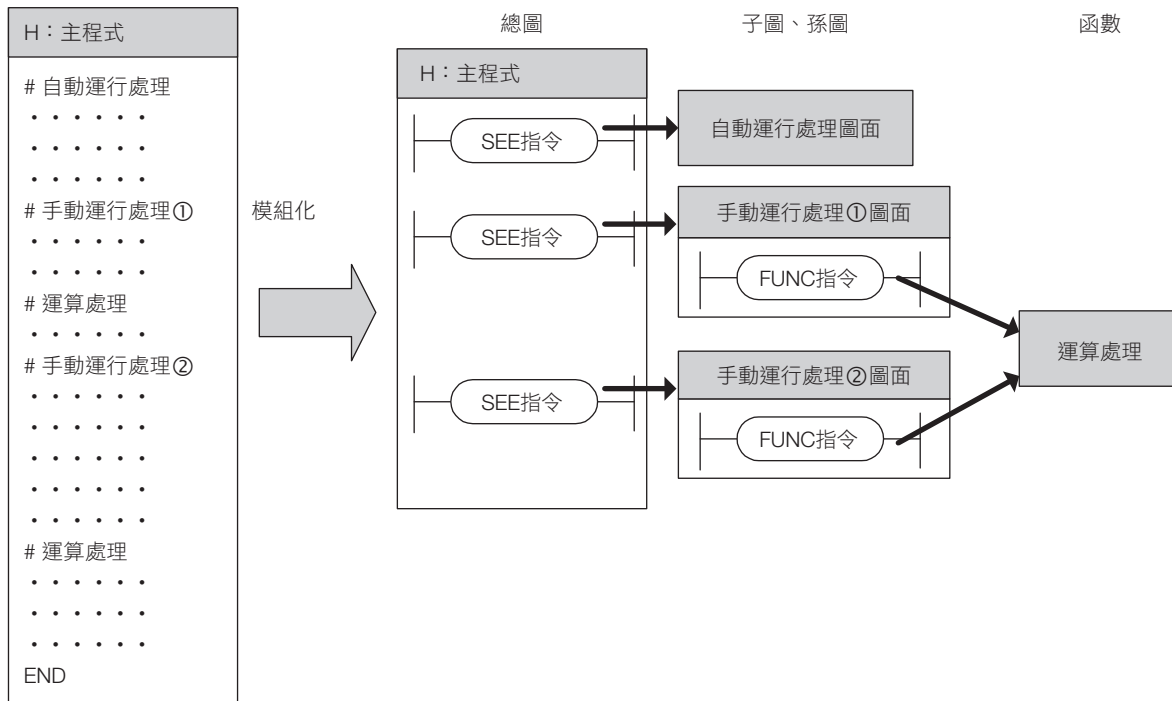
各圖面的執行時間點如下圖所示。

階梯圖	優先度*	執行時間點（處理範例）
DWG.A	1（高）	通電時僅執行一次（資料初始化）
DWG.I	2（↑）	測得中斷訊號時執行（外部訊號中斷處理）
DWG.H	3（↑）	於高速掃描週期中執行（運動控制）
DWG.L	4（低）	於低速掃描週期中執行（觸控螢幕顯示處理）

\* 數字愈小優先度愈高，處理也越優先。

## 程式的模組化

將主程式分為子圖、孫圖，並將函數等處理模組化，能夠形成更顯而易見的程式。下面是程式模組化的範例。

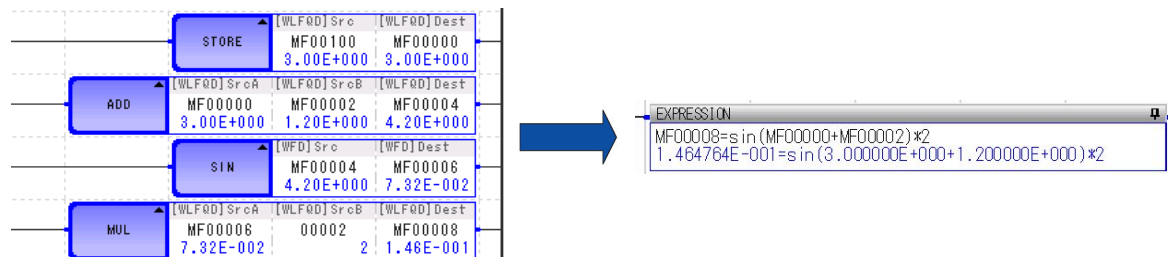


## 複雜數值運算的簡單記述

需要橫跨數行敘述的複雜計算，在使用 EXPRESSION 指令後，能夠以一個指令簡單描述。

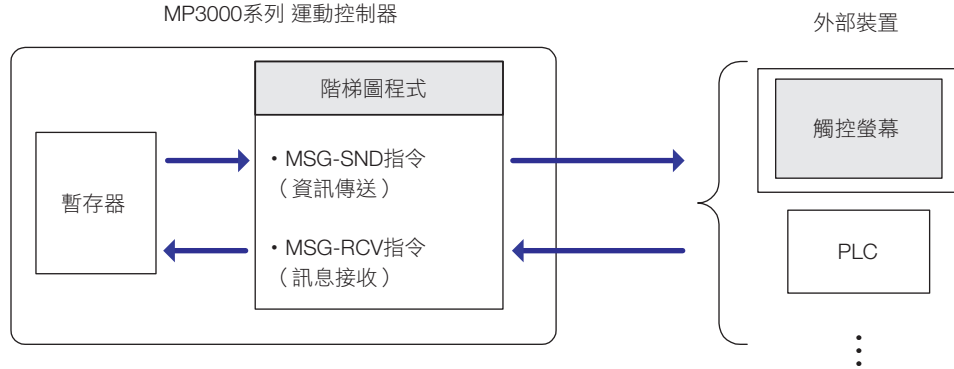
此外，變數、結構體、sin 或 cos 等基本函數能夠如 C 語言般的感觸進行敘述。

再者，也能向其他階梯語言指令般顯示現在值。



## 與外部裝置的通訊控制

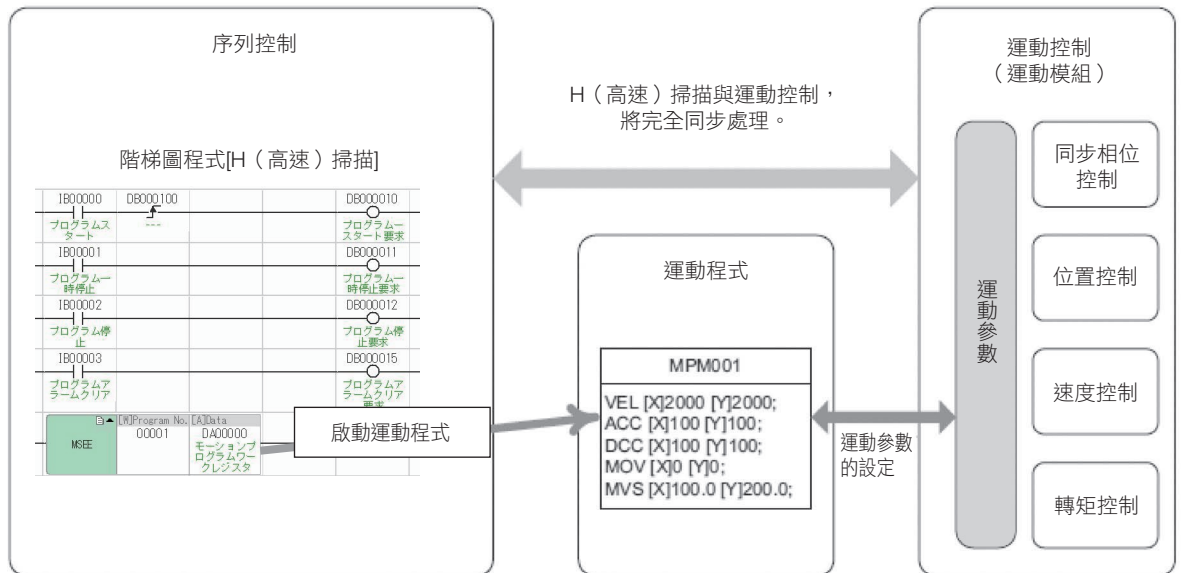
利用支援多種通訊協定的通訊專用階梯語言指令（MSG-SND 指令，MSG-RCV 指令），能夠控制與觸控螢幕、高位 PLC 等多個外部裝置之間的通訊。因而可藉由外部裝置存取運動控制器的暫存器。



- 補充** 也有無須階梯圖程式即可與外部裝置通訊的 I/O 訊息通訊功能或自動收訊功能。詳細內容，請參照以下手冊。
- 📖 MP3000 系列 通訊功能 使用者操作手冊（資料編號：SIJP C880725 12）

## 與運動控制的完全同步

以 H（高速）掃描啟動的階梯圖程式與運動控制，將完全同步處理。此外，能夠呼叫可控制複雜運動的運動程式，與階梯圖程式同步處理。

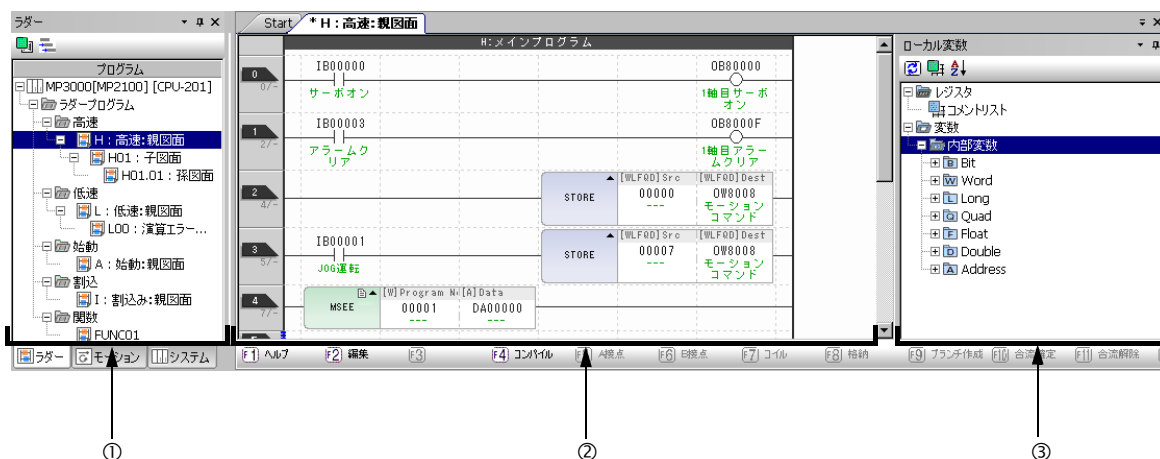


## 1.3 概要

本處將就階梯圖程式之概要進行說明。

### 階梯圖程式編輯器

MPE720 Ver.7 將在下列視窗中進行階梯圖程式的建立與編輯。



#### ① 階梯圖子視窗

以圖面為單位顯示階梯圖程式。

圖面的詳情請參照下列項目。

🔍 階梯圖 (第 1-7 頁)

#### ② 階梯圖程式編輯視窗

編輯階梯圖程式的視窗。

#### ③ 變數子視窗

顯示變數。關於暫存器的詳情，請參照以下章節內容。

🔍 3 章 暫存器

除了此處介紹的視窗以外，還以許多種視窗及工具列。



## 階梯圖

階梯圖程式依圖號（DWG 編號）區分，以圖為單位進行圖面（階梯圖）管理。  
本圖為階梯圖程式的基本結構。

### 種類與階層組成

以下說明階梯圖的種類與階層組成。

#### ◆ 種類

階梯圖依處理目的可分類為以下 4 種。

- DWG.A（啟動處理圖）  
想要任意設定暫存器資訊時使用。高速掃描及低速掃描前執行。
- DWG.I（中斷處理圖）  
想要優先處理選配模組輸入的訊號時使用。  
不論掃描週期為何，將於高速掃描前執行。
- DWG.H（高速掃描處理圖）  
運動控制，或於高速情況下控制 I/O 時使用。
- DWG.L（低速掃描處理圖）  
HMI 及外部機器之通訊與一般 I/O 控制用。

以下表顯示各自圖面的優先度、執行條件及最大圖面數。

圖面種類	優先度*	執行條件	最大圖面數
DWG.A（啟動處理圖）	1	通電 （通電時僅執行一次）	64
DWG.I（中斷處理圖）	2	外部中斷 （選配模組的 DI 中斷，依計數器模組一致中斷執行）	64
DWG.H（高速掃描處理圖）	3	定週期啟動 （每次高速掃描時執行）	200
DWG.L（低速掃描處理圖）	4	定週期啟動 （每次低速掃描時執行）	1000

\* 數字愈小優先度愈高。

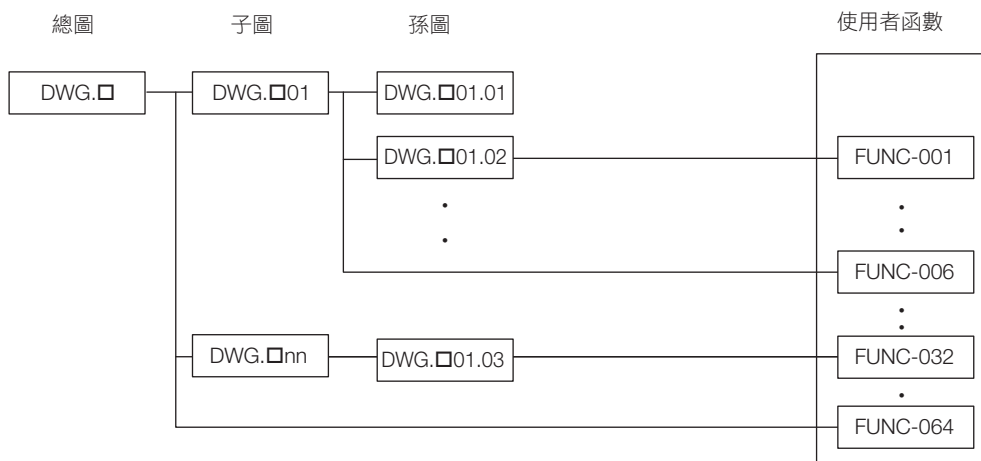
#### ◆ 階層組成

各階梯圖由總圖、子圖、孫圖及運算錯誤處理圖所組成。

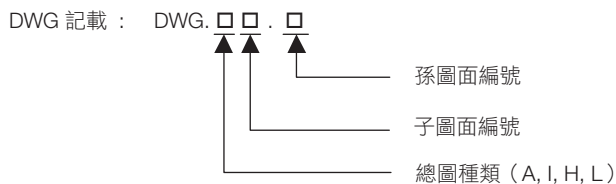
- 總圖  
「執行條件」成立時，系統程式將自動執行。
- 子圖  
引用來自總圖的 SEE 指令後執行。
- 孫圖  
引用來自子圖的 SEE 指令後執行。
- 運算錯誤處理圖  
發生運算錯誤時，系統程式將自動執行。

無法從總圖引用不同種類圖面的子圖，從子圖引用不同種類圖面的孫圖。此外，不可直接從總圖引用孫圖。  
一定要從總圖引用子圖，再從子圖引用孫圖。這稱為圖面的階層組成。

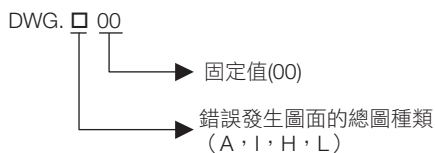
如下圖所示，各處理程式以總圖—子圖—孫圖階層化製作。



(註) o=A,I,H,L



(註) 運算錯誤圖面才會變成下列名稱。



各階梯圖的圖面明細如下表所示。

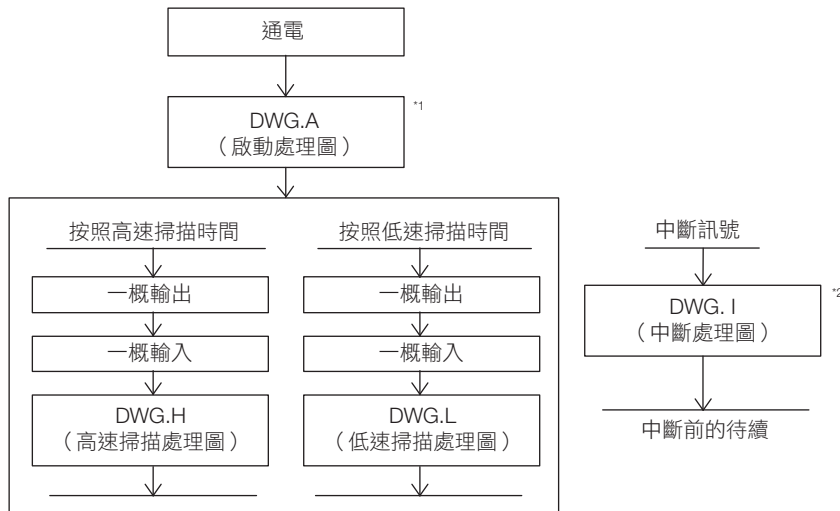
圖面	圖面數			
	DWG.A	DWG.I	DWG.H	DWG.L
總圖	1	1	1	1
運算錯誤處理圖	1	1	1	1
子圖	總合	總合	總合	總合
孫圖	最大 62	最大 62	最大 998	最大 1998

**補充**

與圖面不同，各圖面擁有可自由引用的函數。函數引用來自總圖、子圖及孫圖的 FUNC 指令後執行。可產生之函數最大為 2000。

## 圖面的執行控制

各階梯圖依該優先度執行如下。



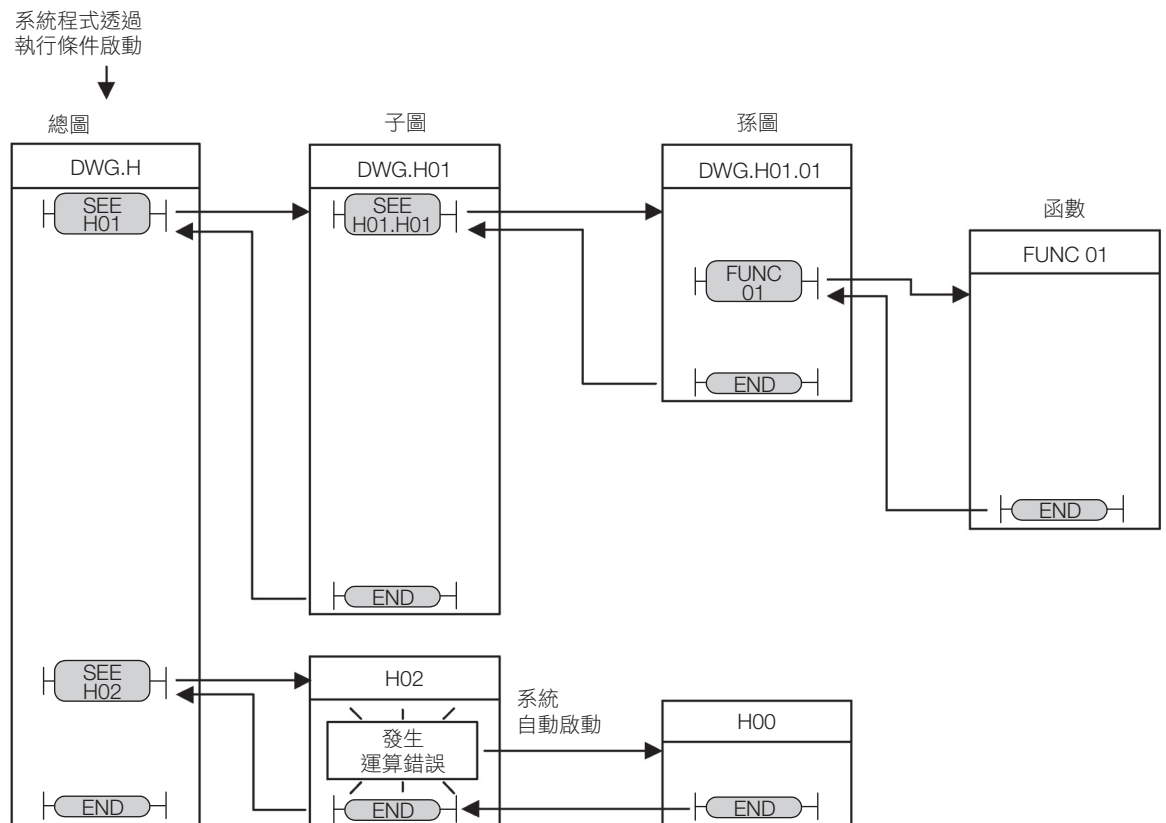
\*1. 製作 DWG.A 時，在通電後開始執行。

\*2. 輸入中斷訊號時，DWG.H 及 DWG.L 即使在執行中，DWG.I 應優先執行。

(註) 各圖面的總圖自系統自動叫出後執行。

## ◆ 圖面的執行處理方式

階層化後的各圖面，自上位圖引用到下位圖的方式執行。以 DWG.H (高速掃描處理圖) 為例，圖面的執行處理方式如下圖所示。



(註) 1. 自系統自動打開總圖。

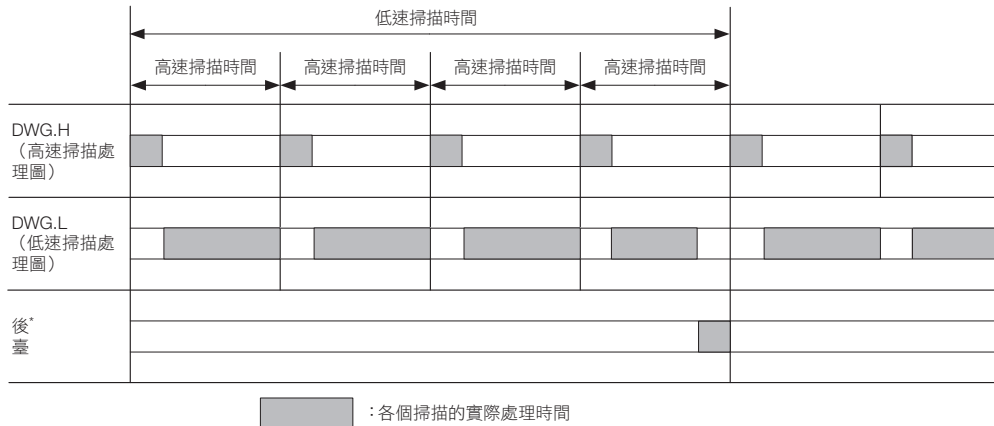
子圖與孫圖，經總圖與子圖的 SEE 指令編程後，執行開啟。

2. 所有的圖面都可引用函數。也可以從函數引用函數。
3. 發生運算錯誤時，各圖的運算錯誤處理圖將會自動開啟。
4. 使用運算錯誤處理圖時，圖號一定要指定為 00。

### ◆ 高速／低速掃描處理圖面的執行排程

DWG.H（高速掃描處理圖面）與 DWG.L（低速掃描處理圖面）無法同時執行。DWG.L 在 DWG.H 的未執行時間執行。

DWG.H 執行的週期稱為高速掃描時間，低速掃描處理圖面被執行的週期稱低速掃描時間。

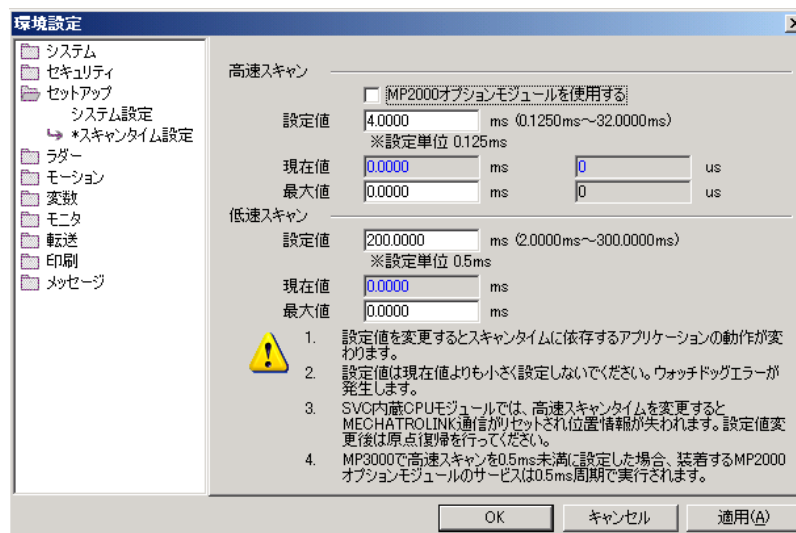


\* 為執行系統內部處理（通訊處理等）者。

### ◆ 高速／低速掃描時間之設定

高速掃描時間及低速掃描時間使用 MPE720 Ver.7，依據以下流程進行設定。

1. 從主選單選取 [ 檔案 ] - [ 環境設定 ]（或點選我的工具視窗的 [ 系統設定 ] 按鈕）後，將顯示 [ 環境設定 ] 對話方塊。
2. 選擇 [ 設定 ] - [ 掃描時間設定 ]，便會顯示下列對話方塊。



設定值：輸入各掃描時間。

現在值：離線時顯示「0.0 ms」。上線時顯示各掃描的實際處理時間。

最大值：顯示各掃描實際處理時間的最大值。此外，可設定最大值，讓超過設定數值前能夠維持該數值。

3. 將高速掃描時間輸入至 [ 高速掃描 ] 的 [ 設定值 ] 欄位，低速掃描時間輸入至 [ 低速掃描 ] 的 [ 設定值 ] 欄位中。

各掃描時間可設定之數值及初始值如下所示。

項目	可設定的值	初始值
高速掃描時間	0.125 ms ~ 32 ms (0.125 ms 單位)	4.0 ms
低速掃描時間	2.0 ms ~ 300.0 ms (0.5 ms 單位)	200.0 ms

(註) 依機種不同，可設定之數值及初始值亦不同。詳細請參閱使用模組對應之使用者操作手冊。



重要

設定高速掃描及低速掃描時間時，請注意以下幾點事項。

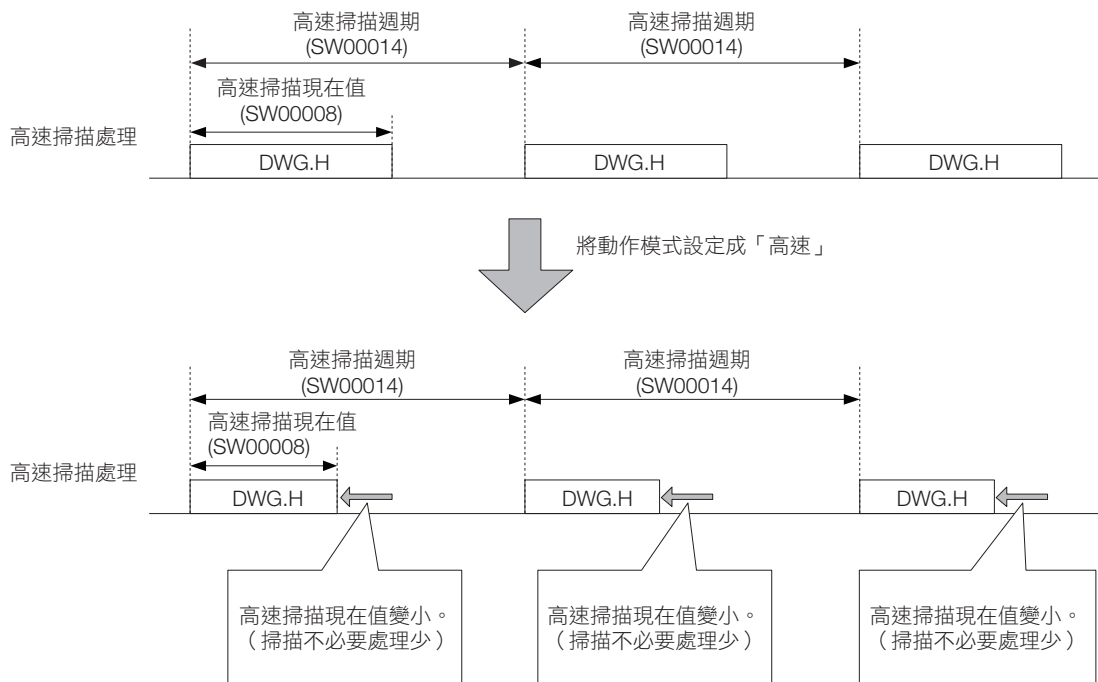
1. 掃描設定值請設定在掃描最大值的 1.25 倍以上。  
掃描設定值設定接近掃描最大值之數值時，MPE720 的視窗之更新將變得異常緩慢，導致發生通訊逾時。此外，掃描最大值超過掃描設定值時，可能因監視器超時，造成運動控制器關閉系統。
2. 使用 MECHATROLINK-II 時的通訊週期，使用 MECHATROLINK-III 時請將傳送週期設定為整數倍。變更通訊週期及傳送週期時，請先確認掃描設定值。
3. 伺服 ON 時請勿變更掃描設定值。特別是軸移動時（馬達轉動時），馬達的轉動動作可能發生異常，請絕對不可進行變更。
4. 執行掃描時間的設定／變更時，請先執行存檔，將資料儲存於快閃記憶體。

#### ◆ 設定高速圖面動作模式

高速圖面動作模式是針對 DWG.H（高速掃描處理圖面）的模式設定。

不使用 DWG.I（中斷處理圖面割）時，請選擇高速模式。可將 DWG.H 的處理最佳化。

要使用 DWG.I（中斷處理圖面）時，請選擇一般模式。若選擇高速模式，便無法執行 DWG.I。



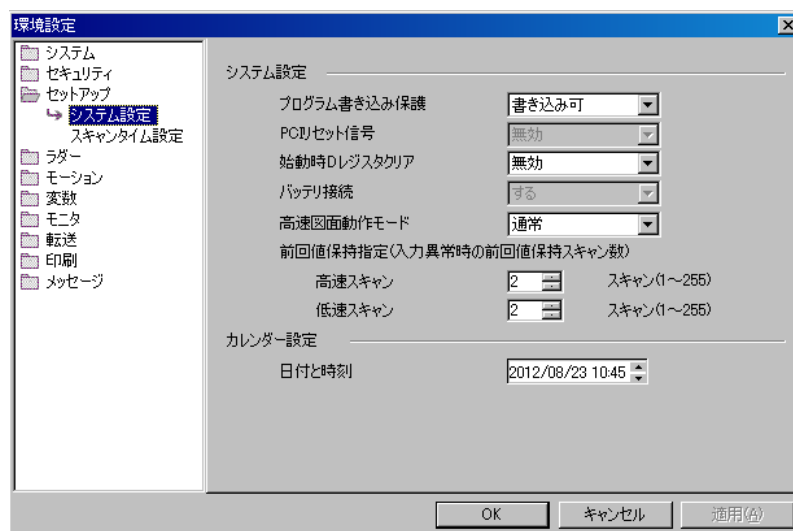
## 補充

- DWG.A（啟動處理圖面）、DWG.I（中斷處理圖面）、DWG.L（低速掃描處理圖面）不具備變更動作模式的功能。
- 使用越多下列指令，便能讓 DWG.H 處理的最佳化效果越好。

種類	功能
繼電器電路指令	上緣觸發 A 接點
	下降沿 A 接點
	上緣觸發 B 接點
	下降沿 B 接點
	上升脈衝
	下降脈衝
	線圈
	反轉線圈
	上升變化偵測線圈
	下降變化偵測線圈
	設定線圈
	重置線圈
數值運算指令	+1 增量
	-1 減量

關於高速圖面動作模式的設定，請使用 MPE720 Ver.7 以下列步驟操作。

1. 從主選單選取 [ 檔案 ] – [ 環境設定 ]（或點選我的工具視窗的 [ 系統設定 ] 按鈕）後，將顯示 [ 環境設定 ] 對話方塊。
2. 選擇 [ 設定 ] – [ 系統設定 ]，便會顯示下列對話方塊。

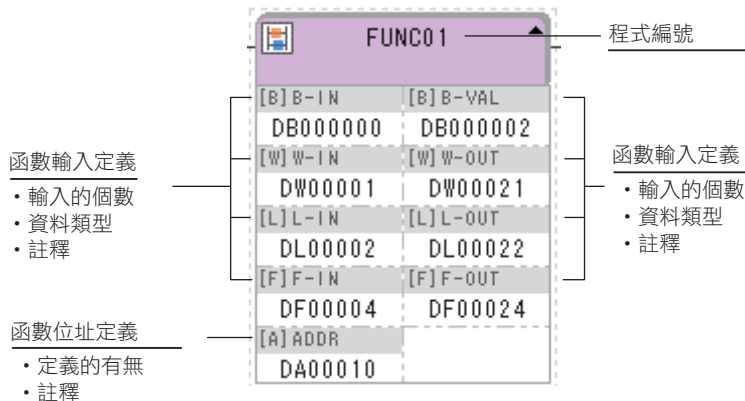


3. 勾選 [ 高速圖面動作模式 ] 欄的 [ 高速 ] 或 [ 一般 ]。

## 使用者函數

### 何謂使用者函數

使用者函數是讓使用者能自由建立函數定義（程式編號與輸出入定義）與處理內容的函數。  
函數定義內容如下圖所示。



### 使用者函數概要

使用者函數本身的處理內容，是以階梯圖程式建立而成。  
使用者函數引用來自總圖、子圖及孫圖的 FUNC 指令後執行。  
亦可如下所示，引用使用者參數。

- 任何圖面均可自由引用。
- 種類及階層不同的圖面可同時引用相同的使用者函數。
- 可由使用者函數引用其他的使用者函數。
- 可從不同的程式引用，次數不限。

使用使用者函數有以下優點。

- 使用者程式容易零組件化
- 使用者程式容易製作與維護

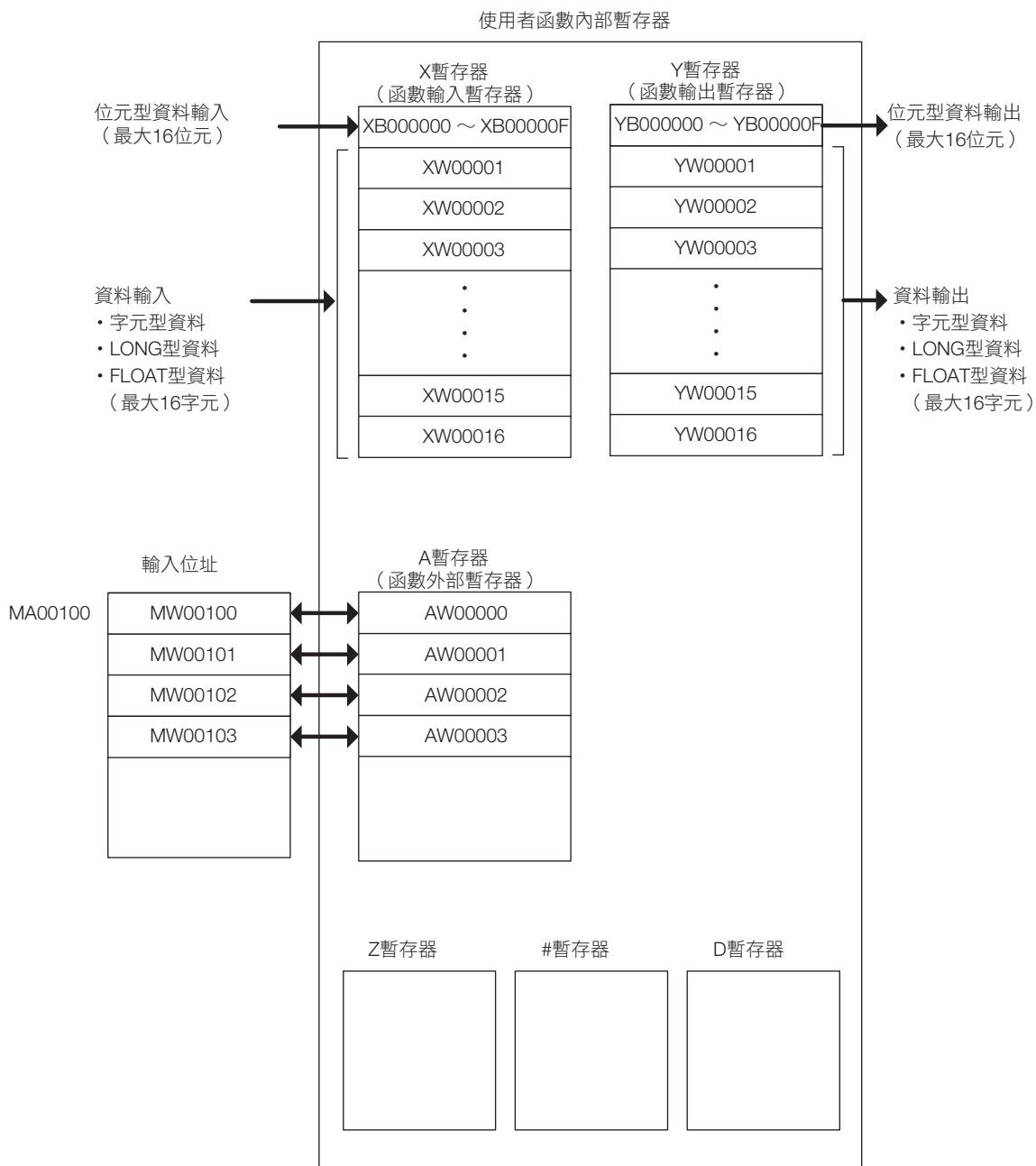


重要

呼叫使用者函數時，請考量使用者函數個別暫存器之數值，進行初始化等處理。  
詳情請參照如下內容。

3 章 暫存器 - ■ 於使用者函數使用局部暫存器時的注意事項（第 3-5 頁）

使用者函數的輸出入資料使用者函數內部暫存器的關係，如下圖所示。



**補充**

1. X 暫存器、Y 暫存器、Z 暫存器、D 暫存器在呼叫使用者函數時，對於初始值的處理不同。詳情請參照如下內容。  
■ 3 章 暫存器 - ■ 於使用者函數使用局部暫存器時的注意事項 (第 3-5 頁)
2. S, M, I, O, C 暫存器亦可由函數內部引用。



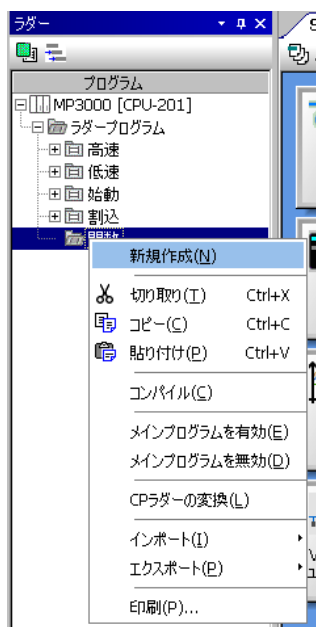
## 建立使用者函數

此處將以下列使用者函數為例，說明使用者函數的建立步驟。

函數定義	名稱	備註
程式編號	FUNC01	-
函數輸入值	IN	整數型
函數輸出值 1	OUT1	整數型
函數輸出值 2	OUT2	整數型
處理內容		
設函數輸入值 (IN) 為 2 倍，輸出至函數輸出值 1 (OUT1)。		
設函數輸入值 (IN) 為 3 倍，輸出至函數輸出值 2 (OUT2)。		

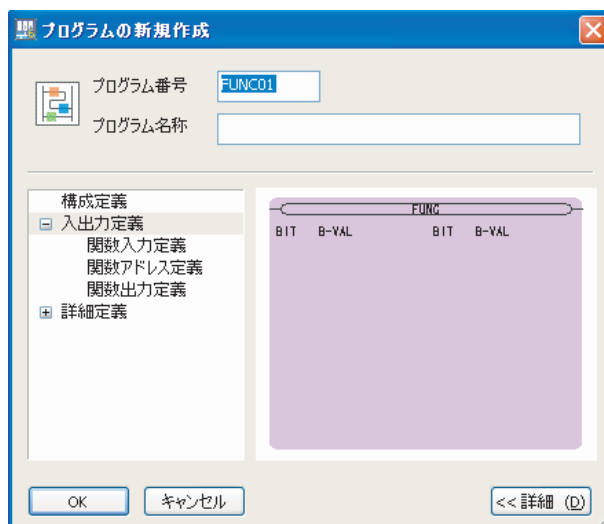
### ◆ 製作步驟

1. 點選啟動器的 [ 編程 ] - [ 階梯圖程式 ]。  
將顯示階梯圖子視窗。
2. 右鍵點選階梯圖程式的 [ 函數 ] 後，選擇 [ 新建 ]。

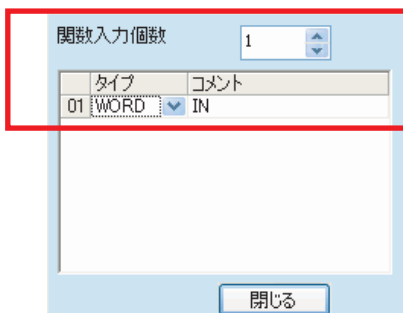


顯示 [ 新建程式 ] 對話方塊。

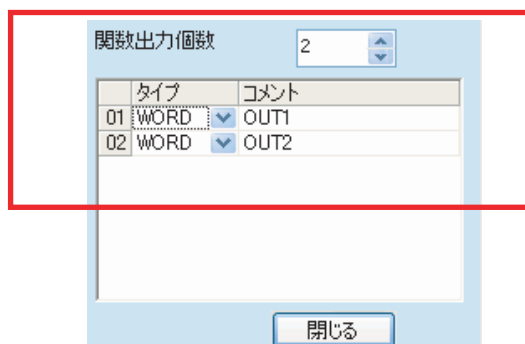
3. 在 [ 程式編號 ] 輸入方塊中輸入「FUNC01」。



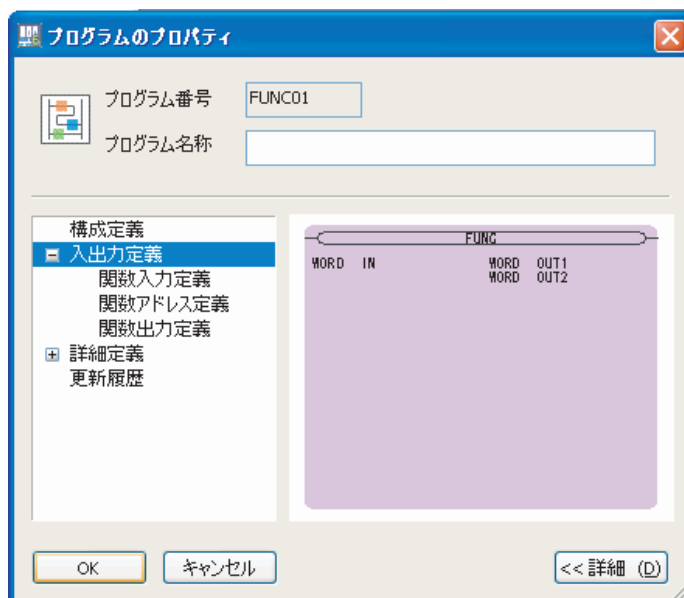
4. 選擇 [ 輸出入定義 ] – [ 函数輸入定義 ]，進行如下的描述。



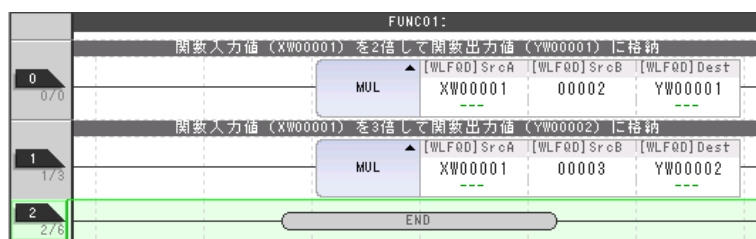
5. 選擇 [ 輸出入定義 ] – [ 函数輸出定義 ]，進行如下的描述。



6. 點選 [OK] 按鈕。至此，函数定義設定完畢。



7. 在步驟 5 建立的使用者函數「FUNC01」的圖面，建立如下所示的階梯圖程式。



8. 在顯示階梯圖程式的狀態下，選擇主選單的 [ 編譯 ]-[ 編譯 ]，進行編譯。  
編譯結束後，會自動儲存階梯圖程式。



重要

若輸出子視窗於編譯過程中顯示錯誤，便不會儲存階梯圖程式。

至此，使用者函數建立完畢。

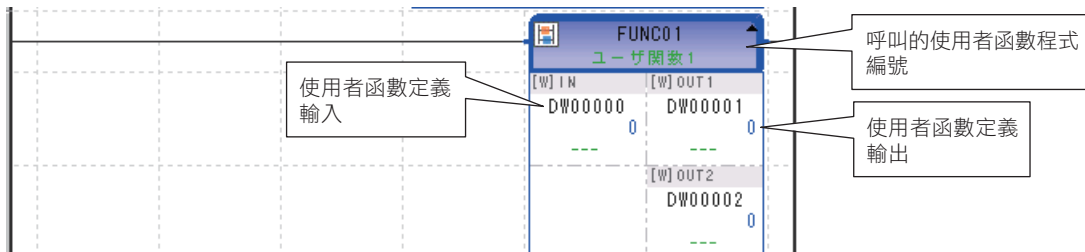
## 呼叫使用者函數

使用者函數可由階梯圖面以 FUNC 指令呼叫後使用。

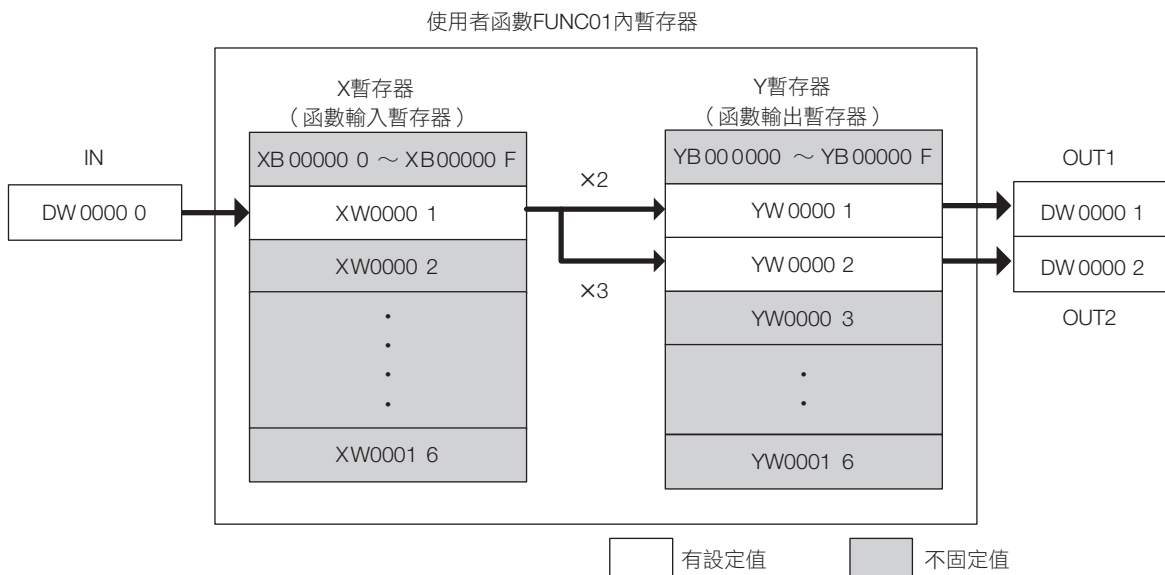
本處將以 DWG.H (高速掃描處理圖) 呼叫已建立的使用者函數為例進行說明。

### ■ 由 DWG.H 呼叫使用者函數 FUNC01 的範例

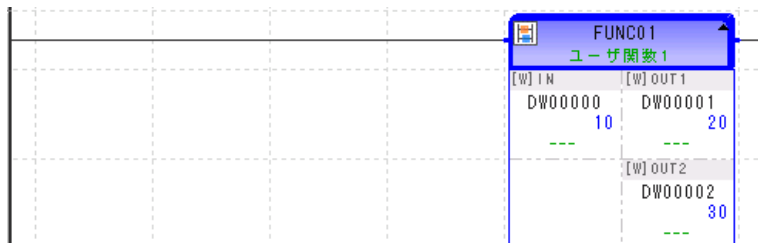
對 DWG.H 敘述 FUNC 指令如下。



如上圖般敘述後，將會是如下般的狀態。



在 DWG.H 設 DW00000 為 10，則 DW00001 為 20，DW00002 為 30，可確認以建立的使用者函數是否被正常呼叫。

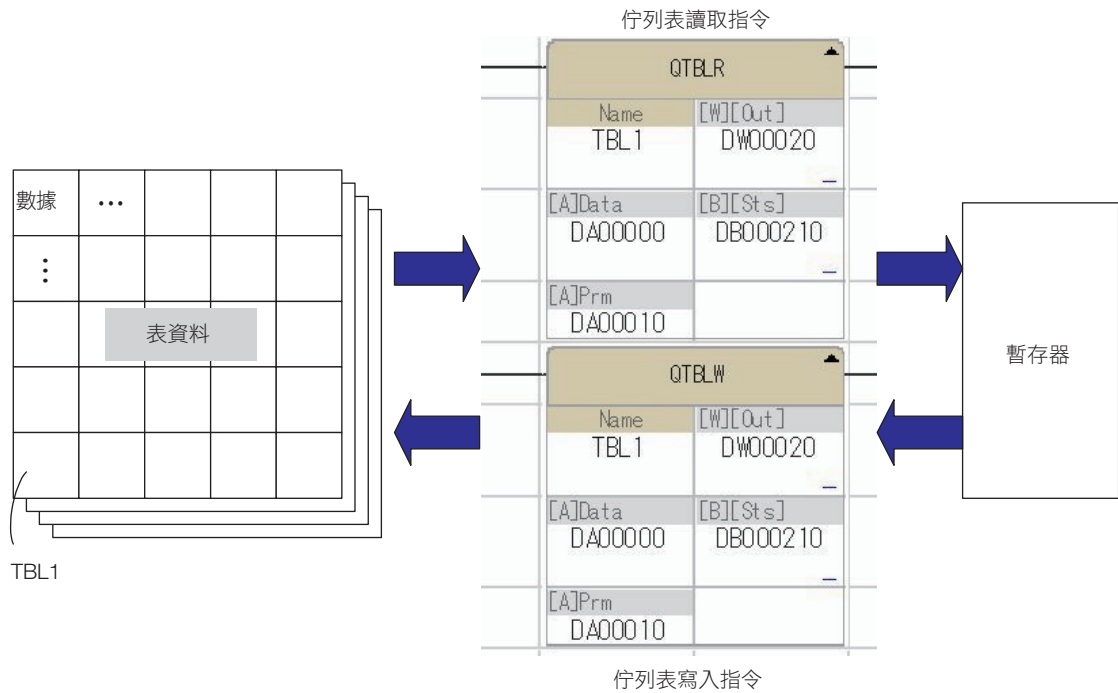


## 表資料

### 何謂表資料

表資料是指以表單格式管理的資料。存放在不同於暫存器的場所。

階梯圖程式執行表操作指令後，可由表資料表對暫存器佈局，或由暫存器填入表資料。當暫存器空間不足時，亦可作為資料的迴避場所。



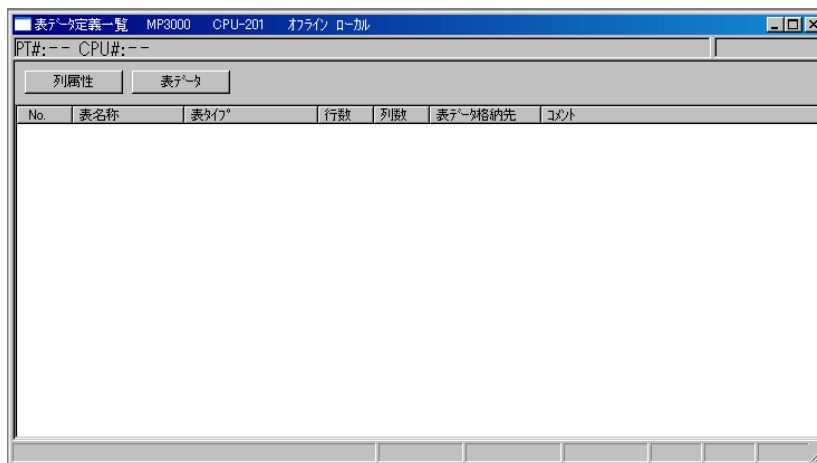
### 建立表資料

建立表資料表時，將設定下列表定義資訊以及列屬性內容。

表定義資訊	內容
表名稱	表的名稱。
表類型	選擇陣列型／記錄型。 <ul style="list-style-type: none"> <li>陣列型：所有列均為同樣列屬性的類型。</li> <li>記錄型：不同列具有不同列屬性的類型。</li> </ul>
列數	表的列數量。(最多 10000 列)
行數	表的行數量。(最多 10000 行)
表註釋	表的註釋。
表資料儲存位置	選擇一般／電池組備用。 <ul style="list-style-type: none"> <li>一般：每 1 張表的最大容量為 5MB。</li> <li>電池組備份：每 1 張表的最大容量為 3MB。</li> </ul> 關於表資料最大容量及電池組備份可容納的機種，請參照以下章節。 <b>附錄 E 運動控制器的規格一覽</b>
列屬性	內容
列名稱	列的名稱。
資料類型	可為整數、2 倍長整數、實數、字串。
尺寸	資料類型的長度。
顯示類型	可為 2 進制、10 進制、16 進制、實數、字串。
列註釋	列的註釋。

### ■ 製作步驟

1. 選擇啟動器的 [ 應用程式 ]-[Engineering Manager] 。  
啟動 Engineering Manager 。
2. 選擇 Engineering Manager 主功能表的 [ 檔案 ]-[ 開啟 ]-[ 表資料定義 ]-[ 表資料一覽 ] 。  
顯示 [ 表資料定義一覽 ] 對話方塊。



3. 讓 [ 表資料定義一覽 ] 對話方塊繼續顯示，並選擇 Engineering Manager 主功能表的 [ 檔案 ]-[ 新建 ] 。  
出現 [ 表定義設定 ] 對話方塊。
4. 設定表定義資訊後，點擊 [OK] 按鈕。



顯示 [ 表資料列屬性 ] 對話方塊。

## 5. 設定表資料列屬性。

(註) 如果在表定義資訊設定的表類型中選擇陣列型，列屬性僅能設定 1 種。

No.	列名称	データ	サイズ	表示タイプ	列コメント
1	POS_NO	整数	002	10進数	データ番号
2	X_DATA	倍長整数	004	10進数	X位置データ
3	Y_DATA	倍長整数	004	10進数	Y位置データ

## 6. 從主選單選擇 [ 檔案 ]-[ 儲存 ]。

步驟 2 中顯示的 [ 表資料定義一覽 ] 對話方塊所建立的表，顯示如下。

No.	表名称	表タイプ	行数	列数	表データ格納先	コメント
000001	TBL	陣列型	00100	00003	通常	XY位置データ

至此，表資料建立完畢。

## 補充

1. 建立表資料時，資料內容將清零進行初始化。
2. 在 [ 表資料定義一覽 ] 對話方塊選擇已建立的資料，點擊 [ 表資料 ] 按鈕，進行表資料的讀出／寫入。
3. 由階梯圖程式操作表資料時，請使用表操作指令。





# 階梯圖程式的開發流程

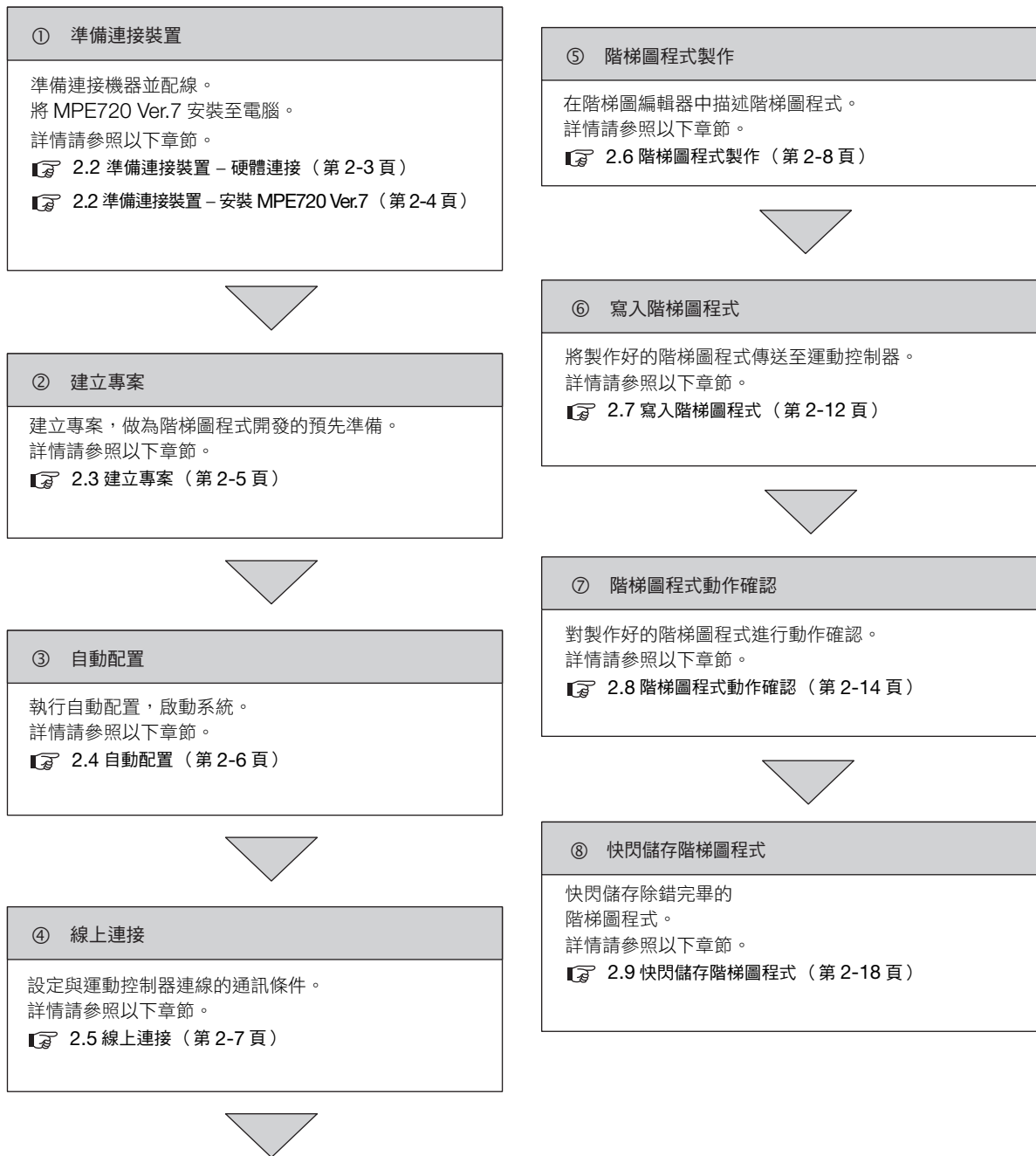
# 2

本章將就階梯圖程式的開發流程進行說明。

<b>2.1</b>	<b>概要</b> .....	<b>2-2</b>
<b>2.2</b>	<b>準備連接裝置</b> .....	<b>2-3</b>
	硬體連接 .....	2-3
	安裝 MPE720 Ver.7 .....	2-4
<b>2.3</b>	<b>建立專案</b> .....	<b>2-5</b>
<b>2.4</b>	<b>自動配置</b> .....	<b>2-6</b>
<b>2.5</b>	<b>線上連接</b> .....	<b>2-7</b>
<b>2.6</b>	<b>階梯圖程式製作</b> .....	<b>2-8</b>
<b>2.7</b>	<b>寫入階梯圖程式</b> .....	<b>2-12</b>
<b>2.8</b>	<b>階梯圖程式動作確認</b> .....	<b>2-14</b>
	動作確認的準備 .....	2-14
	第 0000 行 (AND 迴路) 的動作確認 .....	2-16
	第 0001 行 (計時器迴路) 的動作確認 .....	2-17
<b>2.9</b>	<b>快閃儲存階梯圖程式</b> .....	<b>2-18</b>

## 2.1 概要

本處將以下列流程說明階梯圖程式的開發流程。



(註) 上述流程僅為階梯圖程式設定時的範例其中之一。要將程式安裝至實際裝置時，請進行與外部裝置間的設定。

## 2.2

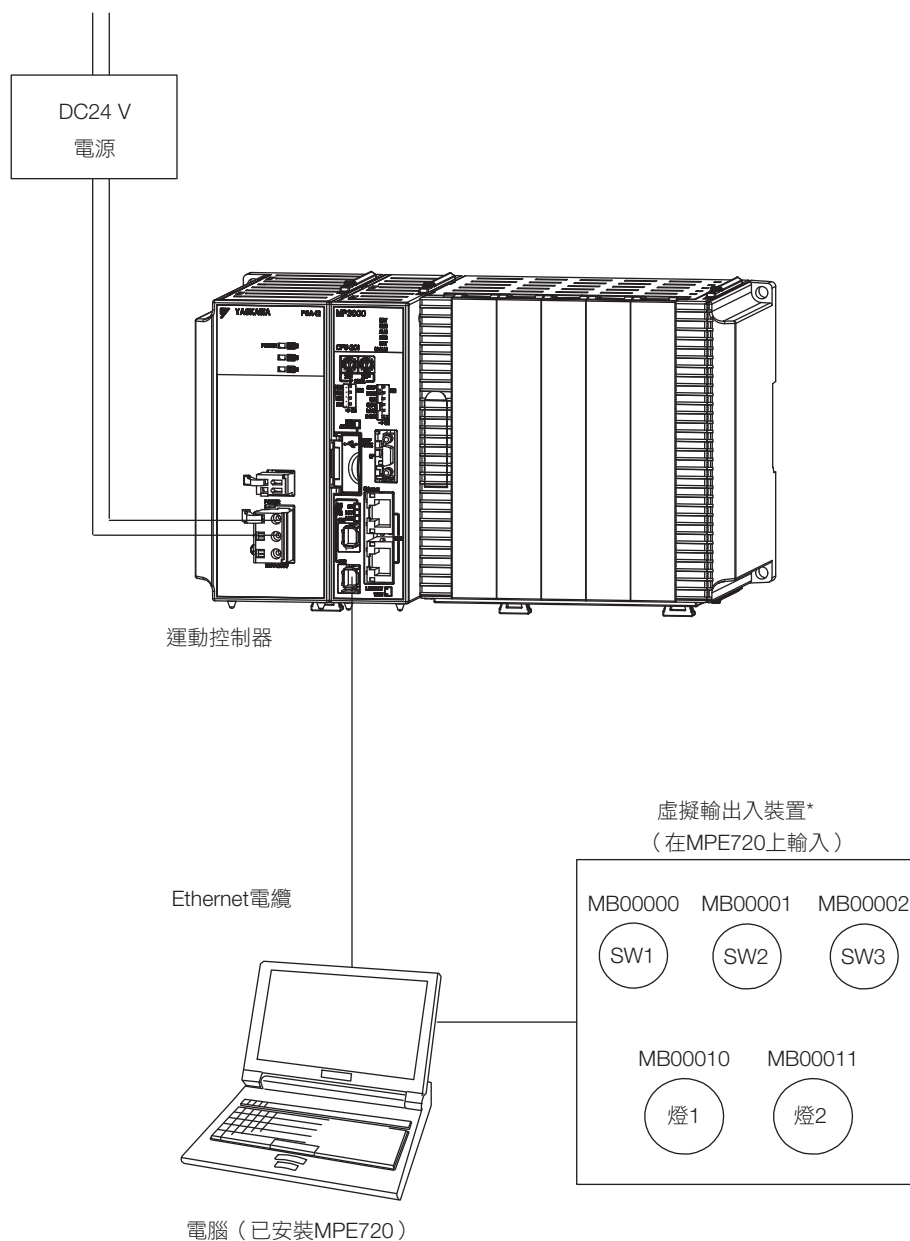
## 準備連接裝置

將說明與硬體的連接以及 MPE720 Ver.7 的安裝步驟。

## 硬體連接

準備連接機器並配線。

本處將以下列系統架構模型進行說明。



\* 本處以利用運動控制器內部 M 暫存器當作虛擬輸出入裝置為例示範。

實際的輸出入訊號，請連接運動控制器的 I/O 模組等，使用 I，O 暫存器製作階梯圖程式。


---

## 安裝 MPE720 Ver.7

---

將 MPE720 Ver.7 安裝至電腦。

安裝步驟請參閱下列手冊。

 MP2000/MP3000 系列 運動控制器系統 安裝手冊（資料編號：SIJP C880725 00）

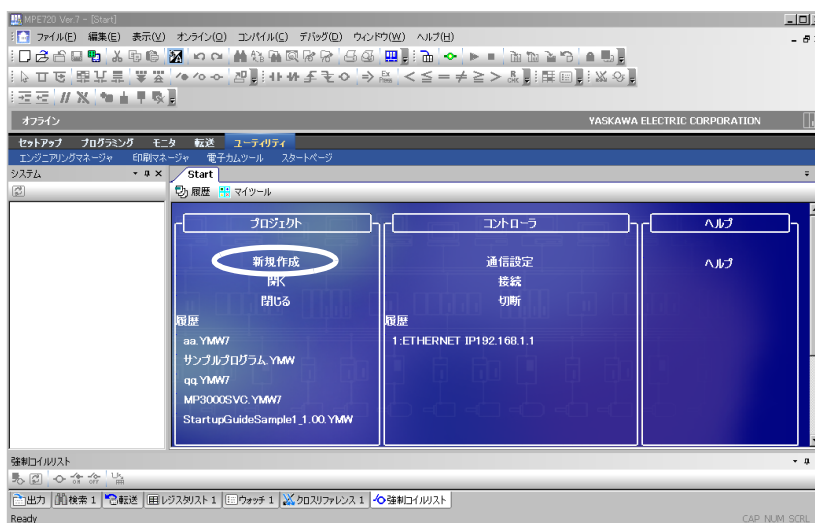
## 2.3 建立專案

以下列步驟建立專案。

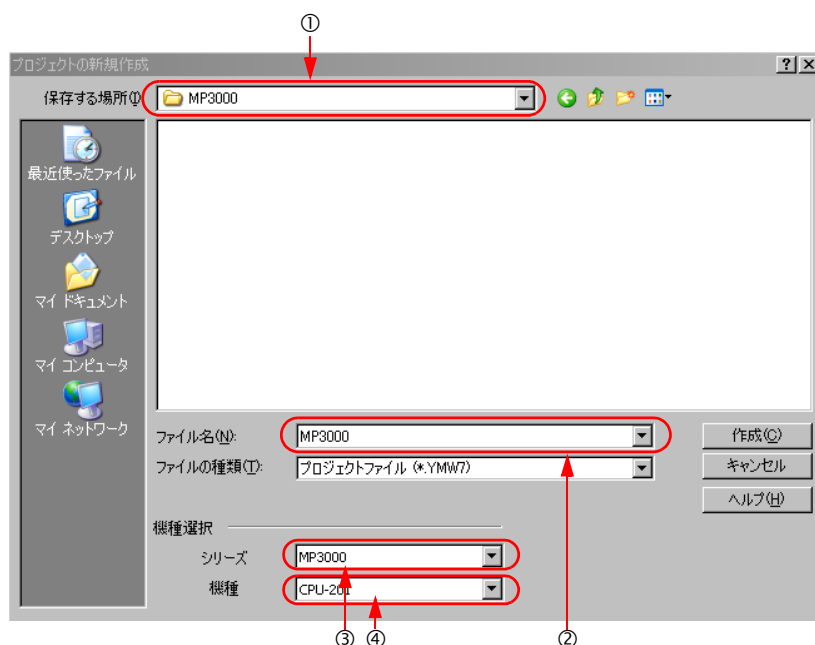
1. 雙擊電腦桌面的下列圖示，啟動 MPE720 Ver.7。



2. 點擊 [開始] 標籤的 [新建]。



3. 指定檔案名稱、檔案儲存路徑、運動控制器系列及機種。



- ① 在 [ 儲存路徑 ] 輸入方塊中指定儲存位置。
- ② 在 [ 檔案名稱 (N) : ] 輸入方塊中輸入檔案名稱。
- ③ 在 [ 系列 ] 輸入方塊中選擇適用系列。
- ④ 在 [ 機種 ] 輸入方塊中選擇適用機種。

4. 點選 [產生] 按鈕。

## 2.4

## 自動配置

使用自動配置功能，啟動系統。自動配置係指自動認識裝備於運動控制器的選配模組及 MECHATROLINK 連接器連接的裝置。使用這項功能，將使系統啟動作業時間更短、更簡便。自動配置可由運動控制器的 DIP 開關或 MPE720 執行。

關於自動配置的步驟，請參閱以下手冊。

📖 MP3000 系列 MP3200 使用者手冊（資料編號：SIJP C880725 10）

📖 MP3000 系列 MP3300 使用者手冊（資料編號：SIJP C880725 21）

## 2.5

## 線上連接

設定運動控制器與電腦連線的通訊條件。

關於通訊條件的步驟，請參閱以下手冊。

📖 MP2000/MP3000 系列 運動控制器系統 安裝手冊（資料編號：SIJP C880725 00）

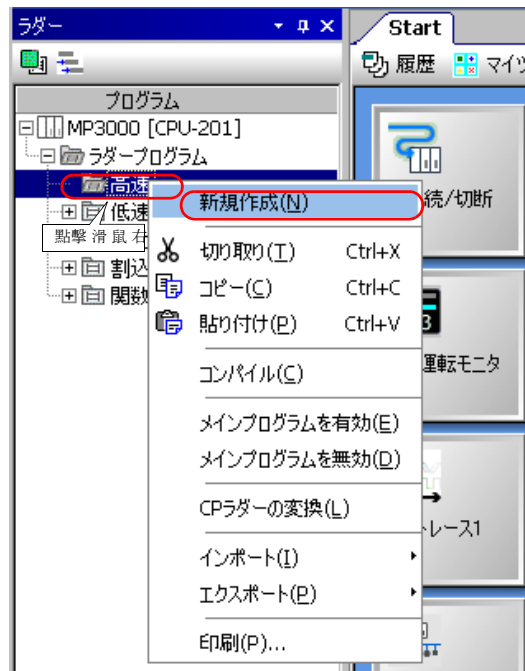
## 2.6 階梯圖程式製作

下列為階梯圖程式的製作步驟。



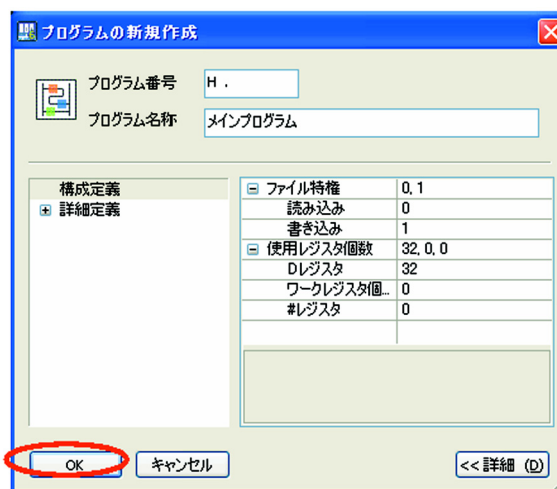
下列雖為製作「高速程式」的操作步驟，但要製作「低速」及「啟動」程式時，亦可使用本步驟。

1. 點選啟動器的 [ 編程 ] – [ 階梯圖程式 ]。  
將顯示階梯圖子視窗。
2. 以右鍵點選階梯圖的 [ 高速 ] 後，選擇 [ 新建檔案 ]。



顯示 [ 新建程式 ] 對話方塊。

3. 點選 [ OK ] 按鈕。



將啟動階梯圖編輯器。



- 製作階梯圖程式。  
階梯圖程式需以插入梯級、插入指令、填入指令之參數的順序依序製作。詳情請參照如下內容。  
📖 階梯圖程式製作範例（第 2-9 頁）
- 在顯示階梯圖程式的狀態下，選擇主選單的 [ 編譯 ]-[ 編譯 ]，進行編譯。  
編譯結束後，會自動儲存階梯圖程式。



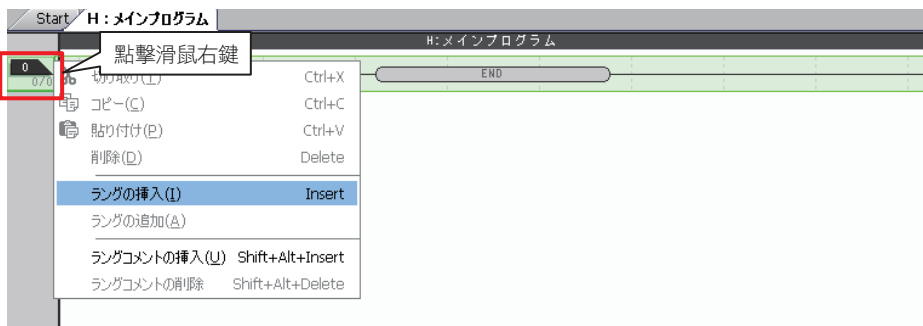
重要

若輸出子視窗於編譯過程中顯示錯誤，便不會儲存階梯圖程式。

## 階梯圖程式製作範例

下列為插入 NOC 指令的範例。

- 右鍵點選有顯示行編號的位置，點選 [ 插入梯級 ]。

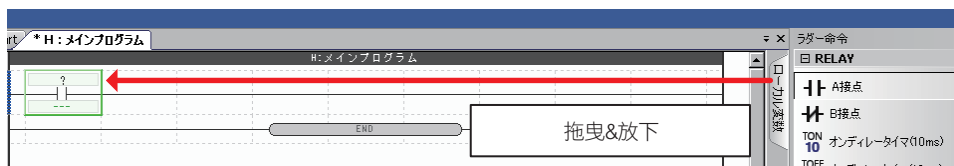


將插入梯級。

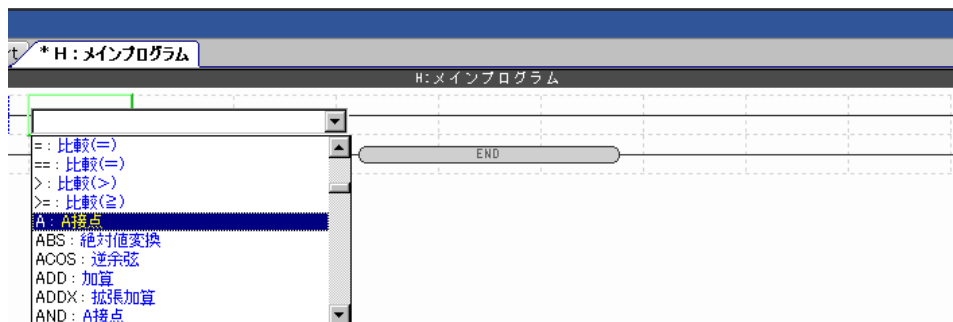



- 使用下列任一方法製作 NOC 指令。

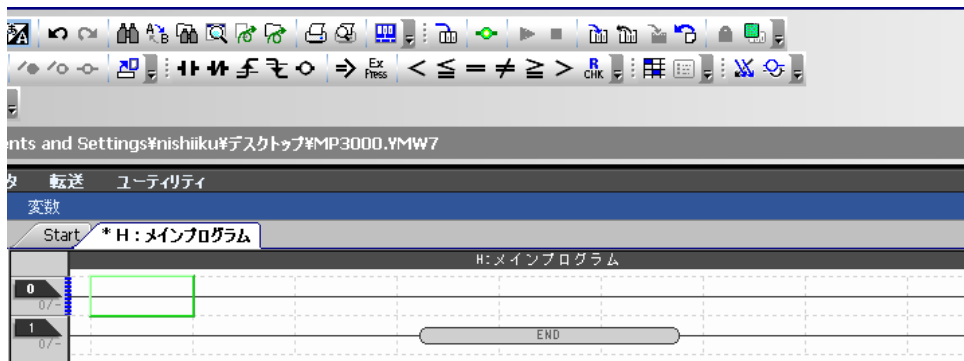
- 從任務窗格子視窗中選擇 [RELAY] – [A 接點] 後，拖曳到剛插入的梯級上。



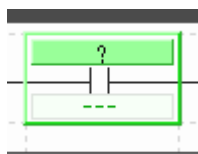
- 雙擊想要在已插入梯級上製作 NOC 指令的位置後，從跳出的清單欄內選擇 [A：A 接點]。



- 選擇想要在已插入梯級上製作 NOC 指令的位置後，點選 [NOC 指令 ] 按鈕。

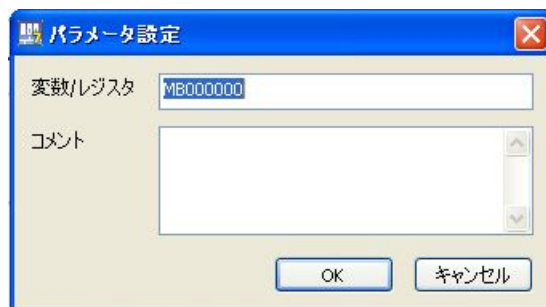


3. 雙擊「？」部位。

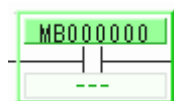


將顯示 [ 參數設定 ] 對話方塊。


4. 在 [ 變數／暫存器 ] 欄內輸入暫存器「MB000000」後，點選 [OK] 按鈕。



NOC 指令將顯示「MB000000」。



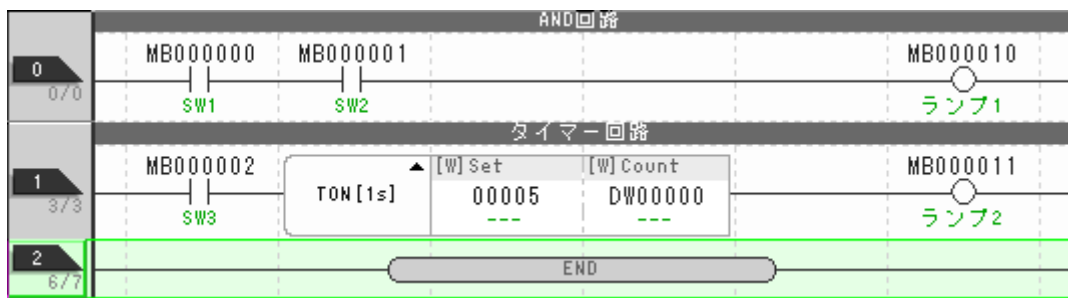
(註) 可使用的類別與資料類型會因指令而異。各指令的詳情請參照以下章節內容。

 4 章 梯形圖語言指令

**補充** 在顯示欲插入註釋的行編號處按下右鍵，選擇 [ 插入梯級註釋 ] 便可插入註釋。

5. 執行步驟 1 ~ 4，製作階梯圖程式。下列的階梯圖程式與時序圖為其中一例。

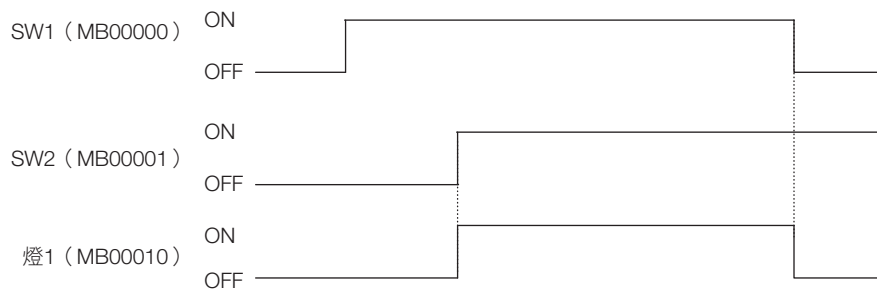
< 階梯圖程式範例 >



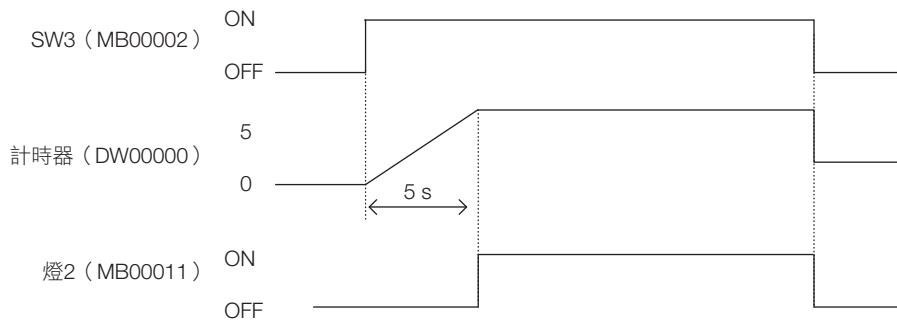
(註) 上述的階梯圖程式為使用 M 暫存器製作開關及燈的範例。  
實際製作階梯圖程式時，請配合實際裝置使用 I、O 暫存器等製作。

< 計時器圖表範例 >

AND 電路的動作



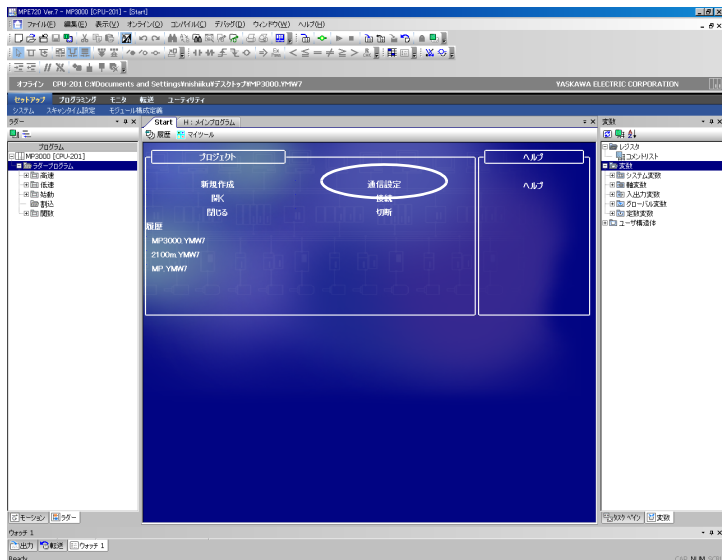
計時器電路的動作



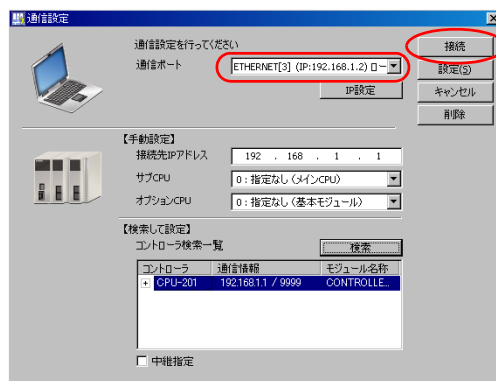
## 2.7 寫入階梯圖式

依下列步驟，將階梯圖式寫入運動控制器。但如果是在線上製作階梯圖式，則可略過此步驟。

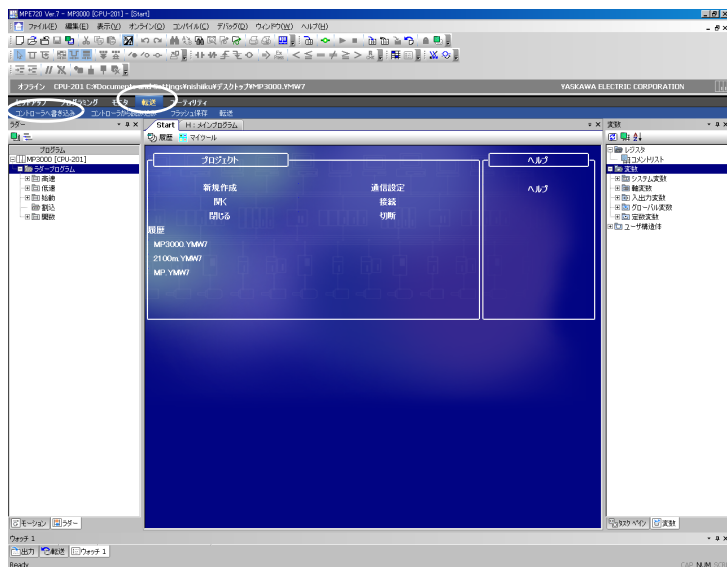
1. 點擊 [開始] 標籤的 [通訊設定]。



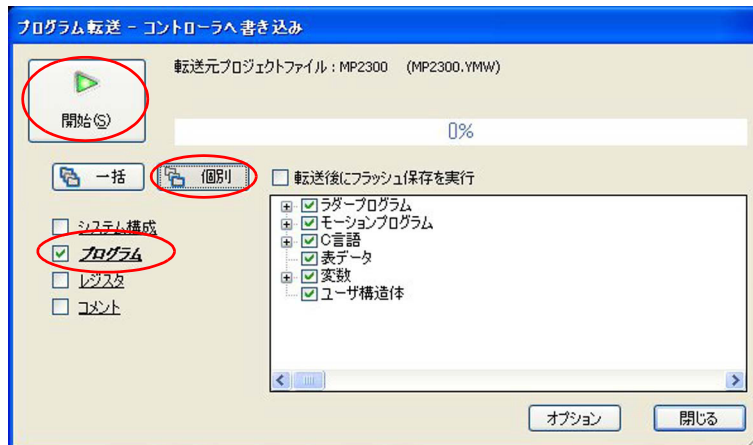
2. 在 [通訊設定] 對話框的 [通訊埠] 方塊中，選擇設定的通信埠並點擊 [連線] 按鈕。



3. 從啟動器選取 [傳輸] – [寫入控制器內]。

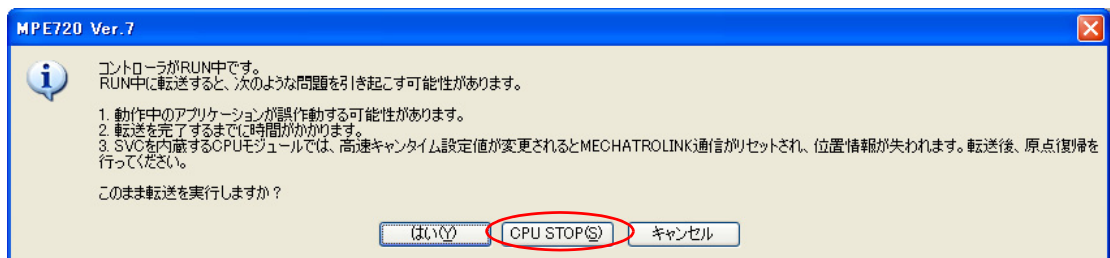


4. 點擊 [ 個別 ] 按鈕，然後只點選 [ 程式 ] 核取方塊。按下 [ 開始 ] 按鈕。



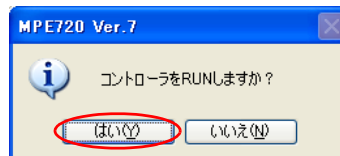
- (註) 1. 個別傳送時，控制器內的相同檔案將被選取的專案檔覆蓋資料。  
2. 一併傳送時，運動控制器的 RAM 在傳送前將被清空，並一併寫入專案資料。

5. 點選 [ CPU STOP ] 按鈕。



開始傳輸。

6. 在下列對話框中點擊 [ 是 ] 按鈕。



運動控制器 RUN。

## 2.8

## 階梯圖程式動作確認

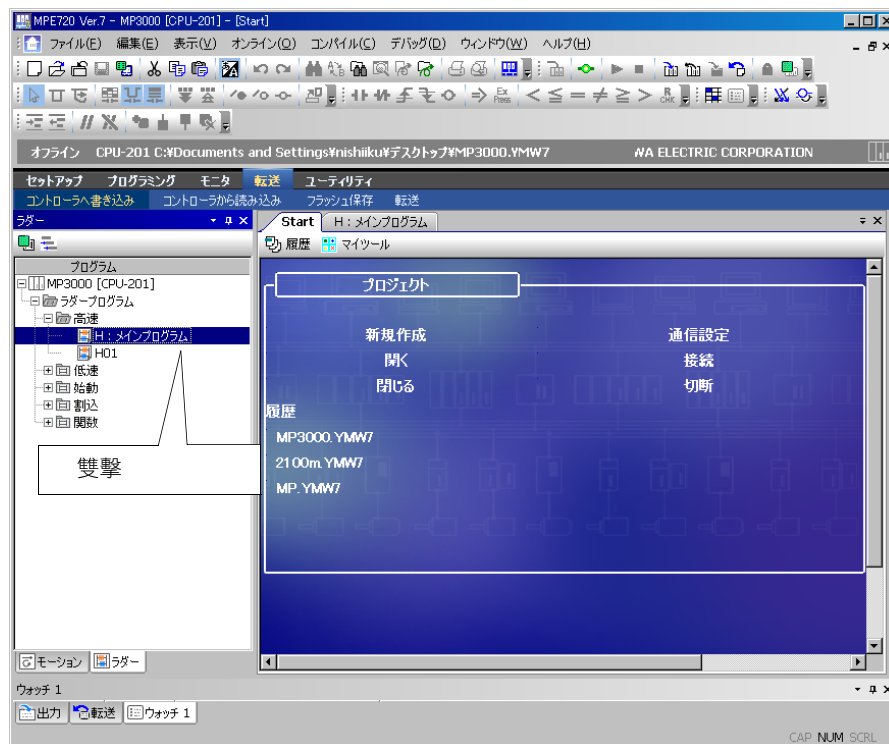
以下列步驟對製作好的階梯圖程式進行動作確認。

使用暫存器列表操作暫存器，並於暫存器列表與階梯圖編輯器的執行中監控確認動作是否正常。

## 動作確認的準備

以下列步驟準備動作確認。

1. 雙擊階梯圖子視窗中已製作的階梯圖程式。



2. 點擊 [ 暫存器清單 1 ] 標籤。

將顯示 [ 暫存器清單 1 ] 對話方塊。

## 補充

如果未顯示 [ 暫存器清單 1 ] 標籤，請以下列任一方法讓 [ 暫存器清單 1 ] 對話框顯示出來。

- 由主功能表選擇 [ 顯示 ] - [ 暫存器清單 ] - [ 暫存器清單 1 ]。
- 由啟動器選擇 [ 監視 ] - [ 暫存器清單 ]。



3. 在 [ 暫存器 ] 輸入方塊中輸入「MB000000」。  
如下所示，開啟暫存器清單。

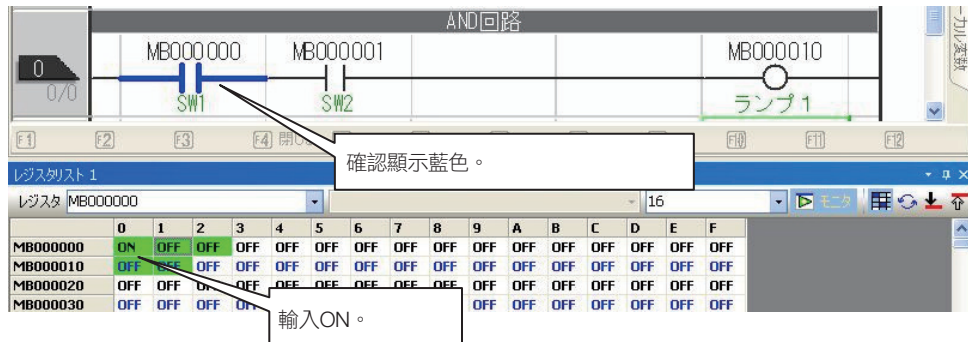
レジスタ	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
MB000000	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
MB000010	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
MB000020	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
MB000030	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
MB000040	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
MB000050	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
MB000060	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

## 第 0000 行 (AND 迴路) 的動作確認

依下列步驟確認第 0000 行的動作。

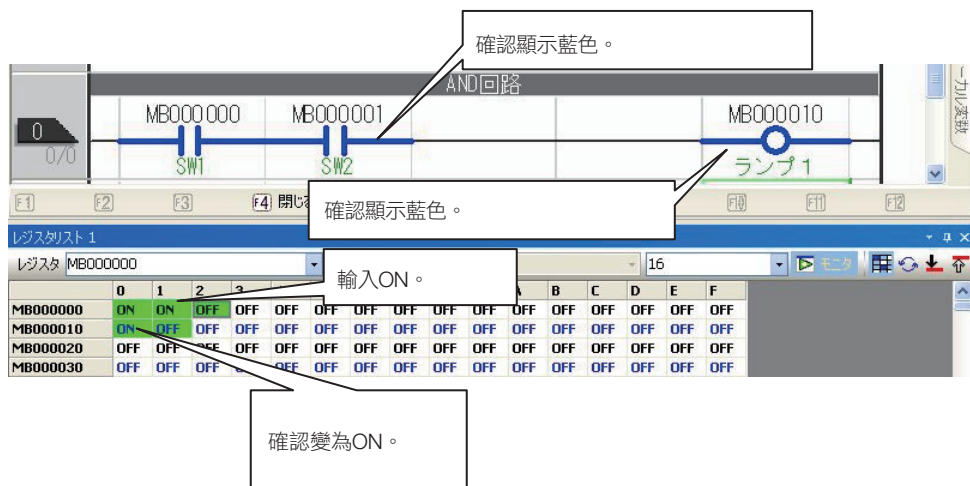
1. 到暫存器清單中，使 MB000000 成為 ON。此時，確認階梯圖編輯器上 MB000000 的 A 接點以藍色顯示。

(註) 當線圈或接點以藍色顯示時，表示 ON 狀態。



2. 到暫存器清單中，使 MB000001 成為 ON。此時確認如下。

- 此時，確認階梯圖編輯器上 MB000001 的 A 接點與 MB000010 的線圈以藍色顯示。
- 暫存器清單上的 MB000010 為 ON。





## 第 0001 行（計時器迴路）的動作確認

依下列步驟確認第 0001 行的動作。

到暫存器清單中，使 MB000002 成為 ON。此時確認如下。

- 確認計時器（DW00000）每秒遞增計數。

確認每秒遞增計數。

レジスタ	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
MB000000	ON	ON	ON	OFF	ON	OFF	OFF	OFF	ON	ON	ON	OFF	OFF	ON	OFF	OFF
MB000010	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF	OFF
MB000020	OFF	OFF	OFF	OFF	ON	OFF	OFF	OFF	ON	ON	ON	OFF	OFF	ON	OFF	OFF

輸入ON。

- 5 秒後，確認階梯圖編輯器上 MB000011 的線圈以藍色顯示。
- 暫存器清單上的 MB000011 為 ON。

確認以藍色顯示。

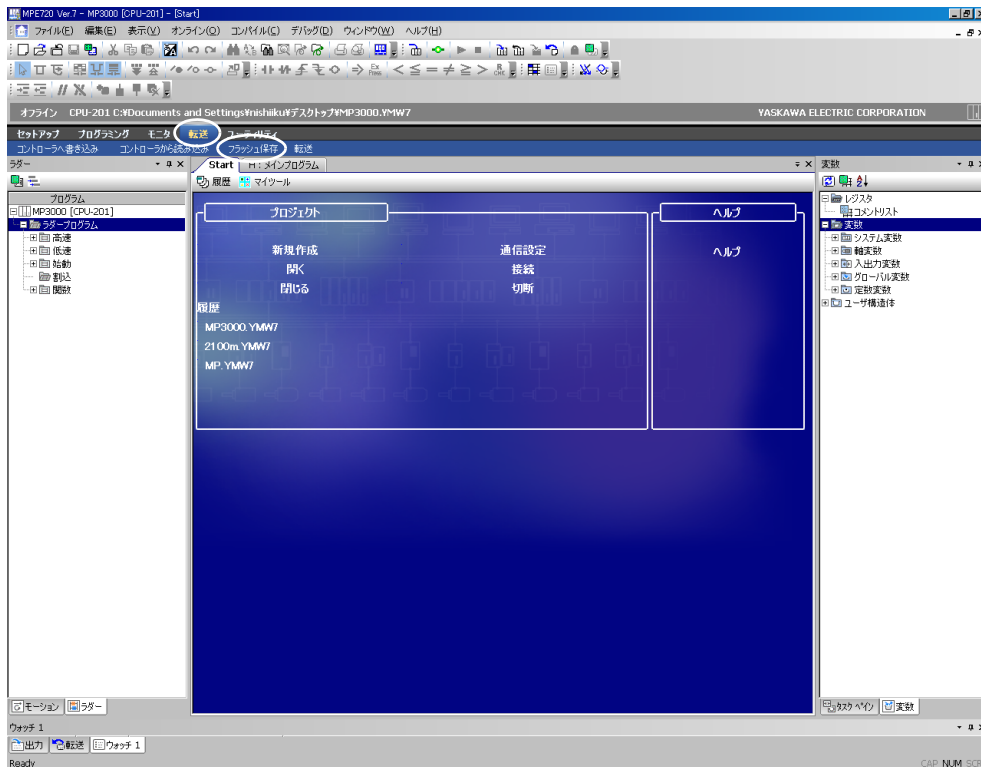
レジスタ	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
MB000000	ON	ON	ON	OFF	ON	OFF	OFF	OFF	ON	ON	ON	OFF	OFF	ON	OFF	OFF
MB000010	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF	OFF
MB000020	OFF	OFF	OFF	OFF	ON	OFF	OFF	OFF	ON	ON	ON	OFF	OFF	ON	OFF	OFF
MB000030	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

確認變為ON。

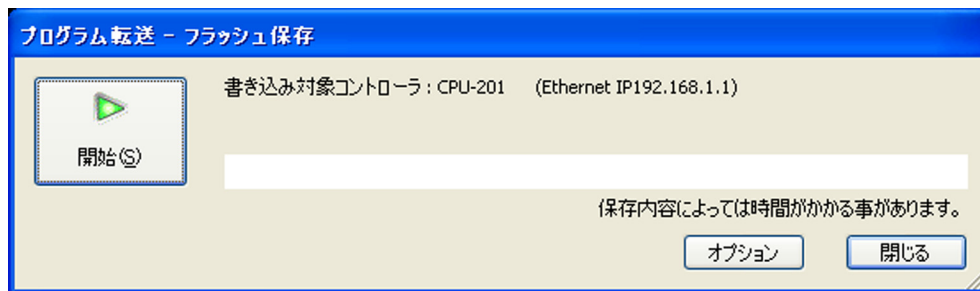
## 2.9 快閃儲存階梯圖程式

以下列步驟將運動控制器 RAM 內的資料，儲存至運動控制器的快閃記憶體內。

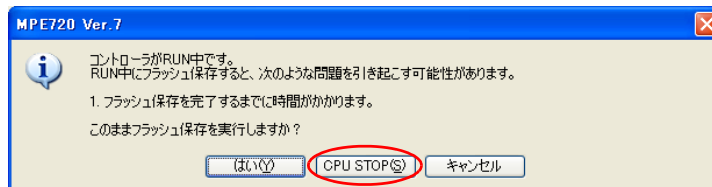
### 1. 選擇啟動器的 [ 傳送 ] – [ 快閃儲存 ]。



### 2. 按下 [ 開始 ] 按鈕。

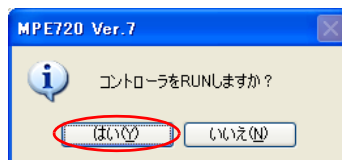


### 3. 點選 [ CPU STOP ] 按鈕。



開始儲存到快閃記憶體。

### 4. 點選 [ 是 ] 按鈕。



運動控制器 RUN。



重要

對運動控制器寫入資料後，請務必執行快閃儲存。  
不執行快閃儲存就重新接電，將恢復最後一次快閃儲存執行後的狀態，並且失去寫入的資料。



# 暫存器

---



本章將就暫存器進行說明。

暫存器 ..... 3-2

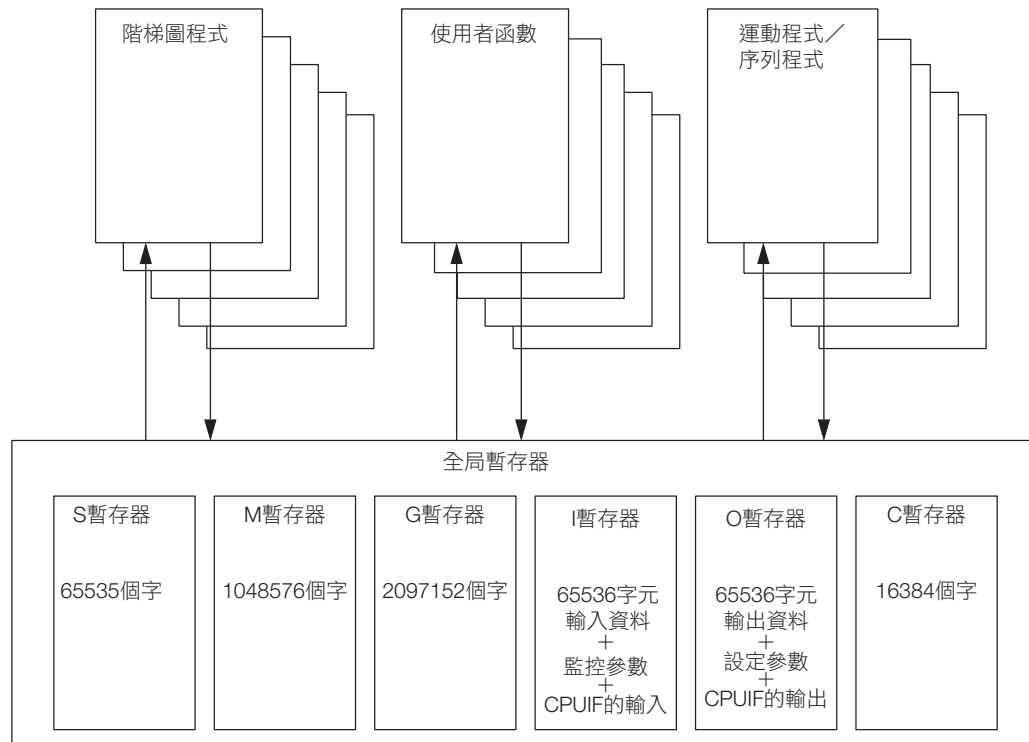
## 暫存器

暫存器為運動控制器內部儲存資料的地方。變數指暫存器與名稱（變數名稱）對應者。

暫存器中分為全部程式共同使用的總體暫存器，與各程式個別使用的局部暫存器。

### 全局暫存器

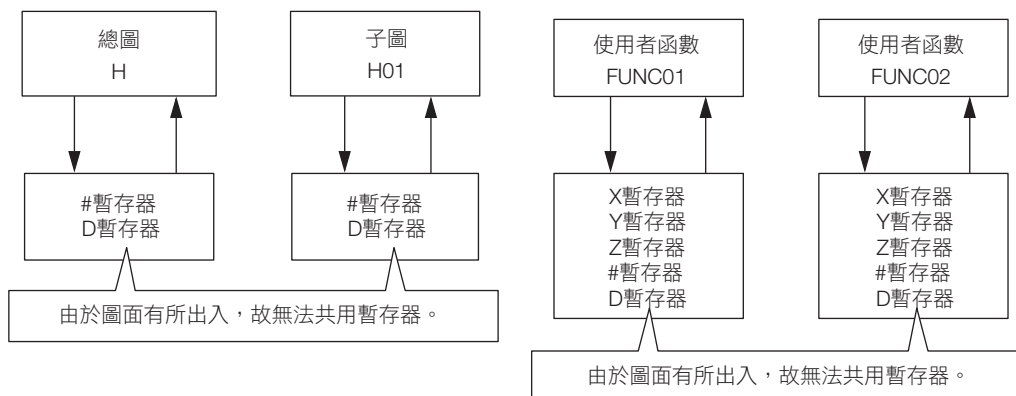
全局暫存器是階梯圖程式、使用者函數、運動程式、序列程式的各程式可共用的暫存器。總體暫存器的大小，依暫存器個別固定於系統中。



