Panasonic

FPWIN Pro 導入

研修講義【上】

第１章～第８章

台灣松下環境方案股份有限公司

地址：台北市中山北路二段 44 號 15 樓　電話：+886 2 2581 6020

目次

第 7 章　使用變數撰寫程式

第 8 章　製作 Function(FUN)/Function Block(FB)

第 9 章　用 Fuction diagram 製作程式

第 10 章　用 Instruction list 製作程式

第 11 章　用 Structured Text 製作程式

第 12 章　用 Sequential function chart 製作程式

第 13 章　監控功能

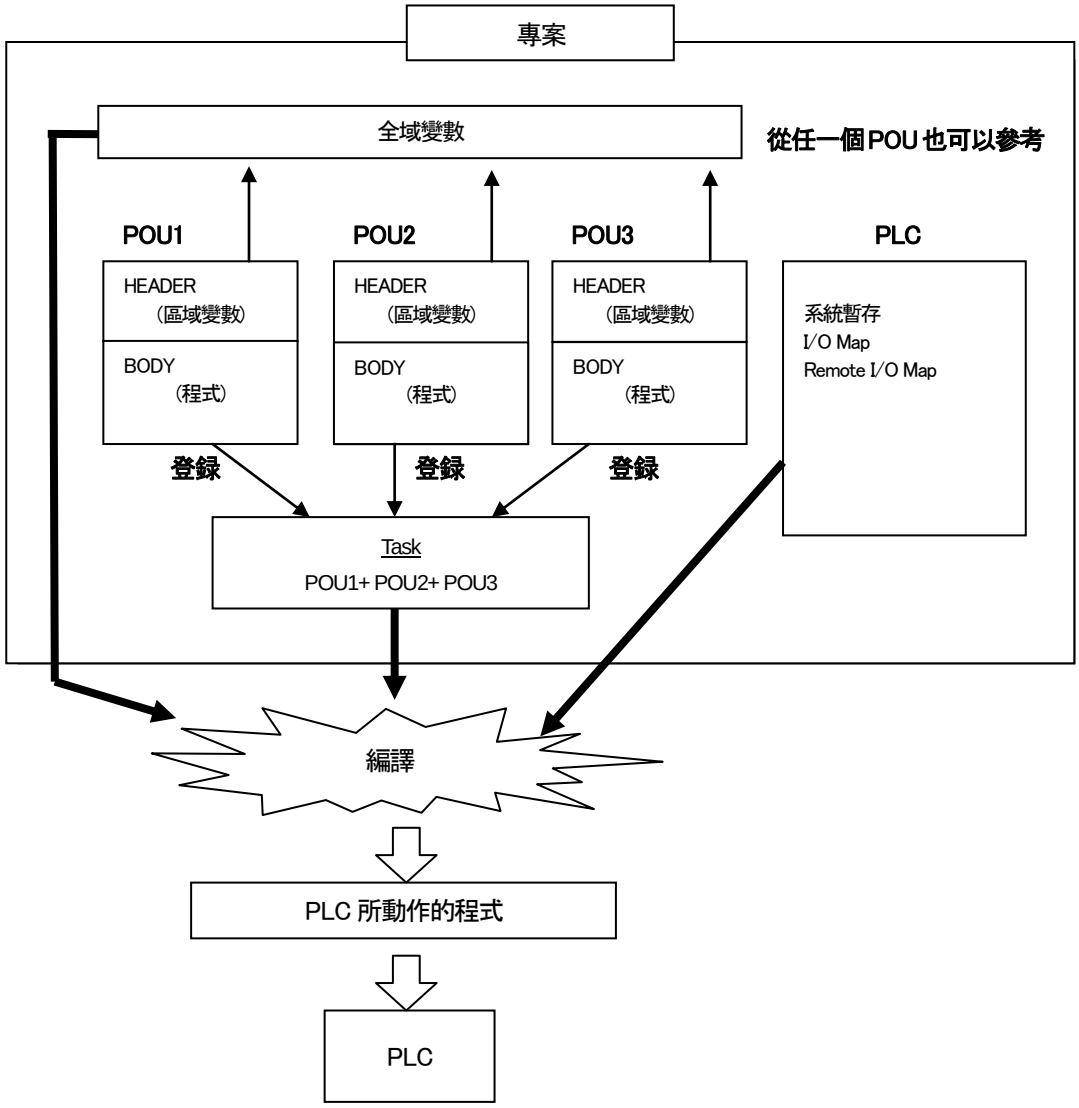
第 14 章　User library

第 15 章　安全性

第1章

FPWIN Pro 的概要

1-1 FPWIN Pro 的概念



■ 可將多個程式組合起來構成全體的專案程式。

- FPWIN Pro 跟 FPWIN GR 有所不同的是、可以將幾個程式組合起來成為全體的專案程式、其中一個程式稱為 POU、從後述的五種語言可以選擇其中語言來做成。
- POU 因為登録於 Task、會被認為全體程式之一。

■ 可以使用變數編寫程式。

- 取代直接指定(使用 X0、Y1、DT100 等)、以變數進行編寫程式。
- 只有在 POU 內、有效的變數稱為區域變數、任一個的 POU 的參考變數稱為全域變數。
- 內部的工作區域、使用繼電器及暫存器時、使用區域變數、Compile 時、會自動分配適當的繼電器及暫存器。在其他的程式中、想再利用 POU 時、可以指定直接設備、使用區域變數方便編寫程式。
- 外部輸入(X)、外部輸出(Y)等、和物理接線有關的設備時、必須作為全域變數進行登録。

■ 將編寫好的程式作為工程文件進行管理。

- 登録幾個 POU(程式)的 Task、全域變數、系統暫存器、I/O 分配等、所有關於程式的全部情報都可作為專案進行管理。

■ 作成的專案必須像高階語言一樣的進行編譯。

- 作成的專案需要經過編譯轉換成PLC 能解讀的程式。編譯前的程式(階梯圖等)想儲存到PLC 內時、PLC 需要有通用記憶體以及註解記憶體。
- 因此 FP0 及 FP1 等無法儲存程式碼。必須儲存到電腦進行管理。

■ 符合 IEC61131-3 規格。(後述)

- 對應五種程式語言(LD, FBD,SFC, ST, IL)。

■ 利用匯入功能、可以充分活用以往開發過的程式。

- 即使是用 FPWIN GR 編寫的程式、也可利用匯入功能充分活用。

1-2 關於 IEC61131-3 規格

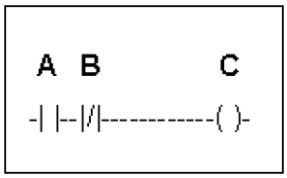
本規格是為了實現以下的各項目的、由 IEC（International Electro-technical Commission）所制定。

- 國際等級的標準化程式
- 無須依賴 PLC 機種的程式
- 構造化編輯程式、容易理解維護的程式
- 再利用性高、模組化的程式
- 使用嚴密文法檢查、錯誤性的程式