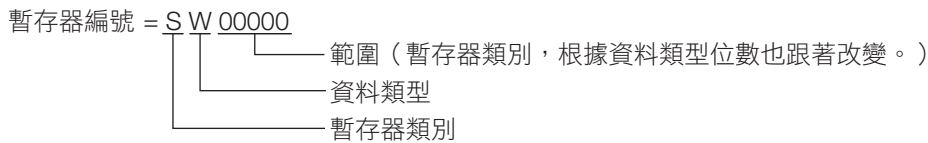
### 局部暫存器

局部暫存器為圖面分別可使用之暫存器。無法自其他圖面使用。

< 階梯圖程式示意圖 >



## 暫存器編號之組成



### 補充

組合做為變數使用的索引暫存器或陣列暫存器，亦可指定暫存器。  
詳情請參照如下內容。

☞ 索引暫存器 (i、j) (第 3-10 頁)

☞ 陣列暫存器 ([ ]) (第 3-12 頁)

## 暫存器的種類

此處就總體暫存器及局部暫存器之種類分別說明如下。

### ◆ 全局暫存器

全局暫存器是階梯圖程式、使用者函數、運動程式、序列程式的各程式可共用的暫存器。也就是說，某階梯圖程式演算結果可做為其他使用者函數及運動程式、序列程式等使用。

類型	名稱	指定方法	可利用範圍	內容
S	系統暫存器 (S 暫存器)	SBnnnnnh, SWnnnnn, SLnnnnn, SQnnnnn, SFnnnnn, SDnnnnn, SAnnnnn	SW00000 ~ SW65534	系統準備的暫存器。回報運動控制器的狀態等。 系統啟動時，SW00000 ~ SW00049 將消除為 0。 會進行電池組備用。
M	資料暫存器 (M 暫存器)	MBnnnnnnnh, MWnnnnnnn, MLnnnnnnn, MQnnnnnnn, MFnnnnnnn, MDnnnnnnn, MAnnnnnnn	MW0000000 ~ MW1048575	程式間的 I/F 等所使用的暫存器。 會進行電池組備用。
G	G 暫存器	GBnnnnnnnh, GWnnnnnnn, GLnnnnnnn, GQnnnnnnn, GFnnnnnnn, GDnnnnnnn, GAnnnnnnn	GW0000000 ~ GW2097151	程式間的 I/F 等所使用的暫存器。 未以電池組備用。
I	輸入暫存器 (I 暫存器)	IBhhhhhh, IWhhhhhh, ILhhhhhh, IQhhhhhh, IFhhhhhh, IDhhhhhh, IAhhhhhh,	IW00000 ~ IW07FFF, IW10000 ~ IW17FFF	輸入資料使用的暫存器。
			IW08000 ~ IW0FFFF, IW18000 ~ IW1FFFF	運動監控參數。 運動模組所使用的暫存器。
			IW20000 ~ IW23FFF	CPUIF 輸入資料使用的暫存器。

(接下頁)

(承上頁)


類型	名稱	指定方法	可利用範圍	內容
O	輸出暫存器 (O 暫存器)	OBhhhhh, OWhhhhh, OLhhhhh, OQhhhhh, OFhhhhh, ODhhhhh, OAhhhh,	OW00000 ~ OW07FFF, OW10000 ~ OW17FFF	輸出資料使用之暫存器。
			OW08000 ~ OW0FFFF, OW18000 ~ OW1FFFF	運動設定參數。 運動模組所使用的暫存器。
			OW20000 ~ OW23FFF	CPUIF 輸出資料使用的暫存器。
C	常數暫存器 (C 暫存器)	CBnnnnnh, CWnnnnn, CLnnnnn, CQnnnnn, CFnnnnn, CDnnnnn, CAnnnnn	CW00000 ~ CW16383	程式中僅供查看的暫存器。 可用 MPE720 設定其值。

(註)n: 10 進制, h: 16 進制

### ◆ 局部暫存器

依程式別固有之暫存器。無法參照其他程式的局部暫存器。

可利用範圍於 MPE720 由使用者指定。

類型	名稱	指定方法	內容	特性
#	# 暫存器	#Bnnnnnh, #Wnnnnn, #Lnnnnn, #Qnnnnn, #Fnnnnn, #Dnnnnn, #Annnnn	程式中僅供查看的暫存器。 可用 MPE720 設定其值。	程式個別
D	D 暫存器	DBnnnnnh, DWnnnnn, DLnnnnn, DQnnnnn, DFnnnnn, DDnnnnn, DAnnnnn	程式內泛用的暫存器。 預設每 1 程式可使用 32 字。 電源再次啟動時的初始值取決於 [ 啟動時 D 暫存器清除 ] 選項的設定。 詳情請參照如下內容。  • [ 啟動時清除 D 暫存器 ] 選項的設定方法 ( 第 3-6 頁 )	
X	函數輸入暫存器	XBnnnnnh, XWnnnnn, XLnnnnn, XQnnnnn, XFnnnnn, XDnnnnn	函數輸入 • 位元輸入: XB000000 ~ XB00000F • 整數輸入: XW00001 ~ XW00016 • 2 倍長整數: XL00001 ~ XL00015 • 4 倍長整數: XQ00001 ~ XQ00013 • 實數: XF00001 ~ XF00015 • 2 倍精度實數: XD00001 ~ XD00013	函數個別
Y	函數輸出暫存器	YBnnnnnh, YWnnnnn, YLnnnnn, YQnnnnn, YFnnnnn, YDnnnnn	自函數輸出 • 位元輸出: YB000000 ~ YB00000F • 整數輸出: YW00001 ~ YW00016 • 2 倍長整數: YL00001 ~ YL00015 • 4 倍長整數: YQ00001 ~ YQ00013 • 實數: YF00001 ~ YF00015 • 2 倍精度實數: YD00001 ~ YD00013	

(接下頁)



(承上頁)

類型	名稱	指定方法	內容	特性
Z	函數內部暫存器	ZBnnnnnh, ZWnnnnn, ZLnnnnn, ZQnnnnn, ZFnnnnn, ZDnnnnn	各函數固有的內部暫存器。可做為函數內部處理使用。 <ul style="list-style-type: none"> <li>• 位元：ZB000000 ~ ZB00063F</li> <li>• 整數：ZW000000 ~ ZW00063</li> <li>• 2 倍長整數：ZL000000 ~ ZL00062</li> <li>• 4 倍長整數：ZQ000000 ~ ZQ00060</li> <li>• 實數：ZF000000 ~ ZF00062</li> <li>• 2 倍精度實數：ZD000000 ~ ZD00060</li> </ul>	函數個別
A	函數外部暫存器	ABnnnnnh, AWnnnnn, ALnnnnn, AQnnnnn, AFnnnnn, ADnnnnn	以位址輸入值為標準位址的外部暫存器。 函數呼叫元，提供 M 暫存器及 D 暫存器位址輸入值，讓函數內部以該位址為標準的函數呼叫元的暫存器可被參照。	

(註) n：10 進制，h：16 進制




使用者函數可從不同的程式引用，次數不限。

重要

### ■ 於使用者函數使用局部暫存器時的注意事項

呼叫使用者函數時，需考量局部暫存器之數值，並進行初始化處理。

名稱	注意事項
X 暫存器 (函數輸入暫存器)	輸入值未被設定時，數值將不固定。 請勿使用輸入定義指定範圍外的 X 暫存器。
Y 暫存器 (函數輸出暫存器)	輸出值未被設定時，數值將不固定。 請務必要在輸出定義指定範圍的 Y 暫存器設定一數值。
Z 暫存器 (函數內部暫存器)	函數被呼叫時，前次設定之數值將消失，變為非固定值。 不適用維持前次數值之指令。 請一定要在函數內設定初始值後再行使用。
# 暫存器	常數暫存器數，數值不變。
D 暫存器	函數被呼叫時，將開啟前次設定值。 不想使用前次設定之數值時，請設定初始值或使用 Z 暫存器。 D 暫存器在電源關閉前將維持資料存在。 電源再次啟動時的初始值取決於 [ 啟動時 D 暫存器清除 ] 選項的設定。詳情請參照如下內容。  • [ 啟動時清除 D 暫存器 ] 選項的設定方法 (第 3-6 頁)

- [ 啟動時清除 D 暫存器 ] 選項的設定方法

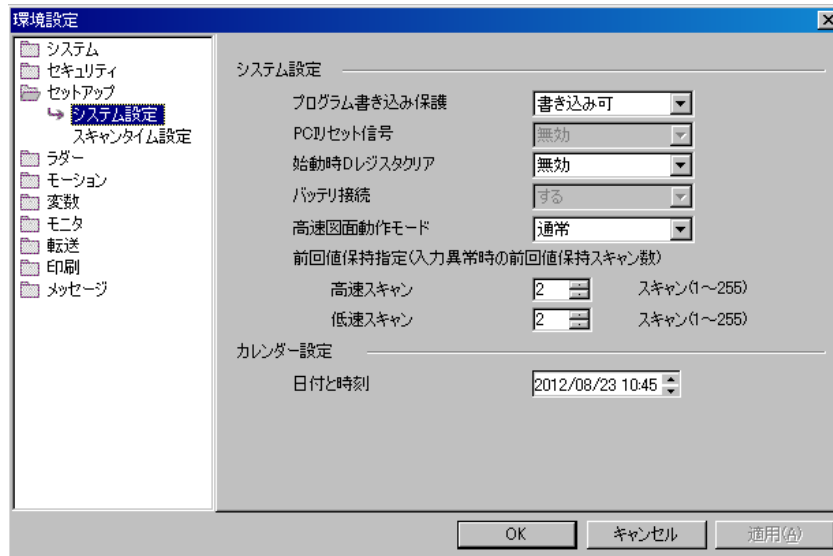
1. MPE720Ver.7 的視窗下選取 [ 檔案 ] 的 [ 環境設定 ]。

2. 選取 [ 設定 ]-[ 系統設定 ]。

3. 選取啟動時清除 D 暫存器的 [ 無效/有效 ]。

無效：初始值不固定

有效：初期值 0



## 資料類型

資料類型如下所示，共分為位元型、整數型、2 倍長整數型、4 倍長整數型、實數型、2 倍精度實數型、位址型，可依目的別使用。

記號	資料類型	數值範圍	資料尺寸	備註
B	位元	1 (ON), 0 (OFF)	-	使用於繼電器電路及 ON / OFF 條件之判定。
W	整數	-32,768 ~ 32,767 (8000H ~ 7FFFH)	1 個字	於數值運算中使用。左邊記述的 ( ) 裡指的是，顯示在邏輯運算中使用的狀況。
L	2 倍長整數	-2,147,483,648 ~ 2,147,483,647 (80000000H ~ 7FFFFFFFH)	2 個字	於數值運算中使用。左邊記述的 ( ) 裡指的是，顯示在邏輯運算中使用的狀況。
Q	4 倍長整數 <sup>*1</sup>	-9223372036854775808 ~ 9223372036854775807 (8000000000000000H ~ 7FFFFFFFFFFFFFFFH)	4 個字	於數值運算中使用。左邊記述的 ( ) 裡指的是，顯示在邏輯運算中使用的狀況。
F	實數	±(1.175E-38 ~ 3.402E+38), 0	2 個字	於高度數值運算中使用。 <sup>*2</sup>
D	2 倍精度實數	±(2.225E-308 ~ 1.798E+308), 0	4 個字	於高度數值運算中使用。 <sup>*2</sup>
A	位址	0 ~ 2,097,152	-	僅限指定指針時使用。

\*1. 無法使用於運動程式之間接指定。

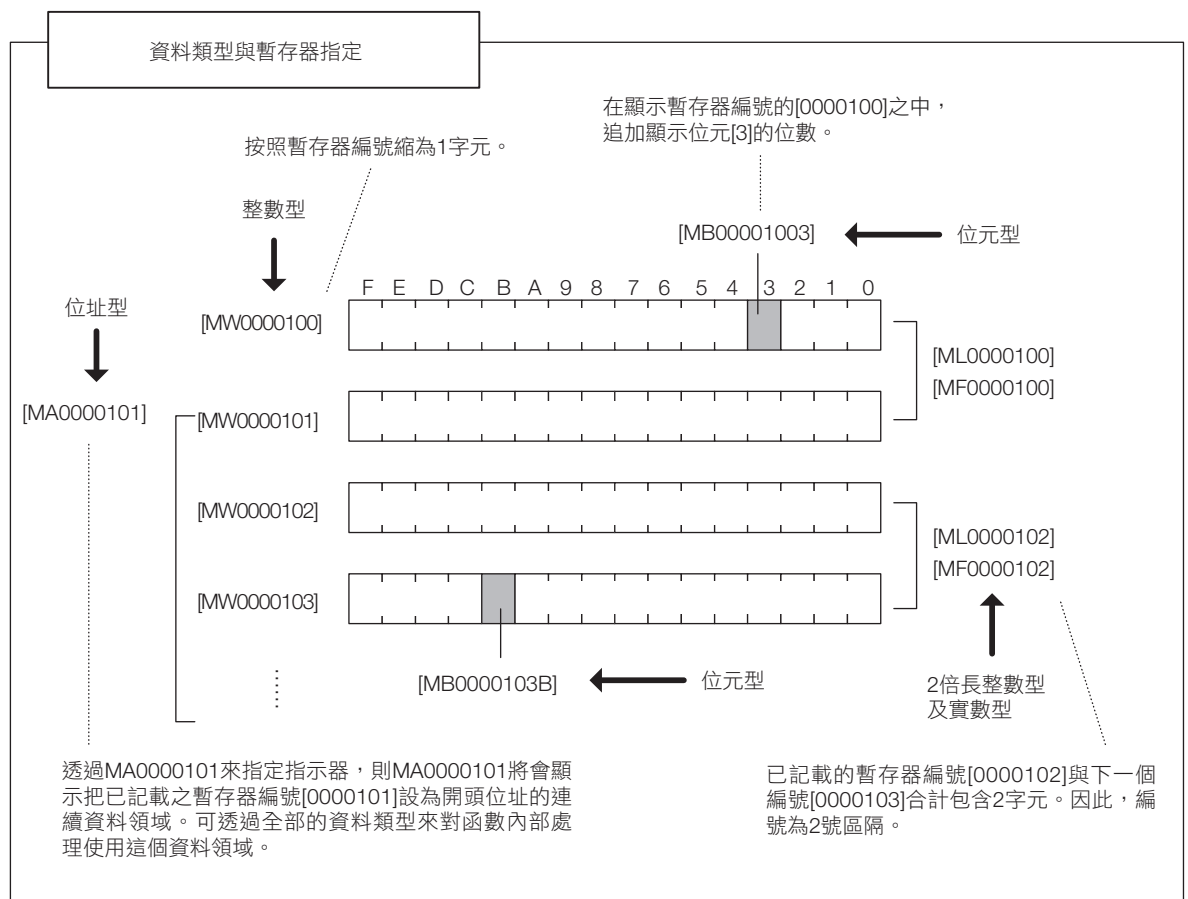
\*2. 依 IEEE754 為準。



重要

MP3000 系列中，並未依不同資料類型準備個別暫存器。如下圖所示，不同的資料類型，只要位址相同，就可以連接到相同的暫存器。

例如，位元型的 MB00001003 與整數型 MW0000100 雖然資料類型不同，但皆可連接至 MW0000100。





術語解說

**指定指針**

將位址做為參數交付至函數，稱為指定指針。

一旦指定完指針，以暫存器編號做為起始位址的連續資料區域，可做為全部的資料類型使用在函數的內部處理。

## ◆ 不同資料類型的演算注意事項

執行不同資料類型之演算時，依據檔案暫存器的檔案類型，將產生以下不同的結果，請注意。

## • 實數型資料儲存於整數型暫存器

MW0000100 = MF0000200；實數值轉換為整數後儲存。

(00001) (1.234)

(註)實數資料儲存於整數型暫存器時，請注意捨入誤差。

從實數到整數的處理方法（無條件捨去／四捨五入），在圖面的屬性項下可進行設定。

■ 實數轉換（CAST）時的動作設定（第 3-9 頁）

MW0000100 = MF0000200 + MF0000202；

(0124) (123.48) (0.02) 演算結果會因演算的變數值而異。

(0123) (123.49) (0.01)

## • 實數型資料儲存於 2 倍長整數型暫存器

ML0000100 = MF0000200；實數值轉換為整數後儲存。

(65432) (65432.1)

## • 2 倍長整數型資料儲存於整數型暫存器

MW0000100 = ML0000200；2 倍長整數型資料的下位 16 位元直接儲存。

(-00001) (65535)

## • 整數型資料儲存於 2 倍長整數型暫存器

ML0000100 = MW0000200；整數型資料轉換成 2 倍長整數型資料後儲存。

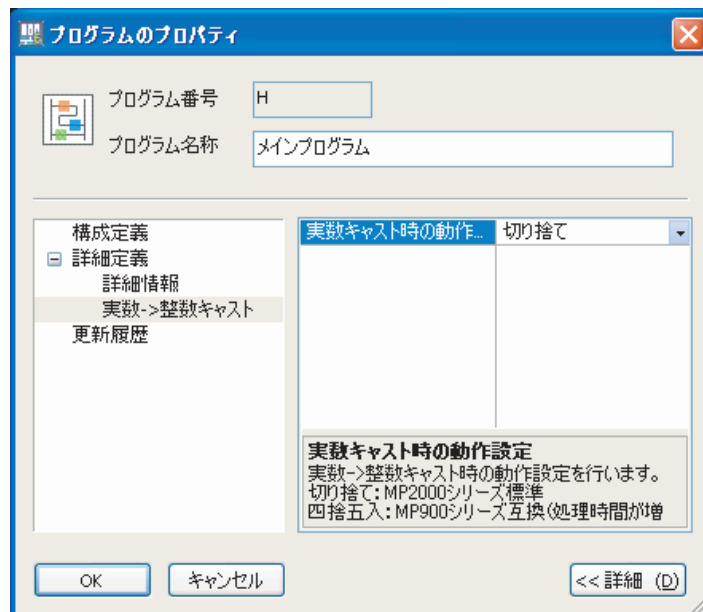
(0001234) (1234)

## ■ 實數轉換（CAST）時的動作設定

程式屬性的詳細定義中可設定實數轉換（CAST）時的動作（無條件捨去／四捨五入）。  
實數轉換 (CAST) 時的動作可依圖面別設定。

以下為 [ 程式屬性 ] 對話方塊的開啟流程。

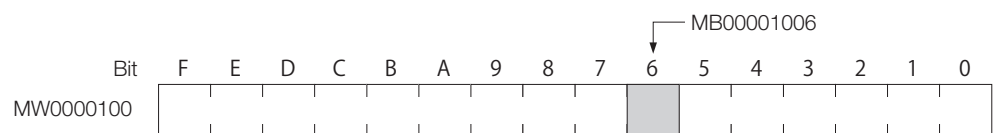
1. 在階梯圖視窗中選取顯示屬性的階梯圖程式檔案。
2. 在被選取的程式點擊滑鼠右鍵，從彈出的選單中選取 [ 屬性 ]。  
出現 [ 程式屬性 ] 對話方塊。



### 補充

資料の排列如以下範例所示，以小端讀取表示。

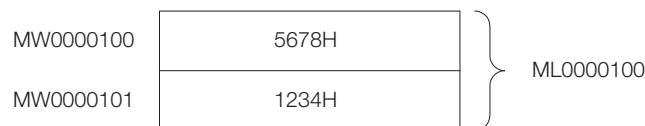
- MB00001006 時



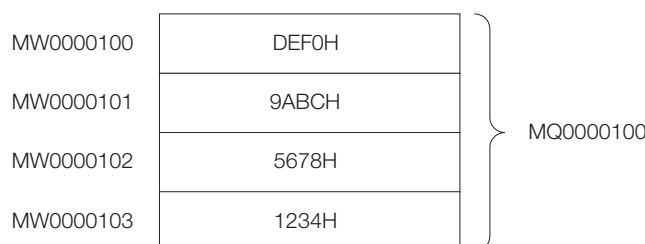
- MW0000100 = 1234H 時



- ML0000100 = 12345678H 時



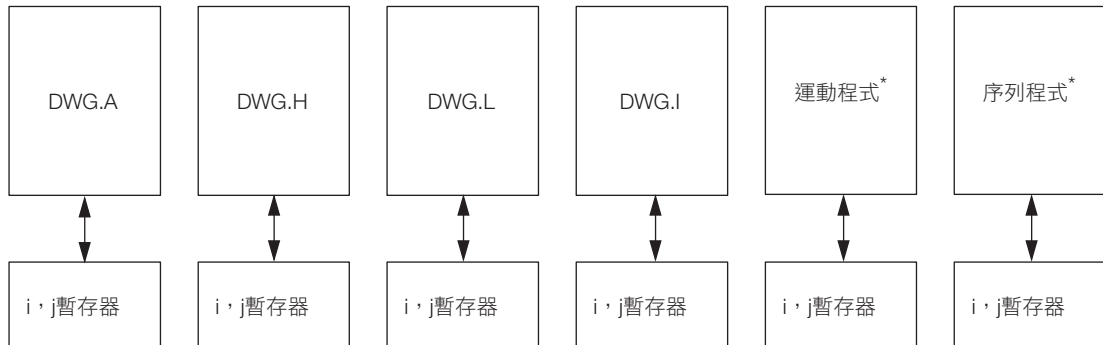
- MQ0000100 = 123456789ABCDEF0H 時



## 索引暫存器 (i、j)

修飾繼電器編號及暫存器編號的專用暫存器，共有 i 及 j 2 種。i 與 j 的功能完全相同。暫存器編號做為變數使用時。

索引暫存器如下圖所示，依程式種類區分。



\* 運動程式及序列程式可依任務維持個別的 i 及 j 暫存器。

(註) 函數參照被開啟圖面用的 i、j 暫存器。

例如，DWG.H 叫出的函數，參照 DWG.H 用的 i、j 暫存器。

以下分別以不同的暫存器的資料類型為例說明索引暫存器的動作。

### ◆ 位元型有添字時

和暫存器編號加計 i 或 j 時的值相同。

例如，i = 2 時的 MB00000000i 與 MB00000002 相同。

i = 2;  
 DB000000 = MB00000000i;  $\longleftrightarrow$  DB000000 = MB00000002;

### ◆ 整數型有添字時

和暫存器編號加計 i 或 j 時的值相同。

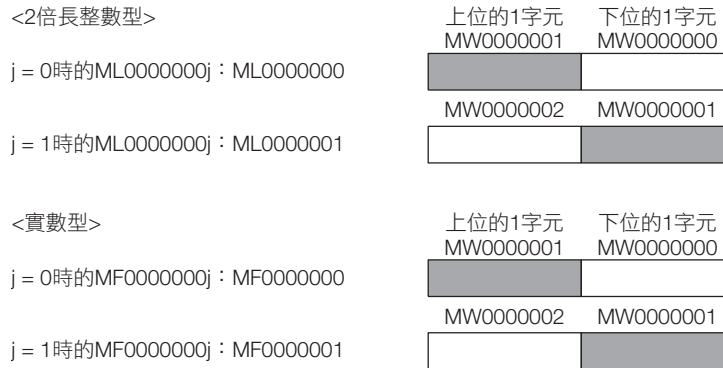
例如，j = 30 時的 MW0000001j 與 MW00000031 相同。

j = 30;  
 DW000000 = MW0000001j;  $\longleftrightarrow$  DW000000 = MW00000031;

### ◆ 2 倍長整數型及實數型有添字時

和暫存器編號加計 i 或 j 時的值相同。

例如，j = 1 時的 ML0000000j 與 ML0000001 相同。另外，j = 1 時的 MF0000000j 與 MF0000001 相同。



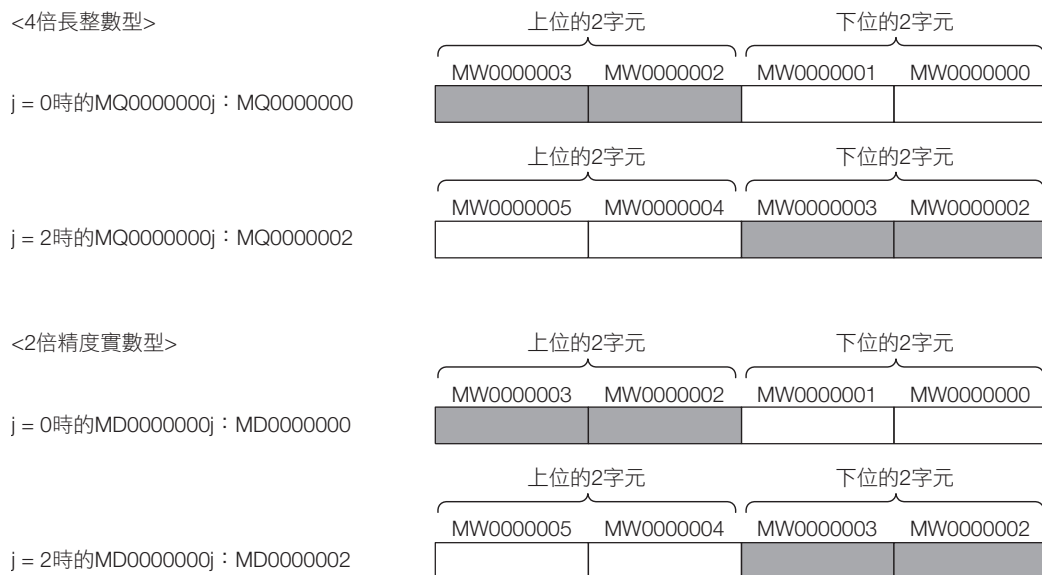
註記

2 倍長整數型、實數型時，使用 2 個字的區域。例如，j = 0 時的 ML0000000j 與 j = 1 時的 ML0000000j，MW0000001 的 1 個字區域會重覆。2 倍長整數型、實數型有添字時，請注意區域是否重覆。

### ◆ 4 倍長整數型及 2 倍精度實數型有添字時

和暫存器編號加計 i 或 j 時的值相同。

例如，j = 2 時的 MQ0000000j 與 MQ0000002 相同。此外，j = 2 時的 MD0000000j 與 MD0000002 相同。



註記

4 倍長整數型、2 倍精度實數型時，使用 4 個字區域。例如，j = 0 時的 MQ0000000j 與 j = 2 時的 MQ0000000j，MW0000002，MW0000003 的 2 個字區域重覆。4 倍長整數型、2 倍精度實數型有添字時，請注意區域有無重覆。

## 陣列暫存器 ([ ])

陣列暫存器係為修飾暫存器編號的專用暫存器，使用符號中的 [ ]。

用於想將暫存器編號做為變數使用時。

與索引暫存器相同，可在暫存器編號加入偏移量。

### ◆ 位元型有陣列時

與暫存器編號加入陣列值者相同。

例如，DW00000 = 2 時的 MB00000000[DW00000] 與 MB00000002 相同。

DW00000 = 2;  
DB000020 = MB00000000[DW00000];  $\longleftrightarrow$  DB000020 = MB00000002;

### ◆ 位元型以外有陣列時

與暫存器編號加入（排列值 × 資料型的字元大小）者相同。

例如，DW00000 = 30 時的 ML00000002[DW00000] 與 ML00000062 相同。

DL00002 = ML00000 (30×2 + 2) = ML0000062

DW00000 = 30;  
DL00002 = ML0000002[DW00000];  $\longleftrightarrow$  DL00002 = ML0000062;



# 梯形圖語言指令

# 4

本章將就階梯語言指令詳細進行說明。

## 4.1 概要 .....4-5

階梯語言指令一覽 .....	4-5
梯形圖語言指令的解讀 .....	4-8

## 4.2 繼電器電路指令 .....4-9

A 接點 (NOC) .....	4-9
上緣觸發 A 接點 (ONP-NOC) .....	4-10
下降沿 A 接點 (OFFP-NOC) .....	4-11
B 接點 (NCC) .....	4-12
上緣觸發 B 接點 (ONP-NCC) .....	4-13
下降沿 B 接點 (OFFP-NCC) .....	4-14
通電延遲計時器 (TON (1 ms)) .....	4-15
斷電延遲計時器 (TOFF (1 ms)) .....	4-17
通電延遲計時器 (TON (10 ms)) .....	4-19
斷電延遲計時器 (TOFF (10 ms)) .....	4-21
通電延遲計時器 (TON (1 s)) .....	4-23
斷電延遲計時器 (TOFF (1 s)) .....	4-25
上升脈衝 (ON-PLS) .....	4-27
下降脈衝 (OFF-PLS) .....	4-29
線圈 (COIL) .....	4-31
反轉線圈 (REV-COIL) .....	4-32
上緣觸發變化偵測線圈 (ONP-COIL) .....	4-33
下降變化偵測線圈 (OFFP-COIL) .....	4-34
設定線圈 (S-COIL) .....	4-35
重置線圈 (R-COIL) .....	4-36

### 4.3 數值運算指令 ..... 4-37

儲存 (STORE) .....	4-37
加法 (ADD (+)) .....	4-39
擴展加法 (ADDX (++) .....	4-41
減法 (SUB (-)) .....	4-43
擴充減法 (SUBX (- -)) .....	4-45
乘法 (MUL (·)) .....	4-47
除法 (DIV (÷)) .....	4-49
整數餘數 (MOD) .....	4-51
實數餘數 (REM) .....	4-53
增量 (INC) .....	4-55
減量 (DEC) .....	4-57
增加時間 (TMADD) .....	4-59
減少時間 (TMSUB) .....	4-61
經過時間 (SPEND) .....	4-63
符號反轉 (INV) .....	4-66
1 的補數 (COM) .....	4-67
絕對值轉換 (ABS) .....	4-68
2 進制轉換 (BIN) .....	4-69
BCD 轉換 (BCD) .....	4-70
同位轉換 (PARITY) .....	4-71
ASCII 轉換 1 (ASCII) .....	4-72
ASCII 轉換 2 (BINASC) .....	4-74
ASCII 轉換 3 (ASCBIN) .....	4-76

### 4.4 邏輯運算／比較指令 ..... 4-78

邏輯乘積 (AND) .....	4-78
邏輯或 (OR) .....	4-80
互斥或 (XOR) .....	4-82
比較 (<) .....	4-84
比較 (≦) .....	4-85
比較 (=) .....	4-86
比較 (≠) .....	4-87
比較 (≧) .....	4-88
比較 (>) .....	4-89
範圍檢查 (RCHK) .....	4-90

### 4.5 程式控制指令 ..... 4-92

呼叫圖面 (SEE) .....	4-92
呼叫運動程式 (MSEE) .....	4-93
呼叫使用者函數 (FUNC) .....	4-95
直接輸入連續執行形 (INS) .....	4-96
直接輸出連續執行形 (OUTS) .....	4-98
執行擴展程式 (XCALL) .....	4-100
WHILE 文 (WHILE, END_WHILE) .....	4-101
FOR 文 (FOR, END_FOR) .....	4-103

IF 文 (IF, END_IF) .....	4-105
IF-ELSE 文 (IF, ELSE, END_IF) .....	4-107
公式敘述 (EXPRESSION) .....	4-109

#### **4.6 基本函數指令 .....4-111**

平方根 (SQRT) .....	4-111
正弦 (SIN) .....	4-113
餘弦 (COS) .....	4-115
正切 (TAN) .....	4-117
反正弦 (ASIN) .....	4-118
反餘弦 (ACOS) .....	4-119
反正切 (ATAN) .....	4-120
指數 (EXP) .....	4-121
自然對數 (LN) .....	4-122
常用對數 (LOG) .....	4-123

#### **4.7 資料移動指令 .....4-124**

位元向左旋轉 (ROTL) .....	4-124
位元向右旋轉 (ROTR) .....	4-126
位元傳送 (MOVB) .....	4-128
文字傳送 (MOVW) .....	4-130
替換傳送 (XCHG) .....	4-132
表初始化 (SETW) .....	4-134
位元組→文字展開 (BEXTD) .....	4-136
文字→位元組壓縮 (BPRESS) .....	4-138
搜尋資料 (BSRCH) .....	4-140
排序 (SORT) .....	4-142
位元向左位移 (SHFTL) .....	4-144
位元向右位移 (SHFTR) .....	4-146
文字複製 (COPYW) .....	4-148
位元組交換 (BSWAP) .....	4-150

#### **4.8 DDC 指令 .....4-151**

不感帶 A (DZA) .....	4-151
不感帶 B (DZB) .....	4-153
上下限值 (LIMIT) .....	4-155
PI 控制 (PI) .....	4-157
PD 控制 (PD) .....	4-162
PID 控制 (PID) .....	4-168
一次延遲 (LAG) .....	4-173
相位前進延遲 (LLAG) .....	4-176
函數產生器 (FGN) .....	4-179
反函數產生器 (IFGN) .....	4-184
直線加減速器 1 (LAU) .....	4-189
直線加減速器 2 (SLAU) .....	4-196
脈衝寬變調 (PWM) .....	4-206

## 4.9 表操作指令 ..... 4-209

區段讀取 (TBLBR) .....	4-209
區段寫入 (TBLBW) .....	4-212
行搜尋 (縱向) (TBLSRL) .....	4-215
列搜尋 (橫向) (TBLSRC) .....	4-218
刪除區段 (TBLCL) .....	4-221
傳送表間區段 (TBLMV) .....	4-224
讀取佇列表 (QTBLR, QTBLRI) .....	4-228
寫入佇列表 (QTBLW, QTBLWI) .....	4-232
佇列指標清除 (QTBLCL) .....	4-236

## 4.10 系統函數指令 ..... 4-238

計數器 (COUNTER) .....	4-238
先進先出 (FINFOUT) .....	4-241
追蹤 (TRACE) .....	4-245
讀取追蹤資料 (DTRC-RD) .....	4-247
資訊傳送 (MSG-SND) .....	4-251
資訊傳送 (擴展) (MSG-SNDE) .....	4-253
訊息接收 (MSG-RCV) .....	4-255
訊息接收 (擴展) (MSG-RCVE) .....	4-257
伺服單元參數寫入 (MLNK-SVW) .....	4-259
伺服單元參數載入 (MLNK-SVR) .....	4-264
快閃操作 (FLASH-OP) .....	4-269
寫入運動暫存器 (MOTREG-W) .....	4-273
載入運動暫存器 (MOTREG-R) .....	4-276
匯入 (IMPORT/IMPORTL) .....	4-279
匯出 (EXPORT/EXPORTL) .....	4-287

## 4.1

## 概要

將說明階梯語言指令的種類與其功能，以及本章的閱讀方式。

## 階梯語言指令一覽

階梯語言指令如次表所示。

種類	指令	含義
繼 電 器 電 路 指 令	NOC	A 接點
	ONP-NOC	上緣觸發 A 接點
	OFFP-NOC	下降沿 A 接點
	NCC	B 接點
	ONP-NCC	上緣觸發 B 接點
	OFFP-NCC	下降沿 B 接點
	TON (1 ms)	通電延遲計時器 (1ms)
	TOFF (1 ms)	斷電延遲計時器 (1ms)
	TON (10 ms)	通電延遲計時器 (10 ms)
	TOFF (10 ms)	斷電延遲計時器 (10 ms)
	TON (1 s)	通電延遲計時器 (1 s)
	TOFF (1 s)	斷電延遲計時器 (1 s)
	ON-PLS	上升脈衝
	OFF-PLS	下降脈衝
	COIL	線圈
	REV-COIL	反轉線圈
	ONP-COIL	上緣觸發變化偵測線圈
	OFFP-COIL	下降變化偵測線圈
	S-COIL	設定線圈
	R-COIL	重置線圈
數 值 運 算 指 令	STORE	儲存
	ADD(+)	加法
	ADDX(+ +)	擴展加法
	SUB(-)	減法
	SUBX(- -)	擴展減法
	MUL(x)	乘法
	DIV(+)	除法
	MOD	整數餘數
	REM	實數餘數
	INC	增量
	DEC	減量
	TMADD	增加時間
	TMSUB	減少時間
	SPEND	時間經過
	INV	符號反轉
	COM	1 的補數
	ABS	絕對值轉換
	BIN	2 進制轉換
	BCD	BCD 轉換
	PARITY	同位轉換
	ASCII	ASCII 轉換 1
	BINASC	ASCII 轉換 2
	ASCBIN	ASCII 轉換 3

(接下頁)

(承上頁)

種類	指令	含義
邏輯運算指令	AND	邏輯乘積
	OR	邏輯或
	XOR	互斥或
	<	比較
	≤	比較
	=	比較
	≠	比較
	≥	比較
	>	比較
	RCHK	範圍檢查
程式控制指令	SEE	叫出圖面
	MSEE	叫出運動程式
	FUNC	呼叫使用者函數
	INS	直接輸入連續執行形
	OUTS	直接輸出連續執行形
	XCALL	執行擴展程式
	WHILE END_WHILE	WHILE 文
	FOR END_FOR	FOR 文
	IF END_IF	IF 文
	IF ELSE END_IF	IF-ELSE 文
	EXPRESSION	公式敘述
	基本函數指令	SQRT
SIN		正弦
COS		餘弦
TAN		正切
ASIN		反正弦
ACOS		反餘弦
ATAN		反正切
EXP		指數
LN		自然對數
LOG		常用對數
資料操作指令		ROTL
	ROTR	位元向右旋轉
	MOVB	位元傳送
	MOVW	文字傳送
	XCHG	替換傳送
	SETW	表初始化
	BEXTD	位元組→文字展開
	BPRESS	文字→位元組壓縮
	BSRCH	資料搜尋
	SORT	篩選
	SHFTL	位元向左位移
	SHFTR	位元向右位移
	COPYW	文字複製
	BSWAP	位元組交換

(接下頁)

(承上頁)

種類	指令	含義
D D C 指令	DZA	不感帶 A
	DZB	不感帶 B
	LIMIT	上下限值
	PI	PI 控制
	PD	PD 控制
	PID	PID 控制
	LAG	一次延遲
	LLAG	相位前進延遲
	FGN	函數產生器
	IFGN	反函數產生器
	LAU	直線加減速器 1
	SLAU	直線加減速器 2
	PWM	脈衝寬變調
	表 操作 指令	TBLBR
TBLBW		區段寫入
TBLSRL		行搜尋 (縱向)
TBLSRC		列搜尋 (橫向)
TBLCL		刪除區段
TBLMV		傳送表間區段
QTBLR		讀取佇列表
QTBLRI		讀取佇列表 (指標步進)
QTBLW		寫入佇列表
QTBLWI		寫入佇列表 (指標步進)
QTBLCL		佇列表指標清除
系統 標準 函數 指令		COUNTER
	FINFOUT	先進先出
	TRACE	追蹤
	DTRC-RD	讀取追蹤資料
	MSG-SND	傳送訊息
	MSG-SNDE	傳送訊息 (擴展)
	MSG-RCV	接收訊息
	MSG-RCVE	接收訊息 (擴展)
	MLNK-SVW	寫入伺服單元的參數
	MLNK-SVR	讀取伺服單元的參數
	FLASH-OP	快閃記憶體操作
	MOTREG-W	寫入運轉暫存器
	MOTREG-R	載入動作暫存器
	IMPORT/IMPORTL	匯入
EXPORT/EXPORTL	匯出	

## 梯形圖語言指令的解讀

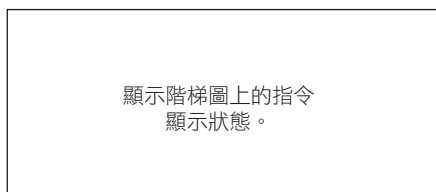
本章以下列架構，就各項指令進行說明。

說明指令的動作。

視需要刊載解說指令動作的圖示。

### 格式

說明指令的格式。



圖示：顯示在MPE720上的圖示。


鍵輸入：顯示階梯圖編輯器上的快捷鍵。

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸出輸入項目 (階梯圖上的標記)	×	○	○	×	○	×	×	○	○

(註) 1. × ... 無法使用該資料類型。

○ ... 所有暫存器均可使用該資料類型。

2. 關於資料類型詳情，請參照以下章節。

 3章 暫存器

### 程式範例

表示使用指令的階梯圖程式範例。

### 補充事項

敘述需要補充說明的事項。無須補充說明的指令，則無此一項目。



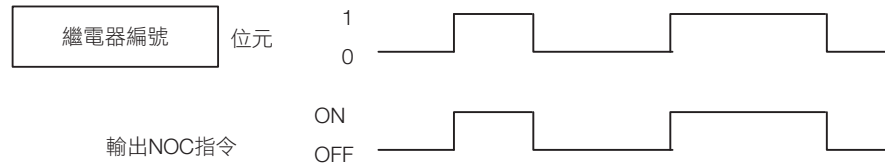
## 4.2

## 繼電器電路指令

## A 接點 (NOC)

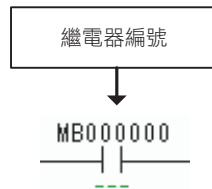
當繼電器編號位元為 1 時，繼電器輸出 ON。

當位元為 0，繼電器輸出 OFF。



## 格式

格式如下所示。



圖示：  
鍵輸入：[]

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
繼電器編號	○	×	×	×	×	×	×	×	×

## 程式範例

NOC 指令的繼電器 (DB000000) 為 ON 時，輸出線圈 (DB000001) 轉為 ON。



## 上緣觸發 A 接點 (ONP-NOC)

位元輸入由 0 變為 1 時，只有 1 掃描輸出 ON。

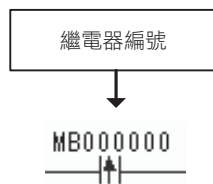
NOC 指令與 ON-PLS 指令結合的動作。

**補充**

- 與 OFFP-NCC 指令的動作相同。
- ONP-NOC 指令無法在使用者函數中使用。使用者函數當中，請使用 A 接點 + 上升脈衝 (ON-PLS)。

### 格式

格式如下所示。



圖示：  
 鍵輸入：JPI

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
繼電器編號	○*	×	×	×	×	×	×	×	×

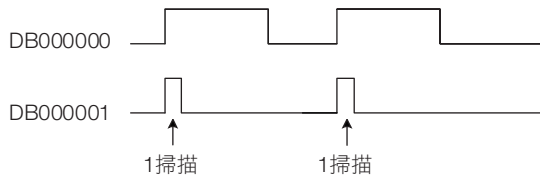
\* #、C 暫存器為常數暫存器，由於數值不會變化，因此沒有目標動作。

### 程式範例

NOC 指令的繼電器 (DB000000) 由 OFF 變為 ON 時，輸出線圈 (DB000001) 轉為 ON。



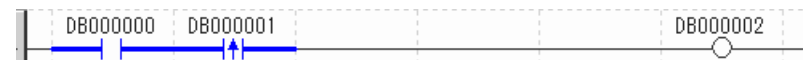
計時器圖表如下所示。



註記

ONP-NOC 指令之前如果組合了其他指令，將輸出與該指令的運算結果。

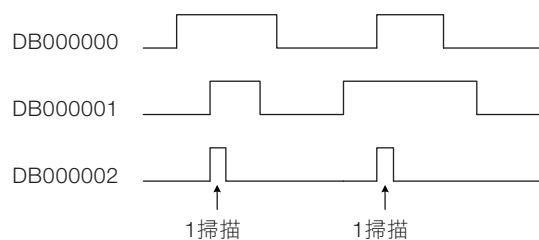
下列範例當中，DB000000 與 DB000001 的 AND 條件由 OFF 變為 ON 時，輸出線圈 (DB000002) 轉為 ON。



上述程式的等價迴路如下所示。



計時器圖表如下所示。



## 下降沿 A 接點 (OFFP-NOC)

位元輸入由 1 變為 0 時，只有 1 掃描輸出 ON。

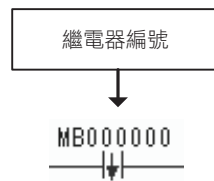
NOC 指令與 OFF-PLS 指令結合的動作。

### 補充

- 與 ONP-NCC 指令的動作相同。
- OFFP-NOC 指令無法在使用者函數中使用。使用者函數當中，請使用 A 接點 + 下降脈衝 (OFF-PLS)。

## 格式

格式如下所示。



圖示：  
鍵輸入：|N|

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
繼電器編號	○*	×	×	×	×	×	×	×	×

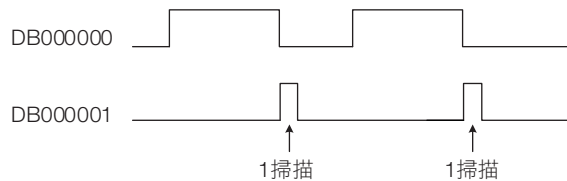
\* #、C 暫存器為常數暫存器，由於數值不會變化，因此沒有目標動作。

## 程式範例

NOC 指令的繼電器 (DB000000) 由 ON 變為 OFF 時，輸出線圈 (DB000001) 轉為 ON。



計時器圖表如下所示。



註記

OFFP-NOC 指令之前如果組合了其他指令，將輸出與該指令的運算結果。

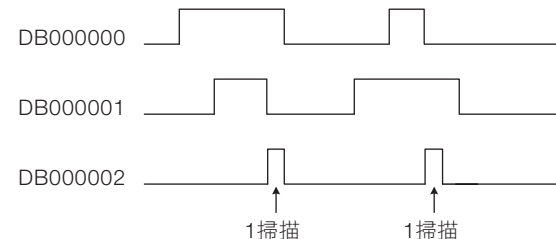
下列範例當中，DB000000 與 DB000001 的 AND 條件由 ON 變為 OFF 時，輸出線圈 (DB000002) 轉為 ON。



上述程式的等價迴路如下所示。



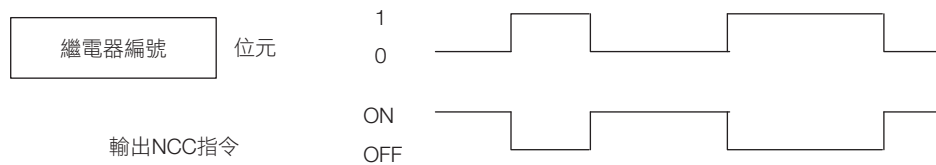
計時器圖表如下所示。



## B 接點 (NCC)

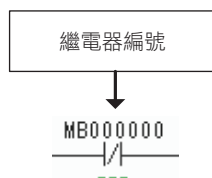
當繼電器編號位元為 1 時，繼電器輸出 OFF。

當位元為 0，繼電器輸出 ON。



### 格式

格式如下所示。

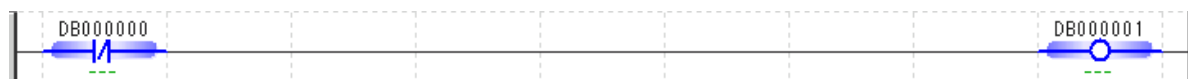


圖示：  
 鍵輸入：V

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
繼電器編號	○	×	×	×	×	×	×	×	×

### 程式範例

NCC 指令的繼電器 (DB000000) 為 OFF 時，線圈 (DB000001) 轉為 ON。



## 上緣觸發 B 接點 (ONP-NCC)

位元輸入由 1 變為 0 時，只有 1 掃描輸出 ON。

NCC 指令與 ON-PLS 指令結合的動作。


### 補充

- 與 OFFP-NOC 指令的動作相同。
- ONP-NCC 指令無法在使用者函數中使用。使用者函數當中，請使用 B 接點 + 上升脈衝 (ON-PLS)。

## 格式

格式如下所示。



圖示：  
鍵輸入：JP/

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
繼電器編號	○*	×	×	×	×	×	×	×	×

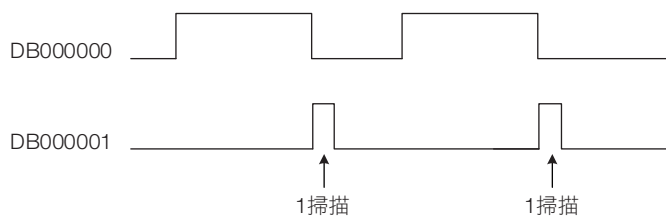
\* #、C 暫存器為常數暫存器，由於數值不會變化，因此沒有目標動作。

## 程式範例

NCC 指令的繼電器 (DB000000) 由 ON 變為 OFF 時，輸出線圈 (DB000001) 轉為 ON。



計時器圖表如下所示。



註記

ONP-NCC 指令之前如果組合了其他指令，將輸出與該指令的運算結果。

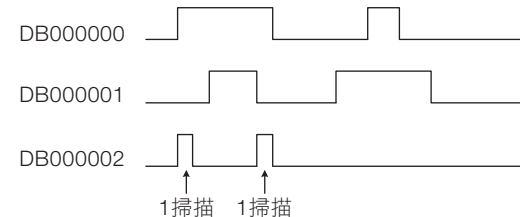
下列範例當中，DB000000 與 DB000001 反轉的 AND 條件由 OFF 變為 ON 時，輸出線圈 (DB000002) 轉為 ON。



上述程式的等價迴路如下所示。



計時器圖表如下所示。



## 下降沿 B 接點 (OFFP-NCC)

位元輸入由 0 變為 1 時，只有 1 掃描輸出 ON。

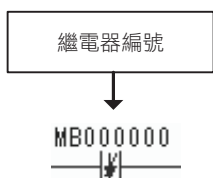
NCC 指令與 OFF-PLS 指令結合的動作。

**補充**

- 與 ONP-NOC 指令的動作相同。
- OFFP-NCC 指令無法在使用者函數中使用。使用者函數當中，請使用 B 接點 + 下降脈衝 (OFF-PLS)。

### 格式

格式如下所示。



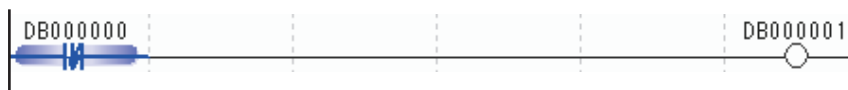
圖示：  
 鍵輸入：JN

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
繼電器編號	○*	×	×	×	×	×	×	×	×

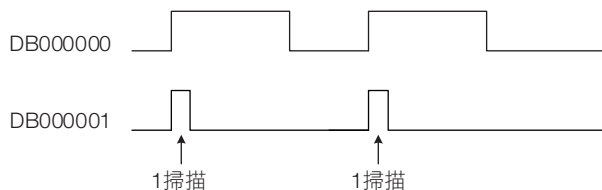
\* #、C 暫存器為常數暫存器，由於數值不會變化，因此沒有目標動作。

### 程式範例

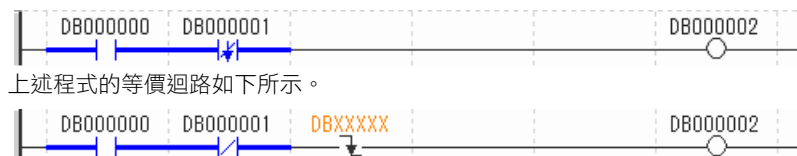
NCC 指令的繼電器 (DB000000) 由 OFF 變為 ON 時，輸出線圈 (DB000001) 轉為 ON。



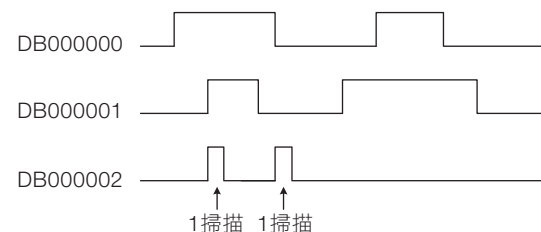
計時器圖表如下所示。



OFFP-NCC 指令之前如果組合了其他指令，將輸出與該指令的運算結果。  
 下列範例當中，DB000000 與 DB000001 反轉的 AND 條件由 ON 變為 OFF 時，輸出線圈 (DB000002) 轉為 ON。



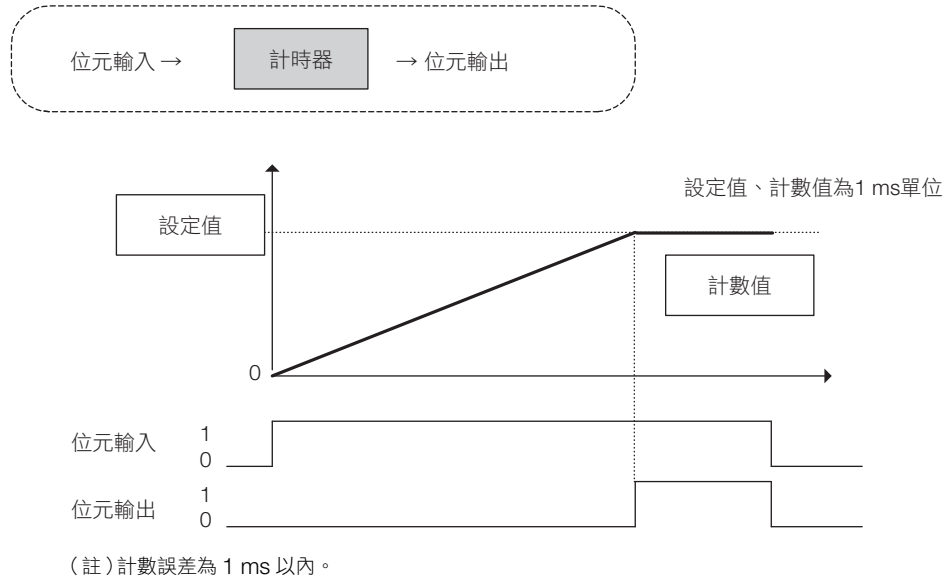
上述程式的等價迴路如下所示。



## 通電延遲計時器 (TON (1 ms))

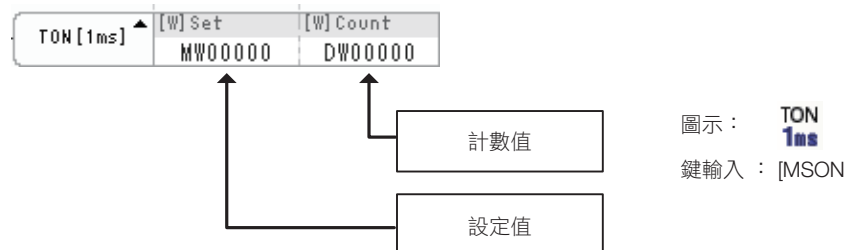
計時器的位元輸入為 1 時，進行時間計數。計數值與設定值相等時，位元輸出為 1。

若位元輸入在計數過程變為 0，則計時器動作停止。若是位元輸入還是 1，則計數從頭 (0) 開始。另外，實際的計數時間 (1 ms 單位) 的數值會儲存於計數用暫存器。



### 格式

格式如下所示。



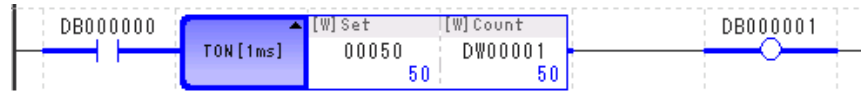
輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
設定值 (Set)	×	○	×	×	×	×	×	×	○
計數值 (Conut)	×	○*	×	×	×	×	×	×	×

\* C、# 暫存器除外

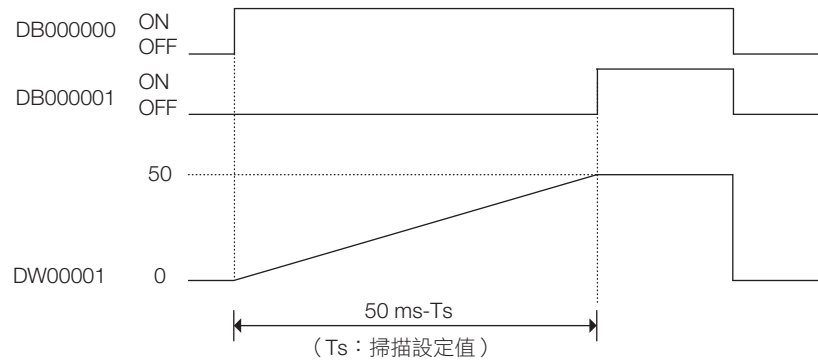
### 程式範例

通電延遲計時器的設定值為 50、計數值儲存於 DW00001 時的程式範例如下所示。

繼電器 (DB000000)ON，50 ms 後線圈 (DB000001)ON。



計時器圖表如下所示。

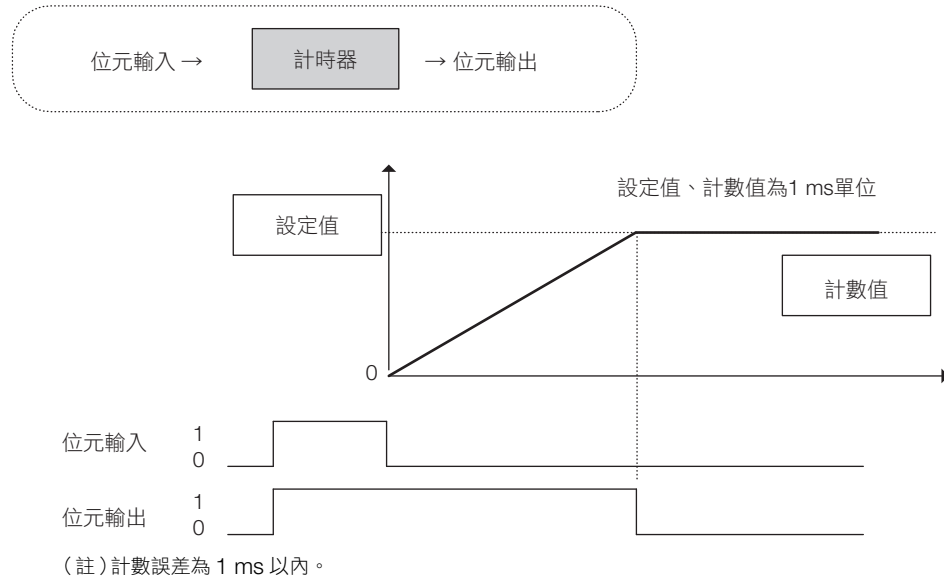




## 斷電延遲計時器 (TOFF (1 ms))

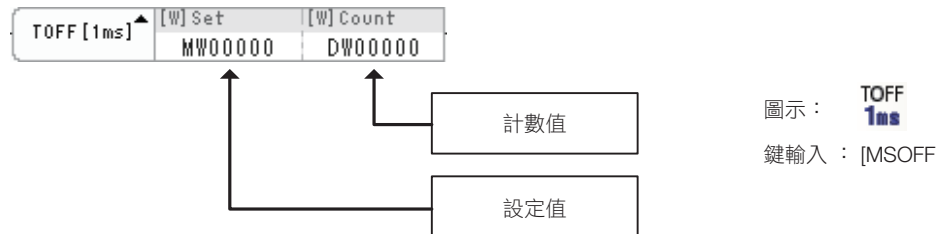
計時器的位元輸入為 0 時，進行時間計數。計數值與設定值相等時，位元輸出為 0。

若位元輸入在計數過程變為 1，則計時器動作停止。若是位元輸入還是 0，則計數從頭 (0) 開始。另外，實際的計數時間 (1 ms 單位) 的數值會儲存於計數用暫存器。



### 格式

格式如下所示。



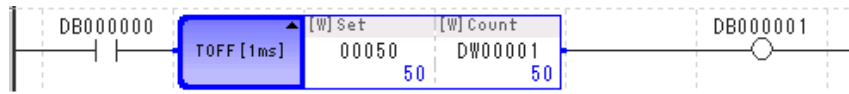
輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
設定值 (Set)	×	○	×	×	×	×	×	×	○
計數值 (Conut)	×	○*	×	×	×	×	×	×	×

\* C、# 暫存器除外

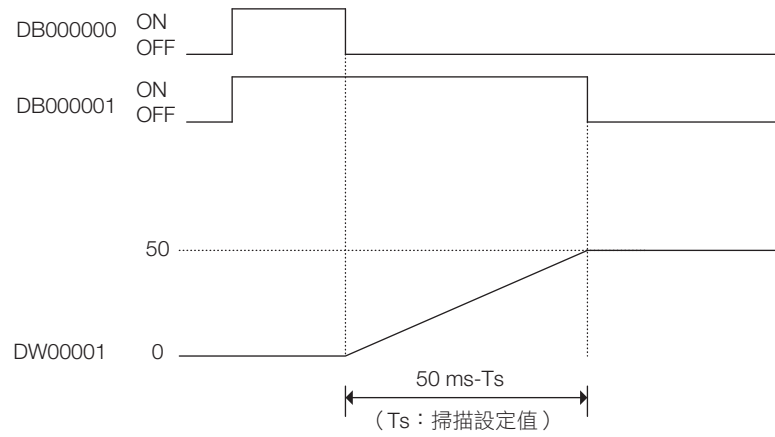
### 程式範例

斷電延遲計時器的設定值為 50、計數值儲存於 DW00001 時的程式範例如下所示。

繼電器 (DB000000)OFF，50 ms 後線圈 (DB000001)OFF。



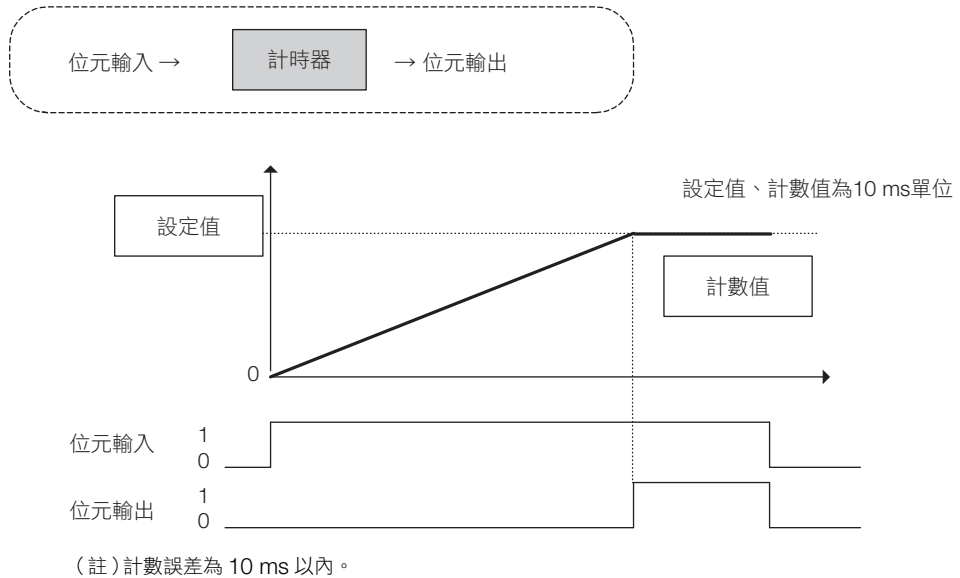
計時器圖表如下所示。



## 通電延遲計時器 (TON (10 ms))

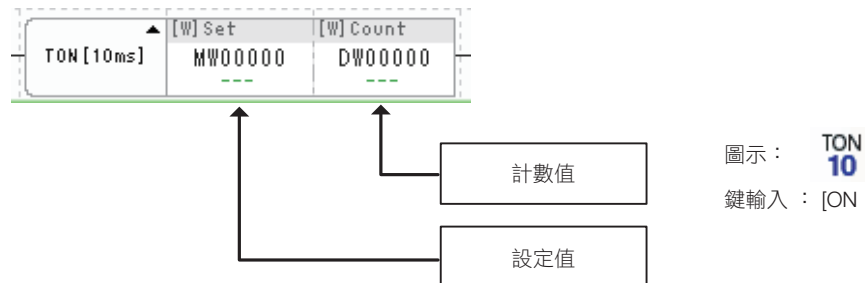
計時器的位元輸入為 1 時，進行時間計數。計數值與設定值相等時，位元輸出為 1。

若位元輸入在計數過程變為 0，則計時器動作停止。若是位元輸入還是 1，則計數從頭 (0) 開始。另外，實際的計數時間 (10 ms 單位) 的數值會儲存於計數用暫存器。



### 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
設定值 (Set)	×	○	×	×	×	×	×	×	○
計數值 (Conut)	×	○*	×	×	×	×	×	×	×

\* C、# 暫存器除外

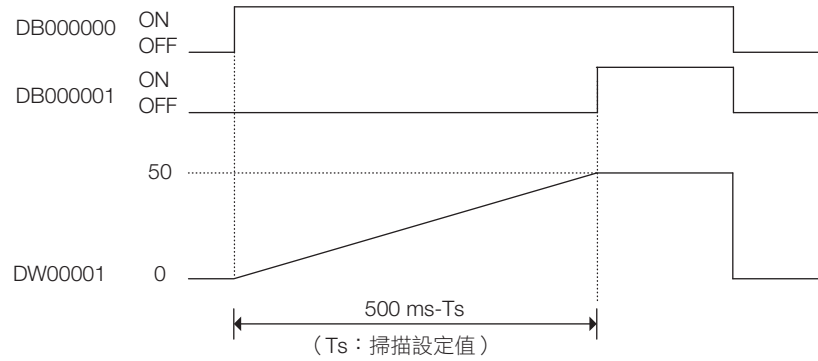
### 程式範例

通電延遲計時器的設定值為 50、計數值儲存於 DW00001 時的程式範例如下所示。

繼電器 (DB000000)ON，500 ms 後線圈 (DB000001)ON。



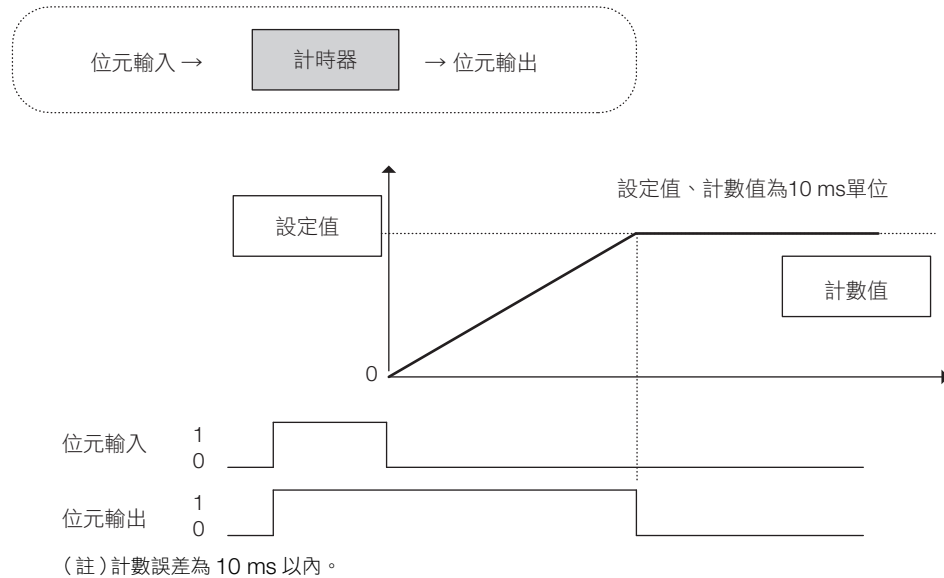
計時器圖表如下所示。



## 斷電延遲計時器 (TOFF (10 ms))

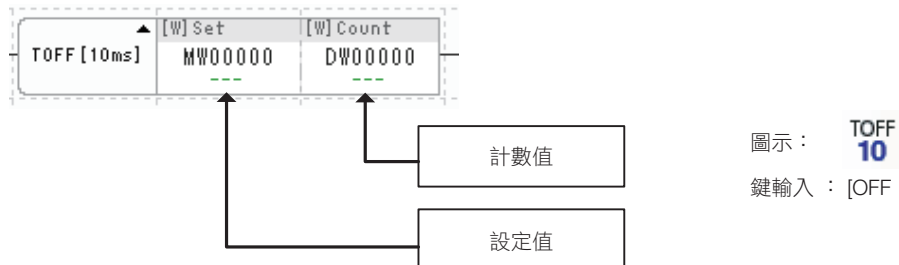
計時器的位元輸入為 0 時，進行時間計數。計數值與設定值相等時，位元輸出為 0。

若位元輸入在計數過程變為 1，則計時器動作停止。若是位元輸入還是 0，則計數從頭 (0) 開始。另外，實際的計數時間 (10 ms 單位) 的數值會儲存於計數用暫存器。



### 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
設定值 (Set)	×	○	×	×	×	×	×	×	○
計數值 (Conut)	×	○*	×	×	×	×	×	×	×

\* C、# 暫存器除外

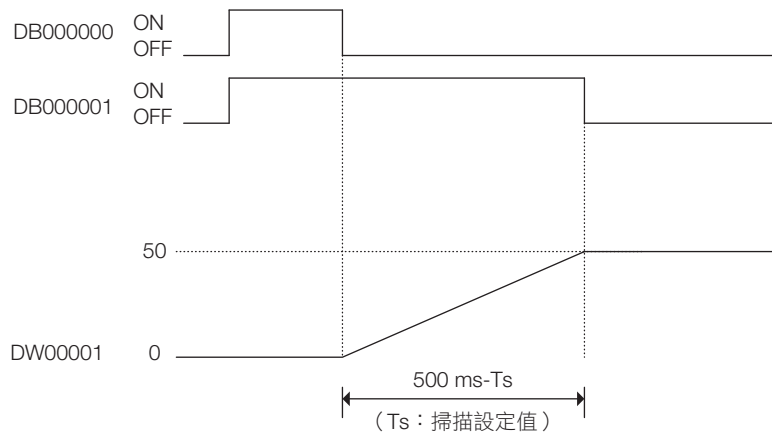
### 程式範例

斷電延遲計時器的設定值為 50、計數值儲存於 DW00001 時的程式範例如下所示。

繼電器 (DB000000)OFF，500 ms 後線圈 (DB000001)OFF。



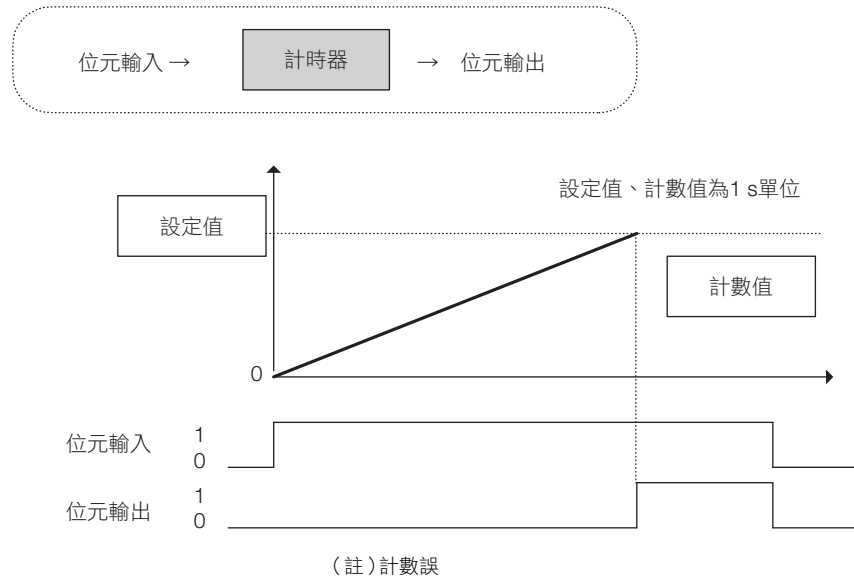
計時器圖表如下所示。



## 通電延遲計時器 (TON (1 s))

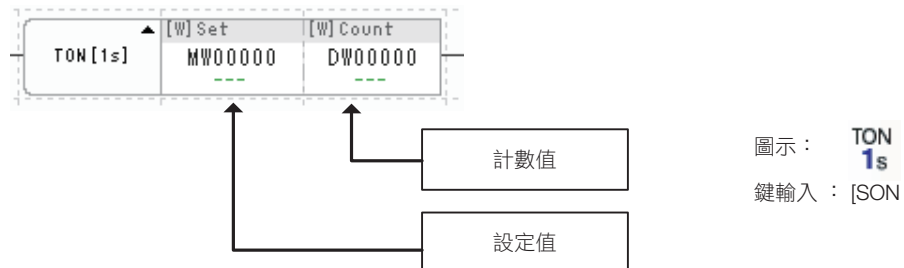
計時器的位元輸入為 1 時，進行時間計數。計數值與設定值相等時，位元輸出為 1。

若位元輸入在計數過程變為 0，則計時器動作停止。位元輸入再次為 1 時，計數則從最初的 0 開始。另外，實際的計數時間（1 s 單位）的數值會儲存於計數用暫存器。



### 格式

格式如下所示。



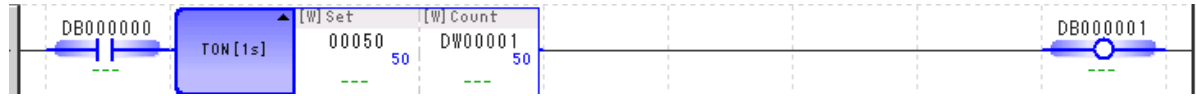
輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
設定值 (Set)	×	○	×	×	×	×	×	×	○
計數值 (Conut)	×	○*	×	×	×	×	×	×	×

\* C、# 暫存器除外

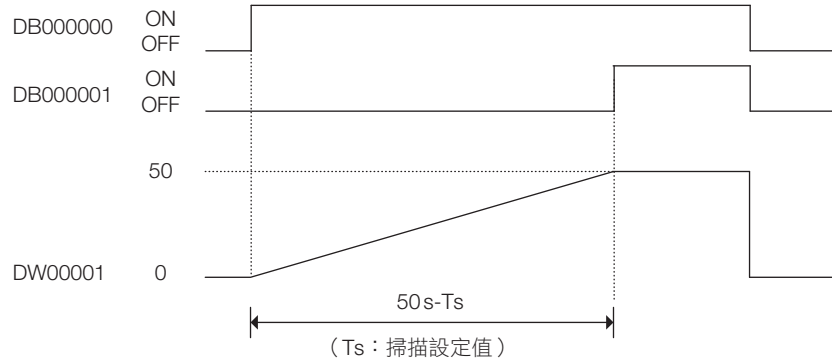
### 程式範例

通電延遲計時器的設定值為 50、計數值儲存於 DW00001 時的程式範例如下所示。

繼電器 (DB000000) 轉為 ON 的 50 秒後，線圈 (DB000001) 轉為 ON。



計時器圖表如下所示。

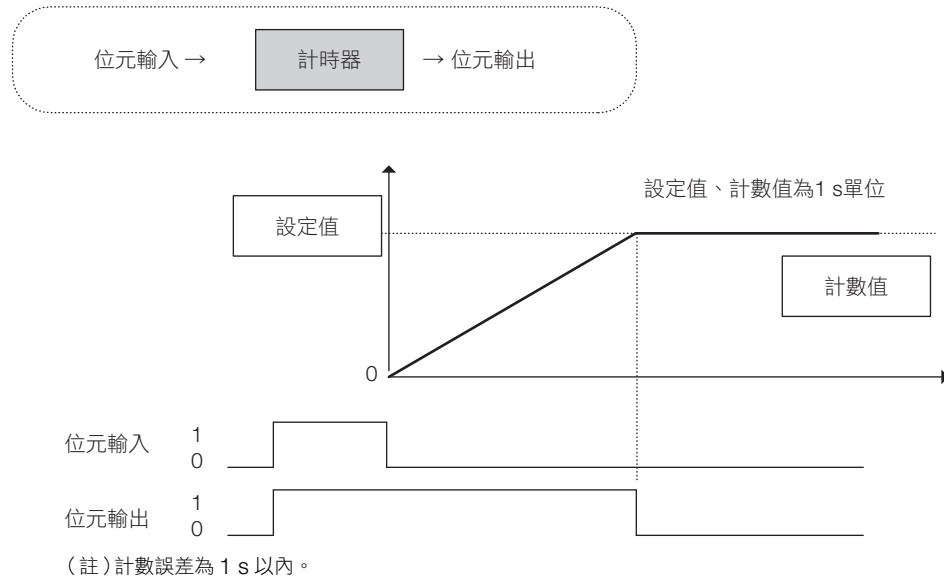




## 斷電延遲計時器 (TOFF (1 s))

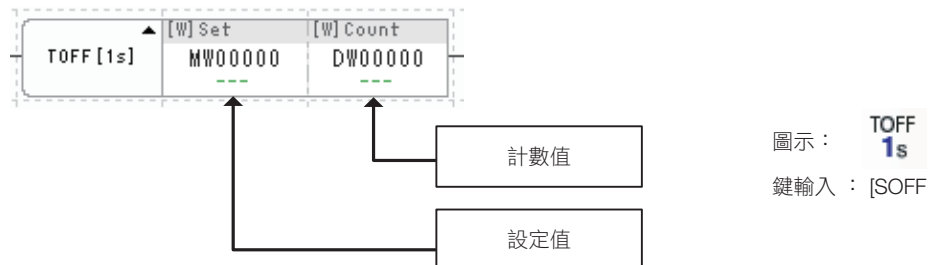
計時器的位元輸入為 0 時，進行時間計數。計數值與設定值相等時，位元輸出為 1。

若位元輸入在計數過程變為 1，則計時器動作停止。若是位元輸入還是 0，則計數從頭 (0) 開始。另外，實際的計數時間 (1 s 單位) 的數值會儲存於計數用暫存器。



### 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
設定值 (Set)	×	○	×	×	×	×	×	×	○
計數值 (Conut)	×	○*	×	×	×	×	×	×	×

\* C、# 暫存器除外

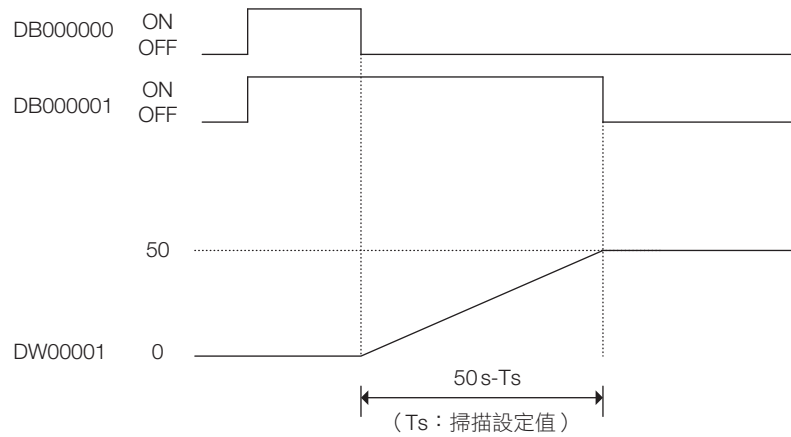
### 程式範例

斷電延遲計時器的設定值為 50、計數值儲存於 DW00001 時的程式範例如下所示。

繼電器 (DB000000) 轉為 OFF 的 50 秒後，線圈 (DB000001) 轉為 OFF。

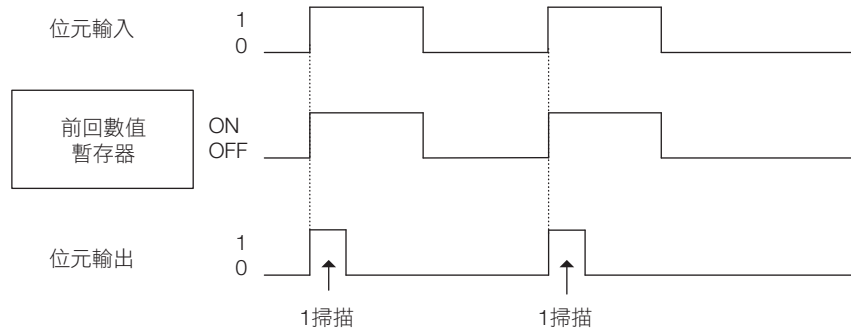


計時器圖表如下所示。



## 上升脈衝（ON-PLS）

位元輸入由 0 變為 1 時，只有 1 掃描之間的位元輸出轉為 1。前一次的位元輸入值，儲存於保有 ON-PLS 指令的前次值暫存器中。



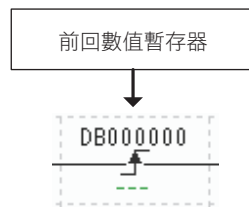
ON-PLS 指令的位元輸入與前次值暫存器位元輸出的關係，如下列真值表所示。


位元輸入	前回數值暫存器	ON-PLS 指令	位元輸出
0	OFF	→	0
0	ON	→	0
1	OFF	→	1
1	ON	→	0

注意第 3 行，位元輸入為 1 而前次值暫存器為 0，也就是說位元輸入由 0 變為 1 時，ON-PLS 指令的位元輸出變為 1。

### 格式

格式如下所示。



圖示：

鍵輸入：JP


輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
前回數值暫存器	○*	×	×	×	×	×	×	×	×

\* C、# 暫存器除外

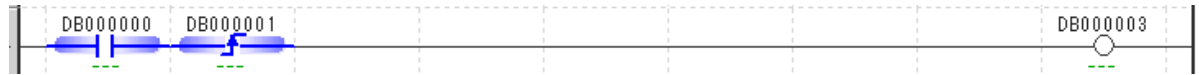
(註) 前回值暫存器為位元輸入前回值儲存用的暫存器。請勿以其他指令設定數值。

### 程式範例

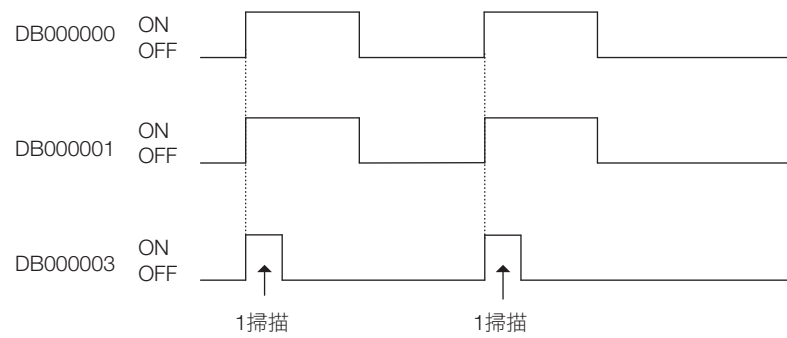
繼電器 (DB000000) 由 OFF 變為 ON 時，線圈 (DB000003) 只有 1 次掃描轉為 ON。DB000001 作為 DB000000 前回值儲存使用。



前回值儲存用暫存器請勿重複使用圖面。  
註記

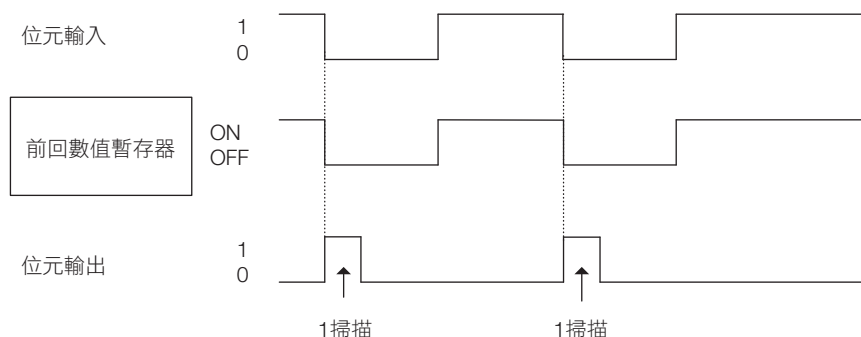


計時器圖表如下所示。



## 下降脈衝（OFF-PLS）

位元輸入由 1 變為 0 時，只有 1 掃描之間的位元輸出轉為 1。前一次的位元輸入值，儲存於保有 OFF-PLS 指令的前次值暫存器中。



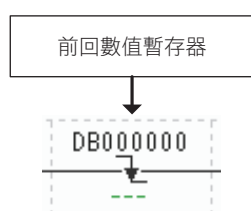
OFF-PLS 指令的位元輸入與前次值暫存器位元輸出的關係，如下列真值表所示。


位元輸入	前回數值暫存器	OFF-PLS 指令	位元輸出
0	OFF	→	0
0	ON	→	1
1	OFF	→	0
1	ON	→	0

注意第 2 行，位元輸入為 0 而前次值暫存器為 1，也就是說位元輸入由 1 變為 0 時，OFF-PLS 指令的位元輸出變為 1。

### 格式

格式如下所示。



圖示：  
鍵輸入：JN

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
前回數值暫存器	○*	×	×	×	×	×	×	×	×

\* C、# 暫存器除外

(註) 前回值暫存器為位元輸入前回值儲存用的暫存器。請勿以其他指令設定數值。

## 程式範例

繼電器 (DB000000) 由 ON 變為 OFF 時，線圈 (DB000003) 只有 1 次掃描轉為 ON。DB000001 作為 DB000000 前回值儲存使用。

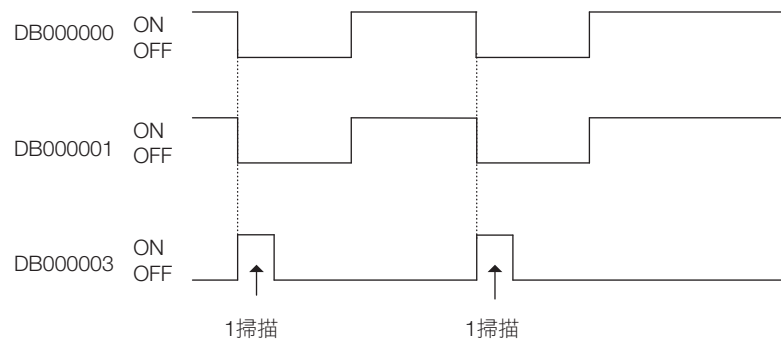


前回值儲存用暫存器請勿重複使用圖面。

註記

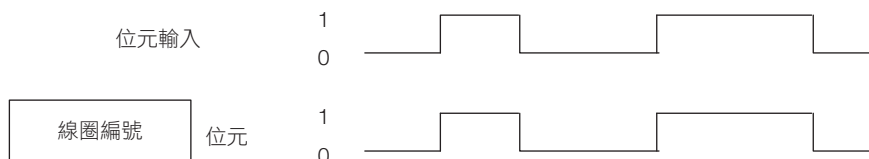


計時器圖表如下所示。



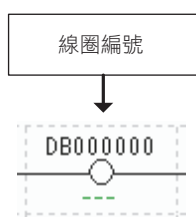
## 線圈 (COIL)

位元輸入為 1 時，線圈編號的位元為 1。位元輸入為 0 時，線圈編號的位元為 0。



### 格式

格式如下所示。



圖示：

鍵輸入：@

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
線圈編號	○*	×	×	×	×	×	×	×	×

\* C、# 暫存器除外

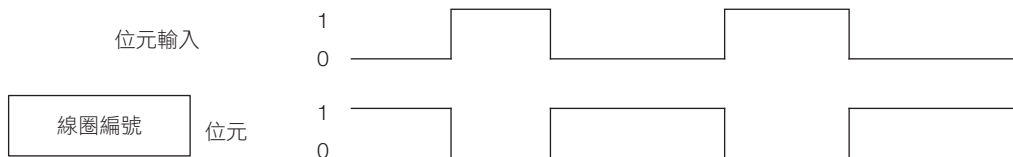
### 程式範例

繼電器 (DB000001) 設為 ON，線圈 (DB000000) 變為 ON。



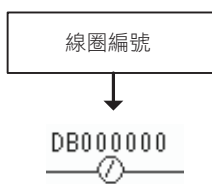
## 反轉線圈 (REV-COIL)

位元輸入為 0 時，線圈編號的位元為 1。位元輸入為 1 時，線圈編號的位元為 0。



### 格式

格式如下所示。



圖示：  
 鍵輸入：@RV

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
線圈編號	○*	×	×	×	×	×	×	×	×

\* #、C 暫存器除外

### 程式範例

繼電器 (DB000001) 設為 ON，線圈 (DB000000) 變為 OFF。





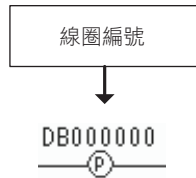
## 上緣觸發變化偵測線圈 (ONP-COIL)


位元輸入由 0 變為 1 時，只有 1 掃描的時間，線圈編號的位元變為 1。

和 ON-PLS 指令與 COIL 指令組合時的動作相同。

### 格式

格式如下所示。



圖示：

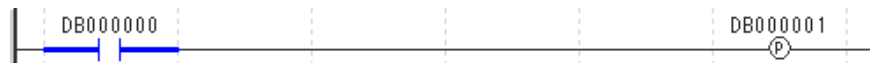
鍵輸入：@P

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
線圈編號	○*	×	×	×	×	×	×	×	×

\* #、C 暫存器除外

### 程式範例

NOC 指令的繼電器 (DB000000) 由 OFF 變為 ON 時，上緣觸發變化偵測線圈 (DB000001) 轉為 ON。



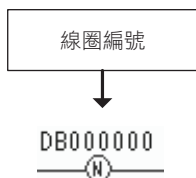
## 下降變化偵測線圈 (OFFP-COIL)


位元輸入由 1 變為 0 時，只有 1 掃描的時間，線圈編號的位元變為 1。

和 OFF-PLS 指令與 COIL 指令組合時的動作相同。

### 格式

格式如下所示。



圖示：

鍵輸入：@N

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
線圈編號	○*	×	×	×	×	×	×	×	×

\* #、C 暫存器除外

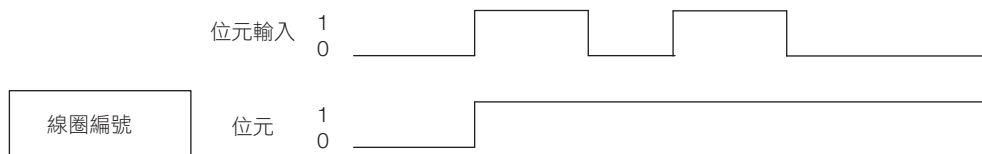
### 程式範例

NOC 指令的繼電器 (DB000000) 由 ON 變為 OFF 時，下降變化偵測線圈 (DB000001) 轉為 ON。



## 設定線圈 (S-COIL)

位元輸入為 1 時，線圈編號的位元數變為 1，設定線圈維持 ON 狀態。



### 格式

格式如下所示。

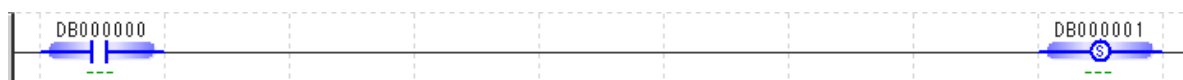


輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
線圈編號	○*	×	×	×	×	×	×	×	×

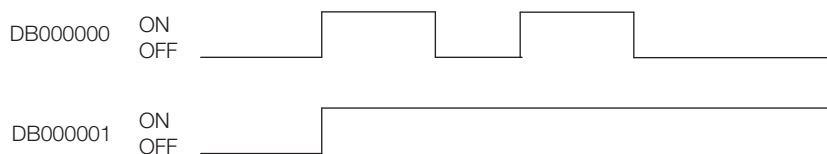
\* C、# 暫存器除外

### 程式範例

繼電器 (DB000000) 設為 ON，設定線圈 (DB000001) 維持 ON 狀態。

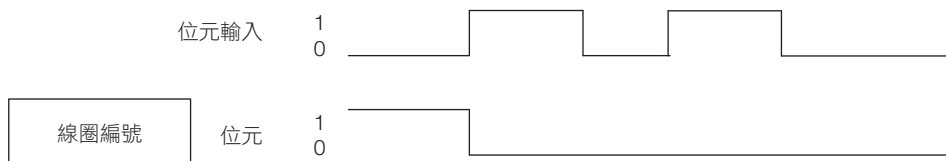


計時器圖表如下所示。



## 重置線圈 (R-COIL)

位元輸入為 1 時，重置線圈編號的位元數變為 0，重置線圈維持 OFF 狀態。



### 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
線圈編號	O*	×	×	×	×	×	×	×	×

\* C、# 暫存器除外

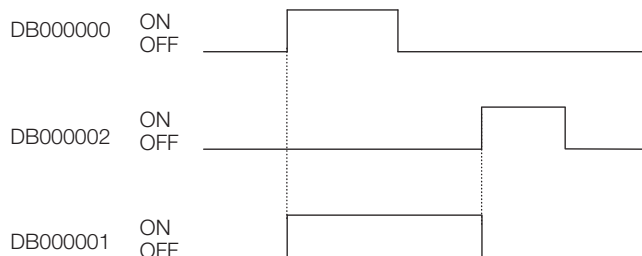
### 程式範例

使用重置線圈，將已為 ON 的第 1 行設定線圈設為 OFF 的程式範例，如下所示。

第 1 行的設定線圈 (DB000001) 為 ON 時，繼電器 (DB000002) 設為 ON，第 2 行的重置線圈 (DB000001) 轉為 ON，設定線圈 (DB000001) 為 OFF。



計時器圖表如下所示。

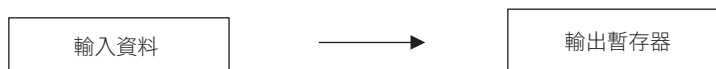


## 4.3

## 數值運算指令

### 儲存 (STORE)

將輸入資料儲存至輸出暫存器。



### 格式

格式如下所示。



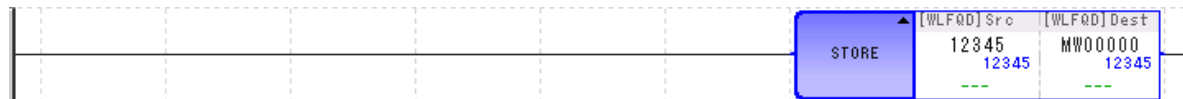
輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 (Src)	×	○	○	○	○	○	○	○	○
輸出暫存器 (Dest)	×	○*	○*	○*	○*	○*	○*	○	×

\* C、# 暫存器除外

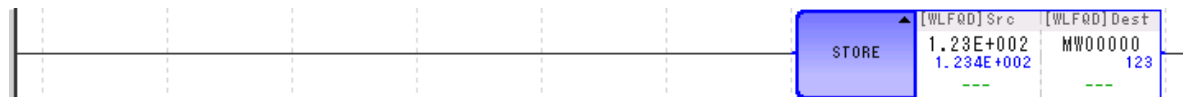
## 程式範例

將輸入資料儲存至輸出暫存器的程式範例，如下所示。

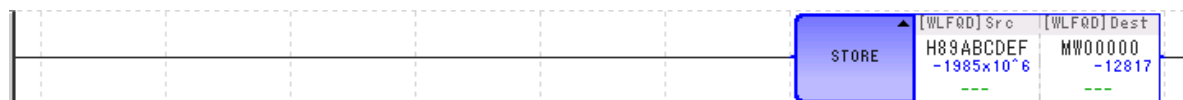
- 輸入資料 12345 (整數) 儲存至輸出暫存器 (MW00000) 時



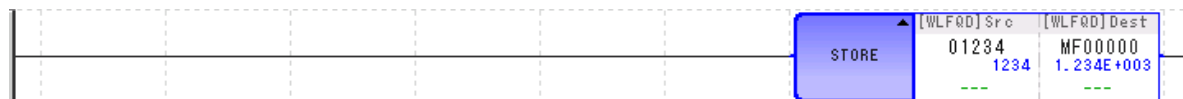
- 輸入資料 12345 (實數) 儲存至輸出暫存器 (MW00000) 時



- 輸入資料 H89ABCDEF (2 倍長整數) 儲存至輸出暫存器 (MW00000) 時  
MW00000 中已儲存了 2 倍長整數的下位字元 -12817 (CDEFH)。




- 輸入資料 1234 (整數) 儲存至輸出暫存器 (MF00000) 時



### 補充

以不同資料類型運算時，運算結果依輸出暫存器的資料類型而異。

 3 章 暫存器 - 不同資料類型的演算注意事項 (第 3-8 頁)

## 加法 (ADD (+))

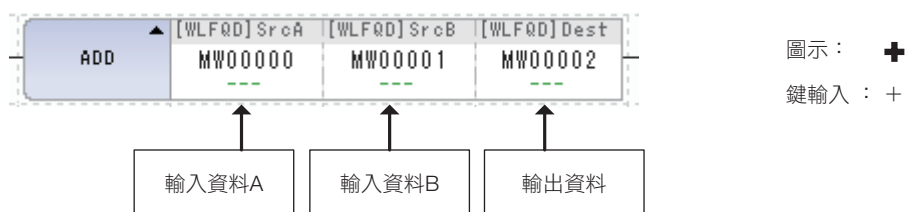
輸入資料 A 及輸入資料 B 相加後儲存於輸出資料。

溢位及欠位會造成運算錯誤。



### 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 A(SrcA)	×	○	○	○	○	○	×	○	○
輸入資料 B(SrcB)	×	○	○	○	○	○	×	○	○
輸出資料 (Dest)	×	○*	○*	○*	○*	○*	×	○	×

\* C、# 暫存器除外

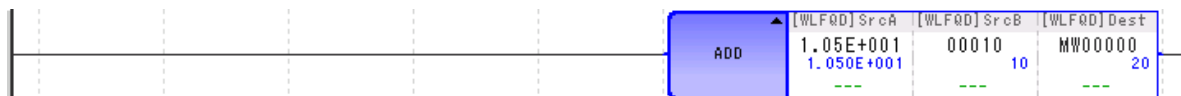
## 程式範例

將輸入資料 A 與輸入資料 B 相加並儲存於輸出資料的程式範例，如下所示。

- 將輸入資料 A 的 100、輸入資料 B 的 200，以及輸出資料儲存於 MW00000 時  
 $100 + 200 \rightarrow MW00000 = 300$



- 將輸入資料 A 的 10.5、輸入資料 B 的 10，以及輸出資料儲存於 MW00000 時  
 $10.5 + 10 \rightarrow MW00000 = 20$  (設為小數點以下四捨五入時)



- 將輸入資料 A (MW00002) 的 20000、輸入資料 B (MW00003) 的 30000，以及輸出資料儲存於 ML00000 時  
 $MW00002 (20000) + MW00003 (30000) \rightarrow ML00000 = 32767^*$



\* 輸入資料 A、B 均為整數型時則以整數型範圍運算，因此前述範例將溢位。


## 補充事項

整數型在運算結果超過 32767 時會造成溢位、- 低於 32768 時會造成欠位的運算錯誤。

2 倍長整數型在運算結果超過 2147483647 時會造成溢位、- 低於 2147483648 時會造成欠位的運算錯誤。

### 補充

以不同資料類型運算時，運算結果依輸出暫存器的資料類型而異。

 3 章 暫存器 - ◆ 不同資料類型的演算注意事項 (第 3-8 頁)

正常情況下，2 倍長整數型的加減算指令 (+、-、++、--) 以 32 位元運算。

但是，僅餘補償運算式 [之前指令為 MUL(x) 指令，之後指令為 DIV(+) 指令] 使用時為 64 位元運算。



## 擴展加法 (ADDX(++))

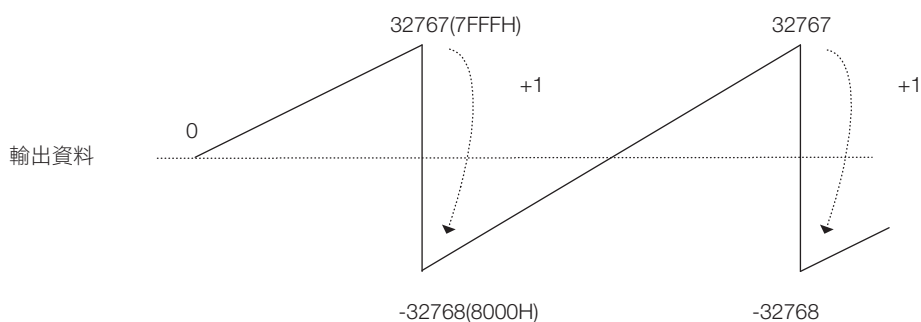
輸入資料 A 及輸入資料 B 相加後儲存於輸出資料。

溢位時不形成運算錯誤，回到負方向最大值後繼續運算。

欠位時不形成運算錯誤，回到正方向最大值後繼續運算。



輸出資料的變遷如下圖所示。



- (註) 1. 上圖為輸出資料為整數型的運算。為 2 倍長整數型時，2147483647 (7FFFFFFFH) 加上 1 後，成為 -2147483648 (80000000)。  
2. 與 ADD(+) 指令或 SUB(-) 指令、EXPRESSION 指令的運算不同，不會發生溢位及欠位。

## 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 A(SrcA)	×	○	○	○	×	×	×	○	○
輸入資料 B(SrcB)	×	○	○	○	×	×	×	○	○
輸出資料 (Dest)	×	○*	○*	○*	×	×	×	○	×

\* C、# 暫存器除外

## 程式範例

將輸入資料 A 與輸入資料 B 擴展相加並儲存於輸出資料的程式範例，如下所示。

- 輸入資料 A 為 32760、輸入資料 B 為 10 及輸出資料儲存於 MW00000 時  
 $32760 \text{ ++ } 10 \rightarrow \text{MW00000} = -32766$

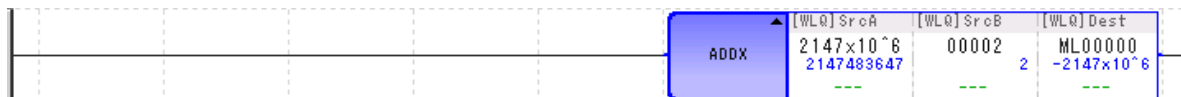


- 將輸入資料 A (MW00002) 的 20000、輸入資料 B (MW00003) 的 30000，以及輸出資料儲存於 ML00000 時  
 $\text{MW00002}(20000) \text{ ++ } \text{MW00003}(30000) \rightarrow \text{ML00000} = -15536^*$



\* 輸入資料 A、B 均為整數型時則以整數型範圍運算，因此前述範例中，ML00000 不會等於 50000。

- 輸入資料 A 為 2147483647、輸入資料 B 為 2 及輸出資料儲存於 ML00000 時  
 $2147483647 \text{ ++ } 2 \rightarrow \text{ML00000} = -241783647$




- 將輸入資料 A 的 -32768、輸入資料 B 的 -1，以及輸出資料儲存於 MW00000 時  
 $-32768 \text{ ++ } -1 \rightarrow \text{MW00000} = 32767$



### 補充

以不同資料類型運算時，運算結果依輸出暫存器的資料類型而異。

 3 章 暫存器 - ◆ 不同資料類型的演算注意事項 (第 3-8 頁)

正常情況下，2 倍長整數型的加減算指令 (+、-、++、--) 以 32 位元運算。

但是，僅餘補償運算式 [之前指令為 MUL(x) 指令，之後指令為 DIV(+) 指令] 使用時為 64 位元運算。

## 減法 (SUB (-))

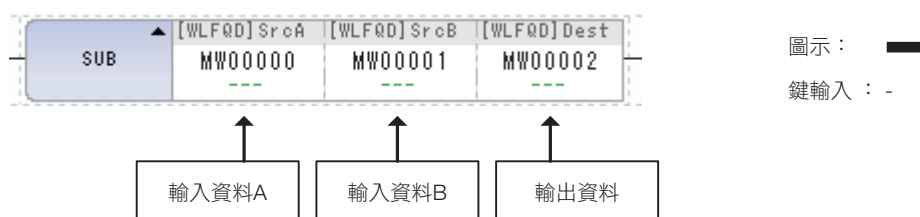
減去輸入資料 A 與輸入資料 B，儲存於輸出資料。

溢位及欠位會造成運算錯誤。



### 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 A(SrcA)	×	○	○	○	○	○	×	○	○
輸入資料 B(SrcB)	×	○	○	○	○	○	×	○	○
輸出資料 (Dest)	×	○*	○*	○*	○*	○*	×	○	×

\* C、# 暫存器除外

## 程式範例

減去輸入資料 A 與輸入資料 B，儲存至輸出資料的程式範例，如下所示。

- 將輸入資料 A 的 100、輸入資料 B 的 200，以及輸出資料儲存於 MW00000 時  
 $100 - 200 \rightarrow MW00000 = -100$



- 將輸入資料 A 的 10.5、輸入資料 B 的 10，以及輸出資料儲存於 MW00000 時  
 $10.5 - 10 \rightarrow MW00000 = 0$  (設為小數點以下捨去時)



- 將輸入資料 A (MW00002) 的 -20000、輸入資料 B (MW00003) 的 30000，以及輸出資料儲存於 ML00000 時  
 $MW00002(-20000) - MW00003(30000) \rightarrow ML00000 = -32768^*$



\* 輸入資料 A、B 均為整數型時則以整數型範圍運算，因此前述範例將欠位。


## 補充事項

整數型在運算結果超過 32767 時會造成溢位、- 低於 32768 時會造成欠位的運算錯誤。

2 倍長整數型在運算結果超過 2147483647 時會造成溢位、- 低於 2147483648 時會造成欠位的運算錯誤。

### 補充

以不同資料類型運算時，運算結果依輸出暫存器的資料類型而異。

 3 章 暫存器 - ◆ 不同資料類型的演算注意事項 (第 3-8 頁)