■ IEC61131-3 規格中、提供以下 5 種程式語言。

・LD（階梯圖程式）

以往的階梯圖程式。



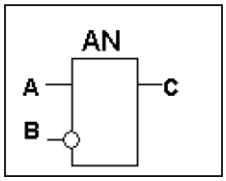
・IL（指令集）

類似組合語言的低階語言。

```
LD    A
ANDN  B
ST    C
```

・FBD（Function Block Diagram）

區塊之間用配線方式類似電路圖之表記方法。



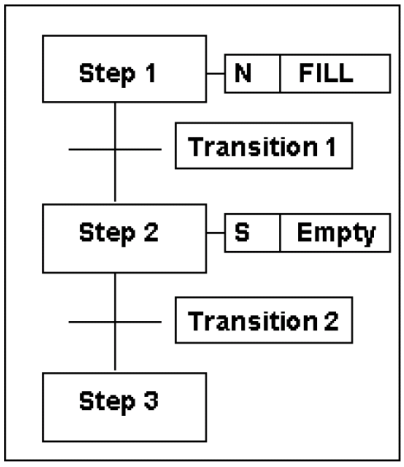
・ST（Structured Text）

針對數值演算等的高階語言。
寫法類似 PASCAL。

```
FOR i:=0 TO 100 DO
  SUM:=SUM+1;
END_FOR;
```

・SFC（Sequential function chart）

程式的順序用圖式來呈現的圖形化語言。
各 STEP 和 Transition 內部可以用上述的 LD、IL、ST の各語言進行記述。



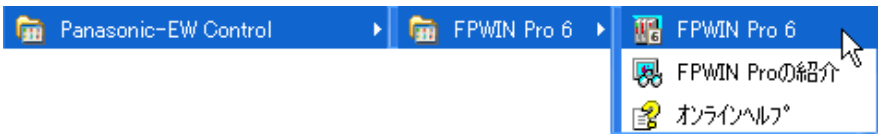
第2章

FPWIN Pro 的起動

2-1 FPWIN Pro 的起動

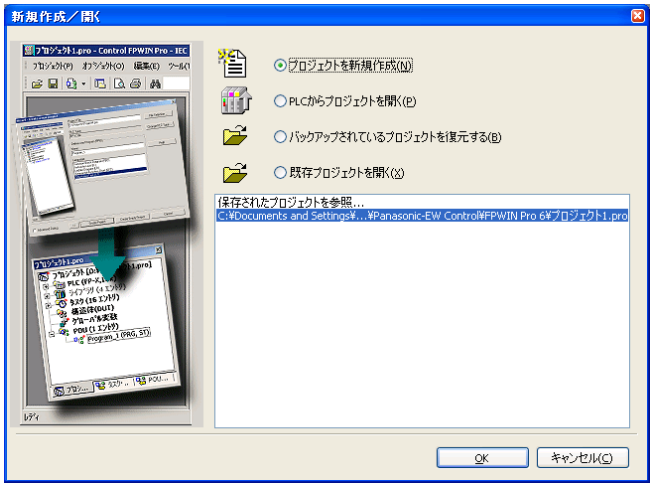
起動的順序、請依照以下任一種方式、起動 FPWIN Pro。

・從 Windows 的開始選單起動。



・點擊桌面上的快捷圖示 2 次即可以起動。

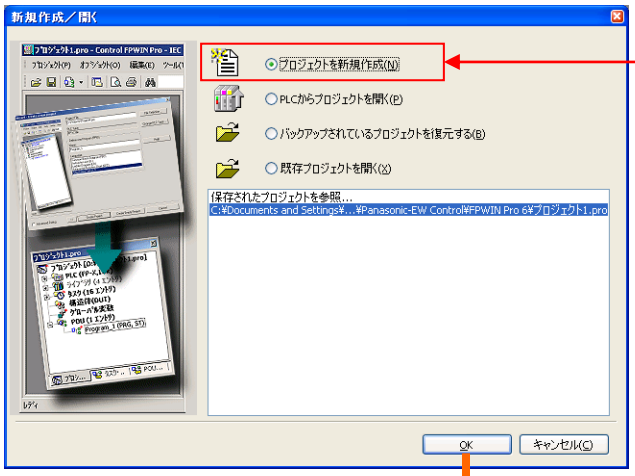
起動後、會顯示以下對話框的畫面。



在此可以選擇建立新 Project、打開舊 Project、或是從 PLC 上傳程式。

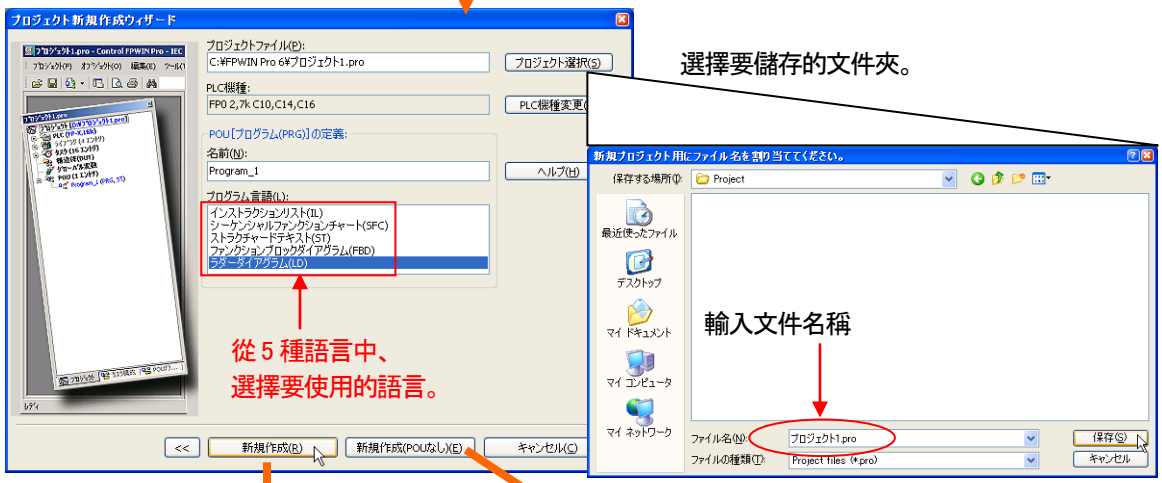
■ 新 Project 作成

製作新 Project、可從「程式」「功能」「FunctionBlock」製成等。此外、無須作成 POU(程式構成単位)、亦可單獨製作 Project。



選擇「新 Project 作成」。

按下「OK」按鍵。

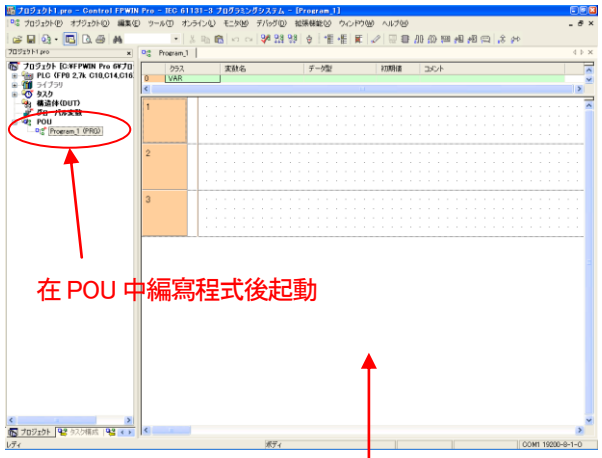


選擇要儲存的文件夾。

從 5 種語言中、
選擇要使用的語言。

輸入文件名稱

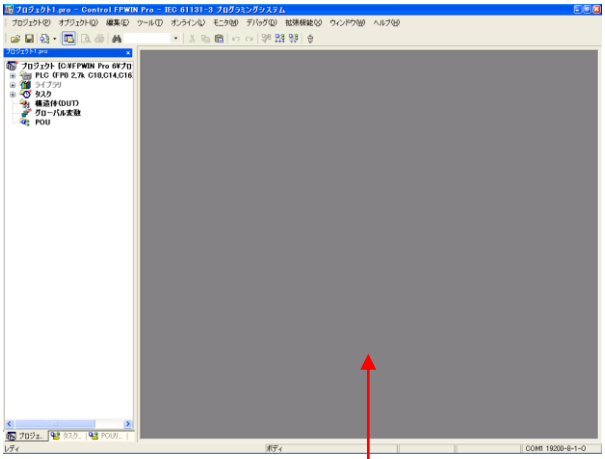
附有 POU 的 Project



在 POU 中編寫程式後起動

上圖 LD 語言時的起動例。

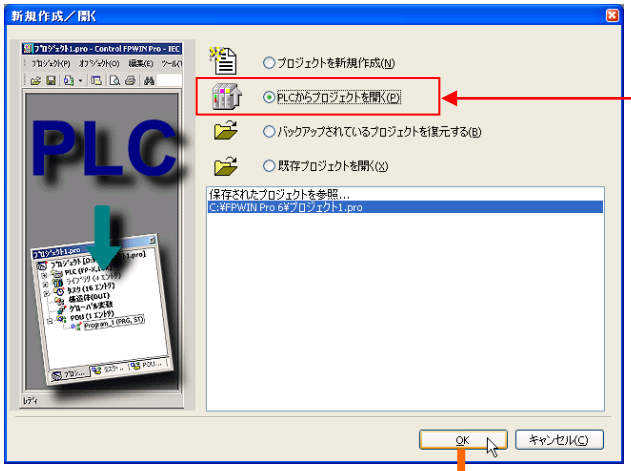
只建立 Project(不在 POU 編寫程式)