正常情況下，2 倍長整數型的加減算指令 (+、-、++、--) 以 32 位元運算。

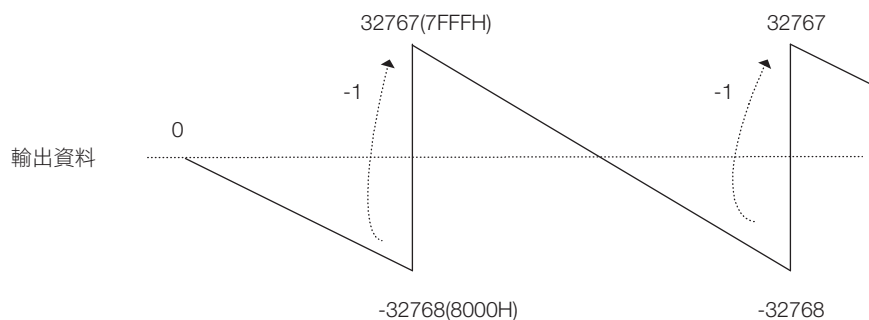
但是，僅餘補償運算式 [之前指令為 MUL(x) 指令，之後指令為 DIV(+) 指令] 使用時為 64 位元運算。

## 擴充減法 (SUBX(--))

輸入資料 A 及輸入資料 B 相加後儲存於輸出資料。  
 溢位時不形成運算錯誤，回到負方向最大值後繼續運算。  
 欠位時不形成運算錯誤，回到正方向最大值後繼續運算。



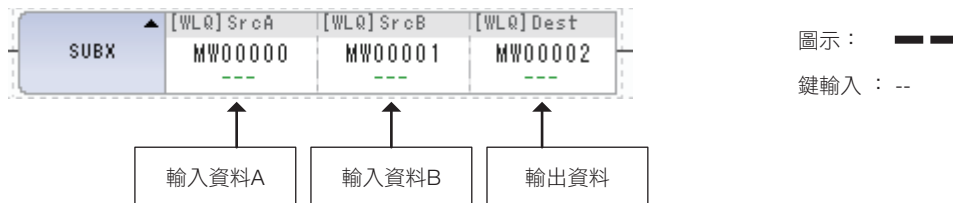
輸出資料的變遷如下所示。



- (註)1. 上圖為輸出資料為整數型的運算。為 2 倍長整數型時，-2147483648 (80000000H) 減去 1 後，成為 2147483647 (7FFFFFFFH)。
- 2. 與 ADD(+) 指令或 SUB(-) 指令、EXPRESSION 指令的運算不同，不會發生溢位及欠位。

### 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 A(SrcA)	×	○	○	○	×	×	×	○	○
輸入資料 B(SrcB)	×	○	○	○	×	×	×	○	○
輸出資料 (Dest)	×	○*	○*	○*	×	×	×	○	×

\* C、# 暫存器除外

## 程式範例

擴展減去輸入資料 A 與輸入資料 B，儲存至輸出資料的程式範例，如下所示。

- 輸入資料 A 為 -32760、輸入資料 B 為 10 及輸出資料儲存於 MW00000 時  
 $-32760 -- 10 \rightarrow MW00000 = 32766$

	[WLQ] SrcA	[WLQ] SrcB	[WLQ] Dest
SUBX	-32760 -32760 ---	00010 10 ---	MW00000 32766 ---

- 將輸入資料 A (MW00002) 的 -20000、輸入資料 B (MW00003) 的 30000，以及輸出資料儲存於 ML00000 時  
 $MW00002(-20000) -- MW00003(30000) \rightarrow ML00000 = 15536^*$

	[WLQ] SrcA	[WLQ] SrcB	[WLQ] Dest
SUBX	-20000 -20000 ---	30000 30000 ---	ML00000 15536 ---

\* 輸入資料 A、B 均為整數型時則以整數型範圍運算，因此前述範例中，ML00000 不會等於 -50000。

- 將輸入資料 A 的 -2147483648、輸入資料 B 的 2，以及輸出資料儲存至 ML00000 時  
 $-2147483648 -- 2 \rightarrow ML00000 = 214783646$


	[WLQ] SrcA	[WLQ] SrcB	[WLQ] Dest
SUBX	$-214 \times 10^7$ $-2147 \times 10^6$ ---	00002 2 ---	ML00000 214783646 ---

- 將輸入資料 A 的 32767、輸入資料 B 的 -1，以及輸出資料儲存於 MW00000 時  
 $32767 -- -1 \rightarrow MW00000 = -32768$

	[WLQ] SrcA	[WLQ] SrcB	[WLQ] Dest
SUBX	32767 32767 ---	-00001 -1 ---	MW00000 -32768 ---

### 補充

以不同資料類型運算時，運算結果依輸出暫存器的資料類型而異。

 3 章 暫存器 - ◆ 不同資料類型的演算注意事項 (第 3-8 頁)

正常情況下，2 倍長整數型的加減算指令 (+、-、++、--) 以 32 位元運算。

但是，僅餘補償運算式 [之前指令為 MUL(x) 指令，之後指令為 DIV(+) 指令] 使用時為 64 位元運算。

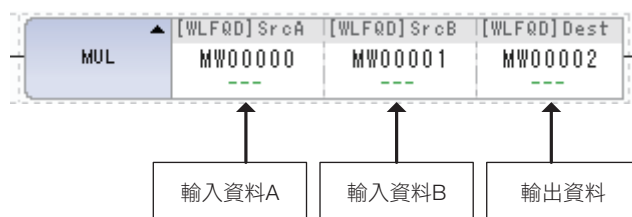
## 乘法 (MUL (x))

輸入資料 A 乘以輸入資料 B，儲存於輸出資料。



### 格式

格式如下所示。



圖示：**×**  
鍵輸入：\*

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 A(SrcA)	×	○	○	○	○	○	×	○	○
輸入資料 B(SrcB)	×	○	○	○	○	○	×	○	○
輸出資料 (Dest)	×	○*	○*	○*	○*	○*	×	○	×

\* C、# 暫存器除外

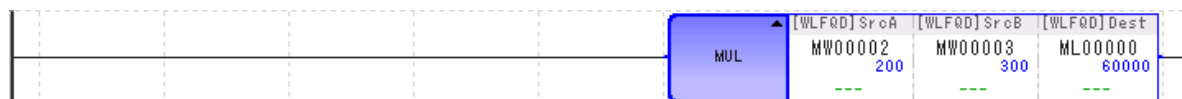
## 程式範例

輸入資料 A 乘以輸入資料 B，儲存至輸出資料的程式範例，如下所示。

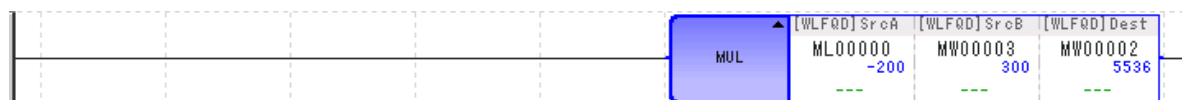
- 輸入資料 A 為 100、輸入資料 B 為 200 及輸出資料儲存於 MW00000 時  
 $100 \times 200 \rightarrow MW00000 = 20000$



- 將輸入資料 A (MW00002) 的 200、輸入資料 B (MW00003) 的 300，以及輸出資料儲存於 ML00000 時  
 $MW00002(200) \times MW00003(300) \rightarrow ML00000 = 60000$




- 將輸入資料 A (ML00000) 的 -200、輸入資料 B (MW00003) 的 300，以及輸出資料儲存於 MW00002 時  
 $-200 \times 300 \rightarrow MW00002 = 5536^*$



\* 輸入資料中有 2 倍長整數型，所以以 2 倍長整數型的範圍運算，但因輸出資料為整數型，因此運算結果如超出整數型範圍時，原本運算結果中的下位 16 位元將被儲存至輸出資料。

### 補充

以不同資料類型運算時，運算結果依輸出暫存器的資料類型而異。

 3 章 暫存器 - ◆ 不同資料類型的演算注意事項 (第 3-8 頁)

正常情況下，2 倍長整數型的加減算指令 (+、-、++、--) 以 32 位元運算。

但是，僅餘補償運算式 [之前指令為 MUL(x) 指令，之後指令為 DIV(+) 指令] 使用時為 64 位元運算。



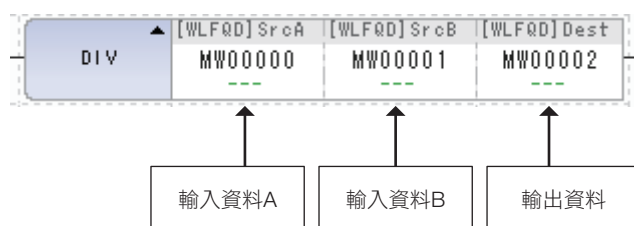
## 除法 (DIV (÷))

輸入資料 A 與輸入資料 B 相除，儲存於輸出資料。



### 格式

格式如下所示。



圖示： ÷

鍵輸入： /

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 A(SrcA)	×	○	○	○	○	○	×	○	○
輸入資料 B(SrcB)	×	○	○	○	○	○	×	○	○
輸出資料 (Dest)	×	○*	○*	○*	○*	○*	×	○	×

\* C、# 暫存器除外

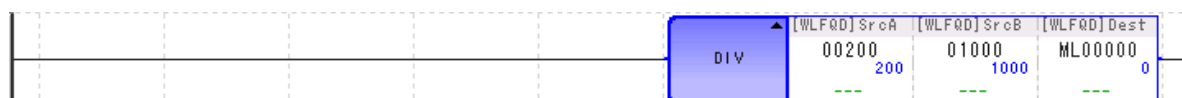
## 程式範例

將輸入資料 A 與輸入資料 B 相除，儲存至輸出資料的程式範例，如下所示。

- 輸入資料 A 為 200、輸入資料 B 為 100 及輸出資料儲存於 MW00000 時  
 $200 \div 100 \rightarrow MW00000 = 2$



- 輸入資料 A 為 200、輸入資料 B 為 1000 及輸出資料儲存於 ML00000 時  
 $200 \div 1000 \rightarrow ML00000 = 0$




- 輸入資料 A 為 200、輸入資料 B 為 1000 及輸出資料儲存於 MF00000 時  
 $200 \div 1000 \rightarrow MF00000 = 0.2$



### 補充

以不同資料類型運算時，運算結果依輸出暫存器的資料類型而異。

 3 章 暫存器 - ◆ 不同資料類型的演算注意事項 (第 3-8 頁)

正常情況下，2 倍長整數型的加減算指令 (+、-、++、--) 以 32 位元運算。

但是，僅餘補償運算式 [之前指令為 MUL(x) 指令，之後指令為 DIV(+) 指令] 使用時為 64 位元運算。

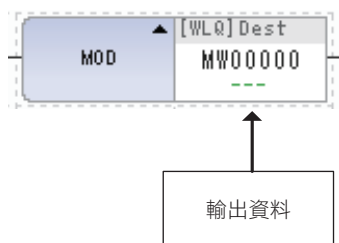
## 整數餘數 (MOD)

將前一次執行的整數型或 2 倍長整數型除法運算後餘數儲存至輸出資料。請務必於 DIV (÷) 指令後立即執行 MOD 指令。DIV (÷) 指令後未立即執行 MOD 指令，則無法保證直到至下一次數值運算指令出現為止的運算結果。



### 格式

格式如下所示。



圖示： MOD  
 鍵輸入：MOD

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸出資料 (Dest)	×	○*	○*	○*	×	×	×	○	×

\* C、# 暫存器除外

## 程式範例

將輸入資料 A 與輸入資料 B 相除，儲存餘數至輸出資料的程式範例，如下所示。

- 之前除法為以下時  
 $12345 \div 123 \rightarrow MW00000 = 100$   
 立即執行 MOD 指令  $\rightarrow MW00001 = 45$



- 之前除法為以下時  
 $123456789 \div 12345 \rightarrow ML00000 = 10000$   
 立即執行 MOD 指令  $\rightarrow ML00002 = 6789$



### 補充

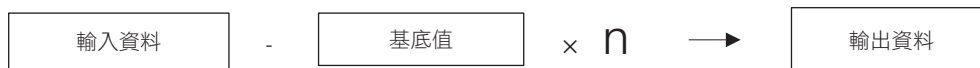
以不同資料類型運算時，運算結果依輸出暫存器的資料類型而異。

3 章 暫存器 - ◆ 不同資料類型的演算注意事項 (第 3-8 頁)

## 實數餘數 (REM)

將實數除法運算後的餘數儲存至輸出資料。此時，餘數是指以來自輸入資料的基準值進行重覆減法運算所獲得的餘數。

也就是說，來自輸入資料的基準值經重覆  $n$  次減法運算時的值 (輸入資料 - 基準值  $\times n$ ) 比基準值為小時，則輸出該值。

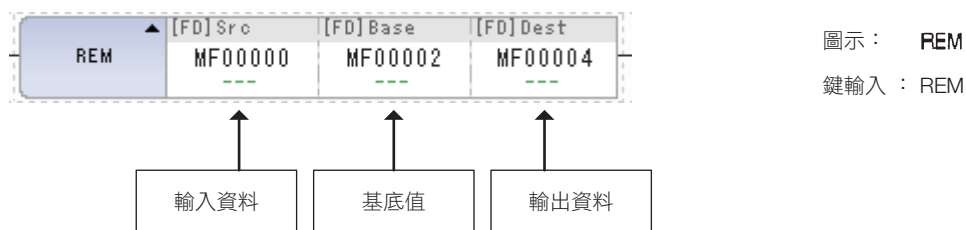


$n$  的值以 0, 1, 2, 3... 遞增，且首次滿足下列公式時，以該  $n$  值進行輸出資料運算。

(輸入資料 - 基準值  $\times n$ ) < 基準值

### 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 (Src)	×	×	×	×	○	○	×	×	○
基準 (Base)	×	×	×	×	○	○	×	×	○
輸出資料 (Dest)	×	×	×	×	○*	○*	×	○	×

\* C、# 暫存器除外

## 程式範例

將來自輸入資料的基準值減法運算 n 次，儲存獲得餘數至輸出資料的程式範例，如下所示。

- 輸入資料為 5.0、基底為 2.0、輸出資料儲存於 MF00000 時  
 $5.0 - 2.0 - 2.0 = 1.0 < \text{基準}(2.0) \rightarrow \text{MF00000} = 1.0$

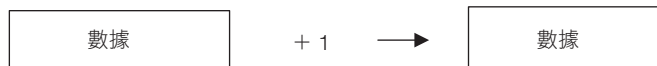
	[FD] Src	[FD] Base	[FD] Dest
REM	5.00E+000	2.00E+000	MF00000
	5.000E+000	2.000E+000	1.000E+000
	---	---	---

- 輸入資料為 3000.0、基底為 3.0、輸出資料儲存於 MF00000 時  
 $3000.0 - 3.0 - 3.0 \dots = 0.0 < \text{基準}(3.0) \rightarrow \text{MF00000} = 0.0$

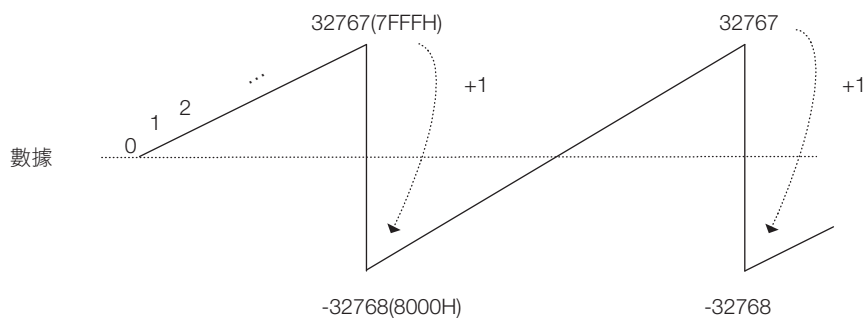
	[FD] Src	[FD] Base	[FD] Dest
REM	3.00E+003	3.00E+000	MF00000
	3.000E+003	3.000E+000	0.000E+000
	---	---	---

## 增量 (INC)

整數型以及 2 倍長整數型的資料加 1。整數型及 2 倍長整數型皆不會發生溢位及欠位。如同 ADDX(++) 指令進行運算。



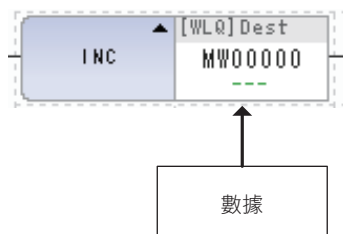
執行 INC 指令時的資料變遷如下圖所示。



(註) 上圖為資料為整數型的運算。為 2 倍長整數型時，2147483647 (7FFFFFFFH) 加上 1 後，成為 -2147483648 (80000000H)。

## 格式

格式如下所示。



圖示： **+1**

鍵輸入：INC

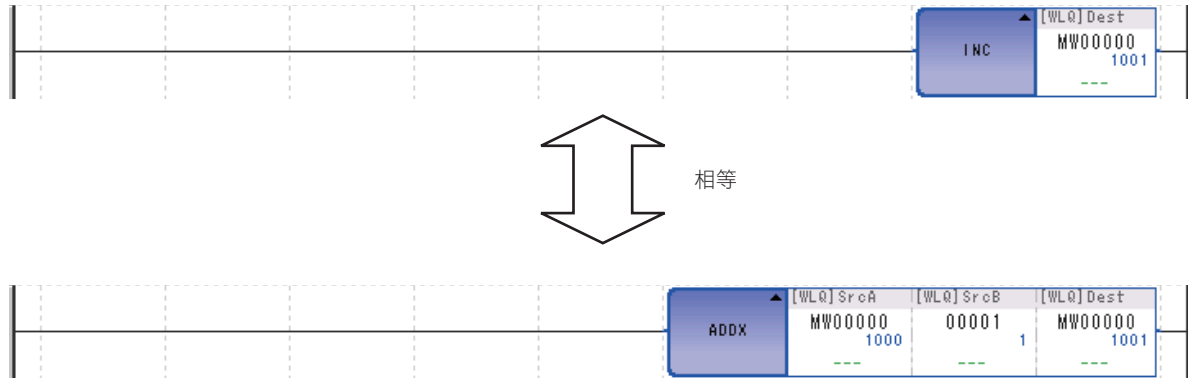
輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
資料 (Dest)	×	○	○	○*	×	×	×	○	×

\* C、# 暫存器除外

## 程式範例

使用 INC 指令，以及使用 ADDX(++) 指令時的程式範例，如下所示。

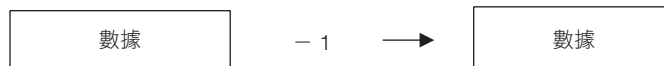
與使用 ADDX(++) 指令讓資料 (MW00000) 1000 加 1 時等價。



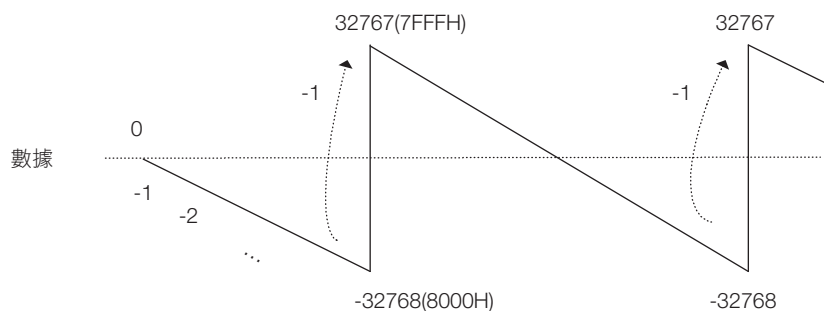


## 減量 (DEC)

整數型以及 2 倍長整數型的資料減 1。整數型及 2 倍長整數型皆不會發生溢位及欠位。如同 SUBX(-- ) 指令進行運算。



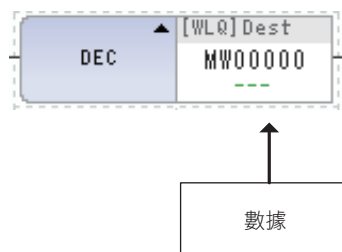
執行 DEC 指令時的資料變遷如下圖所示。



(註) 上圖為資料為整數型的運算。為 2 倍長整數型時，-2147483648 (80000000H) 減去 1 後，成為 2147483647 (7FFFFFFFH)。

## 格式

格式如下所示。



圖示： -1

鍵輸入：DEC

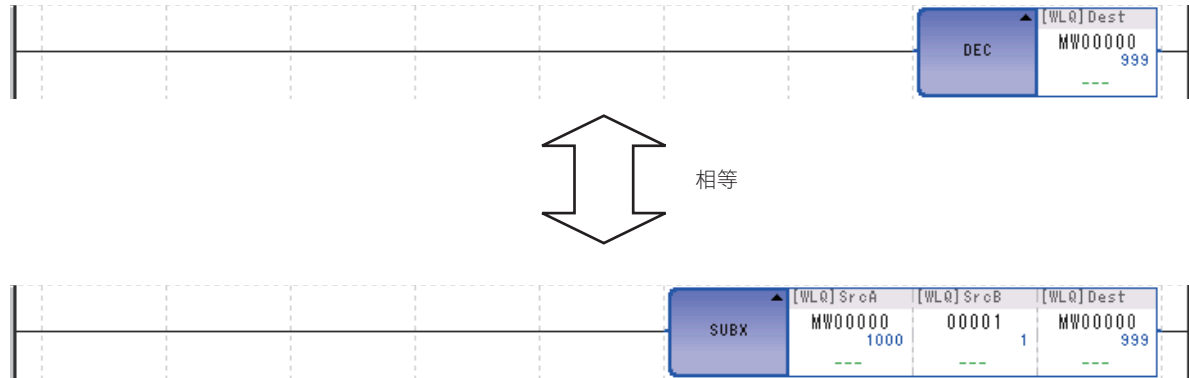
輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
資料 (Dest)	×	○	○	○ <sup>+</sup>	×	×	×	○	×

\* C、# 暫存器除外

### 程式範例

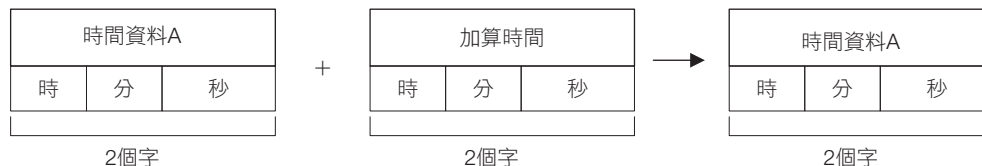
使用 DEC 指令，以及使用 SUBX(-- ) 指令時的程式範例，如下所示。

與用 SUBX(-- ) 指令使資料 (MW00000) 1000 減 1 時等價。



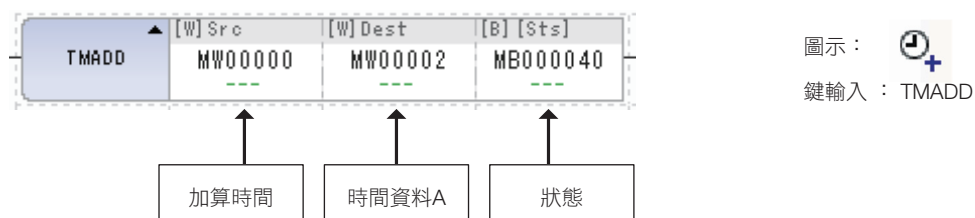
## 增加時間 (TMADD)

將兩個時間資料 (時/分/秒) 進行加法運算。時間資料 A 加上增加時間後，儲存於時間資料 A。時間資料有 2 字元份。



### 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
增加時間 (Src)	×	○ <sup>*2</sup>	×	×	×	×	×	×	×
時間資料 A(Dest)	×	○ <sup>*2</sup>	×	×	×	×	×	×	×
狀態 (Sts) <sup>*1</sup>	○ <sup>*2</sup>	×	×	×	×	×	×	×	×

\*1. 可以省略

\*2. C、# 暫存器除外

時間資料的數據格式顯示如下。

偏移量	資料內容	資料範圍 (BCD)
0	時/分	上位位元組 (時): 00 ~ 23 下位位元組 (分): 00 ~ 59
1	秒	0000 ~ 0059

運算結果如超出前述資料範圍時，時間資料 A 不會更新，秒資料輸出為 9999，狀態位元變成 1。

運算結果為範圍內時，狀態的位元為 0。

## 程式範例

下表為使用 TMADD 指令製作階梯圖程式時的條件範例。表示指令執行前的時間資料 A 與欲增加的時間。

時間	指令執行前的時間資料 A	加算時間
時/分	MW00000 = 0210H (2 時 10 分)	MW00002 = 0050H (0 時 50 分)
秒	MW00001 = 0050H (50 秒)	MW00003 = 0020H (20 秒)

依上表條件加算時間，儲存於時間資料 A 的程式範例，如下所示。

The screenshot shows a ladder logic program with the following components:

- EXPRESSION** (top):
 

```
//時刻データA(加算前);
MW00000 = 0x210; //2時10分;
528=528
MW00001 = 0x0050; //50秒;
80=80

//加算時刻;
MW00002 = 0x0050; //0時50分;
80=80
MW00003 = 0x0020; //20秒;
32=32
```
- TMADD** instruction:
 

[W] Src	[W] Dest	[B] [Sts]
MW00002	MW00000	MB000040
- EXPRESSION** (bottom):
 

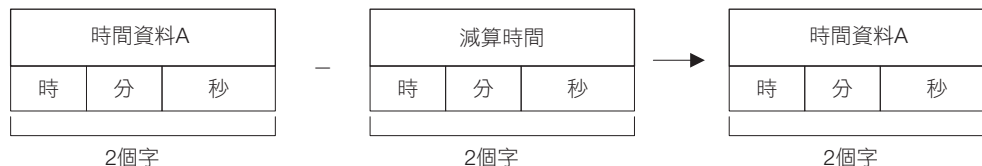
```
//時刻データA(加算後);
MW00000 = MW00000; // 時/分;
769=769
MW00001 = MW00001; // 秒;
16=16
```

指令執行前的時間資料 A 加上增加時間的結果，如下所示。

時間	指令執行後的時間資料 A
時/分	MW00000 = 769 = 0301H (3 時 01 分)
秒	MW00001 = 16 = 0010H (10 秒)

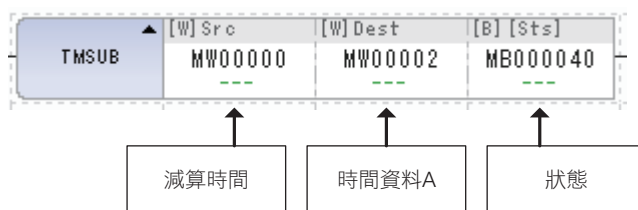
## 減少時間 (TMSUB)


將兩個時間資料 (時/分/秒) 進行減法運算。時間資料 A 減去增加時間後，儲存於時間資料 A。時間資料有 2 字元份。



### 格式

格式如下所示。



圖示：  
鍵輸入：TMSUB

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
減少時間 (Src)	×	○ <sup>*2</sup>	×	×	×	×	×	×	×
時間資料 A(Dest)	×	○ <sup>*2</sup>	×	×	×	×	×	×	×
狀態 (Sts) <sup>*1</sup>	○ <sup>*2</sup>	×	×	×	×	×	×	×	×

\*1. 可以省略

\*2. C、# 暫存器除外

時間資料的數據格式顯示如下。

偏移量	資料內容	資料範圍 (BCD)
0	時/分	上位元組 (時) : 00 ~ 23 下位元組 (分) : 00 ~ 59
1	秒	0000 ~ 0059

運算結果如超出前述資料範圍時，時間資料 A 不會更新，秒資料輸出為 9999，狀態位元變成 1。

運算結果為範圍內時，狀態的位元為 0。

## 程式範例

下表為使用 TMSUB 指令製作階梯圖程式時的條件範例。表示指令執行前的時間資料 A 與欲減少的時間。

時間	指令執行前的時間資料 A	減算時間
時/分	MW00000 = 0210H (2 時 10 分)	MW00002 = 0050H (0 時 50 分)
秒	MW00001 = 0050H (50 秒)	MW00003 = 0020H (20 秒)

依上表條件減去時間，儲存於時間資料 A 的程式範例，如下所示。

The screenshot shows a ladder logic program with the following components:

- EXPRESSION 1:**

```
//時刻データA(減算前);
MW00000 = 0x210; //2時10分;
528=528
MW00001 = 0x0050; //50秒;
80=80

//減算時刻;
MW00002 = 0x0050; //0時50分;
80=80
MW00003 = 0x0020; //20秒;
32=32
```
- TMSUB Instruction:**

[W] Src	[W] Dest	[B] [Sts]
MW00002	MW00000	MB000040
		0
- EXPRESSION 2:**

```
//時刻データA(減算後);
MW00000 = MW00000; // 時/分;
288=288
MW00001 = MW00001; // 秒;
48=48
```

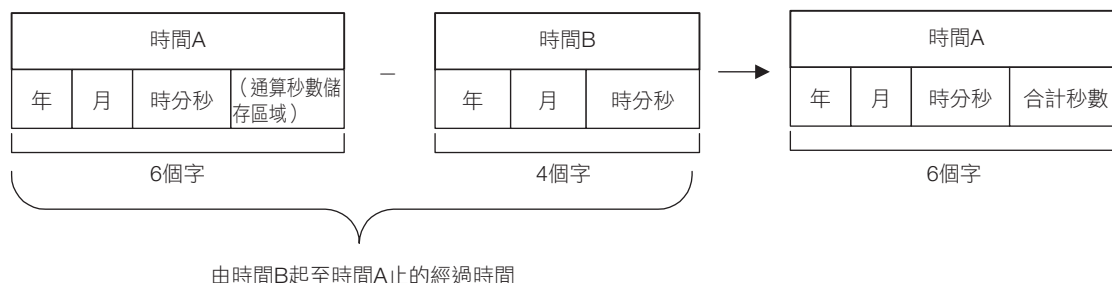
指令執行前的時間資料 A 減去增加時間的結果，如下所示。

時間	指令執行後的時間資料 A
時/分	MW00000 = 288 = 0120H (1 時 20 分)
秒	MW00001 = 48 = 0030H (30 秒)

## 經過時間 (SPEND)

將兩個資料（年／月／日／時／分／秒）相減，計算經過時間。時間 A 減去時間 B（也就是說，從時間 B 起至時間 A 止的經過時間）後，儲存於時間 A。

時間資料最多為 4 個字元。



## 格式

格式如下所示。



圖示：  
鍵輸入：SPEND

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
時間 B (Src)	×	○ <sup>*2</sup>	×	×	×	×	×	×	×
時間 A (Dest)	×	○ <sup>*2</sup>	×	×	×	×	×	×	×
狀態 (Sts) <sup>*1</sup>	○ <sup>*2</sup>	×	×	×	×	×	×	×	×

\*1. 可以省略

\*2. C、# 暫存器除外

以下顯示時間 B 的數據格式。

偏移量	資料內容	資料範圍 (BCD)	輸出輸入
0	年 (BCD)	0000 ~ 0099	IN
1	月/日 (BCD)	上位位元組 (月) : 01 ~ 12 下位位元組 (日) : 01 ~ 31	IN
2	時/分 (BCD)	上位位元組 (時) : 00 ~ 23 下位位元組 (分) : 00 ~ 59	IN
3	秒 (BCD)	0000 ~ 0059	IN

以下顯示時間 A 的數據格式。

偏移量	資料內容	資料範圍 (BCD)	輸出輸入
0	年 (BCD)	0000 ~ 0099	IN/OUT
1	月/日 (BCD)	上位位元組 (月) : 01 ~ 12 下位位元組 (日) : 01 ~ 31	IN/OUT
2	時/分 (BCD)	上位位元組 (時) : 00 ~ 23 下位位元組 (分) : 00 ~ 59	IN/OUT
3	秒 (BCD)	0000 ~ 0059	IN/OUT
4	合計秒數	將運算結果中年/月/日/時/分換算 為秒數的結果 (2 倍長整數)	IN/OUT
5			

運算結果如超出前述資料範圍時，時間 A 不會更新，秒資料輸出為 9999，狀態位元變成 1。

運算結果為範圍內時，狀態的位元為 0。

**補充**

年以 365 日計。不考慮潤年影響。  
此外，月數不計算。以日數計算。



## 程式範例

下表為使用 SPEND 指令製作階梯圖程式時的條件範例。

顯示指令執行前的時間 A (10 年 11 月 20 日 2 時 10 分 50 秒) 與時間 B (9 年 10 月 10 日 0 時 50 分 20 秒)。

	指令執行前的時間 A	時間 B
年	MW00000 = 0010H (10 年)	MW00006 = 0009H (9 年)
月/日	MW00001 = 1120H (11 月 20 日)	MW00007 = 1010H (10 月 10 日)
時/分	MW00002 = 0210H (2 時 10 分)	MW00008 = 0050H (0 時 50 分)
秒	MW00003 = 0050H (50 秒)	MW00009 = 0020H (20 秒)

依上表條件，以時間 A 減去時間 B，儲存握時間於時間 A 的程式範例，如下所示。

The screenshot shows a ladder logic program with two expression blocks and a SPEND instruction. The first expression block calculates the initial time A and B. The SPEND instruction is set to subtract time B from time A. The second expression block shows the result of the subtraction.

```

EXPRESSON
//時刻A(減算前);
MW00000 = 0x0010;//10年;
16=16
MW00001 = 0x1120;//11月20日;
4384=4384
MW00002 = 0x0210;//2時10分;
528=528
MW00003 = 0x0050;//50秒;
80=80

//時間B;
MW00006 = 0x0009;//9年;
9=9
MW00007 = 0x1010;//10月10日;
4112=4112
MW00008 = 0x0050;//0時50分;
80=80
MW00009 = 0x0020;//20秒;
32=32

SPEND [W] Src [W] Dest [B] [Sts]
      MW00006 MW00000 MB000040 0

EXPRESSON
//時間A(減算後)→經過時間;
MW00000 = MW00000;//年;
1=1
MW00001 = MW00001;//月日;
65=65
MW00002 = MW00002;//時分;
288=288
MW00003 = MW00003;//秒;
48=48
ML00004 = ML00004;//通算秒數;
35083230=35083230
    
```

使用 SPEND 指令的結果，如下所示。

	指令執行後的時間 A
年	MW00000 = 1 = 0001H (1 年)
月/日	MW00001 = 65 = 0041H (0 月 41 日)
時/分	MW00002 = 288 = 0120H (1 時 20 分)
秒	MW00003 = 48 = 0030H (30 秒)
合計秒數	ML00004 = 35083230

## 符號反轉 (INV)

反轉輸入資料的符號，儲存至輸出資料。



### 格式

格式如下所示。



圖示： INV

鍵輸入： INV

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 (Src)	×	○	○	○	○	○	×	○	○
輸出資料 (Dest)	×	○*	○*	○*	○*	○*	×	○	×

\* C、# 暫存器除外

### 程式範例

輸入資料 A (MW00000) 為 1234 時，反轉符號，儲存至輸出資料 (ML00002) 的程式範例，如下所示。

$-1 \times MW00000(1234) \rightarrow ML00002 = -1234$



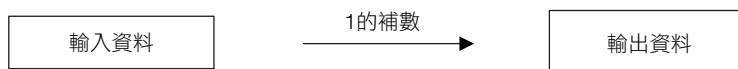
#### 補充

以不同資料類型運算時，運算結果依輸出暫存器的資料類型而異。

3 章 暫存器 - ◆ 不同資料類型的演算注意事項 (第 3-8 頁)

## 1 的補數 (COM)

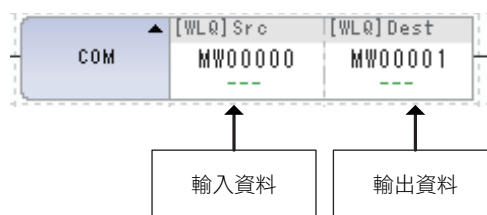
將輸入資料的 1 的補數，儲存至輸出資料。



(註)對於輸入資料的 2 進制表現，反轉 0 與 1，儲存至輸出資料。

### 格式

格式如下所示。



圖示：**COM**  
鍵輸入：COM

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 (Src)	×	○	○	○	×	×	×	○	○
輸出資料 (Dest)	×	○*	○*	○*	×	×	×	○	×

\* C、# 暫存器除外

### 程式範例

將輸入資料 (MW00000) -3856 (HF0F0) 的 1 的補數儲存至輸出資料 (MW00001) 的程式範例，如下所示。

MW00000 = -3856(HF0F0) → MW00001 = 3855(H0F0F)



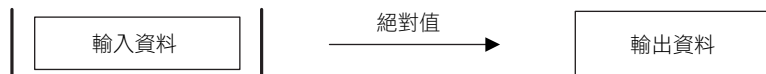
#### 補充

以不同資料類型運算時，運算結果依輸出暫存器的資料類型而異。

☞ 3 章 暫存器 - ◆ 不同資料類型的演算注意事項 (第 3-8 頁)

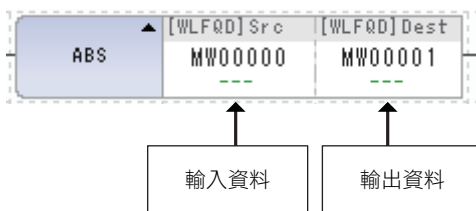
## 絕對值轉換 (ABS)

將輸入資料的絕對值，儲存至輸出資料。



### 格式

格式如下所示。



圖示：

鍵輸入：ABS

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 (Src)	×	○	○	○	○	○	×	○	○
輸出資料 (Dest)	×	○*	○*	○*	○*	○*	×	○	×

\* C、# 暫存器除外

### 程式範例

將輸入資料 (MF00000) -1.23 的絕對值儲存至輸出資料 (MF00002) 的程式範例，如下所示。

| MF00000(-1.23) | → MF00002=1.23



補充

以不同資料類型運算時，運算結果依輸出暫存器的資料類型而異。

3章 暫存器 - ◆ 不同資料類型的演算注意事項 (第 3-8 頁)

## 2 進制轉換 (BIN)

將輸入資料的 BCD 表示數值轉換為二進制數據 (BIN 轉換)，儲存至輸出資料。  
輸入資料不為 BCD 表示 (123FH 等) 時，BIN 轉換無法得到正確結果。



(註) 輸入資料BCD標記為abcd時，輸出資料以下列式子運算。  
輸出資料 = (a × 1000) + (b × 100) + (c × 10) + d

**補充** 輸入資料及輸出資料在畫面以 10 進制顯示。

### 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 (Src)	×	○	○	○	×	×	×	○	○
輸出資料 (Dest)	×	○*	○*	○*	×	×	×	○	×

\* C、# 暫存器除外

### 程式範例

將輸入資料 A (MW00000) 的 BCD(1234H(4660)) 轉換為二進制數據 (轉換為十進制的 1234)，儲存至輸出資料 (MW00001) 的程式範例，如下所示。

$$MW00000 = 1234H : (1 \times 1000) + (2 \times 100) + (3 \times 10) + 4 \rightarrow MW00001 = 1234$$



**補充** 以不同資料類型運算時，運算結果依輸出暫存器的資料類型而異。

3 章 暫存器 - ◆ 不同資料類型的演算注意事項 (第 3-8 頁)

## BCD 轉換 (BCD)

將輸入資料的二進制數據轉換為 BCD，儲存至輸出資料。

輸入資料超過 9999 或為負數時，無法得出正確結果。



(註) 輸入資料10進位標記為abcd時，輸出資料以下列式子運算。

$$\text{輸出資料} = (a \times 4096) + (b \times 256) + (c \times 16) + d$$

**補充** 輸入資料及輸出資料在畫面以 10 進制顯示。

### 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 (Src)	×	○	○	○	×	×	×	○	○
輸出資料 (Dest)	×	○*	○*	○*	×	×	×	○	×

\* C、# 暫存器除外

### 程式範例

將輸入資料 A (MW00000) 的二進制數據 (十進制的 1234) 轉換為 BCD (1234H(4660))，儲存至輸出資料 (MW00001) 的程式範例，如下所示。

$$\text{MW00000} = 1234 : (1 \times 4096) + (2 \times 256) + (3 \times 16) + 4 \rightarrow \text{MW00001} = 1234\text{H}(4660)$$



**補充** 以不同資料類型運算時，運算結果依輸出暫存器的資料類型而異。

3 章 暫存器 - ◆ 不同資料類型的演算注意事項 (第 3-8 頁)

## 同位轉換 (PARITY)

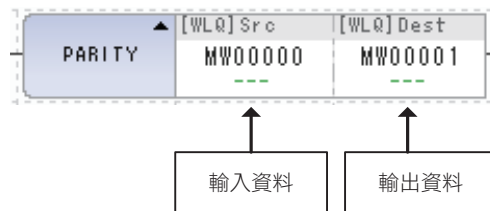
計算輸入資料 2 進位顯示之位元 1 的數後，儲存至輸出資料。

輸入資料2進位顯示之位元1的數



### 格式

格式如下所示。



圖示：**0101**  
**#?**

鍵輸入：PARITY

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 (Src)	×	○	○	○	×	×	×	○	○
輸出資料 (Dest)	×	○*	○*	○*	×	×	×	○	×

\* C、# 暫存器除外

### 程式範例

將輸入資料 (MW00000) 255 (H00FF) 的位元 1 的數儲存至輸出資料 (MW00001) 的程式範例，如下所示。

MW00000(H00FF) 的位元 1 的數 = 8 → MW00001 = 8



#### 補充

以不同資料類型運算時，運算結果依輸出暫存器的資料類型而異。

3 章 暫存器 - ◆ 不同資料類型的演算注意事項 (第 3-8 頁)

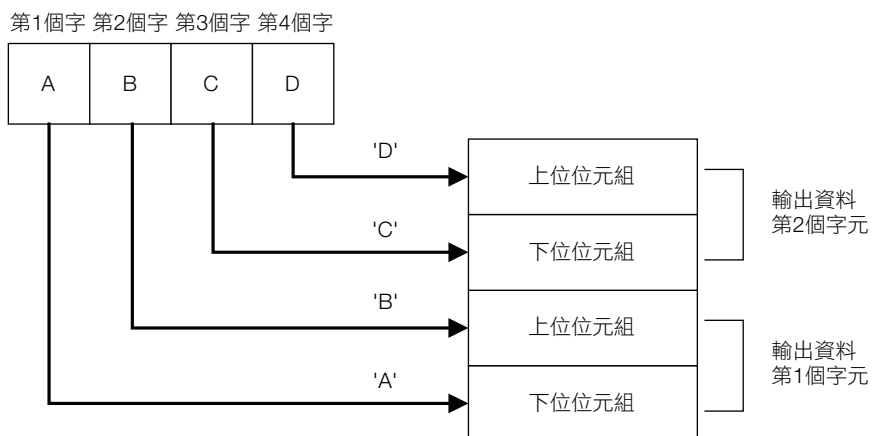
## ASCII 轉換 1 (ASCII)

將輸入字串轉為 ASCII 碼，儲存至輸出資料。可分辨大小寫。

輸入字元數最多 32 個字元 (16 個全形字)。



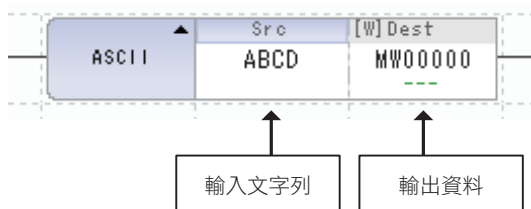
輸入字串的 ASCII 碼儲存地址，如下圖所示。



(註) 字串為奇數字時，最後一個字的上位元組為 0。

### 格式

格式如下所示。



圖示： ASCII

鍵輸入： ASCII

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入字串 (Src)	× <sup>*1</sup>								
輸出資料 (Dest)	×	○ <sup>*2</sup>	×	×	×	×	×	×	×

\*1. ASCII 字元

\*2. C、# 暫存器除外



## 程式範例

將輸入字串 Hello 轉換為 ASCII 碼儲存至輸出資料 (MW00000) 的程式範例，如下所示。

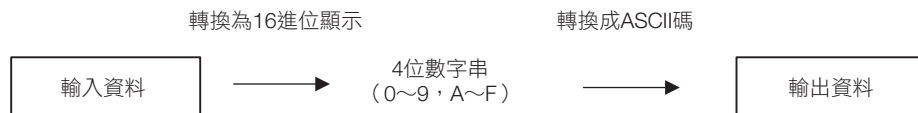


ASCII 碼如下儲存。

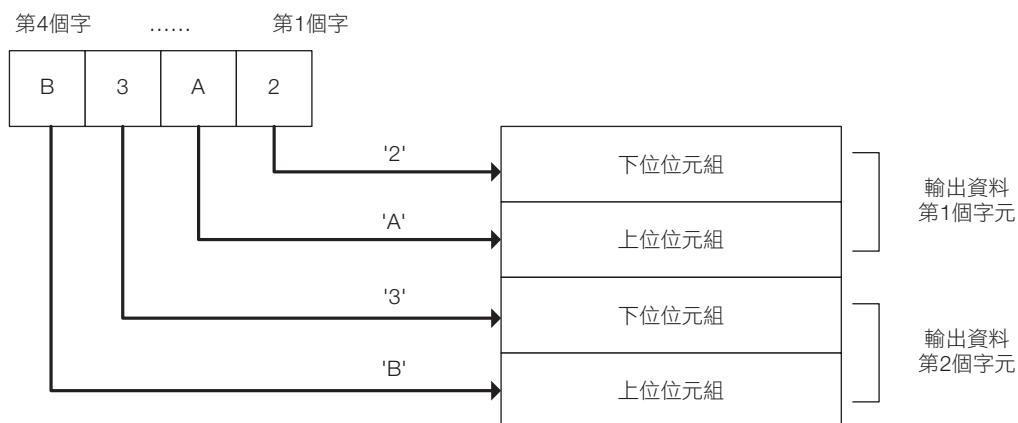
位址	ASCII 碼	文字
MW00000 (下位位元組)	48 H	H
MW00000 (上位位元組)	65 H	e
MW00001 (下位位元組)	6 CH	l
MW00001 (上位位元組)	6 CH	l
MW00002 (下位位元組)	6 FH	o
MW00002 (上位位元組)	0	-

## ASCII 轉換 2 (BINASC)

將儲存在輸入資料 (1 個字) 中的 16 位元二進制數據轉換為 16 進制 4 位數 ASCII 碼，儲存至輸出資料 (2 個字)。



輸入資料為 10811(2A3BH) 時的 ASCII 碼儲存地址，如下圖所示。



### 格式

格式如下所示。



圖示： **BIN**  
**ASC**  
鍵輸入：BINASC

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 (Src)	×	○	×	×	×	×	×	×	○
輸出資料 (Dest)	×	○*	×	×	×	×	×	×	×

\* C、# 暫存器除外

## 程式範例

將輸入資料 10811(2A3BH) 轉換為 ASCII 碼儲存至輸出資料 (MW00000) 的程式範例，如下所示。

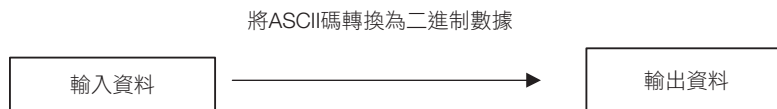


ASCII 碼如下儲存。

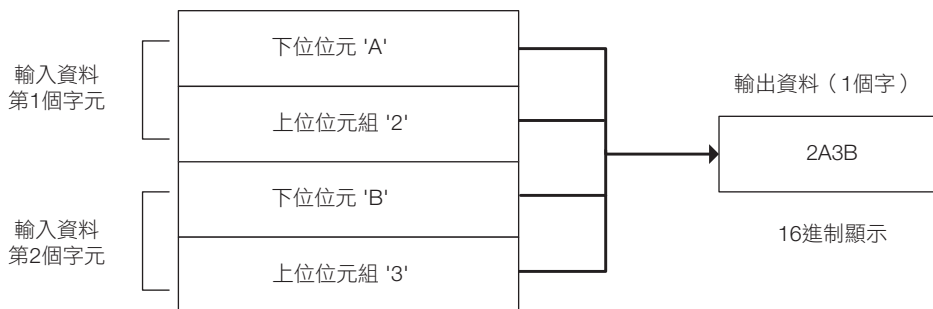
位址	ASCII 碼	文字
MW00000 (下位位元組)	32 H	2
MW00000 (上位位元組)	41 H	A
MW00001 (下位位元組)	33 H	3
MW00001 (上位位元組)	42 H	B

## ASCII 轉換 3 (ASCBIN)

將儲存在輸入資料 (2 個字) 的 16 進制 4 位數 ASCII 碼顯示數值轉換為 16 位元二進制數據，儲存於輸出資料 (1 個字)。



輸入資料的第 1 個字為 4132H ('2' 'A')，第 2 個字為 4232H ('3' 'B') 時的輸出資料，如下圖所示。



### 格式

格式如下所示。



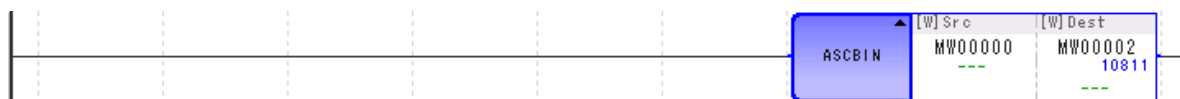
圖示：**ASC  
BIN**  
鍵輸入：ASCBIN

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 (Src)	×	○	×	×	×	×	×	×	×
輸出資料 (Dest)	×	○*	×	×	×	×	×	×	×

\* C、# 暫存器除外

## 程式範例

將輸入資料 (MW00000) 以 ASCBIN 指令儲存至輸出資料 (MW00002) 的程式範例，如下所示。



ASCII 碼將儲存如下。

位址	ASCII 碼	文字
MW00000 (下位位元組)	32 H	2
MW00000 (上位位元組)	41 H	A
MW00001 (下位位元組)	33 H	B
MW00001 (上位位元組)	42 H	3

輸出資料 (MW00000) 將變為 10811(2A3BH)。

## 4.4 邏輯運算／比較指令

### 邏輯乘積 (AND)

計算輸入資料 A 與輸入資料 B 的邏輯乘積，儲存於輸出資料。

僅整數型及 2 倍長整數型可以使用。

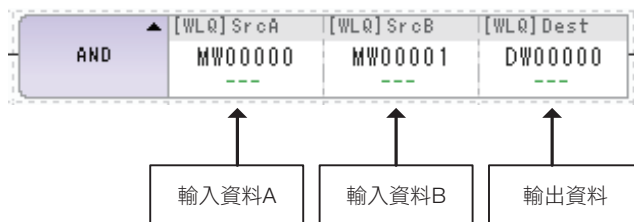



輸入資料的每個位元皆依照以下真理值表運算。

輸入資料 A	輸入資料 B	輸出資料
0	0	0
0	1	0
1	0	0
1	1	1

### 格式

格式如下所示。



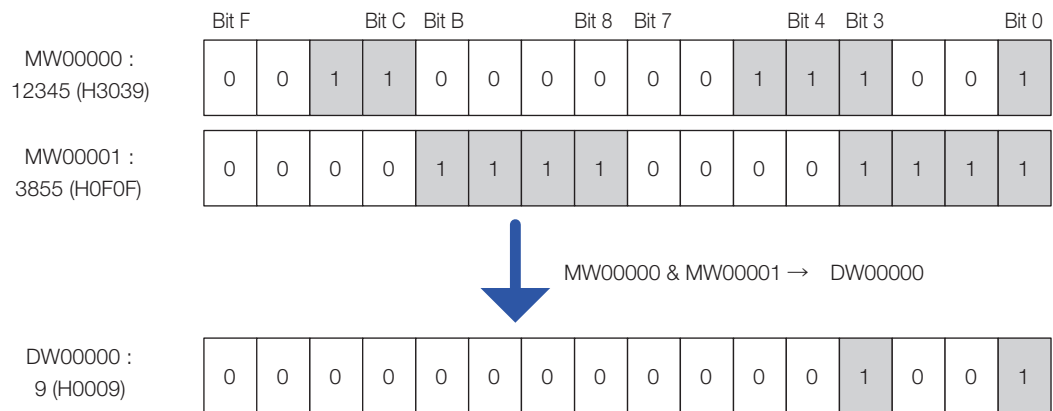
圖示：  
鍵輸入：&

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 A(SrcA)	×	○	○	○	×	×	×	○	○
輸入資料 B(SrcB)	×	○	○	○	×	×	×	○	○
輸出資料 (Dest)	×	○*	○*	○*	×	×	×	○	×

\* C、# 暫存器除外

## 程式範例

將輸入資料 A (MW00000) 的 12345 (H3039)、輸入資料 B (MW00001) 的 3855 (H0F0F) 計算其邏輯乘積，儲存至輸出資料 (DW00000) 的程式範例，如下所示。



## 邏輯或 (OR)

計算輸入資料 A 與輸入資料 B 的邏輯或，儲存於輸出資料。

僅整數型及 2 倍長整數型可以使用。

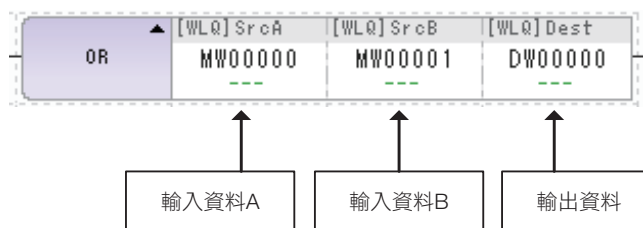


輸入資料的每個位元皆依照以下真理值表運算。

輸入資料 A	輸入資料 B	輸出資料
0	0	0
0	1	1
1	0	1
1	1	1

### 格式

格式如下所示。



圖示：



鍵輸入：|

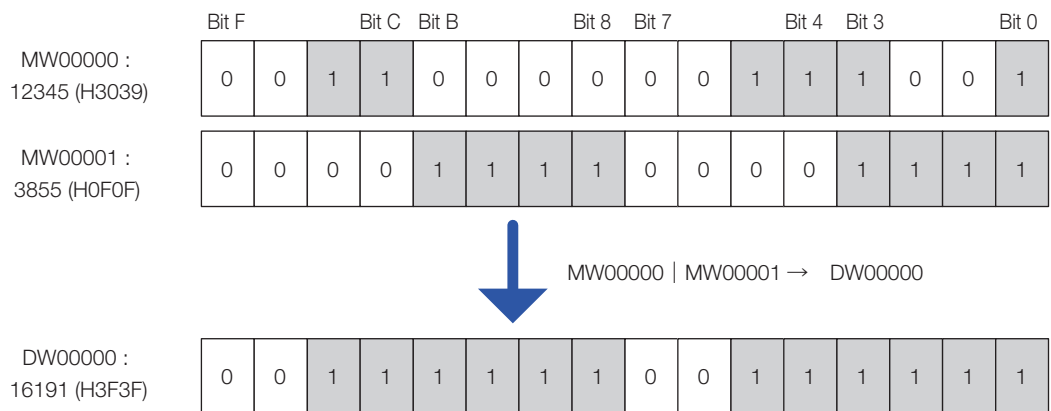
輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 A(SrcA)	×	○	○	○	×	×	×	○	○
輸入資料 B(SrcB)	×	○	○	○	×	×	×	○	○
輸出資料 (Dest)	×	○*	○*	○*	×	×	×	○	×

\* C、# 暫存器除外



## 程式範例

將輸入資料 A (MW00000) 的 12345 (H3039)、輸入資料 B (MW00001) 的 3855 (H0F0F) 計算其邏輯或，儲存至輸出資料 (DW00000) 的程式範例，如下所示。



## 互斥或 (XOR)

計算輸入資料 A 與輸入資料 B 的互斥或，儲存於輸出資料。

僅整數型及 2 倍長整數型可以使用。

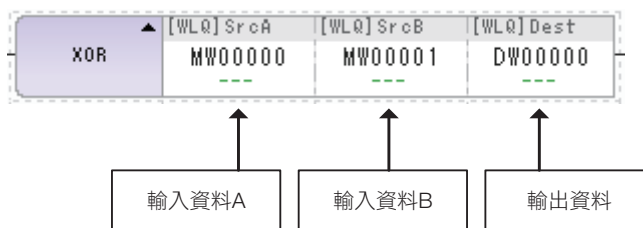



輸入資料的每個位元皆依照以下真理值表運算。

輸入資料 A	輸入資料 B	輸出資料
0	0	0
0	1	1
1	0	1
1	1	0

### 格式

格式如下所示。



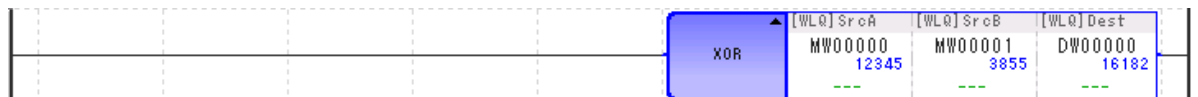
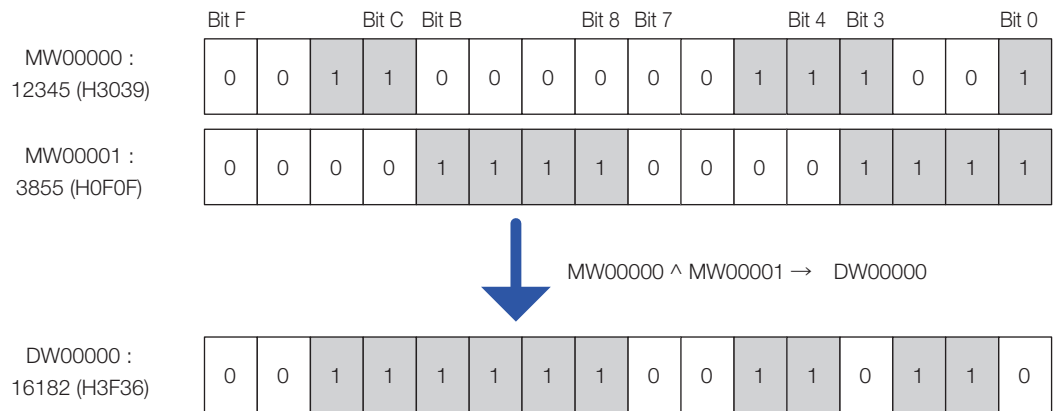
圖示：  
 鍵輸入：^

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 A(SrcA)	×	○	○	○	×	×	×	○	○
輸入資料 B(SrcB)	×	○	○	○	×	×	×	○	○
輸出資料 (Dest)	×	○*	○*	○*	×	×	×	○	×

\* C、# 暫存器除外

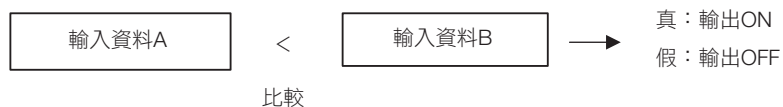
## 程式範例

將輸入資料 A (MW00000) 的 12345 (H3039)、輸入資料 B (MW00001) 的 3855 (H0F0F) 計算其互斥或，儲存至輸出資料 (DW00000) 的程式範例，如下所示。



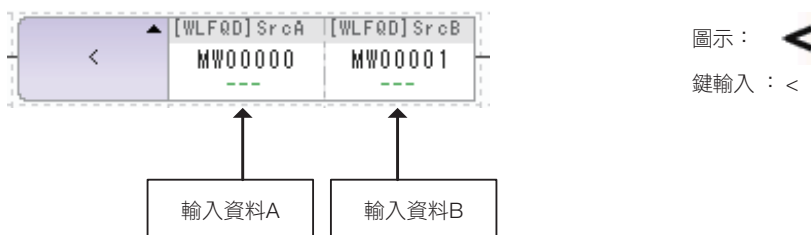
## 比較 (<)

輸入資料 A 與輸入資料 B 相較，結果以位元輸出。



### 格式

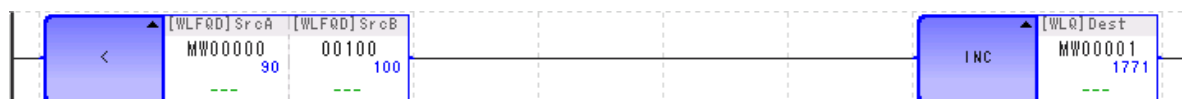
格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 A(SrcA)	×	○	○	○	○	○	×	○	○
輸入資料 B(SrcB)	×	○	○	○	○	○	×	○	○

### 程式範例

輸入資料 A (MW00000) 為 90、輸入資料 B 為常數 100 時，輸入資料 A 因未滿輸入資料 B，比較結果為真，輸出 ON，將執行右端的 INC 指令。



**補充** 若為實數型資料，則 MPE720 顯示數值會有微小精度誤差，因此可能與比較指令執行結果不一致。

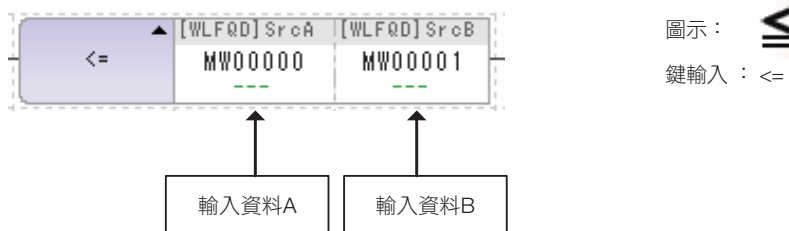
## 比較 (≤)

輸入資料 A 與輸入資料 B 相較，結果以位元輸出。



## 格式

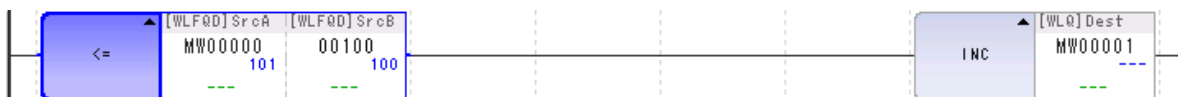
格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 A(SrcA)	×	○	○	○	○	○	×	○	○
輸入資料 B(SrcB)	×	○	○	○	○	○	×	○	○

## 程式範例

輸入資料 A (MW00000) 為 101、輸入資料 B 為常數 100 時，輸入資料 A 因超過輸入資料 B，比較結果為假，輸出 OFF，將不執行右端的 INC 指令。



**補充** 若為實數型資料，則 MPE720 顯示數值會有微小精度誤差，因此可能與比較指令執行結果不一致。

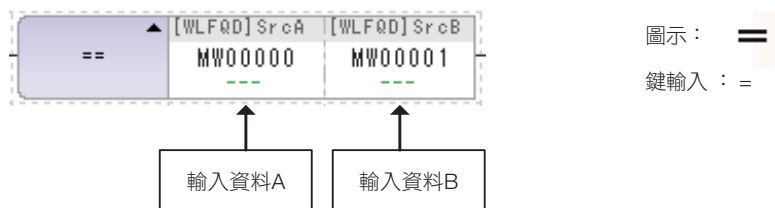
## 比較 (=)

輸入資料 A 與輸入資料 B 相較，結果以位元輸出。



### 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 A(SrcA)	×	○	○	○	○	○	×	○	○
輸入資料 B(SrcB)	×	○	○	○	○	○	×	○	○

### 程式範例

輸入資料 A (MW00000) 為 100、輸入資料 B 為常數 100 時，輸入資料 A 因等於輸入資料 B，比較結果為真，輸出 ON，將執行右端的 INC 指令。



**補充** 若為實數型資料，則 MPE720 顯示數值會有微小精度誤差，因此可能與比較指令執行結果不一致。

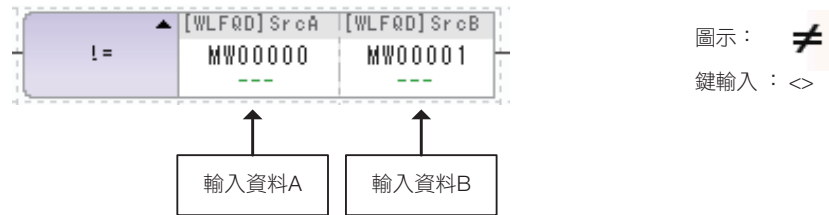
## 比較 (≠)

輸入資料 A 與輸入資料 B 相較，結果以位元輸出。



## 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 A(SrcA)	×	○	○	○	○	○	×	○	○
輸入資料 B(SrcB)	×	○	○	○	○	○	×	○	○

## 程式範例

輸入資料 A (MW00000) 為 100、輸入資料 B 為常數 100 時，輸入資料 A 因不等於輸入資料 B，比較結果為假，輸出 OFF，將不執行右端的 INC 指令。



**補充** 若為實數型資料，則 MPE720 顯示數值會有微小精度誤差，因此可能與比較指令執行結果不一致。

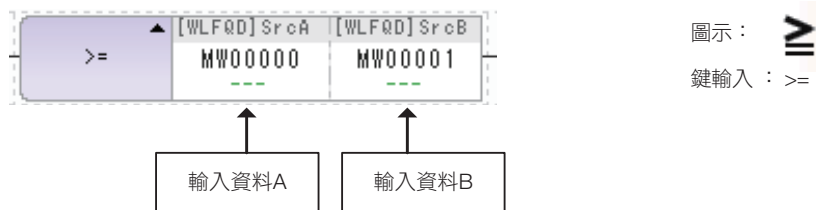
## 比較 (≥)

輸入資料 A 與輸入資料 B 相較，結果以位元輸出。



### 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 A(SrcA)	×	○	○	○	○	○	×	○	○
輸入資料 B(SrcB)	×	○	○	○	○	○	×	○	○

### 程式範例

輸入資料 A (MW00000) 為 100、輸入資料 B 為常數 100 時，輸入資料 A 因大於等於輸入資料 B，比較結果為真，輸出 ON，將執行右端的 INC 指令。



**補充** 若為實數型資料，則 MPE720 顯示數值會有微小精度誤差，因此可能與比較指令執行結果不一致。



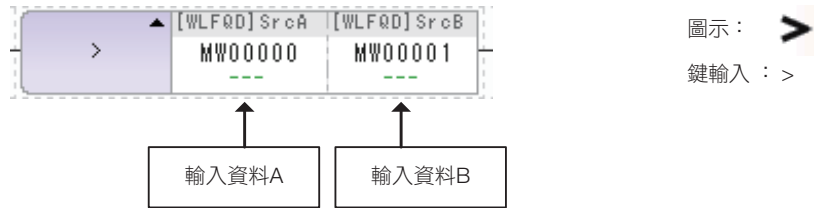
## 比較 (>)

輸入資料 A 與輸入資料 B 相較，結果以位元輸出。



## 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 A(SrcA)	×	○	○	○	○	○	×	○	○
輸入資料 B(SrcB)	×	○	○	○	○	○	×	○	○

## 程式範例

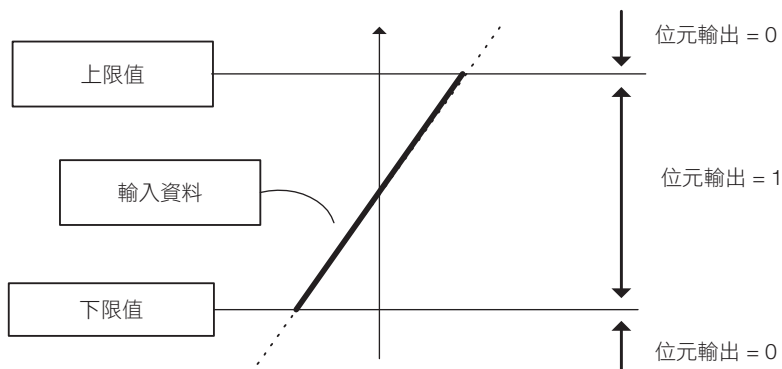
輸入資料 A (MW00000) 為 100、輸入資料 B 為常數 100 時，輸入資料 A 因未超過輸入資料 B，比較結果為假，輸出 OFF，將不執行右端的 INC 指令。



**補充** 若為實數型資料，則 MPE720 顯示數值會有微小精度誤差，因此可能與比較指令執行結果不一致。

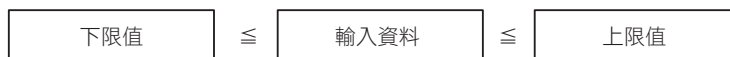
## 範圍檢查 (RCHK)

檢查輸入資料數值是否在上下限值範圍，以位元輸出結果。



### • 位元輸出 = 1

輸入資料數值在下限值以上且上限值以下的範圍時，位元輸出為 1。

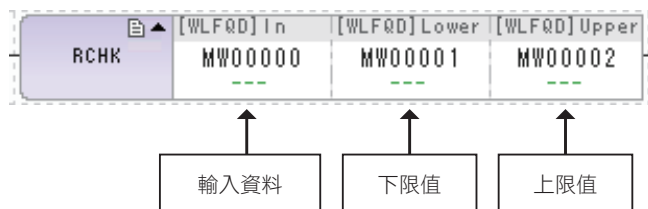


### • 位元輸出 = 0

輸入資料數值不在下限值以上且上限值以下範圍時，位元輸出為 0。

## 格式

格式如下所示。



圖示：  
鍵輸入：RCHK

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 (In)	×	○	○	○	○	○	×	○	○
下限值 (Lower)	×	○	○	○	○	○	×	○	○
上限值 (Upper)	×	○	○	○	○	○	×	○	○

### 補充

請務必設定為下限值 ≤ 上限值。如設為下限值 > 上限值，無法保證其動作。

## 程式範例

執行 RCHK 指令的程式範例，如下所示。

- 輸入資料 (MW00000) = 80，下限值 = 100，上限值 = 1000 時  
輸入資料數值小於下限值，位元輸出為 0，不執行右端的 INC 指令。



- 輸入資料 (MW00000) = 500，下限值 = 100，上限值 = 1000 時  
輸入資料數值在下限值以上且上限值以下範圍，位元輸出為 1，執行右端的 INC 指令。



- 輸入資料 (MW00000) = 1000，下限值 = 100，上限值 = 1000 時  
輸入資料數值在下限值以上且上限值以下範圍，位元輸出為 1，執行右端的 INC 指令。



# 4.5

## 程式控制指令

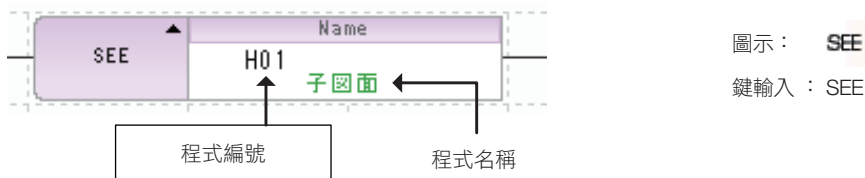
### 呼叫圖面 (SEE)

總圖引用子圖，或子圖引用孫圖。



### 格式

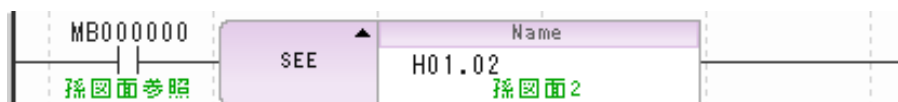
格式如下所示。



輸出輸入項目	可以使用的資料類型
程式編號 (Name)	於暫存器內無法指定，需直接指定程式編號。 指定的程式名稱將顯示在程式編號下方。

### 程式範例

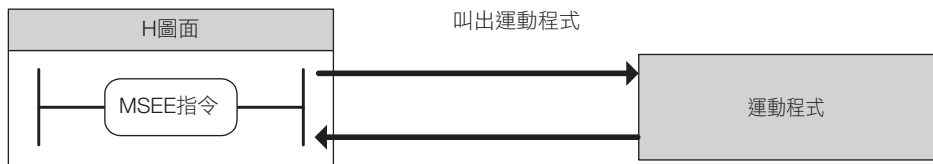
繼電器 (MB000000) 為 ON 時，以 SEE 指令叫 H01.02。其後進行處理，執行 SEE 指令的下一階段。  
繼電器 (MB000000) 為 OFF 時，不呼叫 H01.02。



## 呼叫運動程式 (MSEE)

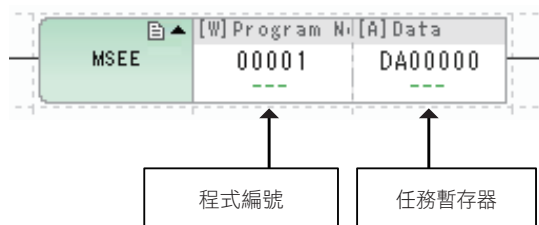
引用指定的運動程式。

運動程式只可由 H 圖面引用。



### 格式

格式如下所示。



圖示：  
M See  
鍵輸入：MSEE

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
程式編號 (Program No.)	×	○*	×	×	×	×	×	×	○
任務暫存器 (Data)	×	×	×	×	×	×	○*	×	×

\* 僅限 M、D 暫存器

工作暫存器的整體構成如下所示。

位址	資料類型	名稱	內容	輸出輸入
0	W	狀態標誌	運動程式的狀態標誌	OUT
1	W	控制訊號	運動程式的控制訊號	IN
2	W	補間用覆寫	執行補間系統指令時的覆寫值 範圍：0 ~ 32767 單位：1 = 0.01%	IN
3	W	系統任務編號	呼叫運動程式系統編號	IN

#### 補充

程式編號請設為 1 ~ 512。

關於運動程式的詳情，請參閱以下手冊。

📖 MP3000 系列 運動程式 程式設計手冊 (資料編號：SIJP C880725 04)

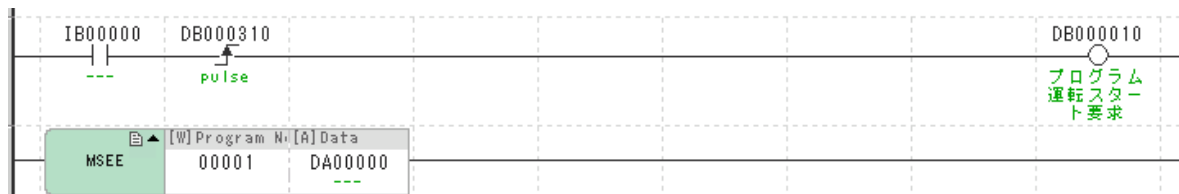
## 程式範例

執行程式編號 1 的運動程式 (MPM001) 時，其程式範例如下所示。

繼電器 (IB00000) 為 ON，控制訊號中的程式開始運行要求 (DB000010) 轉為 ON，執行運動程式 (MPM001)。

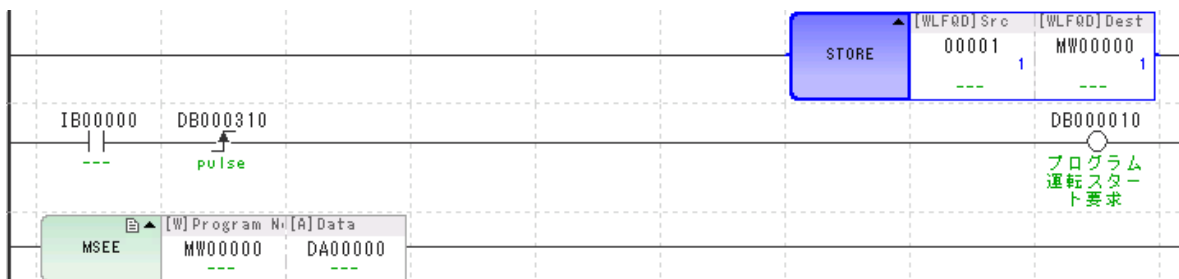
- 直接指定

直接設定程式編號為「1」。



- 間接指定

設定程式編號為「MW00000」。




在運動程式結束執行前，請持續執行 MSEE 指令。

如果是間接指定，在運動程式結束執行前請勿變更暫存器的數值。

## 呼叫使用者函數 (FUNC)

引用使用者函數。欲引用的使用者函數，請事先完善其定義。呼叫使用者函數 (FUNC) 所用的巢套，最多可達 8 階段 (8 個巢套)。

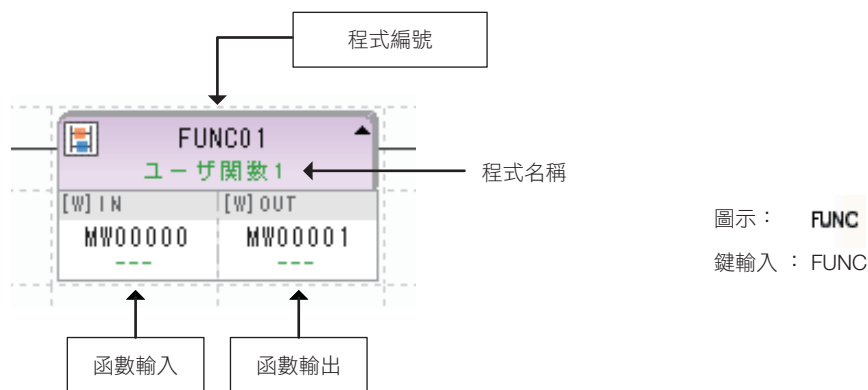
使用者函數的相關詳情，請參照下列項目。

 1.3 概要 – 使用者函數 (第 1-13 頁)



### 格式


格式如下所示。



輸出輸入項目	可以使用的資料類型
程式編號 (Name)	於暫存器內無法指定，需直接指定程式編號。 指定的程式名稱將顯示在指令上方。
函數輸入	可使用函數輸入定義中設定的暫存器。
函數輸出	可使用函數輸出定義中設定的暫存器。

### 程式範例

關於使用者函數的程式範例，請參照下列項目。

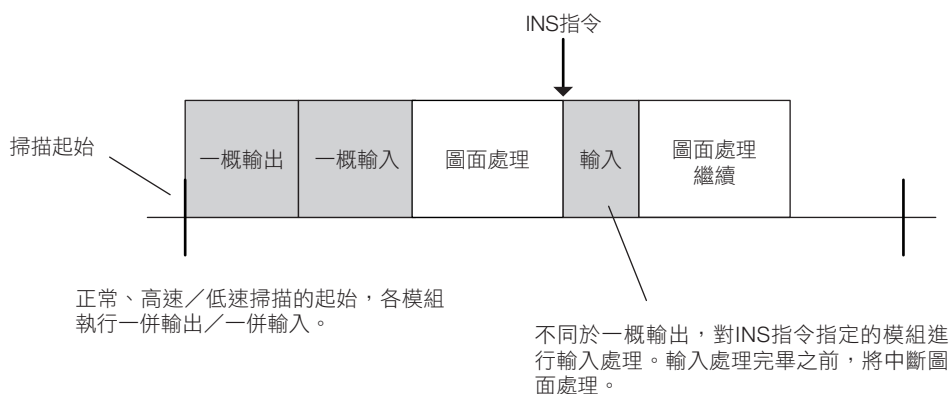
 1.3 概要 – 使用者函數 (第 1-13 頁)

## 直接輸入連續執行形 (INS)

在高速掃描/低速掃描開頭由系統執行的一概輸出入之外，以使用者程式執行 INS 指令。當 INS 指令執行時，依據設定的參數表內容執行來自指定模組的輸入。在輸入動作完畢之前，不執行下一個指令。

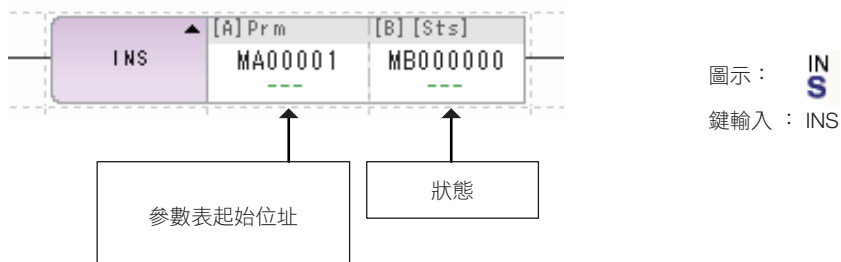
可指定模組如下所示。

- LIO-01/02 模組 (LIO)
- LIO-04/05 模組 (LIO32)
- LIO-06 模組 (MIXIO)
- AI-01 模組 (AI)



### 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
參數表起始位址 (Prm)	×	×	×	×	×	×	O <sup>*1</sup>	×	×
狀態 (Sts) <sup>*2</sup>	O <sup>*1</sup>	×	×	×	×	×	×	×	×

\*1. C、# 暫存器除外

\*2. 可省略



參數表配置如下所示。

位址	資料類型	記號	名稱	規格	輸出輸入
0	W	RSSEL	單元指定 1	指定輸入的模組。	IN
1	W	MDSEL	單元指定 2		IN
2	W	STS	狀態	每 1 個字的輸入狀態，以對應位元輸出。 0：正常 1：異常	OUT
3	W	N	字數	指定連續輸入字數。	IN
4	W	ID1	輸入資料 1	將輸入的資料進行輸出。 如發生錯誤，儲存為 0。	OUT
:	:	:	:		:
N + 3	W	IDN	輸入資料 N		OUT

MP3000 系列的各參數詳情如下所示。

參數	模組名稱			
	LIO-01/02 (LIO)	LIO-04/05 (LIO32)	LIO-06 (MIXIO)	AI-01 (AI)
RSSEL	指定模組固定架／單元／插槽／子插槽編號。 16 進位標示：zxuyH x：固定架編號 (1～7) u：單元編號 (0～4) <sup>*1</sup> y：插槽編號 (0～9) z：子插槽編號 (1～最大值依據模組規格而定)			
MDSEL	0 (未使用)	偏移量： 0～1	頻道編號 -1：0～1	頻道編號 -1：0～7
STS	固定為 0	固定為 0	固定為 0	*2
N	1	1～2 -MDSEL	1～2 -MDSEL	1～8 -MDSEL

\*1. 若單元編號指定為 0 時，將視為單元編號 = 1 執行處理。

\*2. 如果在 AI 模組詳細定義中的分配因 INS 指令而指定成遭到刪除的通道時，以位元輸出符合的通道編號。這是由於分配功能遭到刪除的頻道無法存取資料。位元與頻道之間的關係如下所示。

位元 0：頻道 1  
位元 1：頻道 2  
位元 2：頻道 3  
位元 3：頻道 4  
位元 4：頻道 5  
位元 5：頻道 6  
位元 6：頻道 7  
位元 7：頻道 8

## 程式範例

當位於固定架 1、單元 1 插槽 2 的 LIO-01 模組上的 LIO (子插槽編號：1) 輸入 1 個字時，狀態 (MW00014) 中將儲存 LIO 的輸入資料。

EXPRESSION

```
MW00010 = 0x1102; //ラック、スロット、サブスロット番号
4354=4354
MW00011 = 0; //未使用
0=0
MW00013 = 1; //ワード数
1=1
```

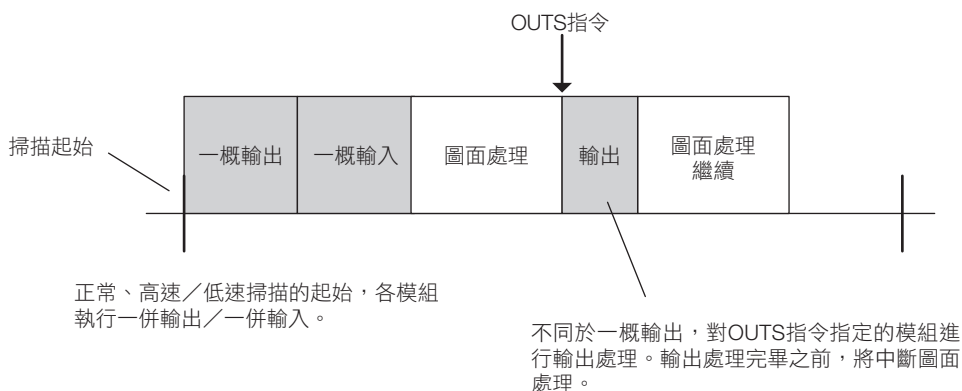
INS [A] Prm MA00010 [B] [Sts] MB000000

## 直接輸出連續執行形 (OUTS)

在高速掃描/低速掃描開頭由系統執行的一概輸出入之外，以使用者程式執行 OUTS 指令。當 OUTS 指令執行時，依據設定的參數表內容執行來自指定模組的輸出。

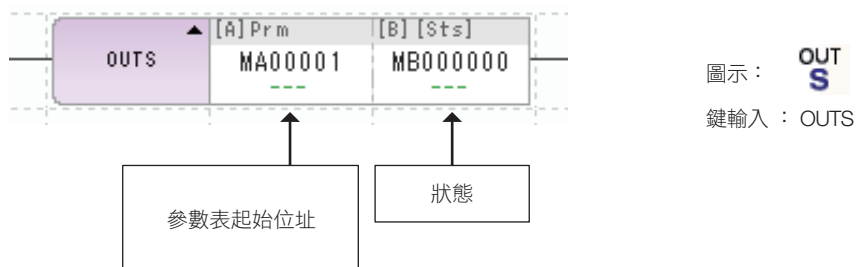
可指定模組如下所示。

- LIO-01/02 模組 (LIO)
- LIO-04/05 模組 (LIO32)
- LIO-06 模組 (MIXIO)
- DO-01 模組 (DO)
- AO-01 模組 (AO)



### 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
參數表起始位址 (Prm)	×	×	×	×	×	×	O <sup>*1</sup>	×	×
狀態 (Sts) <sup>*2</sup>	O <sup>*1</sup>	×	×	×	×	×	×	×	×

\*1. C、# 暫存器除外

\*2. 可省略

參數表配置如下所示。

位址	資料類型	記號	名稱	規格	輸出輸入
0	W	RSSEL	單元指定 1	指定輸出的模組。	IN
1	W	MDSEL	單元指定 2		IN
2	W	STS	狀態	每 1 個字的輸入狀態，以對應位元輸出。 0：正常 1：異常	OUT
3	W	N	字數	指定連續輸出字數（固定為 1）。	IN
4	W	OD1	輸出資料 1	設定欲輸出的資料。	OUT
:	:	:	:		:
N + 3	W	ODN	輸出資料 N		OUT

MP3000 系列的各參數詳情如下所示。

參數	模組名稱				
	LIO-01/02 (LIO)	LIO-04/05 (LIO32)	LIO-06 (MIXIO)	DO-01 (DO)	AO-01 (AO)
RSSEL	指定模組固定架／單元／插槽／子插槽編號。 16 進位標示：zxuyH x：固定架編號（1～7） u：單元編號（0～4）*1 y：插槽編號（0～9） z：子插槽編號（1～最大值依據模組規格而定）				
MDSEL	0 (未使用)	偏移量： 0～1	偏移量： 0～1	偏移量： 0～3	頻道編號 -1：0～3
STS	固定為 0	固定為 0	固定為 0	固定為 0	*2
N	1	1～2 - MDSEL	1～2 - MDSEL	1～4 - MDSEL	1～4 - MDSEL

\*1. 若單元編號指定為 0 時，將視為單元編號 = 1 執行處理。

\*2. 如果在 AO 模組詳細定義中的分配因 OUTS 指令而指定成遭到刪除的通道時，以位元輸出符合的通道編號。這是由於分配功能遭到刪除的頻道無法存取資料。位元與頻道之間的關係如下所示。

位元 0：頻道 1  
位元 1：頻道 2  
位元 2：頻道 3  
位元 3：頻道 4

## 程式範例

當位於固定架 1、單元 1 插槽 2 的 LIO-01 模組上的 LIO（子插槽編號：1）輸出 1 個字時，狀態（MW00014）的資料將輸出至 LIO。

```

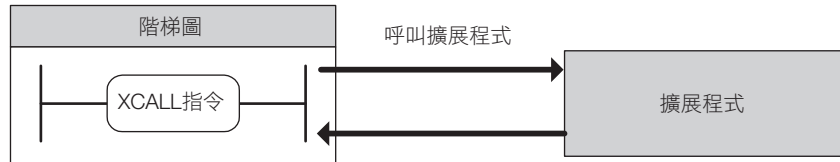
EXPRESSION
MW00010 = 0x1102; //ラック、スロット、サブスロット番号
4354=4354
MW00011 = 0; //未使用
0=0
MW00013 = 1; //ワード数(1固定)
1=1
MW00014 = 0x000F; //出力データ
15=15
  
```

OUTS [A] Prm [B] [Sts]  
MA00010 MB000000 1

## 執行擴展程式 (XCALL)

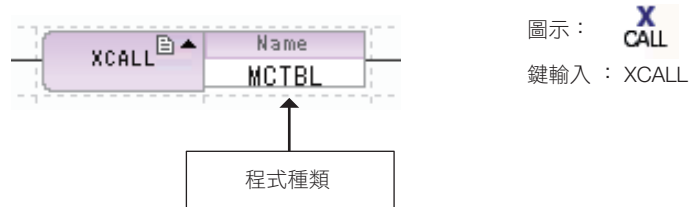
執行擴展程式 (表列程式：常數表)。MPE720 將這些擴展程式轉換為階梯圖程式。轉換後的階梯圖程式，可使用 XCALL 指令執行。

一張圖面中可使用複數 XCALL 指令，但同一擴展程式當中不得呼叫超過 1 次。



### 格式

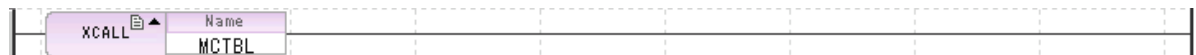
格式如下所示。



輸出輸入項目	可以使用的資料類型
程式種類 (Name)	於暫存器內無法指定，將設定下列種類。 MCTBL：常數表

### 程式範例

呼叫 MCTBL (常數表) 時的範例。

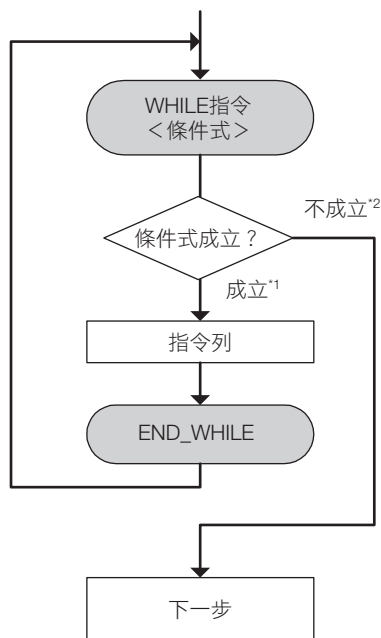


**補充** MP3000 系列無法使用輸出變換表、互鎖表、零件組裝表。

## WHILE 文 (WHILE, END\_WHILE)

當 WHILE 指令的條件式成立，將執行 WHILE 指令與 END\_WHILE 指令間的程式（指令列），並返回 WHILE 指令。條件式成立的期間，將重覆執行。

條件式若不成立，則以 END\_WHILE 指令進到下一個步驟。由 WHILE 指令至 END\_WHILE 指令之間的指令列將不執行。



\*1. 指令列被執行，再次返回 WHILE 指令。

\*2. 指令列未執行，進到下一步驟。

### 格式

格式如下所示。



圖示： WHILE END\_WHILE

鍵輸入： WHILE, WEND

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
條件式	○*	○*	○*	○*	○*	○*	×	○*	○*

\* 以 EXPRESSION 語法記述。

關於可記述的格式，請參照以下章節。

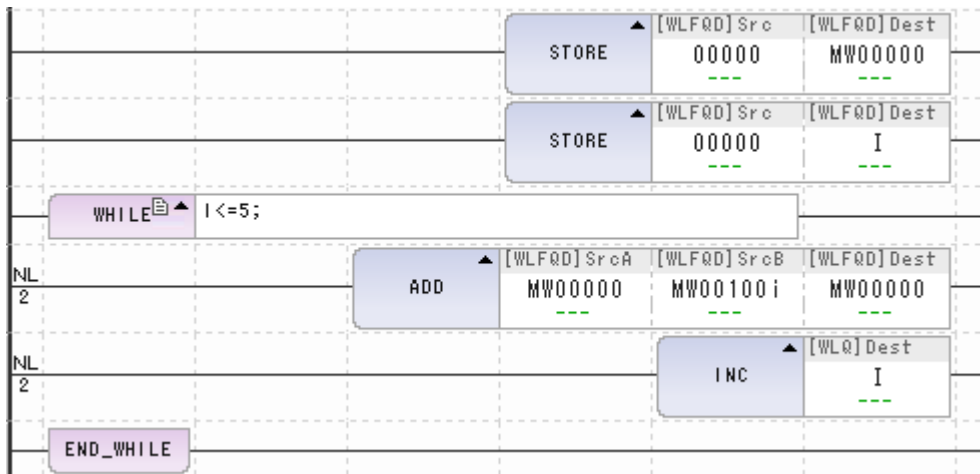
附錄 C EXPRESSION 指令的格式

## 程式範例

MW00100 ~ MW00105 累加計算後儲存於 MW00000 的程式範例如下所示。

條件式為  $I \leq 5$ ，ADD(+) 指令在  $I$  為 0 ~ 5 時執行。

$I = 6$  時，條件式不成立，以 END\_WHILE 指令進到下一個步驟。



WHILE 指令在條件式成立的期間，將重覆指令列。

條件式非不成立，或條件式成立費時時，運動控制器將會癱瘓停機，請注意。  
前述例子中，命令列裡的  $I$  持續未增量時，將形成無限迴圈。

## 補充事項

### ◆ 可使用的條件式

WHILE 指令的條件式，僅能以如下般的 bool 型（真或假）結果輸出之 EXPRESSION 公式敘述。因此無法辨識包含指定運算子在內的公式記述。

公式範例	記述	備註
MB000001 == true	OK	真：ON
MB000001 != false	OK	假：OFF
MW000002 < 100	OK	-
MF000002 < sin(60.0)	OK	-
MW000001 == 0x00FF OK	OK	16 進制顯示使用 0x
MB000001 = true	NG	-
MW000001 = MW000002	NG	-

(註) 關於可使用的指令、運算順序及表示方法等詳情，請參閱下列章節。

附錄 C EXPRESSION 指令的格式

### ◆ 結構語句的深奧（巢狀結構）

FOR 結構文、WHILE 結構文、IF 結構文之內，可包容其他結構文。此即稱為巢狀結構。使用 FOR 結構語句、WHILE 結構語句和 IF 結構語句的巢狀結構，最多可達 8 階段（8 層）。

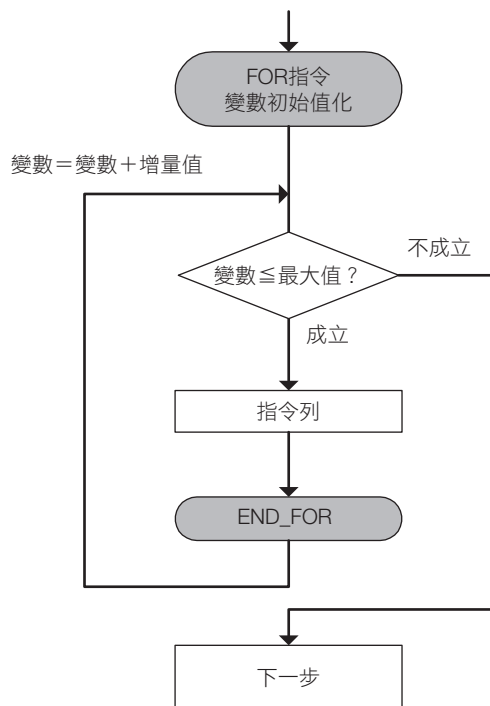
在接點後繼續記述指令時，視同 IF 結構語句，包含在巢狀階層內。

## FOR 文 (FOR, END\_FOR)

重覆執行 FOR 指令與 END\_FOR 指令間的指令列。

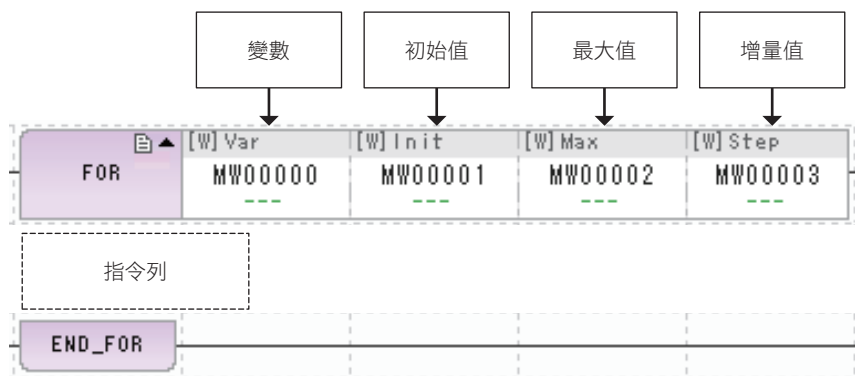
指定為變數的暫存器自初始值起，每重覆執行一次則加算其增加值。

當變數超過最大值，FOR 指令條件式則不成立，進到下一步驟。



### 格式

格式如下所示。



圖示： FOR, END\_FOR  
鍵輸入：FOR, FEND

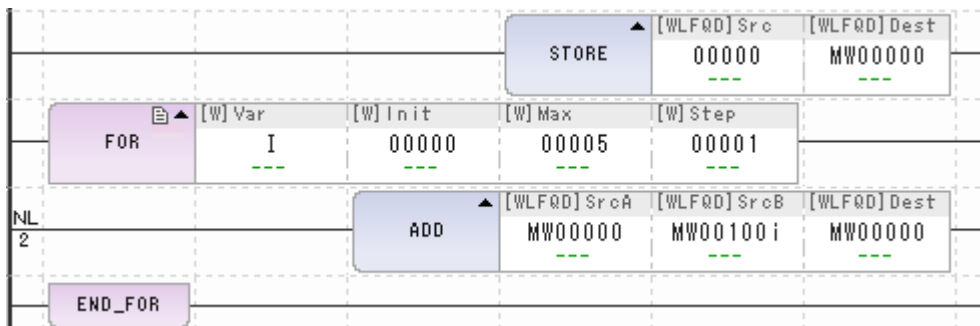
輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
變數 (Var)	×	○*	×	×	×	×	×	○	×
初始值 (Init)	×	○	×	×	×	×	×	○	○
最大值 (Max)	×	○	×	×	×	×	×	○	○
增加值 (Step)	×	○	×	×	×	×	×	○	○

\* C、# 暫存器除外

## 程式範例

MW00100 ~ MW00105 累加計算後儲存於 MW00000 的程式範例如下所示。

將變數 I 初始化 (本例中以 0 儲存)。其後直到變數 (I) 超過最大值 (5) 為止，執行 ADD (+) 指令。I = 6 時，條件式不成立，以 END\_FOR 指令進到下一個步驟。



## 補充事項

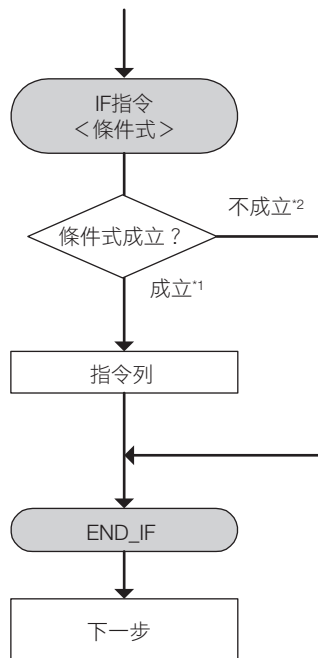
在 FOR 結構語句、WHILE 結構語句和 IF 結構語句中，可包含其他的結構語句。此即稱為巢狀結構。使用 FOR 結構語句、WHILE 結構語句和 IF 結構語句的巢狀結構，最多可達 8 階段 (8 層)。

在接點後繼續記述指令時，視同 IF 結構語句，包含在巢狀階層內。



## IF 文 (IF, END\_IF)

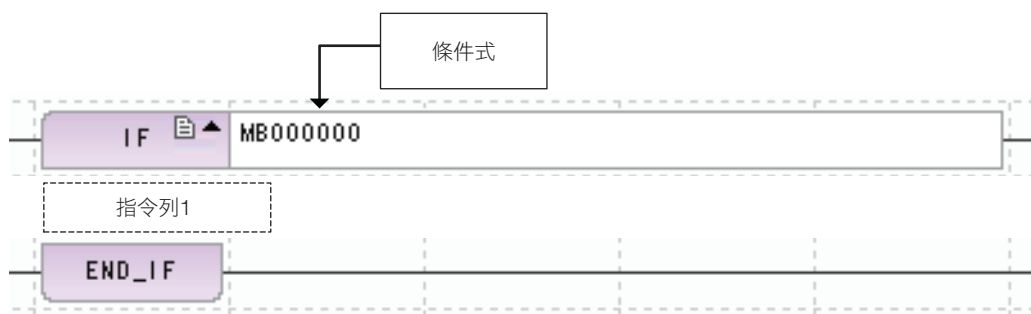
當 IF 指令條件式成立，執行 IF 指令與 END\_IF 指令間的指令列。  
條件式不成立時，不執行指令列。



- \*1. 執行指令列，進到下一步驟。
- \*2. 未執行指令列，進到下一步驟。

### 格式

格式如下所示。



圖示： IF, END\_IF  
鍵輸入：IF, IEND

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
條件式	○*	○*	○*	○*	○*	○*	×	○*	○*

\* 以 EXPRESSION 語法記述。  
關於可記述的格式，請參照以下章節。  
📖 附錄 C EXPRESSION 指令的格式

## 程式範例

IF 指令的條件式 (MB000001) 為 ON 時，MW00010 的值設為 MW01000，並對 MW00011 遞增計數。



## 補充事項

### ◆ 可使用的條件式

IF 指令的條件式，僅能以如下般的 bool 型（真或假）結果輸出之 EXPRESSION 公式敘述。因此無法辨識包含指定運算子在內的公式記述。

公式範例	記述	備註
MB000001 == true	OK	true (真) : ON
MB000001 != false	OK	false (假) : OFF
MW00002 < 100	OK	-
MF00002 < sin(60.0)	OK	-
MW00001 == 0x00FF OK	OK	16 進位標示使用「0x」
MB000001 = true	NG	-
MW00001 = MW00002	NG	-

(註) 關於可使用的指令、運算順序及表示方法等詳情，請參閱下列章節。

📖 附錄 C EXPRESSION 指令的格式

### ◆ 結構語句的深奧（巢狀結構）

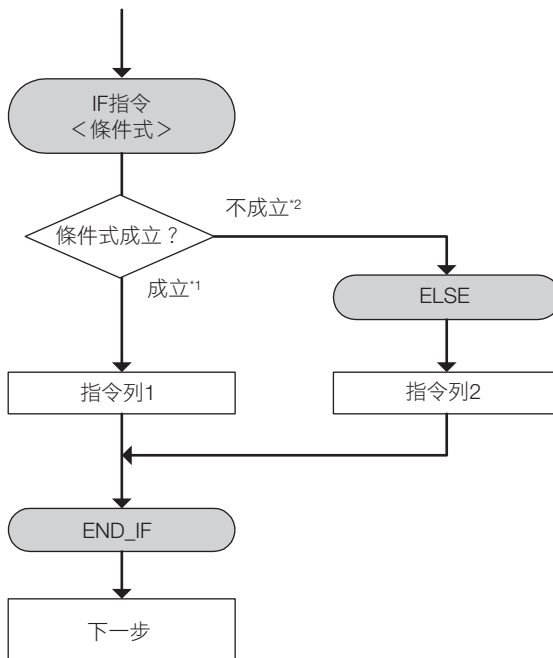
在 FOR 結構語句、WHILE 結構語句和 IF 結構語句中，可包含其他的結構語句。此即稱為巢狀結構。使用 FOR 結構語句、WHILE 結構語句和 IF 結構語句的巢狀結構，最多可達 8 階段（8 層）。

在接點後繼續記述指令時，視同 IF 結構語句，包含在巢狀階層內。

## IF-ELSE 文 (IF, ELSE, END\_IF)

當 IF 指令的條件式成立，僅執行指令列 1，不執行指令列 2。

當條件式不成立，僅執行指令列 2，不執行指令列 1。



\*1. 執行指令列 1，前往下一步。

\*2. 執行指令列 2，前往下一步。

### 格式

格式如下所示。



圖示： IF, ELSE, END\_IF

鍵輸入： IF, ELSE, IEND

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
條件式	○*	○*	○*	○*	○*	○*	×	○*	○*