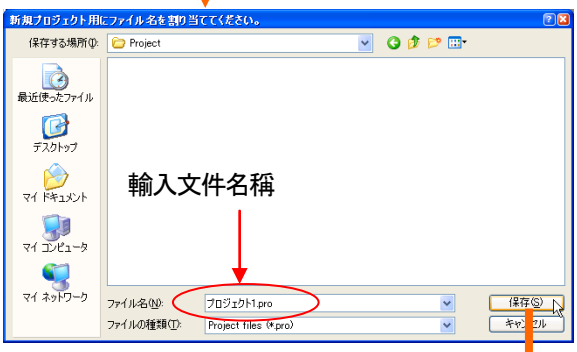
之後在 POU 編寫程式的話會如左圖相同。

■ 從 PLC 開啓 Project

從 PLC 讀出 Project 時選擇此項目。
從 PLC 讀出時、必須已經用 FPWIN Pro 事先將文件儲存到 PLC 之中。

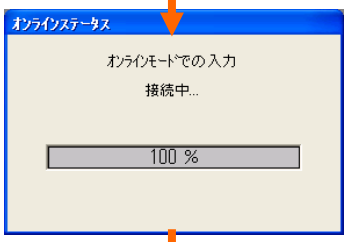


選擇從「PLC 開啓 Project」。

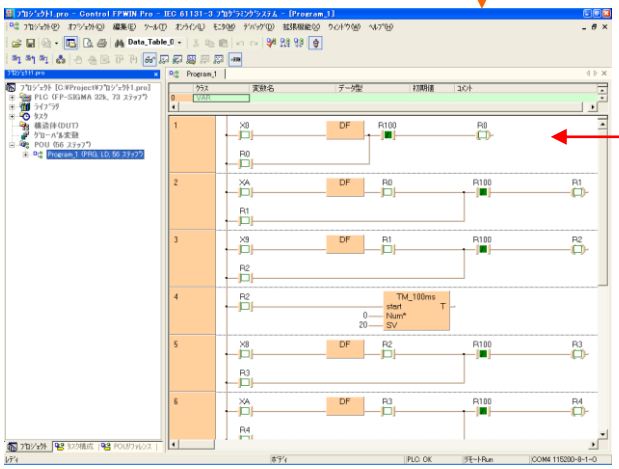


利用跳出的對話框選擇欲讀取的檔案。

按下「儲存」按鍵。



和 PLC 連接。

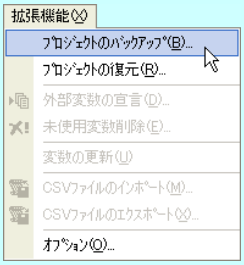


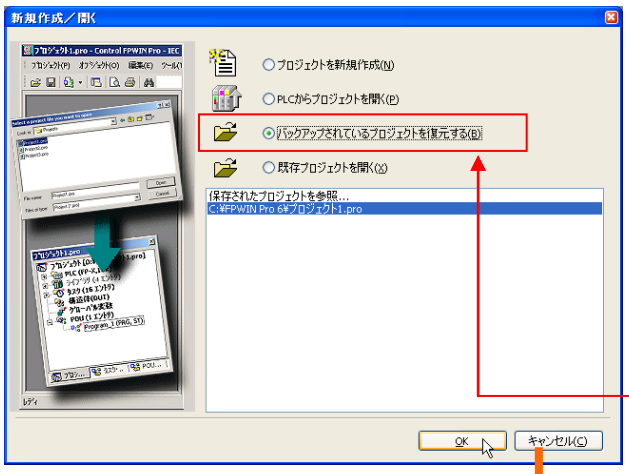
此 Project 為 LD 寫法的例子。

■ 從 Project 復原程式

在壓縮文件打開時選擇。

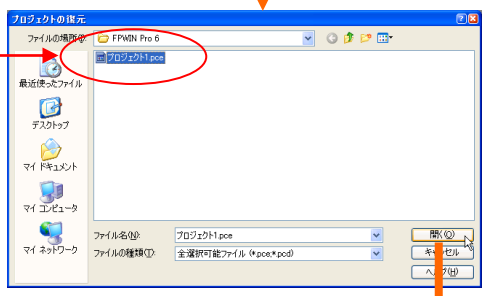
開啓壓縮 Project 時、
必須已使用「Project 備份」功能後才能儲存。





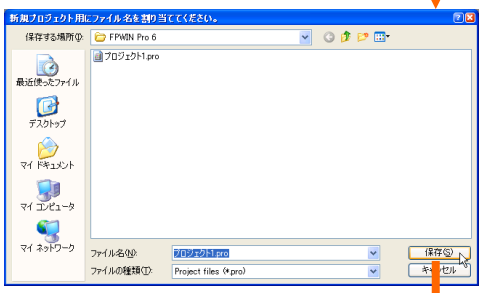
選擇「恢復已經備份文件」。

選擇已備份的 Project



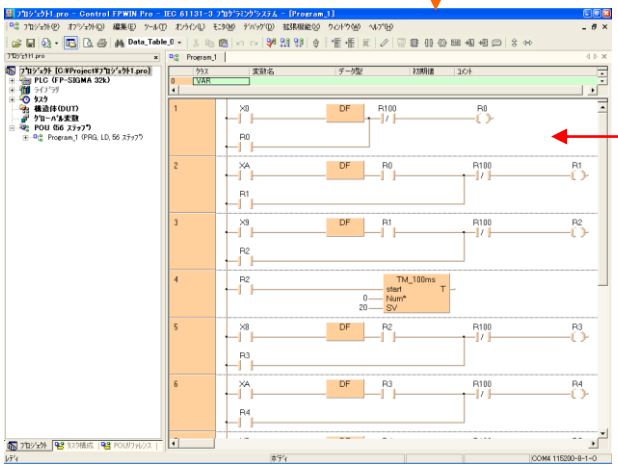
顯示選擇作為恢復對象的備份文件對話框。

按下「開啓」按鍵。



打開提示恢復的文件的保存目標的對話框。

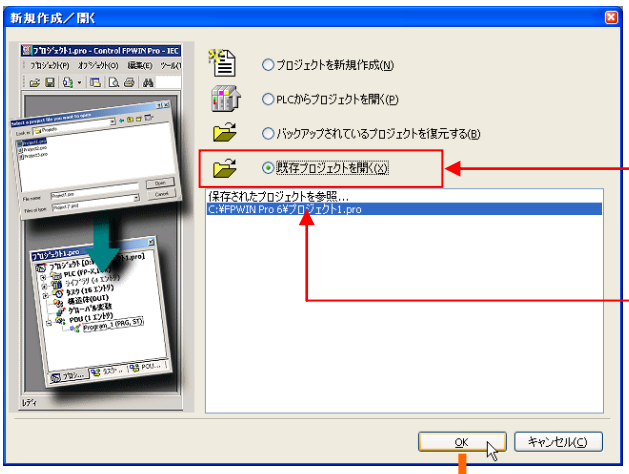
按下「儲存」按鍵。



在 LD 語言下的例子。

■ 打開既有的 Project

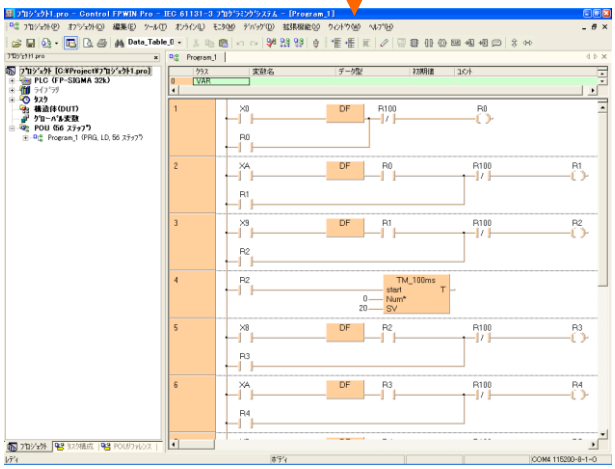
選擇電腦中既有的 Project。



選擇「開啓既有 Project」。

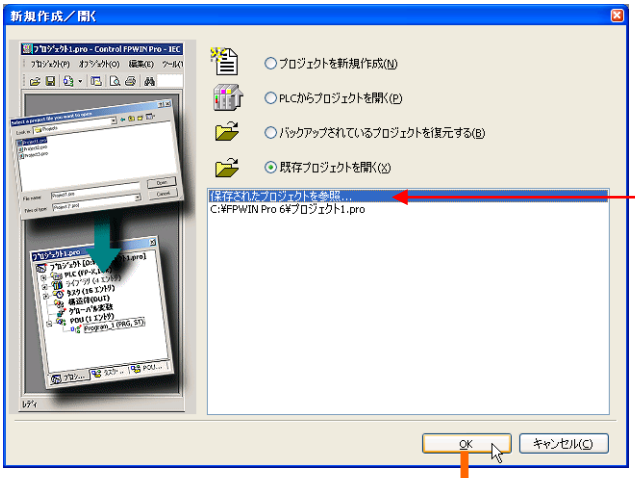
選擇目的 Project

按下「OK」按鍵。



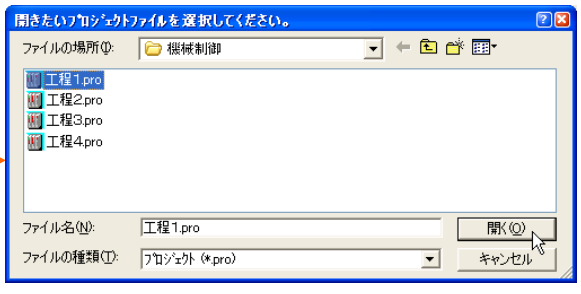
在 LD 語言下的例子。

要開啓 Project 表內沒有記載的 Project 時、請點選「參考儲存 Project」兩下。



選擇「參考儲存 Project」

請選擇目標 Project。

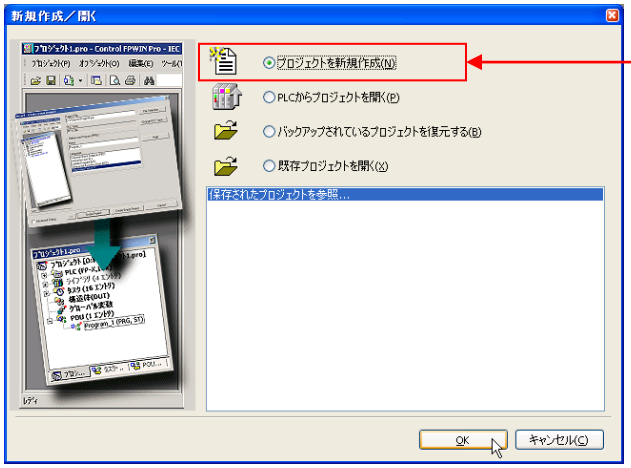


按下「OK」按鍵。

2-2 關於起動程式語言

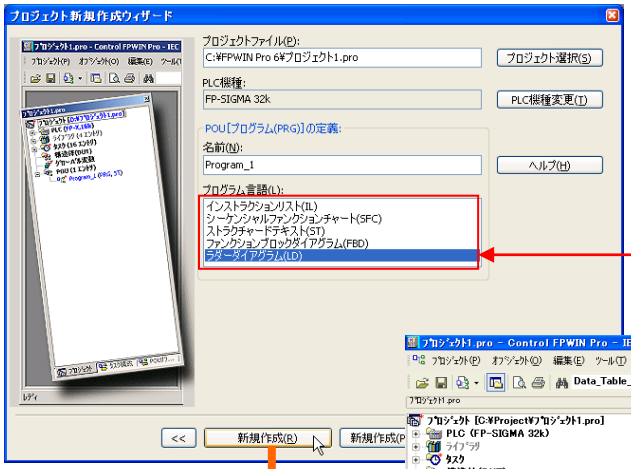
可以選擇 FPWIN Pro 的 5 種語言(LD、FBD、ST、IL、SFC)進行程式編輯。
在此、個別介紹製作新 Project 時、選擇 5 種語言所顯示的內容。