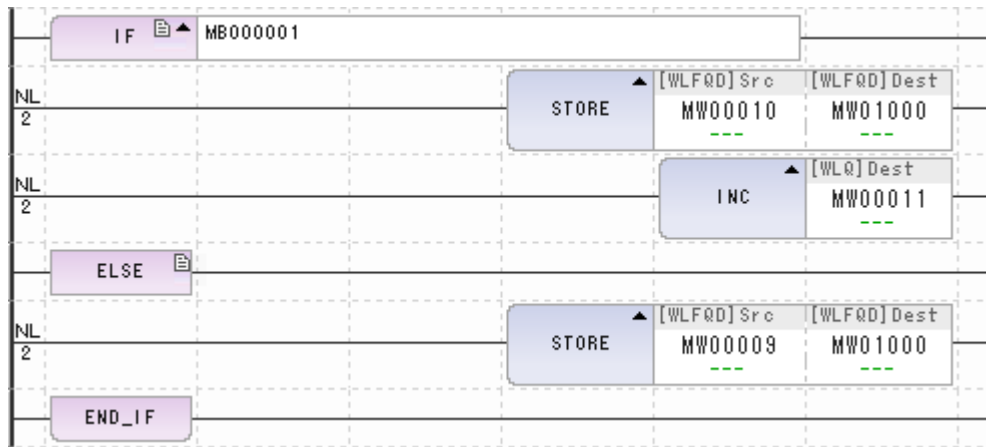
\* 以 EXPRESSION 語法記述。

關於可記述的格式，請參照以下章節。

附錄 C EXPRESSION 指令的格式

## 程式範例

IF 指令的條件式 (MB000001) 為 ON 時, MW00010 的值設為 MW01000, 並對 MW00011 遞增計數。IF 指令的條件式 (MB000001) 為 OFF 時, MW00009 的值設為 MW01000。



## 補充事項

可用的條件式、結構深度均與 IF 結構文相同。

## 公式敘述 (EXPRESSION)

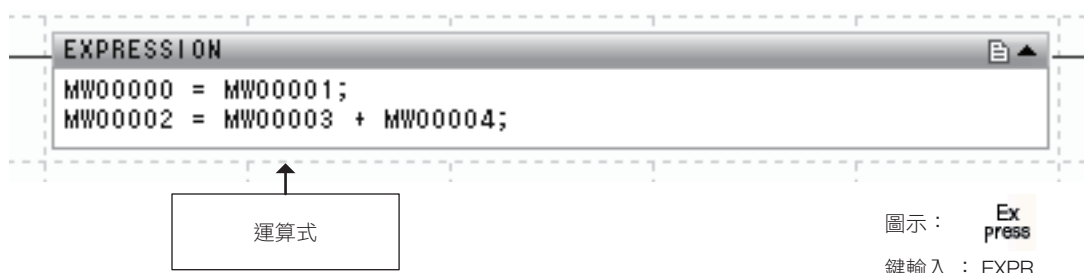
可敘述下列內容。

- 與 C 語言相同，不是暫存器而是變數名稱與結構體的敘述
- sin 函數、cos 函數等基本函數
- 算術運算符、邏輯運算符、比較運算符、代入運算符
- 配列

EXPRESSION指令
<pre>MW00000 = 10; MW00001 = DATA1; ML00002 = MW00000 + 100; MF00004 = sin(MF00006); MW00006 = 0x3FFF;  ⋮</pre>

### 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
運算式	○*	○*	○*	○*	○*	○*	×	○*	○*

\* 以 EXPRESSION 語法記述。

關於可記述的格式，請參照以下章節。

附錄 C EXPRESSION 指令的格式

## 程式範例

EXPRESSION 指令中敘述複數運算時的程式範例，如下所示。

```

EXPRESSION
DW00000 = DW00001 * DW00001 + DW00002 * DW00002;
25=3*3+4*4
DW00004 = 10 + 10 * 2;
30=10+10*2
DW00004 = (10 + 10) * 2;
40=(10+10)*2
DF00008 = sin(DF00006);
5.000000E-001=sin(3.000000E+001)

```

## 補充事項

能以如下般的 bool 型（真或假）結果輸出之外的 EXPRESSION 公式敘述。

公式範例	記述	備註
MB000000 = true;	OK	true (真) : ON
MW00000 = MW00001+10	OK	-
MW00000 = 0x00FF;	OK	16 進位標示使用「0x」
MB000000 == true;	NG	-
MW00001 > MW00000;	NG	-

(註) 關於可使用的指令、運算順序及表示方法等詳情，請參閱下列章節。

 附錄 C EXPRESSION 指令的格式

## 4.6

## 基本函數指令

### 平方根 (SQRT)

計算整數型或實數型輸入資料的平方根，儲存輸出資料。

不使用 2 倍長整數型。

**補充** 當輸入資料未滿 0，則輸出使用輸入資料絕對值的計算結果。

#### ◆ 整數型 SQRT：輸入及輸出資料均為整數型時



**補充** 整數型 SQRT 指令將以有別於數學中開平方的計算式進行計算。

$$\text{sign}(\text{輸入資料}) \times \sqrt{|\text{輸入資料}| \times 32768}$$

換句話說，是將數學開平方後的結果乘以 $\sqrt{32768}$ 之後的值。另外，輸入值為負數時，則計算其絕對值的平方根，以其負數作為計算結果。計算誤差最大為  $\pm 2$ 。

#### ◆ 實數型 SQRT：前述以外的情況



**補充** 將前一個計算結果（實數型資料）當作輸出，將其平方根留作實數型資料。

### 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 (Src)	×	○	×	×	○	○	×	○	○
輸出資料 (Dest)	×	○*	×	×	○*	○*	×	○	×

\* C、# 暫存器除外

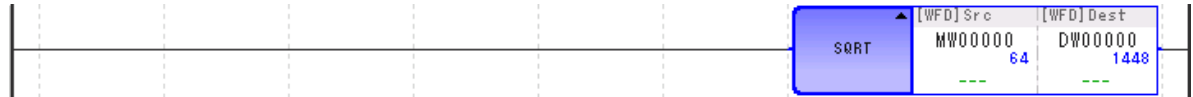
## 程式範例

使用整數型以及實數型輸入資料的 SQRT 指令程式範例，如下所示。

- 整數型 SQRT

整數型輸入資料 (MW00000) 64 的平方根，乘以  $128\sqrt{2}$  後的結果儲存為輸出資料 (DW00000)。

$$\sqrt{64} \times 128\sqrt{2} \rightarrow DW00000 = 1448$$



- 實數型 SQRT

計算實數型輸入資料 (MF00000) 64.0 的平方根，其結果儲存為輸出資料 (DF00000)。

$$\sqrt{64.0} \rightarrow DF00000 = 8.0$$





## 正弦 (SIN)

計算整數型或實數型輸入資料的正弦，儲存輸出資料。

不使用 2 倍長整數型。

### ◆ 輸入資料及輸出資料為整數型時

SIN (  ) × 10000 →

- (註) 1. 輸入資料的單位為「度」(1=0.01)。  
2. 將其 10000 倍的數值作為計算結果輸出為輸出資料。

### ◆ 輸入資料及輸出資料為實數型時

SIN (  ) →

(註) 輸入資料的單位為「度」。

## 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 (Src)	×	○	×	×	○	○	×	○	○
輸出資料 (Dest)	×	○*	×	×	○*	○*	×	○	×

\* C、# 暫存器除外

### ◆ 整數型時

輸入資料的單位為「度」(1 = 0.01)。

因此，-327.78 度 ~ 327.67 度範圍內可進行 SIN 函數運算。

sin 函數輸出時將乘以 10000，因此資料將在 -10000 ~ 10000 的範圍內輸出。

### ◆ 實數型時

輸入資料的單位為「度」。

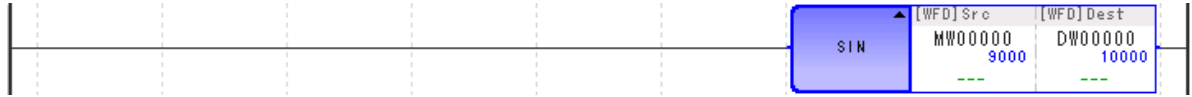
## 程式範例

使用整數型以及實數型輸入資料的 SIN 指令程式範例，如下所示。

- 整數型 SIN

計算整數型輸入資料 (MW00000) 9000 的正弦，其結果儲存為輸出資料 (DW00000)。

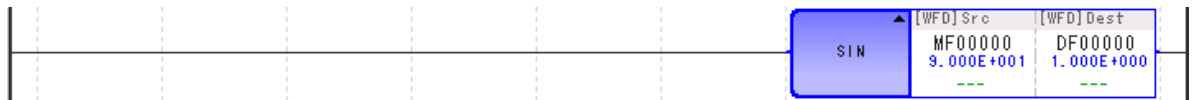
$$\text{SIN} (90.00 \text{ 度}) \times 10000 \rightarrow \text{DW00000} = 10000$$



- 實數型 SIN

計算實數型輸入資料 (MF00000) 90.0 的正弦，其結果儲存為輸出資料 (DF00000)。

$$\text{Sin} (90.0 \text{ 度}) \rightarrow \text{DF00000} = 1.0$$



## 餘弦 (COS)

計算整數型或實數型輸入資料的餘弦，儲存輸出資料。

不使用 2 倍長整數型。

### ◆ 輸入資料及輸出資料為整數型時

COS (  ) × 10000 →

- (註) 1. 輸入資料的單位為「度」(1 = 0.01)。  
 2. 將其 10000 倍的數值作為計算結果儲存為輸出資料。  
 3. 輸入資料範圍為 -327.68 ~ 32.767 度。除此之外則無法得出正確結果。

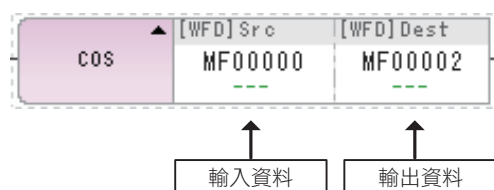
### ◆ 輸入資料及輸出資料為實數型時

COS (  ) →

(註) 輸入資料的單位為「度」。

## 格式

格式如下所示。



圖示： **COS**

鍵輸入：COS

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 (Src)	×	○	×	×	○	○	×	○	○
輸出資料 (Dest)	×	○*	×	×	○*	○*	×	○	×

\* C、# 暫存器除外

### ◆ 整數型時

輸入資料的單位為「度」(1 = 0.01)。

因此，-327.78 度 ~ 327.67 度範圍內可進行 COS 函數運算。

COS 函數輸出時將乘以 10000，因此資料將在 -10000 ~ 10000 的範圍內輸出。

### ◆ 實數型時

輸入資料的單位為「度」。

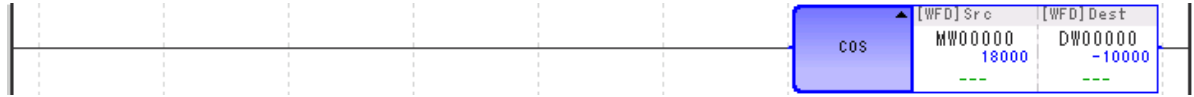
## 程式範例

整數型與實數型的輸入資料，其使用 COS 指令的程式範例如以下所示：

- 整數型 COS

計算整數型輸入資料 (MW00000) 18000 的餘弦，其結果儲存為輸出資料 (DW00000)。

$\text{COS}(180.00 \text{ 度}) \times 10000 \rightarrow \text{DW00000} = -10000$



- 實數型 COS

計算實數型輸入資料 (MF00000) 180.0 的餘弦，其結果儲存為輸出資料 (DF00000)。

$\text{COS}(180.0 \text{ 度}) \rightarrow \text{DF00000} = -1.0$



## 正切 (TAN)

計算整數型或實數型輸入資料的正切，儲存輸出資料。

TAN ( 輸入資料 ) → 輸出資料

(註)輸入資料的單位為「度」。

### 格式

格式如下所示。



圖示：**tan**

鍵輸入：TAN

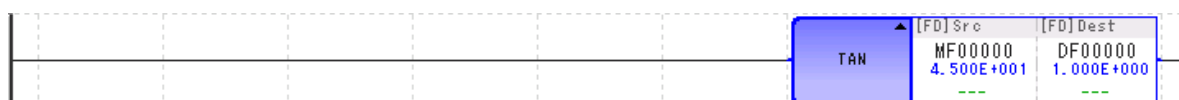
輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 (Src)	×	×	×	×	○	○	×	×	○
輸出資料 (Dest)	×	×	×	×	○*	○*	×	×	×

\* C、# 暫存器除外

### 程式範例

計算輸入資料 (MW00000) 45 的正切，儲存至輸出資料 (DF00000) 的程式範例，如下所示。

TAN (45.0 度) → DF00000 = 1.0



## 反正弦 (ASIN)

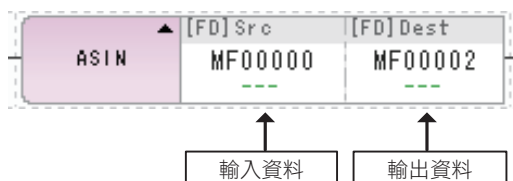
計算整數型或實數型輸入資料的反正弦，儲存輸出資料。


$$\text{SIN} \left( \boxed{\text{輸入資料}} \right)^{-1} \longrightarrow \boxed{\text{輸出資料}}$$

(註)輸出資料的單位為「度」。

### 格式

格式如下所示。



圖示：  
鍵輸入：ASIN

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 (Src)	×	×	×	×	○	○	×	×	○
輸出資料 (Dest)	×	×	×	×	○*	○*	×	×	×

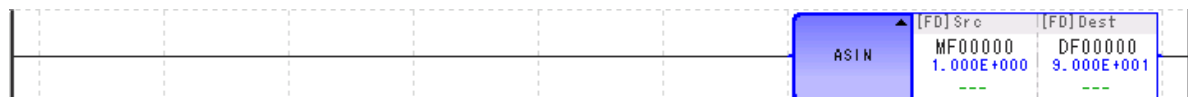
\* C、# 暫存器除外

輸入資料請設定在  $-1.0 \sim 1.0$  的範圍內。若設定超出範圍，則輸出 0。

### 程式範例

計算輸入資料 (MF00000) 1.0 的反正弦，儲存至輸出資料 (DF00000) 的程式範例，如下所示。

$$\text{SIN}(1.0)^{-1} \rightarrow \text{DF00000} = 90.0 \text{ (度)}$$



## 反餘弦 (ACOS)

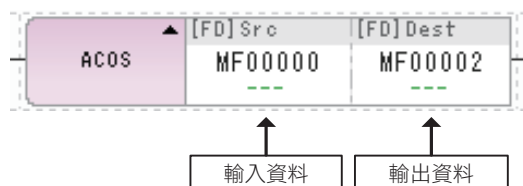
計算整數型或實數型輸入資料的反餘弦，儲存輸出資料。

$$\text{COS} \left( \boxed{\text{輸入資料}} \right)^{-1} \longrightarrow \boxed{\text{輸出資料}}$$

(註)輸出資料的單位為「度」。

### 格式

格式如下所示。



圖示：

鍵輸入：ACOS

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 (Src)	×	×	×	×	○	○	×	×	○
輸出資料 (Dest)	×	×	×	×	○*	○*	×	×	×

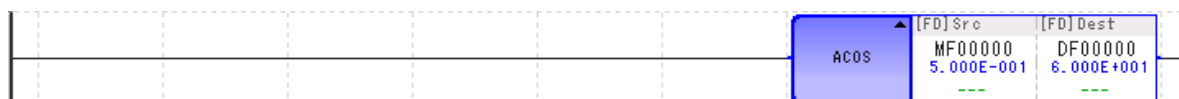
\* C、# 暫存器除外

輸入資料請設定在  $-1.0 \sim 1.0$  的範圍內。若設定超出範圍，則輸出 0。

### 程式範例

計算輸入資料 (MF00000) 0.5 的反餘弦，儲存至輸出資料 (DF00000) 的程式範例，如下所示。

$$\text{COS}(0.5)^{-1} \rightarrow \text{DF00000} = 60.0 \text{ (度)}$$



## 反正切 (ATAN)

計算整數型或實數型輸入資料的反正切，儲存輸出資料。

$$\text{TAN} \left( \boxed{\text{輸入資料}} \right)^{-1} \longrightarrow \boxed{\text{輸出資料}}$$

(註)輸出資料的單位為「度」。

### 格式

格式如下所示。



圖示：  
 鍵輸入：ATAN

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 (Src)	×	○	×	×	○	○	×	×	○
輸出資料 (Dest)	×	○	×	×	○*	○*	×	×	×

\* C、# 暫存器除外

### 程式範例

計算輸入資料 (MF00000) 1.0 的反正切，儲存至輸出資料 (DF00000) 的程式範例，如下所示。

$$\text{TAN}(1.0)^{-1} \rightarrow \text{DF00000} = 45.0 \text{ (度)}$$





## 指數 (EXP)

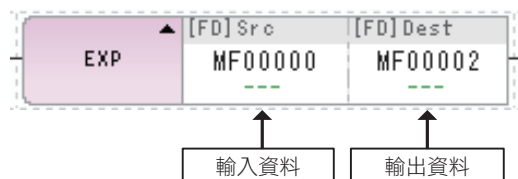
計算自然對數底 (e) 只乘以實數型輸入資料後的值，儲存輸出資料。



(註) e 為自然對數底

### 格式

格式如下所示。



圖示： $X^Y$   
鍵輸入：EXP

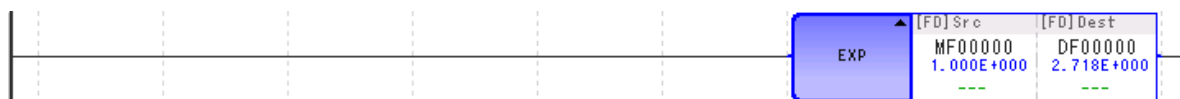
輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 (Src)	×	×	×	×	○	○	×	×	○
輸出資料 (Dest)	×	×	×	×	○*	○*	×	×	×

\* C、# 暫存器除外

### 程式範例

當自然對數底 (e) 輸入資料 (MF00000) 為 1.0，將計算結果儲存至輸出資料 (DF00000) 的程式範例，如下所示。

$$e^{1.0} \rightarrow DF00000 = 2.718282$$



**補充**

當計算結果溢位，輸出極大值 3.402E+38，但不發生計算錯誤。

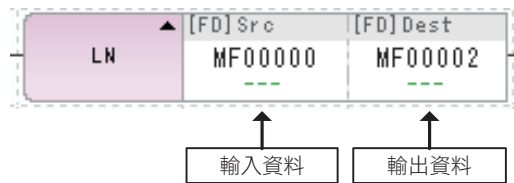
## 自然對數 (LN)

當實數型輸入資料為 X，計算其自然對數 ( $\log_e X$ ) 後，儲存至輸出資料。



### 格式

格式如下所示。



圖示： ln  
鍵輸入：LN

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 (Src)	×	×	×	×	○	○	×	×	○
輸出資料 (Dest)	×	×	×	×	○*	○*	×	×	×

\* C、# 暫存器除外

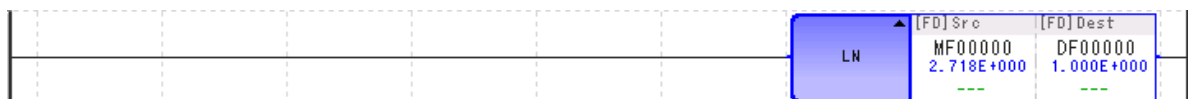
輸入資料 < 0 時，使用輸入資料的絕對值進行運算，並輸出該結果。

輸出資料 = 0 時，輸出 - ∞。

### 程式範例

當輸入資料 (MF00000) 為 2.718282 ( $\approx e$ )，將該自然對數的計算結果儲存至輸出資料 (DF00000) 的程式範例，如下所示。

$$\text{Log}_e 2.718282 \approx \log_e e \rightarrow \text{DF00000} = 1.0$$



## 常用對數 (LOG)

當實數型輸入資料為 X，計算其常用對數 ( $\log_{10} X$ ) 後，儲存至輸出資料。



### 格式

格式如下所示。



圖示： log  
鍵輸入：LOG

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入資料 (Src)	×	×	×	×	○	○	×	×	○
輸出資料 (Dest)	×	×	×	×	○*	○*	×	×	×

\* C、# 暫存器除外

輸入資料 < 0 時，使用輸入資料的絕對值進行運算，並輸出該結果。

輸出資料 = 0 時，輸出  $-\infty$ 。

### 程式範例

當輸入資料 (MF00000) 為 10.0，將該自然對數的計算結果儲存至輸出資料 (DF00000) 的程式範例，如下所示。

$$\log_{10} 10.0 \rightarrow DF00000 = 1.0$$

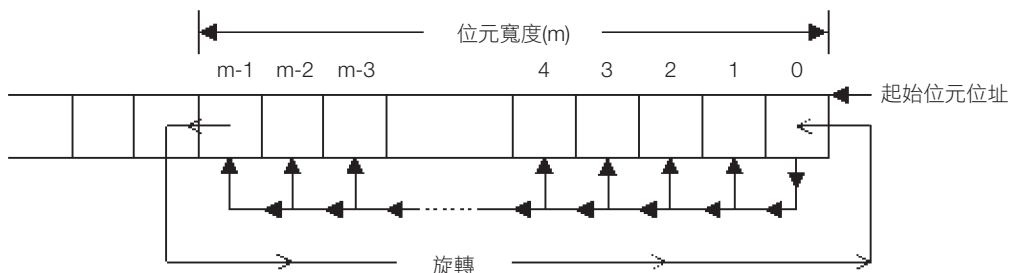


## 4.7

## 資料移動指令

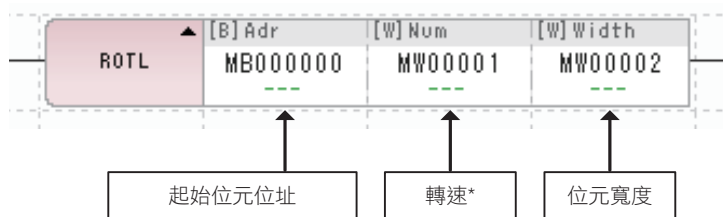
## 位元向左旋轉 (ROTL)

使開頭位元位址與位元寬度指定的資料，向左旋轉指定位元數。



## 格式

格式如下所示。



圖示：  
ROT  
L  
鍵輸入：ROTL

\* 使其旋轉的位元數

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
起始位元位址 (Adr)	○*	×	×	×	×	×	×	×	×
轉速 (Num)	×	○	×	×	×	×	×	○	○
位元寬度 (Width)	×	○	×	×	×	×	×	○	○

\* C、# 暫存器除外

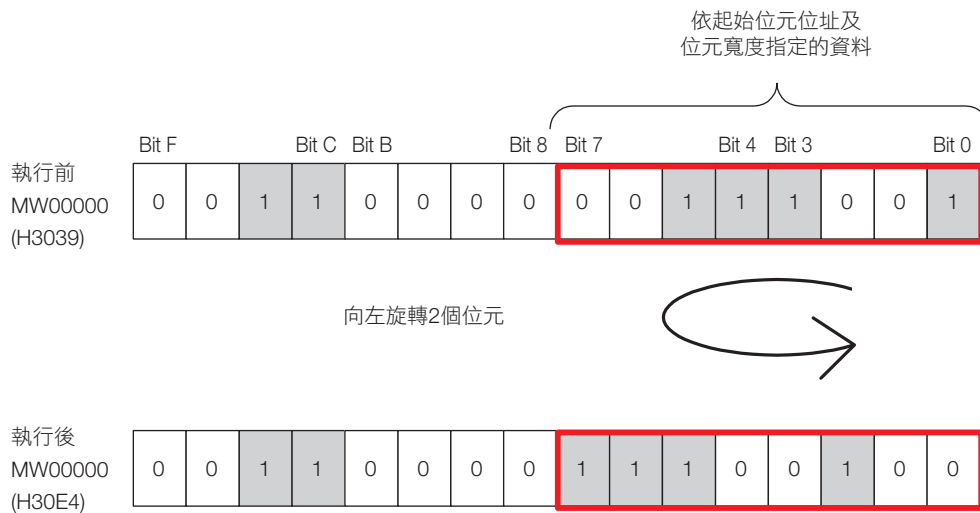
## 程式範例

使開頭位元位址 (MB000000) 與位元寬度 (8) 指定的資料向左旋轉 2 位元的程式範例，如下所示。

當開關 1 (DB000000) 為 ON 時，執行 ROTL 指令。

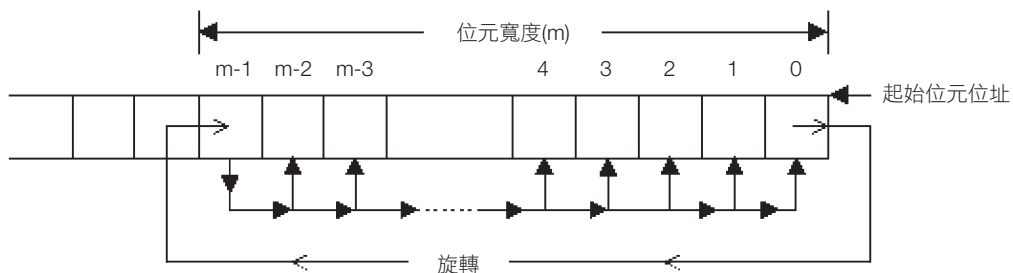


MW00000 為 12345 (H3039) 時的動作如下所示。



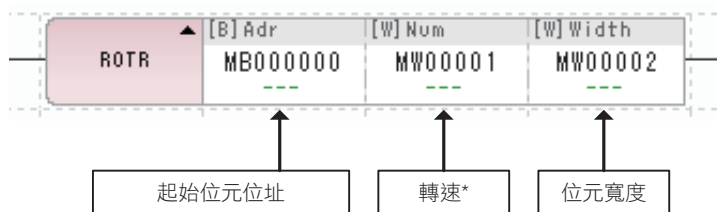
## 位元向右旋轉 (ROTR)

使開頭位元位址與位元寬度指定的資料，向右旋轉指定位元數。



### 格式

格式如下所示。



圖示：  
ROT  
R  
鍵輸入：ROTR

\* 使其旋轉的位元數

輸出輸入項目	可以使用的資料類型									
	B	W	L	Q	F	D	A	指標	常數	
起始位元位址 (Adr)	○*	×	×	×	×	×	×	×	×	
轉速 (Num)	×	○	×	×	×	×	×	○	○	
位元寬度 (Width)	×	○	×	×	×	×	×	○	○	

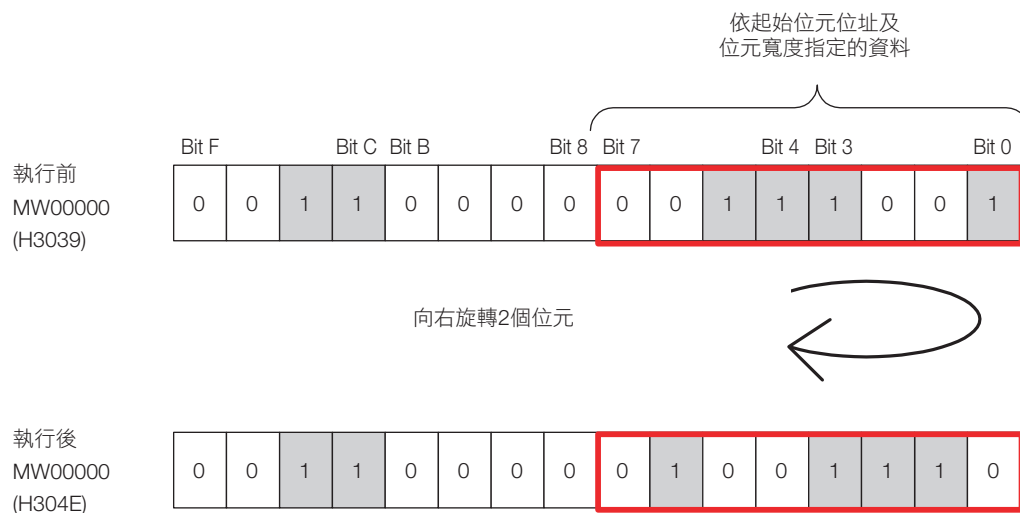
\* C、# 暫存器除外

## 程式範例

使開頭位元位址 (MB000000) 與位元寬度 (8) 指定的資料向右旋轉 2 位元的程式範例，如下所示。  
當開關 1 (DB000000) 為 ON 時，執行 ROTR 指令。

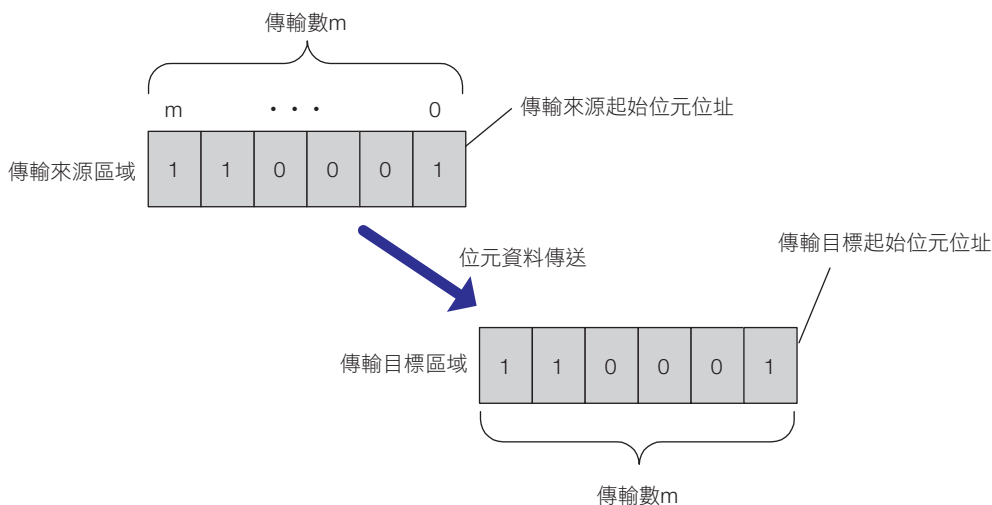


MW00000 為 12345 (H3039) 時的動作如下所示。



## 位元傳送 (MOVB)

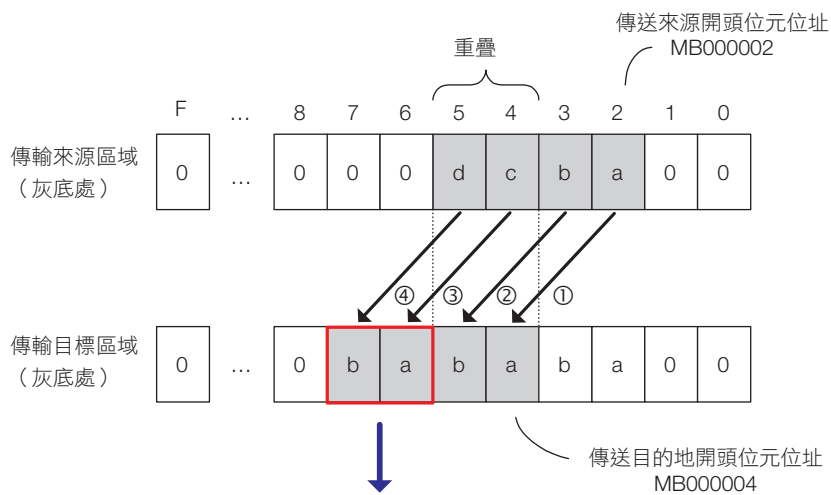
將傳送數指定區域的位元資料，由傳送來源開頭位元位址傳送至傳送對象開頭位元位址。



(註) 按照繼電器編號由小到大，以 1 個位元逐次執行傳送處理。

傳輸來源和傳輸目的地有重疊時，有時會將和執行指令當下的傳輸來源資料不同的資料，傳輸到傳輸目的地。

傳輸來源和傳輸目的地重疊的範例如下所示。

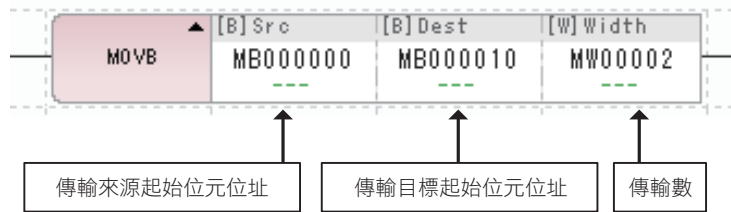


由於以①~④的順序進行傳送，①②傳送執行後的Bit 4，Bit 5資料將以③④傳送。



## 格式

格式如下所示。



圖示：**MOV**  
**B**  
鍵輸入：MOVB

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
傳送來源開頭位元位址 (Src)	○	×	×	×	×	×	×	×	×
傳送對象開頭位元位址 (Dest)	○*	×	×	×	×	×	×	×	×
傳輸數	×	○	×	×	×	×	×	○	○

\* C、# 暫存器除外

## 程式範例

將傳送來源開頭位元位址 (MB000010) 的 4 位元資料傳送至傳送對象開頭位元位址 (MB000020) 的程式範例，如下所示。

當開關 1 (DB000000) 為 ON 時，執行 MOVE 指令。

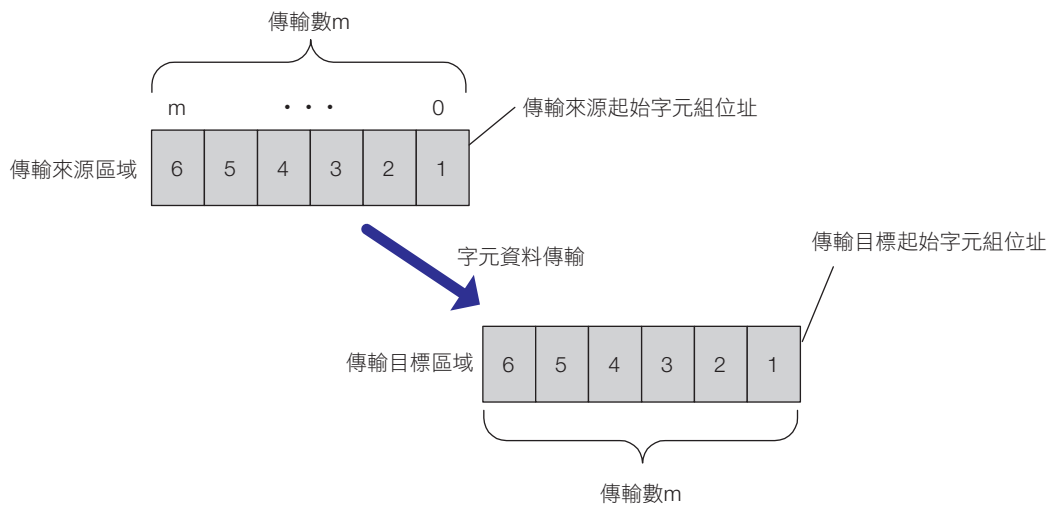


如下所示，傳輸來源的檔案傳輸到傳輸目的地。

傳輸來源區域		⇒	傳輸目標區域	
暫存器	數據		暫存器	執行指令前的資料
MB000010	0	MB000020	0	0
MB000011	1	MB000021	0	1
MB000012	1	MB000022	0	1
MB000013	1	MB000023	0	1

## 文字傳送 (MOVW)

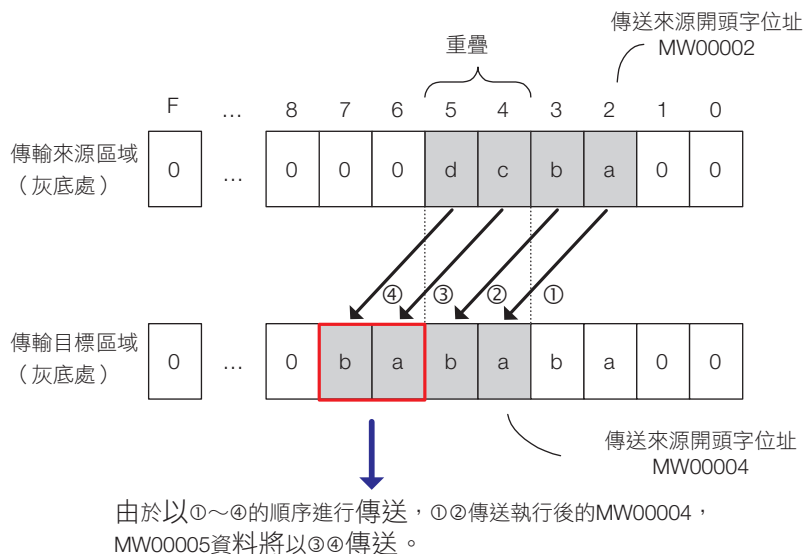
將傳送數指定區域的文字資料，由傳送來源開頭字位址傳送至傳送對象開頭字位址。



(註) 按照暫存器編號由小到大，以 1 個文字逐次執行傳送處理。

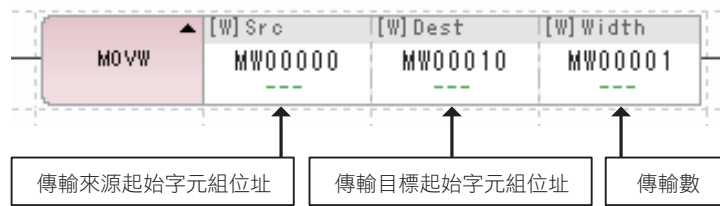
傳輸來源和傳輸目的地有重疊時，有時會將和執行指令當下的傳輸來源資料不同的資料，傳輸到傳輸目的地。

傳輸來源和傳輸目的地重疊的範例如下所示。



## 格式

格式如下所示。



圖示：  
MOV  
W  
鍵輸入：MOVW

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
傳送來源開頭字位址 (Src)	×	○	×	×	×	×	×	○	×
傳送對象開頭字位址 (Dest)	×	○*	×	×	×	×	×	○	×
傳輸次數 (Width)	×	○	×	×	×	×	×	○	○

\* C、# 暫存器除外

## 程式範例

將傳送來源開頭字位址 (MW00010) 的 4 個字資料傳送至傳送對象開頭字位址 (MW00020) 的階梯圖程式範例，如下所示。

當開關 1 (DB000000) 為 ON 時，執行 MOVW 指令。



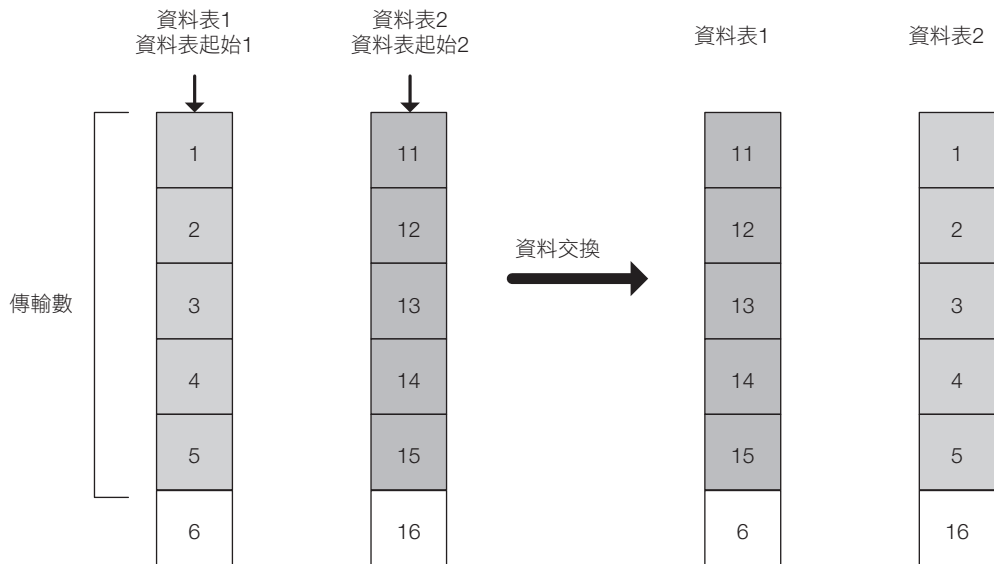
如下所示，傳輸來源的檔案傳輸到傳輸目的地。

傳輸來源區域		⇒	傳輸目標區域		
暫存器	數據		暫存器	執行指令前的資料	執行指令後的資料
MW00010	10		MW00020	0	10
MW00011	20		MW00021	0	20
MW00012	30		MW00022	0	30
MW00013	40		MW00023	0	40

## 替換傳送 (XCHG)

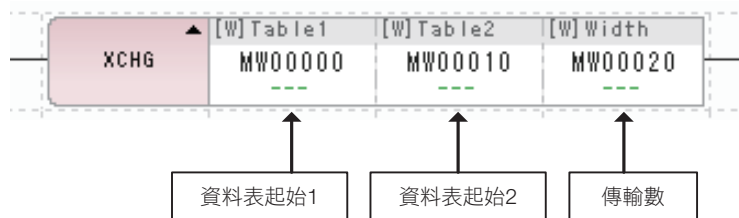
只對表 1 與表 2 間互換傳送數量的資料。

以資料表開頭 1、資料表開頭 2、傳送數量 (文字數) 指定的表 1 與表 2 資料，互換其內容。



## 格式

格式如下所示。



圖示： XCHG

鍵輸入：XCHG

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
資料表開頭 1 (Table1)	×	○*	×	×	×	×	×	×	×
資料表開頭 2 (Table2)	×	○*	×	×	×	×	×	×	×
傳輸次數 (Width)	×	○	×	×	×	×	×	○	○

\* C、# 暫存器除外

## 程式範例

表 1 (MW00010) 與表 2 (MW00020) 僅替換 4 個字資料時的程式範例，如下所示。  
當開關 1 (DB000000) 為 ON 時，執行 XCHG 指令。

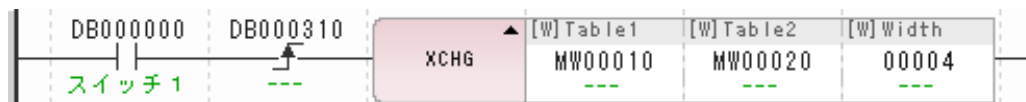
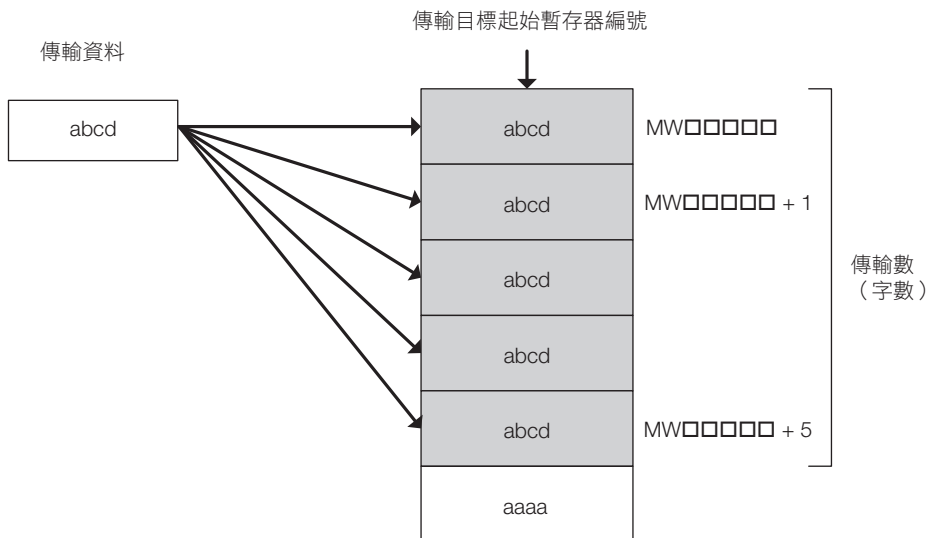


表 1 與表 2 資料將如下般替換。

資料表 1				資料表 2		
暫存器	執行指令前的資料	執行指令後的資料		暫存器	執行指令前的資料	執行指令後的資料
MW00010	10	123	⇔	MW00020	123	10
MW00011	20	234		MW00021	234	20
MW00012	30	345		MW00022	345	30
MW00013	40	456		MW00023	456	40

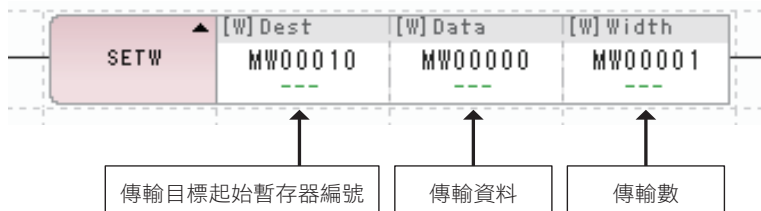
## 表初始化 (SETW)

在傳送目標開頭暫存器編號與傳送數量所指定的全部暫存器中儲存傳送資料。按照暫存器編號由小到大，以 1 個文字逐次儲存。



### 格式

格式如下所示。



圖示：**SETW**  
鍵輸入：SETW

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
傳送目標開頭暫存器編號 (Dest)	×	○*	×	×	×	×	×	×	×
傳送資料 (Data)	×	○	×	×	×	×	×	○	○
傳輸次數 (Width)	×	○	×	×	×	×	×	○	○

\* C、# 暫存器除外

## 程式範例

在通電後的高速掃描第 1 次掃描，將來自 MW00000 的 1000 個字區域初始化為傳送資料 (0) 的程式範例，如下所示。

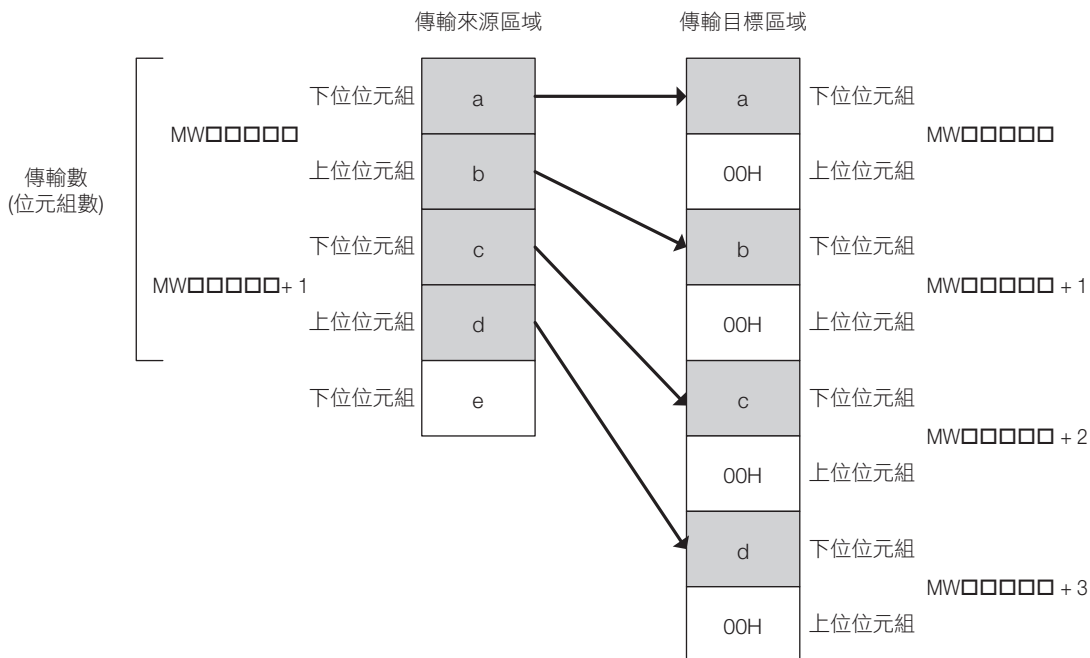


通電後執行高速掃描的 1 次掃描後，將如下初始化為 0。

暫存器	數據
MW00000	0
MW00001	0
:	:
MW00998	0
MW00999	0

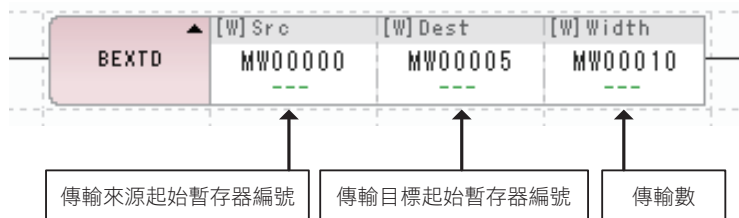
## 位元組→文字展開 (BEXTD)

將傳送來源開頭暫存器編號起算的傳送數量區域位元組資料，以 1 個位元組逐次展開為 1 字元資料，並傳送至傳送對象開頭暫存器編號起算的傳送數量區域。展開為 1 個字時，上位位元組為 0。



### 格式

格式如下所示。



圖示：  
鍵輸入：BEXTD

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
傳送來源開頭暫存器編號 (Src)	×	○	×	×	×	×	×	×	×
傳送目標開頭暫存器編號 (Dest)	×	○*	×	×	×	×	×	×	×
傳輸次數 (Width)	×	○	×	×	×	×	×	○	○

\* C、# 暫存器除外



## 程式範例

傳送傳送來源開頭暫存器編號 (MW00010) 起算 4 個位元組區域資料至傳送對象開頭暫存器編號 (MW00020) 起算 4 個位元組區域的程式範例，如下所示。

當開關 1 (DB000000) 為 ON 時，執行 BEXTD 指令。



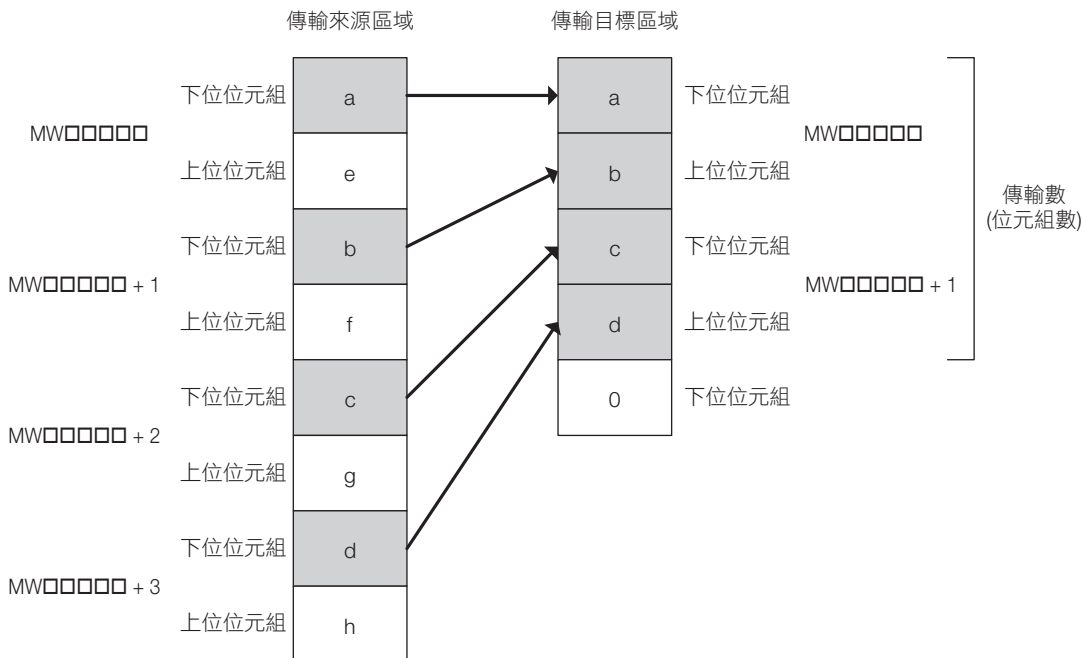
傳送來源的位元組資料，如下般對傳送對象區域展開為文字資料後傳送。

傳輸來源區域			⇒	傳輸目標區域		
暫存器		數據		暫存器		數據
MW00010	下位位元組	10 H		MW00020	下位位元組	10 H
	上位位元組	20 H			上位位元組	00 H
MW00011	下位位元組	30 H		MW00021	下位位元組	20 H
	上位位元組	40 H			上位位元組	00 H
				MW00022	下位位元組	30 H
					上位位元組	00 H
				MW00023	下位位元組	40 H
					上位位元組	00 H

## 文字→位元組壓縮 (BPRESS)

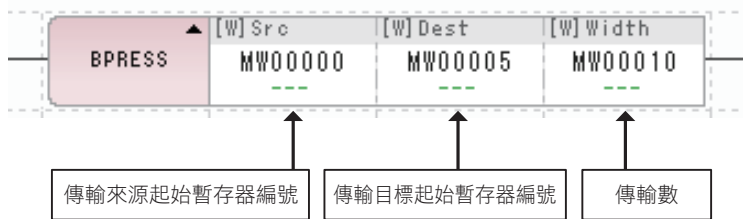
將傳送來源開頭暫存器編號起算的傳送數量區域內儲存的文字資料下位位元組，以 1 個位元組逐次儲存至傳送對象開頭暫存器編號起算的傳送數量區域。與 BEXTED 指令動作相反。

將捨棄上位位元組。



### 格式

格式如下所示。



圖示：**B**  
**PRESS**  
鍵輸入：BPRESS

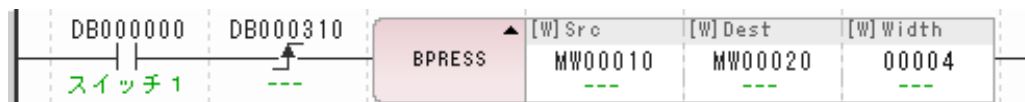
輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
傳送來源開頭暫存器編號 (Src)	×	○	×	×	×	×	×	×	×
傳送目標開頭暫存器編號 (Dest)	×	○*	×	×	×	×	×	×	×
傳輸次數 (Width)	×	○	×	×	×	×	×	○	○

\* C、# 暫存器除外

## 程式範例

儲存傳送來源開頭暫存器編號 (MW00010) 起算 4 個字元區域資料的下位位元組至傳送對象開頭暫存器編號 (MW00020) 起算 4 個位元組區域的程式範例，如下所示。

當開關 1 (DB000000) 為 ON 時，執行 BPRESS 指令。

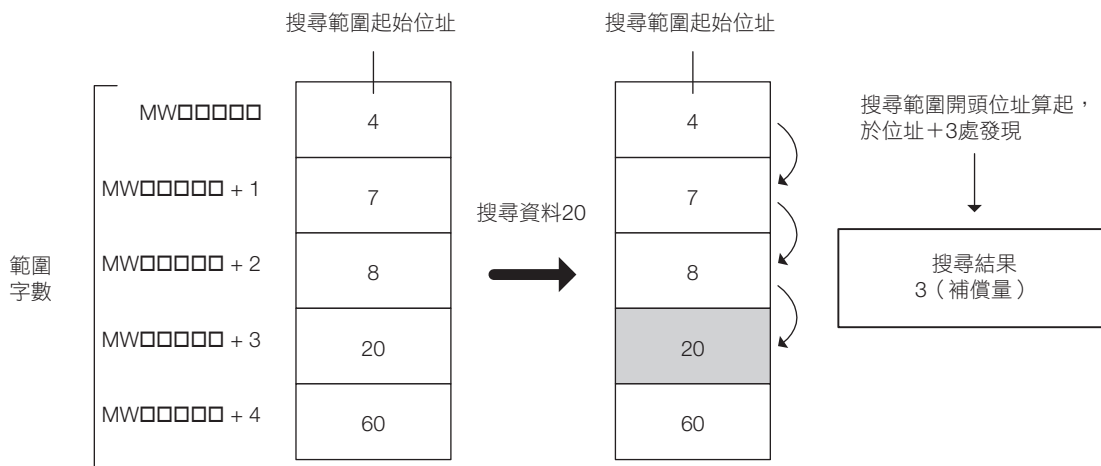


傳送來源的文字資料，如下般將傳送對象區域壓縮為位元組資料後傳送。

傳輸來源區域			⇒	傳輸目標區域		
暫存器	數據	暫存器		數據		
MW00010	下位位元組	12 H		MW00020	下位位元組	12 H
	上位位元組	23 H		MW00021	上位位元組	34 H
MW00011	下位位元組	34 H		MW00021	下位位元組	56 H
	上位位元組	45 H		MW00021	上位位元組	78 H
MW00012	下位位元組	56 H				
	上位位元組	67 H				
MW00013	下位位元組	78 H				
	上位位元組	89 H				

## 搜尋資料 (BSRCH)

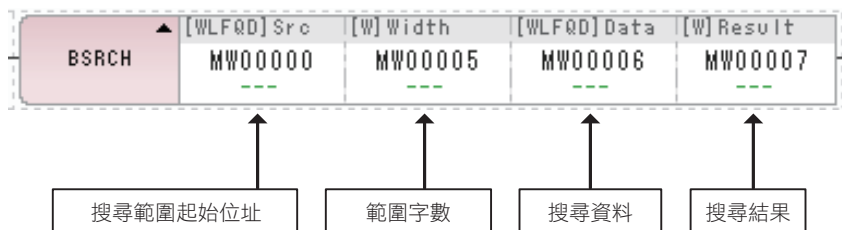
對搜尋開頭位址起算的範圍字數區域，以二進制方式進行搜尋。與搜尋資料相符的開始搜尋暫存器起算的偏移量字元組數，將輸出作為搜尋結果。搜尋範圍請務必先以遞增排序。



- (註) 1. 執行 BSRCH 指令前，請先將搜尋區域以遞增排序。  
 2. 以下為整數型的示意圖。2 倍長整數型與實數型亦同。  
 3. 搜尋後如未尋獲，搜尋結果將以 -1 儲存。

### 格式

格式如下所示。



圖示：**BSRCH**  
 鍵輸入：BSRCH

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
搜尋範圍開頭位址 (Src)	×	○	○	○	○	○	×	×	×
範圍字數 (Width)	×	○	×	×	×	×	×	○	○
搜尋資料 (Data)	×	○	○	○	○	○	×	○	○
搜尋結果 (Result)	×	○*	×	×	×	×	×	×	×

\* C、# 暫存器除外



#### 二進制搜尋

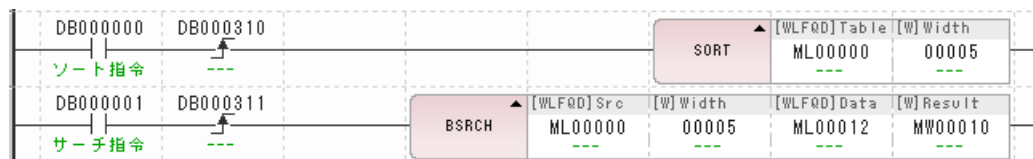
為資料搜尋演算法的一種，在已排序區域中高速進行搜尋。

將區域中央的數值與搜尋資料比對，搜尋資料為大則由中央往右區域，搜尋資料為小則由中央往左區域，持續搜尋符合者。為了能以此種步驟搜尋，需要事先將資料以遞增排序。

術語解說

## 程式範例

排序指令 (DB00000) 為 ON 時。對 ML00000 ~ ML00008 排序。之後當搜尋指令 (DB000001) ON, 在排序後的資料區域尋找搜尋資料 (ML00012)。



當 ML00000 ~ ML00008 為如下資料，搜尋資料 (ML00012) 為 70，執行第 1 行時將排序如下表。執行第 2 行，則將搜尋 70 的成果 4 儲存至搜尋結果 (MW00010)。

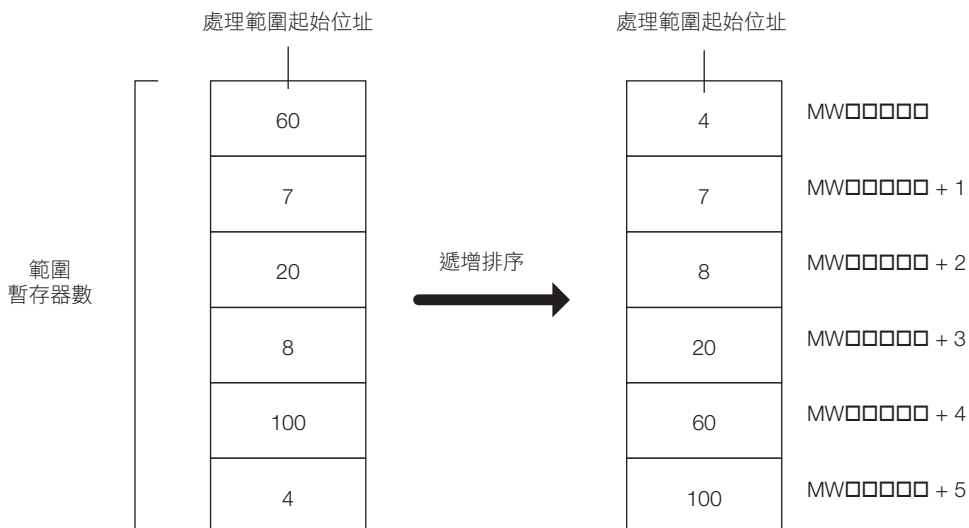
暫存器	第 1 行執行前資料	第 1 行執行後資料	第 2 行執行結果
ML00000	100	15	ML00004 = 70， 故 MW00010 = 4
ML00002	30	30	
ML00004	90	70	
ML00006	15	90	
ML00008	70	100	

## 排序 (SORT)

在排序範圍內，將開頭位址起算的範圍暫存器數量資料進行遞增排序。

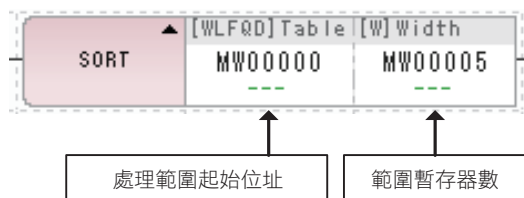
下圖將以整數型為例進行說明。2 倍長整數型、實數型也將同樣排序。

可排序的最多資料個數為 128 個。



### 格式

格式如下所示。



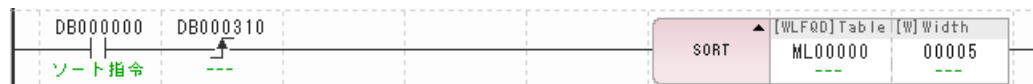
圖示： SORT  
鍵輸入： SORT

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
排序範圍開頭位址 (Table)	×	○*	○*	○*	○*	○*	×	×	×
範圍暫存器數 (Width)	×	○	×	×	×	×	×	○	○

\* C、# 暫存器除外

## 程式範例

排序指令 (DB000000) ON 時，對 ML00000 ~ ML00008 遞增排序的程式範例，如下所示。



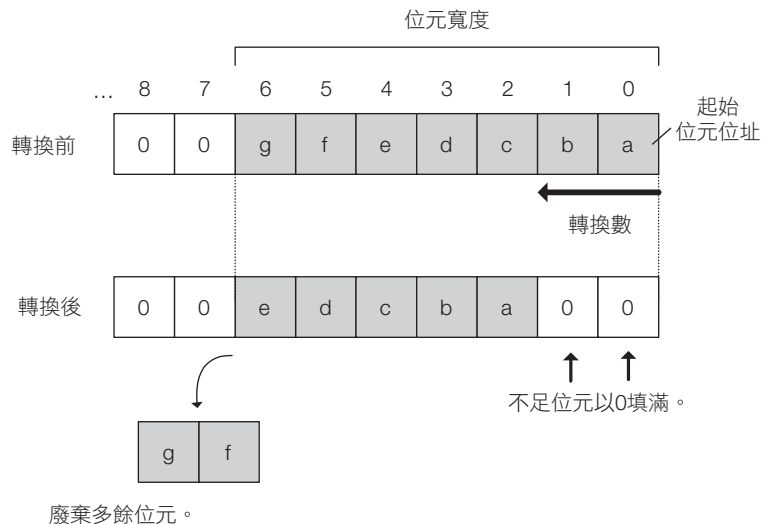
當 ML00000 ~ ML00008 為如下資料，執行 SORT 指令後將排序如下表。

暫存器	執行指令前的資料	執行指令後的資料
ML00000	100	15
ML00002	30	30
ML00004	90	70
ML00006	15	90
ML00008	70	100

## 位元向左位移 (SHFTL)

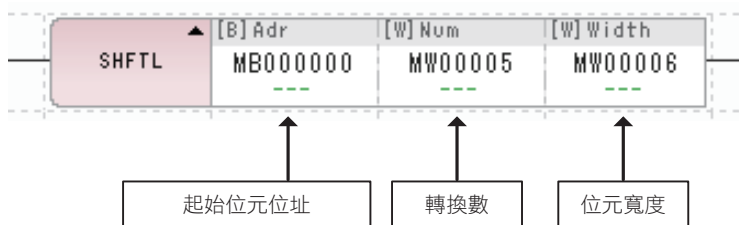
使開頭位元位址與位元寬度指定的資料，向左位移指定位元數。

超出位元寬度的資料將被捨棄，不足的部分則位元為 0。



### 格式

格式如下所示。



圖示：  
SHFTL  
鍵輸入：SHFTL

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
起始位元位址 (Adr)	○*	×	×	×	×	×	×	×	×
移位數 (Num)	×	○	×	×	×	×	×	○	○
位元寬度 (Width)	×	○	×	×	×	×	×	○	○

\* C、# 暫存器除外

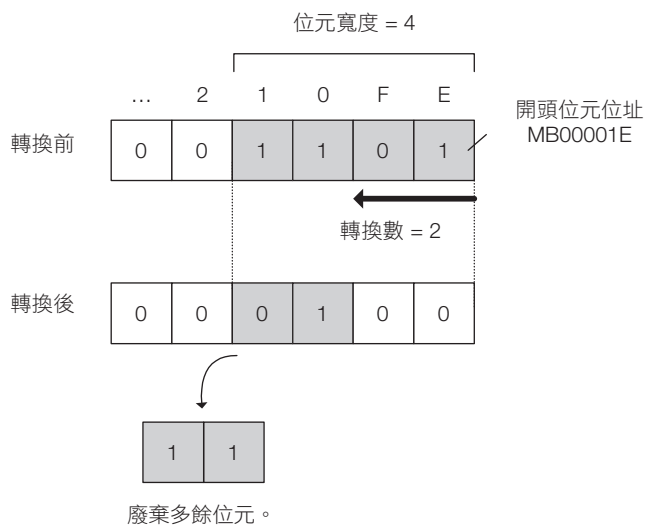


## 程式範例

開關 1 (DB000000) ON 時，開頭位元位址 (MB00001E) 4 個位元向左位移 2 位元的程式範例，如下所示。



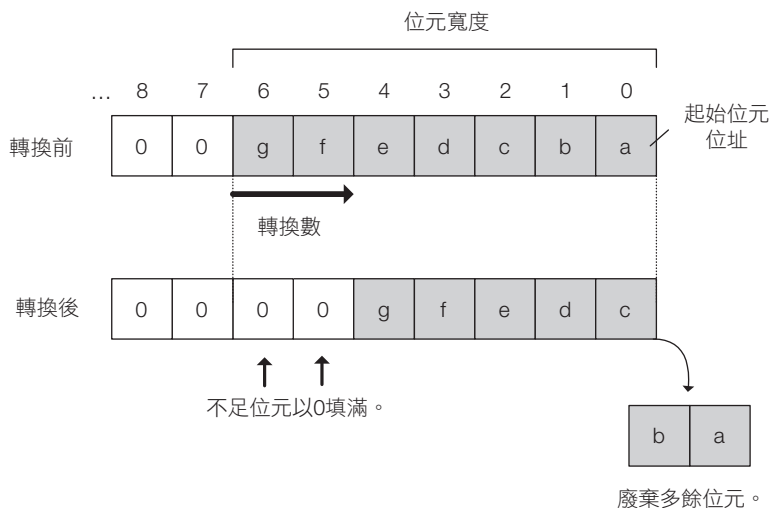
執行上列程式後，即變為下圖情形。



## 位元向右位移 (SHFTR)

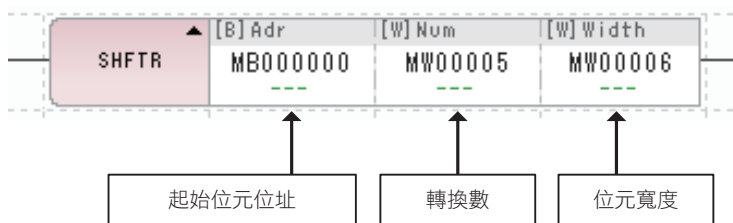
使開頭位元位址與位元寬度指定的資料，向右位移指定位元數。

超出位元寬度的資料將被捨棄，不足的部分則位元為 0。



### 格式

格式如下所示。



圖示：  
SHFT  
R  
鍵輸入：SHFTR

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
起始位元位址 (Adr)	○*	×	×	×	×	×	×	×	×
移位數 (Num)	×	○	×	×	×	×	×	○	○
位元寬度 (Width)	×	○	×	×	×	×	×	○	○

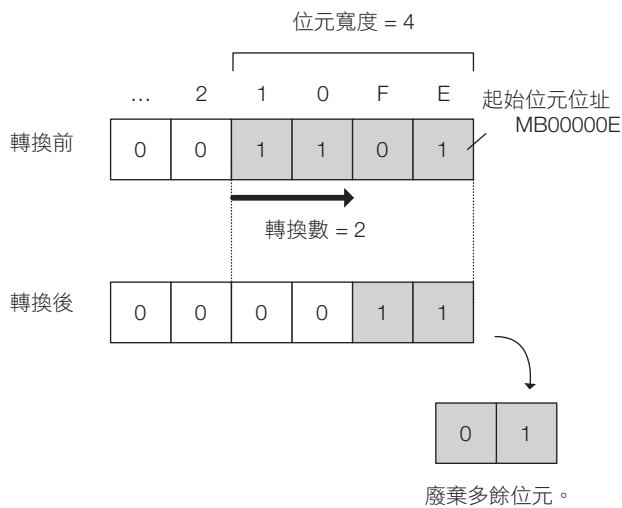
\* C、# 暫存器除外

## 程式範例

開關 1 (DB000000) ON 時，開頭位元位址 (MB00001E) 4 個位元向右位移 2 位元的程式範例，如下所示。



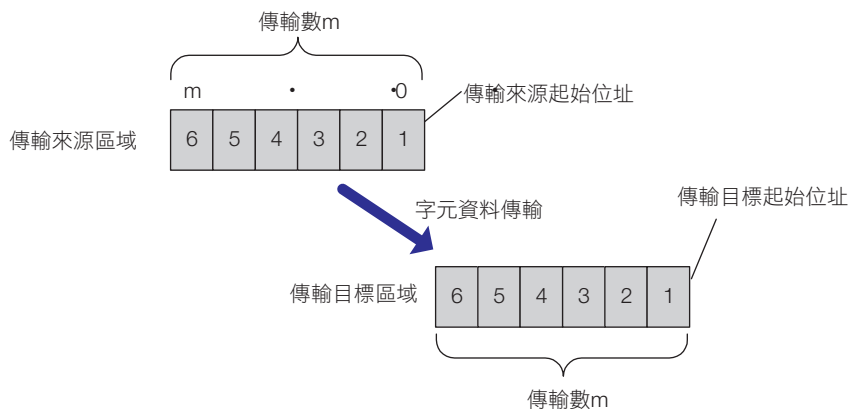
執行上列程式後，即變為下圖情形。



## 文字複製 (COPYW)

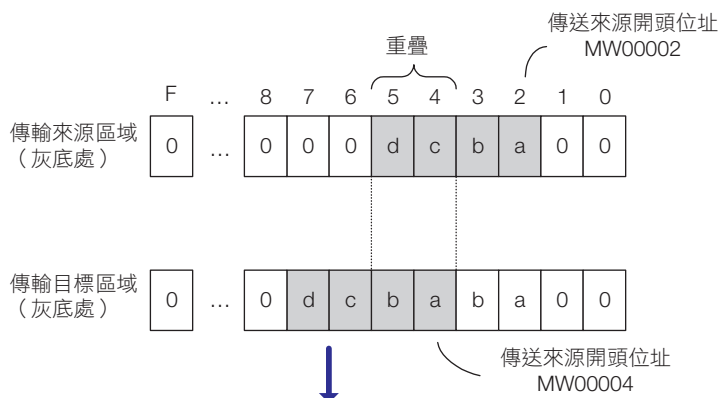
將傳送數區域文字資料，由傳送來源區域傳送至傳送對象區域。

資料以區塊方式由傳送來源逐一複製至傳送對象。因此，與文字傳送指令 (MOVW) 不同，即使傳送來源與對象重覆，也能將欲傳送的資料區塊照樣複製至傳送對象。



(註) 傳送來源與對象重覆時，動作與 MOVW 指令時不同。

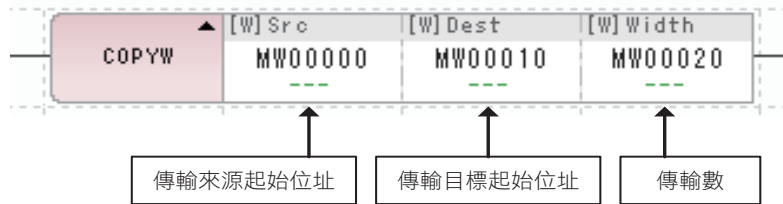
傳輸來源和傳輸目的地重疊的範例如下所示。



不同於MOVW指令，即使發生重覆，傳送來源區域的資料也將傳至傳送對象區域。

## 格式

格式如下所示。



圖示：  
  
 鍵輸入：COPYW

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
傳送來源開頭位址 (Src)	×	○	×	×	×	×	×	○	×
傳送對象開頭位址 (Dest)	×	○*	×	×	×	×	×	○	×
傳輸次數 (Width)	×	○	×	×	×	×	×	○	○

\* C、# 暫存器除外

## 程式範例

開關 1 (DB000000) ON 時，傳送來源開頭位址 (MW00000) 起算 5 個字的資料將複製至傳送對象開頭位址 (MW00100) 起算 5 個字元區域的程式範例，如下所示。

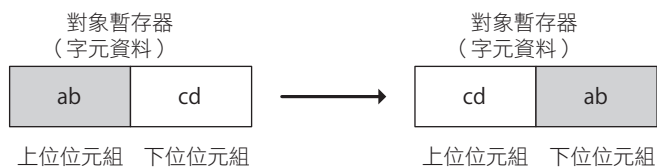


執行上列程式後，即變為下圖情形。

暫存器	數據	暫存器	執行指令前的資料	執行指令後的資料
MW00000	1	MW00100	123	1
MW00001	2	MW00101	234	2
MW00002	3	MW00102	345	3
MW00003	4	MW00103	456	4
MW00004	5	MW00104	567	5

## 位元組交換 (BSWAP)

將對象暫存器的上位位元組與下位位元組互換。



### 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								指標	常數
	B	W	L	Q	F	D	A			
對象暫存器 (Dest)	×	○*	×	×	×	×	×	×	×	×

\* C、# 暫存器除外

### 程式範例

開關 1 (DB000000) ON 時，對象暫存器 (MW00000) 的上位位元組與下位位元組互換的程式範例，如下所示。

當 MW00000 為 00FFH，執行 BSWAP 指令後，MW00000 變為 FF00H。



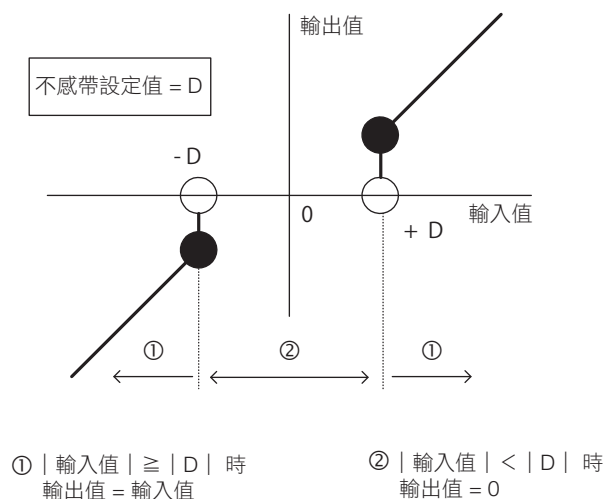
## 4.8 DDC 指令

### 不感帶 A (DZA)

針對輸入值，依據所設定的不感帶進行輸出值運算。

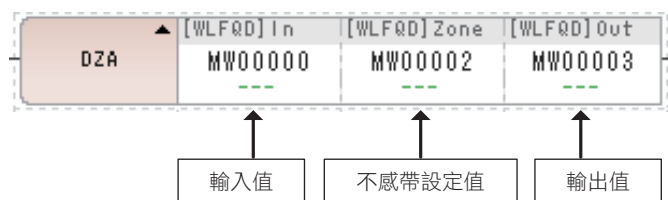
如下圖所示，當  $| \text{輸入值} | \geq | D |$ ，位於不感帶外，輸出輸入值。


當  $| \text{輸入值} | < | D |$ ，則位於不感帶內，輸出 0。



### 格式

格式如下所示。



圖示：  
鍵輸入：DZA

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入值 (In)	×	○	○	○	○	○	×	○	○
不感帶設定值 (Zone)	×	○	○	○	○	○	×	○	○
輸出值 (Out)	×	○*	○*	○*	○*	○*	×	○	×

\* C、# 暫存器除外

## 程式範例

不感帶設定值為 10000 時，將運算結果儲存於輸出值 (MW00000) 的程式範例如下所示。

對於輸入值 (MW00001 ~ MW00003) 的輸出值計算如下。

- 不感帶外時

| MW00001(12345) |  $\geq$  | 10000 | ，故 MW00000 為 12345 。



| MW00002(-12345) |  $\geq$  | 10000 | ，故 MW00000 為 -12345 。



- 不感帶內時

| MW00003(6789) | < | 10000 | ，故 MW00000 為 0 。



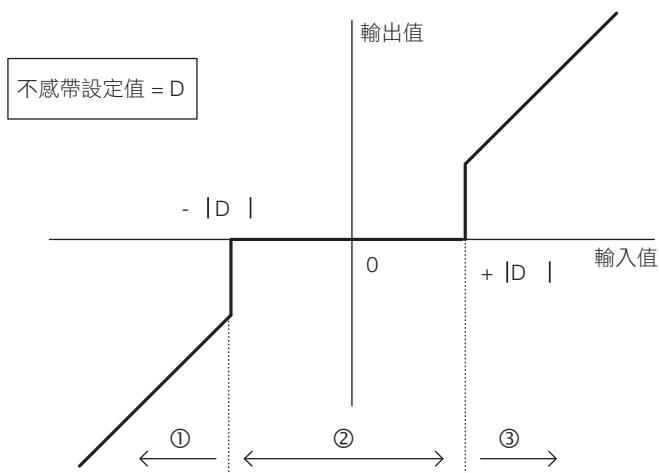


## 不感帶 B (DZB)

針對輸入值，依據所設定的不感帶進行輸出值運算。

如下圖所示，當  $| \text{輸入值} | < | D |$ ，位於不感帶內，輸出 0。

與 DZA 指令不同，輸入值為不感帶外時將根據輸入值的  $\pm$ ，輸出輸入值  $\pm | D |$  後的數值。



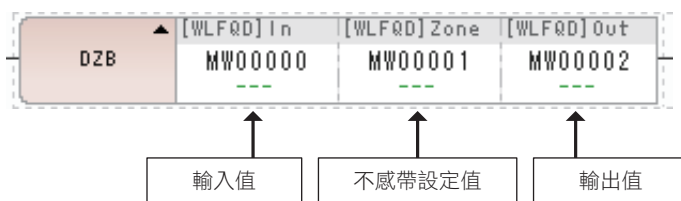
① 輸入值  $< 0$  且  $| \text{輸入值} | \geq | D |$  時  
輸出值 = 輸入值 -  $| D |$


②  $| \text{輸入值} | < | D |$  時  
輸出值 = 0

③ 輸入值  $> 0$  且  $| \text{輸入值} | \geq | D |$  時  
輸出值 = 輸入值 +  $| D |$

### 格式

格式如下所示。



圖示：  
鍵輸入：DZB

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入值 (In)	×	○	○	○	○	○	×	○	○
不感帶設定值 (Zone)	×	○	○	○	○	○	×	○	○
輸出值 (Out)	×	○ <sup>+</sup>	○ <sup>+</sup>	○ <sup>+</sup>	○ <sup>+</sup>	○ <sup>+</sup>	×	○	×

\* C、# 暫存器除外

## 程式範例

不感帶設定值為 10000 時，將運算結果儲存於輸出值 (MW00000) 的程式範例如下所示。

對於輸入值 (MW00001) 的輸出值計算如下。

- 不感帶外時

$MW00001(12345) > 0$ ， $|MW00001(12345)| \geq |10000|$ ，故  $MW00000 = 12345 - |10000| = 2345$ 。



$MW00001(-12345) < 0$ ， $|MW00001(-12345)| \geq |10000|$ ，故  $MW00000 = -12345 + |10000| = -2345$ 。



- 不感帶內時

$|MW00001(6789)| < |10000|$ ，故  $MW00000$  為 0。

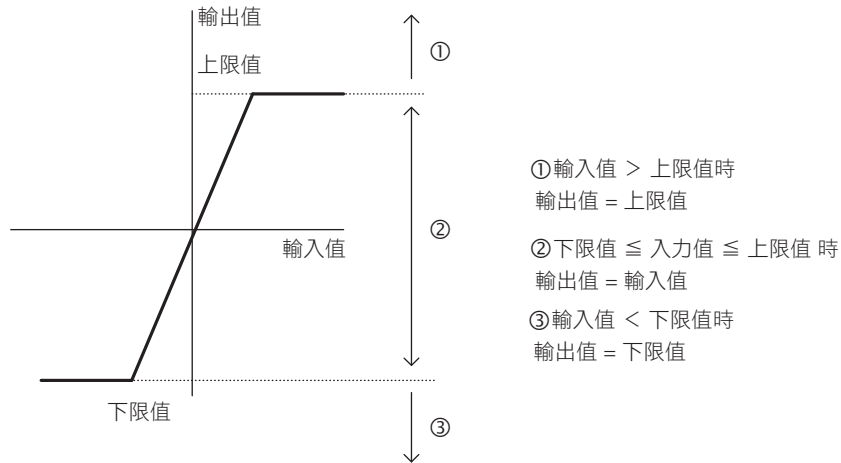


## 上下限值 (LIMIT)

對輸入值控制其輸出值，不使超過指定上下限。

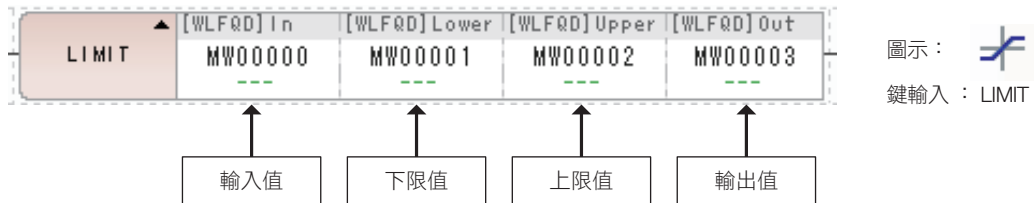
如下圖所示，輸入值在上下限以內時，輸出輸入值。

當輸入值 > 上限值時輸出上限值，輸入值 < 下限值時，輸出下限值。



### 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入值 (In)	×	○	○	○	○	○	×	○	○
下限值 (Lower)	×	○	○	○	○	○	×	○	○
上限值 (Upper)	×	○	○	○	○	○	×	○	○
輸出值 (Out)	×	○*	○*	○*	○*	○*	×	○	×

\* C、# 暫存器除外

**補充** 請務必設定為下限值 ≤ 上限值。

## 程式範例

LIMIT 指令的下限值設為 -100、上限值設為 10000 時，計算結果儲存為輸出值 (MW00000) 的程式範例，如下所示。

對於輸入值 (MW00001) 的輸出值計算如下。

- 輸入值在上下限值範圍外

MW00001(12345) > 上限值 (10000)，故 MW00000 為上限值 (10000)。



MW00001(-12345) < 下限值 (-100)，故 MW00000 為下限值 (-100)。



- 輸入值在上下限值範圍內

下限值 (-100) < MW00001(6789) < 上限值 (10000)，故 MW00000 為 6789。



## PI 控制 (PI)

輸入偏差 X，則根據已設定的參數表內容執行 PI 運算及範圍運算，輸出補償值 Y。

參數表內的積分重設為 ON 時，以 I 補償值為 0 執行 PI 補償值運算。

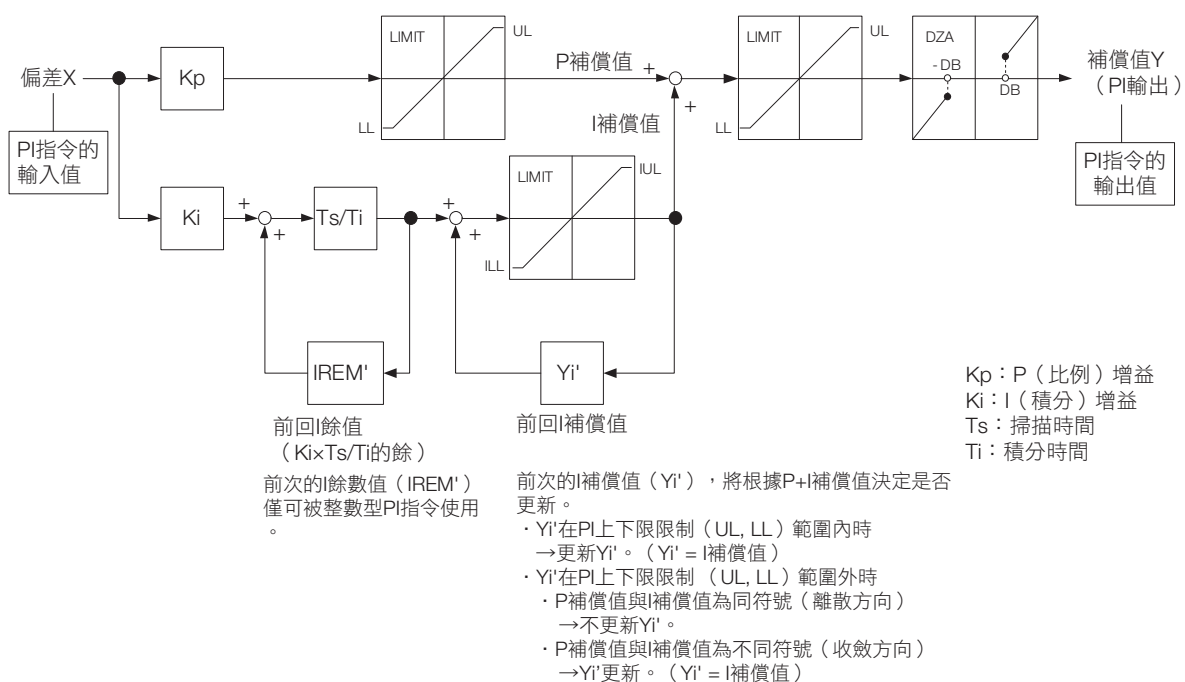
輸入至 PI 指令的值，可使用整數型與實數型。不使用 2 倍長整數型。

此外，整數型與實數型的參數表配置有所不同。



重要

以整數型使用時，請將掃描時間設定值設為 1 ms 的整數倍。

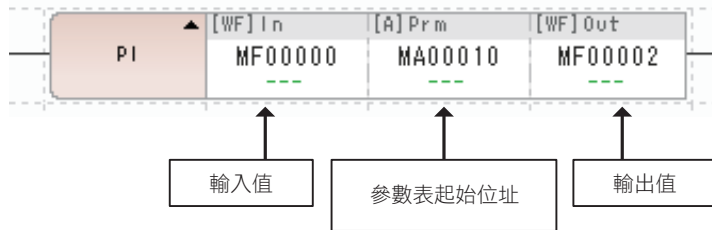


PI 指令的運算，若將輸入值設為  $X(s)$ ，輸出值設為  $Y(s)$ ，則公式如下所示。

$$\frac{Y(s)}{X(s)} = K_p + K_i \times \frac{1}{T_i \times s}$$

## 格式

格式如下所示。



圖示： PI

鍵輸入：PI

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入值 (In)	×	○	×	×	○	×	×	○	○
參數表起始位址 (Prm)	×	×	×	×	×	×	○*	○	○
輸出值 (Out)	×	○*	×	×	○*	×	×	○	×

\* C、# 暫存器除外

### ◆ 整數型 PI 指令的參數表

位址	資料類型	記號	名稱	規格	輸出輸入
0	W	RLY	繼電器輸入輸出	繼電器輸入，繼電器輸出*	IN/OUT
1	W	Kp	P 增益	P 補償增益 (增益 1 倍為 100)	IN
2	W	Ki	I 增益	積分迴路輸入增益 (增益 1 倍為 100)	IN
3	W	Ti	積分時間	積分時間 (ms)	IN
4	W	IUL	積分上限 LIMIT	I 校正值的相對上限限制值	IN
5	W	ILL	積分下限 LIMIT	I 校正值的相對下限限制值	IN
6	W	UL	PI 上限 LIMIT	P + I 校正值的相對上限限制值	IN
7	W	LL	PI 下限 LIMIT	P + I 校正值的相對下限限制值	IN
8	W	DB	PI 輸出不感帶	P + I 校正值的相對不感帶寬度	IN
9	W	Y	PI 輸出	PI 校正值輸出 (輸出至 Out)	OUT
10	W	Yi	I 補償值	I 校正值儲存	OUT
11	W	IREM	I 餘數	I 餘數儲存	OUT

\* 繼電器輸入輸出的分配如下所示。

Bit	記號	名稱	規格	輸出輸入
0	IRST	積分重設	積分重設時輸入 ON	IN
1 ~ 7	-	(備用)	輸入用備用繼電器	IN
8 ~ F	-	(備用)	輸出用備用繼電器	OUT

## ◆ 實數型 PI 指令的參數表

位址	資料類型	記號	名稱	規格	輸出輸入
0	W	RLY	繼電器輸入輸出	繼電器輸入，繼電器輸出*	IN/OUT
1	W	-	(備用)	備用暫存器	-
2	F	Kp	P 增益	P 補償增益 (增益 1 倍為 1.0)	IN
4	F	Ki	I 增益	積分迴路輸入增益 (增益 1 倍為 1.0)	IN
6	F	Ti	積分時間	積分時間 (s)	IN
8	F	IUL	積分上限 LIMIT	I 校正值的相對上限限制值	IN
10	F	ILL	積分下限 LIMIT	I 校正值的相對下限限制值	IN
12	F	UL	PI 上限 LIMIT	P + I 校正值的相對上限限制值	IN
14	F	LL	PI 下限 LIMIT	P + I 校正值的相對下限限制值	IN
16	F	DB	PI 輸出不感帶	P + I 校正值的相對不感帶寬度	IN
18	F	Y	PI 輸出	PI 校正值輸出 (輸出至 Out)	OUT
20	F	Yi	I 補償值	I 校正值儲存	OUT

\* 繼電器輸入輸出的分配和整數型相同。

## ◆ 指令內部運算

對輸入的偏差 X 進行下列運算，可求出輸出值 (PI 補償值)。

下列計算式中的 Yi' 為 Yi 前一次的 I 補償值，Ts 為掃描時間設定值。

**補充** 積分重設 (IRST) 為 ON 時，以 I 補償值為 0 執行 PI 補償值運算。

P 校正值 = (Kp × X) 的上下限限制 (UL、LL)

Yi (I 校正值) = { (Ki × X + IREM) /  $\frac{T_i}{T_s}$  + Yi' } 的上下限限制 (IUL、ILL)

Y (PI 補償值) = P 補償值 + I 補償值上下限限制 (UL、LL)，不感帶 A (寬度 DB)

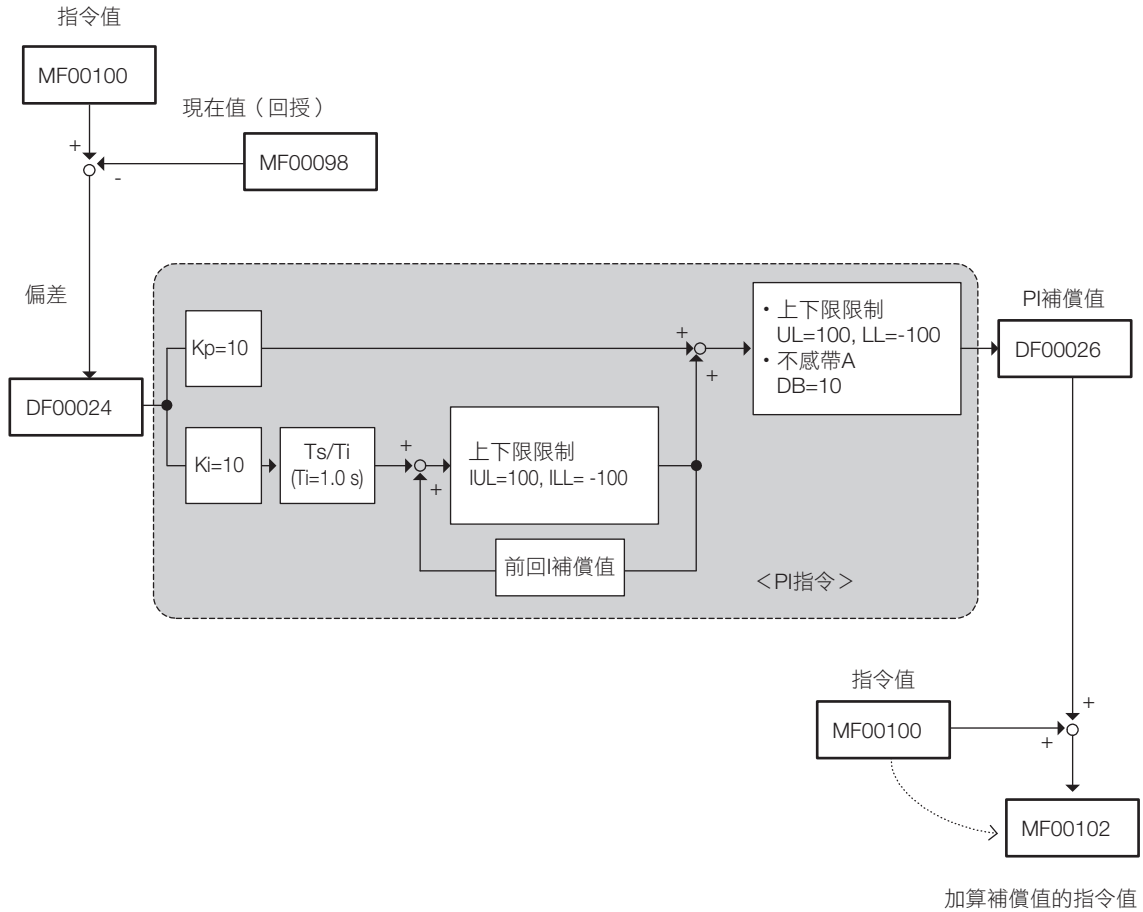
## 程式範例

考慮到 PI 補償值的指令值 (MF00102) 計算程式範例，如下所示。

求出指令值 (MF00100) 與現在值 (MF00098) 的偏差 (DF00024)，輸入至 PI 指令。

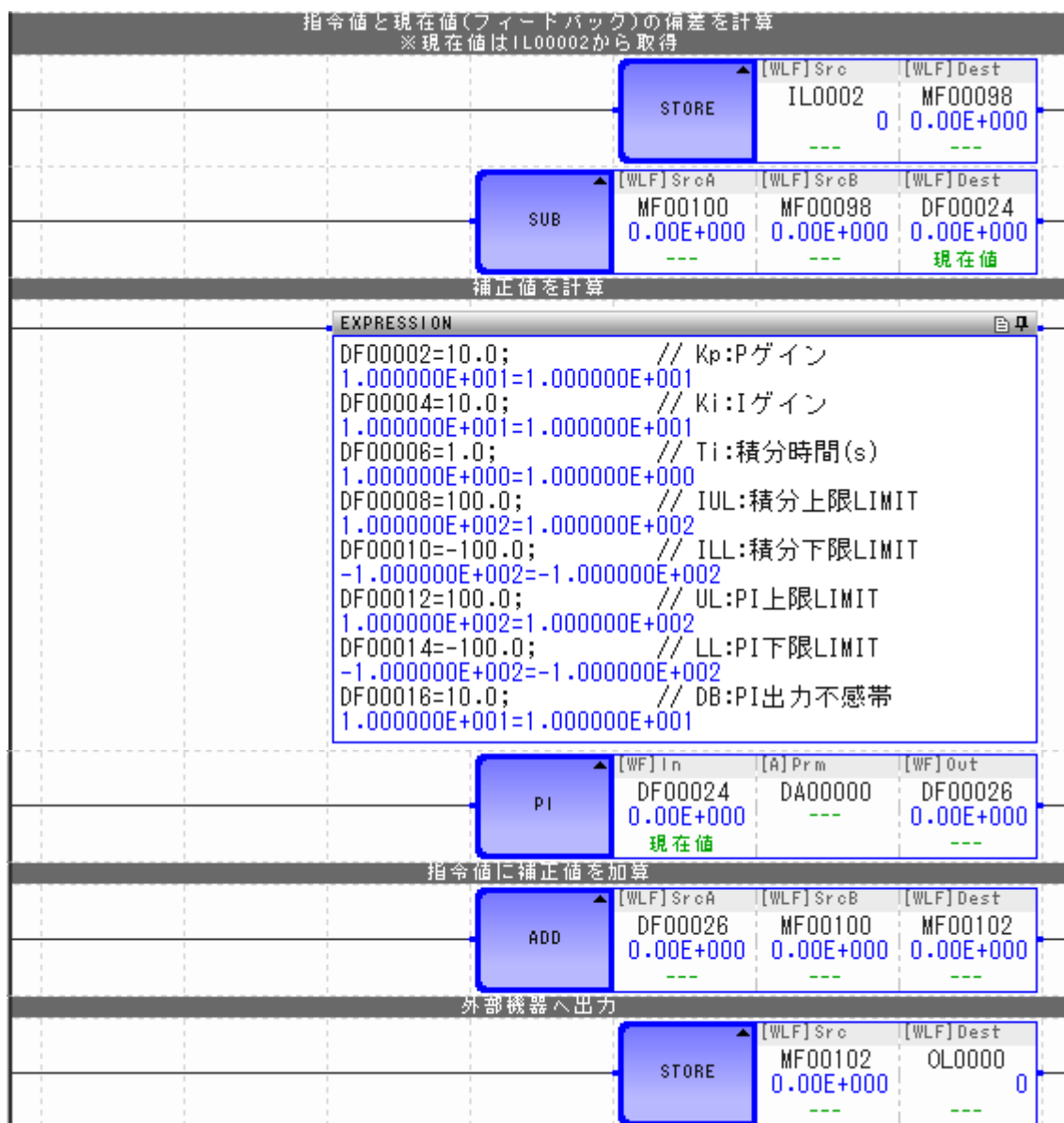
輸出的 PI 補償值 (DF00026) 加上原本的指令值 (MF00100) 後，求出本次輸出的指令值。

程式框圖如下所示。





程式範例如下所示。



(註)OL00000 (指令値) 及 IL00002 (回授値) 為分配給外部設備の暫存器。

## PD 控制 (PD)

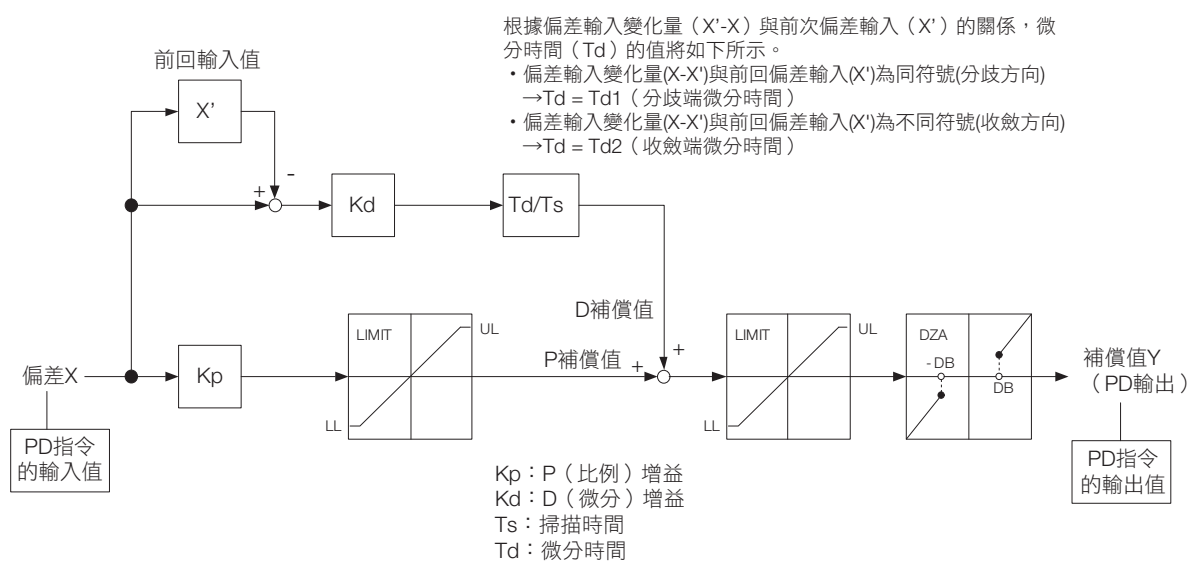
輸入偏差 X，則根據已設定的參數表內容執行 PD 運算及範圍運算，輸出補償值 Y。

輸入至 PD 指令的值，可使用整數型與實數型。不使用 2 倍長整數型。

此外，整數型與實數型的參數表配置有所不同。



以整數型使用時，請將掃描時間設定值設為 1 ms 的整數倍。

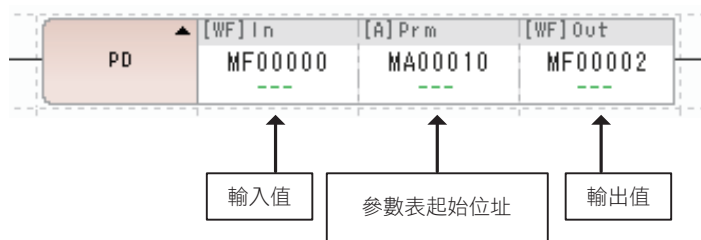


PD 指令運算如以輸入值為  $X(s)$ 、輸出值為  $Y(s)$ ，則可用下列式子表示。

$$\frac{Y(s)}{X(s)} = Kp + Kd \times Td \times S$$

## 格式

格式如下所示。



圖示： PD  
鍵輸入： PD

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入值 (In)	×	○	×	×	○	×	×	○	○
參數表起始位址 (Prm)	×	×	×	×	×	×	○*	○	○
輸出值 (Out)	×	○*	×	×	○*	×	×	○	×

\* C、# 暫存器除外

### ◆ 整數型 PD 指令的參數表

位址	資料類型	記號	名稱	規格	輸出輸入
0	W	RLY	繼電器輸入輸出	繼電器輸入，繼電器輸出*	IN/OUT
1	W	Kp	P 增益	P 補償增益 (增益 1 倍為 100)	IN
2	W	Kd	D 增益	微分迴路輸入增益 (增益 1 倍為 100)	IN
3	W	Td1	發散端微分時間	來自發散方向的輸入所使用的微分時間 (ms)	IN
4	W	Td2	收束端微分時間	來自收束方向的輸入所使用的微分時間 (ms)	IN
5	W	UL	PD 上限 LIMIT	P + D 校正值的相對上限限制值	IN
6	W	LL	PD 下限 LIMIT	P + D 校正值的相對下限限制值	IN
7	W	DB	PD 輸出不感帶	P + D 校正值的相對不感帶寬度	IN
8	W	Y	PD 輸出	PD 補償值輸出 (輸出至 Out)	OUT
9	W	X	儲存輸入值	儲存本次輸入偏差值	OUT

\* 繼電器輸入輸出的分配如下所示。

Bit	記號	名稱	規格	輸出輸入
0 ~ 7	-	(備用)	輸入用備用繼電器	IN
8 ~ F	-	(備用)	輸出用備用繼電器	OUT

## ◆ 實數型 PD 指令的參數表

位址	資料類型	記號	名稱	規格	輸出輸入
0	W	RLY	繼電器輸入輸出	繼電器輸入，繼電器輸出*	IN/OUT
1	W	-	(備用)	備用暫存器	-
2	F	Kp	P 增益	P 補償增益 (增益 1 倍為 1.0)	IN
4	F	Kd	D 增益	微分迴路輸入增益 (增益 1 倍為 1.0)	IN
6	F	Td1	發散端微分時間	來自發散方向的輸入所使用的微分時間 (s)	IN
8	F	Td2	收束端微分時間	來自收束方向的輸入所使用的微分時間 (s)	IN
10	F	UL	PD 上限 LIMIT	P + D 校正值的相對上限限制值	IN
12	F	LL	PD 下限 LIMIT	P + D 校正值的相對下限限制值	IN
14	F	DB	PD 輸出不感帶	P + D 校正值的相對不感帶寬度	IN
16	F	Y	PD 輸出	PD 補償值輸出 (輸出至 Out)	OUT
18	F	X	儲存輸入值	儲存本次輸入偏差值	OUT