根據不同的語言、工具列和程式組體的顯示會有改變。

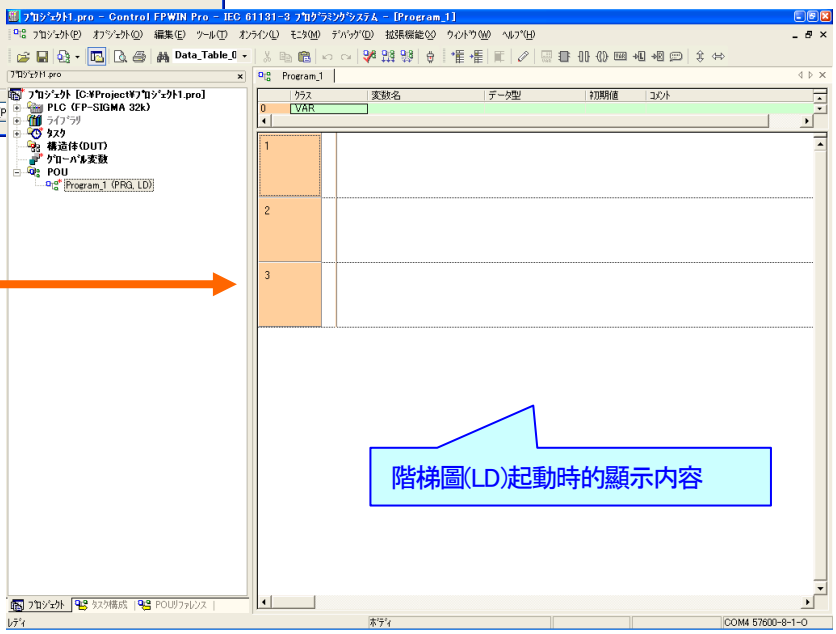


選擇「製作新 Project」。

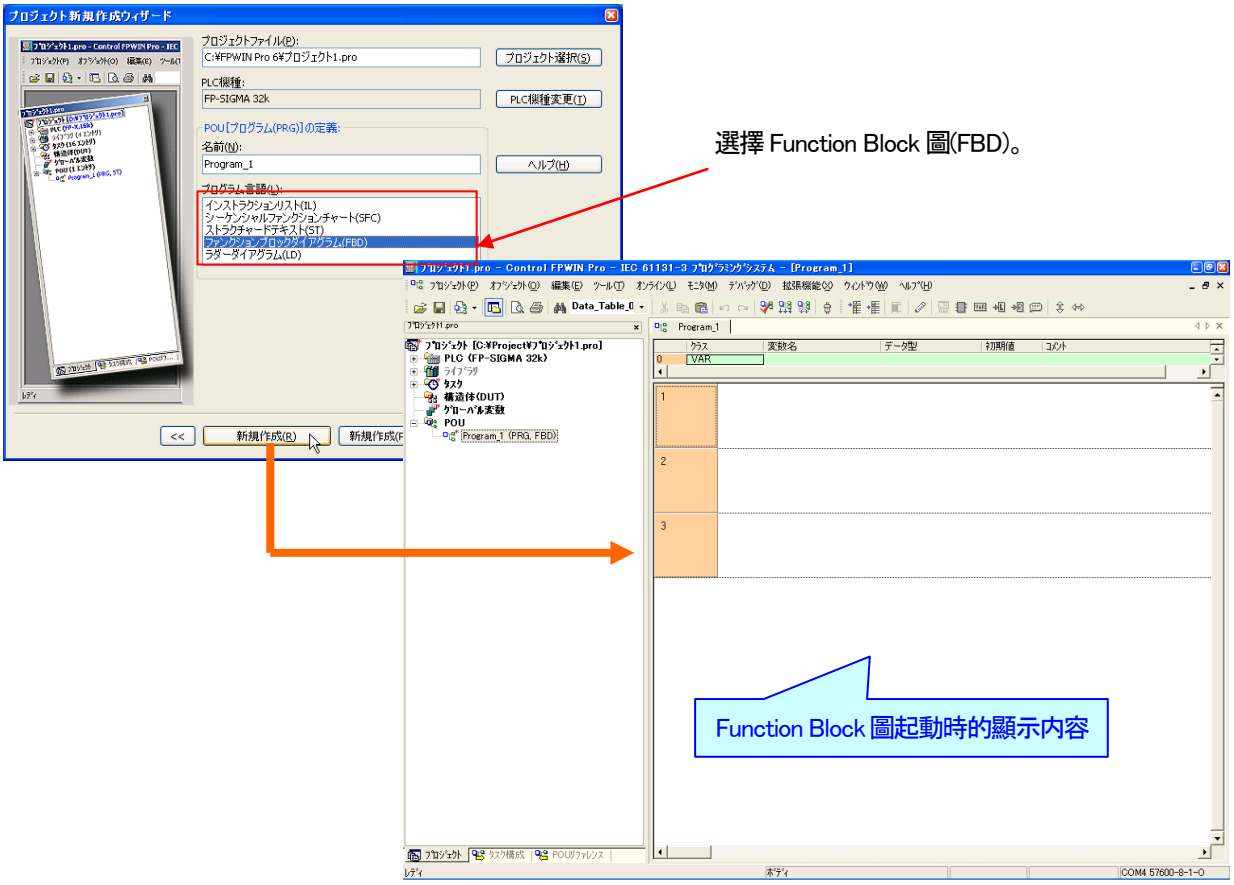
■ 階梯圖(LD)