\* 繼電器輸入輸出的分配和整數型相同。

## ◆ 指令內部運算

對輸入的偏差 X 進行下列運算，可求出輸出值 (PD 補償值)。

以下計算式中的 X' 為 X 的前一次輸入值，Ts 為掃描時間設定值，Td 為微分時間。

微分時間 (Td) 在 (X - X') 與 X' 符號相同時為 Td1，(X - X') 與 X' 不同符號時為 Td2。

P 補償值 = (Kp × X) 的上下限限制 (UL、LL)

D 補償值 =  $Kd \times (X - X') \times \frac{Td}{Ts}$  的上下限限制 (IUL、ILL)

PD 補償值 = (P 補償值 + D 補償值) 的上下限限制 (UL、LL)，不感帶 A (不感帶寬度 = DB)

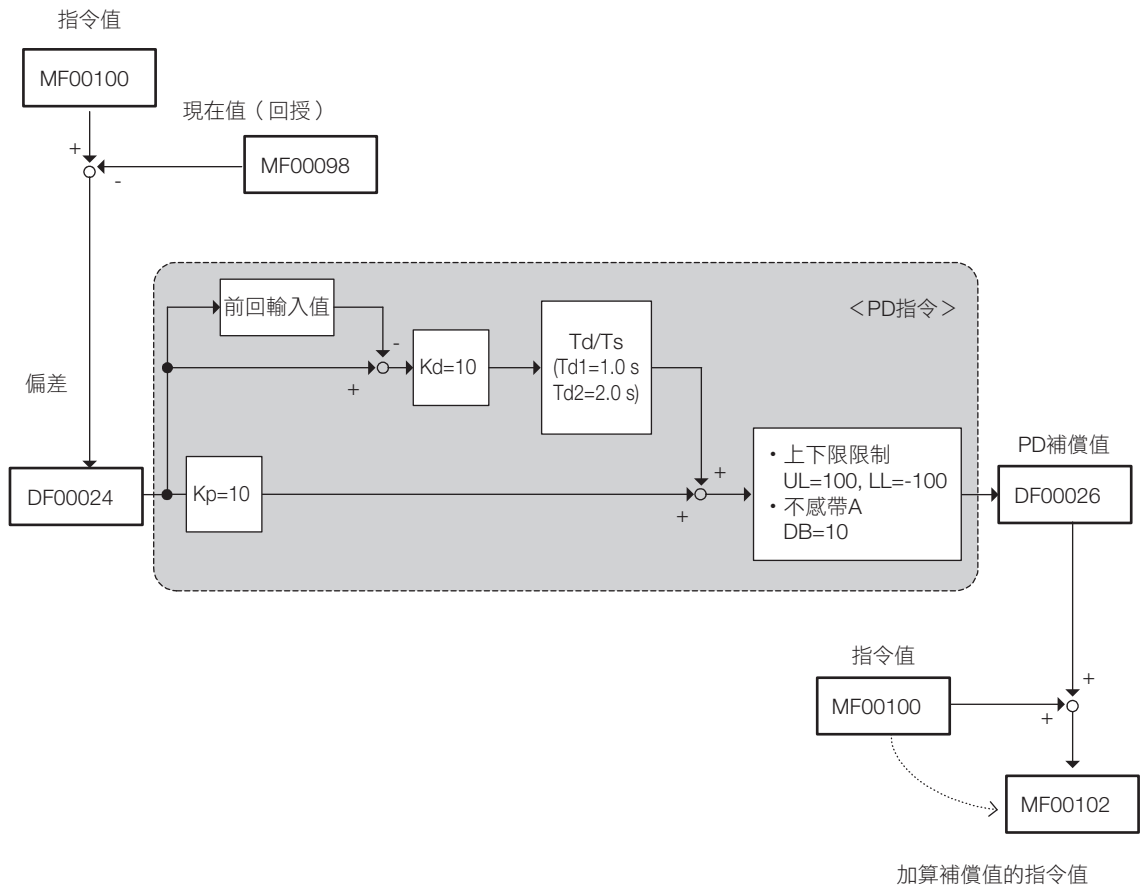
## 程式範例

考慮到 PD 補償值的指令值 (MF00102) 計算程式範例，如下所示。

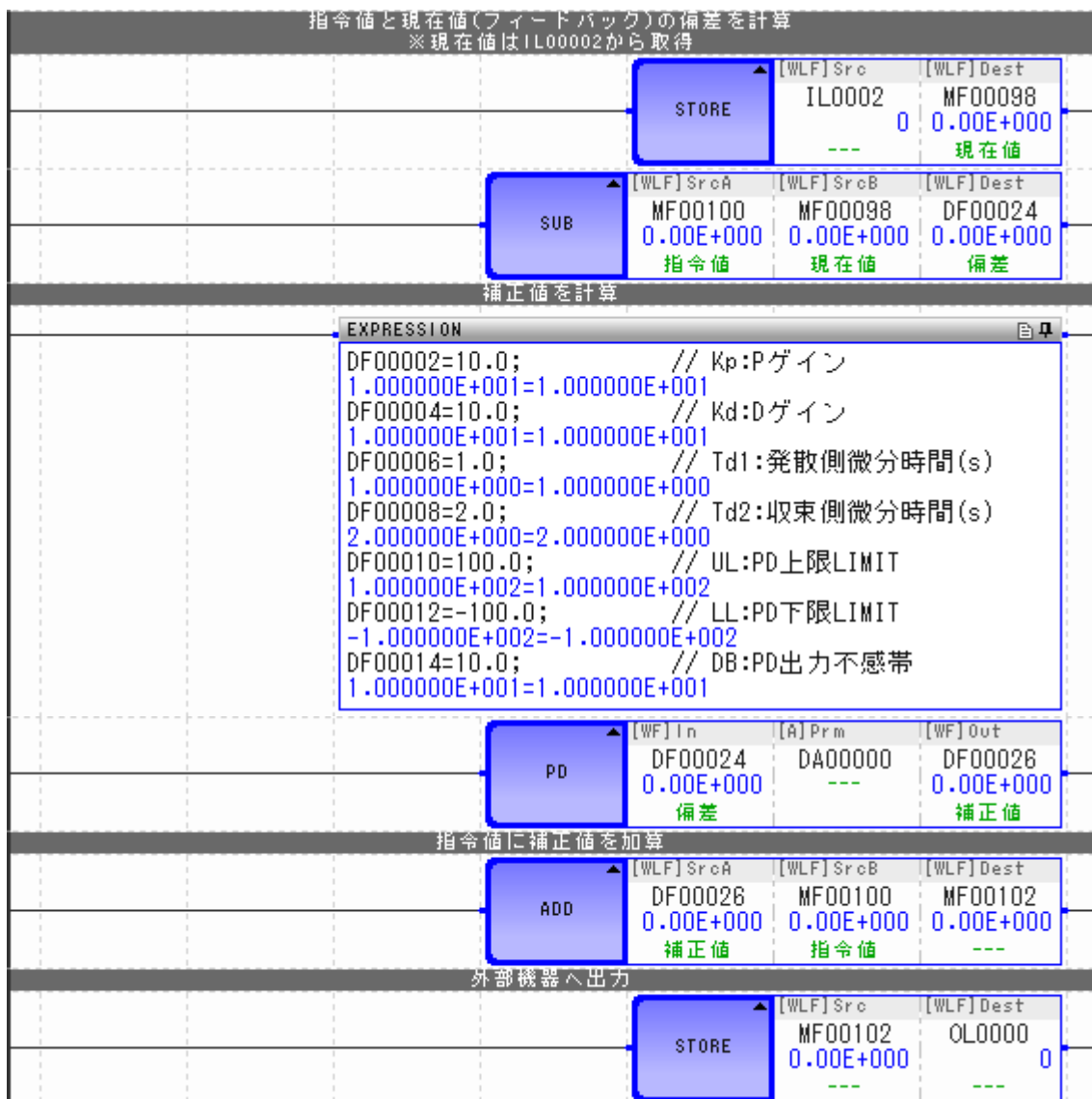
求出指令值 (MF00100) 與現在值 (MF00098) 的偏差 (DF00024)，輸入至 PD 指令。

輸出的 PD 補償值 (DF00026) 加上原本的指令值 (MF00100) 後，求出本次輸出的指令值。

程式框圖如下所示。



程式範例如下所示。



(註)OL00000 (指令値) 及 IL00002 (回授値) 為分配給外部設備の暫存器。

## 補充事項

### ◆ 傳送函數

PD 運算的傳送函數可以下列式子表示。

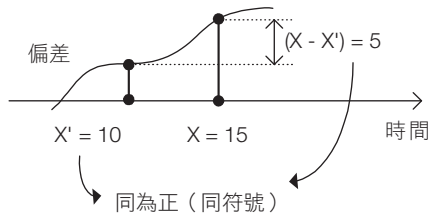
$X(s)$  為輸入值， $Y(s)$  為輸出值。

$$\frac{Y(s)}{X(s)} = K_p + K_d \times T_d \times S$$

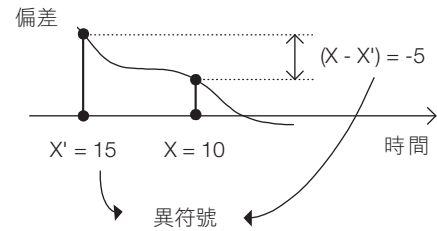
### ◆ 離散端與收束端的本次偏差 $X$ 與前次偏差 $X'$ 關係

離散端與收束端的本次偏差  $X$  與前次偏差  $X'$  關係，如下圖所示。

< 離散端 (偏差離散時) 範例 >



< 收束端 (偏差收束時) 範例 >



## PID 控制 (PID)

輸入偏差 X，則根據已設定的參數表內容執行 PID 運算及範圍運算，輸出補償值 Y。

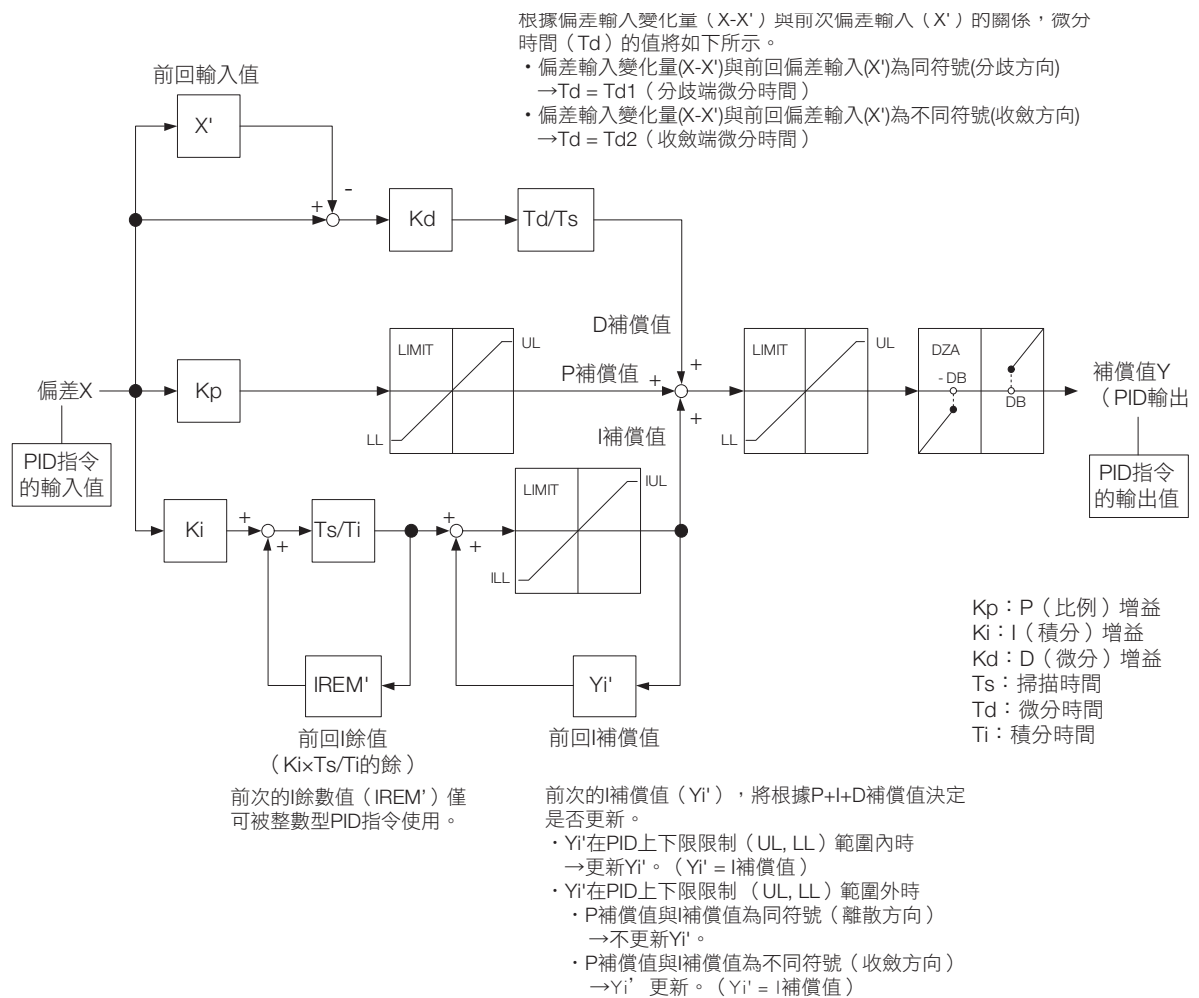
參數表內的積分重設為 ON 時，以 I 補償值 = 0 執行 PI 補償值運算。

輸入至 PID 指令的值，可使用整數型與實數型。不使用 2 倍長整數型。

此外，整數型與實數型的參數表配置有所不同。



以整數型使用時，請將掃描時間設定值設為 1 ms 的整數倍。



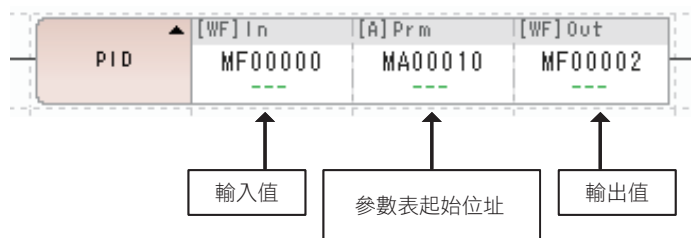
PID 指令運算如以輸入值為 X(s)、輸出值為 Y(s)，則可用下列式子表示。

$$\frac{Y(s)}{X(s)} = Kp + Ki \times \frac{1}{Ti \times s} + Kd \times Td \times S$$



## 格式

格式如下所示。



圖示： PID  
鍵輸入：PID

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入值 (In)	×	○	×	×	○	×	×	○	○
參數表起始位址 (Prm)	×	×	×	×	×	×	○*	○	○
輸出值 (Out)	×	○*	×	×	○*	×	×	○	×

\* C、# 暫存器除外

### ◆ 整數型 PID 指令的參數表

位址	資料類型	記號	名稱	規格	輸出輸入
0	W	RLY	繼電器輸入輸出	繼電器輸入，繼電器輸出*	IN/OUT
1	W	Kp	P 增益	P 補償增益 (增益 1 倍為 100)	IN
2	W	Ki	I 增益	積分迴路輸入增益 (增益 1 倍為 100)	IN
3	W	Kd	D 增益	微分迴路輸入增益 (增益 1 倍為 100)	IN
4	W	Ti	積分時間	積分時間 (ms)	IN
5	W	Td1	發散端微分時間	來自發散方向的輸入所使用的微分時間 (ms)	IN
6	W	Td2	收束端微分時間	來自收束方向的輸入所使用的微分時間 (ms)	IN
7	W	IUL	積分上限 LIMIT	I 校正值的相對上限限制值	IN
8	W	ILL	積分下限 LIMIT	I 校正值的相對下限限制值	IN
9	W	UL	PID 上限 LIMIT	P + I 校正值的相對上限限制值	IN
10	W	LL	PID 下限 LIMIT	P + I 校正值的相對下限限制值	IN
11	W	DB	PID 輸出不感帶	P + I 校正值的相對不感帶寬度	IN
12	W	Y	PID 輸出	PI 校正值輸出 (輸出至 Out)	OUT
13	W	Yi	I 補償值	I 校正值儲存	OUT
14	W	I REM	I 餘數	I 餘數儲存	OUT
15	W	X	儲存輸入值	儲存本次輸入偏差值	OUT

\* 繼電器輸入輸出的分配如下所示。

Bit	記號	名稱	規格	輸出輸入
0	IRST	積分重設	積分重設時輸入 ON	IN
1 ~ 7	-	(備用)	輸入用備用繼電器	IN
8 ~ F	-	(備用)	輸出用備用繼電器	OUT

## ◆ 實數型 PID 指令的參數表

位址	資料類型	記號	名稱	規格	輸出輸入
0	W	RLY	繼電器輸入輸出	繼電器輸入，繼電器輸出*	IN/OUT
1	W	-	(備用)	備用暫存器	IN
2	F	Kp	P 增益	P 補償增益 (增益 1 倍為 1.0)	IN
4	F	Ki	I 增益	積分迴路輸入增益 (增益 1 倍為 1.0)	IN
6	F	Kd	D 增益	微分迴路輸入增益 (增益 1 倍為 1.0)	IN
8	F	Ti	積分時間	積分時間 (s)	IN
10	F	Td1	發散端微分時間	來自發散方向的輸入所使用的微分時間 (s)	IN
12	F	Td2	收束端微分時間	來自收束方向的輸入所使用的微分時間 (s)	IN
14	F	IUL	積分上限 LIMIT	I 校正值的相對上限限制值	IN
16	F	ILL	積分下限 LIMIT	I 校正值的相對下限限制值	IN
18	F	UL	PID 上限 LIMIT	相對 P+I+D 補償值的上限限制值	IN
20	F	LL	PID 下限 LIMIT	相對 P+I+D 補償值的下限限制值	IN
22	F	DB	PID 輸出不感帶	相對 P+I+D 補償值的不感帶寬度	IN
24	F	Y	PID 輸出	PID 補償值輸出 (輸出至 Out)	OUT
26	F	Yi	I 補償值	I 校正值儲存	OUT
28	F	X	儲存輸入值	儲存本次輸入偏差值	OUT

\* 繼電器輸入輸出的分配和整數型相同。

## ◆ 指令內部運算

對輸入的偏差 X 進行下列運算，可求出輸出值 (PID 補償值)。

以下計算式中的 X' 為 X 的前一次輸入值，Y' 為 Y 的前一次 I 補償值，Ts 為掃描時間設定值，Td 為微分時間。

微分時間 (Td) 在 (X - X') 與 X' 符號相同時為 Td1，(X - X') 與 X' 不同符號時為 Td2。

**補充** 積分重設 (IRST) 為 ON 時，以 I 補償值 = 0 執行 PID 補償值運算。

P 校正值 = (Kp × X) 的上下限限制 (UL、LL)

Yi (I 校正值) = { (Ki × X + IREM) /  $\frac{T_i}{T_s}$  + Y' } 的上下限限制 (IUL、ILL)

D 補償值 = Kd × (X - X') ×  $\frac{T_d}{T_s}$  的上下限限制 (IUL、ILL)

Y (PID 補償值) = (P + I + DI 補償值) 補償值上下限限制 (UL、LL)，不感帶 A (寬度 DB)

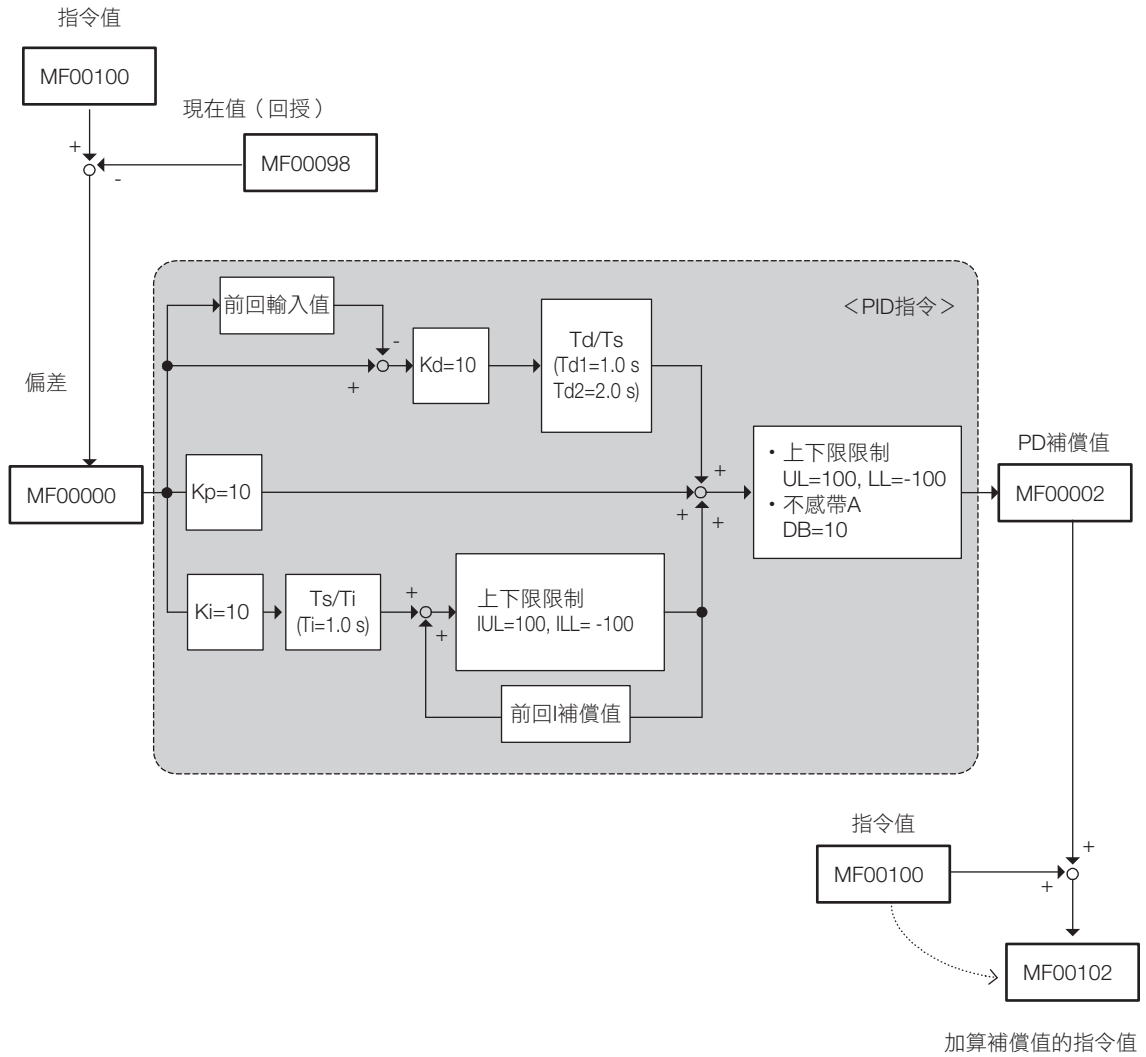
## 程式範例

考慮到 PID 補償值的指令值 (MF00102) 計算程式範例，如下所示。

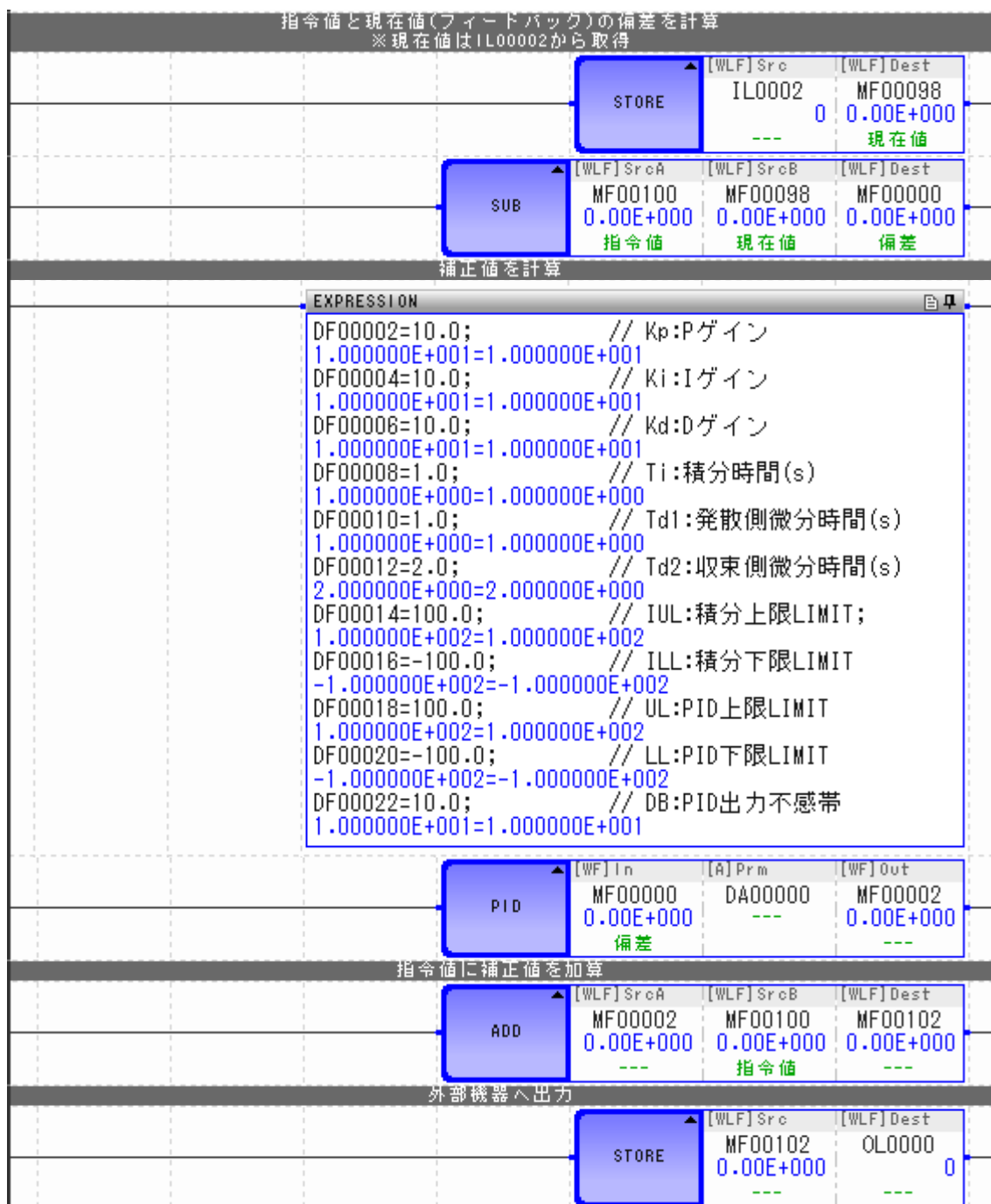
求出指令值 (MF00100) 與現在值 (MF00098) 的偏差 (MF00000)，輸入至 PID 指令。

輸出的 PID 補償值 (MF00002) 加上原本的指令值 (MF00100) 後，求出本次輸出的指令值。

程式框圖如下所示。



程式範例如下所示。



(註) OL00000 (指令値) 及 IL00002 (回授値) 為分配給外部設備の暫存器。

## 一次延遲 (LAG)

根據預設的參數表內容運算一次延遲。

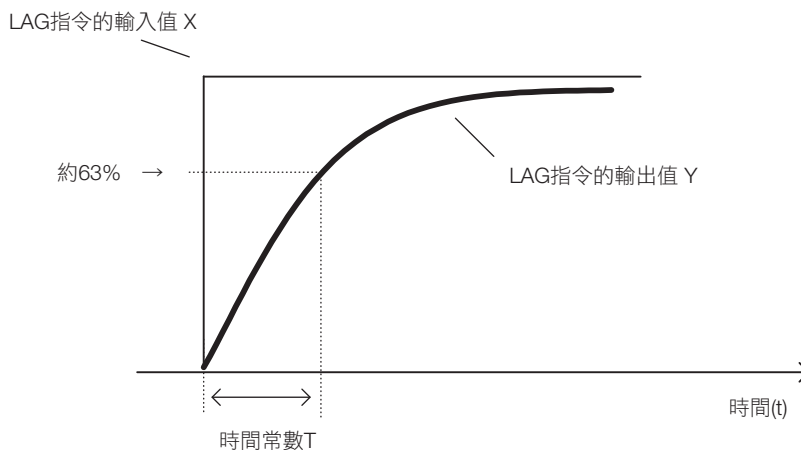
輸入至 LAG 指令的值，可使用整數型與實數型。不使用 2 倍長整數型。

整數型與實數型的參數表配置有所不同。



重要

以整數型使用時，請將掃描時間設定值設為 1 ms 的整數倍。



上圖的 LAG 運算可以下列式子表示。

$$\frac{Y(s)}{X(s)} = \frac{1}{1+T \times s}$$

即

$$T \times \frac{dY}{dt} + Y = X$$

LAG 指令內以  $dt = Ts$ 、 $dY = Y - Y'$ ，運算如下。

以下式子中的  $Y'$  為前一次輸出值， $Ts$  為掃描時間設定值， $REM$  為餘數。

$Ts$  的單位與  $T$  相同。

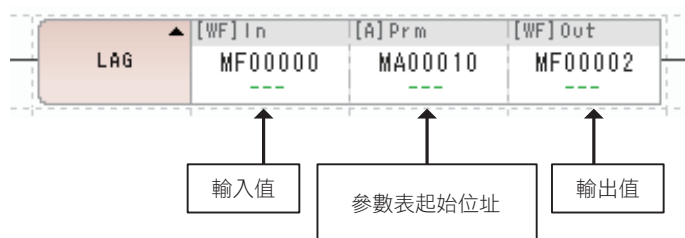
$$Y = \frac{T \times Y' + Ts \times X + REM}{T + Ts}$$

補充

LAG 重置 (IRST) 為 ON 時，輸出  $Y = 0$ 、 $REM = 0$ 。

## 格式

格式如下所示。



圖示： LAG

鍵輸入：LAG

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入值 (In)	×	○	×	×	○	×	×	○	○
參數表起始位址 (Prm)	×	×	×	×	×	×	○*	○	○
輸出值 (Out)	×	○*	×	×	○*	×	×	○	×

\* C、# 暫存器除外

### ◆ 整數型 LAG 指令的參數表

位址	資料類型	記號	名稱	規格	輸出輸入
0	W	RLY	繼電器輸入輸出	繼電器輸入，繼電器輸出*	IN/OUT
1	W	T	一次延遲時間常數	一次延遲時間常數 (ms)	IN
2	W	Y	LAG 輸出	LAG 輸出 (輸出至輸出值)	OUT
3	W	REM	餘數	餘數儲存	OUT

\* 繼電器輸入輸出的分配如下所示。

Bit	記號	名稱	規格	輸出輸入
0	IRST	LAG 重置	LAG 重置時輸入 ON	IN
1~7	-	(備用)	輸入用備用繼電器	IN
8~F	-	(備用)	輸出用備用繼電器	OUT

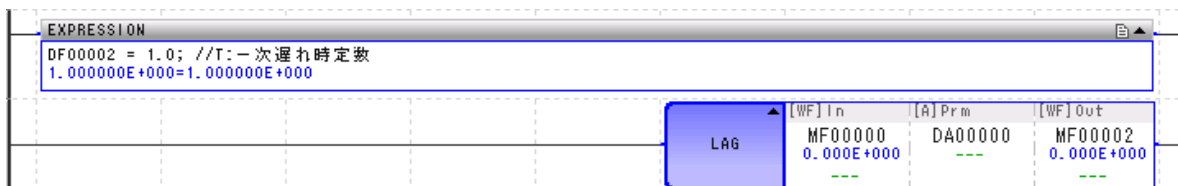
### ◆ 實數型 LAG 指令的參數表

位址	資料類型	記號	名稱	規格	輸出輸入
0	W	RLY	繼電器輸入輸出	繼電器輸入，繼電器輸出*	IN/OUT
1	W	-	(備用)	備用暫存器	-
2	F	T	一次延遲時間常數	一次延遲時間常數 (s)	IN
4	F	Y	LAG 輸出	LAG 輸出 (輸出至輸出值)	OUT

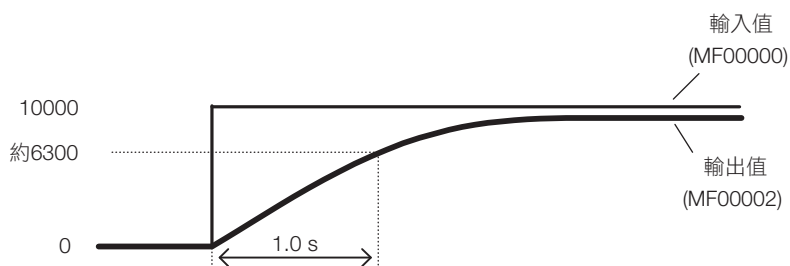
\* 繼電器輸入輸出的分配和整數型相同。

## 程式範例

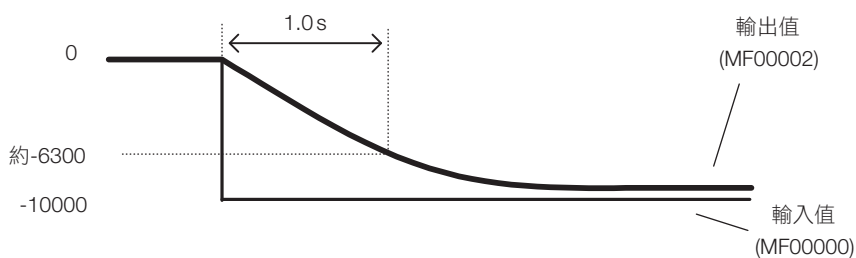
一次延遲時間常數為 1.0 時，參數表的輸入值設為 MF00000、輸出值設為 MF00002，執行 LAG 指令的程式範例，如下所示。



MF00000 產生由 0 至 10000 的變化時，MF00002 的轉變如下所示。



MF00000 產生由 0 至 -10000 的變化時，MF00002 的轉變如下所示。



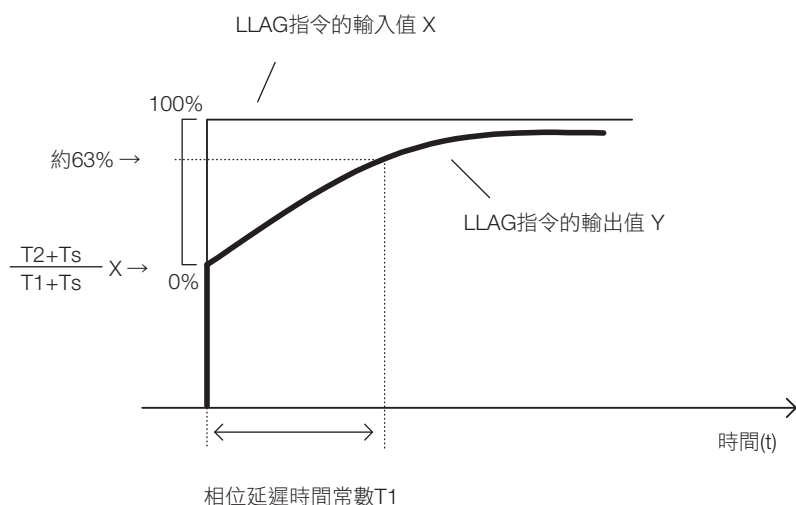
## 相位前進延遲 (LLAG)

根據預設的參數表內容運算相位前進延遲。輸入至 LLAG 指令的值，可使用整數型與實數型。不使用 2 倍長整數型。

整數型與實數型的參數表配置有所不同。



以整數型使用時，請將掃描時間設定值設為 1 ms 的整數倍。



上圖的 LLAG 運算可以下列式子表示。

$$\frac{Y(s)}{X(s)} = \frac{1+T2 \times s}{1+T1 \times s}$$

即

$$T1 \times \frac{dY}{dt} + Y = T2 \times \frac{dX}{dt} + X$$

LLAG 指令內以  $dt = Ts$ 、 $dY = Y - Y'$ 、 $dX = X - X'$ ，運算如下。

以下式子中的  $Y'$  為前一次輸出值， $X'$  為前一次輸入值， $Ts$  為掃描時間設定值， $REM$  為餘數。

$Ts$  的單位與  $T1$  相同。

$$Y = \frac{T1 \times Y' + (T2 + Ts) \times X - T2 \times X' + REM}{T1 + Ts}$$

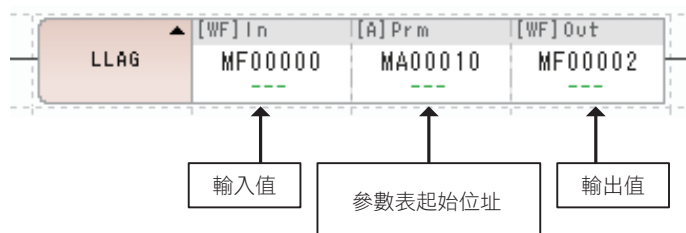
**補充**

LLAG 重置 (IRST) 為 ON 時，輸出  $Y = 0$ 、 $REM = 0$ 、 $X = 0$ 。



## 格式

格式如下所示。



圖示： LLAG

鍵輸入： LLAG

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入值 (In)	×	○	×	×	○	×	×	○	○
參數表起始位址 (Prm)	×	×	×	×	×	×	○*	○	○
輸出值 (Out)	×	○*	×	×	○*	×	×	○	×

\* C、# 暫存器除外

### ◆ 整數型 LLAG 指令的參數表

位址	資料類型	記號	名稱	規格	輸出輸入
0	W	RLY	繼電器輸入輸出	繼電器輸入，繼電器輸出*	IN/OUT
1	W	T2	相位前進時間常數	相位前進時間常數 (ms)	IN
2	W	T1	相位延遲時間常數	相位延遲時間常數 (ms)	IN
3	W	Y	LLAG 輸出	LLAG 輸出 (輸出至輸出值)	OUT
4	W	REM	餘數	餘數儲存	OUT
5	W	X	儲存輸入值	儲存輸入值	OUT

\* 繼電器輸入輸出的分配如下所示。

Bit	記號	名稱	規格	輸出輸入
0	IRST	LLAG 重置	LLAG 重置時輸入 ON	IN
1~7	-	(備用)	輸入用備用繼電器	IN
8~F	-	(備用)	輸出用備用繼電器	OUT

### ◆ 實數型 LAG 指令的參數表

位址	資料類型	記號	名稱	規格	輸出輸入
0	W	RLY	繼電器輸入輸出	繼電器輸入，繼電器輸出*	IN/OUT
1	W	-	(備用)	備用暫存器	-
2	F	T2	相位前進時間常數	相位前進時間常數 (s)	IN
4	F	T1	相位延遲時間常數	相位延遲時間常數 (s)	IN
6	F	Y	LLAG 輸出	LLAG 輸出 (輸出至輸出值)	OUT
8	F	X	儲存輸入值	儲存輸入值	OUT

\* 繼電器輸入輸出的分配和整數型相同。

### 程式範例

相位前進時間常數為 1.0 秒、相位延遲時間常數為 2.0 秒時，參數表的輸入值設為 MF00000、輸出值設為 MF00002，執行 LLAG 指令的程式範例，如下所示。

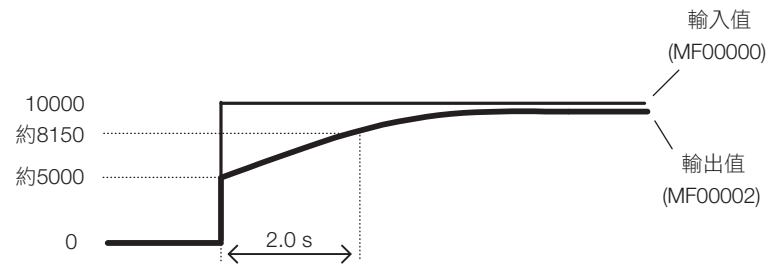
**EXPRESSION**

```

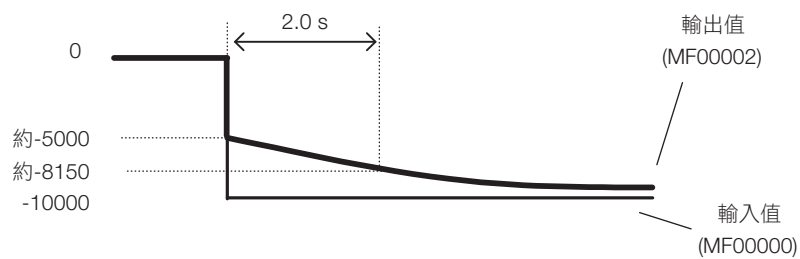
DF00002 = 1.0; //T2:位相進取時定數(s)
1.000000E+000=1.000000E+000
DF00004 = 2.0; //T1:位相遲取時定數(s)
2.000000E+000=2.000000E+000
        
```

	[WF] In	[A] Prm	[WF] Out
<b>LLAG</b>	MF00000 0.000E+000 ---	DA00000 ---	MF00002 0.000E+000 ---

MF00000 產生由 0 至 10000 的變化時，MF00002 的轉變如下所示。



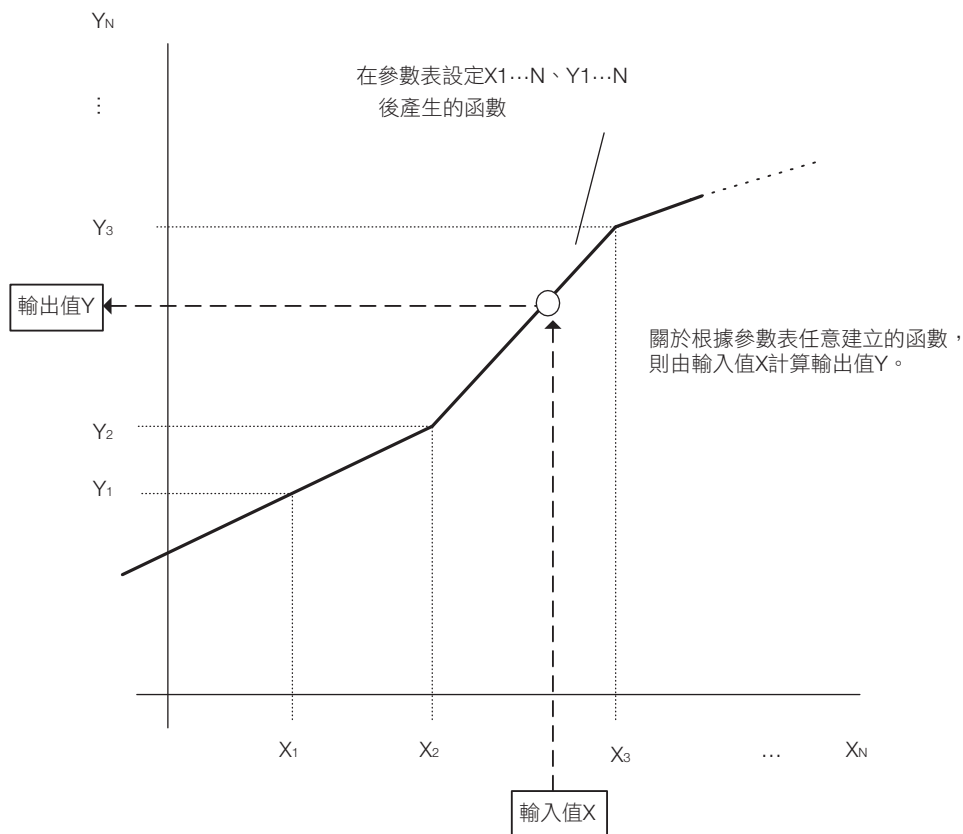
MF00000 產生由 0 至 -10000 的變化時，MF00002 的轉變如下所示。



## 函數產生器 (FGN)

根據已設定的參數表產生函數，計算對該函數給予輸入值  $X$  時的輸出值  $Y$ 。

隨著輸入值  $X$  的資料類型，將分別成為整數型、2 倍長整數型、實數型、4 倍長整數型以及 2 倍精度實數型的 FGN 指令，其參數表也將改變。

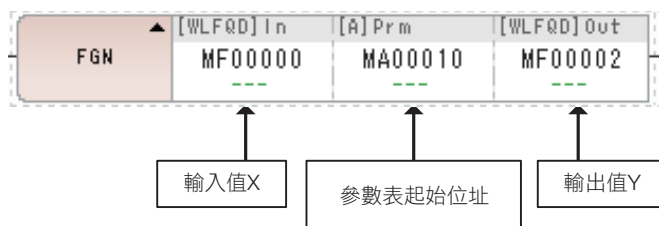


註記

請製作形成  $X_1 < X_2 < \dots < X_N$  的參數表。

## 格式

格式如下所示。



圖示： FGN

鍵輸入：FGN

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入值 X (In)	×	○	○	○	○	○	×	○	○
參數表 開頭位址 (Prm)	×	×	×	×	×	×	○	×	×
輸出值 (Out)	×	○*	○*	○*	○*	○*	×	○	×

\* C、# 暫存器除外

### ◆ 整數型 FGN 指令的參數表

輸入值 X 為整數型時，則為整數型 FGN 指令。

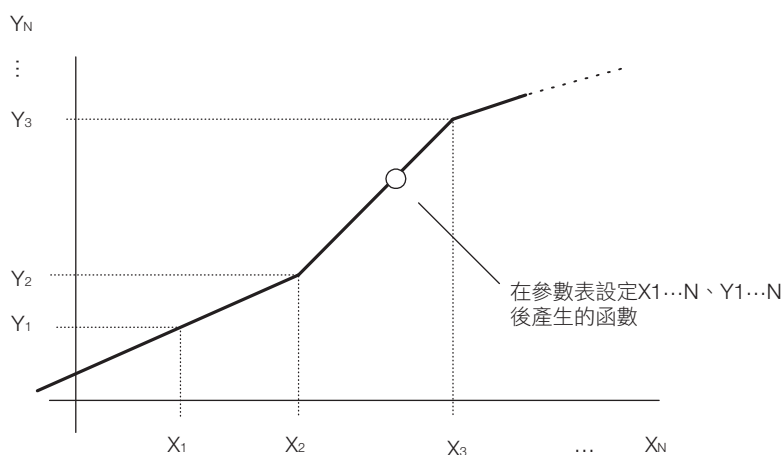
請如下所示建構參數表。

位址	資料類型	記號	名稱
0	W	N	X、Y 組數
1	W	X <sub>1</sub>	資料 X <sub>1</sub>
2	W	Y <sub>1</sub>	資料 Y <sub>1</sub>
3	W	X <sub>2</sub>	資料 X <sub>2</sub>
4	W	Y <sub>2</sub>	資料 Y <sub>2</sub>
:	:	:	:
2N-1	W	X <sub>N</sub>	資料 X <sub>N</sub>
2N	W	Y <sub>N</sub>	資料 Y <sub>N</sub>

### ◆ 2 倍長整數型／實數型 FGN 指令的參數表

輸入值 X 為 2 倍長整數型時則為 2 倍長整數型 FGN 指令，為實數型時則為實數型 FGN 指令。  
請如下所示建構參數表。

位址	資料類型	記號	名稱
0	W	N	X、Y 組數
1	W	-	保留
2	L/F	$X_1$	資料 $X_1$
4	L/F	$Y_1$	資料 $Y_1$
6	L/F	$X_2$	資料 $X_2$
8	L/F	$Y_2$	資料 $Y_2$
:	:	:	:
$4N-2$	L/F	$X_N$	資料 $X_N$
$4N$	L/F	$Y_N$	資料 $Y_N$



### ◆ 4 倍長整數型／2 倍精度實數型 FGN 指令參數表

輸入值 X 為 4 倍長整數型時，為 4 倍長整數型 FGN 指令，輸入值 X 為 2 倍精度實數型時，為 2 倍精度實數型 FGN 指令。

請如下所示建構參數表。

位址	資料類型	記號	名稱
0	W	N	X、Y 組數
1	W	-	保留
2	L	-	保留
4	Q/D	$X_1$	資料 $X_1$
8	Q/D	$Y_1$	資料 $Y_1$
12	Q/D	$X_2$	資料 $X_2$
16	Q/D	$Y_2$	資料 $Y_2$
:	:	:	:
$8N-4$	Q/D	$X_N$	資料 $X_N$
$8N$	Q/D	$Y_N$	資料 $Y_N$



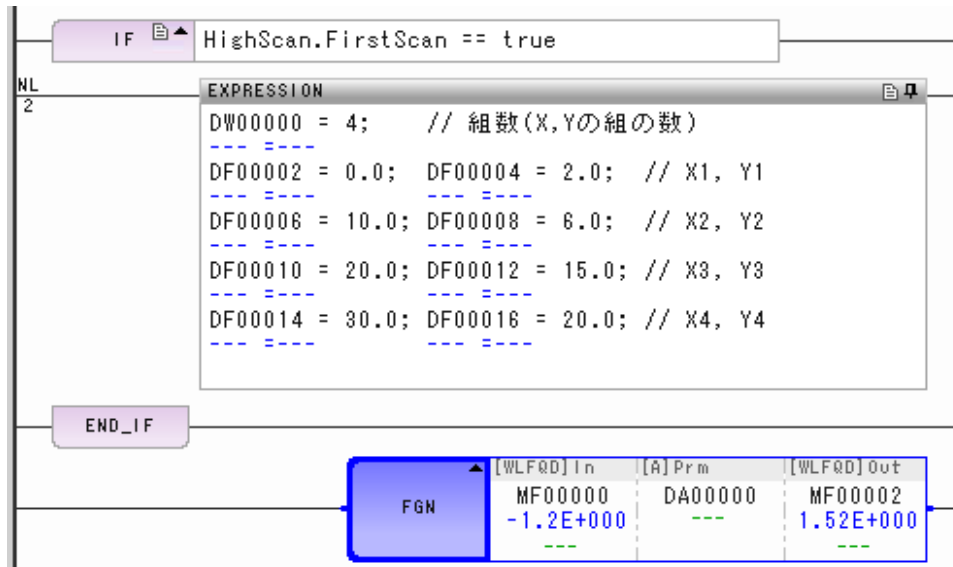
註記

整數型、2 倍長整數型、實數型、4 倍長整數型以及 2 倍精度實數型的任何一組參數表，請務必以  $X_1 < X_2 < \dots < X_N$  進行設定。

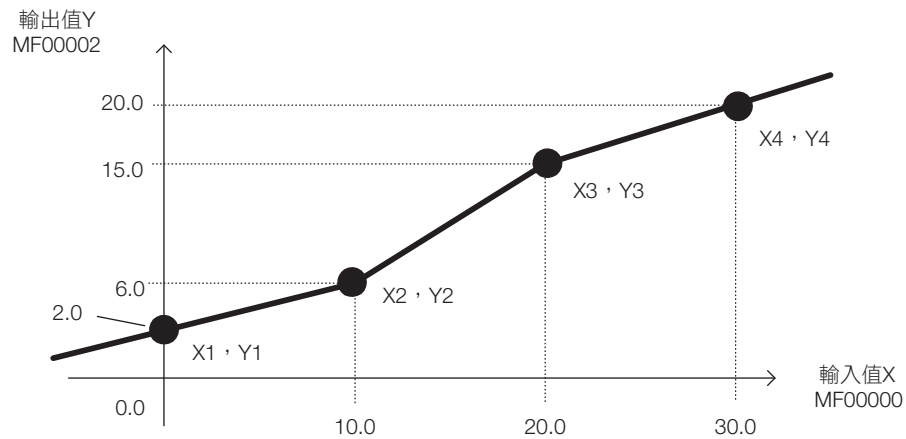
### 程式範例

依下列參數表，使用實數型 FGN 指令產生函數時的程式範例。如下所示。

組數	4
X1, Y1	0.0, 2.0
X2, Y2	10.0, 6.0
X3, X4	20.0, 15.0
X4, Y4	30.0, 20.0



輸入值 X (MF00000) 與輸出值 Y (MF00002) 的關係，如下圖所示。



## 補充事項

輸出值 Y 計算如下。

- 可為  $X_n \leq \text{輸入值 } X \leq X_{n+1}$  的  $X_n$ 、 $Y_n$  組合存在時

$$\text{輸出值 } Y = Y_n + \frac{Y_{n+1} - Y_n}{X_{n+1} - X_n} \times (\text{輸入值 } X - X_n) \quad (1 \leq n \leq N - 1)$$

- 可為  $X_n \leq \text{輸入值 } X \leq X_{n+1}$  的  $X_n$ 、 $Y_n$  組合不存在時

輸入值  $X < X_1$  時

$$\text{輸出值 } Y = Y_1 + \frac{Y_2 - Y_1}{X_2 - X_1} \times (\text{輸入值 } X - X_1)$$

輸入值  $X > X_N$  時

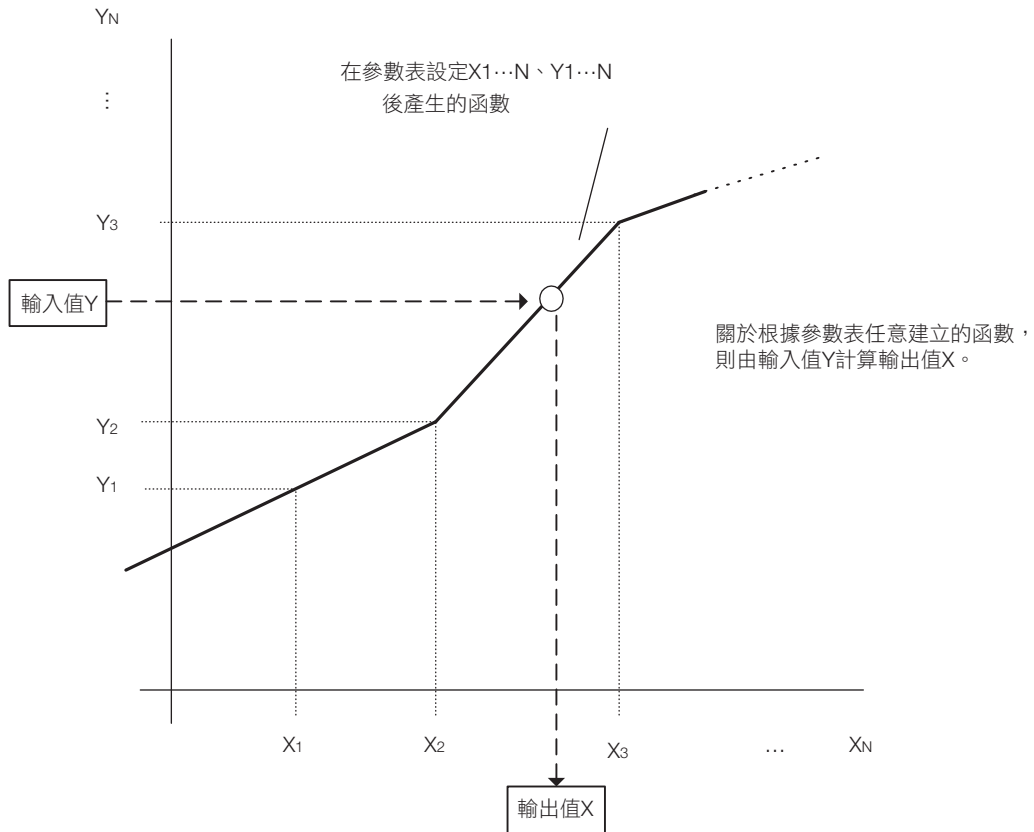
$$\text{輸出值 } Y = Y_N + \frac{Y_N - Y_{N-1}}{X_N - X_{N-1}} \times (\text{輸入值 } X - X_N)$$

## 反函數產生器 (IFGN)

根據已設定的參數表產生函數，計算對該函數給予輸入值  $Y$  時的輸出值  $X$ 。

隨著輸入值  $X$  的資料類型，將分別成為整數型、2 倍長整數型、實數型、4 倍長整數型以及 2 倍精度實數型的 IFGN 指令。

參數表的結構則與 FGN 指令相同。



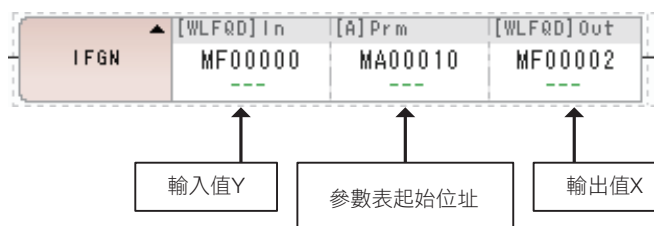
註記

請設定形成  $Y_1 < Y_2 < \dots < Y_N$  的參數表。



## 格式

格式如下所示。



圖示：[1](#)  
**IFGN**

鍵輸入：IFGN

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入值 Y (In)	×	○	○	○	○	○	×	○	○
參數表起始位址 (Prm)	×	×	×	×	×	×	○	×	×
輸出值 X (Out)	×	○*	○*	○*	○*	○*	×	○	×

\* C、# 暫存器除外

### ◆ 整數型 IFGN 指令的參數表

輸入值 X 為整數型時，則為整數型 IFGN 指令。

請如下所示建構參數表。

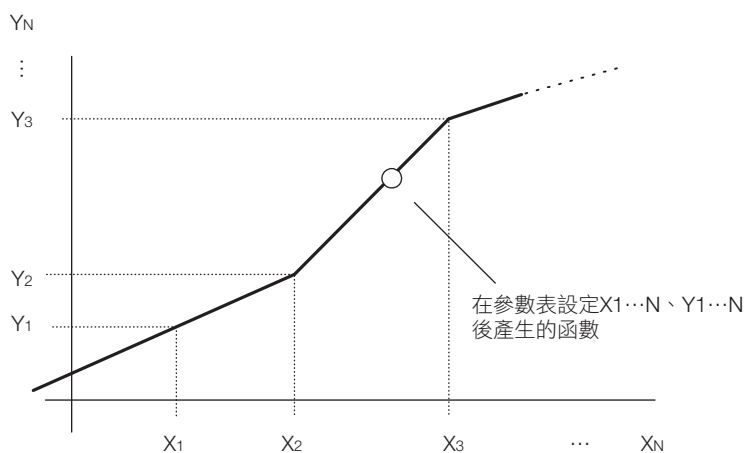
位址	資料類型	記號	名稱
0	W	N	X、Y 組數
1	W	X <sub>1</sub>	資料 X <sub>1</sub>
2	W	Y <sub>1</sub>	資料 Y <sub>1</sub>
3	W	X <sub>2</sub>	資料 X <sub>2</sub>
4	W	Y <sub>2</sub>	資料 Y <sub>2</sub>
:	:	:	:
2N-1	W	X <sub>N</sub>	資料 X <sub>N</sub>
2N	W	Y <sub>N</sub>	資料 Y <sub>N</sub>

## ◆ 2 倍長整數型／實數型 IFGN 指令的參數表

輸入值 Y 為 2 倍長整數型時則為 2 倍長整數型 IFGN 指令，為實數型時則為實數型 IFGN 指令。

請如下所示建構參數表。

位址	資料類型	記號	名稱
0	W	N	X、Y 組數
1	W	-	保留
2	L/F	$X_1$	資料 $X_1$
4	L/F	$Y_1$	資料 $Y_1$
6	L/F	$X_2$	資料 $X_2$
8	L/F	$Y_2$	資料 $Y_2$
:	:	:	:
$4N-2$	L/F	$X_N$	資料 $X_N$
$4N$	L/F	$Y_N$	資料 $Y_N$



## ◆ 4 倍長整數型／2 倍精度實數型 FGN 指令參數表

輸入值 X 為 4 倍長整數型時，為 4 倍長整數型 FGN 指令，輸入值 X 為 2 倍精度實數型時，為 2 倍精度實數型 FGN 指令。

請如下所示建構參數表。

位址	資料類型	記號	名稱
0	W	N	X、Y 組數
1	W	-	保留
2	L	-	保留
4	Q/D	$X_1$	資料 $X_1$
8	Q/D	$Y_1$	資料 $Y_1$
12	Q/D	$X_2$	資料 $X_2$
16	Q/D	$Y_2$	資料 $Y_2$
:	:	:	:
$8N-4$	Q/D	$X_N$	資料 $X_N$
$8N$	Q/D	$Y_N$	資料 $Y_N$



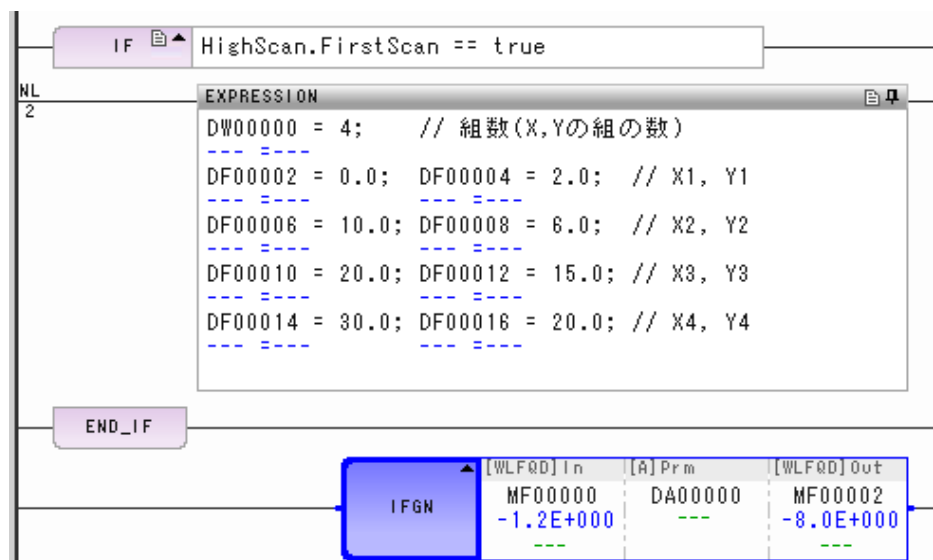
整數型、2 倍長整數型、實數型、4 倍長整數型以及 2 倍精度實數型的任何一組參數表，請務必以  $Y_1 < Y_2 < \dots < Y_N$  進行設定。

註記

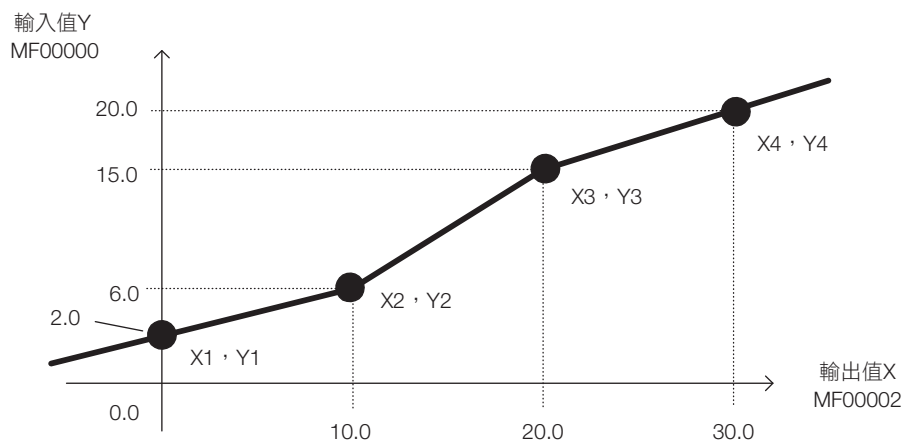
## 程式範例

依下列參數表，使用實數型 IFGN 指令產生函數時的程式範例。如下所示。

組數	4
X1, Y1	0.0, 2.0
X2, Y2	10.0, 6.0
X3, Y3	20.0, 15.0
X4, Y4	30.0, 20.0



輸入值 Y (MF00000) 與輸出值 X (MF00002) 的關係，如下圖所示。



## 補充事項

輸出值 X 計算如下。

- 可為  $Y_n \leq \text{輸入值 } Y \leq Y_{n+1}$  的  $X_n$ 、 $Y_n$  組合存在時

$$\text{輸出值 } X = X_n + \frac{X_{n+1} - X_n}{Y_{n+1} - Y_n} \times (\text{輸入值 } Y - Y_n) \quad (1 \leq n \leq N - 1)$$

- 可為  $Y_n \leq \text{輸入值 } Y \leq Y_{n+1}$  的  $X_n$ 、 $Y_n$  組合不存在時

輸入值  $Y < Y_1$  時

$$\text{輸出值 } X = X_1 + \frac{X_2 - X_1}{Y_2 - Y_1} \times (\text{輸入值 } Y - Y_1)$$

輸入值  $Y > Y_N$  時

$$\text{輸出值 } X = X_N + \frac{X_N - X_{N-1}}{Y_N - Y_{N-1}} \times (\text{輸入值 } Y - Y_N)$$

## 直線加減速器 1 (LAU)

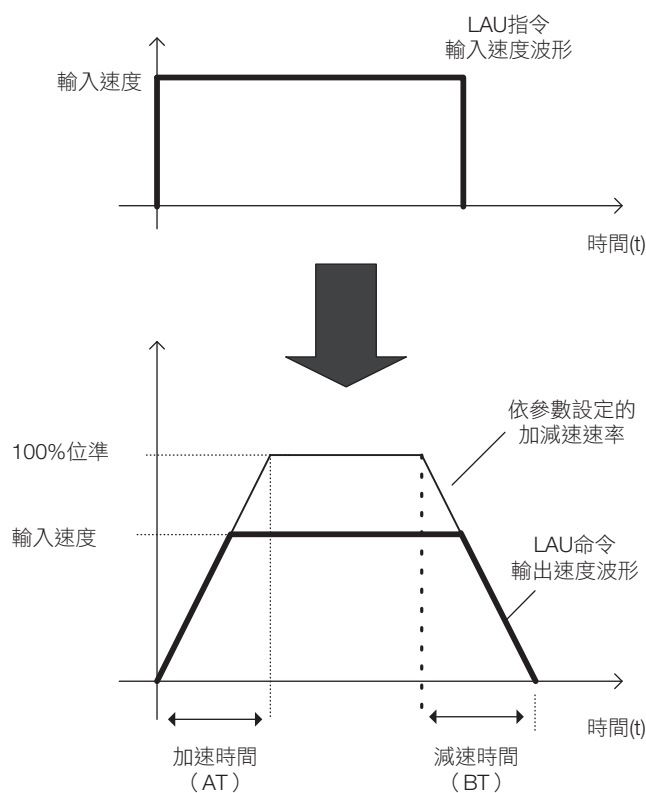
對於輸入速度，輸出以固定加減速率加速／減速後的速度結果。依據已設定的參數表內容執行加減速。輸入至 LAU 運算的值，可使用整數型與實數型。但無法使用 2 倍長整數型、4 倍長整數型以及 2 倍精度實數型。

整數型與實數型的參數表配置有所不同。



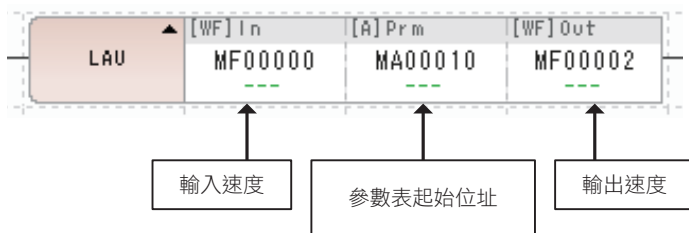
以整數型使用時，請將掃描時間設定值設為 1 ms 的整數倍。


重要



## 格式

格式如下所示。



圖示：

鍵輸入：LAU

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入速度 (In)	×	○	×	×	○	×	×	○	○
參數表起始位址 (Prm)	×	×	×	×	×	×	○	×	×
輸出速度 (Out)	×	○*	×	×	○*	×	×	○	×

\* C、# 暫存器除外

### ◆ 整數型 LAU 指令的參數表

位址	資料類型	記號	名稱	規格	輸出輸入
0	W	RLY	繼電器輸入輸出	繼電器輸入，繼電器輸出*	IN/OUT
1	W	LV	輸入 100% 等級	輸入 100% 的光學尺	IN
2	W	AT	加速時間	0% ~ 100% 的加速時間 (0.1 s)	IN
3	W	BT	減速時間	100% ~ 0% 的減速時間 (0.1 s)	IN
4	W	QT	急停止時間	100% ~ 0% 的急停止時間 (0.1 s)	IN
5	W	V	當前速度	LAU 輸出 (輸出至輸出速度)	OUT
6	W	DVDT	當前加減速	通常加速率之比例為 5000	OUT
7	W	-	(備用)	備用暫存器	-
8	W	VIM	前次速度指令	儲存速度指令輸入的前次值	OUT
9	W	DVDTK	DVDT 係數	目前加速度 (DVDT) 的比率係數	IN
10	L	REM	餘數	加減速率餘數	OUT

\* 繼電器輸入輸出的分配如下所示。


Bit	記號	名稱	規格	輸出輸入
0	RN	線路運轉中	線路運轉中輸入 ON	IN
1	QS	緊急停止	急停止時輸入 OFF	IN
2	DVDTF	DVDT 運算不執行	不執行 DVDT 運算輸入 ON	IN
3	DVDT S	DVDT 運算選擇	選擇 DVDT 運算方式	IN
4 ~ 7	-	(備用)	輸入用備用繼電器	IN
8	ARY	加速中	加速中輸出 ON	OUT
9	BRY	減速中	減速中輸出 ON	OUT
A	LSP	零速	零速時輸出 ON	OUT
B	EQU	一致	輸入速度 = 以輸出速度輸出 ON	OUT
C ~ F	-	(備用)	輸出用備用繼電器	OUT

(註) 緊急停止 (QS) 為 OFF 時，使用緊急停止時間 (QT) 作為加減速時間。

## ◆ 實數型 LAU 指令的參數表

位址	資料類型	記號	名稱	規格	輸出輸入
0	W	RLY	繼電器輸入輸出	繼電器輸入、繼電器輸出*2	IN/OUT
1	W	-	(備用)	備用暫存器	-
2	F	LV*1	輸入 100% 等級	輸入 100% 的光學尺	IN
4	F	AT	加速時間	0% ~ 100% 的加速時間 (s)	IN
6	F	BT	減速時間	100% ~ 0% 的減速時間 (s)	IN
8	F	QT	急停止時間	100% ~ 0% 的急停止時間 (s)	IN
10	F	V	當前速度	LAU 輸出 (輸出至輸出速度)	OUT
12	F	DVDT	當前加減速	輸出當前的加減速	OUT

\*1. 實際的加減速時間，由輸入 100% 等級的設定值 (LV) 與輸入速度比率決定。  
LAU 指令的內部運算詳情，請參照以下項目。

 補充事項 (第 4-193 頁)

\*2. 繼電器輸入輸出的分配如下所示。

Bit	記號	名稱	規格	輸出輸入
0	RN	線路運轉中	線路運轉中輸入 ON	IN
1	QS	緊急停止	急停止時輸入 OFF	IN
2 ~ 7	-	(備用)	輸入用備用繼電器	IN
8	ARY	加速中	加速中輸出 ON	OUT
9	BRY	減速中	減速中輸出 ON	OUT
A	LSP	零速	零速時輸出 ON	OUT
B	EQU	一致	輸入速度 = 以輸出速度輸出 ON	OUT
C ~ F	-	(備用)	輸出用備用繼電器	OUT

## 補充

當緊急指 (QS) OFF，以緊急停止時間 (QT) 為加減速時間。

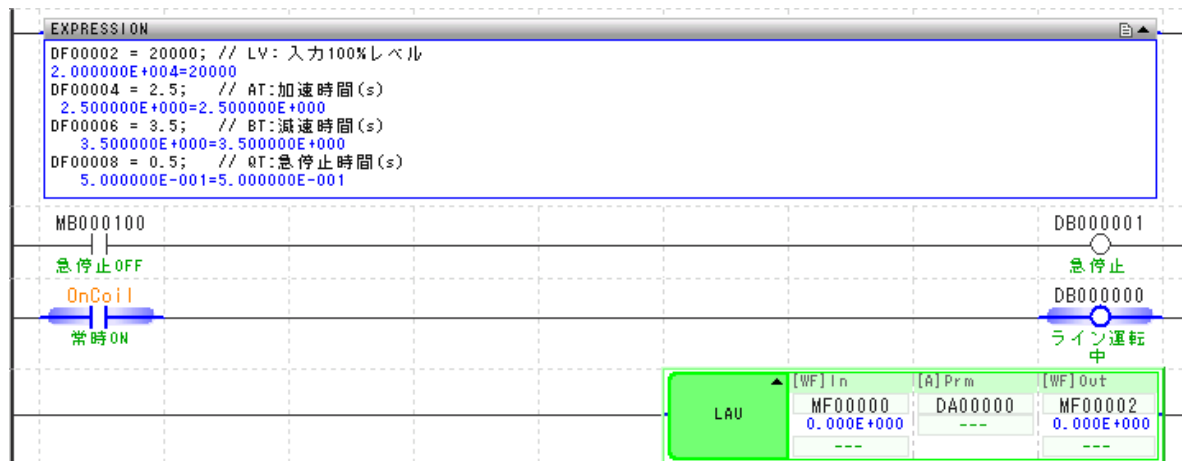
如要緊急停止，請於緊急停止 (QS) 設為 OFF 的同時讓輸入速度成為 0。

## 程式範例

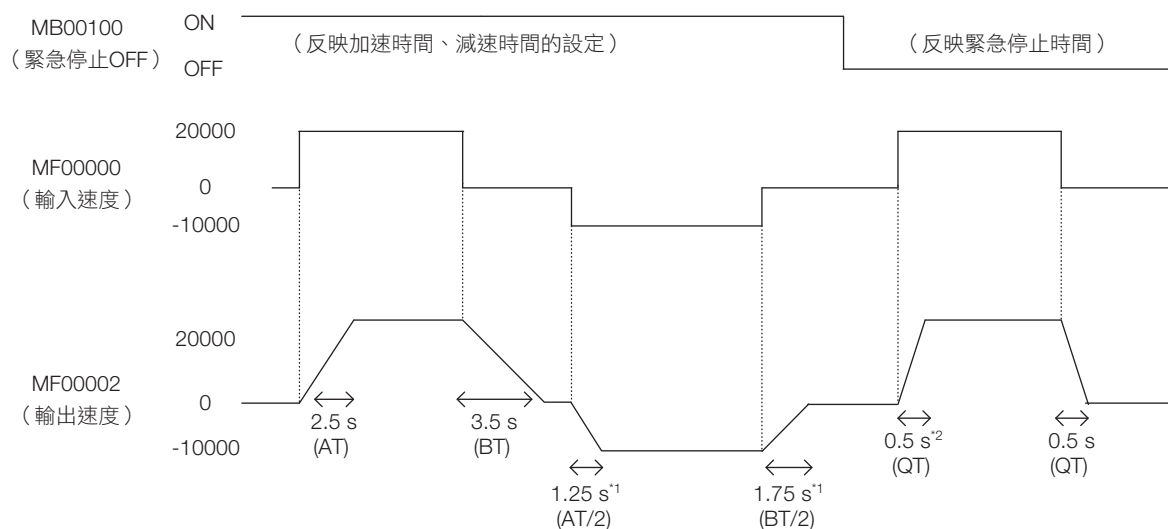
使用建立的加減速率，當輸入速度為 MF00000、輸出速度為 MF00002，執行實數型 LAU 指令時的程式範例，如下所示。

下列是為建構加減速率所設定的參數。

- 加減速率的輸入 100% 等級 = 20000
- 加速時間 = 2.5 s
- 減速時間 = 3.5 s
- 急停止時間 = 0.5 s



各暫存器的動作如下所示。



\*1. 套用遠離 0 的方向為加速時間，接近 0 的方向為減速時間。

\*2. 緊急停止時間也將反映至加速時間。



## 補充事項

### ◆ 計算速度輸出值及當前加減速的運算公式

以下說明求出速度輸出值（加速時、減速時、緊急停止時）以及目前加減速度的運算式。

#### ■ 整數型 LAU 指令

在整數型 LAU 指令中，下列運算式將依據已設定參數求出速度輸出值（加速時、減速時、緊急停止時）以及目前加減速度。

運算公式的 V 為速度輸出值，V' 為前次速度輸出值，VI 為速度指令的輸入值，Ts 為掃描時間設定值。

##### • 加速時的速度輸出值

可使用下列運算公式求出加速時的速度輸出值。

若  $VI > V'$  ( $V' \geq 0$ ) 時，則  $V = V' + ADV$ 。

若  $VI < V'$  ( $V' \leq 0$ ) 時，則  $V = V' - ADV$ 。

$$\text{加速速率 (ADV)} = \frac{LV \times Ts (0.1 \text{ ms}) + \text{REM}}{AT (0.1 \text{ s}) \times 1000}$$

##### • 減速時的速度輸出值

可使用下列運算公式求出減速時的速度輸出值。

$VI > V'$  ( $V' < 0$ )，則  $V = V' + BDV$ 。

$VI < V'$  ( $V' > 0$ )，則  $V = V' - BDV$ 。

$$\text{減速速率 (BDV)} = \frac{LV \times Ts (0.1 \text{ ms}) + \text{REM}}{BT (0.1 \text{ s}) \times 1000}$$

##### • 急停止時的速度輸出值

可使用下列運算公式求出急停止時的速度輸出值。

QS = OFF ( $VI > V'$ ,  $V' < 0$ )，則  $V = V' + QDV$ 。

QS = OFF ( $VI < V'$ ,  $V' > 0$ )，則  $V = V' - QDV$ 。

$$\text{緊急停止速率 (QDV)} = \frac{LV \times Ts (0.1 \text{ ms}) + \text{REM}}{QT (0.1 \text{ s}) \times 1000}$$

##### • 當前加減速

DVDT 運算不執行 (DVDTF) 為 ON 時，將遵照 DVDT 運算選擇 (DVDTs)，以下列任一運算式求出目前加減速度 (DVDT)。當 DVDTF 為 OFF，DVDT 變為 0。

$$\text{DVDTs} = \text{ON}, \text{ 則 } DVDT = \frac{(V - V') \times 5000}{ADV}$$

$$\text{DVDTs} = \text{ON}, \text{ 則 } DVDT = (V - V') \times \text{DVDTK}$$

#### 補充

1. 加速中 (ARY) 的繼電器為 ON 的條件如下所示。  
 $V' \geq 0$  且  $ADV > 0$  時，或  $V' \leq 0$  且  $ADV < 0$  時
2. 減速中 (BRY) 的繼電器為 ON 的條件如下所示。  
  - $V' < 0$  且  $BDV > 0$  時，或  $V' > 0$  且  $BDV < 0$  時
  - $V' < 0$  且  $QDV > 0$  時，或  $V' > 0$  且  $QDV < 0$  時
3. 零速 (LSP) 的繼電器為 ON 的條件為  $V = 0$ 。
4. 一致 (EQU) 的繼電器為 ON 的條件為  $VI = V$ 。
5. 當直線運轉中 (RN) 為 OFF 時，輸出  $V = 0$ 、 $DVDT = 0$ 、 $\text{REM} = 0$ 。

### ■ 實數型 LAU 指令

在實數型 LAU 指令中，下列運算式將依據已設定參數求出速度輸出值（加速時、減速時、緊急停止時）以及目前加減速度。

運算式中的 V 為速度輸出值，V' 為前次速度輸出值，VI 為速度指令輸入值，Ts 為掃描時間設定值。

#### • 加速時的速度輸出值

可使用下列運算公式求出加速時的速度輸出值。

若  $VI > V'$  ( $V' \geq 0$ ) 時，則  $V = V' + ADV$ 。

若  $VI < V'$  ( $V' \leq 0$ ) 時，則  $V = V' - ADV$ 。

$$\text{加速速率 (ADV)} = \frac{LV \times Ts (0.1 \text{ ms})}{AT (s) \times 10000}$$

#### • 減速時的速度輸出值

可使用下列運算公式求出減速時的速度輸出值。

$VI < V'$  ( $V' > 0$ )，則  $V = V' + BDV$ 。

$VI > V'$  ( $V' < 0$ )，則  $V = V' - BDV$ 。

$$\text{減速速率 (BDV)} = \frac{-LV \times Ts (0.1 \text{ ms})}{BT (s) \times 10000}$$

#### • 急停止時的速度輸出值

可使用下列運算公式求出急停止時的速度輸出值。

QS = OFF ( $VI < V'$ ,  $V' > 0$ )，則  $V = V' + QDV$ 。

QS = OFF ( $VI > V'$ ,  $V' < 0$ )，則  $V = V' - QDV$ 。

$$\text{緊急停止速率 (QDV)} = \frac{-LV \times Ts (0.1 \text{ ms})}{QT (s) \times 10000}$$

#### • 當前加減速

目前加減速度 (DVDT) 將於運算速度輸出值 (V) 後，以下列運算式求出。

$$DVDT = V - V'$$

#### 補充

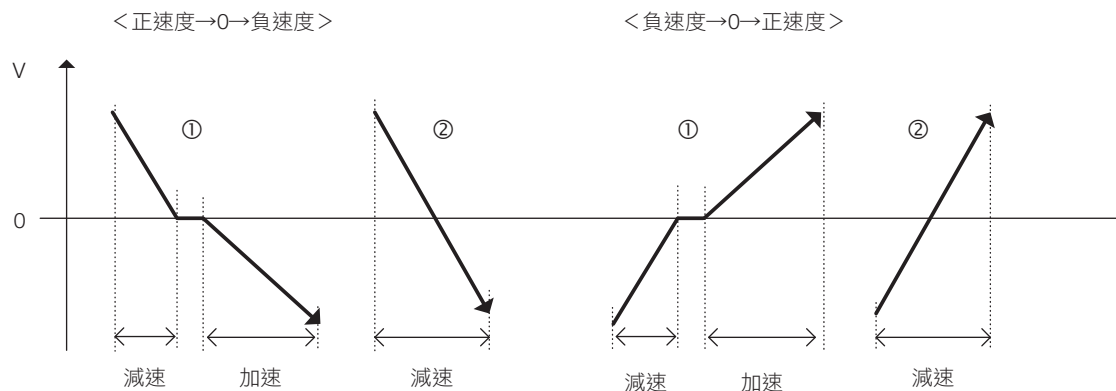
1. 加速中 (ARY) 的繼電器為 ON 的條件如下所示。  
 $V' \geq 0$  且  $ADV > 0$  時，或  $V' \leq 0$  且  $ADV < 0$  時
2. 減速中 (BRY) 的繼電器為 ON 的條件如下所示。
  - $V' < 0$  且  $BDV > 0$  時，或  $V' > 0$  且  $BDV < 0$  時
  - $V' < 0$  且  $QDV > 0$  時，或  $V' > 0$  且  $QDV < 0$  時
3. 零速 (LSP) 繼電器變為 ON 的條件為  $V = 0$ 。
4. 一致 (EQR) 繼電器變為 ON 的條件為  $VI = V$ 。
5. 加速中 (ARY) 繼電器變為 ON 的條件如下所示。  
 $V \neq V'$  且 DVDT 與 V 為同符號
6. 減速中 (BRY) 繼電器變為 ON 的條件如下所示。  
 $V \neq V'$  且 DVDT 與 V 為異符號
7. 當直線運轉中 (RN) 為 OFF 時，輸出  $V = 0$ 、 $DVDT = 0$ 。

### ◆ 輸入速度變更時的加減速

以下就輸入速度變更時的加減速進行說明。

當速度 = 0 而暫停後，將以設定的減速時間及加速時間進行加減速。(參照 ①)

在速度 = 0 前後下指令時，將在不變動速度情況下以減速時間進行加減速。(參照 ②)



## 直線加減速器 2 (SLAU)

對於輸入速度，輸出以可變加減速率加速／減速後的速度結果。依據已設定的參數表內容執行 S 型曲線加減速。

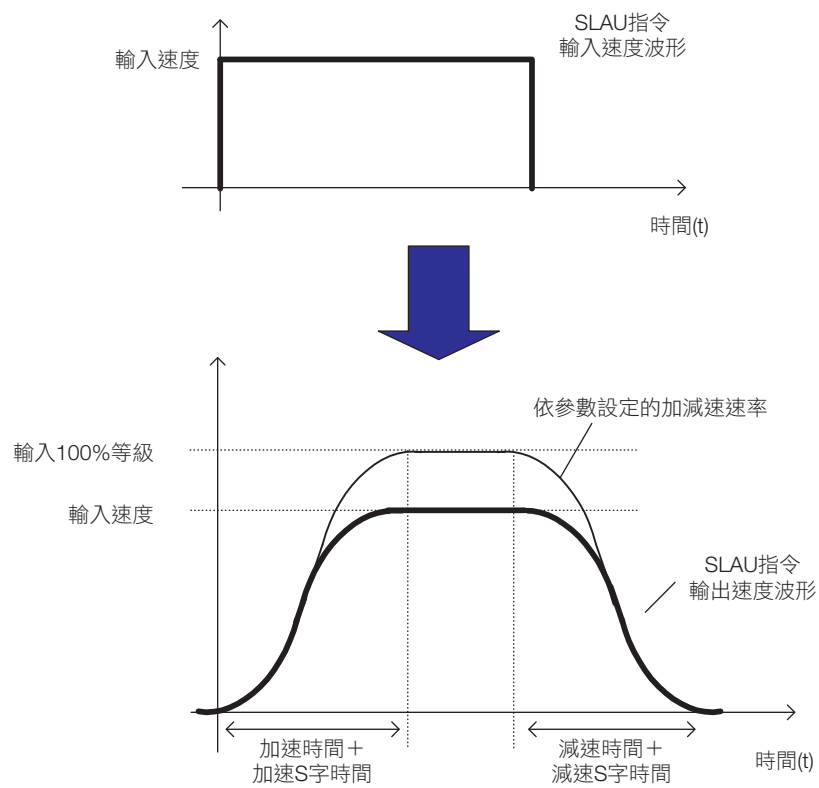
輸入至 SLAU 運算的值，可使用整數型、2 倍長整數型或實數型。但無法使用 4 倍長整數型以及 2 倍精度實數型。

整數型與實數型的參數表配置有所不同。



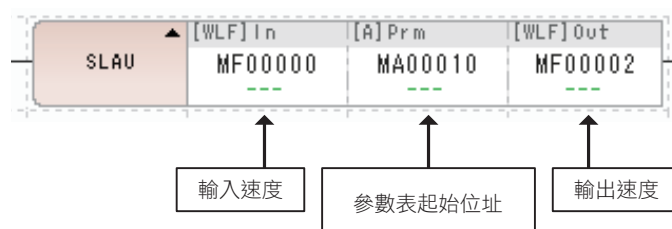
重要


以整數型使用時，請將掃描時間設定值設為 1 ms 的整數倍。



## 格式

格式如下所示。



圖示：  
鍵輸入：SLAU

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入速度 (In)	×	○	○	×	○	×	×	○	○
參數表起始位址 (Prm)	×	×	×	×	×	×	○*	○*	×
輸出速度 (Out)	×	○*	○*	×	○*	×	×	○	×

\* C、# 暫存器除外

### ◆ 整數型 SLAU 指令的參數表

位址	資料類型	記號	名稱	規格	輸出輸入
0	W	RLY	繼電器輸入輸出	繼電器輸入，繼電器輸出*	IN/OUT
1	W	LV	輸入 100% 等級	輸入 100% 的光學尺	IN
2	W	AT	加速時間	0% ~ 100% 的加速時間 (0.1 s)	IN
3	W	BT	減速時間	100% ~ 0% 的減速時間 (0.1 s)	IN
4	W	QT	急停止時間	100% ~ 0% 的急停止時間 (0.1 s)	IN
5	W	AAT	加速 S 字時間	加速時的 S 型領域時間 (0.01 ~ 32.00 s)	IN
6	W	BBT	減速 S 字時間	減速時的 S 型領域時間 (0.01 ~ 32.00 s)	IN
7	W	V	當前速度	SLAU 輸出 (輸出至輸出速度)	OUT
8	W	DVDT1	當前加減速 1	通常加速率之比例為 5000	OUT
9	W	-	(備用)	備用暫存器	-
10	W	ABMD	保持時速度上升	下達保持指令後到穩定時的速度變化量	OUT
11	W	REM1	餘數	加減速率餘數	OUT
12	W	-	(備用)	備用暫存器	-
13	W	VIM	前次速度指令	儲存速度指令輸入的前次值用	OUT
14	L	DVDT2	當前加減速 2	以實際加減速度乘以 1000 倍	OUT
16	L	DVDT3	當前加減速 3	目前加減速度 (= DVDT2/1000)	OUT
18	L	REM2	餘數	S 型區間加減速率餘數	OUT
20	W	REM3	餘數	目前速度餘數	OUT
21	W	DVDTK	DVDT1 係數	目前加速度 1 (DVDT) 的比率係數 (-32768 ~ 32767)	OUT

\* 繼電器輸入輸出的分配如下所示。

Bit	記號	名稱	規格	輸出輸入
0	RN	線路運轉中	線路運轉中輸入 ON	IN
1	QS	緊急停止	急停止時輸入 OFF	IN
2	DVDTF	DVDT1 運算不執行	不執行 DVDT 運算輸入 ON	IN
3	DVDT S	DVDT1 運算選擇	選擇 DVDT 運算方式	IN
4~7	-	(備用)	輸入用備用繼電器	IN
8	ARY	加速中	加速中輸出 ON	OUT
9	BRY	減速中	減速中輸出 ON	OUT
A	LSP	零速	零速時輸出 ON	OUT
B	EQU	一致	輸入速度 = 出力速度時輸出 ON	OUT
C	-	(備用)	輸出用備用繼電器	OUT
D	CCF	輔助繼電器	系統內輔助繼電器	OUT
E	BBF	輔助繼電器	系統內輔助繼電器	OUT
F	AAF	輔助繼電器	系統內輔助繼電器	OUT

(註) 急停止 (QS) 為 OFF 時，使用急停止時間 (QT) 作為減速時間。

### ◆ 2 倍長整數型 SLAU 指令的參數表

位址	資料類型	記號	名稱	規格	輸出輸入
0	W	RLY	繼電器輸入輸出	繼電器輸入、繼電器輸出 <sup>*2</sup>	IN/OUT
1	W	-	(備用)	-	-
2	L	LV	輸入 100% 等級	輸入值 100% 的比例	IN
4	L	AT	加速時間	0% ~ 100% 的加速時間 (0.1 s)	IN
6	L	BT	減速時間	100% ~ 0% 的減速時間 (0.1 s)	IN
8	L	QT	急停止時間	100% ~ 0% 的緊急停止時間 (0.1 s)	IN
10	L	AAT	加速 S 字時間	加速時的 S 型領域時間 (0.01 s)	IN
12	L	BBT	減速 S 字時間	減速時的 S 型領域時間 (0.01 s)	IN
14	L	V	當前速度	SLAU 輸出 (同時輸出至 A 暫存器)	OUT
16	L	DVDT	當前加減速	輸出當前的加減速 (小數點以下捨去後輸出)	OUT
18	L	ABMD	保持時速度上升	下達保持指令後到穩定時的速度變化量	OUT
20	D <sup>*1</sup>	V_D	當前速度	系統用 SLAU 輸出 (2 倍精度實數)	IN/OUT
24	D <sup>*1</sup>	DVDT_D	當前加減速	系統用 目前加減速度 (2 倍精度實數)	IN/OUT

\*1. D 為 2 倍精度實數，4 個字元。MPE720 無法作為實數值顯示。

\*2. 繼電器輸入輸出的分配如下所示。

Bit	記號	名稱	規格	輸出輸入
0	RN	線路運轉中	線路運轉中輸入 ON	IN
1	QS	緊急停止	急停止時輸入 OFF	IN
2	DVDTF	加減速度標記	ON 輸入時，輸出目前加減速度 (DVDT) 乘以 1000 倍的值	IN
3~7	-	(備用)	輸入用備用繼電器	IN
8	ARY	加速中	加速中輸出 ON	OUT
9	BRY	減速中	減速中輸出 ON	OUT
A	LSP	零速	零速時輸出 ON	OUT
B	EQU	一致	輸入值 = 輸出值時 ON 輸出	OUT
C~F	-	輔助繼電器	系統內輔助繼電器	IN/OUT

(註) 急停止 (QS) 為 OFF 時，使用急停止時間 (QT) 作為減速時間。

## ◆ 實數型 SLAU 指令的參數表

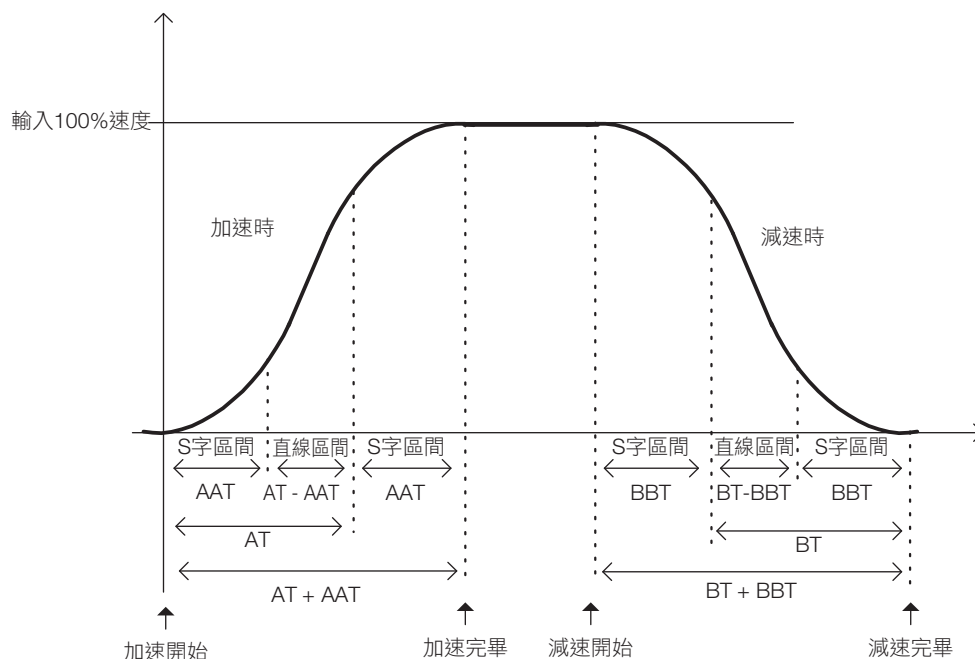
位址	資料類型	記號	名稱	規格	輸出輸入
0	W	RLY	繼電器輸入輸出	繼電器輸入，繼電器輸出*	IN/OUT
1	W	-	(備用)	備用暫存器	-
2	F	LV	輸入 100% 等級	輸入 100% 的光學尺	IN
4	F	AT	加速時間	0% ~ 100% 的加速時間 (s)	IN
6	F	BT	減速時間	100% ~ 0% 的減速時間 (s)	IN
8	F	QT	急停止時間	100% ~ 0% 的急停止時間 (s)	IN
10	F	AAT	加速 S 字時間	加速時的 S 型領域時間 (s)	IN
12	F	BBT	減速 S 字時間	減速時的 S 型領域時間 (s)	IN
14	F	V	當前速度	SLAU 輸出 (輸出至輸出速度)	OUT
16	F	DVDT1	當前加減速 1	輸出實際加減速度	OUT
18	F	ABMD	保持時速度上升	下達保持指令後到穩定時的速度變化量	OUT

\* 繼電器輸入輸出的分配如下所示。

Bit	記號	名稱	規格	輸出輸入
0	RN	線路運轉中	線路運轉中輸入 ON	IN
1	QS	緊急停止	緊急停止時輸入 OFF	IN
2 ~ 7	-	(備用)	輸入用備用繼電器	IN
8	ARY	加速中	加速中輸出 ON	OUT
9	BRY	減速中	減速中輸出 ON	OUT
A	LSP	零速	零速時輸出 ON	OUT
B	EQU	一致	輸入速度 = 以輸出速度輸出 ON	OUT
C ~ F	-	(備用)	輸出用備用繼電器	OUT

(註) 急停止 (QS) 為 OFF 時，使用急停止時間 (QT) 作為減速時間。

例 參數在實際指令當中的運用，如下所示。



(註) 關於 SLAU 指令的內部運算，請參照以下項目。

📖 補充事項 (第 4-201 頁)

補充

緊急停止 (QS) 設為 OFF，則在緊急停止時間 (QT) 減速，輸出速度變為 0。輸入速度則無須設為 0。

此外，緊急停止時不進行 S 型動作，而是直線減速。請設定為直線加減速時間 (AT/BT) ≥ 加減速 S 型時間 (AAT/BBT)。

## 程式範例

使用建立的加減速率，當輸入速度為 MF00000、輸出速度為 MF00002，執行實數型 SLAU 指令時的程式範例，如下所示。

下列是為建構加減速率所設定的參數。

- 加減速率的輸入 100% 等級速度 = 20000
- 加速時間 = 1.5 s
- 減速時間 = 2.5 s
- 急停止時間 = 0.5 s
- 加速 S 型時間 = 0.5 s
- 減速 S 型時間 = 1.0 s

EXPRESSION

```

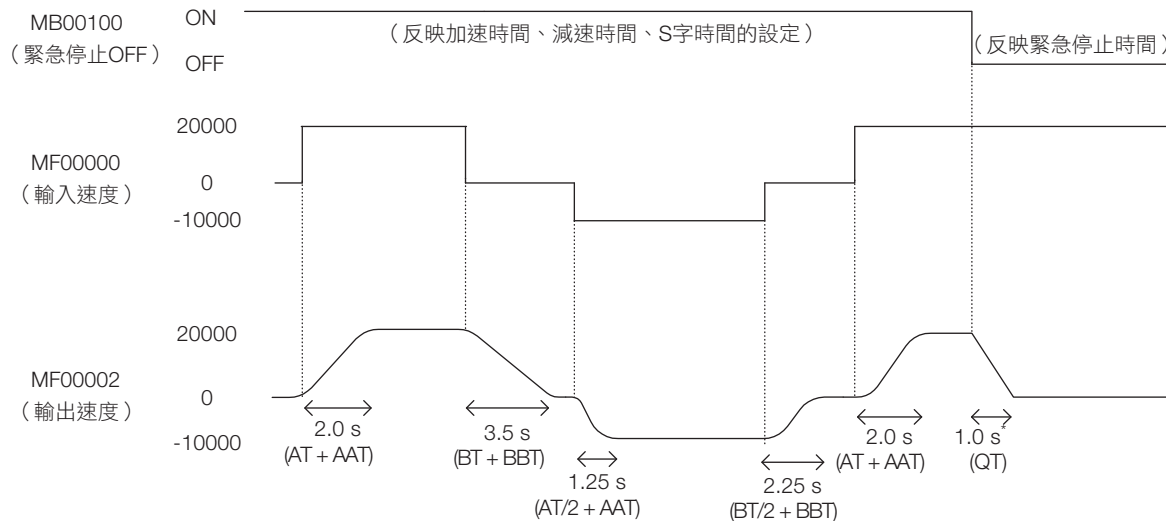
DF00002 = 20000; // LV: 入力100%レベル
2.000000E+004=20000
DF00004 = 1.5; // AT:加速時間(s)
1.500000E+000=1.500000E+000
DF00006 = 2.5; // BT:減速時間(s)
2.500000E+000=2.500000E+000
DF00008 = 1.0; // QT:急停止時間(s)
1.000000E+000=1.000000E+000
DF00010 = 0.5; // AAT:加速S字時間(s)
5.000000E-001=5.000000E-001
DF00012 = 1.0; // BBT:減速S字時間(s)
1.000000E+000=1.000000E+000
  
```

MB000100 急停止OFF OnCoil 常時ON

DB000001 急停止 DB000000 ライン運転中

SLAU [WLF] In MF00000 0.000E+000 [A] Prm DA00000 --- [WLF] Out MF00002 0.000E+000

各暫存器的動作如下所示。



\* 緊急停止輸入時將於緊急停止時間減速停止，與 S 型時間、輸入速度無關。



## 補充事項

### ◆ 計算速度輸出值及當前加減速的運算公式

以下說明求出速度輸出值（加速時、減速時、緊急停止時、S 型區段加速時、S 型區段減速時）以及目前加減速度的運算式。

#### ■ 整數型 SLAU 命令的運算

在整數型 SLAU 指令中，下列運算式將依據已設定參數求出速度輸出值（加速時、減速時、緊急停止時、S 型區段加速時、S 型區段減速時）以及目前加減速度。

運算公式的 V 為速度輸出值，V' 為前次速度輸出值，VI 為速度指令的輸入值，Ts 為掃描時間設定值。

##### • 加速時的速度輸出值

可使用下列運算公式求出加速時的速度輸出值。

S 字段落外 (ADVS > ADV) 且 VI > V' (V' ≥ 0)，則 V = V' + ADV。

S 字段落外 (ADVS > ADV) 且 VI < V' (V' ≤ 0)，則 V = V' - ADV。

$$\text{加速速率 (ADV)} = \frac{\text{LV} \times \text{Ts} (0.1 \text{ ms}) + \text{REM1}}{\text{AT} (0.1 \text{ s}) \times 1000}$$

##### • 減速時的速度輸出值

可使用下列運算公式求出減速時的速度輸出值。

S 字段落外 (BDVS > BDV) 且 VI > V' (V' < 0)，則 V = V' + BDV。

S 字段落外 (BDVS > BDV) 且 VI < V' (V' > 0)，則 V = V' - BDV。

$$\text{減速速率 (BDV)} = \frac{\text{LV} \times \text{Ts} (0.1 \text{ ms}) + \text{REM1}}{\text{BT} (0.1 \text{ s}) \times 1000}$$

##### • 急停止時的速度輸出值

可使用下列運算公式求出急停止時的速度輸出值。

QS = OFF (VI > V', V' < 0)，則 V = V' + QDV。

QS = OFF (VI < V', V' > 0)，則 V = V' - QDV。

$$\text{緊急停止速率 (QDV)} = \frac{\text{LV} \times \text{Ts} (0.1 \text{ ms}) + \text{REM1}}{\text{QT} (0.1 \text{ s}) \times 1000}$$

**補充** 急停止時並非呈現 S 形動作，而是以直線動作減速。

##### • S 形區間加速時的速度輸出值

可使用下列運算公式求出 S 形區間加速時的速度輸出值。

S 字段落內 (ADVS < ADV) 且 VI > V' (V' ≥ 0)，則 V = V' + ADVS。

S 字段落內 (ADVS < ADV) 且 VI < V' (V' ≤ 0)，則 V = V' - ADVS。

S 字段落加速率 (ADVS) = ADVS' ± AADVS

$$\text{AADVS} = \frac{\text{ADV} \times \text{Ts} (0.1 \text{ ms}) + \text{REM2}}{\text{AAT} (0.01 \text{ s}) \times 100}$$

- S 形區間減速時的速度輸出值

可使用下列運算公式求出 S 形區間減速時的速度輸出值。

若 S 字區間內 (BDVS < BDV) 為  $V > V'$  ( $V' < 0$ ) 時，則  $V = V' + BDVS$ 。

若 S 字區間內 (BDVS < BDV) 為  $V < V'$  ( $V' > 0$ ) 時，則  $V = V' - BDVS$ 。

S 字區間減速速率 (BDVS) =  $BDVS' \pm BBDVS$

$$BBDVS = \frac{BDV \times Ts (0.1ms) + REM2}{BBT (0.01 s) \times 100}$$

- 當前加減速

DVDT1 運算不執行 (DVDTF) 為 ON 時，目前加減速度 1 (DVDT1) 將遵照 DVDT1 運算選擇 (DVDTs)，以下列任一運算式求出目前加減速度 (DVDT)。當 DVDTF 為 OFF，DVDT1 變為 0。

$$DVDTs = ON, \text{ 則 } DVDT1 = \frac{(V - V') \times 5000}{ADV}$$

$$DVDTs = OFF, \text{ 則 } DVDT1 = (V - V') \times DVDTK$$

目前加減速度 2 (DVDT2) 則變為以下情況。

加速中：S 型區段內：DVDT2 =  $\pm ADVS$

S 型區段外：DVDT2 =  $\pm ADV$

減速中：S 型區段內：DVDT2 =  $\pm BDVS$

S 型區段外：DVDT2 =  $\pm BDV$

緊急停止中：DVDT =  $\pm QDV$

保持時速度上升部分 (ABMD) 經過如下運算後進行輸出。

$$ABMD = \frac{DVDT2' \times DVDT2'}{2 \times ADVS (BBDVS)}$$

DVDT2'：目前加減速度 2 (DVDT2) 的前一次值

**補充**

1. 加速中 (ARY) 的繼電器為 ON 的條件如下所示。
  - $V' \geq 0$  且  $ADV > 0$  時，或  $V' \leq 0$  且  $ADV < 0$  時
  - S 型區段內 當  $V' \geq 0$  且  $ADVS > 0$ ，或 S 型區段內  $V' \leq 0$  且  $ADVS < 0$  時
2. 減速中 (BRY) 的繼電器為 ON 的條件如下所示。
  - $V' < 0$  且  $BDV > 0$  時，或  $V' > 0$  且  $BDV < 0$  時
  - $V' < 0$  且  $QDV > 0$  時，或  $V' > 0$  且  $QDV < 0$  時
  - S 型區段內 當  $V' < 0$  且  $BDVS > 0$ ，或 S 型區段內  $V' > 0$  且  $BDVS < 0$  時
3. 零速 (LSP) 繼電器變為 ON 的條件為  $V = 0$ ，
4. 一致 (EQU) 的繼電器為 ON 的條件為  $V_I = V$ 。
5. 當直線運轉中 (RN) 為 OFF 時，輸出  $V = 0$ 、 $DVDT1$ 、 $DVDT2$ 、 $DVDT3 = 0$ 、 $REM1$ ， $REM2$ ， $REM3 = 0$ 。

## ■ 2 倍長整數型／實數型 SLAU 命令的運算

在 2 倍長整數型／實數型 SLAU 指令中，將由下列運算式求出速度輸出值（加速時、減速時、緊急停止時、S 型區段加速時、S 型區段減速時）以及目前加減速度。

運算式中的 V 為速度輸出值，V' 為前次速度輸出值，VI 為速度指令輸入值，Ts 為掃描時間設定值，ADVS' 為 ADVS 前一次的值，BDVS' 為 BDVS 前一次的值。

### • 加速時的速度輸出值

可使用下列運算公式求出加速時的速度輸出值。

S 字段落外 (ADVS > ADV) 且 VI > V' (V' ≥ 0)，則 V = V' + ADV。

S 字段落外 (ADVS > ADV) 且 VI < V' (V' ≤ 0)，則 V = V' - ADV。

$$\text{加速速率 (ADV)} = \frac{\text{LV} \times \text{Ts} (0.1\text{ms})}{\text{AT} (\text{s}) \times 10000}$$

### • 減速時的速度輸出值

可使用下列運算公式求出減速時的速度輸出值。

S 字段落外 (BDVS > BDV) 且 VI > V' (V' < 0)，則 V = V' + BDV。

S 字段落外 (BDVS > BDV) 且 VI < V' (V' > 0)，則 V = V' - BDV。

$$\text{減速速率 (BDV)} = \frac{-\text{LV} \times \text{Ts} (0.1\text{ms})}{\text{BT} (\text{s}) \times 10000}$$

### • 急停止時的速度輸出值

可使用下列運算公式求出急停止時的速度輸出值。

QS = OFF (VI > V', V' < 0)，則 V = V' + QDV。

QS = OFF (VI < V', V' > 0)，則 V = V' - QDV。

$$\text{緊急停止速率 (QDV)} = \frac{-\text{LV} \times \text{Ts} (0.1\text{ms})}{\text{QT} (\text{s}) \times 10000}$$

#### 補充

急停止時並非呈現 S 形動作，而是以直線動作減速。

### • S 形區間加速時的速度輸出值

可使用下列運算公式求出 S 形區間加速時的速度輸出值。

S 字段落內 (ADVS < ADV) 且 VI > V' (V' ≥ 0)，則 V = V' + ADVS。

S 字段落內 (ADVS < ADV) 且 VI < V' (V' ≤ 0)，則 V = V' - ADVS。

S 字段落加速率 (ADVS) = ADVS' ± AADVS

$$\text{AADVS} = \frac{\text{ADV} \times \text{Ts} (0.1\text{ms})}{\text{AAT} (\text{s}) \times 10000}$$

### • S 形區間減速時的速度輸出值

可使用下列運算公式求出 S 形區間減速時的速度輸出值。

若 S 字區間內 (BDVS < BDV) 為 VI > V' (V' < 0) 時，則 V = V' + BDVS。

若 S 字區間內 (BDVS < BDV) 為 VI < V' (V' > 0) 時，則 V = V' - BDVS。

S 字區間減速速率 (BDVS) = BDVS' ± BBDVS

$$\text{BBDVS} = \frac{\text{BDV} \times \text{Ts} (0.1\text{ms})}{\text{BBT} (\text{s}) \times 10000}$$

- 當前加減速

目前加減速度 1 (DVDT1) 於進行下列運算後輸出。

加速中：S 型區段內：DVDT = ADVS

S 型區段外：DVDT = ADV

減速中：S 型區段內：DVDT = BDVS

S 型區段外：DVDT = BDV

緊急停止中：DVDT = QDV

保持時速度上升部分 (ABMD) 經過下列運算後進行輸出。

$$ABMD = \frac{DVDT \times DVDT}{2 \times AADVS(BBDVS)}$$

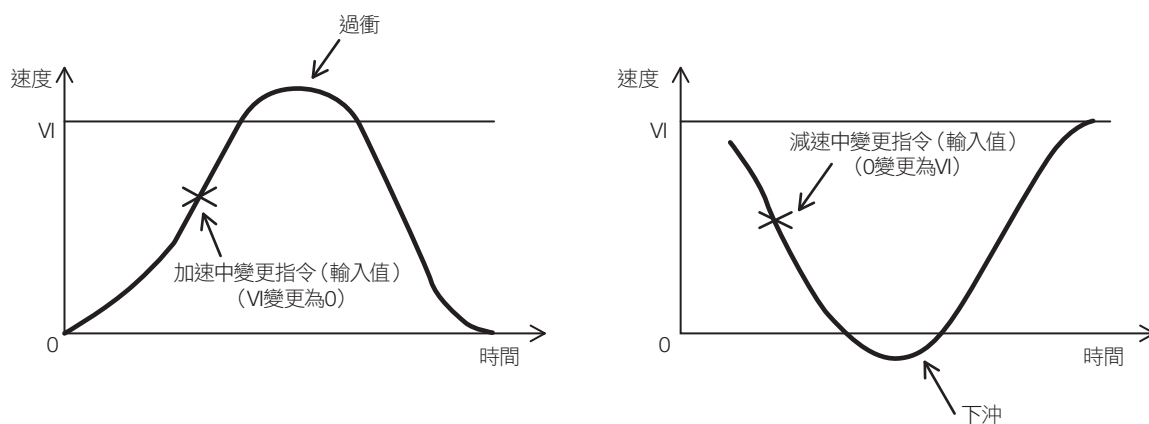
**補充**

1. 零速 (LSP) 的繼電器為 ON 的條件為  $V = 0$ 。
2. 一致 (EQU) 的繼電器為 ON 的條件為  $VI = V$ 。
3. 當直線運轉中 (RN) 為 OFF 時，輸出  $V = 0$ 、 $DVDT = 0$ 、 $AVMD = 0$ 。

### ◆ 使用 SLAU 指令時的注意事項

#### ■ 加減速中對 [ 到達輸入速度 (VI) 前 ] 變更輸入值時

加減速中變更輸入值 (VI) 時，請勿使用整數型 SLAU 指令。有可能如下圖般發生過衝/下衝。

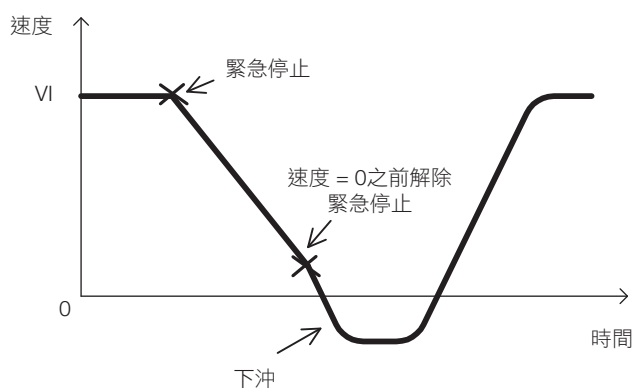


加減速中要變更輸入值 (VI) 時，請使用應用程式採取下列任一對策。

- 請使用實數型 SLAU 指令。
- 實數型 SLAU 指令請併用 LIMIT 指令。也就是說，請將整數型 SLAU 指令的輸出值作為 LIMIT 指令的輸入值，限制過衝/下衝。

### ■ 在緊急停止減速中解除緊急停止時

緊急停止減速中，在輸出速度未變為 0 之前，請勿解除緊急停止。有可能在朝向輸入速度的中途發生下衝。



如逼不得已必須在輸出速度變為 0 之前解除緊急停止，請對輸出速度使用 LIMIT 指令，避免下衝發生。

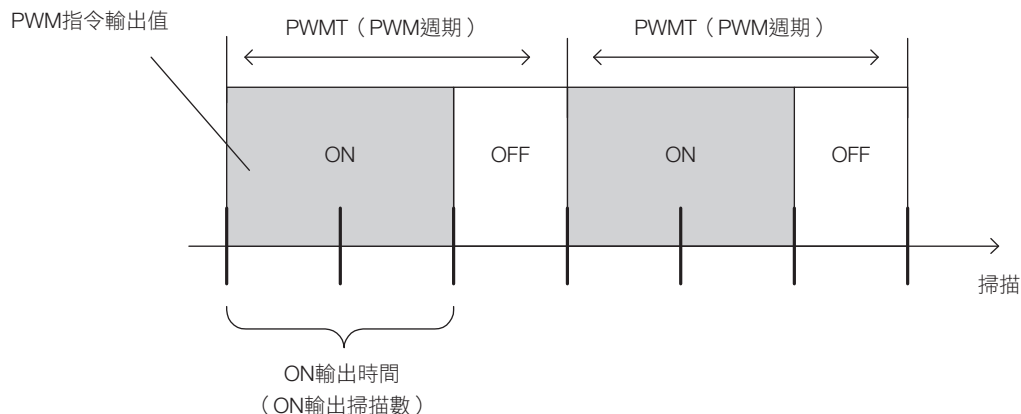
## 脈衝寬變調 (PWM)

將輸入值 (-100.00 ~ 100.00%) 轉換為 PMW，將其結果輸出至輸出值以及參數表。輸入值僅能使用整數型，輸出值僅能使用位元型。無法使用 2 倍長整數型以及實數型。



重要

以整數型使用時，請將掃描時間設定值設為 1 ms 的整數倍。



PWM 指令的 ON 輸出時間以及 ON 輸出掃描數。可由下列式子求出。

X 為輸入值，PWMT 為 PWM 週期 (ms)，Ts 為掃描時間設定值 (ms)。

$$\text{ON輸出時間} = \frac{\text{PWMT} (X + 10000)}{20000}$$

$$\text{ON輸出掃描數} = \frac{\text{PWMT} (X + 10000)}{T_s \times 20000}$$

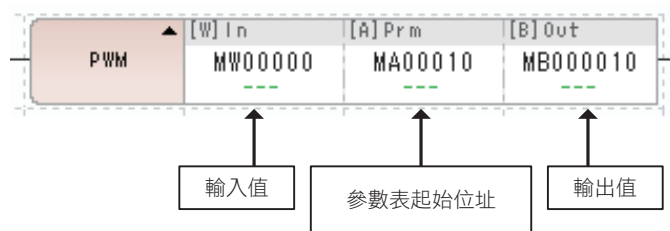


註記

- 輸入值與 PWM 輸出 ON 比例的關係如下。
  - 輸入值 100.00% → 100% ON (ON 輸出時間 = PWMT)
  - 輸入值 0.00% → 50% ON (ON 輸出時間 = PWMT/2)
  - 輸入值 -100.00% → 0% ON (ON 輸出時間 = 0)
- 通電後，請先令 PWM 重置 (PWMRST) 為 ON 以清除內部運算，再使用 PWM 指令。以 PWM 重置當時為開始點，進行 PMW 運算。

## 格式

格式如下所示。



圖示： PWM

鍵輸入： PWM

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸入值 (In)	×	○	×	×	×	×	×	○	○
參數表起始位址 (Prm)	×	×	×	×	×	×	○*	○*	×
輸出值 (Out)	○*	×	×	×	×	×	×	○	×

\* C、# 暫存器除外

### ◆ 輸入值／輸出值的範圍

輸入值的範圍為  $-10000 \sim 10000$  (單位: 0.01%)。

範圍外時，以上限值 10000、下限值  $-10000$  進行處理。

輸出值在 PWM 輸出 ON 時為 1，輸出 OFF 時為 0。

### ◆ 參數表

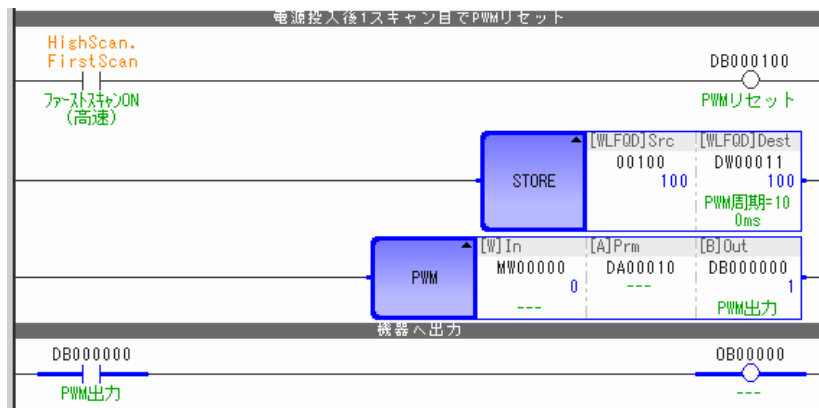
位址	資料類型	記號	名稱	規格	輸出輸入
0	W	RLY	繼電器輸入輸出	繼電器輸入、繼電器輸出 <sup>*1</sup>	IN/OUT
1	W	RWMT	PWM 週期	PWM 週期 (1 ms) 範圍: 1 ~ 32767 ms	IN
2	W	ONCNT	ON 輸出設定計時器	ON 輸出設定計時器 (1 ms)	OUT
3	W	CVON	ON 輸出計數計時器	ON 輸出計數計時器 (1 ms)	OUT
4	W	CVONREM	ON 輸出計數計時器餘數	ON 輸出計數計時器餘數 (0.1 ms)	OUT
5	W	OFFCNT	OFF 輸出設定計時器	OFF 輸出設定計時器 (1 ms)	OUT
6	W	CVOFF	OFF 輸出計數計時器	OFF 輸出計數計時器 (1 ms)	OUT
7	W	CVOFFREM	OFF 輸出計數計時器餘數	OFF 輸出計數計時器餘數 (0.1 ms)	OUT

\* 繼電器輸入輸出的分配如下所示。

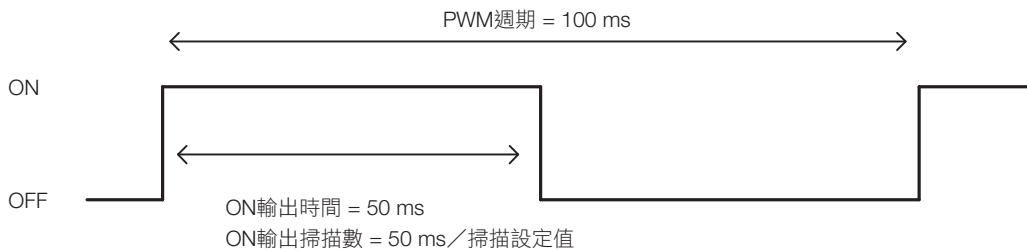
Bit	記號	名稱	規格	輸出輸入
0	PWMRST	PWM 重置	PWM 重置時輸入 ON	IN
2 ~ 7	-	(備用)	輸入用備用繼電器	IN
8	PWMOUT	PWM 輸出	PWM 的輸出 (輸出 ON 時為 1，輸出 OFF 時為 0)	OUT
9 ~ F	-	(備用)	輸出用備用繼電器	OUT

### 程式範例

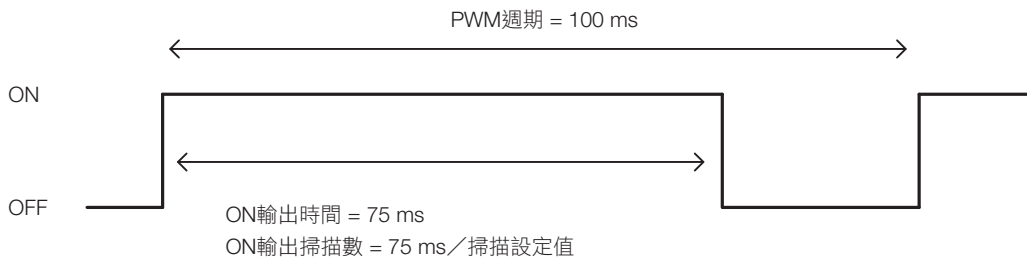
當 PWM 週期為 100 ms，對輸入值 (MW00000) 儲存 PWM 輸出至 OB000000 的程式範例，如下所示。



MW00000 為 0 (0% : ON 輸出時間 = PWM 週期 / 2) 時的 OB000000 將會如下。



MW00000 為 5000 (50% : ON 輸出時間 = PWM 週期 × 3/4) 時的 OB000000 將會如下。





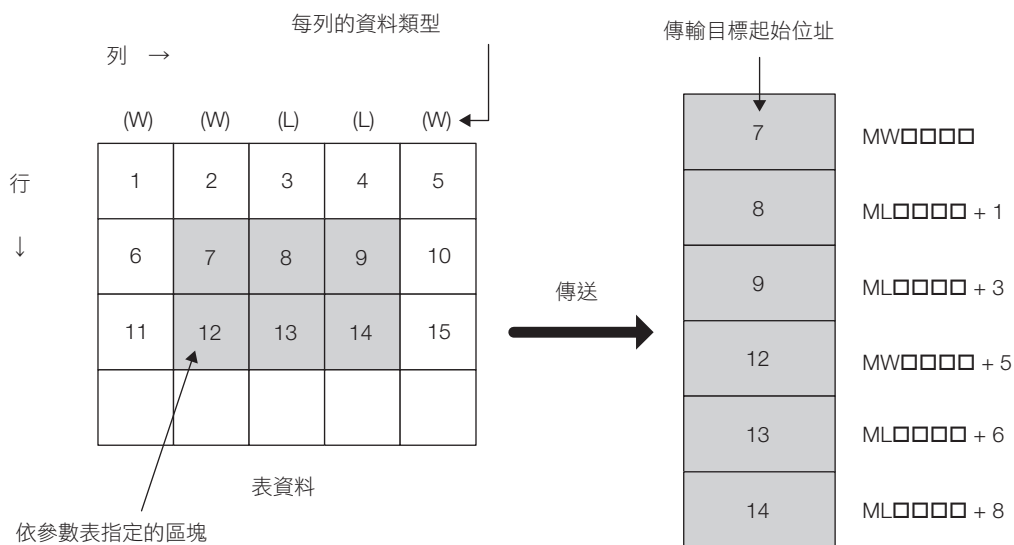
# 4.9 表操作指令

## 區段讀取 (TBLBR)

將表名稱、行編號、列編號指定的表資料區段，傳送至傳送對象開頭位址起的連續區域。依據讀取要素的資料類型，儲存至傳送對象區域。

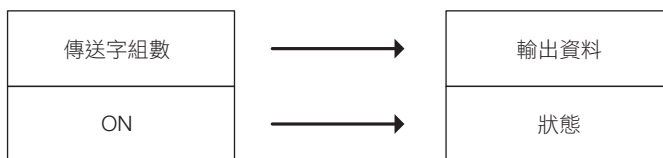
引用表格時，當發生範圍外、傳送對象資料長度不足等錯誤時，將會報告錯誤，不讀取資料，維持傳送對象區域的原本內容。

正常結束時，輸出傳輸字組數，狀態設為 OFF。發生錯誤時，輸出錯誤代碼，狀態設為 ON。

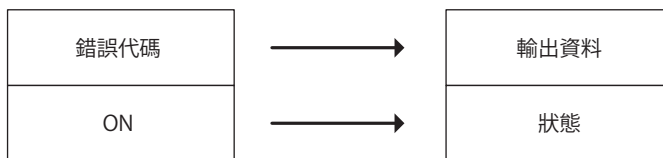


配合表資料的資料類型進行存入。

• 傳輸成功時



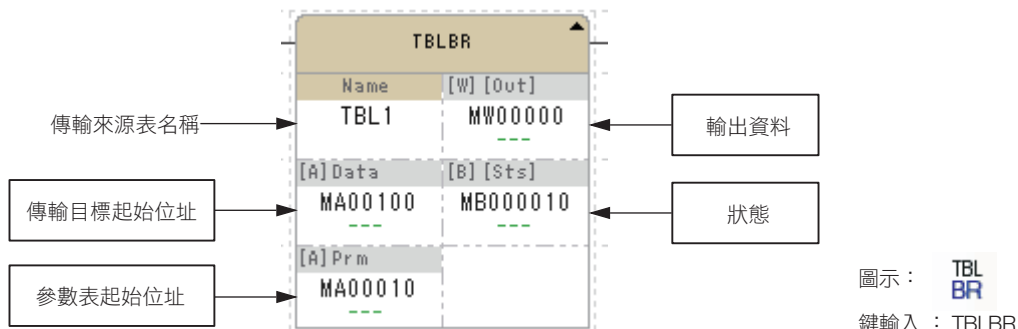
• 傳輸失敗時



**補充** 傳送失敗時，傳送對象區域將保持指令執行之前的狀態。

### 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
傳送對象開頭位址 (Data)	×	×	×	×	×	×	O <sup>*2</sup>	×	×
參數表起始位址 (Prm)	×	×	×	×	×	×	O	×	×
輸出資料 (Out) <sup>*1</sup>	×	O <sup>*2</sup>	×	×	×	×	×	O	×
狀態 (Sts) <sup>*1</sup>	O <sup>*2</sup>	×	×	×	×	×	×	×	×

\*1. 可以省略  
\*2. C、# 暫存器除外

### ◆ 參數表

位址	資料類型	記號	名稱	規格	輸出輸入
0	L	ROW1	表元素開始行編號	對象之表元素開始行編號 (1 ~ 65535)	IN
2	L	COL1	表元素開始列編號	對象之表元素開始列編號 (1 ~ 32767)	IN
4	W	RLEN	行元素個數	行元素個數 (1 ~ 32767)	IN
5	W	CLEN	列元素個數	列元素個數 (1 ~ 32767)	IN

### ◆ 錯誤代碼一覽

錯誤代碼	錯誤名稱	內容
0001 H	未定義對照表	未定義目標表。
0002 H	行編號範圍外	表元素的行編號不在目標表範圍內。
0003 H	列編號範圍外	表元素的列編號不在目標表範圍內。
0004 H	元素個數不良	目標元素的個數不恰當。
0005 H	儲存目的地空間不足	儲存空間不足。
0006 H	元素型不足	所指定的元素型異常。
0007 H	佇列緩衝區錯誤	當佇列緩衝區為空白時執行讀取，或佇列緩衝區已滿時以指標步進執行寫入。
0008 H	佇列表錯誤	指定之表並非佇列表。
0009 H	系統錯誤	執行指令時檢測到系統內部產生非預期的錯誤。

(註) 錯誤代碼為所有表操作指令共通。

## 程式範例

開關 1 (DB00000) ON 時，傳送記錄型表資料 TBL1 指定的區段至參數表開頭位址 (MW00100) 起的連續區域的程式範例，如下所示。

參數表請設定如下。

暫存器	數據	備註
DL00000	2	表元素開始行編號
DL00002	2	表元素開始列編號
DW00004	3	行元素個數
DW00005	3	列元素個數

表資料 TBL1 內容如下所示。

行	列				
	1 (W)*	2 (W)*	3 (L)*	4 (L)*	5 (F)*
1	1000	1001	10000	10001	1.1
2	2000	2002	20000	20002	1.2
3	3000	3003	30000	30003	1.3
4	4000	4004	40000	40004	1.4
5	5000	5005	50000	50005	1.5

傳輸區域

\* 顯示資料類型。

指令執行後，資料將傳送至從 MW00100 開始的連續區域，如下所示。

此外，輸出資料為 (MW00000)15 (字組數)，狀態 (MB000010) 為 0 (傳輸成功)。

暫存器	數據	暫存器	數據	暫存器	數據
MW00100	2002	ML00101	20000	ML00103	20002
MW00105	3003	ML00106	30000	ML00108	30003
MW00110	4004	ML00111	40000	ML00113	40004

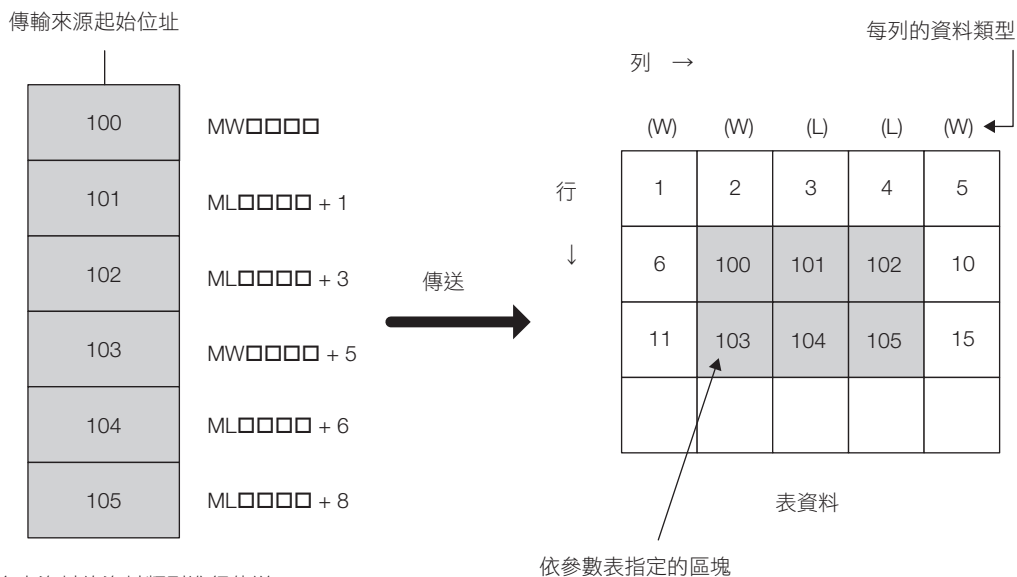
(註) 暫存器與前述表格採相同配置。

## 區段寫入 (TBLBW)

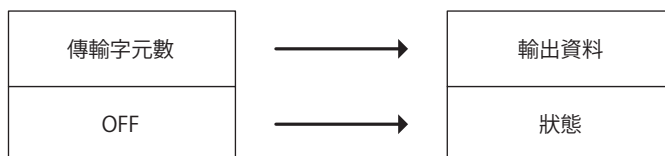
將傳送來源開頭位址起的連續區域資料，傳送至表名稱、行編號、列編號指定的表資料區段。傳送來源區域與表資料各要素的資料類型視為一致，進行儲存。

引用表格時，當發生範圍外、傳送來源資料長度不足等錯誤時，將會報告錯誤，不寫入資料，維持傳送對象區域的原本內容。

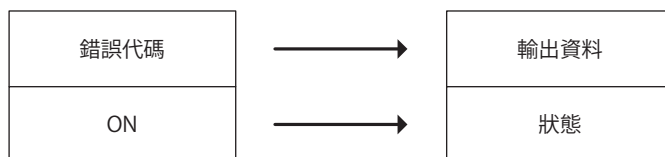
正常結束時，輸出傳輸字組數，狀態設為 OFF。發生錯誤時，輸出錯誤代碼，狀態設為 ON。



• 傳輸成功時



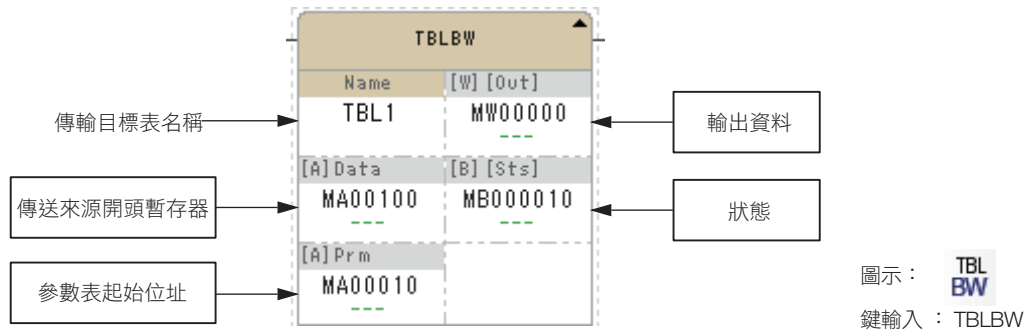
• 傳輸失敗時



**補充** 傳送失敗時，傳送對象表資料將保持指令]執行之前的狀態。

## 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
傳送來源開頭位址 (Data)	×	×	×	×	×	×	○ <sup>*2</sup>	×	×
參數表起始位址 (Prm)	×	×	×	×	×	×	○	×	×
輸出資料 (Out) <sup>*1</sup>	×	○ <sup>*2</sup>	×	×	×	×	×	○	×
狀態 (Sts) <sup>*1</sup>	○ <sup>*2</sup>	×	×	×	×	×	×	×	×

\*1. 可以省略

\*2. C、# 暫存器除外

### ◆ 參數表

位址	資料類型	記號	名稱	規格	輸出輸入
0	L	ROW1	表元素開始行編號	對象之表元素開始行編號 (1 ~ 65535)	IN
2	L	COL1	表元素開始列編號	對象之表元素開始列編號 (1 ~ 32767)	IN
4	W	RLEN	行元素個數	行元素個數 (1 ~ 32767)	IN
5	W	CLEN	列元素個數	列元素個數 (1 ~ 32767)	IN

### ◆ 錯誤代碼一覽

錯誤代碼	錯誤名稱	內容
0001 H	未定義對照表	未定義目標表。
0002 H	行編號範圍外	表元素的行編號不在目標表範圍內。
0003 H	列編號範圍外	表元素的列編號不在目標表範圍內。
0004 H	元素個數不良	目標元素的個數不恰當。
0005 H	儲存目的地空間不足	儲存空間不足。
0006 H	元素型不足	所指定的元素型異常。
0007 H	佇列緩衝區錯誤	當佇列緩衝區為空白時執行讀取，或佇列緩衝區已滿時以指標步進執行寫入。
0008 H	佇列表錯誤	指定之表並非佇列表。
0009 H	系統錯誤	執行指令時檢測到系統內部產生非預期的錯誤。

(註) 錯誤代碼為所有表操作指令共通。

### 程式範例

開關 1 (DB00000) ON 時，傳送參數表開頭位址 (MW00100) 起的連續區域資料至記錄型表資料 TBL1 指定的區段的程式範例，如下所示。

參數表請設定如下。

暫存器	數據	備註
DL00000	2	表元素開始行編號
DL00002	2	表元素開始列編號
DW00004	3	行元素個數
DW00005	3	列元素個數

傳送的資料如下。

暫存器	數據	暫存器	數據	暫存器	數據
MW00100	1	ML00101	2	ML00103	3
MW00105	4	ML00106	5	ML00108	6
MW00110	7	ML00111	8	ML00113	9



指令執行後的表資料 TBL1 內容如下。

此外，輸出資料為 (MW00000)15 (字組數)，狀態 (MB000010) 為 0 (傳輸成功)。

行	列				
	1 (W)*	2 (W)*	3 (L)*	4 (L)*	5 (F)*
1					
2		1	2	3	
3		4	5	6	
4		7	8	9	
5					

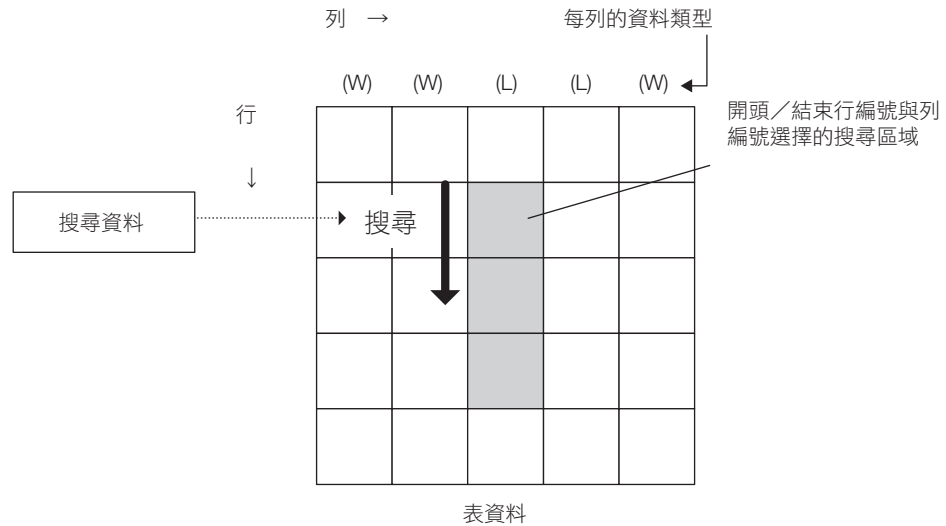
A callout box labeled '傳輸區域' (Transfer Area) points to the red-bordered cells containing the data values 1, 2, 3, 4, 5, 6, 7, 8, and 9.

\* 顯示資料類型。

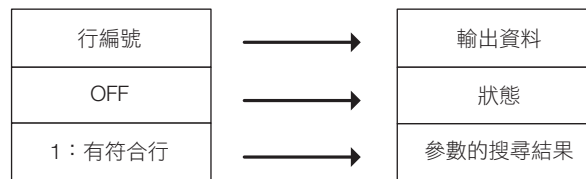
## 行搜尋（縱向）（TBL SRL）

搜尋表名稱、行編號、列編號指定的表資料列要素，當發現與搜尋資料一致的資料時，向輸出資料報告其行編號。搜尋對象的資料類型，將基於指定列要素資料類型自動判斷。

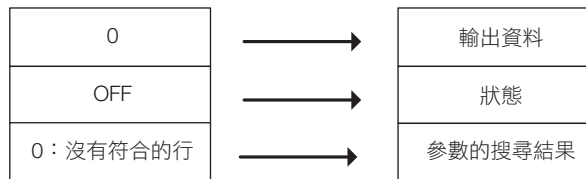
正常結束時，如發現一致資料，則輸入的參數表搜尋結果設為 1，輸出資料設定行編號，狀態設為 OFF。如無一致資料，則搜尋結果與輸出資料均設 0。發生錯誤時，於輸出資料設定錯誤代碼，狀態設為 ON。



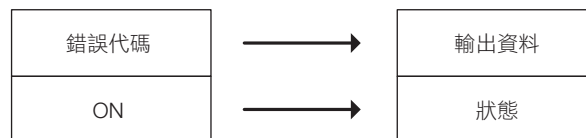
- 有一致的資料



- 無一致的資料

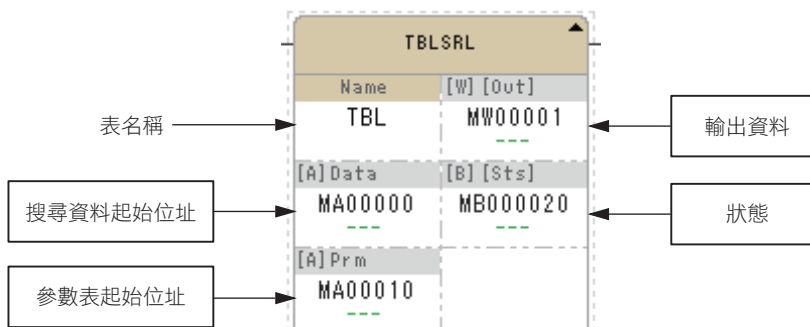


- 發生錯誤時



### 格式

格式如下所示。



圖示：  
TBL SRL  
鍵輸入：TBL SRL

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
搜尋資料起始位址 (Data)	×	×	×	×	×	×	○	×	×
參數表起始位址 (Prm)	×	×	×	×	×	×	○	×	×
輸出資料 (Out) <sup>*1</sup>	×	○ <sup>*2</sup>	×	×	×	×	×	○	×
狀態 (Sts) (W) <sup>*1</sup>	○ <sup>*2</sup>	×	×	×	×	×	×	×	×

\*1. 可以省略  
\*2. C、# 暫存器除外

### ◆ 參數表

位址	資料類型	記號	名稱	規格	輸出輸入
0	L	ROW1	表元素第一行編號	作為對象的表元素開頭行編號 (1 ~ 65535)	IN
2	L	ROW2	表元素最尾行編號	作為對象的表元素最終行編號 (1 ~ 65535)	IN
4	L	COLUMN	表元素列編號	作為對象的表元素列編號 (1 ~ 32767)	IN
6	W	FIND	搜尋結果	搜尋結果 0：沒有符合的行 1：有符合的行	OUT

### ◆ 錯誤代碼一覽

錯誤代碼	錯誤名稱	內容
0001 H	未定義對照表	未定義目標表。
0002 H	行編號範圍外	表元素的行編號不在目標表範圍內。
0003 H	列編號範圍外	表元素的列編號不在目標表範圍內。
0004 H	元素個數不良	目標元素的個數不恰當。
0005 H	儲存目的地空間不足	儲存空間不足。
0006 H	元素型不足	所指定的元素型異常。
0007 H	佇列緩衝區錯誤	當佇列緩衝區為空白時執行讀取，或佇列緩衝區已滿時以指標步進執行寫入。
0008 H	佇列表錯誤	指定之表並非佇列表。
0009 H	系統錯誤	執行指令時檢測到系統內部產生非預期的錯誤。



## 程式範例

由陣列型表資料 TBL1 循表的縱向進行搜尋資料（MW00000）32 的搜尋時的程式範例，如下所示。

參數表請設定如下。

暫存器	數據	備註
DL00010	2	表元素第一行編號
DL00012	5	表元素最尾行編號
DL00014	2	表元素列編號

表資料 TBL1 內容如下所示。（表元素為整數型）

行	列				
	1 (W)*	2 (W)*	3 (W)*	4 (W)*	5 (W)*
1	11	12	13	14	15
2	21	22	23	24	25
3	31	32	33	34	35
4	41	42	43	44	45
5	51	52	53	54	55

\* 顯示資料類型。

搜尋區域

32 與搜尋區域中的行編號 3 一致，因此輸出 3 至輸出資料（DW00001）。

The screenshot shows a ladder logic program with two main components:

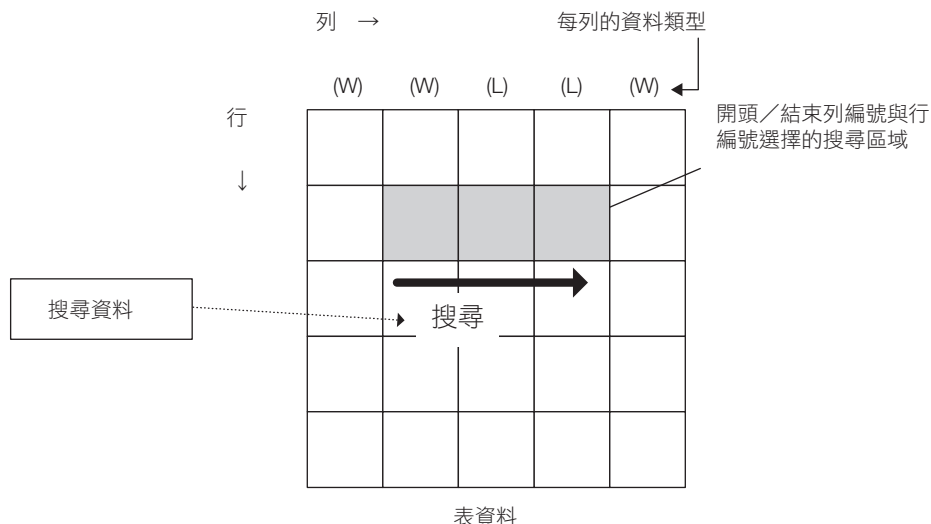
- EXPRESSION:** A block containing the following logic:
 

```
DL00010 = 2; DL00012 = 5; // 開始・最終行番号
2 = 2; 5 = 5
DL00014 = 2; // 列番号
2 = 2
MW00000 = 32; // 検索データ
32 = 32
```
- TBLSRC:** A block representing the search results for table TBL1. It shows:
  - Name: TBL1
  - [W] [Out]: DW00001
  - Value: 3
  - [A] Data: MA00000
  - [B] [Sts]: DB000000
  - Label: PWM出力
  - [A] Prm: DA00010

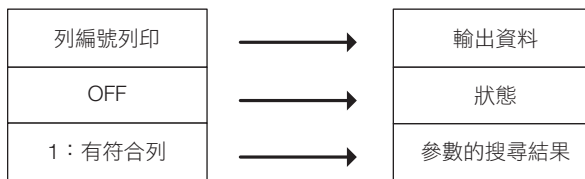
## 列搜尋（橫向）(TBLSRC)

搜尋表名稱、行編號、列編號指定的表資料行要素，當發現與搜尋資料一致的資料時，向輸出資料報告其列編號。搜尋對象的資料類型，將基於指定行要素資料類型自動判斷。

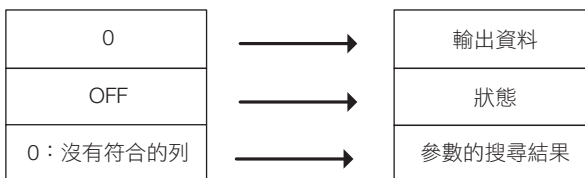
正常結束時，如發現一致資料，則輸入的參數表搜尋結果設為 1，輸出資料設定列編號，狀態設為 OFF。如無一致資料，則搜尋結果與輸出資料均設 0。發生錯誤時，於輸出資料設定錯誤代碼，狀態設為 ON。



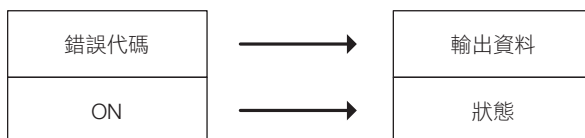
• 有一致的資料



• 無一致的資料

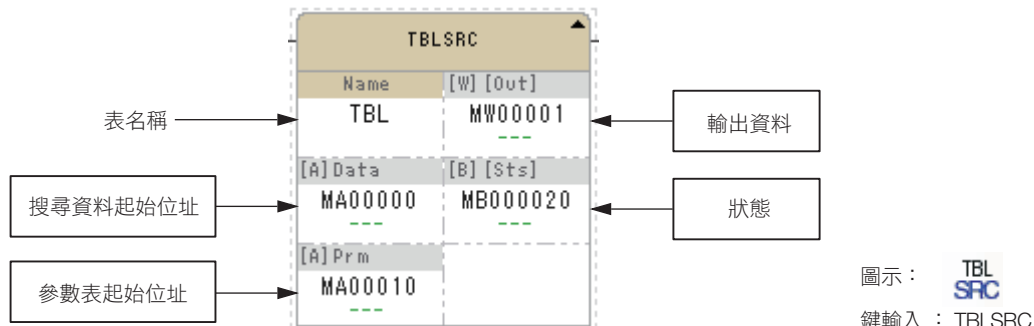


• 發生錯誤時



## 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
搜尋資料起始位址 (Data)	×	×	×	×	×	×	○	×	×
參數表起始位址 (Prm)	×	×	×	×	×	×	○	×	×
輸出資料 (Out) <sup>*1</sup>	×	○ <sup>*2</sup>	×	×	×	×	×	○	×
狀態 (Sts) <sup>*1</sup>	○ <sup>*2</sup>	×	×	×	×	×	×	×	×

\*1. 可以省略

\*2. C、# 暫存器除外

### ◆ 參數表

位址	資料類型	記號	名稱	規格	輸出輸入
0	L	ROW1	表元素行編號	作為對象的表元素行編號（1～65535）	IN
2	L	COL-UMN1	表元素第一列編號	作為對象的表元素開頭列編號（1～32767）	IN
4	L	COL-UMN2	表元素最尾列編號	作為對象的表元素最終列編號（1～32767）	IN
6	W	FIND	搜尋結果	搜尋結果 0：沒有符合的列 1：有符合的列	OUT

### ◆ 錯誤代碼一覽

錯誤代碼	錯誤名稱	內容
0001 H	未定義對照表	未定義目標表。
0002 H	行編號範圍外	表元素的行編號不在目標表範圍內。
0003 H	列編號範圍外	表元素的列編號不在目標表範圍內。
0004 H	元素個數不良	目標元素的個數不恰當。
0005 H	儲存目的地空間不足	儲存空間不足。
0006 H	元素型不足	指定要素的資料類型異常。
0007 H	佇列緩衝區錯誤	當佇列緩衝區為空白時執行讀取，或佇列緩衝區已滿時以指標步進執行寫入。
0008 H	佇列表錯誤	指定之表並非佇列表。
0009 H	系統錯誤	執行指令時檢測到系統內部產生非預期的錯誤。

(註) 錯誤代碼為所有表操作指令共通。

## 程式範例

由陣列型表資料 TBL1 循表的橫向進行搜尋資料（MW00000）34 的搜尋時的程式範例，如下所示。

參數表請設定如下。

暫存器	數據	備註
DL00010	3	表元素行編號
DL00012	2	表元素第一列編號
DL00014	5	表元素最尾列編號

表資料 TBL1 內容如下所示。（表元素為整數型）

行	列				
	1 (W)	2 (W)	3 (W)	4 (W)	5 (W)
1	11	12	13	14	15
2	21	22	23	24	25
3	31	32	33	34	35
4	41	42	43	44	45
5	51	52	53	54	55

搜尋區域

\* 顯示資料類型。

34 與搜尋區域中的行編號 4 一致，因此輸出 4 至輸出資料（DW00001）。

The screenshot shows a debugger interface with two main components:

- EXPRESSION window:** Contains the following code:
 

```
DL00010 = 3; // 行番号;
3=3
DL00012 = 2; DL00014 = 5; // 開始・最終列番号;
2=2; 5=5
MW00000 = 34; // 検索データ
34=34
```
- TBLSRC window:** Shows the configuration for the TBLSRC instruction:
 

Name	[W] [Out]
TBL1	DW00001
	4
[A] Data	[B] [Sts]
MA00000	DB000000
	0
[A] Pr m	
DA00010	

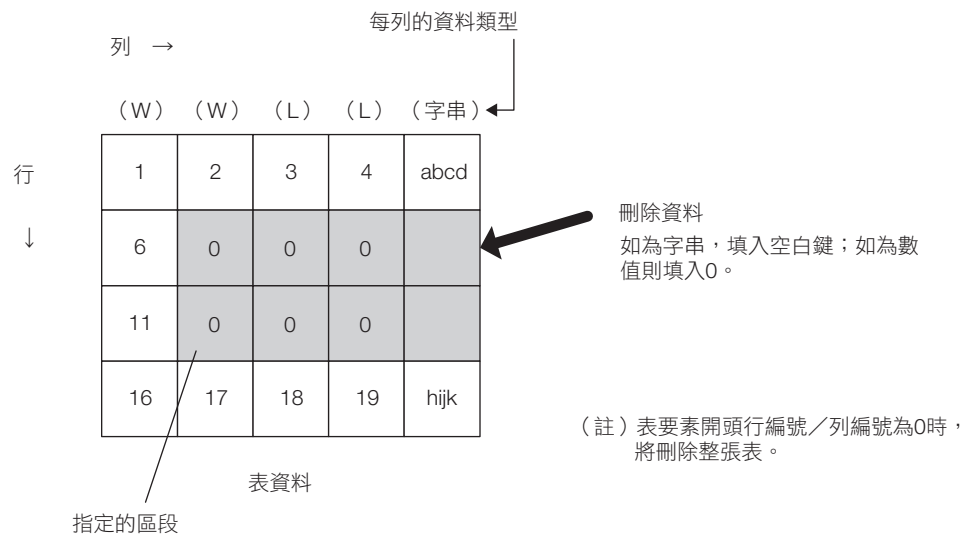
## 刪除區段 (TBLCL)

將刪除表名稱、行編號、列編號指定的表資料區段資料。表要素資料類型為字串時填入空白，為數值時填入 0。

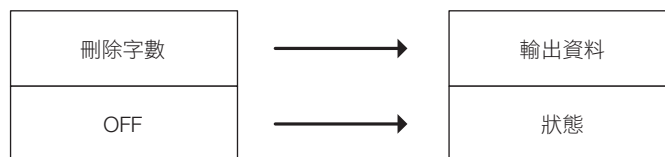
表要素開頭行編號、表要素開頭列編號均為 0 時，刪除整張表。

引用表格時，當發生範圍外、傳送資料長度不足等錯誤時，將會報告錯誤，不寫入資料。

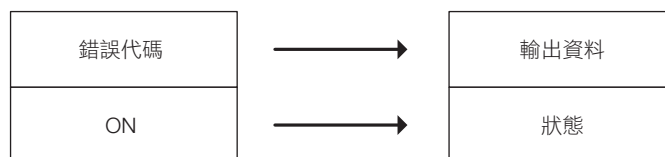
正常結束時，對輸出資料設定刪除字數，設狀態為 OFF。發生錯誤時，於輸出資料設定錯誤代碼，狀態設為 ON。



### • 刪除成功時



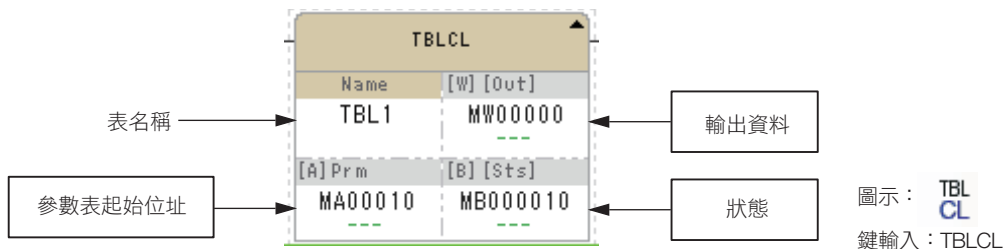
### • 刪除失敗時



**補充** 刪除失敗時，資料將保持指令執行之前的狀態。

### 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
參數表起始位址 (Prm) <sup>*1</sup>	×	×	×	×	×	×	○	×	×
輸出資料 (Out) <sup>*1</sup>	×	○ <sup>*2</sup>	×	×	×	×	×	○	×
狀態 (Sts) <sup>*1</sup>	○ <sup>*2</sup>	×	×	×	×	×	×	×	×

\*1. 可以省略  
\*2. C、# 暫存器除外

### ◆ 參數表

位址	資料類型	記號	名稱	規格	輸出輸入
0	L	ROW	表元素開始行編號	對象之表元素開始行編號 (1 ~ 65535)	IN
2	L	COL	表元素開始列編號	對象之表元素開始列編號 (1 ~ 32767)	IN
4	W	RLEN	行元素個數	行元素個數 (1 ~ 32767)	IN
5	W	CLEN	列元素個數	列元素個數 (1 ~ 32767)	IN

### ◆ 錯誤代碼一覽

錯誤代碼	錯誤名稱	內容
0001 H	未定義對照表	未定義目標表。
0002 H	行編號範圍外	表元素的行編號不在目標表範圍內。
0003 H	列編號範圍外	表元素的列編號不在目標表範圍內。
0004 H	元素個數不良	目標元素的個數不恰當。
0005 H	儲存目的地空間不足	儲存空間不足。
0006 H	元素型不足	所指定的元素型異常。
0007 H	佇列緩衝區錯誤	當佇列緩衝區為空白時執行讀取，或佇列緩衝區已滿時以指標步進執行寫入。
0008 H	佇列表錯誤	指定之表並非佇列表。
0009 H	系統錯誤	執行指令時檢測到系統內部產生非預期的錯誤。

(註) 錯誤代碼為所有表操作指令共通。

## 程式範例

開關 1 (DB000100) ON 時，由記錄型表資料 TBL1 刪除指定區段的程式範例，如下所示。

參數表請設定如下。

暫存器	數據	備註
DL00000	2	表元素開始行編號
DL00002	2	表元素開始列編號
DW00004	3	行元素個數
DW00005	3	列元素個數

表資料 TBL1 內容如下所示。

行	列				
	1 (W)*	2 (W)*	3 (L)*	4 (文字列)*	5 (F)*
1	1000	1001	10000	ABCD	1.1
2	2000	2002	20000	BCDE	1.2
3	3000	3003	30000	CDEF	1.3
4	4000	4004	40000	DEFG	1.4
5	5000	5005	50000	EFGH	1.5

刪除的區域

\* 顯示資料類型。

The screenshot shows a ladder logic program with the following components:

- EXPRESSION:**

```
DL00000 = 2; DL00002 = 2; // 表要素開始行・列番号
2=2; 2=2
DW00004 = 3; // 行要素數
3=3
DW00005 = 3; // 列要素數
3=3
```
- DB000100:** A switch labeled "スイッチ1" (Switch 1).
- TBLCL Instruction:**

Name	[W] [Out]
TBL1	MW00000
[A] Prm	[B] [Sts]
DA00000	MB000010

指令執行後將刪除資料，如下所示。

行	列				
	1 (W)*	2 (W)*	3 (L)*	4 (文字列)*	5 (F)*
1	1000	1001	10000	ABCD	1.1
2	2000	0	0		1.2
3	3000	0	0		1.3
4	4000	0	0		1.4
5	5000	5005	50000	EFGH	1.5

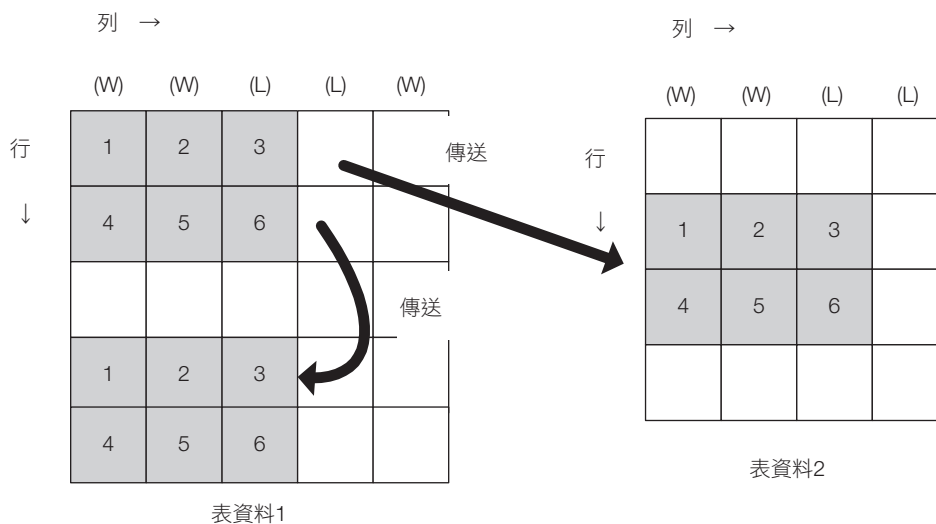
已刪除的區域

\* 顯示資料類型。

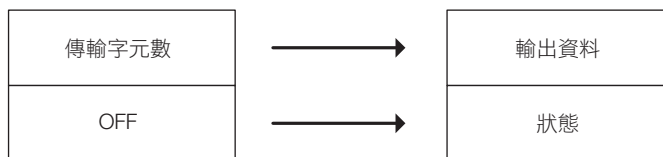
## 傳送表間區段 (TBLMV)

將傳送表名稱、行編號、列編號指定的表資料區段資料至該表的其他區段。可在不同表間或同一表間傳送。傳送來源與對象的列要素資料類型不同時，將報告錯誤，不傳送資料。

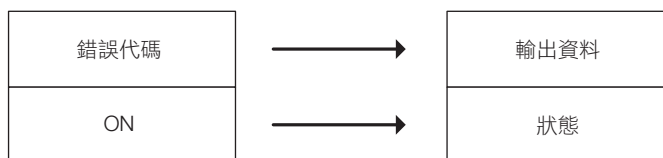
正常結束時，輸出傳輸字組數，狀態設為 OFF。發生錯誤時，輸出錯誤代碼，狀態設為 ON。



• 傳輸成功時



• 傳輸失敗時

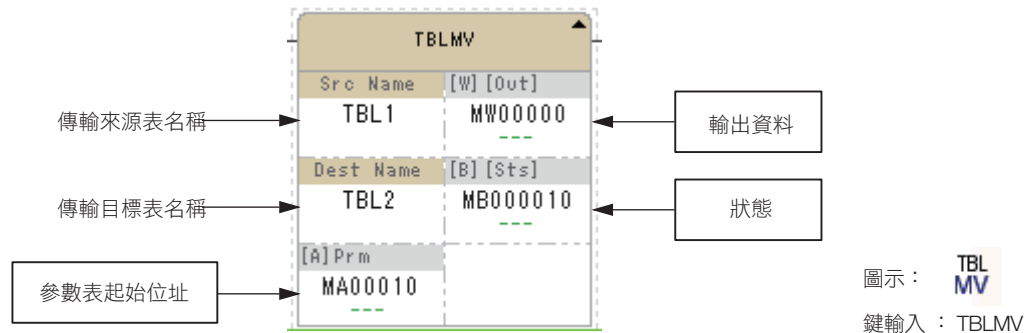


**補充** 傳送失敗時，資料將保持指令執行之前的狀態。



## 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
參數表起始位址 (Prm)	×	×	×	×	×	×	○	×	×
輸出資料 (Out) <sup>*1</sup>	×	○ <sup>*2</sup>	×	×	×	×	×	○	×
狀態 (Sts) <sup>*1</sup>	○ <sup>*2</sup>	×	×	×	×	×	×	×	×

\*1. 可以省略

\*2. C、# 暫存器除外

### ◆ 參數表

位址	資料類型	記號	名稱	規格	輸出輸入
0	L	ROW1	表元素開始行編號	作為傳送來源的表要素開始行編號 (1 ~ 65535)	IN
2	L	COLUMN1	表元素開始列編號	作為傳送來源的表要素開始列編號 (1 ~ 32767)	IN
4	W	RLEN	行元素個數	行元素個數 (1 ~ 32767)	IN
5	W	CLEN	列元素個數	列元素個數 (1 ~ 32767)	IN
6	L	ROW2	表元素開始行編號	作為傳送對象的表要素開始行編號 (1 ~ 65535)	IN
8	L	COLUMN2	表元素開始列編號	作為傳送對象的表要素開始列編號 (1 ~ 32767)	IN

### ◆ 錯誤代碼一覽

錯誤代碼	錯誤名稱	內容
0001 H	未定義對照表	未定義目標表。
0002 H	行編號範圍外	表元素的行編號不在目標表範圍內。
0003 H	列編號範圍外	表元素的列編號不在目標表範圍內。
0004 H	元素個數不良	目標元素的個數不恰當。
0005 H	儲存目的地空間不足	儲存空間不足。
0006 H	元素型不足	所指定的元素型異常。
0007 H	佇列緩衝區錯誤	當佇列緩衝區為空白時執行讀取，或佇列緩衝區已滿時以指標步進執行寫入。
0008 H	佇列表錯誤	指定之表並非佇列表。
0009 H	系統錯誤	執行指令時檢測到系統內部產生非預期的錯誤。

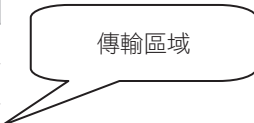
(註) 錯誤代碼為所有表操作指令共通。

### 程式範例

開關 1 (DB000100) ON 時，傳送記錄型表資料 TBL1 的指定區段至表資料 TBL2 指定區段的程式範例，如下所示。

表資料 TBL1 內容如下所示。

行	列		
	1 (W)*	2 (W)*	3 (L)*
1	1000	1001	10000
2	2000	2002	20000
3	3000	3003	30000
4	4000	4004	40000
5	5000	5005	50000



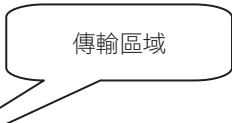
\* 顯示資料類型。

參數表請設定如下。

暫存器	數據	備註
DL00000	2	傳送來源開始行編號
DL00002	1	傳送來源開始列編號
DW00004	3	行元素數
DW00005	3	列元素數
DL00006	2	傳送對象開始行編號
DL00008	2	傳送對象開始列編號

表資料 TBL2 的內容如下所示。

行	列				
	1 (W)*	2 (W)*	3 (W)*	4 (W)*	5 (W)*
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0



\* 顯示資料類型。

**EXPRESSION**

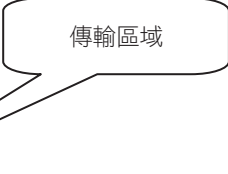
```
DL00000 = 2; DL00002 = 1; // 表要素開始行・列番号
2=2; 1=1
DW00004 = 3; // 行要素數
3=3
DW00005 = 3; // 列要素數;
3=3
DL00006 = 2; DL00008 = 2; // 転送先開始行：列番号
2=2; 2=2
```

DB000100  
スイッチ1

TBLMV	
Src Name	[W] [Out]
TBL1	MW00000 12
---	
Dest Name	[B] [Sts]
TBL2	MB000010 0
---	
[A] Pr m	DA00000 ---

指令執行後的表資料 TBL2 內容如下。

行	列				
	1 (W)*	2 (W)*	3 (W)*	4 (W)*	5 (W)*
1	0	0	0	0	0
2	0	2000	2002	20000	0
3	0	3000	3003	30000	0
4	0	4000	4004	40000	0
5	0	0	0	0	0



\* 顯示資料類型。

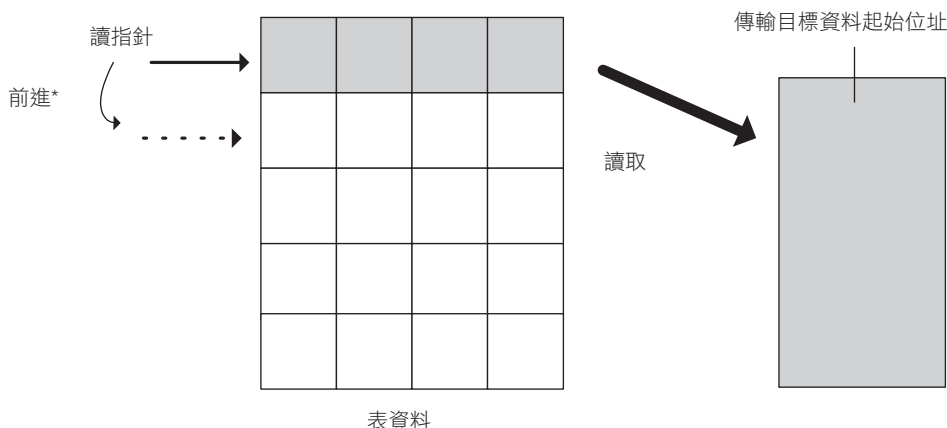
## 讀取佇列表 (QTBLR, QTBLRI)

連續讀取表明稱、行編號、列編號指定的表資料列要素，儲存至指定暫存器起的連續區域。讀取的要素資料類型，依據指定表自動進行判別。儲存對象的暫存器資料類型將被忽視，且在不轉換讀取值的資料類型情況下，依據表的資料類型進行儲存。

如為 QTBLR 指令，佇列表讀取指標將不會改變。如為 QTBLRI 指令，指令，佇列表讀取指標將步進 1 行。

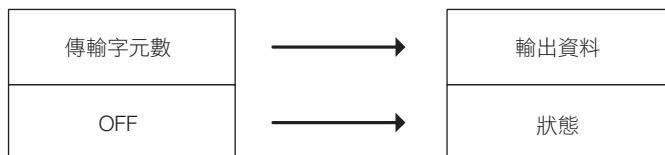
引用表時，如發生表名稱錯誤、行編號範圍外、佇列表緩存清空等錯誤，將報告錯誤，不讀取資料，指標也不會步進。保留儲存目的地暫存器的內容。

正常結束時，輸出傳輸字組數，狀態設為 OFF。發生錯誤時，輸出錯誤代碼，狀態設為 ON。

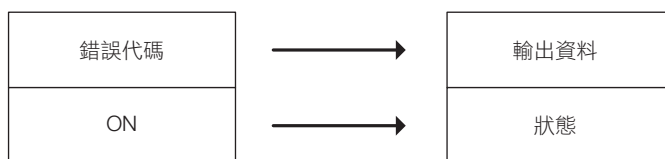


\* 執行 QTBLR 指令後不產生指標步進。  
執行 QTBLRI 指令後產生指標步進。

• 讀取成功時



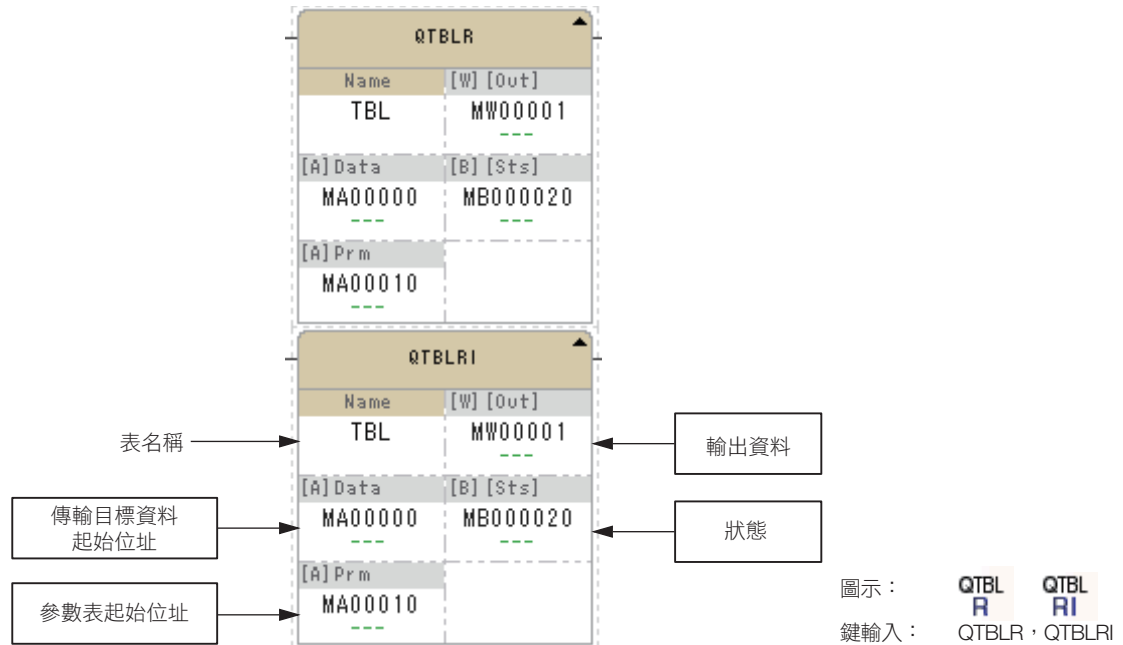
• 讀取失敗時



**補充** 讀取失敗時，傳送對象表資料將保持指令執行之前的狀態。

## 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
傳送對象資料開頭位址 (Data)	×	×	×	×	×	×	○	×	×
參數表起始位址 (Prm)	×	×	×	×	×	×	○ <sup>*2</sup>	×	×
輸出資料 (Out) <sup>*1</sup>	×	○ <sup>*2</sup>	×	×	×	×	×	○	×
狀態 (Sts) <sup>*1</sup>	○ <sup>*2</sup>	×	×	×	×	×	×	×	×

\*1. 可以省略

\*2. C、# 暫存器除外

### ◆ 參數表

位址	資料類型	記號	名稱	規格	輸出輸入
0	L	ROW	表元素相對行編號	作為傳送來源的表元素相對行編號 (1 ~ 65535)	IN
2	L	COLUMN	表元素第一列編號	作為傳送來源的表元素開頭列編號 (1 ~ 32767)	IN
4	W	CLEN	列元素個數	傳輸之列元素個數 (1 ~ 32767)	IN
5	W	保留			
6	L	RPTR	讀指針	執行後佇列表的讀指針	OUT
8	L	WPTR	輕型指針	執行後佇列表的寫入指標	OUT

## ◆ 錯誤代碼一覽

錯誤代碼	錯誤名稱	內容
0001 H	未定義對照表	未定義目標表。
0002 H	行編號範圍外	表元素的行編號不在目標表範圍內。
0003 H	列編號範圍外	表元素的列編號不在目標表範圍內。
0004 H	元素個數不良	目標元素的個數不恰當。
0005 H	儲存目的地空間不足	儲存空間不足。
0006 H	元素型不足	所指定的元素型異常。
0007 H	佇列緩衝區錯誤	當佇列緩衝區為空白時執行讀取，或佇列緩衝區已滿時以指標步進執行寫入。
0008 H	佇列表錯誤	指定之表並非佇列表。
0009 H	系統錯誤	執行指令時檢測到系統內部產生非預期的錯誤。

(註) 錯誤代碼為所有表操作指令共通。

## ◆ 表元素相對行編號設定

相對行編號	讀取行	備註
0	讀指針行	僅 QTBLRI 指令指標步進
1	讀指針行	無讀取步進
2	讀指針行 -1	無讀取步進
3	讀指針行 -2	無讀取步進
:	:	:
n	讀取指標行 - (n-1)	無讀取步進

## 程式範例

開關 2 (DB000002) ON 時，讀取陣列表資料 TBL1 的指定列要素的 DW00010 至 DW00012 區域的程式範例，如下所示。

在開關 2 設為 ON 之前，於變更 DW00000 至 DW00002 的值的同時將開關 1 設 3 次 ON，表資料內容將設定如下。詳情請參照如下內容。

 寫入佇列表 (QTBLW, QTBLWI) - 程式範例 (第 4-234 頁)

表資料 TBL1 內容如下所示。

行	列		
	1 (W)*	2 (W)*	3 (W)*
1	11	12	13
2	21	22	23
3	31	32	33

\* 顯示資料類型。

參數表請設定如下。

暫存器	數據	備註
DL00010	0	相對行編號
DL00012	1	開始列編號
DW00014	3	行元素數



此處將開關 2 (DB000002) 設 3 次 ON，則從第 1 次到第 3 次的讀取資料將變化如下。

暫存器	第 1 次 資料	第 2 次 資料	第 3 次 資料
DW00010	11	21	31
DW00011	12	22	32
DW00012	13	23	33

第 1 次的讀取指標為第 1 行，第 2 次的讀取指標為第 2 行 ... 依此類推，每次執行後指標均會步進，故為前述的結果。



註記

由於輸入電源時的讀指針及寫入指針皆為不定，因此在使用 QTBLR、QTBLRI、QTBLW 或 QTBLWI 指令前，請執行 QTBLCL 指令。

未執行 QTBLCL 指令即執行 QTBLR、QTBLRI、QTBLW 或 QTBLWI 指令，可能產生運算錯誤。

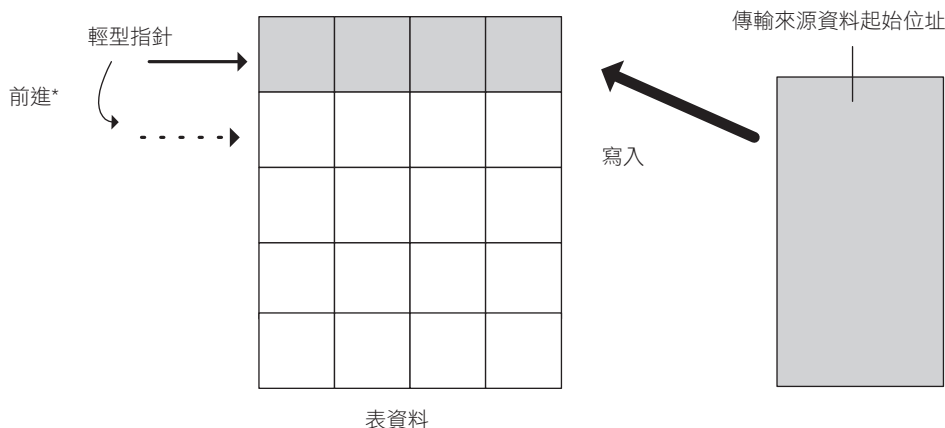
## 寫入佇列表 (QTBLW, QTBLWI)

將指定暫存器起的連續區域資料，連續寫入指定表資料列。儲存來源與對象的資料類型視為一致，進行處理。

如為 QTBLW 指令，佇列表寫入指標將不會改變。如為 QTBLWI 指令，佇列表寫入指標將步進 1 行。

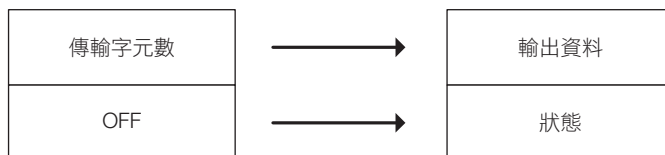
引用表時，如發生表名稱錯誤、行編號範圍外、佇列表緩存已滿等錯誤，將報告錯誤，不寫入資料，指標也不會步進。保留儲存目的地暫存器的內容。

正常結束時，輸出傳輸字組數，狀態設為 OFF。發生錯誤時，輸出錯誤代碼，狀態設為 ON。

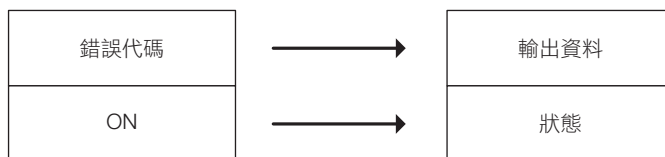


\* 執行 QTBLR 指令後不產生指標步進。

• 寫入成功時



• 寫入失敗時

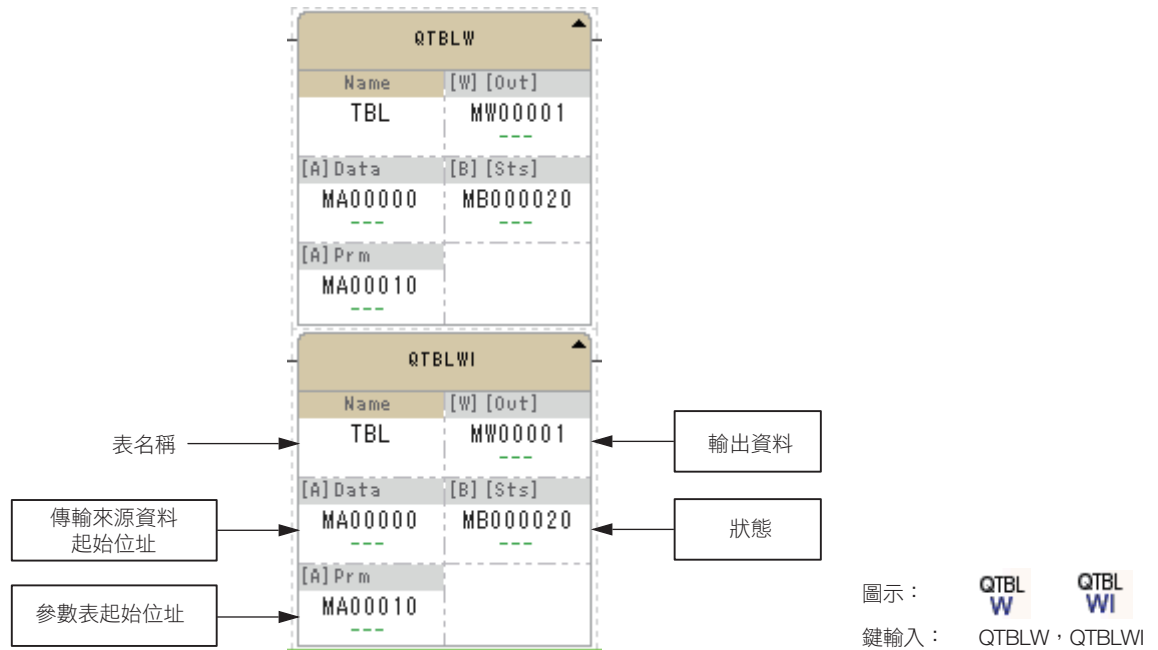


**補充** 寫入失敗時，表資料將保持指令執行之前的狀態。



## 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
傳送來源資料開頭位址 (Data)	×	×	×	×	×	×	○	×	×
參數表起始位址 (Prm)	×	×	×	×	×	×	○ <sup>*2</sup>	×	×
輸出資料 (Out) <sup>*1</sup>	×	○ <sup>*2</sup>	×	×	×	×	×	○	×
狀態 (Sts) <sup>*1</sup>	○ <sup>*2</sup>	×	×	×	×	×	×	×	×

\*1. 可以省略

\*2. C、# 暫存器除外

### ◆ 參數表

位址	資料類型	記號	名稱	規格	輸出輸入
0	L	ROW	表元素相對行編號	作為傳送對象的表元素相對行編號 (1 ~ 65535)	IN
2	L	COLUMN	表元素第一列編號	作為傳送對象的表元素開頭列編號 (1 ~ 32767)	IN
4	W	CLEN	列元素個數	傳輸之列元素個數 (1 ~ 32767)	IN
5	W	保留			
6	L	RPTR	讀指針	執行後佇列表的讀指針	OUT
8	L	WPTR	輕型指針	執行後佇列表的寫入指標	OUT

## ◆ 錯誤代碼一覽

錯誤代碼	錯誤名稱	內容
0001 H	未定義對照表	未定義目標表。
0002 H	行編號範圍外	表元素的行編號不在目標表範圍內。
0003 H	列編號範圍外	表元素的列編號不在目標表範圍內。
0004 H	元素個數不良	目標元素的個數不恰當。
0005 H	儲存目的地空間不足	儲存空間不足。
0006 H	元素型不足	所指定的元素型異常。
0007 H	佇列緩衝區錯誤	當佇列緩衝區為空白時執行讀取，或佇列緩衝區已滿時以指標步進執行寫入。
0008 H	佇列表錯誤	指定之表並非佇列表。
0009 H	系統錯誤	執行指令時檢測到系統內部產生非預期的錯誤。

(註) 錯誤代碼為所有表操作指令共通。

## ◆ 表元素相對行編號設定

相對行編號	讀取行	備註
0	寫入指標行	僅 QTBLWI 指令指標步進
1	寫入指標行	無讀取步進
2	寫入指標行 -1	無讀取步進
3	寫入指標行 -2	無讀取步進
:	:	:
n	讀取指標行 - (n-1)	無讀取步進

## 程式範例

開關 1 (DB000001) ON 時，對陣列表資料 TBL1 的指定列寫入 MW00000 至 MW00002 資料的程式範例，如下所示。

將執行前的表資料 TBL1 初始化。

行	列		
	1 (W)*	2 (W)*	3 (W)*
1	0	0	0
2	0	0	0
3	0	0	0

\* 顯示資料類型。

參數表請設定如下。

暫存器	數據	備註
DL00010	0	相對行編號
DL00012	1	開始列編號
DW00014	3	行元素數



此處將改變 MW00000 至 MW00002 的資料，如下所示，同時使開關 1 (DB000001) 進行 3 次 ON。

暫存器	第 1 次 資料	第 2 次 資料	第 3 次 資料
MW00000	11	21	31
MW00001	12	22	32
MW00002	13	23	33

第 1 次的寫入指標為第 1 行，第 2 次的寫入指標為第 2 行 ...，每次執行時寫入指標都會步進，3 次執行後的 TBL1 資料將如下。

行	列		
	1 (W)	2 (W)	3 (W)
1	11	12	13
2	21	22	23
3	31	32	33

\* 顯示資料類型。

在第一回寫入

在第二回寫入

在第三回寫入



註記

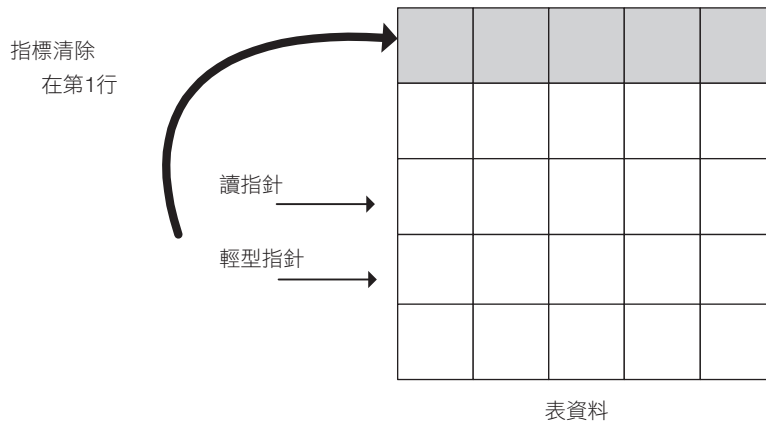
由於輸入電源時的讀指針及寫入指標皆為不定，因此在使用 QTBLR、QTBLRI、QTBLW 或 QTBLWI 指令前，請執行 QTBLCL 指令。

未執行 QTBLCL 指令即執行 QTBLR、QTBLRI、QTBLW 或 QTBLWI 指令，可能產生運算錯誤。

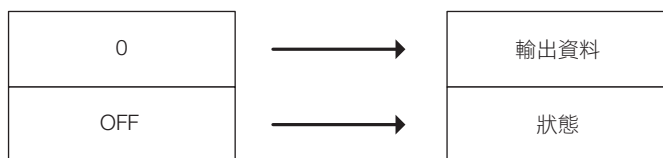
## 佇列指標清除 (QTBLCL)

將表名稱指定表資料的佇列讀取指標與佇列寫入指標還原為初始狀態 (第 1 行)。

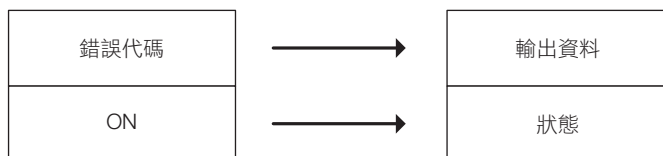
正常結束時，將輸出資料設為 0，設狀態為 OFF。發生錯誤時，於輸出資料設定錯誤代碼，狀態設為 ON。



- 佇列清除成功時



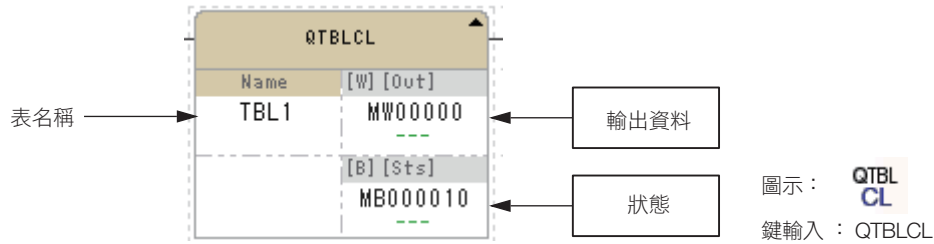
- 佇列清除失敗時



**補充** 清除失敗時，佇列將保持指令執行之前的狀態。

## 格式

格式如下所示。



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
輸出資料 (Out) <sup>*1</sup>	×	○ <sup>*2</sup>	×	×	×	×	×	○	×
狀態 (Sts) <sup>*1</sup>	○ <sup>*2</sup>	×	×	×	×	×	×	×	×

\*1. 可以省略

\*2. C、# 暫存器除外

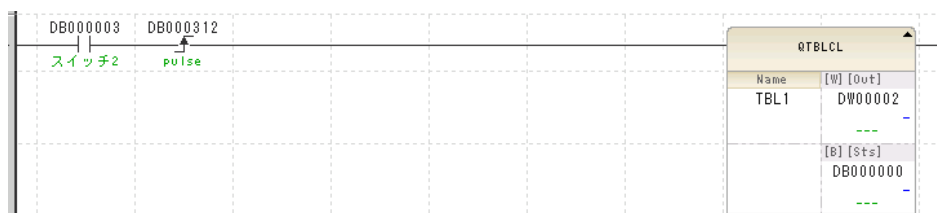
### ◆ 錯誤代碼一覽

錯誤代碼	錯誤名稱	內容
0001 H	未定義對照表	未定義目標表。
0002 H	行編號範圍外	表元素的行編號不在目標表範圍內。
0003 H	列編號範圍外	表元素的列編號不在目標表範圍內。
0004 H	元素個數不良	目標元素的個數不恰當。
0005 H	儲存目的地空間不足	儲存空間不足。
0006 H	元素型不足	所指定的元素型異常。
0007 H	佇列緩衝區錯誤	當佇列緩衝區為空白時執行讀取，或佇列緩衝區已滿時以指標步進執行寫入。
0008 H	佇列表錯誤	指定之表並非佇列表。
0009 H	系統錯誤	執行指令時檢測到系統內部產生非預期的錯誤。

(註) 錯誤代碼為所有表操作指令共通。

## 程式範例

開關 2 (DB000003) ON 時，將指定佇列表的佇列指標初始化的程式範例，如下所示。



註記

由於輸入電源時的讀指針及寫入指標皆為不定，因此在使用 QTBLR、QTBLRI、QTBLW 或 QTBLWI 指令前，請執行 QTBLCL 指令。

未執行 QTBLCL 指令即執行 QTBLR、QTBLRI、QTBLW 或 QTBLWI 指令，可能產生運算錯誤。

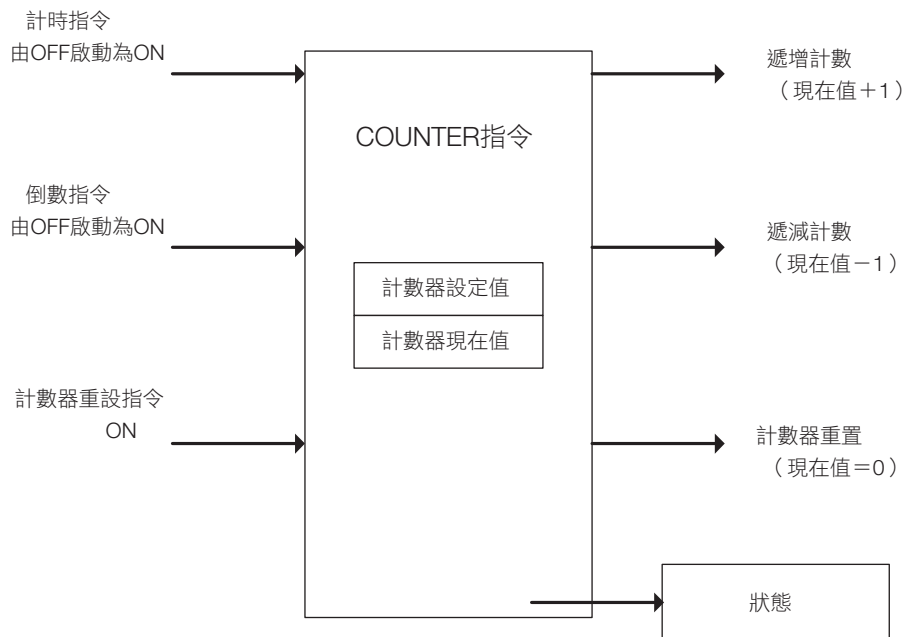
## 4.10 系統函數指令

### 計數器 (COUNTER)

遞增／遞減計數指令由 OFF 變為 ON 時，目前值將遞增／遞減。

計數器重置指令為 ON 時，計數器目前值為 0。此外會將計數器目前值與設定值比對，並輸出結果。

計數器錯誤（目前值 > 設定值）時，目前值不會遞增／遞減。

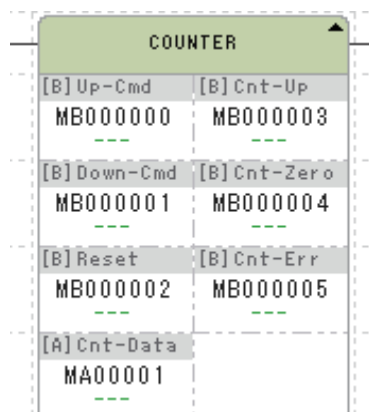


將輸出下列 3 種狀態。

- 計數器一致（目前值 = 設定值）
- 計數器 0（目前值 = 0）
- 計數器錯誤  
（目前值 > 設定值、目前值 < 0）

## 格式

格式如下所示。



圖示：**COUNTER**  
鍵輸入：COUNTER

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
計時指令 (Up-Cmd)	○	×	×	×	×	×	×	×	×
倒數指令 (Down-Cmd)	○	×	×	×	×	×	×	×	×
計數器重設指令 (Reset)	○	×	×	×	×	×	×	×	×
計數器處理用資料區域開頭位址 (Cnt-Data)	×	×	×	×	×	×	○ <sup>*1</sup>	×	×
遞增計數 (Cnt-Up)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×
零計數 (Cnt-Zero)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×
計數錯誤 (Cnt-Err)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×

\*1. 僅限 M、D 暫存器

\*2. C、# 暫存器除外

以下說明各輸入輸出項目的內容。

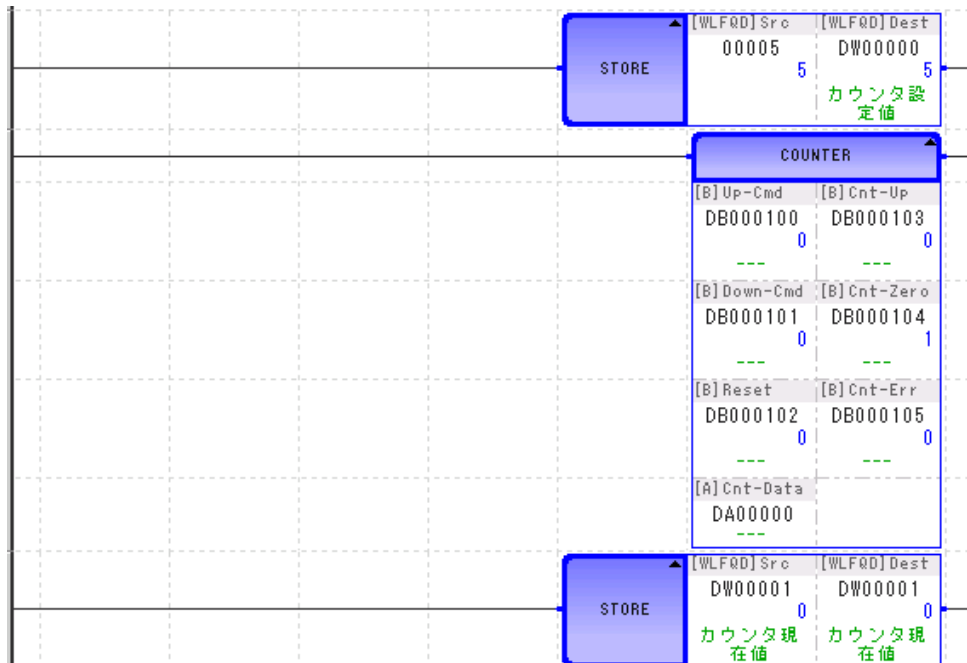
輸出輸入項目	內容	輸出輸入
遞增計數指令 (Up-Cmd)	OFF → ON 時，目前值將遞增計數*。	IN
遞減計數指令 (Down-Cmd)	OFF → ON 時，目前值將遞減計數*。	IN
計數器重置指令 (Reset)	ON 時，目前值重置為 0。	IN
計數器處理用資料區域開頭位址 (Cnt-Data)	+ 0 字組：設定值	IN
	+ 1 字組：現在值	OUT
	+ 2 字組：工作標記	OUT
遞增計數 (Cnt-Up)	目前值 = 設定值時 ON。	OUT
零計數 (Cnt-Zero)	目前值 = 0 時 ON。	OUT
計數錯誤 (Cnt-Err)	目前值 > 設定值時 ON。 或目前值 < 0 時 ON。	OUT

\* 遞增以及遞減計數指令同時由 OFF 變為 ON 時，目前值不變。

## 程式範例

下列程式範例的第 1 行，將計數器設定值設為 5，第 3 行則是進行計數器目前值 (DW00001) 的監視。

此時，當遞增計數指令 (DB000100) 由 OFF 變為 ON，DW00001 將遞增計數；遞減計數指令 (DB000101) 由 OFF 變為 ON 時，DW00001 將遞減計數。



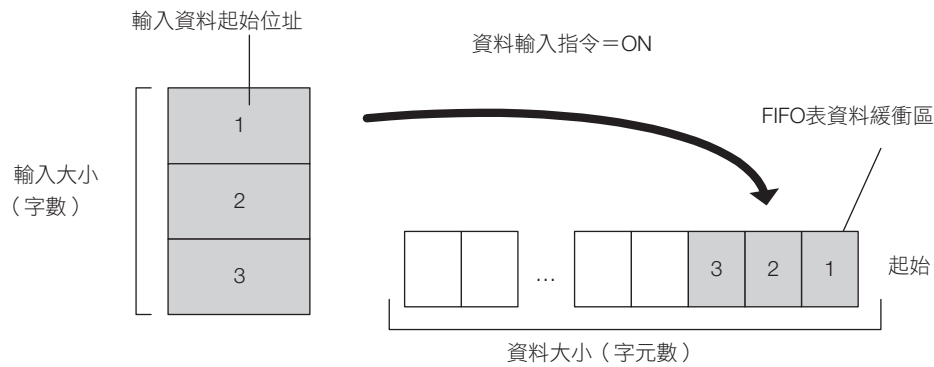


## 先進先出 (FINFOUT)

先進先出型的區段資料傳送函數。FIFO 表是由 4 個字的表頭部分與資料緩存所構成。在 FINFOUT 指令呼叫前，請務必設定表頭部分的 3 個字（資料大小、輸入大小、輸出大小）。

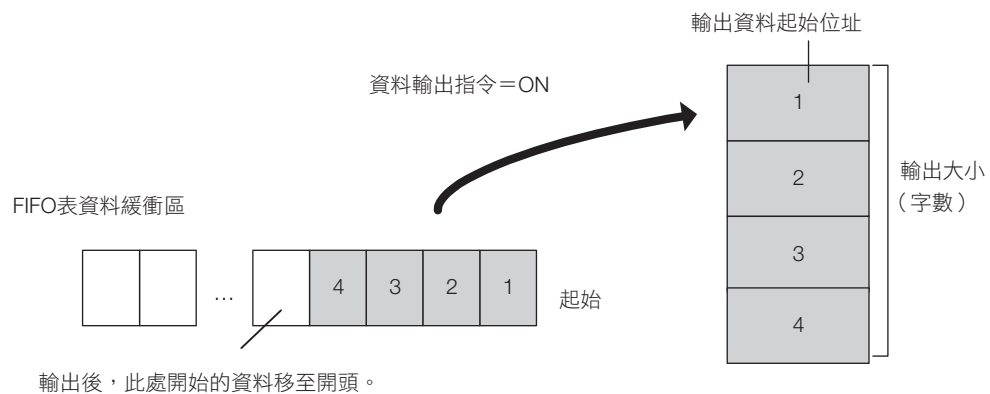
### ■ 資料輸入指令 (In-Cmd) = ON 時

資料輸出指令為 ON 時，指定個數資料將由指定的輸入資料區域依序儲存至 FIFO 表的資料區域。



### ■ 資料輸出指令 (Out-Cmd) = ON 時

資料輸出指令為 ON 時，將指定個數資料由 FIFO 表的資料區域開頭傳送至指定的輸出資料區域。



### ■ 重置指令 (Reset) = ON 時

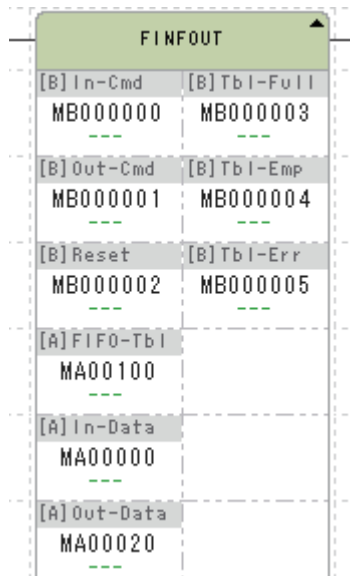
FIFO 表的儲存數設為 0，FIFO 表空白 (Tbl-Emp) 變為 ON。

(註) 資料緩存內容則將保持，不會清零。

**補充** 資料剩餘空間 < 輸入大小，或是資料大小 < 輸出大小時，FIFO 表錯誤 (Tbl-Err) 變為 ON。

## 格式

格式如下所示。



圖示：**FIN FOUT**  
鍵輸入：FINFOUT

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
資料輸入指令 (In-Cmd)	○	×	×	×	×	×	×	×	×
資料輸出指令 (Out-Cmd)	○	×	×	×	×	×	×	×	×
重設指令 (Reset)	○	×	×	×	×	×	×	×	×
FIFO 表開頭位址 (FIFO-Tbl)	×	×	×	×	×	×	○ <sup>*1</sup>	×	×
輸入資料起始位址 (in-Data)	×	×	×	×	×	×	○ <sup>*1</sup>	×	×
輸出資料起始位址 (Out-Data)	×	×	×	×	×	×	○ <sup>*1</sup>	×	×
FIFO 表已滿 (Tbl-Full)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×
FIFO 表空白 (Tbl-Emp)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×
FIFO 表錯誤 (Tbl-Err)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×

\*1. 僅限 M、D 暫存器

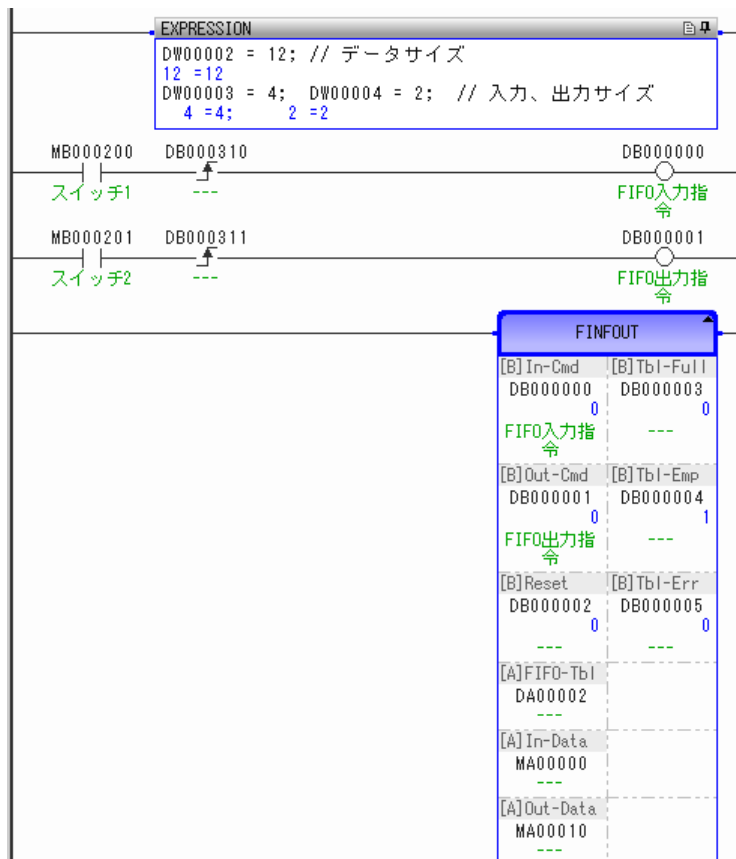
\*2. C、# 暫存器除外

以下說明各輸入輸出項目的內容。

輸出輸入項目	内容	輸出輸入
資料輸入指令 (In-Cmd)	ON 時儲存至 FIFO 表。	IN
資料輸出指令 (Out-Cmd)	ON 時由 FIFO 表傳送。	IN
重設指令 (Reset)	ON 時儲存數為 0。	IN
FIFO 表開頭位址 (FIFO-Tbl)	+ 0 字組：資料尺寸	IN
	+ 1 字：輸入大小	IN
	+ 2 字：輸出大小	IN
	+ 3 字：資料儲存數	OUT
+ 4 字～：資料	OUT	
輸入資料開頭位址 (in-Data)	輸入資料開頭位址	IN
輸出資料開頭位址 (Out-Data)	輸出資料開頭位址	IN
FIFO 表已滿 (Tbl-Full)	FIFO 表已滿時 ON。	OUT
FIFO 表空白 (Tbl-Emp)	FIFO 表空白時 ON。	OUT
FIFO 表錯誤 (Tbl-Err)	FIFO 表發生錯誤時 ON。	OUT

## 程式範例

建立資料大小為 12 個字、輸入大小 4 個字、輸出大小 2 個字的 FIFO 表，執行 FINFOUT 指令的程式範例，如下所示。



開關 1 (MB000200) ON 時，MW00000 至 MW00003 的資料將儲存於 FIFO 表緩存之中。此外，資料儲存數 (DW00005) 變為 4。

暫存器	數據	FIFO表緩衝區	數據
MW00000	123	DW00006	123
MW00001	234	DW00007	234
MW00002	345	DW00008	345
MW00003	456	DW00009	456
		DW00010	0
		:	:
		DW00017	0

其後，開關 2 (MB000201) ON 時，FIFO 表緩存的開頭 2 個字資料將輸出至 MW00010 至 MW00011 的區域。此外，資料儲存數 (DW00005) 變為 2。

FIFO表緩衝區	數據	暫存器	數據
DW00006	123 →345	MW00010	123
		MW00011	234
DW00007	234 →456	MW00012	0

## 追蹤 (TRACE)

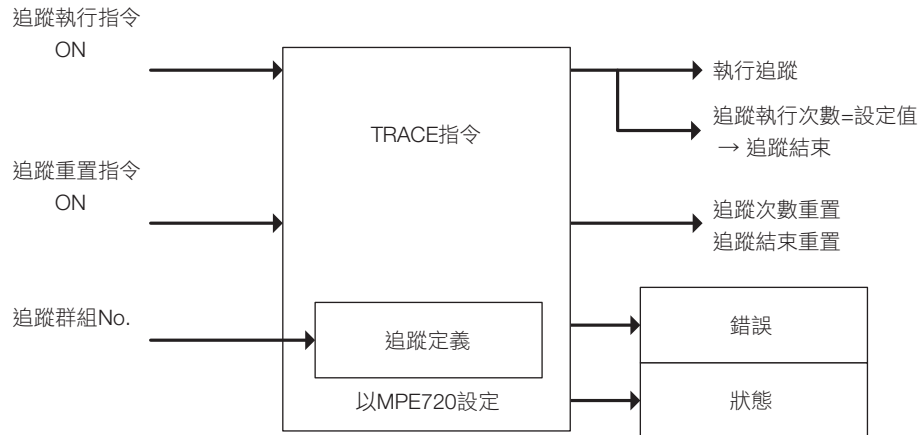
對於追蹤群組 No. (1 ~ 4) 指定的追蹤資料執行追蹤執行控制。

### 補充

追蹤定義的設定，以 MPE720 的「資料追蹤定義」執行。詳細內容，請參照以下手冊。

📖 運動控制器 MP2000/3000 系列 運動控制器系統 安裝手冊 (資料編號: SIJP C880725 12)

- 追蹤執行指令 (Execute) 為 ON 時，執行追蹤。
- 追蹤重置指令 (Reset) 為 ON 時，將重置追蹤次數計數器。此時，追蹤結束 (Trc-End) 也將重置。
- 執行了追蹤設定的次數後，追蹤結束 (Trc-End) 將變為 ON。



## 格式

格式如下所示。

TRACE	
[B] Execute	[B] Trc-End
MB000000	MB000002
[B] Reset	[B] Error
MB000001	MB000003
[W] Group-No	[W] Status
MW000002	MW000001

圖示： TRACE  
鍵輸入：TRACE

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
追蹤執行指令 (Execute)	○	×	×	×	×	×	×	×	×
追蹤重設指令 (Reset)	○	×	×	×	×	×	×	×	×
追蹤群組 No. (Group-No)	×	○	×	×	×	×	×	○	○
結束追蹤指令 (Trc-End)	○*	×	×	×	×	×	×	×	×
錯誤 (Error)	○*	×	×	×	×	×	×	×	×
狀態 (Status)	×	○*	×	×	×	×	×	○	×

\* C、# 暫存器除外

## 追蹤 (TRACE)

以下說明各輸入輸出項目的內容。

輸出輸入項目	內容	輸出輸入
追蹤執行指令 (Execute)	ON 則開始執行追蹤。	IN
追蹤重設指令 (Reset)	ON 則重置追蹤。	IN
追蹤群組 No.(Group-No)	追蹤群組 No. (1 ~ 4)	IN
結束追蹤指令 (Trc-End)	追蹤結束時 ON。	OUT
錯誤 (Error)	發生錯誤而 ON。	OUT
狀態 (Status)	追蹤執行狀態	OUT

狀態組成如下所示。

Bit	名稱	備註
0	追蹤資料已滿	指定群組的數據追蹤記憶體內旋轉 1 次後，變為 ON。
1 ~ 7	系統預約	-
8	無追蹤定義	不會執行函數。
9	指定群組 No. 錯誤	不會執行函數。
A ~ C	系統預約	-
D	執行時間點錯誤	不會執行函數。
E	系統預約	-
F	系統預約	-

## 程式範例

使用追蹤群組 No.1 的定義執行追蹤的程式範例，如下所示。

追蹤執行指令 (DB000000) ON 時，開始追蹤。

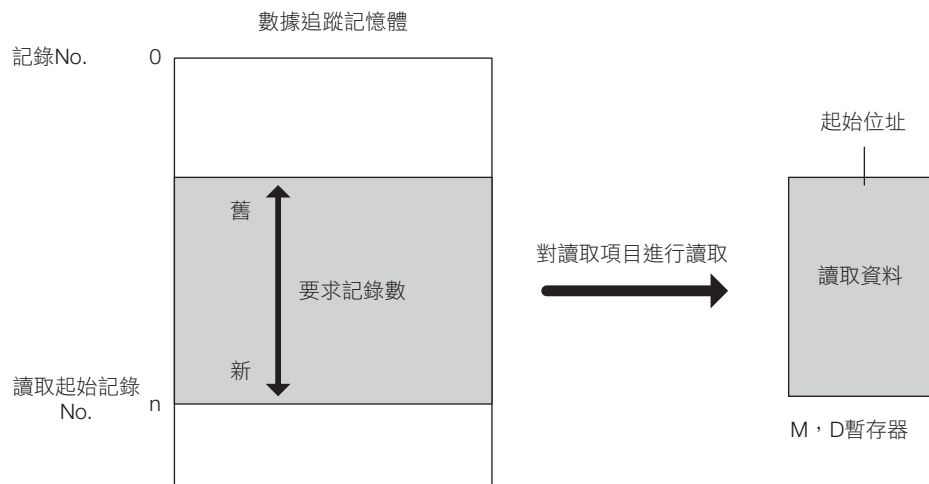


請先在 MPE720 上設定追蹤群組 No.1 的數據追蹤定義。此時，取樣條件請設為「程式指定」。

TRACE	
[B] Execute DB000000 0	[B] Trc-End DB000002 0
[B] Reset DB000001 0	[B] Error DB000003 0
[W] Group-No 00001 1	[W] Status DW00001 0

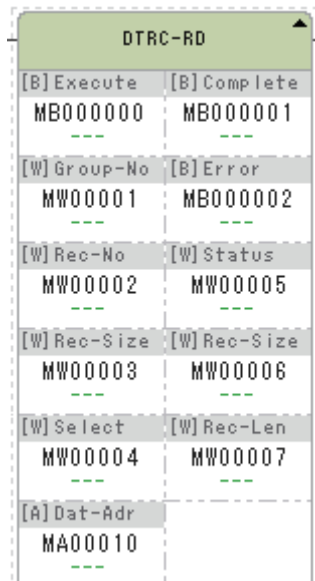
## 讀取追蹤資料 (DTRC-RD)

讀取運動控制器本體的追蹤資料，儲存至暫存器。指定記錄 No.、記錄數後，可讀取追蹤記憶體內的資料。即使只指定記錄中的必要項目，也可進行讀取。



## 格式

格式如下所示。



圖示：  
鍵輸入：DTRCRD

輸出輸入項目	可以使用的資料類型								指標	常數
	B	W	L	Q	F	D	A			
追蹤讀取執行指令 (Execute)	○	×	×	×	×	×	×	×	×	×
追蹤群組 No. (Group-No)	×	○	×	×	×	×	×	×	×	×
記錄 No.(Rec-No)	×	○	×	×	×	×	×	×	×	×
記錄數 (Rec-Size)	×	○	×	×	×	×	×	×	×	×
項目選擇 (Select)	×	○	×	×	×	×	×	×	×	×
起始位址 (Dat-Adr)	×	×	×	×	×	×	○ <sup>*1</sup>	×	×	×
完成追蹤 (Complete)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×	×
錯誤 (Error)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×	×
狀態 (Status)	×	○ <sup>*2</sup>	×	×	×	×	×	×	×	×
讀取記錄數 (Rec-Size)	×	○ <sup>*2</sup>	×	×	×	×	×	×	×	×
讀取 1 記錄數 (Rec-Len)	×	○ <sup>*2</sup>	×	×	×	×	×	×	×	×

\*1. 僅限 M、D 暫存器

\*2. C、# 暫存器除外



以下說明各輸入輸出項目的內容。

輸出輸入項目	內容	輸出輸入
追蹤讀取執行指令 (Execute)	資料追蹤讀取執行指定	IN
追蹤群組 No.(Group-No)	資料追蹤群組 No. (1 ~ 4)	IN
記錄 No.(Rec-No)	讀取起始記錄 No. (0 ~ 最大記錄數 -1)	IN
記錄數 (Rec-Size)	讀取要求記錄數 (0 ~ 最大記錄數 -1)	IN
項目選擇 (Select)	讀取項目 (0001H ~ FFFFH) 位元 0 ~ F 對應了追蹤定義的指定資料 1 ~ 16。	IN
起始位址 (Dat-Adr)	讀取開頭暫存器 No. (MA, DA)	IN
完成追蹤 (Complete)	追蹤讀取結束時 ON。	OUT
錯誤 (Error)	發生錯誤而 ON。	OUT
狀態 (Status)	資料追蹤讀取執行狀態	OUT
讀取記錄數 (Rec-Size)	讀取記錄數	OUT
讀取 1 記錄數 (Rec-Len)	讀取 1 記錄數 (字數)	OUT

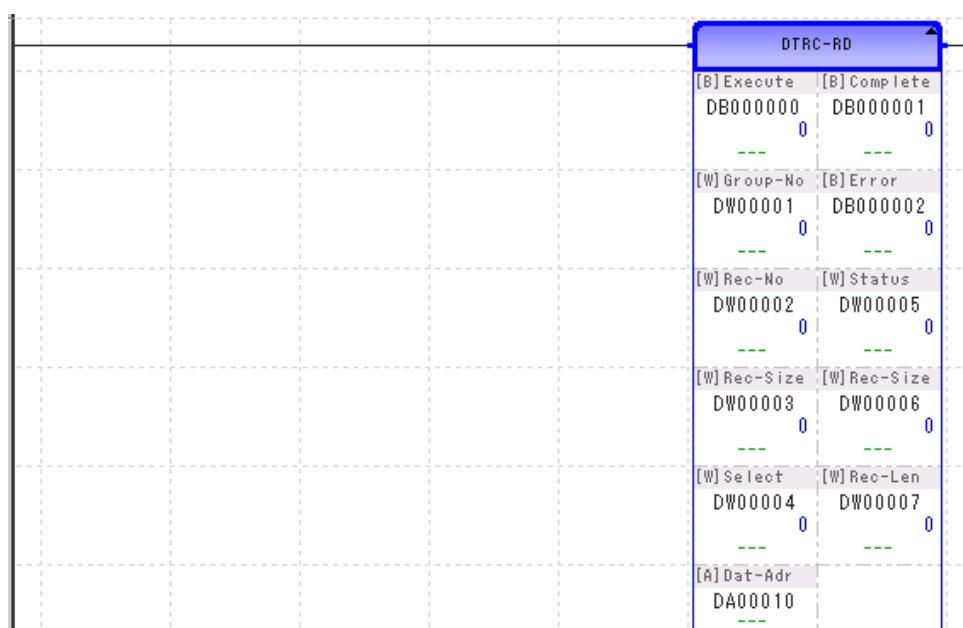
狀態組成如下所示。

Bit	名稱	備註
0 ~ 7	系統預約	-
8	無追蹤定義	不會執行函數。
9	群組 No. 錯誤	不會執行函數。
A	指定記錄 No. 錯誤	不會執行函數。
B	指定記錄數錯誤	不會執行函數。
C	資料儲存錯誤	不會執行函數。
D	系統預約	-
E	系統預約	-
F	位址輸入錯誤	不會執行函數。

## 程式範例

讀取群組定義 No.1 的資料追蹤的程式範例，如下所示。

追蹤執行指令 (DB000000) ON 時，開始讀取追蹤資料。

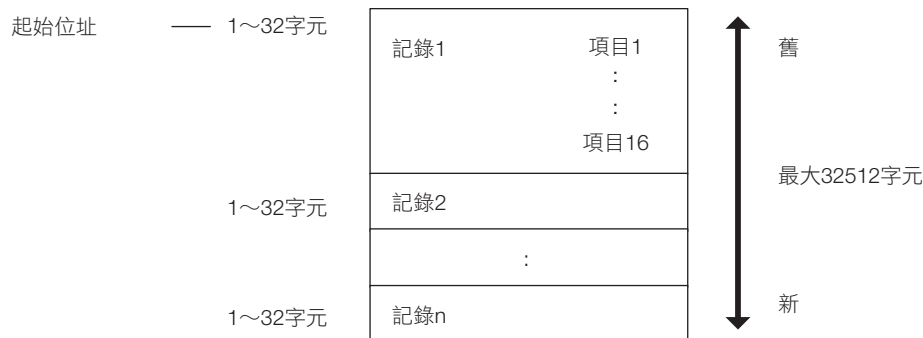


## 補充事項

### ◆ 讀取資料結構

記錄長度由選擇項目資料所構成，每 1 筆記錄大小為 1 ~ 32 個字。

最大記錄數隨記錄長度變動，範圍為 1015 ~ 32511。



### ◆ 記錄長度

記錄由選擇的項目資料構成。

記錄長度 (1 個記錄的字數) 將根據選擇的暫存器或點數而不同。

- 1 筆記錄的文字數 =  $B_n \times 1$  個字 +  $W_n \times 1$  個字 +  $L_n \times 2$  個字 +  $F_n \times 2$  個字

$B_n$  : 位元型暫存器的選擇點數

$W_n$  : 整數型暫存器的選擇點數

$L_n$  : 2 倍長整數型暫存器的選擇點數

$F_n$  : 實數型暫存器的選擇點數

合計最多 16 點。

- 記錄長度最大 = 32 個字 (當 2 倍長整數型或實數型計數器為 16 點時)
- 記錄長度最小 = 1 個字 (當位元型或整數型計數器為 1 點時)

### ◆ 記錄數

指定記錄數將因為記錄長度而有如下所示的改變。

- 記錄長度最大時的記錄數 : 0 ~ 1015
- 記錄長度最小時的記錄數 : 0 ~ 32511 (上限 : 32521 / 記錄長度 - 1)

### ◆ 最新記錄編號

各追蹤群組的最新記錄編號，將如下表般儲存於系統暫存器中。

系統暫存器編號	內容
SW00100	群組 1 最新記錄編號
SW00101	群組 2 最新記錄編號
SW00102	群組 3 最新記錄編號
SW00103	群組 4 最新記錄編號
SW00104	-
SW00105	-
SW00106	-
SW00107	-

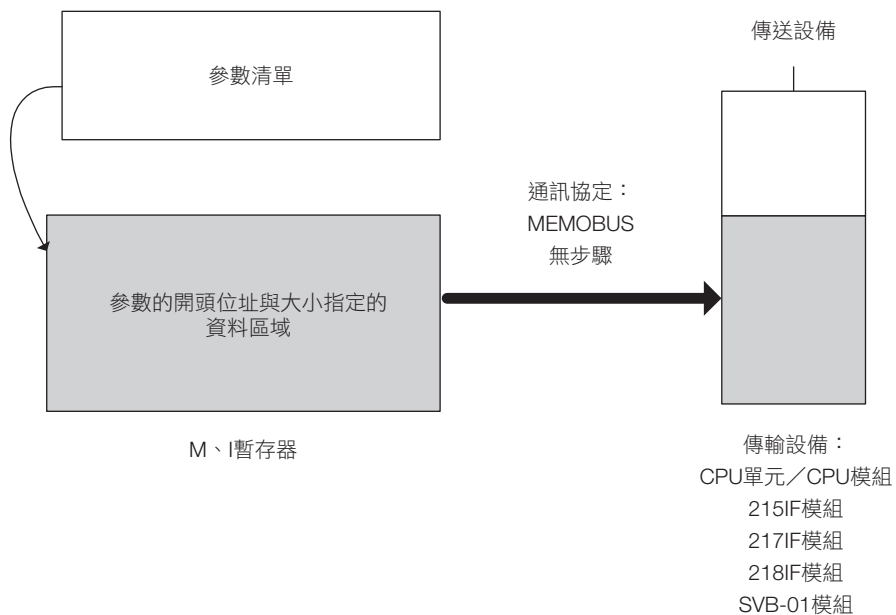
## 資訊傳送 (MSG-SND)

依據傳輸設備種類，傳送訊息至指定線路上的對方端。

支援下列傳輸設備和通訊協定。

傳輸設備：CPU 單元／CPU 模組、215IF 模組、217IF 模組、218IF 模組、SVB-01 模組

通訊協定：MEMOBUS 通訊協定，無步驟



## 格式

格式如下所示。

MSG-SND	
[B] Execute	[B] Busy
MB000000	MB000002
[B] Abort	[B] Complete
MB000001	MB000003
[W] Dev-Typ	[B] Error
MW000001	MB000004
[W] Pro-Typ	
MW000002	
[W] Cir-No	
MW000003	
[W] Ch-No	
MW000004	
[A] Param	
MA000010	

圖示： MSG-SND

鍵輸入：MSG-SND

輸出輸入項目	可以使用的資料類型								指標	常數
	B	W	L	Q	F	D	A			
傳送執行指令 (Execute)	○	×	×	×	×	×	×	×	×	×
傳送強制中止指令 (Abort)	○	×	×	×	×	×	×	×	×	×
傳輸設備種類 (Dev-Typ)	×	○	×	×	×	×	×	○	○	○
傳輸協定 (Pro-Typ)	×	○	×	×	×	×	×	○	○	○
線路編號 (Cir-No)	×	○	×	×	×	×	×	○	○	○
傳輸緩衝器頻道編號 (Ch-No)	×	○	×	×	×	×	×	○	○	○
參數表起始位址 (Param)	×	×	×	×	×	×	○ <sup>*1</sup>	×	×	×
處理中 (Busy)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×	×
處理完成 (Complete)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×	×
發生錯誤 (Error)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×	×

\*1. 僅限 M、G、D 暫存器

\*2. C、# 暫存器除外

關於輸入輸出項目、參數詳情和程式範例，請參照以下手冊。

📖 運動控制器 MP2000 系列 使用者手冊 階梯圖程式篇 (資料編號：SI-C887-1.2)

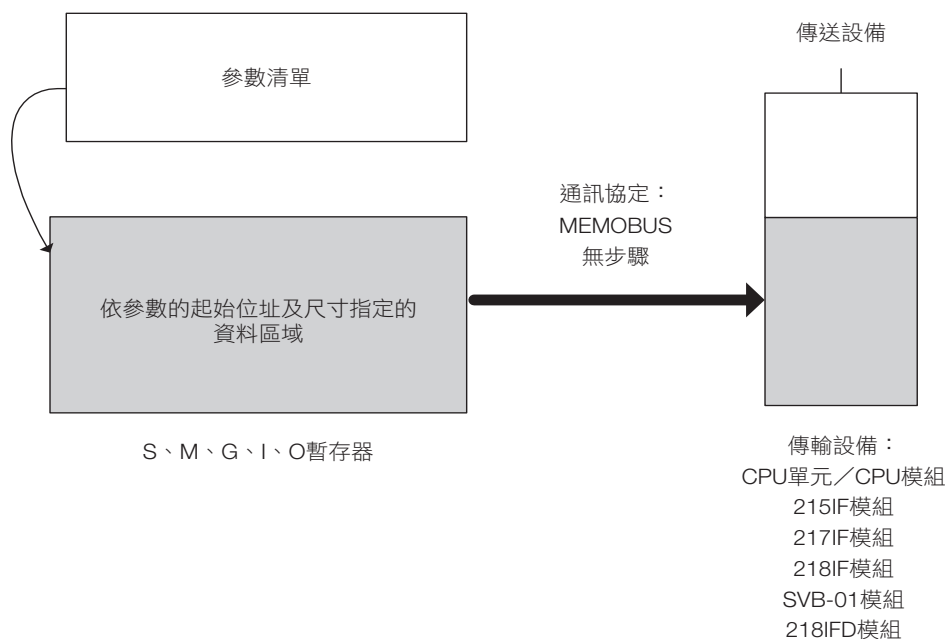
## 資訊傳送（擴展）（MSG-SNDE）

依據傳輸設備種類，傳送訊息至指定線路上的對方端。

支援下列傳輸設備和通訊協定。

傳輸設備：CPU 單元／CPU 模組、215IF 模組、217IF 模組、218IF 模組、SVB-01 模組、218IFD 模組

通訊協定：MEMOBUS 通訊協定，無步驟



基本動作如同 MSG-SND 指令，但平常請使用符合 MP3000 規格的 MSG-SNDE 指令。

MSG-SND 指令是符合 MP2000 的指令。以下的暫存器存取有所不同。

暫存器名稱	MSG-SNDE 的存取範圍		MSG-SND 的存取範圍	
	地址範圍	存取權限	地址範圍	存取權限
系統暫存器	SW00000 ~ 65534	RW	-	-
保留暫存器	MW0000000 ~ 1048575	RW	MW0000000 ~ 0065534	RW
資料暫存器	GW0000000 ~ 2097151	RW	-	-
輸入暫存器	IW00000 ~ 17FFF	R	IW00000 ~ 0FFFF	R
輸出暫存器	OW00000 ~ 17FFF	RW	-	-

(註) R：僅可讀取、RW：可讀取／寫入

## 格式

格式如下所示。

MSG-SNDE	
[B] Execute	[B] Busy
MB000000	MB000002
[B] Abort	[B] Complete
MB000001	MB000003
[W] Dev-Typ	[B] Error
MW000001	MB000004
[W] Pro-Typ	
MW000002	
[W] Cir-No	
MW000003	
[W] Ch-No	
MW000004	
[A] Param	
MA000010	

圖示：**MSG-SNDE**

鍵輸入：MSG-SNDE

輸出輸入項目	可以使用的資料類型								指標	常數
	B	W	L	Q	F	D	A			
傳送執行指令 (Execute)	○	×	×	×	×	×	×	×	×	×
傳送強制中止指令 (Abort)	○	×	×	×	×	×	×	×	×	×
傳輸設備種類 (Dev-Typ)	×	○	×	×	×	×	×	○	○	○
傳輸協定 (Pro-Typ)	×	○	×	×	×	×	×	○	○	○
線路編號 (Cir-No)	×	○	×	×	×	×	×	○	○	○
傳輸緩衝器頻道編號 (Ch-No)	×	○	×	×	×	×	×	○	○	○
參數表起始位址 (Param)	×	×	×	×	×	×	○ <sup>*1</sup>	×	×	×
處理中 (Busy)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×	×
處理完成 (Complete)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×	×
發生錯誤 (Error)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×	×

\*1. 僅限 M、G、D 暫存器

\*2. C、# 暫存器除外

關於輸入輸出項目、參數詳情和程式範例，請參照以下手冊。

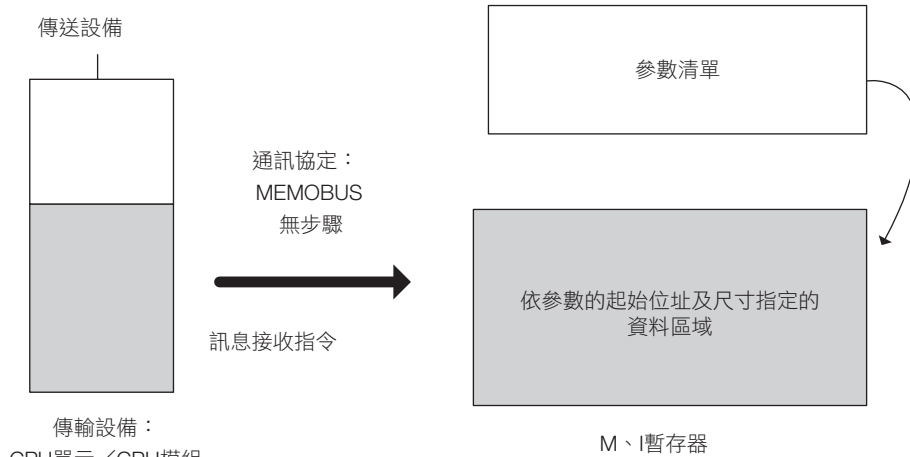
📖 MP3000 系列 通訊功能 使用者操作手冊 (資料編號：SIJP C880725 12)

## 訊息接收 (MSG-RCV)

依據傳輸設備種類，從指定線路上的對方端接收訊息。在完成接收（訊息接收完成為 ON）前，請讓訊息接收指令保持為 ON。支援下列傳輸設備和通訊協定。

傳輸設備：CPU 單元／CPU 模組、215IF 模組、217IF 模組、218IF 模組、SVB-01 模組

通訊協定：MEMOBUS 通訊協定，無步驟



(註) 傳輸完成後，訊息接收完成便會轉為 ON。  
在此之前，請讓訊息接收指令保持為 ON。

## 格式

格式如下所示。

MSG-RCV	
[B] Execute	[B] Busy
MB000000	MB000002
[B] Abort	[B] Complete
MB000001	MB000003
[W] Dev-Typ	[B] Error
MW000001	MB000004
[W] Pro-Typ	
MW000002	
[W] Cir-No	
MW000003	
[W] Ch-No	
MW000004	
[A] Param	
MA000010	

圖示：  
 鍵輸入：MSG-RCV

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
接收執行指令 (Execute)	○	×	×	×	×	×	×	×	×
接收強制中止指令 (Abort)	○	×	×	×	×	×	×	×	×
傳輸設備種類 (Dev-Typ)	×	○	×	×	×	×	×	○	○
傳輸協定 (Pro-Typ)	×	○	×	×	×	×	×	○	○
線路編號 (Cir-No)	×	○	×	×	×	×	×	○	○
傳輸緩衝器頻道編號 (Ch-No)	×	○	×	×	×	×	×	○	○
參數表起始位址 (Param)	×	×	×	×	×	×	○ <sup>*1</sup>	×	×
處理中 (Busy)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×
處理完成 (Complete)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×
發生錯誤 (Error)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×

\*1. 僅限 M、G、D 暫存器

\*2. C、# 暫存器除外

關於輸入輸出項目、參數詳情和程式範例，請參照以下手冊。

 MP2000 系列 使用者手冊 階梯圖程式篇 (資料編號：SI-C887-1.2)

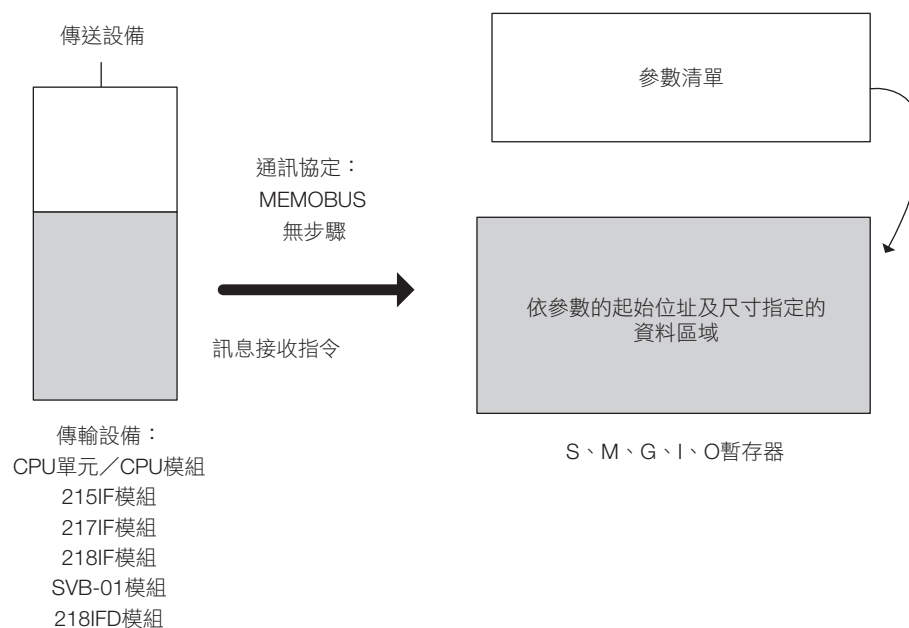


## 訊息接收（擴展）（MSG-RCVE）

依據傳輸設備種類，從指定線路上的對方端接收訊息。在完成接收（訊息接收完成為 ON）前，請讓訊息接收指令保持為 ON。支援下列傳輸設備和通訊協定。

傳輸設備：CPU 單元／CPU 模組、215IF 模組、217IF 模組、218IF 模組、SVB-01 模組、218IFD 模組

通訊協定：MEMOBUS 通訊協定，無步驟



（註）傳輸完成後，訊息接收完成便會轉為 ON。  
在此之前，請讓訊息接收指令保持為 ON。

基本動作如同 MSG-RCV 指令，但平常請使用符合 MP3000 規格的 MSG-RCVE 指令。

MSG-RCV 指令是符合 MP2000 的指令。以下的暫存器存取有所不同。

暫存器名稱	MSG-RCVE 的存取範圍		MSG-RCV 的存取範圍	
系統暫存器	SW00000 ~ 65534	RW	-	-
保留暫存器	MW0000000 ~ 1048575	RW	MW0000000 ~ 0065534	RW
資料暫存器	GW0000000 ~ 2097151	RW	-	-
輸入暫存器	IW00000 ~ 17FFF	R	IW00000 ~ 0FFFF	R
輸出暫存器	OW00000 ~ 17FFF	RW	-	-

（註）R：僅可讀取、RW：可讀取／寫入

## 格式

格式如下所示。

MSG-RCVE	
[B] Execute	[B] Busy
MB000000	MB000002
[B] Abort	[B] Complete
MB000001	MB000003
[W] Dev-Typ	[B] Error
MW000001	MB000004
[W] Pro-Typ	
MW000002	
[W] Cir-No	
MW000003	
[W] Ch-No	
MW000004	
[A] Param	
MA000010	


圖示：  
 鍵輸入：MSG-RCVE

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
接收執行指令 (Execute)	○	×	×	×	×	×	×	×	×
接收強制中止指令 (Abort)	○	×	×	×	×	×	×	×	×
傳輸設備種類 (Dev-Typ)	×	○	×	×	×	×	×	○	○
傳輸協定 (Pro-Typ)	×	○	×	×	×	×	×	○	○
線路編號 (Cir-No)	×	○	×	×	×	×	×	○	○
傳輸緩衝器頻道編號 (Ch-No)	×	○	×	×	×	×	×	○	○
參數清單開頭位址 (Param)	×	×	×	×	×	×	○ <sup>*1</sup>	×	×
處理中 (Busy)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×
處理完成 (Complete)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×
發生錯誤 (Error)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×

\*1. 僅限 M、G、D 暫存器

\*2. C、# 暫存器除外

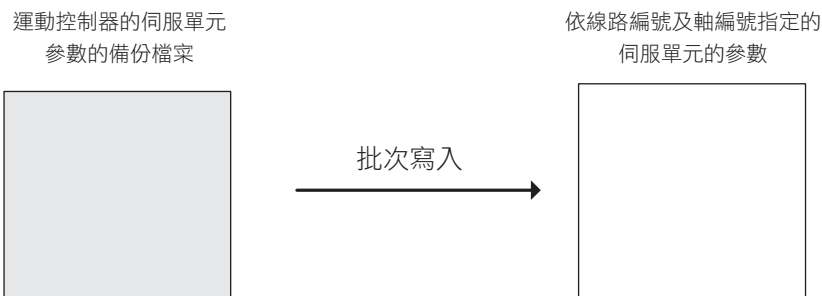
關於輸入輸出項目、參數詳情和程式範例，請參照以下手冊。

 MP3000 系列 通訊功能 使用者操作手冊 (資料編號：SIJP C880725 12)

## 伺服單元參數寫入 (MLNK-SVW)

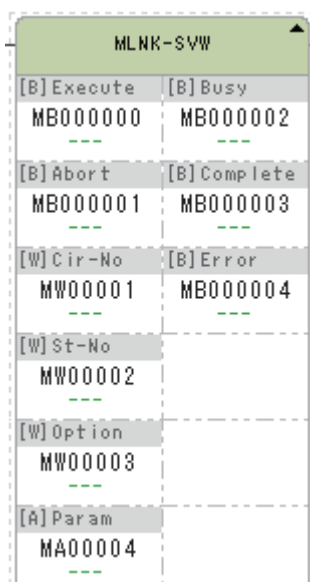
將運動控制器內儲存的參數一概儲存至指定線路編號以及軸編號的伺服單元，作為伺服單元參數備份檔案。因使用 MLNK-SVW 指令，即使沒有 MPE720，亦可僅靠階梯圖處理寫入伺服單元參數。

用於更換伺服單元等時相當方便的指令。



### 格式

格式如下所示。



圖示：**MLNK-SVW**

鍵輸入：MLNK-SVW

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
寫入指令 (Execute)	○	×	×	×	×	×	×	×	×
寫入處理強制中斷指令 (Abort)	○	×	×	×	×	×	×	×	×
線路編號 (Cir-No)	×	○	×	×	×	×	×	×	×
軸編號 (St-No)	×	○	×	×	×	×	×	×	×
選項設定 (Option)	×	○	×	×	×	×	×	×	×
參數表起始位址 (Param)	×	×	×	×	×	×	○ <sup>*1</sup>	×	×
寫入中 (Busy)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×
寫入結束 (Complete)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×
錯誤 (Error)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×

\*1. 僅限 M、D 暫存器

\*2. C、# 暫存器除外

以下說明各輸入輸出項目的內容。

輸出輸入項目	內容	輸出輸入
寫入指令 (Execute)	ON 則開始寫入伺服單元的參數。	IN
寫入處理強制中斷指令 (Abort)	ON 則強制中斷寫入處理。	IN
線路編號 (Cir-No)	目標線路編號 (1 ~ 16)	IN
軸編號 (St-No)	對象軸編號 1 ~ 16	IN
選項設定 (Option)	指令選項 Bit 設定 Bit E: ID 確認有效/無效設定 0 (有效)、1 (無效) Bit F: 版本確認有效/無效設定 0 (有效)、1 (無效) 未使用其他位元。即使設定也會遭到無視。	IN
參數表起始位址 (Param)	使用函數工作起始位址	IN
寫入中 (Busy)	伺服單元參數寫入中為 ON。	OUT
寫入結束 (Complete)	伺服單元參數寫入結束，僅 1 次掃描為 ON。	OUT
錯誤 (Error)	發生錯誤時，僅 1 掃描為 ON。 (錯誤內容會輸出至 PARAM00、PARAM01。)	OUT

選項設定內容如下所示。

Bit	內容
0 ~ D	未使用 (無視設定。)
E	ID 確認有效/無效設定 (0:有效 / 1:無效) 原本檔案的 ID 資訊與寫入對象的 ID 資訊不同時，產生錯誤 (ID 資訊不一致)。 選擇「1 (關閉)」，將不偵測此錯誤，可進行寫入處理。 用於對步進馬達驅動程式寫入時。 選擇「1 (關閉)」，將不檢查機種資訊，可能發生寫入不同機種的參數而導致問題，請千萬小心。 使用在線編輯或已儲存的伺服單元參數檔案執行函數時，也同樣會發生錯誤 (ID 資訊不一致)。此時請於確認沒有問題後，選擇「1 (無效)」。
F	版本確認有效/無效設定 (0:有效 / 1:無效) 原本的伺服單元 (通訊 I/F) 版本與寫入對象版本不同時，發生錯誤 (版本不一致)。 若版本不同，伺服單元參數的項目和設定範圍可能有所不同。請於確認沒有問題後，選擇「1 (無效)」。 可執行寫入處理。 使用在線編輯或已儲存的伺服單元參數檔案執行函數時，也同樣會發生錯誤 (版本不一致)。此時請於確認沒有問題後，選擇「1 (無效)」。

### ◆ 使用函數工作詳情

說明使用函數的工作詳情。參數 No. 為起始位址起算的字組偏移量。

例 起始位址為 MA00100 時，在設定 PARAM 05 時，請將數值設定為 MW00105。

參數 No.	IN/OUT	內容
PARAM 00	OUT	處理結果
PARAM 01	OUT	錯誤代碼
PARAM 02	OUT	Cir-No 複製
PARAM 03	OUT	ST-No 複製
PARAM 04	SYS	系統用 #1
PARAM 05	SYS	系統用 #2
PARAM 06	SYS	系統用 #3

### ■ 處理結果 (PARAM 00)

將處理結果輸出至伺服單元。

- 0000H：處理中 (Busy)
- 1000H：處理完成 (Complete)
- 8□□□H：發生錯誤 (Error)

錯誤分類如下所示。

錯誤代碼	內容
8100H	預約
8200H	位址設定錯誤 (資料位址設定在範圍外。)
8300H	預約
8400H	線路編號設定錯誤 (線路編號超出設定範圍。)
8500H	預約
8600H	軸編號設定錯誤 (軸編號超出設定範圍。)
8700H	預約
8800H	通訊 I/F 任務錯誤 (通訊 I/F 任務傳回錯誤。)
8900H	預約
8A00H	重複執行函數錯誤 (同時執行複數個 MLNK-SVW 函數。或執行了 MLNK-SVR 函數。)

#### ■ 錯誤代碼 (PARAM 01)

輸出通訊 I/F 任務的錯誤代碼。僅當處理結果 (PARAM00) 為 8800H 時有效。

錯誤代碼	內容
0000H	預約
0001H	無伺服單元參數備份檔
0002H	錯誤的備份檔
0003H	ID 訊息不一致
0004H	版本不一致
0005H	模組異常
0006H	伺服單元控制器部指令重複錯誤
0007H	通訊異常
0008H	非定義指令
0009H	錯誤參數
000AH	系統內部錯誤

#### ■ Cir-No 複製 (PARAM 02)

輸入資料 Cir-No 的複製。

#### ■ St-No 複製 (PARAM 03)

輸入資料 St-No 的複製。

#### ■ 系統用 #1(PARAM 04)

用於系統。以使用者程式啟動電源的首次掃描，請設定為「0000H」。除此之外的時間點，請勿進行變更。

#### ■ 系統用 #2(PARAM 05)

用於系統。以使用者程式啟動電源的首次掃描，請設定為「0000H」。除此之外的時間點，請勿進行變更。

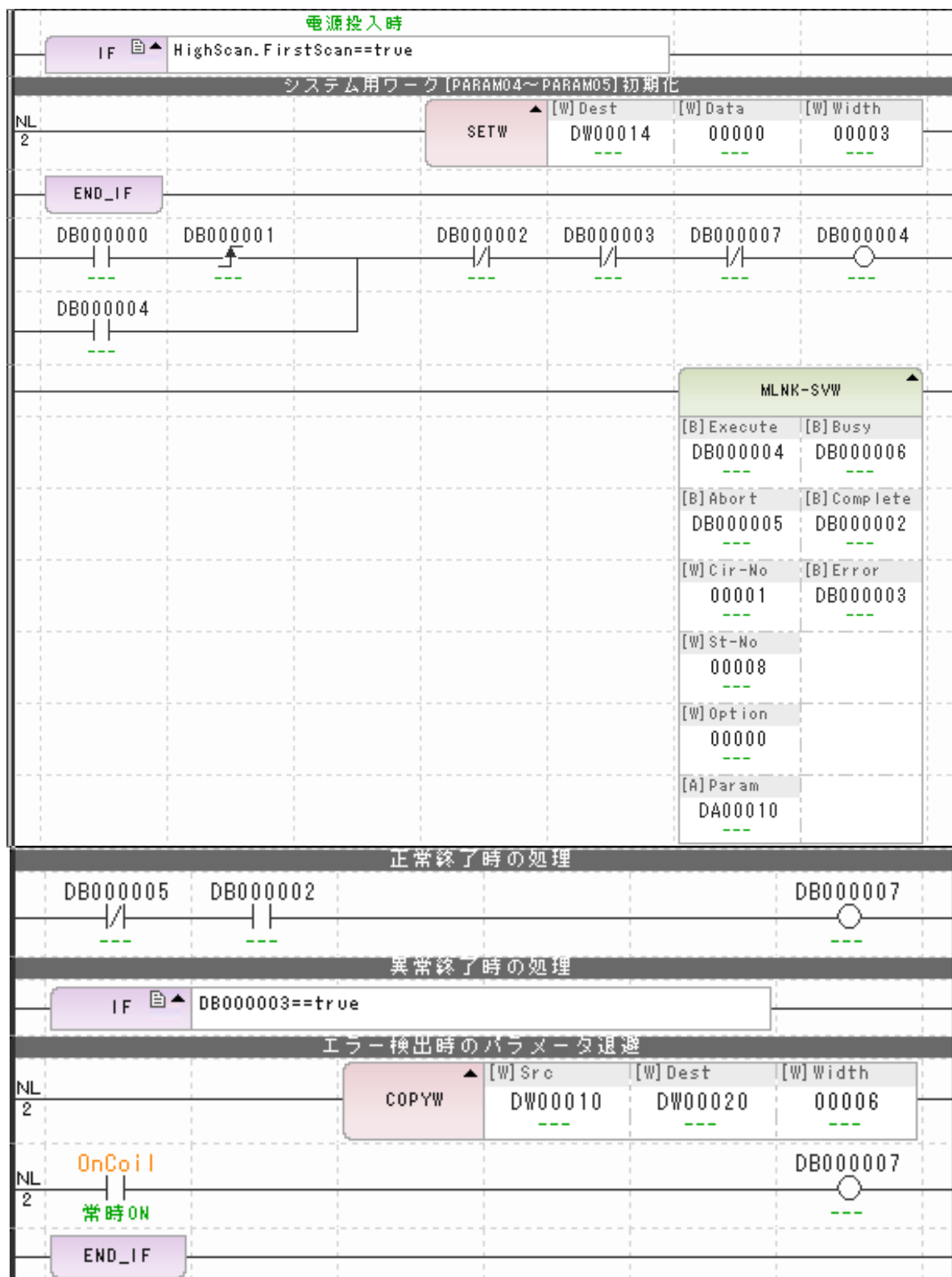
#### ■ 系統用 #3(PARAM 06)

用於系統。以使用者程式啟動電源的首次掃描，請設定為「0000H」。除此之外的時間點，請勿進行變更。

## 程式範例

對伺服單元寫入參數時的程式範例，如下所示。

運動控制器中如存在伺服單元參數的備份檔時，將因為 DB000000 設為 ON，對模組構成定義中 MECHATROLINK 線路編號為 1、MECHATROLINK 詳細定義中 ST#8 的伺服單元參數執行只有 1 次的寫入。



## 伺服單元參數載入 (MLNK-SVR)

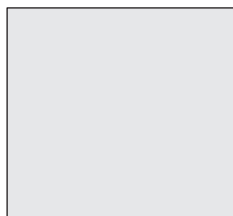
由指定線路編號以及軸編號的伺服單元 RAM 區域一概載入參數，覆蓋儲存至運動控制器儲存的伺服單元參數備份檔。因使用 MLNK-SVR 指令，即使沒有 MPE720，亦可僅靠階梯圖處理載入伺服單元參數。

用於更換伺服單元等時相當方便的指令。

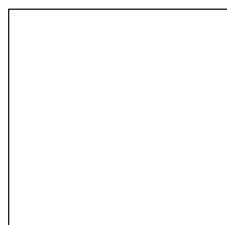


- 執行 MLNK-SVR 指令前，請檢查運動控制器軟體版本為 1.23 或更新版本，且目前使用的 MPE720 軟體版本為 Ver7.34 或更新版本。
- MLNK-SVR 指令將由伺服單元的 RAM 區域載入參數。因此，當伺服單元的 RAM 區域與非揮發性記憶體的參數設定值有差異，則伺服單元電源開與關將使控制器載入的參數設定值與伺服單元 RAM 區域的參數設定值發生不一致，敬請注意。

運動控制器的伺服單元  
參數的備份檔案



依線路編號及軸編號指定的  
伺服單元的參數




批次讀取



### 格式

格式如下所示。

MLNK-SVR	
[B] Execute	[B] Busy
MB000000	MB000002
[B] Abort	[B] Complete
MB000001	MB000003
[W] Cir-No	[B] Error
MW00001	MB000004
[W] St-No	
MW00002	
[W] Option	
MW00003	
[A] Param	
MA00004	

圖示：  
鍵輸入：MLNK-SVR



輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
讀取指令 (Execute)	○	×	×	×	×	×	×	×	×
載入處理強制中斷指令 (Abort)	○	×	×	×	×	×	×	×	×
線路編號 (Cir-No)	×	○	×	×	×	×	×	○	○
軸編號 (St-No)	×	○	×	×	×	×	×	○	○
選項設定 (Option)	×	○	×	×	×	×	×	○	○
參數表起始位址 (Param)	×	×	×	×	×	×	○ <sup>*1</sup>	×	×
讀取中 (Busy)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×
載入結束 (Complete)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×
錯誤 (Error)	○ <sup>*2</sup>	×	×	×	×	×	×	×	×

\*1. 僅限 M、D 暫存器

\*2. C、# 暫存器除外

以下說明各輸入輸出項目的內容。

輸出輸入項目	內容	輸出輸入
讀取指令 (Execute)	ON 則開始載入伺服單元的參數。 執行指令中需保持為 ON。	IN
載入處理強制中斷指令 (Abort)	ON 則強制中斷載入處理。	IN
線路編號 (Cir-No)	目標線路編號 (1 ~ 16)	IN
軸編號 (St-No)	對象軸編號 (1 ~ 32)	IN
選項設定 (Option)	指令選項 Bit 設定 Bit E: ID 確認有效/無效設定 0 (有效)、1 (無效) Bit F: 版本確認有效/無效設定 0 (有效)、1 (無效) 未使用其他位元。即使設定也會遭到無視。	IN
參數表起始位址 (Param)	使用函數工作起始位址	IN
讀取中 (Busy)	伺服單元參數載入中為 ON。	OUT
載入結束 (Complete)	伺服單元參數載入結束，僅 1 次掃描為 ON。	OUT
錯誤 (Error)	發生錯誤時，僅 1 掃描為 ON。 (錯誤內容會輸出至 PARAM00、PARAM01。)	OUT

選項設定內容如下所示。

Bit	內容
0 ~ D	未使用 (無視設定。)
E	ID 確認有效/無效設定 (0:有效 / 1:無效) 原本檔案的 ID 資訊與載入對象的 ID 資訊不同時, 產生錯誤 (ID 資訊不一致)。 選擇「1 (關閉)」, 將不偵測此錯誤, 可進行載入處理。 用於對步進馬達驅動程式載入時。 選擇「1 (關閉)」, 將不檢查機種資訊, 可能發生載入不同機種的參數而導致問題, 請千萬小心。 更換伺服單元時, 請選擇「1 (關閉)」。 使用在線編輯或已儲存的伺服單元參數檔案執行函數時, 也同樣會發生錯誤 (ID 資訊不一致)。此時請於確認沒有問題後, 選擇「1 (無效)」。
F	版本確認有效/無效設定 (0:有效 / 1:無效) 原本的伺服單元 (通訊 I/F) 版本與載入對象版本不同時, 發生錯誤 (版本不一致)。 若版本不同, 伺服單元參數的項目和設定範圍可能有所不同。請於確認沒有問題後, 選擇「1 (無效)」, 可執行載入處理。 使用在線編輯或已儲存的伺服單元參數檔案執行函數時, 也同樣會發生錯誤 (版本不一致)。此時請於確認沒有問題後, 選擇「1 (無效)」。

### ◆ 使用函數工作詳情

說明使用函數的工作詳情。參數 No. 為起始位址起算的字組偏移量。

**例** 起始位址為 MA00100 時, 在設定 PARAM 05 時, 請將數值設定為 MW00105。

參數 No.	IN/OUT	內容
PARAM 00	OUT	處理結果
PARAM 01	OUT	錯誤代碼
PARAM 02	OUT	Cir-No 複製
PARAM 03	OUT	ST-No 複製
PARAM 04	SYS	系統用 #1
PARAM 05	SYS	系統用 #2
PARAM 06	SYS	系統用 #3

### ■ 處理結果 (PARAM 00)

將處理結果輸出至伺服單元。

- 0000H: 處理中 (Busy)
- 1000H: 處理完成 (Complete)
- 8□□□H: 發生錯誤 (Error)

錯誤分類如下所示。

錯誤代碼	內容
8100H	預約
8200H	預約
8300H	預約
8400H	線路編號設定錯誤 (線路編號超出設定範圍。)
8500H	預約
8600H	軸編號設定錯誤 (軸編號超出設定範圍。)

(接下頁)

(承上頁)

錯誤代碼	內容
8700H	預約
8800H	通訊 I/F 任務錯誤 (通訊 I/F 任務傳回錯誤。)
8900H	預約
8A00H	重複執行函數錯誤 (同時執行複數個 MLNK-SVR 函數。或執行了 MLNK-SVW 函數。)

#### ■ 錯誤代碼 (PARAM 01)

輸出通訊 I/F 任務的錯誤代碼。僅當處理結果 (PARAM00) 為 8800H 時有效。

錯誤代碼	內容
0000H	無錯誤
0001H	無伺服單元參數備份檔
0002H	錯誤的備份檔
0003H	ID 訊息不一致
0004H	版本不一致
0005H	模組異常
0006H	伺服單元控制器部指令重複錯誤
0007H	通訊異常
0008H	預約
0009H	預約
000AH	系統內部錯誤

#### ■ Cir-No 複製 (PARAM 02)

輸入資料 Cir-No 的複製。

#### ■ St-No 複製 (PARAM 03)

輸入資料 St-No 的複製。

#### ■ 系統用 #1(PARAM 04)

用於系統。以使用者程式啟動電源的首次掃描，請設定為「0000H」。除此之外的時間點，請勿進行變更。

#### ■ 系統用 #2(PARAM 05)

用於系統。以使用者程式啟動電源的首次掃描，請設定為「0000H」。除此之外的時間點，請勿進行變更。

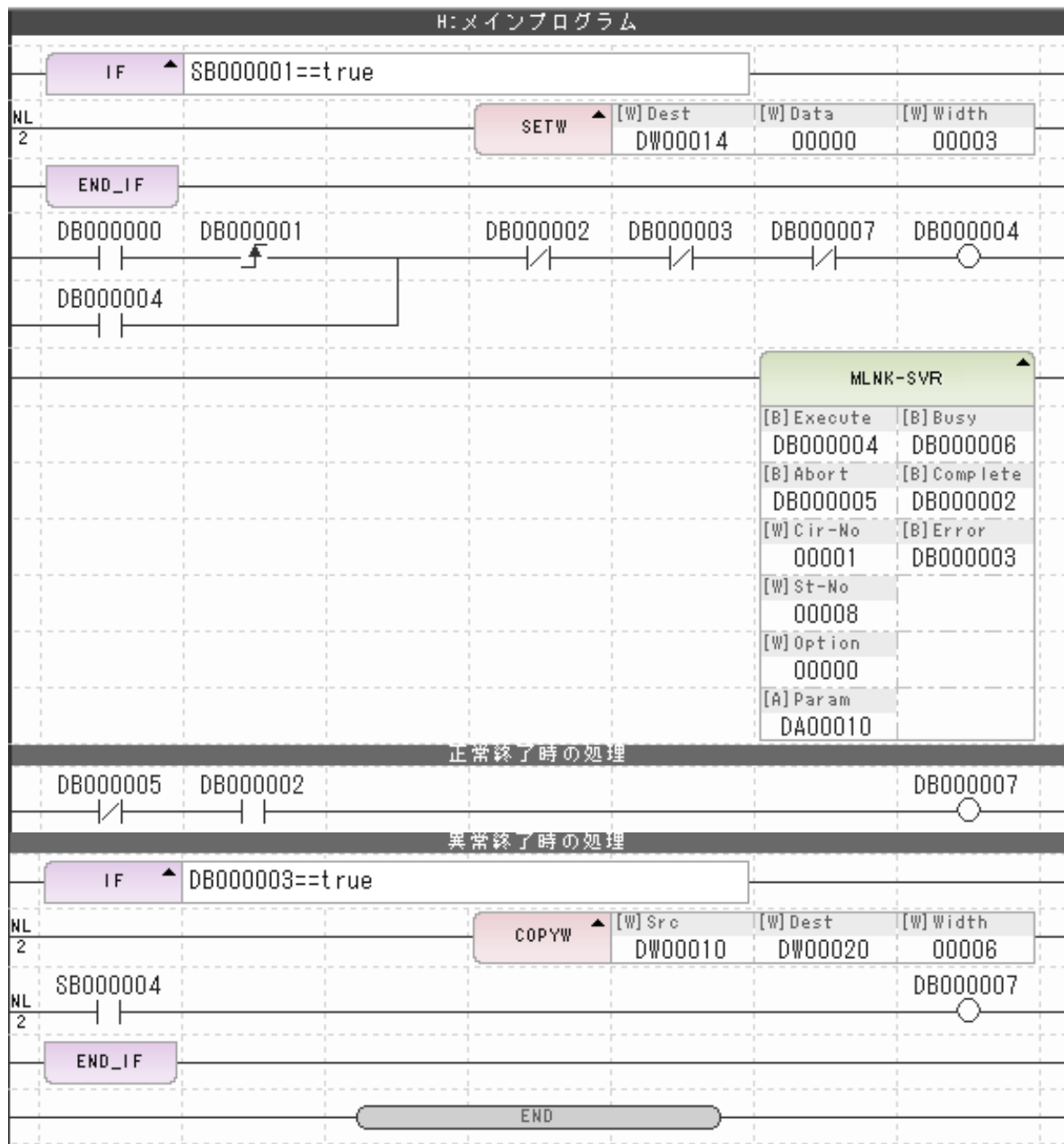
#### ■ 系統用 #3(PARAM 06)

用於系統。以使用者程式啟動電源的首次掃描，請設定為「0000H」。除此之外的時間點，請勿進行變更。

### 程式範例

由伺服單元載入參數時的程式範例，如下所示。

運動控制器中如存在伺服單元參數的備份檔時，將因為 DB000000 設為 ON，對模組構成定義中 MECHATROLINK 線路編號為 1、MECHATROLINK 詳細定義中 ST#8 的伺服單元參數執行只有 1 次的載入。



## 快閃操作 (FLASH-OP)

運動控制器內的快閃記憶體可與 RAM 上的資料比對，或將 RAM 上的資料儲存至快閃記憶體。因使用 FLASH-OP 指令，即使沒有 MPE720，亦可僅靠階梯圖快閃儲存。

是使用 MLNK-SVR 指令讀出伺服單元的參數後快閃儲存時的方便指令。



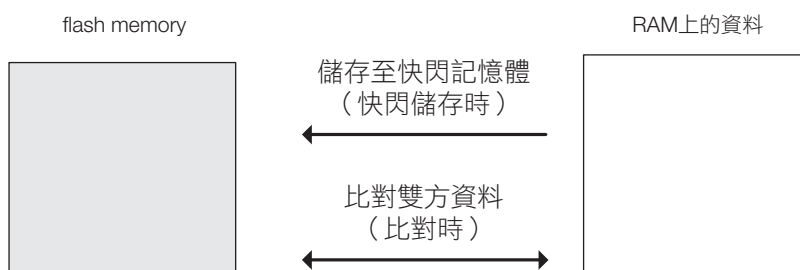
重要

執行 FLASH-OP 指令前，請檢查運動控制器軟體版本為 1.23 或更新版本，且目前使用的 MPE720 軟體版本為 Ver7.34 或更新版本。



註記

請勿在快閃記憶體儲存結束前切斷運動控制器電源。  
若在快閃儲存中切斷運動控制器電源，將導致資料消失。  
若在本狀態下重新開啟運動控制器的電源，將以出廠設定狀態啟動運動控制器。



### 格式

格式如下所示。

FLASH-OP	
[B] Execute MB000000 ---	[B] Busy MB000002 ---
[B] Reserve MB000001 ---	[B] Complete MB000003 ---
[W] Option MW000001 ---	[B] Error MB000004 ---
[A] Param MA000010 ---	[W] ErrorCode MW000002 ---
	[W] Status MW000003 ---

圖示：**FLASH-OP**  
鍵輸入：FLASH-OP

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
快閃記憶體操作指令 (Execute)	○	×	×	×	×	×	×	×	×
系統預約 (Reserve)	○	×	×	×	×	×	×	×	×
選項設定 (Option)	×	○	×	×	×	×	×	○	○
參數表起始位址 (Param)	×	×	×	×	×	×	○*1	×	×
執行中 (Busy)	○*2	×	×	×	×	×	×	×	×
執行完成 (Complete)	○*2	×	×	×	×	×	×	×	×
錯誤 (Error)	○*2	×	×	×	×	×	×	×	×
錯誤代碼 (Error Code)	×	○	×	×	×	×	×	○	○
比較結果 (Status)	×	○	×	×	×	×	×	○	○

\*1. 僅限 M、G、D 暫存器

\*2. C、# 暫存器除外

以下說明各輸入輸出項目的內容。

輸出輸入項目	內容	輸出輸入
快閃記憶體操作指令 (Execute)	ON 時開始快閃操作指令。 執行指令中需保持為 ON。	IN
系統預約 (Reserve)	—	—
選項設定 (Option)	指令選項 Bit 設定 Bit D：CPU 動作 0 (CPU RUN 狀態時執行)，1 (CPU STOP 後執行) Bit E：校驗關閉/啟用設定 0 (關閉)，1 (啟用) Bit F：快閃儲存關閉/啟用設定 0 (關閉)，1 (啟用) 未使用其他位元。即使設定也會遭到無視。	IN
參數表起始位址 (Param)	使用函數工作起始位址	IN
執行中 (Busy)	快閃操作中為 ON。	OUT
執行完成 (Complete)	快閃操作完畢，僅 1 次掃描時 ON。	OUT
錯誤 (Error)	發生錯誤時，僅 1 掃描為 ON。	OUT
錯誤代碼 (Error Code)	發生錯誤時，僅 1 掃描為 ON。	OUT
比較結果 (Status)	校驗完畢時，僅輸出 1 次掃描的比對結果。 除此之外輸出 0。 比對結果 1 (無差異)，2 (有差異)	OUT

選項設定內容如下所示。

Bit	內容
0 ~ C	未使用 (無視設定。)
D	快閃操作執行時的 CPU 動作 (0: RUN, 1: STOP) 選擇快閃操作執行時的 CPU 動作狀態。 選擇「1(STOP)」時，快閃操作執行時 CPU 將會 STOP，等到快閃操作執行完成後，CPU 變為 RUN。 選擇「1(STOP)」時，CPU 會停止，請確認不會發生問題後再行選擇。
E	校驗關閉/啟用設定 (0: 關閉/1: 啟用) 選擇是否比對快閃記憶體與 RAM 上的資料。 選擇「1 (啟用)」，將比對快閃記憶體與 RAM 上的資料。 將校驗無效/有效設定和快閃記憶體儲存無效/有效兩者都設為有效時，將比較快閃記憶體和 RAM 上的資料，當檢測到差異時，將 RAM 上的資料儲存至快閃記憶體。
F	快閃儲存關閉/啟用設定 選擇是否執行快閃儲存。 選擇「1 (啟用)」，RAM 上的資料將儲存至快閃記憶體。 將校驗無效/有效設定和快閃記憶體儲存無效/有效兩者都設為有效時，將比較快閃記憶體和 RAM 上的資料，當檢測到差異時，將 RAM 上的資料儲存至快閃記憶體。

### ◆ 使用函數工作詳情

說明使用函數的工作詳情。參數 No. 為起始位址起算的字組偏移量。

**例** 開頭位址為 MA00100 時，要設定 PARAM 01，請對 MW00101 設定數值。

參數 No.	IN/OUT	內容
PARAM 00	SYS	系統用 #1
PARAM 01	SYS	系統用 #2

#### ■ 系統用 #1(PARAM 00)

用於系統。以使用者程式啟動電源的首次掃描，請設定為「0000H」。除此之外的時間點，請勿進行變更。

#### ■ 系統用 #2(PARAM 01)

用於系統。以使用者程式啟動電源的首次掃描，請設定為「0000H」。除此之外的時間點，請勿進行變更。

#### ■ 錯誤代碼

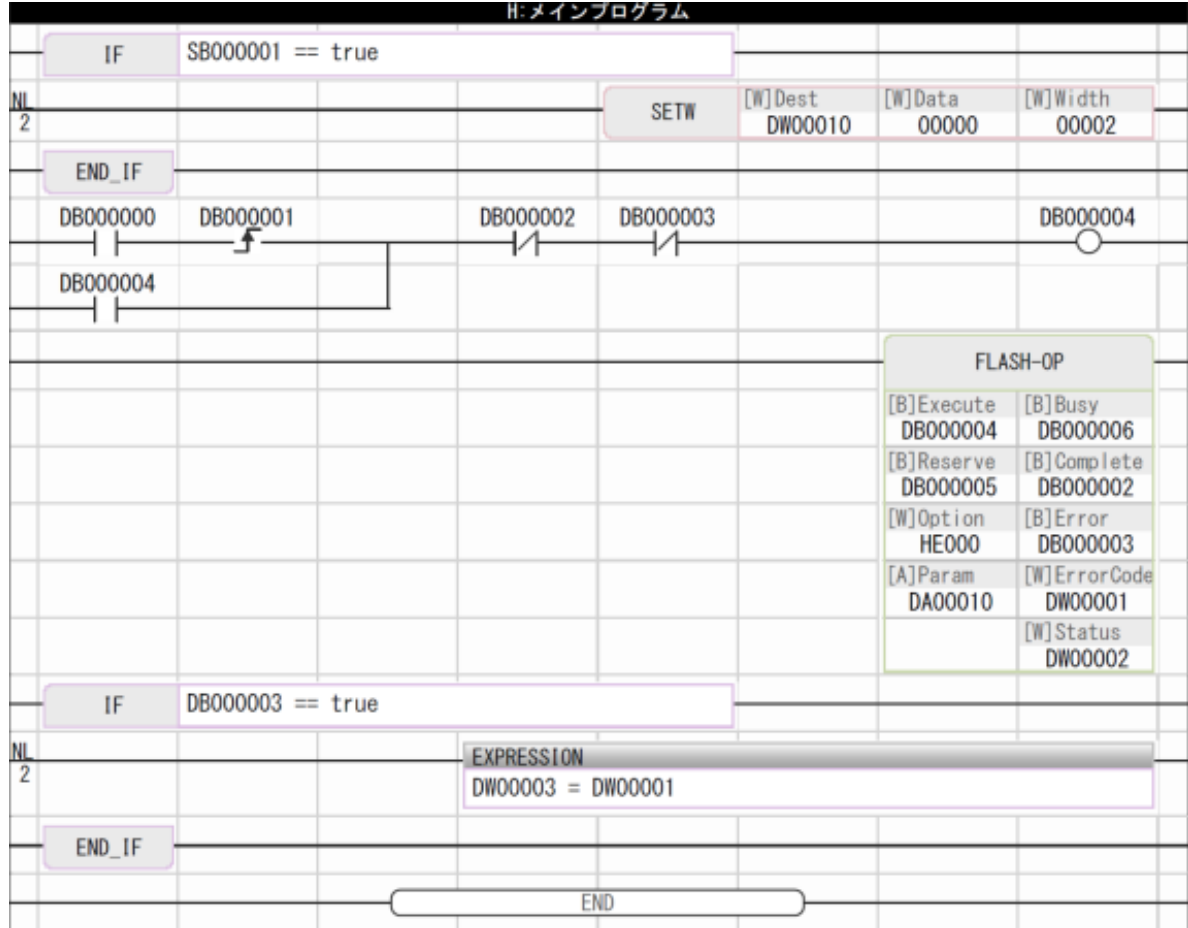
將輸出下列錯誤代碼。

錯誤代碼	內容
0000H	正常
0001H	指令重覆
0002H	系統內部錯誤
0003H	未指定快閃儲存或校驗

### 程式範例

快閃儲存時的程式範例，如下所示。

因 DB000000 為 ON，執行校驗。快閃記憶體與 RAM 上的資料有差異時，CPU 將 STOP，將 RAM 上的資料快閃儲存。快閃儲存完畢後，CPU 自動恢復為 RUN。





## 寫入運動暫存器 (MOTREG-W)

用於存取所指定的運動暫存器的系統函數。

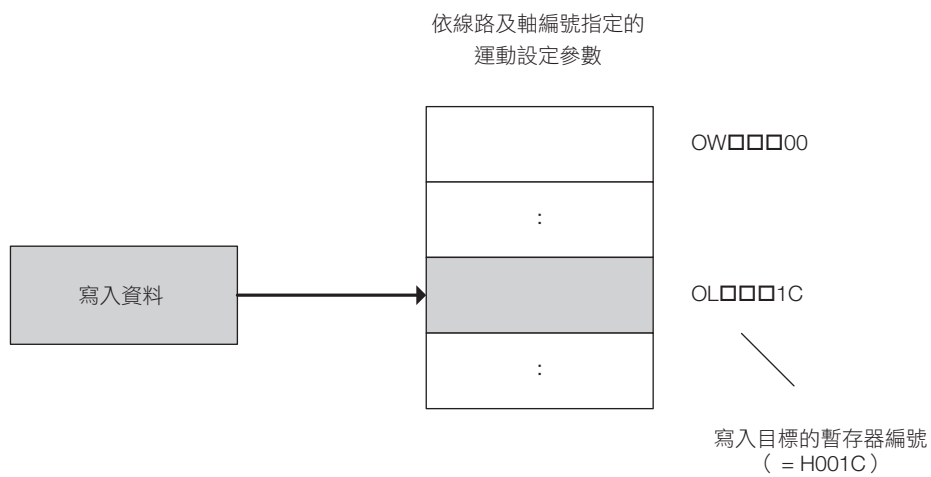
指定線路編號、軸編號、暫存器編號，對運動暫存器寫入數值。

以運動設定參數為對象。



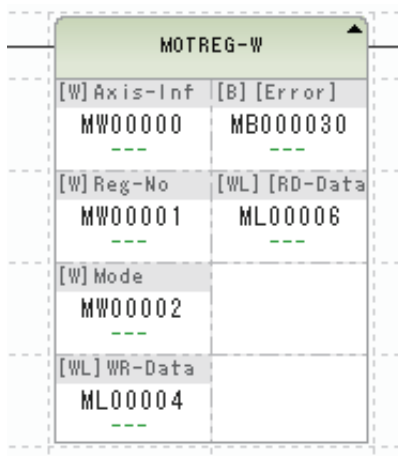
註記

將相同運動設定參數儲存至不同線路編號以及軸編號的複數軸時啟用。以 STORE 指令或 EXPRESSION 指令寫入時，需依據線路編號或軸編號考量補償值。



## 格式

格式如下所示。



圖示：**MOT REG-W**  
鍵輸入：MOTREG-W

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
軸資訊 (Axis-Inf)	×	○	×	×	×	×	×	×	×
暫存器編號 (Reg-No)	×	○	×	×	×	×	×	×	×
模式 (Mode)	×	○	×	×	×	×	×	×	×
寫入資料 (WR-Data)	×	○	○	×	×	×	×	×	×
錯誤 (Error)	○*	×	×	×	×	×	×	×	×
讀取資料 (RD-Data)	×	○*	○*	×	×	×	×	×	×

\* 除 C、# 暫存器，其餘可省略

以下說明各輸入輸出項目的內容。

輸出輸入項目	內容	輸出輸入
軸資訊 (Axis-Inf)	線路編號，軸編號 上位位元組：線路編號 (01 ~ 10) (hex) 下位位元組：軸編號 (01 ~ 10) (hex)	IN
暫存器編號 (Reg-No)	指定整數型時：0000 ~ 007F(hex) 指定 2 倍長整數型時：0000 ~ 007E(hex)	IN
模式 (Mode)	存取類別、存取大小 • 上位位元組：存取類別 0：對指定暫存器寫入 WR-Data 1：將指定暫存器與 WR-Data 的邏輯或寫入指定暫存器 2：將指定暫存器與 WR-Data 的邏輯乘積寫入指定暫存器 前述以外：對指定暫存器寫入 WR-Data。 • 下位位元組：存取規格 0：整數型 1：2 倍長整數型 上述之外：整數型	IN
寫入資料 (WR-Data)	模式的存取大小為整數型，輸入類型為 2 倍長整數型時，僅使用下位字元。	IN
錯誤 (Error)	發生錯誤 (發生錯誤時為 ON) 原因 線路編號、軸編號、暫存器編號為範圍外或沒有模組，無法寫入／讀取暫存器。 發生錯誤時輸出 0 至 RD-Data。	OUT
讀取資料 (RD-Data)	寫入後有資料讀取。 指定整數型時，輸出符號擴展資料。	OUT

## 程式範例

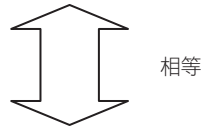
在線路 3 的軸 10 上的 STEP 移動量 (OL□□□44) 將寫入資料 (ML00000) 數值寫入時的程式範例，如下所示。

請進行下列設定。

- 軸資訊 = H030A (線路 3、軸 10)
- 暫存器編號 = H0044
- 模式 = H0001 (2 倍長整數型)

MOTREG-W	
[W] Axis-Inf	[B] [Error]
H030A	DB000000
778	0
---	---
[W] Reg-No	[WL] [RD-Data]
H0044	ML00002
68	10000
---	---
[W] Mode	
H0001	
	1
[WL] WR-Data	
ML00000	
10000	
---	

直接指定暫存器編號並以 STORE 指令儲存，也能得到相同結果。



STORE	[WLRD] Src	[WLRD] Dest
	ML00000	OL94C4
	10000	10000
	---	---

## 載入運動暫存器 (MOTREG-R)

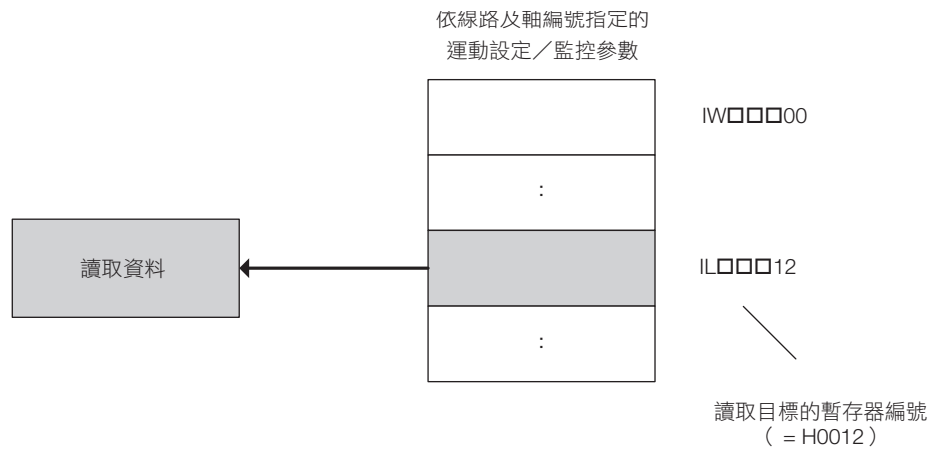
用於存取所指定的運動暫存器的系統函數。

指定線路編號、軸編號、暫存器編號，讀出運動暫存器數值。

以運動設定參數、運動監控參數為對象。



將相同運動設定參數自不同線路編號以及軸編號的複數軸中讀取時啟用。以 STORE 指令或 EXPRESSION 指令讀取時，需依據線路編號或軸編號考量補償值。



## 格式

格式如下所示。

MOTREG-R	
[W] Axis-Inf	[B] [Error]
MW00000	MB000030
[W] Reg-No	[WL] RD-Data
MW00001	ML00004
[W] Mode	
MW00002	

圖示： MOT  
REG-R

鍵輸入： MOTREG-R

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
軸資訊 (Axis-Inf)	×	○	×	×	×	×	×	×	×
暫存器編號 (Reg-No)	×	○	×	×	×	×	×	×	×
模式 (Mode)	×	○	×	×	×	×	×	×	×
錯誤 (Error)	○*	×	×	×	×	×	×	×	×
讀取資料 (RD-Data)	×	○*	○*	×	×	×	×	×	×

\* 除 C、# 暫存器，其餘可省略

以下說明各輸入輸出項目的內容。

輸出輸入項目	內容	輸出輸入
軸資訊 (Axis-Inf)	線路編號，軸編號 上位位元組：線路編號 (01 ~ 10) (hex) 下位位元組：軸編號 (01 ~ 10) (hex)	IN
暫存器編號 (Reg-No)	指定整數型時：0000 ~ 007F(hex) 指定 2 倍長整數型時：0000 ~ 007E(hex)	IN
模式 (Mode)	暫存器類別、存取大小 • 上位位元組：暫存器類別 0：I 暫存器（運動監控參數） 1：O 暫存器（運動設定參數） 前述以外：I 暫存器 • 下位位元組：存取規格 0：整數型 1：2 倍長整數型 上述之外：整數型	IN
錯誤 (Error)	發生錯誤（發生錯誤時為 ON）原因 線路編號、軸編號、暫存器編號為範圍外或沒有模組，無法寫入／讀取暫存器。 發生錯誤時輸出 0 至 RD-Data。	OUT
讀取資料 (RD-Data)	指定整數型時，輸出符號擴展資料。	OUT

### 程式範例

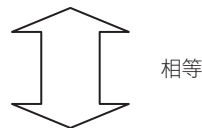
讀取線路 1 的軸 2 上的機械座標系回授位置 (IL8096) 的程式範例，如下所示。

請進行下列設定。

- 軸資訊 = H0102 (線路 1、軸 2)
- 暫存器編號 = H0016
- 模式 = H0001 (運動監控參數，2 倍長整數型)

回線 1、軸 2 の機械座標系フィードバック位置 (IL8096) を読み出す

MOTREG-R	
[W] Axis-Inf	[B] [Error]
H0102	DB000000
258	1
---	
[W] Reg-No	[WL] RD-Data
H0016	DL00002
22	4965999
---	
[W] Mode	
H0001	
	1



直接指定暫存器編號並以 STORE 指令儲存至 DL00002，也能得到相同結果。

	[W] [W] [W] Src	[W] [W] [W] Dest
STORE	IL8096	DL00002
	4965999	4965999
	---	---

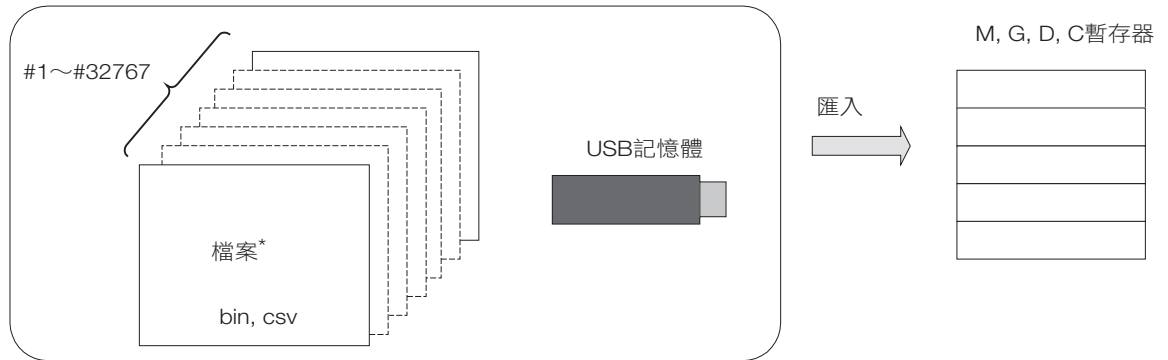
## 匯入 (IMPORT/IMPORTL)

由 USB 隨身碟匯入暫存器資料，寫入暫存器。

匯入資料的格式，可選擇 2 進制資料 (bin)、CSV 資料 (csv)。

匯入的對象暫存器，可選擇 M 暫存器、G 暫存器、D 暫存器或 C 暫存器。

IMPORT/IMPORTL 指令可同時平行執行 2 個指令。



\* 將下列 USB 隨身碟中的檔案匯入。

¥MP\_DATA¥DAT00001.BIN (CSV)

:

¥MP\_DATA¥DAT32767.BIN (CSV)

### 格式

格式如下所示。

IMPORT	
[B] Execute	[B] Busy
MB000000	MB000002
[B] Abort	[B] Complete
MB000001	MB000003
[W] Drv-No	[B] Error
MW00001	MB000004
[W] Data-No	
MW00002	
[W] Size	
MW00003	
[W] Ch-No	
MW00004	
[A] Dest	
MA00100	
[A] Param	
MA00010	

圖示： IM PORT インポート IM PORTL インポート(拡張)

鍵輸入：IMPORT/IMPORTL

## 匯入 (IMPORT/IMPORTL)

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
匯入指令 (Execute)	○	×	×	×	×	×	×	×	×
匯入強制中止指令 (Abort)	○	×	×	×	×	×	×	×	×
驅動器編號 (Drv-No)	×	○	×	×	×	×	×	○	○
資料編號 (Data-No)	×	○	×	×	×	×	×	○	○
傳輸字組數 (Size)	×	○*1	○*2	×	×	×	×	○	○
平行執行頻道編號 (Ch-No)	×	○	×	×	×	×	×	○	○
傳輸目的地起始暫存器編號 (Dest)	×	×	×	×	×	×	○	×	×
參數表起始位址 (Param)	×	×	×	×	×	×	○	×	×
執行匯入中 (Busy)	○	×	×	×	×	×	×	×	×
匯入執行完畢 (Complete)	○	×	×	×	×	×	×	×	×
錯誤 (Error)	○	×	×	×	×	×	×	×	×

\*1. IMPORT 指令時

\*2. IMPORTL 指令時

以下說明各輸入輸出項目的內容。

輸出輸入項目	內容	輸出輸入
匯入指令 (Execute)	ON 則匯入指令開始。 執行指令中需保持為 ON。	IN
匯入強制中止指令 (Abort)	ON 則強制中斷匯入處理。	IN
驅動器編號 (Drv-No)	驅動器編號 (1 : USB 隨身碟)	IN
資料編號 (Data-No)	資料編號 (1 ~ 32767)	IN
傳輸字組數 (Size)	傳輸字元數 (IMPORT:1 ~ 32767 , IMPORTL:1 ~ 2147483647)	IN
平行執行頻道編號 (Ch-No)	平行執行頻道編號 (1 , 2)	IN
傳輸目的地起始暫存器編號 (Dest)	傳輸目標起始暫存器編號 (MA , GA , DA , CA)	IN
參數表起始位址 (Param)	參數表起始位址 (MA , GA , DA)	IN
執行匯入中 (Busy)	匯入執行時為 ON。	OUT
匯入執行完畢 (Complete)	匯入完畢後 ON。	OUT
錯誤 (Error)	發生錯誤而 ON。	OUT

## ◆ 參數詳情

說明參數詳情。

&lt;IMPORT&gt;

位址	資料類型	參數 No.	IN/OUT	內容
0	W	PARAM00	OUT	處理結果
1	W	PARAM01	IN	格式種類
2	W	PARAM02	IN	CSV 檔案時的檔案內偏移行數
3	W	PARAM03	IN	檔案內資料的字組偏移量
4	W	PARAM04	OUT	系統預約
5	W	PARAM05	OUT	系統預約



## &lt;IMPORTL&gt;

位址	資料類型	參數 No.	IN/OUT	內容
0	W	PARAM00	OUT	處理結果
1	W	PARAM01	IN	格式種類
2	L	PARAM02	IN	CSV 檔案時的檔案內偏移行數
4	L	PARAM03	IN	檔案內資料的字組偏移量
6	W	PARAM04	OUT	系統預約
7	W	PARAM05	OUT	系統預約

### ■ 處理結果 (PARAM00)

將報告 IMPORT/IMPORTL 指令的處理結果。

- 00□□H：處理中 (Busy)
- 10□□H：處理完成 (Complete)
- 8□□□H：發生錯誤 (Error)

錯誤分類如下所示。

錯誤代碼	內容
8101H	超出範圍錯誤 (驅動器編號)
8102H	超出範圍錯誤 (資料編號)
8103H	超出範圍錯誤 (傳輸數)
8104H	超出範圍錯誤 (平行執行頻道編號)
8105H	超出範圍錯誤 (傳輸目的地/傳輸來源暫存器編號)
8106H	超出範圍錯誤 (格式種類)
8107H	超出範圍錯誤 (開啟類型)
8108H	超出範圍錯誤 (檔案內資料的字組偏移量)
8109H	超出範圍錯誤 (參數表起始位址)
810AH	範圍外錯誤 (檔案內的偏移行數)
8201H	未安插 USB 隨身碟
8202H	檔案開啟錯誤
8203H	檔案搜尋錯誤
8204H	檔案寫入錯誤
8205H	檔案讀取錯誤
8206H	檔案關閉錯誤
8301H	由於超過可處理檔案數，因此無法處理
8302H	檔案輸入輸出逾時

### ■ 格式種類 (PARAM01)

設定匯入檔案的格式。

匯入 MPE720 的暫存器清單資料時，請設為 2。

1：由 2 進制檔案 (DAT□□□□□.BIN) 匯入。

□□□□□ 為資料編號 (Data-No) 所設定的數值。

2：由 CSV 檔案 (DAT□□□□□.CSV) 匯入。

□□□□□ 為資料編號 (Data-No) 所設定的數值。

### ■ 為 CSV 檔案時的檔案內偏移行數 (PARAM02)

設定 CSV 檔案時的偏移行數。

匯入 MPE720 的暫存器清單資料時，請指定為 2。

如為 2 進制檔案，將忽視此參數。

## 匯入 (IMPORT/IMPORTL)

## ■ 檔案內資料的字組偏移量 (PARAM03)

設定偏移的字數。

<IMPORT>

設定範圍為 0 ~ 32766。

<IMPORTL>

依據 MP3000 系列運動控制器的軟體版本，設定範圍有所不同。  
請參照下列設定。

版本	設定範圍
Ver1.21 以下	0 ~ 32766
Ver1.22 以上	0 ~ 2147483646

## ■ 系統預約 (PARAM04)

用於系統的工作區域。

## ■ 系統預約 (PARAM05)

用於系統的工作區域。

## 程式範例

將 MPE720 的暫存器清單資料匯入 MW01234 ~ MW01243 時的程式範例，如下所示。關於 MPE720 的操作步驟，請參照下列項目。

☞ 補充事項 (第 4-284 頁)

The screenshot displays a software interface with two main components:

- EXPRESSION Window:** Contains the following code:
 

```
DW00011=2; // csv;
DW00012=2; // 行オフセット
```
- IMPORT Dialog Box:** A floating window with the title 'IMPORT'. It contains the following parameters:
 

[B] Execute	[B] Busy
DB000000	DB000002
[B] Abort	[B] Complete
DB000001	DB000003
[W] Drv-No	[B] Error
00001	DB000004
[W] Data-No	
00001	
[W] Size	
00010	
[W] Ch-No	
00001	
[A] Dest	
MA01234	
[A] Param	
DA00010	

執行指令前

暫存器	值
MW01233	0
MW01234	0
MW01235	0
MW01236	0
MW01237	0
MW01238	0
MW01239	0
MW01240	0
MW01241	0
MW01242	0
MW01243	0
MW01244	0



執行指令後

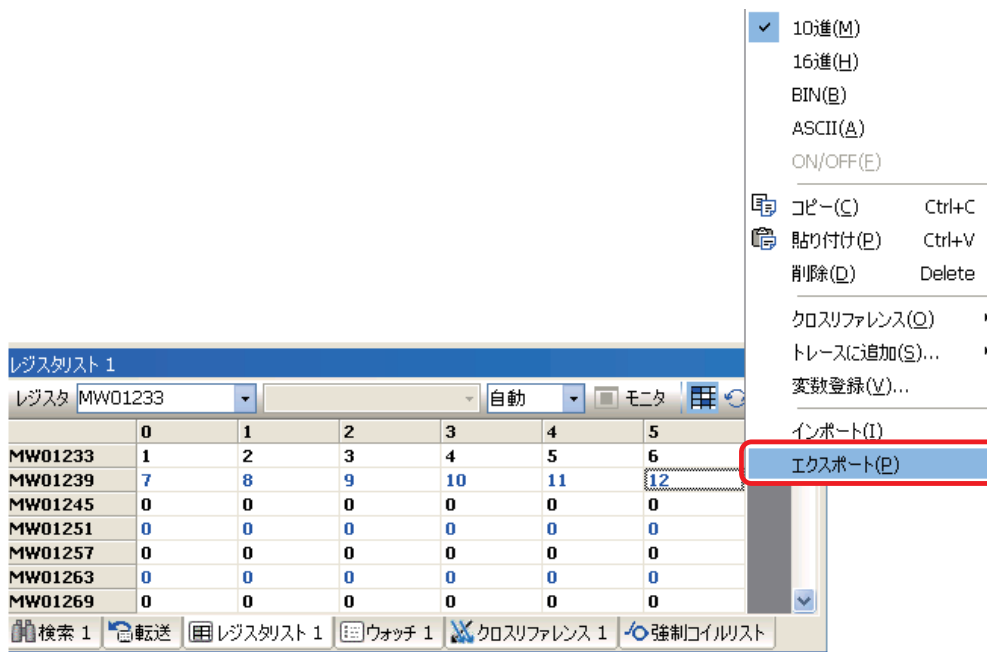
暫存器	值
MW01233	0
MW01234	2
MW01235	3
MW01236	4
MW01237	5
MW01238	6
MW01239	7
MW01240	8
MW01241	9
MW01242	10
MW01243	11
MW01244	0

## 補充事項

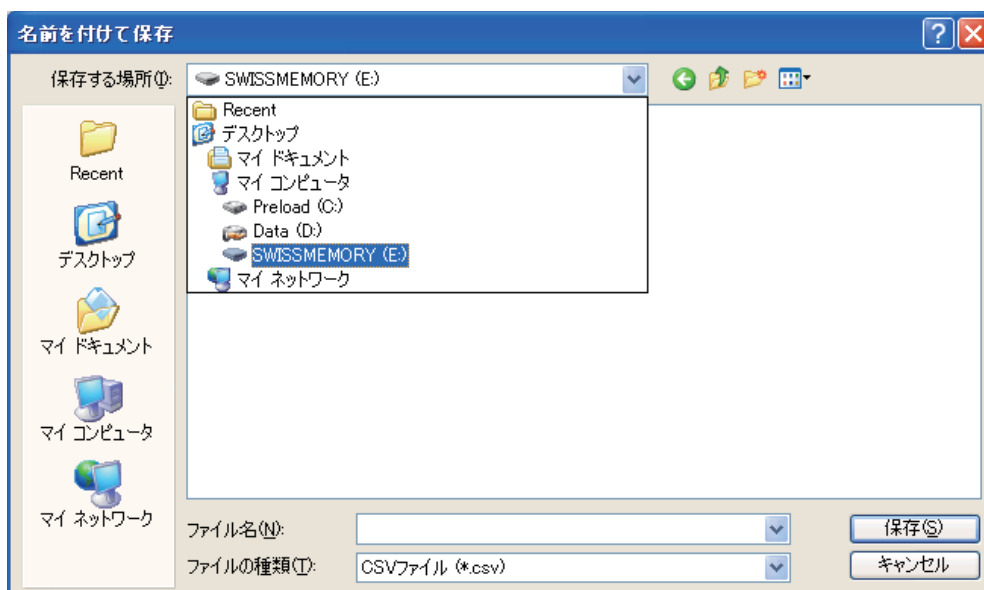
請依下列步驟匯出 MPE720 的暫存器清單資料。

另外，此處將敘述依據前述程式範例內容的步驟。

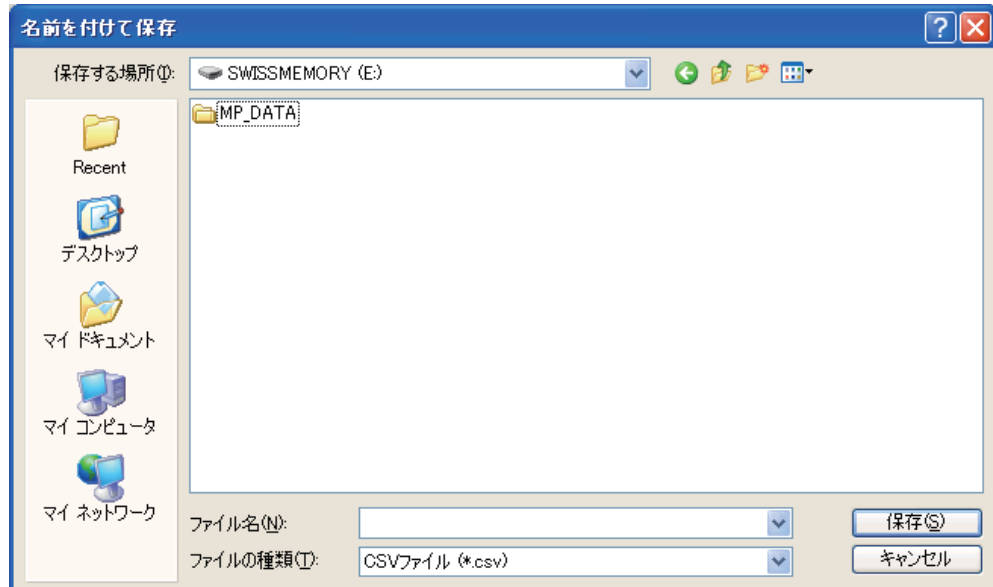
1. 將 USB 隨身碟插入 PC。
2. 在 MPE720 上顯示暫存器清單。
3. 右鍵點擊暫存器清單，在跳顯的選單中選擇 [ 匯出 ]。



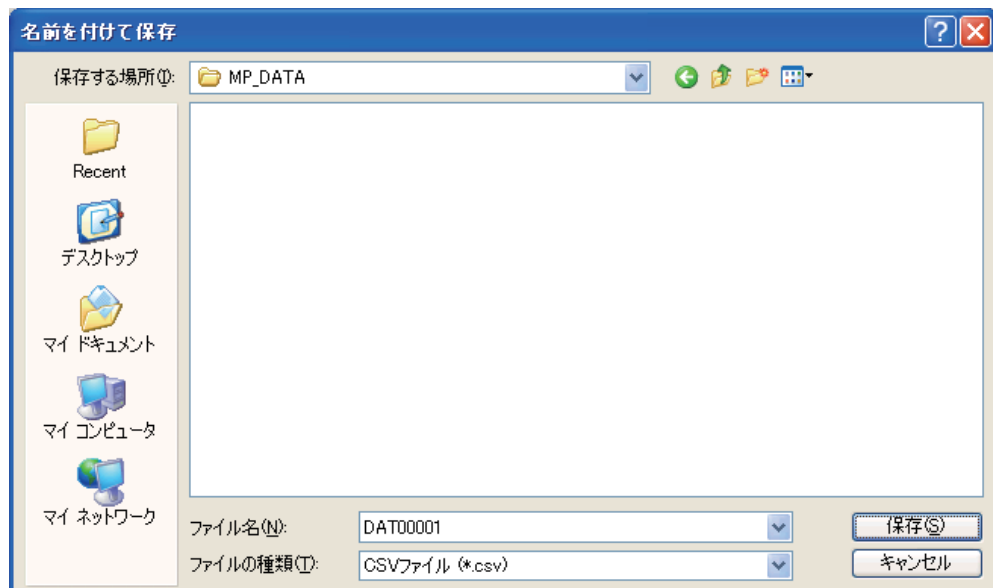
4. 選擇 USB 隨身碟磁碟路徑。



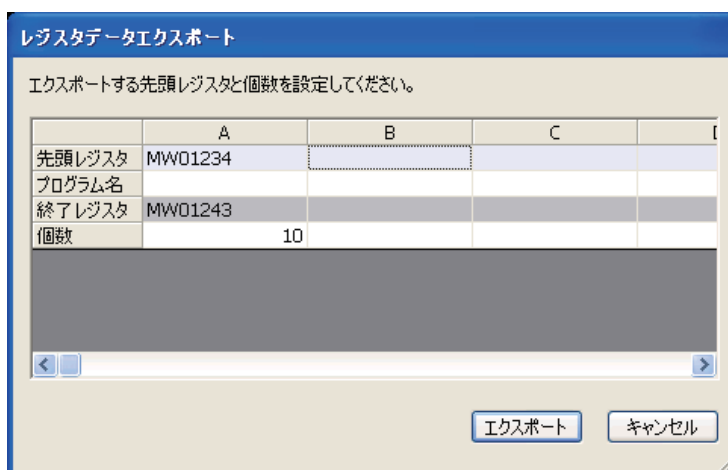
5. 點擊 [MP\_DATA] 資料夾。  
[MP\_DATA] 資料夾如不存在，請新建資料夾。



6. 在 [ 檔案名稱 ] 輸入框輸入「DAT00001」，點擊 [ 儲存 ] 按鈕。



7. 在 [開頭暫存器] 輸入「MW01234」、[個數] 輸入「10」後，點擊 [輸出] 按鈕。



8. 由 PC 取出 USB 隨身碟。
9. 將 USB 隨身碟插入運動控制器。
10. 等待 USB ACCESS LED 亮燈。

USB  
ACTIVE 

11. 建立前述程式範例中的程式。
12. 執行 IMPORT/IMPORTL 指令。

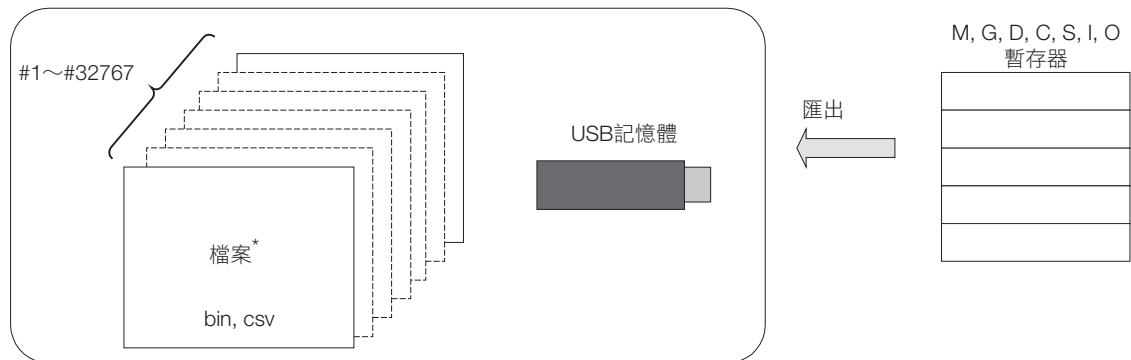
## 匯出 (EXPORT/EXPORTL)

將暫存器資料匯出至 USB 隨身碟。

匯出資料的格式，可選擇 2 進制資料 (bin)、CSV 資料 (csv)。

匯出的對象暫存器，可選擇 M 暫存器、G 暫存器、D 暫存器、C 暫存器、S 暫存器、I 暫存器或 O 暫存器。

EXPORT/EXPORTL 指令可同時平行執行 2 個指令。





\* 以下列檔名匯出至 USB 記憶體。  
 ¥MP\_DATA¥DAT00001.BIN (CSV)  
 :  
 ¥MP\_DATA¥DAT32767.BIN (CSV)

### 格式

格式如下所示。

EXPORT	
[B] Execute	[B] Busy
MB000000	MB000002
[B] Abort	[B] Complete
MB000001	MB000003
[W] Drv-No	[B] Error
MW000001	MB000004
[W] Data-No	
MW000002	
[W] Size	
MW000003	
[W] Ch-No	
MW000004	
[A] Src	
MA00100	
[A] Str	
MA00200	
[A] Param	
MA00010	

圖示： 匯出  匯出(擴張)

鍵輸入：EXPORT/EXPORTL

4.10 系統函數指令

匯出 (EXPORT/EXPORTL)

輸出輸入項目	可以使用的資料類型								
	B	W	L	Q	F	D	A	指標	常數
匯出指令 (Execute)	○	×	×	×	×	×	×	×	×
匯出強制中止指令 (Abort)	○	×	×	×	×	×	×	×	×
驅動器編號 (Drv-No)	×	○	×	×	×	×	×	○	○
資料編號 (Data-No)	×	○	×	×	×	×	×	○	○
傳輸字組數 (Size)	×	○*1	○*2	×	×	×	×	○	○
平行執行頻道編號 (Ch-No)	×	○	×	×	×	×	×	○	○
傳輸來源起始暫存器編號 (Src)	×	×	×	×	×	×	○	×	×
文字列輸出用暫存器編號 (Str)	×	×	×	×	×	×	○	×	×
參數表起始位址 (Param)	×	×	×	×	×	×	○	×	×
執行匯出中 (Busy)	○	×	×	×	×	×	×	×	×
匯出執行完成 (Complete)	○	×	×	×	×	×	×	×	×
錯誤 (Error)	○	×	×	×	×	×	×	×	×

\*1. EXPORT 指令時

\*2. EXPORTL 指令時

以下說明各輸入輸出項目的內容。

輸出輸入項目	內容	輸出輸入
匯出指令 (Execute)	ON 則匯出指令開始。 執行指令中需保持為 ON。	IN
匯出強制中止指令 (Abort)	ON 則強制中斷匯出處理。	IN
驅動器編號 (Drv-No)	驅動器編號 (1 : USB 隨身碟)	IN
資料編號 (Data-No)	資料編號 (1 ~ 32767)	IN
傳輸字組數 (Size)	傳輸字元數 (EXPORT:1 ~ 32767, EXPORTL:1 ~ 2147483647)	IN
平行執行頻道編號 (Ch-No)	平行執行頻道編號 (1, 2)	IN
傳輸來源起始暫存器編號 (Src)	傳輸來源起始暫存器編號 (MA, GA, DA, CA, SA, IA, OA)	IN
文字列輸出用暫存器編號 (Str)	字串輸出用暫存器編號*1*2 (MA, GA, DA, CA)	IN
參數表起始位址 (Param)	參數表起始位址 (MA, GA, DA)	IN
執行匯出中 (Busy)	匯出執行時為 ON。	OUT
匯出執行完成 (Complete)	匯出完畢後為 ON。	OUT
錯誤 (Error)	發生錯誤而 ON。	OUT

\*1. CSV 檔案時啟用。如為 2 進制檔案，將忽視此項目。

\*2. 請務必設定表示字串結尾的 0 (NULL 字元)。



## ◆ 參數詳情

說明參數詳情。

<EXPORT>

位址	資料類型	參數 No.	IN/OUT	內容
0	W	PARAM00	OUT	處理結果
1	W	PARAM01	IN	格式種類
2	W	PARAM02	IN	檔案開啟類型
3	W	PARAM03	IN	檔案內資料的字組偏移量
4	W	PARAM04	OUT	系統預約
5	W	PARAM05	OUT	系統預約

<EXPORTL>

位址	資料類型	參數 No.	IN/OUT	內容
0	W	PARAM00	OUT	處理結果
1	W	PARAM01	IN	格式種類
2	L	PARAM02	IN	檔案開啟類型
4	L	PARAM03	IN	檔案內資料的字組偏移量
6	W	PARAM04	OUT	系統預約
7	W	PARAM05	OUT	系統預約

## ■ 處理結果 (PARAM00)

將報告 EXPORT/EXPORTL 指令的處理結果。

- 00□□H：處理中 (Busy)
- 10□□H：處理完成 (Complete)
- 8□□□H：發生錯誤 (Error)

錯誤分類如下所示。

錯誤代碼	內容
8101H	超出範圍錯誤 (驅動器編號)
8102H	超出範圍錯誤 (資料編號)
8103H	超出範圍錯誤 (傳輸數)
8104H	超出範圍錯誤 (平行執行頻道編號)
8105H	超出範圍錯誤 (傳輸目的地/傳輸來源暫存器編號)
8106H	超出範圍錯誤 (格式種類)
8107H	超出範圍錯誤 (開啟類型)
8108H	超出範圍錯誤 (檔案內資料的字組偏移量)
8109H	超出範圍錯誤 (參數表起始位址)
810BH	字串以上 (未偵測出 NULL 字元)
8201H	未安插 USB 隨身碟
8202H	檔案開啟錯誤
8203H	檔案搜尋錯誤
8204H	檔案寫入錯誤
8205H	檔案讀取錯誤
8206H	檔案關閉錯誤
8301H	由於超過可處理檔案數，因此無法處理
8302H	檔案輸入輸出逾時

## 匯出 (EXPORT/EXPORTL)

## ■ 格式種類 (PARAM01)

設定匯出檔案的格式。

匯出 MPE720 的暫存器清單資料時，請設為 2。

1：由 2 進制檔案 (DAT□□□□□.BIN) 匯出。

□□□□□ 為資料編號 (Data-No) 所設定的數值。

2：由 CSV 檔案 (DAT□□□□□.CSV) 匯出。

□□□□□ 為資料編號 (Data-No) 所設定的數值。

## ■ 檔案開啟類別 (PARAM02)

如為 2 進制檔案，設定檔案開啟類別。

1：新建檔案並匯出。

2：將既有檔案匯出。

選擇此類時，僅有部分資料可進行變更。

CSV 檔案將無視這項參數。

## ■ 檔案內資料的字組偏移量 (PARAM03)

如為 2 進制檔案，設定偏移字數。

CSV 檔案將無視這項參數。

<EXPORT>

設定範圍為 0 ~ 32766。

<EXPORTL>

依據 MP3000 系列運動控制器的軟體版本，設定範圍有所不同。  
請參照下列設定。

版本	設定範圍
Ver1.21 以下	0 ~ 32766
Ver1.22 以上	0 ~ 2147483646

## ■ 系統預約 (PARAM04)

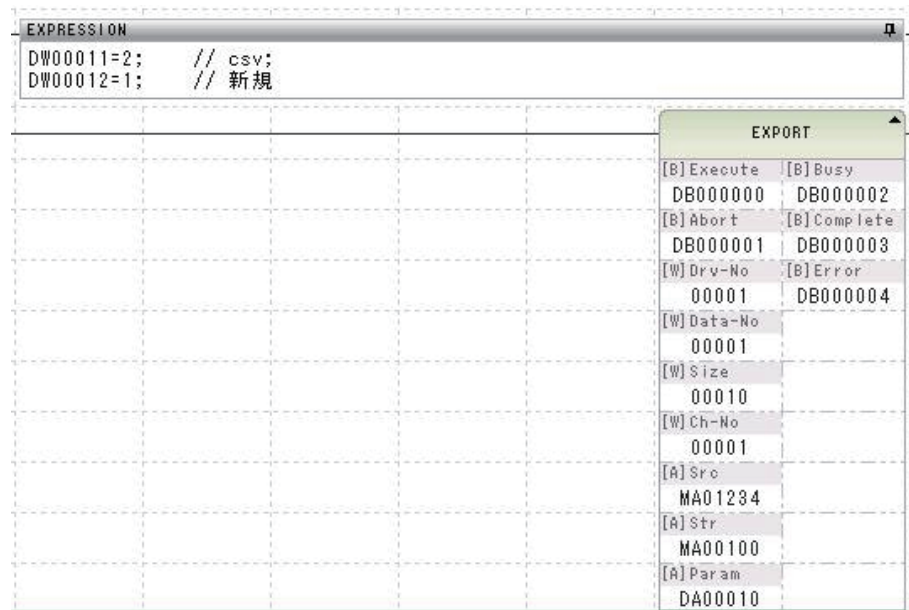
用於系統的工作區域。

## ■ 系統預約 (PARAM05)

用於系統的工作區域。

## 程式範例

將 MW01234 ~ MW01243 的資料匯出為 CSV 檔案的程式範例，如下所示。



暫存器資料

暫存器	值
MW01233	1
MW01234	2
MW01235	3
MW01236	4
MW01237	5
MW01238	6
MW01239	7
MW01240	8
MW01241	9
MW01242	10
MW01243	11
MW01244	12

USB 隨身碟的檔案 (DAT00001.CSV)

純文字檔案的內容
MW1234 ↵
↵
00002 ↵
00003 ↵
00004 ↵
00005 ↵
00006 ↵
00007 ↵
00008 ↵
00009 ↵
00010 ↵
00011 ↵





# 工程工具 (MPE720)

## 功能介紹

# 5

本章節將就階梯圖程式工程工具 MPE720 的部分功能進行介紹。

- 5.1 階梯圖程式的執行中監視 ..... 5-4**
- 5.2 搜尋／取代 ..... 5-5**
  - 程式內的搜尋／取代 ..... 5-5
  - 專案檔案中的搜尋／取代 ..... 5-8
- 5.3 相互參照 ..... 5-10**
- 5.4 檢查雙重線圈 ..... 5-13**
- 5.5 線圈強制 ON / OFF ..... 5-14**
  - 從階梯圖程式設定強制 ON / OFF ..... 5-14
  - 從強制線圈清單子視窗變更為強制 ON / OFF ..... 5-14
- 5.6 引用呼叫程式 ..... 5-17**
- 5.7 暫存器清單 ..... 5-18**
  - 顯示暫存器對照表 ..... 5-18
  - 切換暫存器對照表的顯示內容 ..... 5-19
  - 編輯資料 ..... 5-20
- 5.8 調整面板 ..... 5-21**
- 5.9 階梯圖程式的有效／無效 ..... 5-22**

<b>5.10</b>	<b>監看</b> .....	<b>5-23</b>
	顯示監看資料 .....	5-23
	編輯 [ 數值 ] 欄位 .....	5-24
<b>5.11</b>	<b>安全功能</b> .....	<b>5-25</b>
<b>5.12</b>	<b>追蹤功能</b> .....	<b>5-26</b>
<b>5.13</b>	<b>進階使用方式</b> .....	<b>5-27</b>
	運動程式 .....	5-27

---

本章節將介紹支援階梯圖編程以及除錯的工程工具 MPE720 Ver.7 的下列功能。

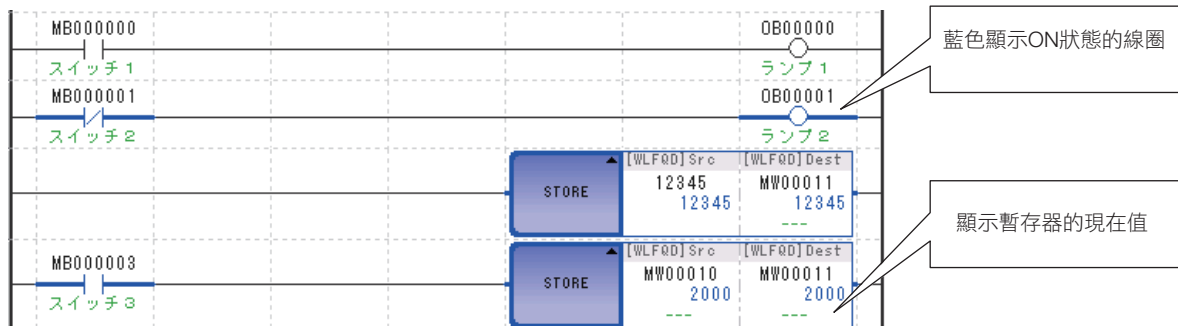
- 階梯圖程式的執行中監視
- 搜尋／取代
- 相互參照
- 雙重線圈
- 線圈強制 ON / OFF
- 引用呼叫程式
- 暫存器清單
- 調整面板
- 階梯圖程式的有效／無效
- 監看
- 安全功能
- 追蹤功能
- 使用運動程式

## 5.1 階梯圖程式的執行中監視

可監視各指令執行狀態的功能。可以在連接運動控制器的狀態下使用。

關於「繼電器輸出」已轉為 ON 的指令，將以藍色顯示。

此外，也會顯示暫存器現在值（位於執行中指令的參數內）。





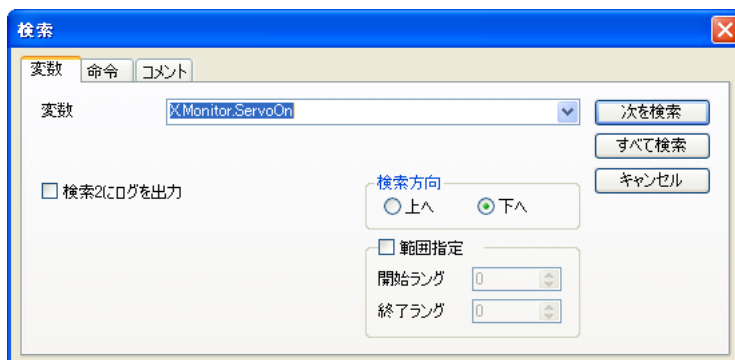
## 5.2 搜尋／取代

### 程式內的搜尋／取代

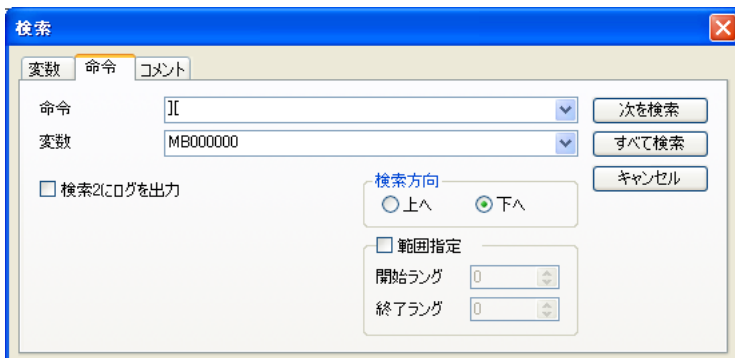
可指定程式並搜尋程式內使用之變數、指令、註釋的功能。此外亦可搜尋、取代暫存器及註釋。  
以下說明搜尋及取代的操作方法。

#### 程式內的搜尋

1. 將要搜尋的程式顯示在階梯圖編輯畫面的最前面，從主選單選擇 [編輯] - [搜尋]。  
顯示 [搜尋] 對話方塊。
2. 選擇 [變數]、[指令]、[註釋] 的其中任一項標籤，設定搜尋內容。



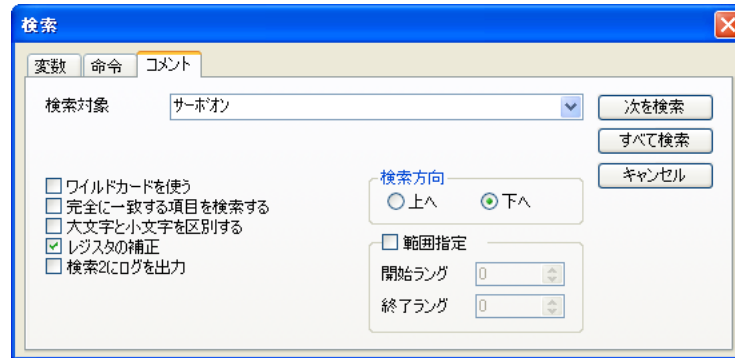
[變數] 標籤：以變數及暫存器為搜尋對象。變數也可從變數子視窗進行複製 & 貼上而輸入。



[指令] 標籤：請在 [指令] 欄內輸入指令名稱或指令鍵分配。

在 [指令] 欄內輸入指令後，便會顯示 [變數] 欄。另外，若在 [指令] 欄內輸入 SEE 指令，  
[變數] 欄會變更成 [程式名稱]。

變數也可從變數子視窗進行複製 & 貼上而輸入。



[ 註釋 ] Tab：以目標的註釋、梯級註釋、程式註釋、式中的註釋為對象。

- 使用萬用字元（註釋）：搜尋詞可使用萬用字元「\*」及「?」。
- 搜尋完全一致的項目（註釋）：搜尋註釋欄內的字串和搜尋字串完全一致的項目。但大小寫區別則視 [ 區分大小寫 ] 的設定。
- 區分大小寫（註釋）：設定是否區分半形英文字母的大小寫。
- 暫存器補償（註釋）：設定當作為搜尋目標輸入的字串被辨識為暫存器時，是否將字串修正為正確的暫存器格式。
- 將履歷輸出至搜尋 2：勾選後搜尋結果便會被輸出至搜尋 2 子視窗，不會變更搜尋 1 子視窗的輸出內容。全取消時，輸出至搜尋 1 子視窗。
- 指定範圍：打勾時可指定開始梯級、結束梯級，可設定搜尋範圍。

### 3. 選擇 [ 搜尋次項 ] 或 [ 搜尋全部 ] 按鈕後，開始搜尋。

點選 [ 搜尋次項 ] 按鈕後，便會讓符合的指令對象進入被選擇狀態。

點選 [ 搜尋全部 ] 按鈕後，將在搜尋 1 或搜尋 2 子視窗顯示搜尋結果。

#### 検索 1

```

検索開始 'XMonitor.ServoOn'
H02.01 : [Rung 0000, Step 0004, NOC, Operand 00] : XMonitor.ServoOn
H02.01 : [Rung 0006, Step 0023, NOC, Operand 00] : XMonitor.ServoOn
検索終了 2個見つかりました。

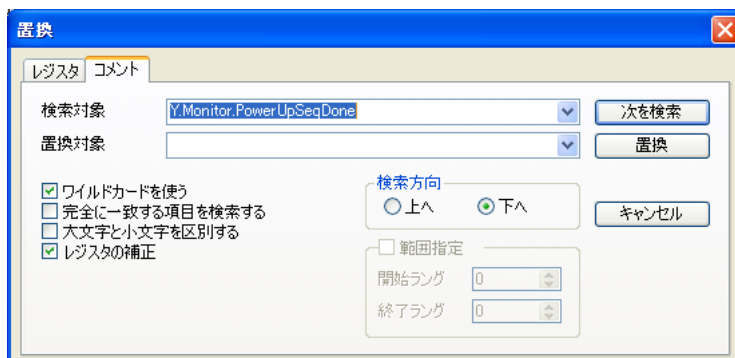
```

## 程式內的取代

1. 將要搜尋、取代的程式顯示在階梯圖編輯畫面的最前面，從主選單選擇 [編輯] – [取代]。顯示 [取代] 對話方塊。
2. 選擇 [暫存器]、[註釋] 的其中任一項標籤，設定搜尋內容及取代內容。



[暫存器] 標籤：以暫存器為搜尋對象。



[註釋] Tab：以目標的註釋、梯級註釋、程式註釋、式中的註釋為對象。

- 使用萬用字元（註釋）：搜尋詞可使用萬用字元「\*」及「?」。  
 （註）若在 [取代暫存器] 及 [取代對象] 內使用「\*」及「?」，便會將其代換為字串，而非作為萬用字元使用。
  - 指定範圍：打勾時可指定開始梯級、結束梯級，可設定搜尋範圍。  
 但在使用 [註釋] 標籤時無法使用。
3. 開始搜尋、取代。  
 點選 [搜尋次項] 按鈕後，便會讓符合的指令對象進入被選擇狀態，點選 [取代] 按鈕後便會開始取代 [取代暫存器] 或 [取代對象] 的語句。  
 若在 [暫存器] 標籤中點選 [取代全部] 按鈕，便會取代符合的暫存器，並在輸出子視窗顯示取代結果。

```
出力
置換開始 'MB300000'を'MB300000'に置き換えます。
成功: H01 [Rune 0002, Step 0004, NOC, Operand 00]: [置換前]MB300000 -> [置換後]MB300000
成功: H01 [Rune 0003, Step 0007, NOC, Operand 00]: [置換前]MB300000 -> [置換後]MB300000
置換終了。2個見つかりました。
成功2、失敗0
```

## 專案檔案中的搜尋／取代

能以專案檔內所有階梯圖程式與運動程式（或指定的程式）為對象搜尋變數。此外亦可搜尋／取代暫存器及位址。



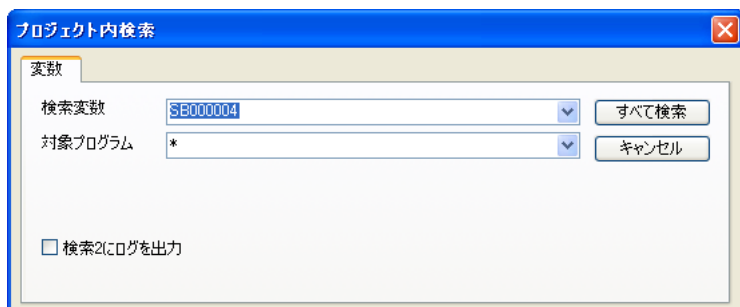
本功能僅可在未與運動控制器連線時（離線時）有效。

註記

下列為搜尋及取代專案檔案內部內容的操作方法。

### 專案檔案內的搜尋

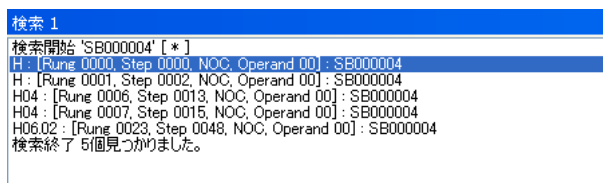
1. 將要搜尋的程式顯示在階梯圖編輯畫面的最前面，從主選單選擇 [編輯] – [搜尋專案檔內]。將顯示 [搜尋專案檔內] 對話方塊。
2. 設定搜尋的變數（位址）及搜尋對象的程式名稱。



- (註) 1. 變數也可從變數子視窗進行複製？貼上而輸入。
2. 若在 [目標程式] 內輸入「，」（逗號）及空格，便可指定複數程式。另外 [目標程式] 亦可使用下列「\*」（萬用字元）。
    - \*、H\*、L\*、I\*、A\*、F\*（函數全）、MPM\*、MPS\*
 僅可在上述的形態下使用通配。其他（「H01.\*」等）無法使用。
  3. 若勾選 [將履歷輸出至搜尋 2]，便會將搜尋結果輸出至搜尋 2 子視窗，不會變更搜尋 1 子視窗的輸出內容。全取消時，輸出至搜尋 1 子視窗。

### 3.開始進行搜尋。

點選 [搜尋全部] 按鈕後，會顯示代表搜尋中的進度條，並在搜尋子視窗中顯示搜尋結果。



## 專案檔案內的取代



註記

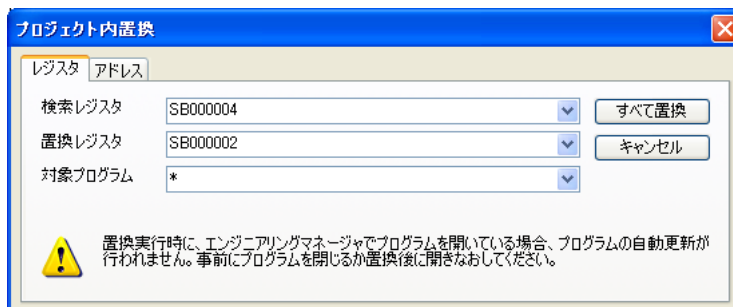
- 關於專案檔案內的取代動作，若在取代後執行編譯及儲存，便無法復原回取代前的狀態。若是重要檔案，請在取代前製作備份檔。
- 執行取代之前，如果以 MPE720 的 Engineering Manager 開啟了運動程式，程式將不會自動更新。請先關閉運動程式，再進行取代。

- 將要搜尋的程式顯示在階梯圖編輯畫面的最前面，從主選單選擇 [編輯] – [取代專案檔內]。將顯示 [取代專案檔內] 對話方塊。
- 設定搜尋的變數（位址）及搜尋對象的程式名稱。
 

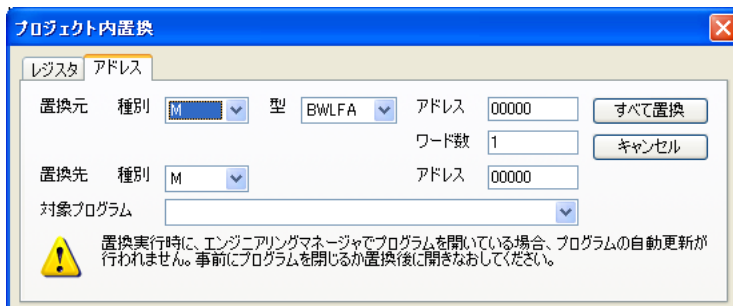
(註) 1. 變數也可從變數子視窗進行複製 & 貼上而輸入。

2. 若在 [目標程式] 內輸入「，」（逗號）及空格，便可指定複數程式。另外「目標程式」亦可使用下列「\*」（萬用字元）。

\*、H\*、L\*、I\*、A\*、F\*（函數全）、MPM\*、MPS\*  
僅可在上述的形態下使用通配。其他（「H01.\*」等）無法使用。
- 選擇 [暫存器]、[位址] 的其中任一項標籤，設定搜尋內容及取代內容。



[暫存器] 標籤：以單個暫存器為取代對象。

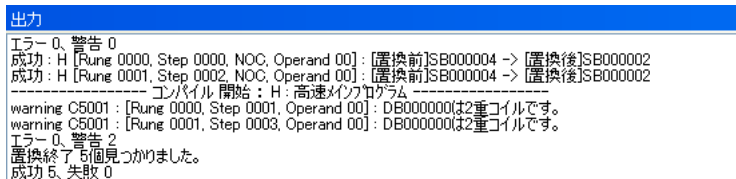


[位址] 標籤：以符合條件的暫存器為取代對象。

(註) 另外「目標程式」中亦可使用下列「\*」（萬用字元）。

\*、H\*、L\*、I\*、A\*、F\*、MPM\*、MPS\*

- 開始搜尋、取代。  
點選 [取代全部] 按鈕後，將在輸出子視窗顯示取代結果。



(註) 若是在編譯中發生錯誤的程式，不會執行取代動作。  
執行取代後，會顯示被取代的暫存器變數或位址。

## 5.3 相互参照

相互参照是種可確認某暫存器是否被程式使用中，若被使用時是在何處使用的搜尋功能。  
在搜尋結果中，會以紅色標示作為輸出暫存器使用的位置，以藍色標示作為輸入暫存器使用的位置。

相互参照実行

搜尋結果顯示  
紅色：輸出暫存器  
藍色：輸入暫存器

レジスタ	プログラム	実行命令	実行ステップ	書込/読込	コメント
<b>同一レジスタ</b>					
MW00000	H02: 1軸目手動運転	LOAD: 整数型置数	46	読込	
MW00000	H02: 1軸目手動運転	STORE: 格納	49	書込	
<b>重複アドレス</b>					
MB000004	H02: 1軸目手動運転	NOC: A接点	50	読込	
MB000005	H02: 1軸目手動運転	COIL: コイル	51	書込	
ML00000	H03: 2軸目手動運転	LOAD: 整数型置数	4	読込	
ML00000	H03: 2軸目手動運転	STORE: 格納	7	書込	

當暫存器值和設定值不同時，表示有可能已在程式內的其他地方覆寫數值。此時可透過相互参照功能搜尋該暫存器，確認以紅色標示的位置後，鎖定出正在執行覆寫的程式。

例 使用陣列時，會以下列方式搜尋。

1. 使用暫存器 [ 暫存器 ] 時

→ MW00000及MW00001為搜尋對象。

2. 使用暫存器 [ 參數 ] 時

→ MW00000及MW00005為搜尋對象。

3. 使用暫存器 [ 參數 ] 時 ( LONG 型 )



→ ML00000及ML00010為搜尋對象。

交互條件具有下列設定項目。以下將說明各功能。



ローカルレジスタは開いているプログラムを検索する

勾選	搜尋方法
有效	針對在 MPE720 畫面上轉為主動 (Active) 的畫面，搜尋其局部暫存器 (D 暫存器)。
無效	針對指定畫面，搜尋其局部暫存器 (D 暫存器)。

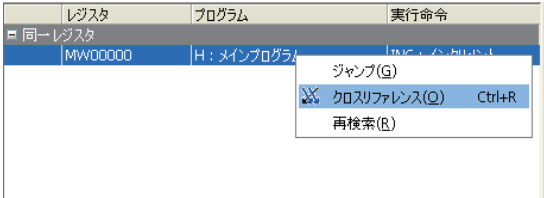
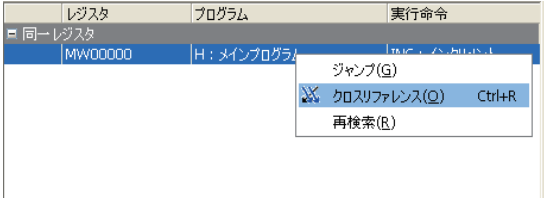
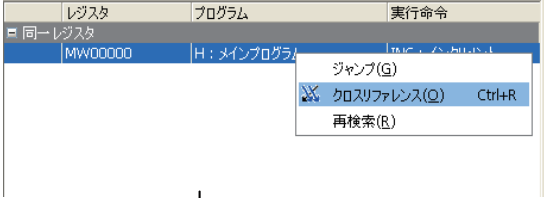
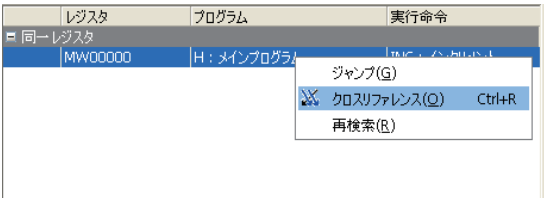
同じレジスタを検索する(同一レジスタ)

勾選	搜尋方法
有效	<p>搜尋和已搜尋暫存器相同的暫存器。 在以下程式中以「MW00000」搜尋變數時，若要顯示在清單中，請勾選核取方塊。</p> 
無效	<p>不會搜尋資料類型和已搜尋暫存器相同的暫存器。 在以下程式中以「MW00000」搜尋變數時，若不想要顯示在清單中，請勾除核取方塊。</p> 

 アドレスが重複するレジスタを検索する(重複アドレス)

勾選	搜尋方法
有效	<p>搜尋重複位址。 在以下程式中以「ML00000」等其他資料類型搜尋變數時，若要顯示在清單中，請勾選核取方塊。</p> 
無效	<p>不搜尋重複位址。 在以下程式中以「ML00000」等其他資料類型搜尋變數時，若不想要顯示在清單中，請勾除核取方塊。</p> 

次のクロスリファレンス2に検索結果を表示する

勾選	搜尋方法
<p>有効</p>	<p>從相互参照視窗中執行相互参照時，將在其他視窗顯示搜尋結果。 最多可在 3 個視窗中保留相互参照結果。</p> <p>相互参照1</p>  <p>↓ 視窗切換</p> <p>相互参照2</p>  <p>↓ 視窗切換</p> <p>相互参照3</p>  <p>視窗切換</p>
<p>無效</p>	<p>從相互参照視窗中執行相互参照時，將在同一個視窗中更新並顯示搜尋結果。</p> <p>相互参照1</p>  <p>視窗更新</p>



## 5.4

## 檢查雙重線圈

本功能以所有階梯圖程式為對象，搜尋程式內的雙重線圈（使用同一個暫存器的線圈），並顯示搜尋結果。



註記

由於使用「專案連接」連接時會使用專案檔案的資料，因此有可能會顯示和專案連接中的運動控制器資料不一致的結果。

要透過專案連接連線搜尋雙重線圈時，請預先以 [ 從控制器讀取 ] 將資料傳輸至專案檔內。

從主選單選取 [ 除錯 ] - [ 檢查雙重線圈 ]。

將搜尋雙重線圈，並將結果輸出至雙重線圈檢查子視窗中。

出力タイプ	レジスタ	プログラム	実行ステップ
-(-)	OB00000	H: メインプログラム	1
-(-)	OB00000	H: メインプログラム	3

A callout box points to the table with the text: 顯示2重線圈的搜尋結果



註記

在編譯設定中有效開啟 [ 檢查雙重線圈 ] 選項，在編譯過程中也會搜尋雙重線圈，並將結果作為警告顯示在輸出子視窗上。

## 5.5 線圈強制 ON / OFF

本功能可對階梯圖編輯器上的指定線圈進行強制 ON / OFF。

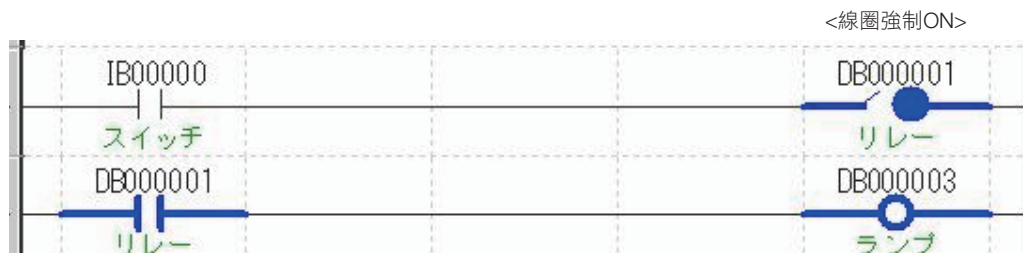
使用本功能時，無論線圈左側的輸出如何，都將進行線圈輸出。

舉例來說，製作下列階梯圖時，即使實際上並沒有開關（IB00000），但讓繼電器（DB000001）強制 ON 後，就能模擬按下開關後的動作。

### 從階梯圖程式設定強制 ON / OFF

可在階梯圖編輯畫面，將指定線圈物件強制為 ON 或 OFF，對程式進行監視。

1. 選擇要強制 ON 或 OFF 的線圈（進入選擇狀態）。
2. 從主選單選取 [ 除錯 ] - [ 強制 ON ] 或 [ 強制 OFF ]。  
被選擇的線圈將強制連接或強制斷線。



補充


若要取消 [ 強制 ON ]（或 [ 強制 OFF ]），則從主選單選取 [ 除錯 ] - [ 強制取消 ]。

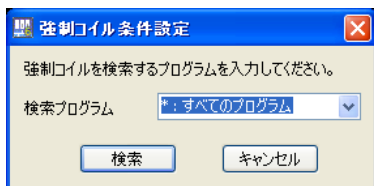
### 從強制線圈清單子視窗變更為強制 ON / OFF

強制線圈清單子視窗中，可以清單方式顯示階梯圖程式中的強制線圈 ON / OFF 狀態。

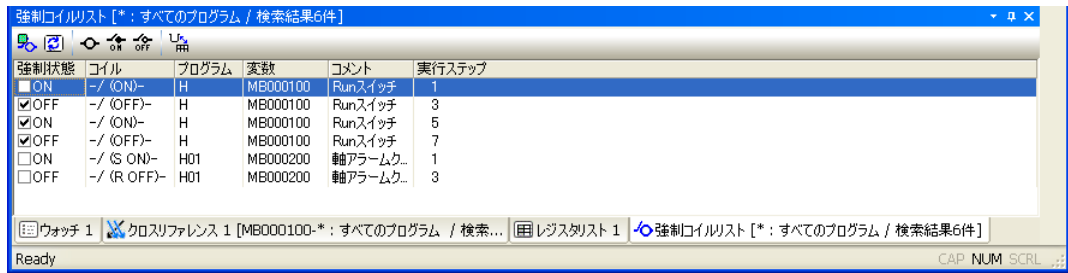
此外也能以所有程式為對象，變更及解除強制線圈的 ON / OFF / 取消狀態。

#### 搜尋強制線圈清單子視窗內的強制線圈

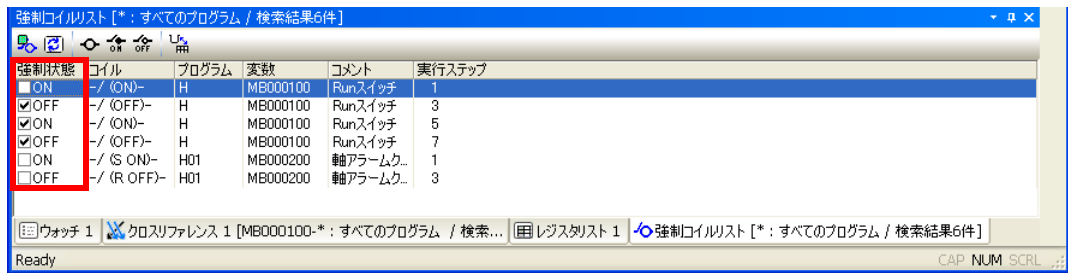
1. 將顯示強制線圈清單子視窗。  
(註) 在主選單選擇 [ 顯示 ] - [ 其他視窗 ] - [ 強制線圈清單 ]，可以切換顯示 / 隱藏強制線圈清單子視窗。
2. 從主選單選取 [ 除錯 ] - [ 強制線圈清單 ]。  
(註) 在上述狀況時，會以所有程式為對象搜尋強制線圈。要指定作為搜尋對象的程式時，請點選強制線圈條件設定按鈕 (  )，顯示強制線圈條件設定視窗。



搜尋結果將顯示在強制線圈清單子視窗內。



### 3. 勾选想要強制 ON / OFF 的線圈。

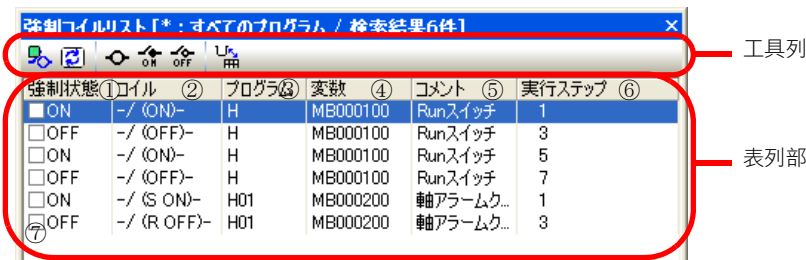


註記


- 若在強制線圈清單子視窗的清單區點擊右鍵，從跳出的選單中選擇 [ 全選 ]、[ 全部取消 ]，便能一次勾选或一次取消線圈強制狀態的勾选欄。
- 在強制線圈清單子視窗的搜尋結果畫面中選擇任意行或雙擊後，便會切換到該階梯圖程式內的線圈。此外，若在強制線圈清單子視窗的清單區點擊右鍵，從跳出的選單中選擇 [ 切換 ]，也能進行相同操作。若程式未開啟，則會自動開啟程式並切換至該程式內的線圈。
- 若在強制線圈清單子視窗的清單區點擊右鍵，從跳出的選單中選擇 [ 相互参照 ]，或從主選單選擇 [ 除錯 ]-[ 相互参照 ]，便會以相互参照方式搜尋線圈所設定的暫存器，並將搜尋結果顯示在相互参照子視窗上。
- 若在顯示搜尋結果途中編輯階梯圖程式，便會以灰色標示出被編輯程式的線圈。

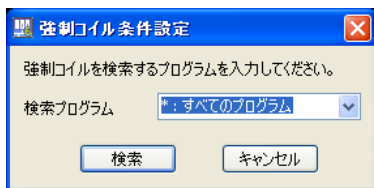
## 強制線圈清單子視窗的各部位名稱及概要






強制線圈清單子視窗由顯示強制線圈的「清單區」、強制線圈搜尋、重新搜尋與用以變更強制狀態的「工具列」所組成。



## ◆ 工具列

- 強制線圈條件設定按鈕 (  )  
將顯示強制線圈條件設定對話方塊。可指定要搜尋強制線圈的程式，開始執行強制線圈搜尋動作。



- 重新搜尋按鈕 (  )  
可重新搜尋在 [ 強制線圈條件設定 ] 對話方塊中所指定之程式的強制線圈。
- 強制取消按鈕 (  )  
取消在勾選欄位中打勾之線圈的強制狀態。
- 強制 ON 按鈕 (  )  
讓在勾選欄位中打勾之線圈強制進入 ON。
- 強制 OFF 按鈕 (  )  
讓在勾選欄位中打勾之線圈強制進入 OFF。
- 變數顯示切換按鈕 (  )  
將線圈使用中之暫存器的顯示方式切換為暫存器及變數。

## ◆ 表列部

## ① 強制狀態

以「ON」或「OFF」表示搜尋結果線圈的強制狀態。

## ② 線圈

顯示搜尋結果線圈。

線圈全部可分為以下 6 種。

線圈種類	線圈符號	
	ON	OFF
線圈	-/ (ON)-	-/ (OFF)-
設定線圈	-/ (S ON)-	-/ (S OFF)-
重置線圈	-/ (R ON)-	-/ (R OFF)-

## ③ 程式

顯示搜尋線圈的程式名稱。

## ④ 變數

顯示搜尋結果線圈所設定的變數或暫存器。

## ⑤ 註釋

顯示變數的註釋。

## ⑥ 執行步驟

顯示搜尋結果線圈的執行步驟編號。

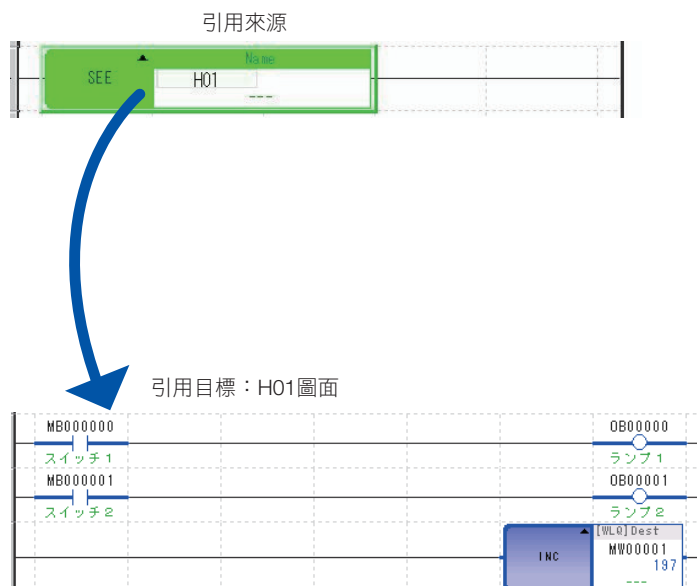
## ⑦ 核取方塊

勾選後便會成為強制操作 ( ON / OFF / 取消 ) 的對象。可使用工具列按鈕或彈出式選單，將線圈一次全部設為強制 ON / OFF / 取消狀態。

## 5.6 引用呼叫程式

可開啟 SEE 指令及 FUNC 指令所引用的圖面。

選擇想要確認程式的 SEE 指令目標或 FUNC 指令目標後，從主選單選取 [ 除錯 ] - [ 從現在位置引用呼叫 (R) ]。



## 5.7

## 暫存器清單

在「暫存器清單」功能中，可透過暫存器清單子視窗（1、2、3）監視連續範圍的暫存器現在值（暫存器對照表）。運動控制器連接時可即時監視。還可進行值的編輯。



註記

- 暫存器對照表在直接連接情況下也將顯示專案檔的資料。  
由於使用「專案連接」連接時會使用運動控制器的資料，因此在顯示暫存器對照表時，有可能會出現和專案連接中的專案檔案不一致的結果。  
要以專案連接連線顯示暫存器對照表時，請預先以 [ 從控制器讀取 ] 功能將資料傳輸至專案檔內。
- 暫存器清單功能支援 S、I、O、M、C、D、G 暫存器。但由於 C 暫存器為讀取專用，因此只可引用。

## 顯示暫存器對照表

下列為暫存器對照表的背景顏色意義。

綠	被使用在階梯圖程式中的暫存器
紅	重複暫存器（被複數資料型使用的暫存器）

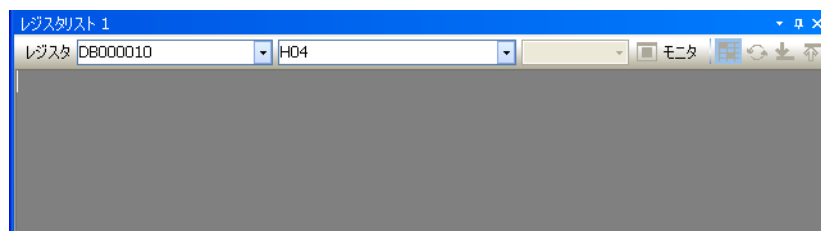
下列為顯示暫存器對照表的操作方式。

### 1. 點選暫存器清單子視窗（1、2、3）的任一標籤。

點選啟動器的 [ 監視 ] - [ 暫存器清單 ] 後，將自動顯示暫存器清單 1 子視窗。

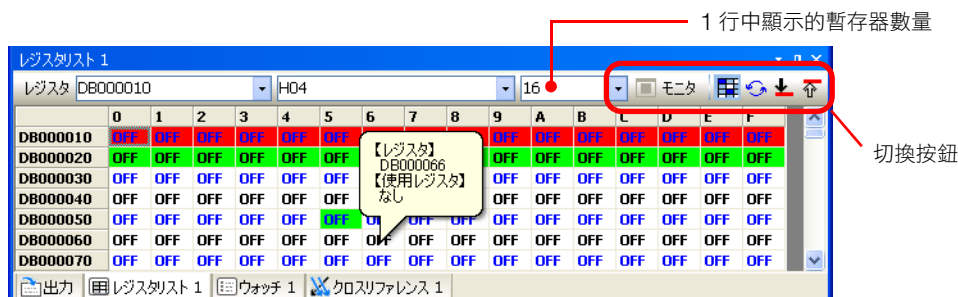
（註）關於暫存器子視窗（1、2、3），可從主選單選擇 [ 顯示 ] - [ 暫存器清單 ] - [ 暫存器清單 1 / 暫存器清單 2 / 暫存器清單 3 ]，以切換清單顯示顯示。

### 2. 在 [ 暫存器 ] 欄內輸入想要顯示暫存器對照表之暫存器的位址。要顯示 D 暫存器的清單時，亦請輸入程式編號（如下列內容所示）。



### 3. 按一下 Enter 鍵。

顯示指定暫存器顯示在最上一行的暫存器對照表。



D 暫存器的暫存器對照表範例及快顯範例

レジスタ	0	1	2	3	4	5	6	7
MW00014	0	0	0	0	0	0	0	0
MW00022	0	0	0	0	0	0	0	0
MW00030	0	0	0	0	0	0	0	0
MW00038	0	0	0	0	0	0	0	0
MW00046	0	0	0	0	0	0	0	0
MW00054	0	0	0	0	0	0	0	0
MW00062	0	0	0	0	0	0	0	0

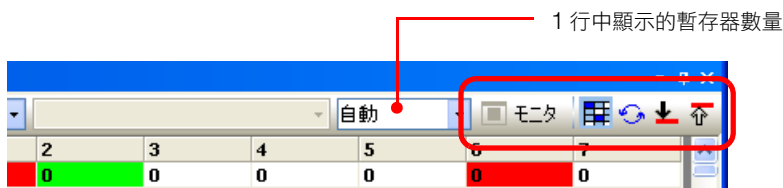
M 暫存器的暫存器對照表範例



- 若在暫存器對照表上移動游標，便會以快顯方式顯示游標停留位置上的暫存器及暫存器使用狀況。
- 可變更 1 行中能顯示的暫存器個數。此外亦可透過清單右上角的 5 顆按鈕，切換顯示內容。
- 若在暫存器清單上點擊右鍵，從彈出式選單中選擇 [10 進制 / 16 進制 / BIN / ASCII]，便可變更「數值」欄的資料類型。但資料類型「B」及「F」無法變更。
- 每隔 1 行以藍字或黑字標示。
- [監視] 圖示在連接運動控制器時（線上連線時）為有效。

## 切換暫存器對照表的顯示內容

可變更要在 1 行中顯示的暫存器個數。此外亦可透過右上角的 5 顆按鈕，切換暫存器對照表的顯示內容。



### ◆ 1 行中顯示的暫存器數量

可透過直接輸入數值或從清單中選擇數字，在 1 ~ 16 的範圍內變更 1 行中能顯示的暫存器個數（Bit 型固定為「16」）。若選擇 [自動]，將配合暫存器清單子視窗的顯示大小，自動設定要顯示的暫存器個數。

### ◆ 監視 ON ( ) / OFF ( ) 按鈕

僅在線上連線時有效。點選後便可切換 ON/OFF。

在監視 ON 時，會持續更新並顯示暫存器資料。監視 OFF 時不會更新。

### ◆ 顯示 ( ) / 隱藏 ( ) 暫存器對照表按鈕

點選後便可切換顯示 / 隱藏。

[顯示] 時：階梯圖程式使用暫存器的背景為綠色顯示，兩個以上料類型使用暫存器的背景為「紅色」顯示。


[隱藏] 時：所有暫存器的背景皆為「白色」顯示。

### ◆ 重新整理暫存器對照表按鈕 ( )

點選後便可更新暫存器對照表的顯示內容。



註記

前述 [ 顯示 / 不顯示暫存器對照表 ] 按鈕設定成不顯示時 (  ) 為無效。


### ◆ 搜尋重複暫存器按鈕 ( / )

可搜尋並顯示重複暫存器的按鈕。[↑] 按鈕可朝上，[↓] 按鈕可往下搜尋重複暫存器。

結果將顯示在 [ 暫存器清單 ] 子視窗，重覆的暫存器顯示藍色背景。



註記

當 [ 顯示 / 不顯示暫存器對照表 ] 按鈕設定成不顯示時 (  ) 為無效。

レジスタ	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MW00014	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MW00029	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MW00044	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MW00059	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MW00074	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MW00089	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MW00104	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

## 編輯資料

在暫存器對照表雙擊資料儲存格或按下 F2 鍵，將顯示文字游標，可進行下列編輯操作。

- 直接輸入資料
- 刪除 ( 0 設定 )
- 複製、貼上

按下 Enter 鍵，便可確定編輯內容。線上時資料的變更直接反映在運動控制器的動作上。



## 5.8

## 調整面板

調整面板中可登錄任意暫存器，顯示目前數值資料並進行編輯。不僅可顯示數值，也能顯示註解標記或指示燈顯示。

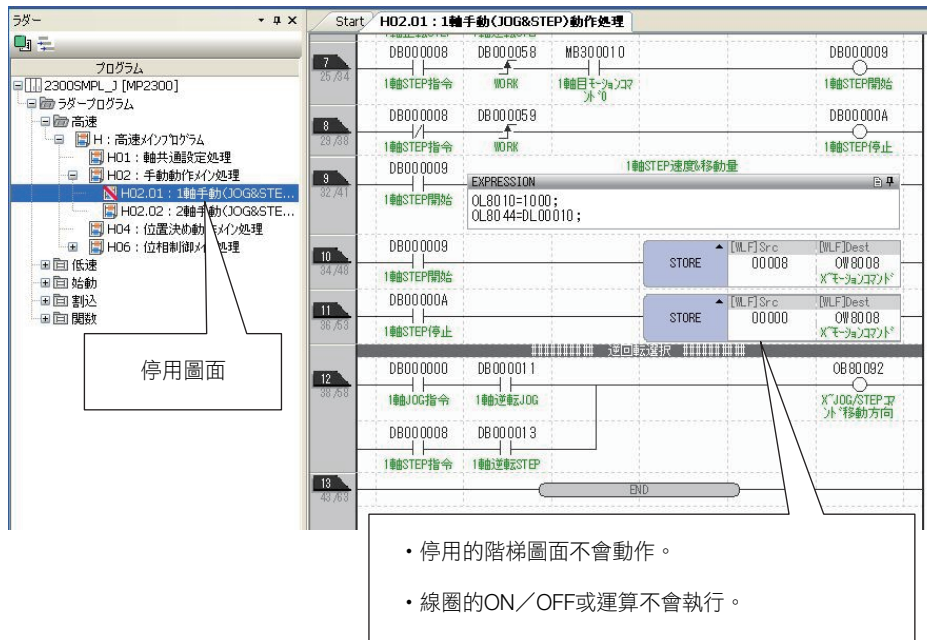
能夠進行製作應用程式的運轉操作或動作確認。

調整 Visual 監視器，可配合狀況顯示資料。

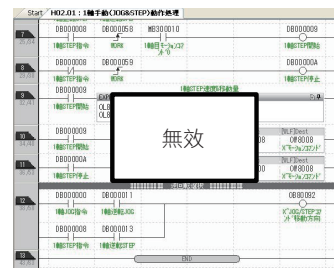
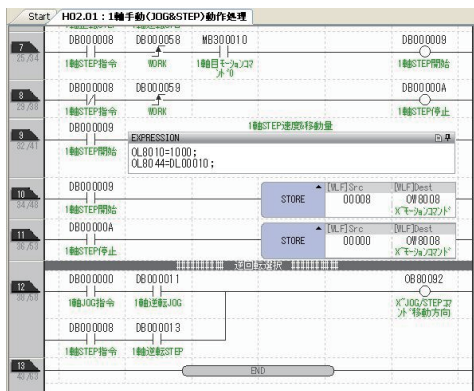
変数	コメント	現在値	単位	Visualモニタ
H02: 手動動作メイン処理				
DW00010 [L]		0		0
IB80000		ON		●
IB80800		OFF		○
IL8016		0		0
IL8096		0		0
DW00010 [L]		0		0
MB300000		OFF		○
MB300001		OFF		○
DW00010 [L]		0		0
DB000010 [H02.01]	1軸正転JOG	OFF		○
DB000011 [H02.01]	1軸逆転JOG	OFF		○
DB000010 [H02.02]	2軸正転JOG	OFF		○
DB000011 [H02.02]	2軸逆転JOG	OFF		○
DB000012 [H02.01]	1軸正転STEP	OFF		○
DB000013 [H02.01]	1軸逆転STEP	OFF		○
DB000012 [H02.02]	2軸正転STEP	OFF		○

# 5.9 階梯圖程式的有效／無效

階梯圖程式的有效／無效功能，能夠以圖面為單位切換階梯圖程式的有效／無效。



可暫時停用加入伺服馬達的伺服 ON 或 JOG 運轉等程序的階梯圖，進行 MPE720 試運轉及能否以模組架構定義正確進行操作的確認。



停用階梯圖圖面



因階梯圖面移動，無法順利由MPE720操作馬達。

可由MPE720順利操作馬達。

## 5.10 監看

在「監看」功能中，可透過監看子視窗（1、2、3）監視指定的暫存器 S、I、O、M、C、D 的數值及註釋。運動控制器連接時可即時監視。還可進行值的編輯。



註記

使用專案連結時，只有已登錄在監看內的資料會儲存到運動控制器。若欲將監看資料套用至專案檔案，請從運動控制器一併傳輸。

### 顯示監看資料

1. 選擇任一個監看子視窗（1、2、3）標籤。

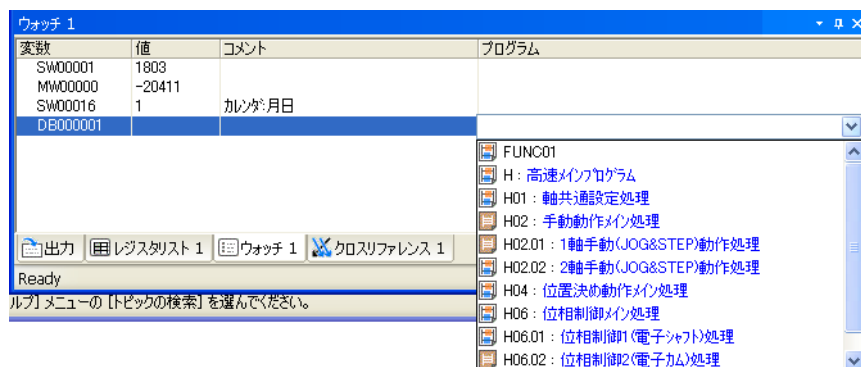
點選啟動器的 [ 監視 ] - [ 監看 ] 後，將自動顯示監看 1 子視窗。

（註）關於監看子視窗（1、2、3），可從主選單選擇 [ 顯示 ] - [ 監看 ] - [ 監看 1 / 監看 2 / 監看 3 ]，以切換監看顯示。

2. 雙擊 [ 變數 ] 欄或按下 F2 鍵顯示文字游標後，輸入要監看的暫存器或變數暫存器。

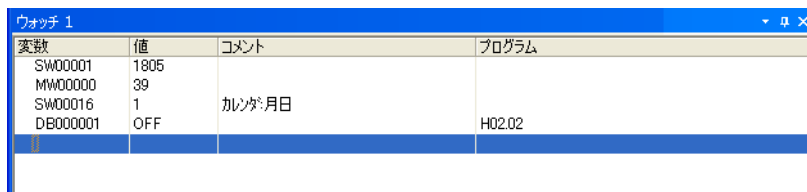
（註）1. 亦可從階梯圖程式及變數子視窗中進行拖曳或複製 & 貼上。

2. 要監看 D 暫存器時，亦請輸入程式編號（如下列內容所示）。



3. 按一下 Enter 鍵。

將顯示指定暫存器的內容。



在行按右鍵，從彈出式選單中選擇 [10 進制 / 16 進制 / BIN / ASCII]，便可變更 [ 數值 ] 欄的資料類型。

## 編輯 [ 數值 ] 欄位

雙擊 [ 數值 ] 欄或按下 F2 鍵後，可在 [ 數值 ] 欄顯示文字游標，可直接輸入資料或複製 & 貼上。



註記

無法編輯註解。

輸入資料並按下 Enter 鍵後，便可確定編輯內容。



註記

線上時資料的變更直接反映在運動控制器的動作上。

## 5.11 安全功能

MPE720 Ver.7 具備下列安全功能。以專案或程式（圖面）為單位，利用這些安全功能設定存取權限，就能保護您的資料。

- **用戶管理（用戶名稱／密碼設定）**  
能夠登錄或變更具具有開啟專案權限的用戶。  
在連接運動控制器情況下進行本設定時，將成為設定對運動控制器的存取權限設定。
- **設定專案密碼**  
可設定專案檔的開啟密碼。
- **設定程式密碼**  
可設定階梯圖程式以及運動程式的開啟密碼。可對不同程式分別設定密碼。
- **線上安全功能**  
可設定讀取運動控制器中資料的安全金鑰（密碼）與限制權限。能夠將由運動控制器讀取程式資料或開啟程式等的功能，限制開放給具有限制權限以上權限的用戶。

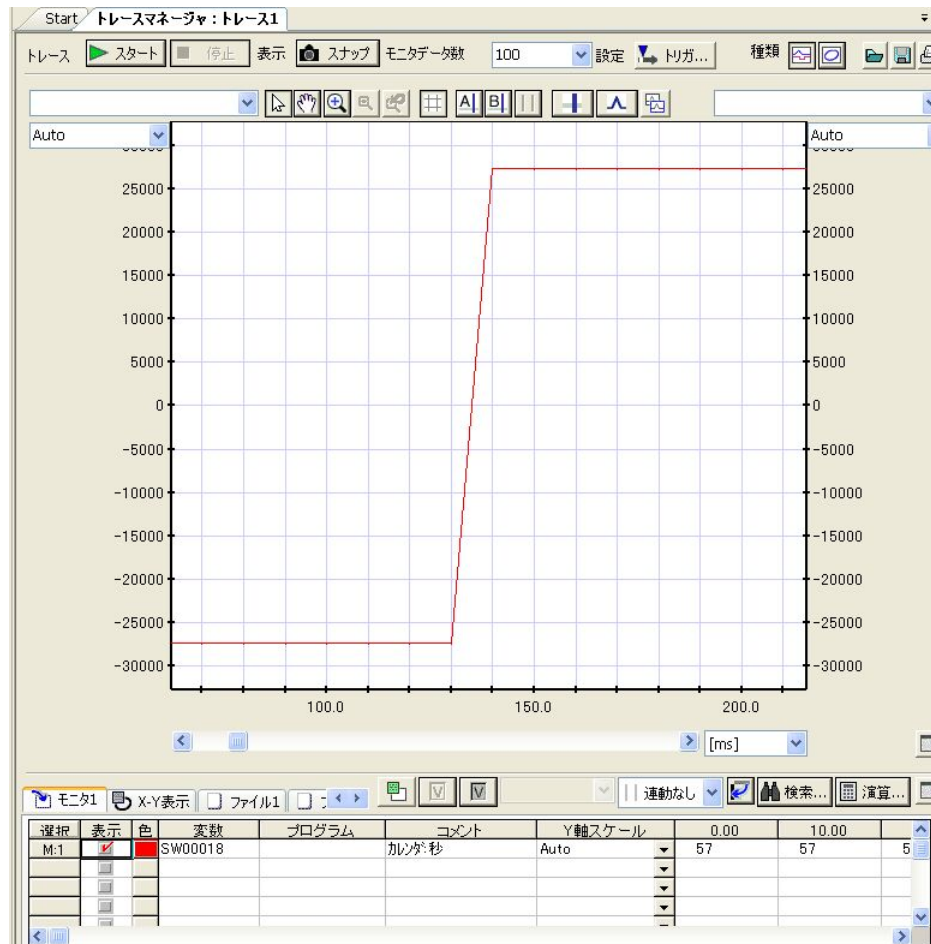
## 5.12 追跡機能

MPE720 Ver.7 具有下列 3 項追跡功能。

- 即時追跡  
可以使用指定暫存器即時監視圖表。
- 數據追跡  
可在指定時間由運動控制器收集指定的暫存器，運算其資料，並描繪成圖表。  
分析指定時間的暫存器資料，可以作為階梯圖程式的除錯功能利用。
- XY 追跡  
每次掃描可取得 X 軸、Y 軸的位置資料，並以 2 維平面圖表顯示。

3 項功能均可將追跡的數據輸出成 CSV 檔案。  
請利用於階梯圖程式及運動程式的動作確認及除錯。

數據追跡視窗的範例，如下所示。

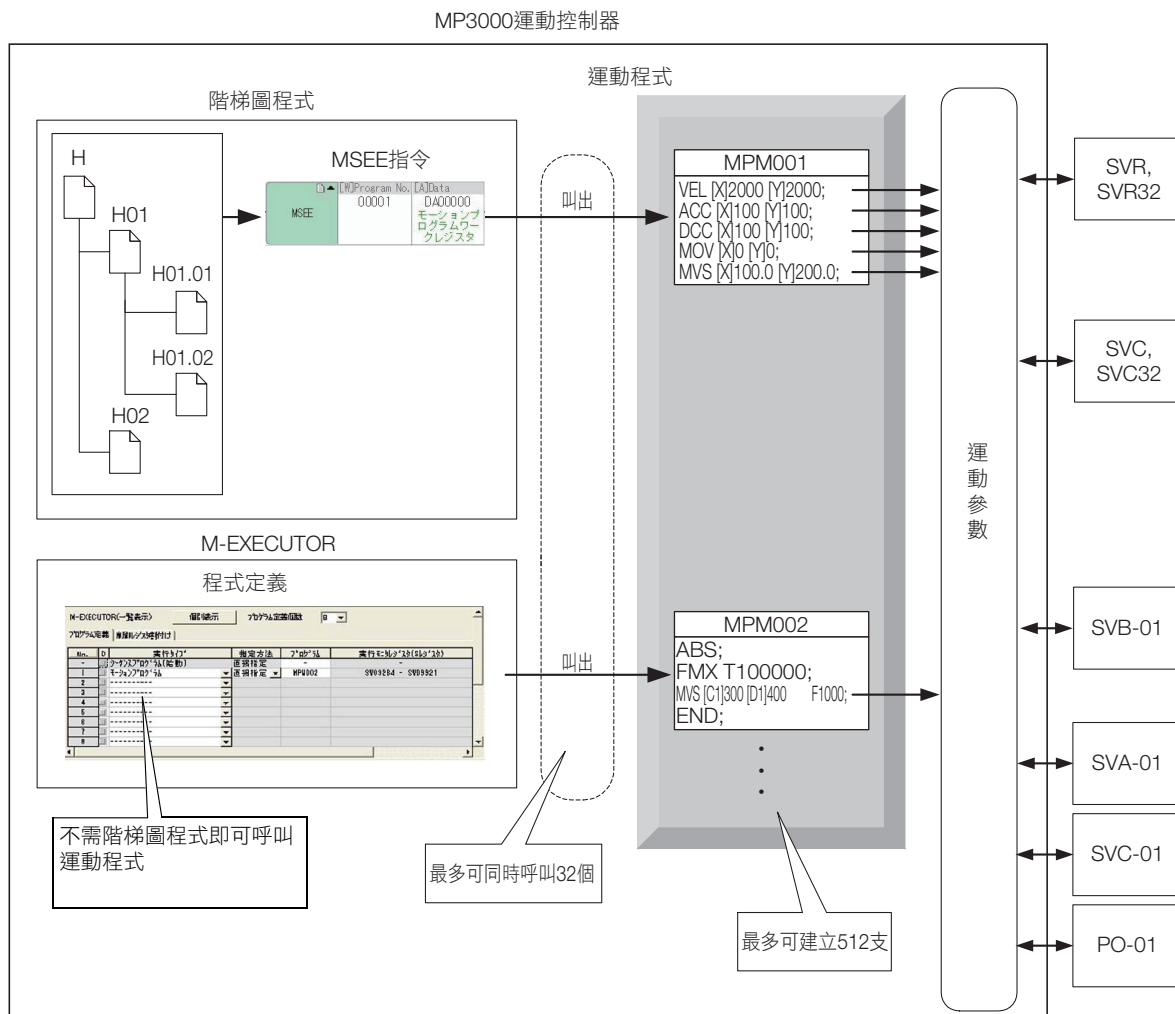


## 5.13 進階使用方式

### 運動程式

運動程式是以純文字語法的動作語言進行敘述的程式。使用運動程式，不管是基本的運動控制或運算或是直線補間、圓弧補間等複雜動作，都能夠簡單進行敘述。

運動程式能透過階梯圖程式的 DWG.H（高速掃描處理圖）敘述 MSEE 指令，或是登錄於 M-EXECUTOR 模組的執行登錄視窗後執行。



關於運動程式的詳情，請參閱以下手冊。

📖 MP3000 系列 運動程式 程式設計手冊（資料編號：SIJP C880725 14）





# 系統服務暫存器一覽

## 附錄 A

本章將就運動控制器系統準備的系統暫存器當中的系統服務暫存器部分，進行說明。

全圖面共通 .....	A-2
DWG.H（高速掃描處理圖面）專用 .....	A-3
DWG.L（低速掃描處理圖面）專用 .....	A-4
掃描執行狀態和日曆 .....	A-5
系統程式軟體編號和程式記憶內容剩餘容量 .....	A-6

系統暫存器藉由運動控制器所準備的暫存器，即可讀取系統的錯誤資訊及運轉狀況等。

	內容
SW000000	系統服務暫存器
SW000030	系統狀態
SW000050	系統錯誤狀態
SW000080	使用者運算錯誤狀態（概要）
SW000090	系統服務執行狀態
SW000110	使用者運算錯誤狀態（詳細資訊）
SW000190	警報計數器及警報清除
SW000200	系統輸入輸出錯誤狀態
SW000504	系統預約
SW000652	CF 卡相關系統暫存器 （僅限 MP2200/CPU-02，CPU-03）
SW000698	中斷狀態
SW000800	模組資訊
SW001312	系統預約
SW001411	MPU-01 模組系統狀態
SW002048	系統預約
SW003200	運動程式資訊
SW005200 ～ SW008191	系統預約

系統服務暫存器分為下列 5 種。

- 全圖面共通
- DWG.H（高速掃描處理圖面）專用
- DWG.L（低速掃描處理圖面）專用
- 掃描執行狀態和日曆
- 系統程式軟體編號和程式記憶內容剩餘容量

## 全圖面共通

名稱	暫存器編號	備註
系統預約	SB000000	（未使用）
H（高速）掃描	SB000001	高速掃描啟動後，僅 1 次掃描 ON
L（低速）掃描	SB000003	低速掃描啟動後，僅 1 次掃描 ON
常時（ON）	SB000004	常時 ON（=1）
系統預約	SB000005， SB000006	（未使用）
H（高速）掃描執行中	SB000007	高速掃描執行中 ON（=1）
系統預約	SB000008 ～ SB00000F	（未使用）

## DWG.H (高速掃描處理圖面) 專用

H (高速) 在掃描啟動後開始動作。

名稱	暫存器編號	備註
1 次掃描閃爍繼電器	SB000010	
0.5 s 閃爍繼電器	SB000011	
1.0 s 閃爍繼電器	SB000012	
2.0 s 閃爍繼電器	SB000013	
0.5 s 取樣繼電器	SB000014	
1.0 s 取樣繼電器	SB000015	
2.0 s 取樣繼電器	SB000016	
60.0 s 取樣繼電器	SB000017	
掃描處理開始 1.0 s 後繼電器	SB000018	
掃描處理開始 2.0 s 後繼電器	SB000019	
掃描處理開始 5.0 s 後繼電器	SB00001A	

## DWG.L (低速掃描處理圖面) 專用

L (低速) 在掃描啟動後開始動作。

名稱	暫存器編號	備註
1 次掃描閃爍繼電器	SB000030	
0.5 s 閃爍繼電器	SB000031	
1.0 s 閃爍繼電器	SB000032	
2.0 s 閃爍繼電器	SB000033	
0.5 s 取樣繼電器	SB000034	
1.0 s 取樣繼電器	SB000035	
2.0 s 取樣繼電器	SB000036	
60.0 s 取樣繼電器	SB000037	
掃描處理開始 1.0 s 後繼電器	SB000038	
掃描處理開始 2.0 s 後繼電器	SB000039	
掃描處理開始 5.0 s 後繼電器	SB00003A	

## 掃描執行狀態和日曆

名稱	暫存器編號	備註
高速掃描設定值	SW00004	高速掃描設定值 (0.1 ms)
高速掃描現在值	SW00005	高速掃描現在值 (0.1 ms)
高速掃描最大值	SW00006	高速掃描最大值 (0.1 ms)
高速掃描設定值 2	SW00007	高速掃描設定值 (1 μs)
高速掃描現在值 2	SW00008	高速掃描現在值 (1 μs)
高速掃描最大值 2	SW00009	高速掃描最大值 (1 μs)
低速掃描設定值	SW00010	低速掃描設定值 (0.1 ms)
低速掃描現在值	SW00011	低速掃描現在值 (0.1 ms)
低速掃描最大值	SW00012	低速掃描最大值 (0.1 ms)
系統預約	SW00013	(未使用)
執行掃描現在值	SW00014	現在執行中掃描現在值 (0.1 ms)
日曆：年	SW00015	西元 1999 年：0099 (BCD) (僅後面兩位數)
日曆：月日	SW00016	12 月 31 日：1231(BCD)
日曆：時分	SW00017	23 時 59 分：2359 (BCD)
日曆：秒	SW00018	59 秒：59(BCD)
日曆：週	SW00019	0：週日、1：週一、2：週二、3：週三、4：週四、5：週五、6：週六

## 系統程式軟體編號和程式記憶內容剩餘容量

名稱	暫存器編號	備註
系統程式軟體編號	SW00020	Sxxxx (xxxx 容納於 BCD 值內)
系統編號	SW00021 ~ SW00025	(未使用)
程式記憶體剩餘容量	SL00026	位元組單位
記憶體總容量	SL00028	位元組單位

# 程式範例

---

## 附錄 B

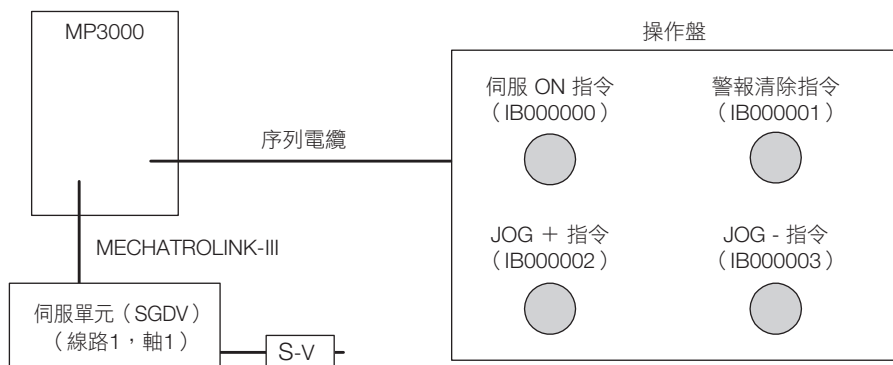
本章將就進行試運轉時的階梯圖程式範例進行說明。

- B.1** 由操作盤進行 JOG 運轉 ..... B-2
- B.2** 運動程式控制 ..... B-3
- B.3** 使用虛擬軸的簡易 2 軸同步運轉 ..... B-4

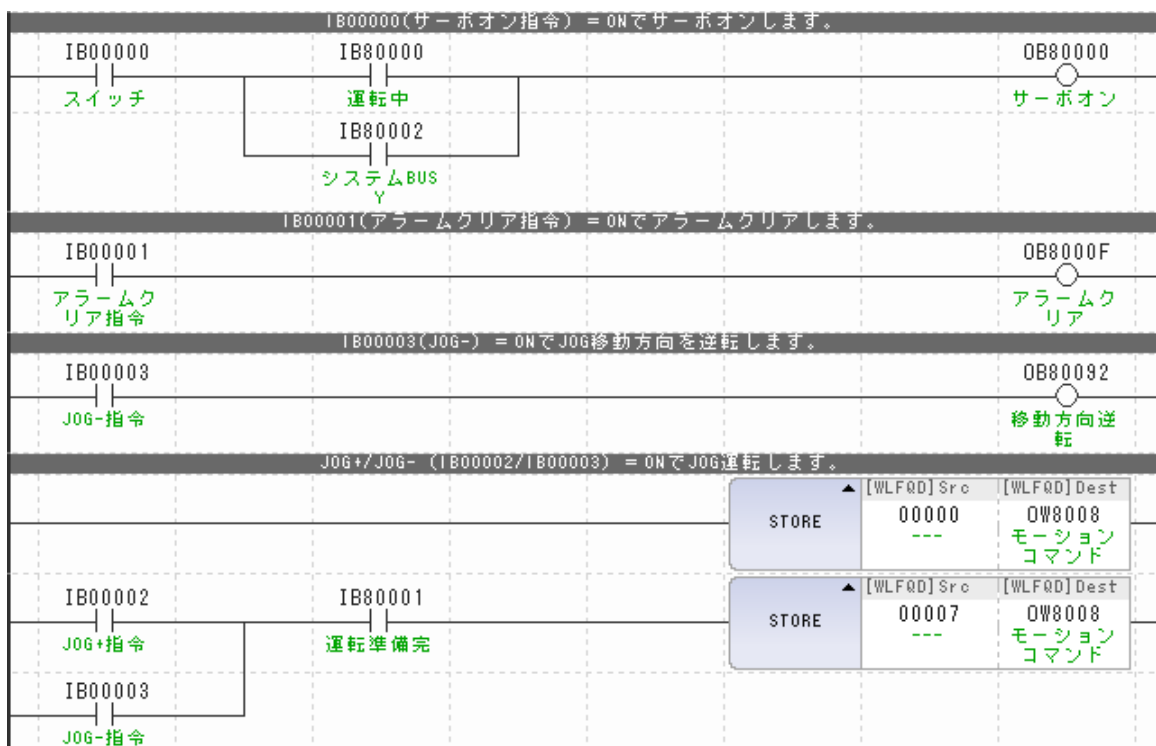
## B.1 由操作盤進行 JOG 運轉

將操作盤與馬達如下連接至運動控制器，由操作盤開關運轉馬達時的架構範例，以及階梯圖程式範例如下。

### ■ 架構範例



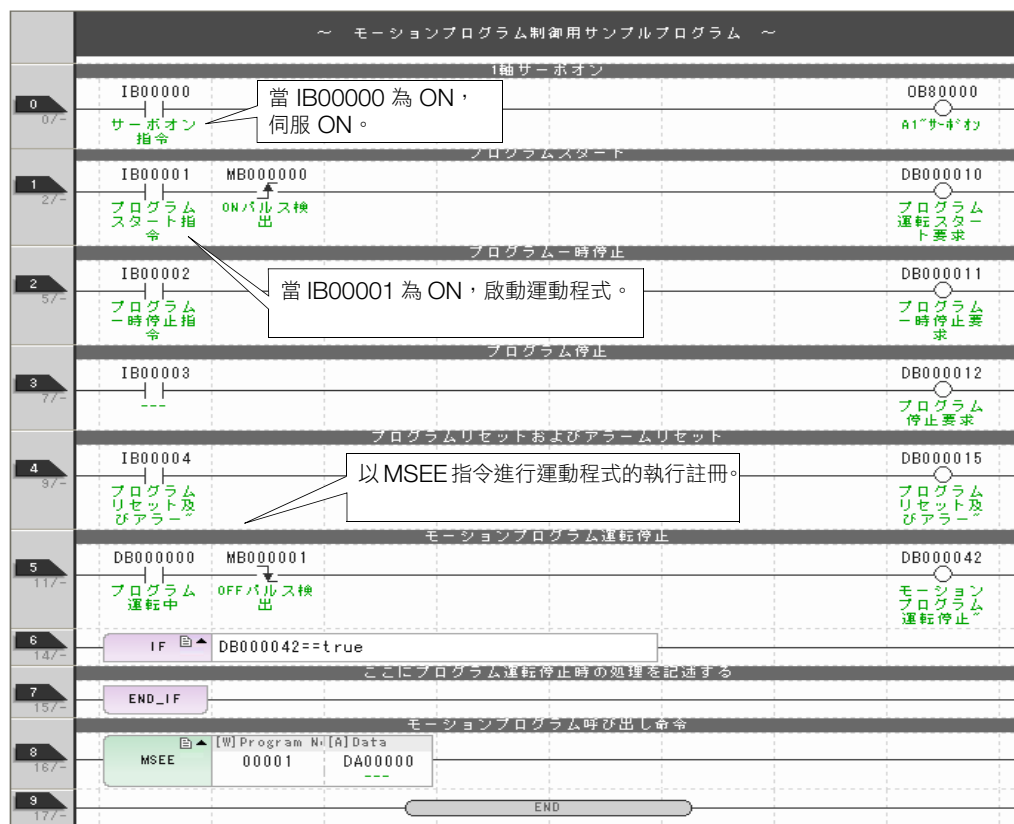
### ■ 階梯圖程式範例





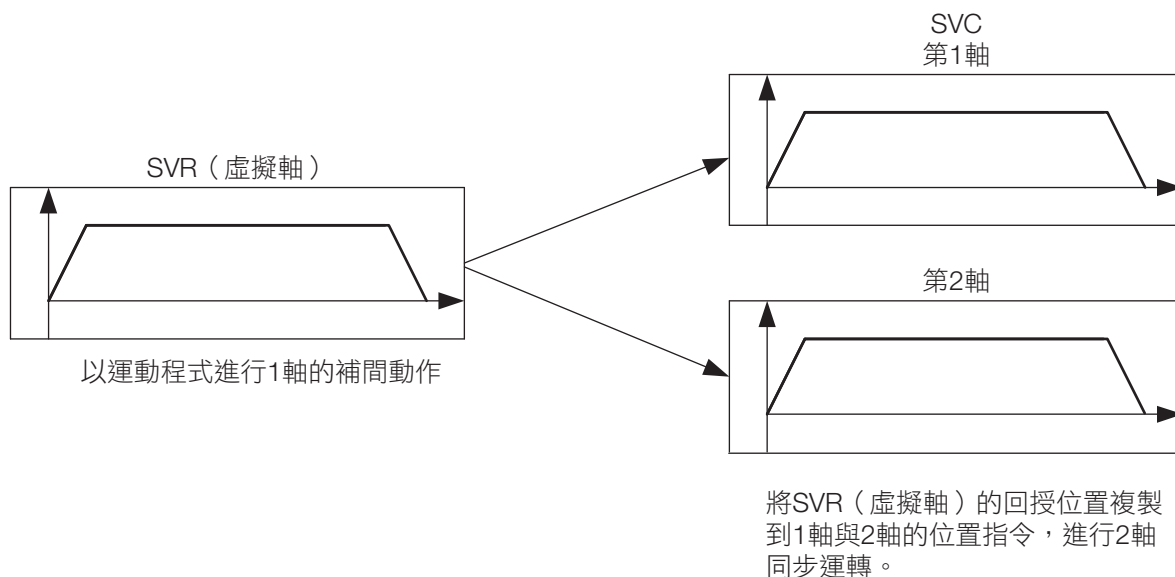
## B.2 運動程式控制

控制運動程式執行的階梯圖程式範例如下所示。



## B.3 使用虛擬軸的簡易 2 軸同步運轉

以運動程式推動 SVR（虛擬軸），以階梯圖程式分配 SVR（虛擬軸）回饋位置至二支實際軸，進行 2 軸同步運轉。



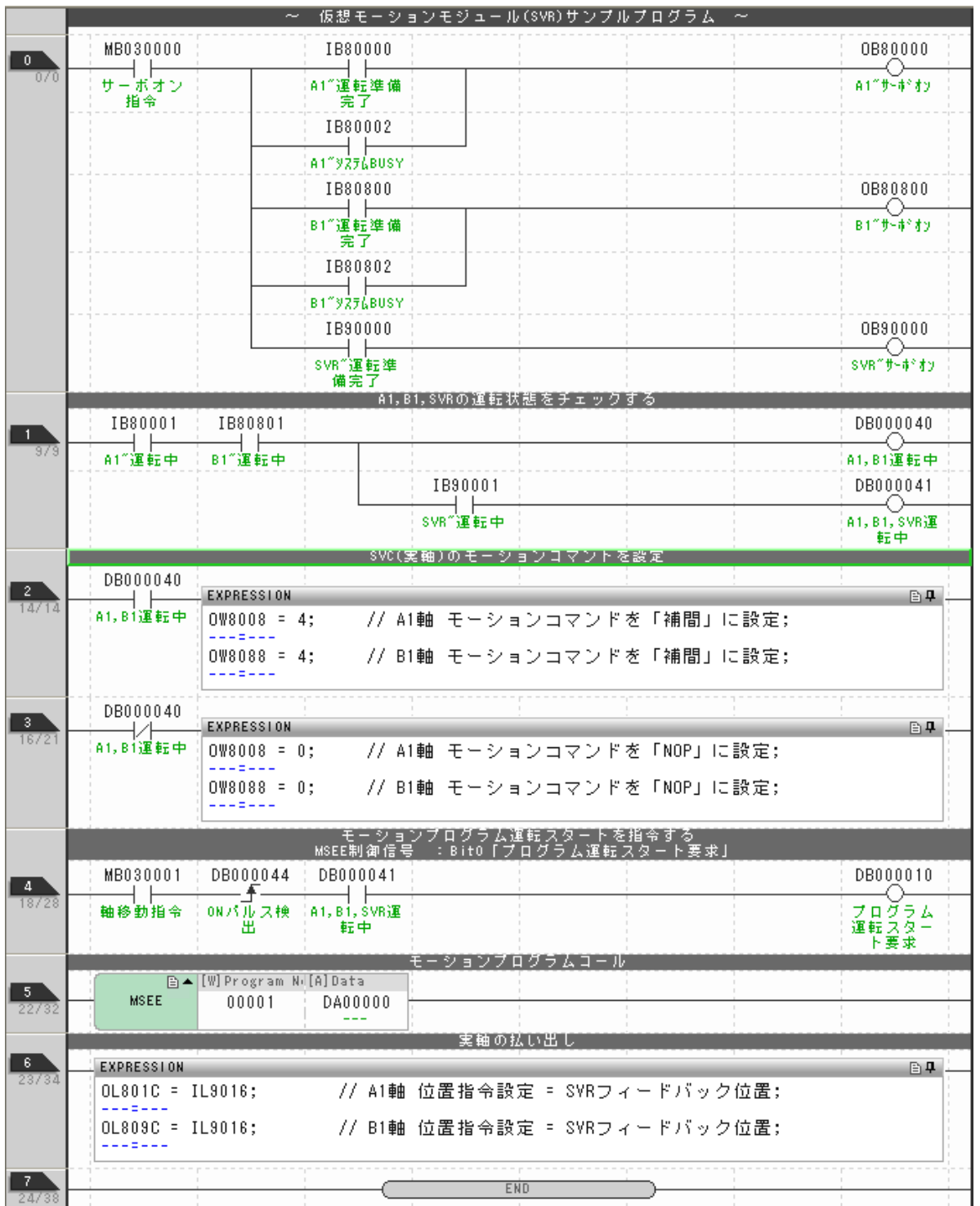
前述運動程式的範例與階梯圖程式範例如下所示。

### ■ 運動程式範例

```

FMX T10000K;          " 最大補間速度設定 K = 1000
INC;                  " 增量模式
IAC T500;             " 補間加速時間 = 500 ms
IDC T500;             " 補間減速時間 = 500 ms
MVS [SVR] 1000K F10000K; " 移動距離 1000000 的補間動作
END;
```

■ 階梯圖程式範例



本程式將略過軸發生異常時的復歸方法。套用本程式時，應預設發生異常時的狀況，新增程式以利安全運轉。

重要



# EXPRESSION 指令 的格式

## 附錄 C

本章將就 EXPRESSION 指令的格式進行說明。

### C.1 可用公式與構成要素 . . . . . C-2

運算子 . . . . . C-2

操作數 . . . . . C-3

EXPRESSION 指令可以使用的指令 . . . . . C-4

### C.2 敘述方式限制 . . . . . C-5

算術運算子 . . . . . C-5

比較運算子 . . . . . C-5

邏輯運算子 . . . . . C-5

指定運算子 . . . . . C-6

函數 . . . . . C-6

括弧 . . . . . C-6

## C.1 可用公式與構成要素

EXPRESSION 指令可用的公式，是由運算子、運算數（常數與變數）以及函數所構成。在此說明前述的構成要素。

### 運算子

#### 運算子種類與可用運算子

運算子種類與可用運算子如下所示。

種類	可用運算子	
算術運算子	+	加法
	-	減法
	*	乘法
	/	除法
	%	剩餘
	&	每位元的積
		每位元的和
邏輯運算子 (僅可對位元型使用)	&&	邏輯乘積
		邏輯總和
	!	邏輯否定
比較運算子	==	等於右邊值
	!=	不等於右邊值
	>	大於右邊值
	>=	大於或等於右邊值
	<	小於右邊值
	<=	小於或等於右邊值
指定運算子	=	將右邊值代入左邊值
預約語	true	對邏輯式的數值（真）
	false	對邏輯式的數值（假）
控制指令	IF, ELSE, IEND	ELSE 可省略

#### 優先順序與組合規則

運算子有其優先順序，組合規則（運算數受評價的順序）將如下表套用。

優先度	運算子	說明	組合規則
高	[] ()	式	從左到右
	- !	單項	由右至左
↑	* / %	乘、除、餘數	從左到右
	+ -	加、減	
	< > <= >=	關係	
	== !=	關係（等價）	
	&	每位元的積	
		每位元的和	
↓	&&	邏輯乘積	
		邏輯總和	
低			

(註) 同行的運算子具有相同優先順序，並依組合規則評價。

## 操作數

### 常數

常數使用整數或實數。

- 整數得使用 64 位元整數值（4 倍長整數）可表現範圍的值。  
(-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807)
- 實數得使用 64 位元 double 型（2 倍精度實數）可表現範圍的值。  
± (2.225E-308 ~ 1.798E+308)

#### 補充

EXPRESSION 指令、IF 指令、WHILE 指令中以 16 進位顯示時，必須記述為 0x□□□□。以 H□□□□ 記述時，將出現錯誤。  
例) H012F ... NG      0x012F ... OK  
其他指令（STORE 等）則需敘述為 H□□□□。

### 變數


EXPRESSION 指令中，可將 C 語言可用的任意變數名稱指定敘述為運動控制器暫存器對應的型式。

此外，C 語言雖不存在 bool 型變數，但運動控制器的位元型暫存器將被當作 bool 型處理。bool 型變數只取 true 或 false 任一值，僅可用於邏輯運算式。

#### ■ 變數名稱限制

可用變數名稱有下列限制。

- 需以數值以外的字元起頭。
- ASCII 字元中僅可使用英文大小寫、底線「\_」以及數字。
- 不得使用與下列函數名稱相同的變數名稱。

 EXPRESSION 指令可以使用的指令（第 C-4 頁）

例	Abc	OK
	Get_input()	OK
	1ab	NG
	Sin	NG

## EXPRESSION 指令可以使用的指令

EXPRESSION 指令的可用指令如下所示。

指令	內容	例	預約語
+	加法	MW00001 = MW00002 + MW00003	○
-	減法	MW00001 = MW00002 - MW00003	○
*	乘法	MW00001 = MW00002 × MW00003	○
/	除法	MW00001 = MW00002 / MW00003	○
%	剩餘	MW00001 = MW00002 % MW00003	○
&	每位元的積	MW00001 = MW00002 & 4096	○
	每位元的和	MW00001 = MW00002   4096	○
&&	邏輯乘積	MB000010 = MB000011 && MB000012	○
	邏輯總和	MB000010 = MB000011    MB000012	○
!	邏輯否定	MB000010 = !MB000011	○
==	等於右邊值	MB000010 = MB000011 == true	○
>=	右邊值在左邊值之上	MB000010 = MW00020 >= MW00021	○
>	右邊值比左邊值大	MB000010 = MW00020 > MW00021	○
<	右邊值比左邊值小	MB000010 = MW00020 < MW00021	○
<=	右邊值在左邊值之下	MB000010 = MW00020 <= MW00021	○
=	將右邊值代入左邊值	MW00001 = MW00002	○
true	真	MB000010 = MB000011 == true	○
false	假	MB000010 = MB000011 == false	○
sin()	SIN	MW00001 = sin(MW00002)	○
cos()	COS	MF00002 = cos(MF00004)	○
atan()	ARCTAN	MF00002 = atan(MF00004)	○
tan()	TAN	MF00002 = tan(MF00004)	○
()	括弧	MW00001 = (MW00002 + MW00003) / MW00004	○
asin()	ARCSIN	MF00002 = asin(MF00004)	○
acos()	ARCCOS	MF00002 = acos(MF00004)	○
sqrt()	SQRT	MW00001 = sqrt(MW00002)	○
abs()	ABS	MW00001 = abs(MW00002)	○
exp()	EXP	MF00002 = exp(MF00004)	○
log()	LOG 自然對數	MF00002 = log(MF00004)	○
log10()	LOG10 常用對數	MF00002 = log10(MF00004)	○



## C.2

## 敘述方式限制

公式由運算數與運算子組合描述，但其描述方式有著部分限制。未滿足限制條件，將不會辨識為公式。在此說明前述限制。

## 算術運算子

此運算子可以對整數型及實數型的操作數使用。單項負只可使用 1 次。位元運算僅可使用整數型。位元型運算數不得用於算術運算。此外，當計算值超過暫存器範圍，不會自動變換資料類型，使用者應分配適當資料類型至暫存器。

例	MW00001 = MW00002 + MW00003	OK
	MW00001 = MW00002 / 345	OK
	MF00002 = (MW00004 + MF00002) / (ML00018 + MW00008)	OK
	MW00001 = MW00002 & 4096	OK
	MB000010 = MB000011 - MB000012	NG
	MW00001 = MB000011 * MW00001	NG

## 比較運算子

此運算子可以對整數型及實數型的操作數使用。左邊必須以位元型暫存器敘述。以「==」或「!=」對整數位元性型運算數進行比較時，比較對象請使用 true/false。

例	MB000010 = MW00002 != MW00003	OK
	MB000010 = MF00002 < 99.99	OK
	MB000010 = MW00002 >= MW00003	OK
	MB000010 = MB000011 == true	OK
	MB000010 = MB000011 != 0	NG
	MB000010 = MB000011 == 1	NG

## 邏輯運算子

此運算子僅可對位元型運算數使用。

例	MB000010 = MB000011 && MB000012	OK
	MB000010 = !MB000011	OK
	MB000010 = (MW000020 >= 50) && MB000011	OK
	MB000010 = MW00001    MW00002	NG
	MB000010 = !MW00001	NG

## 指定運算子

只要是實數型或整數型，即使是不同的資料類型也可對實數型及整數型暫存器進行代入。但實數型代入整數型暫存器時，將產生捨入誤差。

而位元型暫存器僅可以邏輯值（位元型暫存器或 true / false）代入。邏輯值以外的值代入位元型暫存器時，將與 0 或 0.0 比較，轉換為真假將代入的編碼。

位元型不得代入非位元型的暫存器。

例	MW00001 = MW00002;	OK
	MF00000 = MW00002 / 345;	OK
	MB000010 = MB000010;	OK
	MW00010 = MB0000101;	NG
	MW00001 = true;	NG

## 函數

與函數相對的引數及返回值，依循運動控制器的函數規格。

也既是說，對於 sin（）、cos（）及 atan（）輸入整數及整數型暫存器則返回整數輸出值，輸入實數及實數型暫存器則返回實數輸出值。

tan（）引數為實數，因此輸入整數型暫存器也將視為實數型。

例	MW00001 = sin(MW00002);	OK
	MF00002 = cos(MF00000×3.14);	OK
	MW00001 = -atan(MF00002);	OK

## 括弧

### ■ 組合

可用「（」與「）」組合多個式子。

例	MW00001 = -(MW00002 + 10) / (MW00003 - MW00005);	OK
---	--	----

### ■ 配列

與 C 語言相同，可以「[」與「]」指定陣列。

例	MW00001 = MW00002[100];	OK
	MW00001 = MW00002[MW00003];	OK
	MB000010 = MB000020[0];	OK

# 運動參數的相關注意事項

---

## 附錄 D

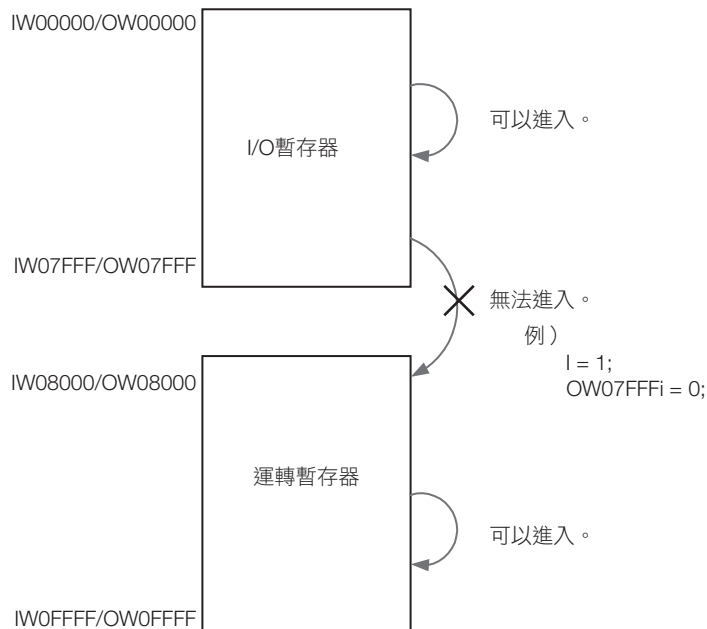
本章將就運動參數相關注意事項進行說明。

運動參數相關注意事項，說明如下。

■ 切勿使用附註說明，由 I/O 暫存器參照運動暫存器。

I/O 暫存器與運動暫存器並未分配至相連記憶內容。

要使用附註說明，請使其僅存取 I/O 暫存器或動作暫存器範圍內的暫存器。



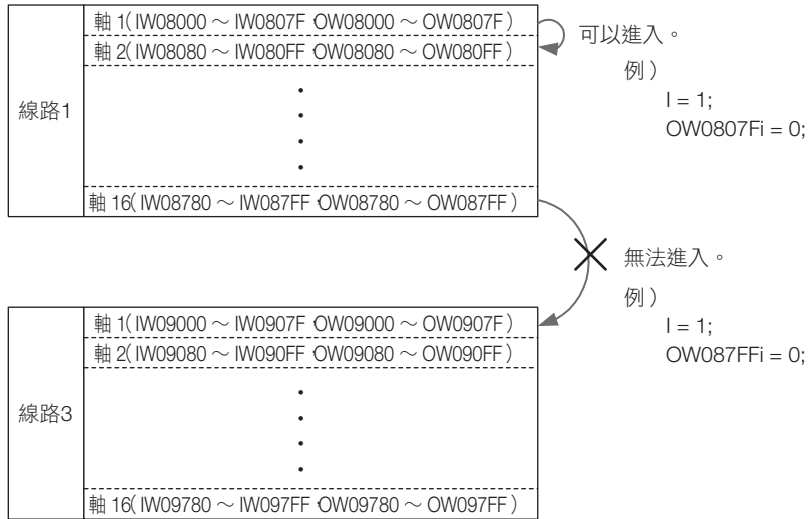
■ 切勿使用附註說明，參照其他線路的運動暫存器。

不同線路的運動暫存器並未分配至相連記憶內容。

要使用附註說明，請使其僅存取各線路動作暫存器範圍內的暫存器。

如為同一線路，亦可存取不同軸的動作暫存器。

線路編號	1 軸	2 軸	...	16 軸
1	OW08000 ~ OW0807F	OW08080 ~ OW080FF	...	OW08780 ~ OW087FF
3	OW09000 ~ OW0907F	OW09080 ~ OW090FF	...	OW09780 ~ OW097FF
5	OW0A000 ~ OW0A07F	OW0A080 ~ OW0A0FF	...	OW0A780 ~ OW0A7FF
7	OW0B000 ~ OW0B07F	OW0B080 ~ OW0B0FF	...	OW0B780 ~ OW0B7FF
9	OW0C000 ~ OW0C07F	OW0C080 ~ OW0C0FF	...	OW0C780 ~ OW0C7FF
11	OW0D000 ~ OW0D07F	OW0D080 ~ OW0D0FF	...	OW0D780 ~ OW0D7FF
13	OW0E000 ~ OW0E07F	OW0E080 ~ OW0E0FF	...	OW0E780 ~ OW0E7FF
15	OW0F000 ~ OW0F07F	OW0F080 ~ OW0F0FF	...	OW0F780 ~ OW0F7FF
17	OW18000 ~ OW1807F	OW18080 ~ OW180FF	...	OW18780 ~ OW187FF
19	OW19000 ~ OW1907F	OW19080 ~ OW190FF	...	OW19780 ~ OW197FF
21	OW1A000 ~ OW1A07F	OW1A080 ~ OW1A0FF	...	OW1A780 ~ OW1A7FF
23	OW1B000 ~ OW1B07F	OW1B080 ~ OW1B0FF	...	OW1B780 ~ OW1B7FF
25	OW1C000 ~ OW1C07F	OW1C080 ~ OW1C0FF	...	OW1C780 ~ OW1C7FF
27	OW1D000 ~ OW1D07F	OW1D080 ~ OW1D0FF	...	OW1D780 ~ OW1D7FF
29	OW1E000 ~ OW1E07F	OW1E080 ~ OW1E0FF	...	OW1E780 ~ OW1E7FF
31	OW1F000 ~ OW1F07F	OW1F080 ~ OW1F0FF	...	OW1F780 ~ OW1F7FF





# 運動控制器的規格一覽

---

## 附錄 E

本章將說明運動控制器支援的程式規格。

運動控制器程式的相關規格，如下所示。

規格		CPU 單元 / CPU 模組	備註
運動 程 式	程式數	最大 512 支	可以製作運動程式 / 序列程式合計 512 支程式。
	群組數	16 群組	-
	作業數	最多 32 個任務 (可同時執行的運動程式數)	-
	平行處理數 (每 1 任務)	最多 8 組平行 (由下列 4 個模式選擇) <ul style="list-style-type: none"> <li>• 主程式 4 平行 × 子程式 2 平行</li> <li>• 主程式 8 平行 × 子程式 1 平行</li> <li>• 主程式 2 平行 × 子程式 4 平行</li> <li>• 主程式 1 平行 × 子程式 8 平行</li> </ul>	可使用 MPE720 切換模式。
	執行登錄	<ul style="list-style-type: none"> <li>• 由階梯圖程式使用 MSEE 指令</li> <li>• 使用 M-EXECUTOR</li> </ul>	-
	啟動方法	控制訊號 Bit 0 (程式運轉開始要求) 偵測到上升沿，則程式啟動	-
	定位速度對應的覆寫數	可設定 0.01% ~ 327.67% 的範圍	-
	動作模式	ABS/INC 模式	使用專用命令 (ABS/INC)，可以切換模式。
	指令單位	<ul style="list-style-type: none"> <li>• SVC/SVC32/SVC-01/SVB-01/SVR/SVR32 pulse, mm, deg, inch, μm</li> <li>• SVA-01/PO-01 pulse, mm, deg, inch</li> </ul>	-
	最小指令單位	<ul style="list-style-type: none"> <li>• pulse 1</li> <li>• mm, deg, inch, μm 1, 0.1, 0.01, 0.001, 0.0001, 0.00001</li> </ul>	-
	指令範圍	- 2147483648 ~ + 2147483647 (附 32 位元符號)	-
	同時控制軸數 (每 1 任務)	<ul style="list-style-type: none"> <li>• 定位、直線補間、原點重設、具略過功能的直線補間、時間指定定位 最多 32 軸</li> <li>• 循環補間 2 軸</li> <li>• 螺旋補間 3 軸</li> <li>• 外部定位 1 軸</li> </ul>	-
	同時控制鏡頭數	最多 32 顆鏡頭	-
	同時控制機構數	最多 4 個機構	-
序 列 程 式	程式數	最大 512 支 (可由始動處理、高速掃描處理、低速掃描處理選擇執行時間點)	可以製作運動程式 / 序列程式合計 512 支程式。
	作業數	最多 32 個任務 (可同時執行的序列程式數)	-
	平行處理數 (每 1 任務)	平行處理 (PFORK 指令) 不可使用。	-
	執行登錄	使用 M-EXECUTOR	-
	啟動方法	由系統啟動	對 M-EXECUTOR 登錄系統後啟動。

(接下頁)



(承上頁)

規格	CPU 單元 / CPU 模組	備註	
可 存 取 的 暫 存 器	M 暫存器	1048576 個字	電池組備用記憶體
	S 暫存器	65535 個字	電池組備用記憶體
	G 暫存器	2097152 個字	各程式間共用之無電池備援暫存器。
	I 暫存器	65536 個字 + 設定參數 + CPUIF 用	-
	O 暫存器	65536 個字 + 監控參數 + CPUIF 用	-
	C 暫存器	16384 個字	-
	D 暫存器	可在 0 ~ 16384 個字當中指定	各 DWG 固有的內部暫存器。僅符合的 DWG 可使用。

# 索引

<b>符號</b>	
# 暫存器	3-4
<b>數字</b>	
1 的補數 (COM)	4-67
2 倍長整數	3-7
2 倍精度實數	3-7
2 進制轉換 (BIN)	4-69
4 倍長整數	3-7
<b>A</b>	
ASCII 轉換 1 (ASCII)	4-72
ASCII 轉換 2 (BINASC)	4-74
ASCII 轉換 3 (ASCBIN)	4-76
A 接點 (NOC)	4-9
<b>B</b>	
BCD 轉換 (BCD)	4-70
B 接點 (NCC)	4-12
<b>D</b>	
DDC 指令	4-7, 4-151
DWG.A	1-8
DWG.H	1-8
DWG.I	1-8
DWG.L	1-8
D 暫存器	3-4
<b>E</b>	
EXPRESSION 指令	C-2
<b>F</b>	
FOR 文 (FOR, END_FOR)	4-103
<b>G</b>	
G 暫存器	3-3
<b>I</b>	
IF_ELSE 文 (IF, ELSE, END_IF)	4-107
IF 文 (IF, END_IF)	4-105
<b>M</b>	
MPE720 Ver.7 安裝	2-4
<b>P</b>	
PD 控制 (PD)	4-162
PID 控制 (PID)	4-168
PI 控制 (PI)	4-157
<b>V</b>	
Visual 監視器	5-21
<b>W</b>	
WHILE 文 (WHILE, END_WHILE)	4-101

## X

XY 追蹤	5-26
-------	------

## 一畫

一次延遲 (LAG)	4-173
一併傳送	2-13

## 三畫

下降沿 A 接點 (OFFP-NOC)	4-11
下降沿 B 接點 (OFFP-NCC)	4-14
下降脈衝 (OFF-PLS)	4-29
下降變化偵測線圈 (OFFP-COIL)	4-34
上下限值 (LIMIT)	4-155
上升脈衝 (ON-PLS)	4-27
上緣觸發 A 接點 (ONP-NOC)	4-10
上緣觸發 B 接點 (ONP-NCC)	4-13
上緣觸發變化偵測線圈 (ONP-COIL)	4-33
子圖	1-7

## 四畫

不感帶 A (DZA)	4-151
不感帶 B (DZB)	4-153
互斥或 (XOR)	4-82
公式敘述 (EXPRESSION)	4-109
切換暫存器對照表的顯示	5-19
反正切 (ATAN)	4-120
反正弦 (ASIN)	4-118
反函數產生器 (IFGN)	4-184
反餘弦 (ACOS)	4-119
反轉線圈 (REV-COIL)	4-32
引用呼叫程式	5-17
文字→位元組壓縮 (BPRESS)	4-138
文字傳送 (MOVW)	4-130
文字複製 (COPYW)	4-148
比較 (<)	4-84
比較 (=)	4-86
比較 (≠)	4-87
比較 (>)	4-89
比較 (≤)	4-85
比較 (≥)	4-88
比較運算子	C-2, C-5

## 五畫

加法 (ADD (+))	4-39
平方根 (SQRT)	4-111
正切 (TAN)	4-117
正弦 (SIN)	4-113
用戶管理	5-25

## 六畫

交互條件設定	5-10
先進先出 (FINFOUT)	4-241
全局暫存器	3-2
列搜尋 (橫向) (TBLSRC)	4-218

同位轉換 (PARITY) .....	4-71
安全功能 .....	5-25
安全金鑰 .....	5-25
自動配置 .....	2-6
自然對數 (LN) .....	4-122
行搜尋 (縱向) (TBL SRL) .....	4-215

## 七畫

位元 .....	3-7
位元向右位移 (SHFTR) .....	4-146
位元向右旋轉 (ROTR) .....	4-126
位元向左位移 (SHFTL) .....	4-144
位元向左旋轉 (ROTL) .....	4-124
位元組→文字展開 (BEXTD) .....	4-136
位元組交換 (BSWAP) .....	4-150
位元傳送 (MOVB) .....	4-128
位址 .....	3-7
佇列指標清除 (QTBLCL) .....	4-236
伺服單元參數載入 (MLNK-SVR) .....	4-264
伺服單元參數寫入 (MLNK-SVW) .....	4-259
刪除區段 (TBLCL) .....	4-221
即時追蹤 .....	5-26
局部暫存器 .....	3-2
快閃操作 (FLASH-OP) .....	4-269
快閃儲存階梯圖程式 .....	2-18
系統函數指令 .....	4-238
系統服務暫存器 .....	A-2
系統架構模型 .....	2-3
系統暫存器 .....	3-3
系統標準函數指令 .....	4-7

## 八畫

使用者函數 .....	1-13
函數內部暫存器 .....	3-5
函數外部暫存器 .....	3-5
函數產生器 (FGN) .....	4-179
函數輸入暫存器 .....	3-4
函數輸出暫存器 .....	3-4
「取代」對話方塊 .....	5-7
取代 .....	5-7
「取代專案檔內」對話方塊 .....	5-9
呼叫使用者函數 .....	1-18
呼叫使用者函數 (FUNC) .....	4-95
呼叫運動程式 (MSEE) .....	4-93
呼叫圖面 (SEE) .....	4-92
直接輸入連續執行形 (INS) .....	4-96
直接輸出連續執行形 (OUTS) .....	4-98
直線加減速器 1 (LAU) .....	4-189
直線加減速器 2 (SLAU) .....	4-196
表初始化 (SETW) .....	4-134
表資料 .....	1-19

## 九畫

建立使用者函數 .....	1-15
建立表資料 .....	1-19
建立專案 .....	2-5
指定運算子 .....	C-2, C-6
指數 (EXP) .....	4-121
相互參照 .....	5-10
相位前進延遲 (LLAG) .....	4-176
背景程式 .....	1-10
計數器 (COUNTER) .....	4-238
重置線圈 (R-COIL) .....	4-36
限制權限 .....	5-25

## 十畫

乘法 (MUL (?)) .....	4-47
個別傳送 .....	2-13
孫圖 .....	1-7
索引暫存器 (i、j) .....	3-10
脈衝寬變調 (PWM) .....	4-206
訊息接收 (MSG-RCV) .....	4-255
訊息接收 (擴展) (MSG-RCVE) .....	4-257
追蹤 (TRACE) .....	4-245
追蹤功能 .....	5-26
除法 (DIV (KY)) .....	4-49
高速/低速掃描時間之設定 .....	1-10
高速/低速掃描處理圖面的執行排程 .....	1-10

## 十一畫

區段寫入 (TBLBW) .....	4-212
區段讀取 (TBLBR) .....	4-209
基本函數指令 .....	4-6, 4-111
執行擴展程式 (XCALL) .....	4-100
專案檔內的搜尋 .....	5-8
專案檔案內的取代 .....	5-9
常用對數 (LOG) .....	4-123
常數 .....	C-3
常數暫存器 .....	3-4
強制 OFF .....	5-14
強制 ON .....	5-14
控制指令 .....	C-2
排序 (SORT) .....	4-142
符號反轉 (INV) .....	4-66
設定高速圖面動作模式 .....	1-11
設定專案密碼 .....	5-25
設定程式密碼 .....	5-25
設定線圈 (S-COIL) .....	4-35
通電延遲計時器 (TON (1 ms)) .....	4-15
通電延遲計時器 (TON (1 s)) .....	4-23
通電延遲計時器 (TON (10 ms)) .....	4-19

## 十二畫

插入指令	2-9
插入梯級	2-9
替換傳送 (XCHG)	4-132
減少時間 (TMSUB)	4-61
減法 (SUB (-))	4-43
減量 (DEC)	4-57
硬體連接	2-3
程式內的搜尋/取代	5-5
程式控制命令	4-6
絕對值轉換 (ABS)	4-68
階梯圖	1-7
階梯圖子視窗	1-6
階梯圖面的執行時間點	1-3
階梯圖程式	1-2
階梯圖程式的執行中監視	5-4
階梯圖程式動作確認	2-14
階梯圖程式啟用/停用	5-22
階梯圖程式製作	2-8
階梯圖程式編輯視窗	1-6
階梯圖程式編輯器	1-6
階梯語言指令一覽	4-5

## 十三畫

傳送表間區段 (TBLMV)	4-224
匯入 (IMPORT/IMPORTL)	4-279
匯出 (EXPORT/EXPORTL)	4-287
「搜尋」對話方塊	5-5
搜尋	5-5
搜尋/取代	5-5
「搜尋專案檔內」對話方塊	5-8
搜尋資料 (BSRCH)	4-140
準備連接裝置	2-3
經過時間 (SPEND)	4-63
資料表操作指令	4-7, 4-209
資料追蹤	5-26
資料移動指令	4-124
資料暫存器	3-3
資料操作指令	4-6
資料類型	3-7
資訊傳送 (MSG-SND)	4-251
資訊傳送 (擴展) (MSG-SNDE)	4-253
載入運動暫存器 (MOTREG-R)	4-276
運動程式	5-27
運算子	C-2
運算錯誤處理圖	1-7
預約語	C-2

## 十四畫

圖面的執行處理方式	1-9
圖面執行控制	1-9

實數	3-7
實數餘數 (REM)	4-53
對話方塊	
取代	5-7
取代專案檔內	5-9
搜尋	5-5
搜尋專案檔內	5-8
監看	5-23
算術運算子	C-2, C-5

## 十五畫

增加時間 (TMADD)	4-59
增量 (INC)	4-55
寫入佇列表 (QTBLW, QTBLWI)	4-232
寫入階梯圖程式	2-12
寫入運動暫存器 (MOTREG-W)	4-273
數值運算指令	4-5, 4-37
暫存器	3-2
暫存器的種類	3-3
暫存器清單	2-14, 5-18
暫存器對照表	5-18
暫存器編號之組成	3-3
範圍檢查 (RCHK)	4-90
編譯	2-9
線上安全功能	5-25
線上連接	2-7
線圈 (COIL)	4-31
線圈強制 ON / OFF	5-14
從強制線圈清單子視窗變更為強制 ON / OFF	5-14
從階梯圖程式設定強制 ON / OFF	5-14
搜尋強制線圈清單子視窗內的強制線圈	5-14
調整面板	5-21
餘弦 (COS)	4-115

## 十六畫

操作數	C-3
整數	3-7
整數餘數 (MOD)	4-51
輸入暫存器	3-3
輸出暫存器	3-4

## 十七畫

儲存 (STORE)	4-37
總圖	1-7

## 十八畫

擴充減法 (SUBX (-))	4-45
擴展加法 (ADDX (++)	4-41
斷電延遲計時器 (TOFF (1 ms))	4-17
斷電延遲計時器 (TOFF (1 s))	4-25
斷電延遲計時器 (TOFF (10 ms))	4-21
雙重線圈	5-13

**二十畫**

繼電器迴路指令 ----- 4-5, 4-9

**二十二畫**

讀取佇列表 (QTBLR, QTBLRI) ----- 4-228

讀取追蹤資料 (DTRC-RD) ----- 4-247

**二十三畫**

變數 ----- C-3

變數子視窗 ----- 1-6

邏輯或 (OR) ----- 4-80

邏輯乘積 (AND) ----- 4-78

邏輯運算/比較指令 ----- 4-78

邏輯運算子 ----- C-2, C-5

邏輯運算指令 ----- 4-6

顯示監看資料 ----- 5-23

## 改版履歷

有關資料改版的資訊，與資料編號一起記載在本資料封底的右下角。

資料編號 YTWMNCO-15004B <1>-1  
 改版追加序號  
 改版編號  
 Published in Taiwan 2012年 6月  
 發行年月

發行日期	改版編號	改版追加序號	項目編號	變更內容
2016年8月	<7>	1	4.8	新增：掃描時間設定值相關注意事項
2016年1月		0	-	反映與 Web 用 SIJP C880725 13G<6>-1 相同之內容
2015年12月	<6>	1	4.2	變更：關於斷電延遲計時器（TOFF (1 ms)）的敘述
			4.9	變更：關於區段寫入（TBLBW）的表格資料
2015年10月	0	4.8		變更：實數型 PI 指令、PD 指令、PID 指令參數表的 P 增益、I 增益、D 增益規格
				變更：PI 控制、PD 控制、PID 控制的程式範例
			4.10	新增：伺服單元參數載入（MLNK-SVR） 新增：快閃操作（FLASH-OP）
封底		變更：位址		
2015年5月	<5>	0	-	反映與 Web 用 SIJP C880725 13E<4>-1 相同之內容
2015年4月	<4>	1	4.2, 4.5	新增：關於使用者參數的注意事項
			封面，封底，書背	變更：格式
2015年2月	0	-	-	反映與 Web 用 SIJP C880725 13D<3>-1 相同之內容
2015年1月	<3>	1	4.2	新增：補述與時間表及其他指令組合時的內容
			封底	變更：位址
2014年4月	0	所有章節		追加：MP3300 相關敘述
			4.10	追加：IMPORTL 指令及 EXPORTL 指令相關敘述
			封底	變更：位址
2013年8月	<2>	0	-	反映與 Web 用 SIJP C880725 13B<1>-1 相同之內容
			序言	變更：PL 的記述內容
			4.5	變更：JNS 指令及 OUTS 指令的參數 RSSEL 固定架編號 1 ~ 4 → 1 ~ 7 插槽編號 0 ~ 8 → 0 ~ 9
			封底	變更：位址
2012年6月	<1>	1	附錄 C.1	追加：運算子「控制指令」
2012年5月		-	所有章節	全面修改
		4.10	追加：伺服單元參數寫入（MLNK-SVW）	
	封底	變更：位址		
2011年9月	-	-	-	初版刊行

運動控制器 MP3000系列

# 階梯圖程式

## 程式設計手冊

---

### 台灣安川電機股份有限公司

事務所/技術服務中心

地址：23143新北市新店區北新路3段207號12樓

TEL：(02)8913-1333 FAX：(02)8913-1513/1519

台南服務中心

地址：74144台南市新市區創業路18號2樓

TEL：(06)505-1432 FAX：(06)505-6405

代理商 / 經銷商

---

**YASKAWA**

安川電機

本產品的終端使用者若為軍事相關單位，或用於製造軍事武器之用途，可能會被列為《外幣匯率及國際貿易法》所規定之限制出口的對象產品，出口時請務必經過審慎的審查，並辦妥必要的出口手續。

部分額定、規格、尺寸等可能因產品改良而有所變更，恕不另行通知。  
有關此資料內容相關資訊，請洽詢本公司代理經銷商或上述營業部門。

© 2011-2016 YASKAWA ELECTRIC CORPORATION

資料編號 YTWMNCO-15004B

Published in Taiwan 2016年 8月  
16-1-12