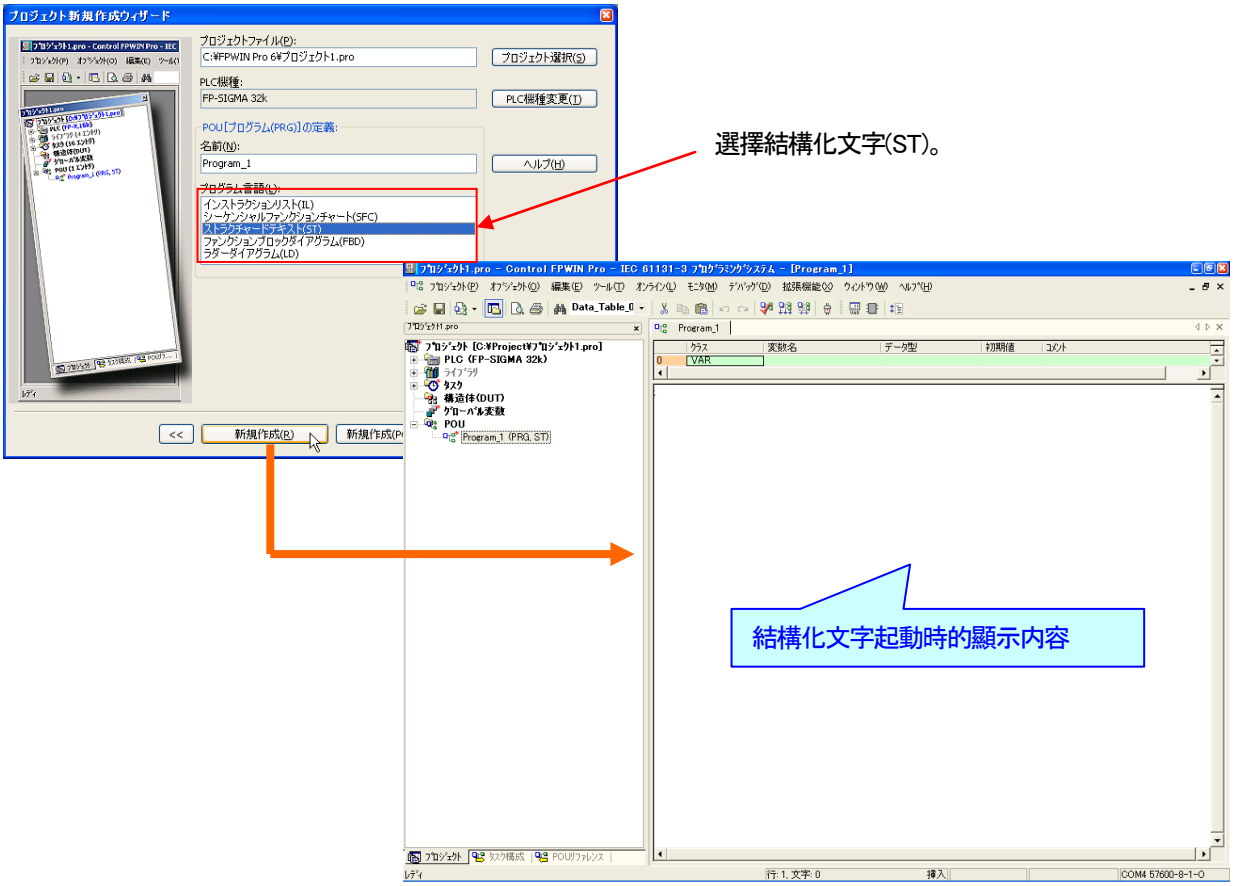
選擇階梯圖(LD)。



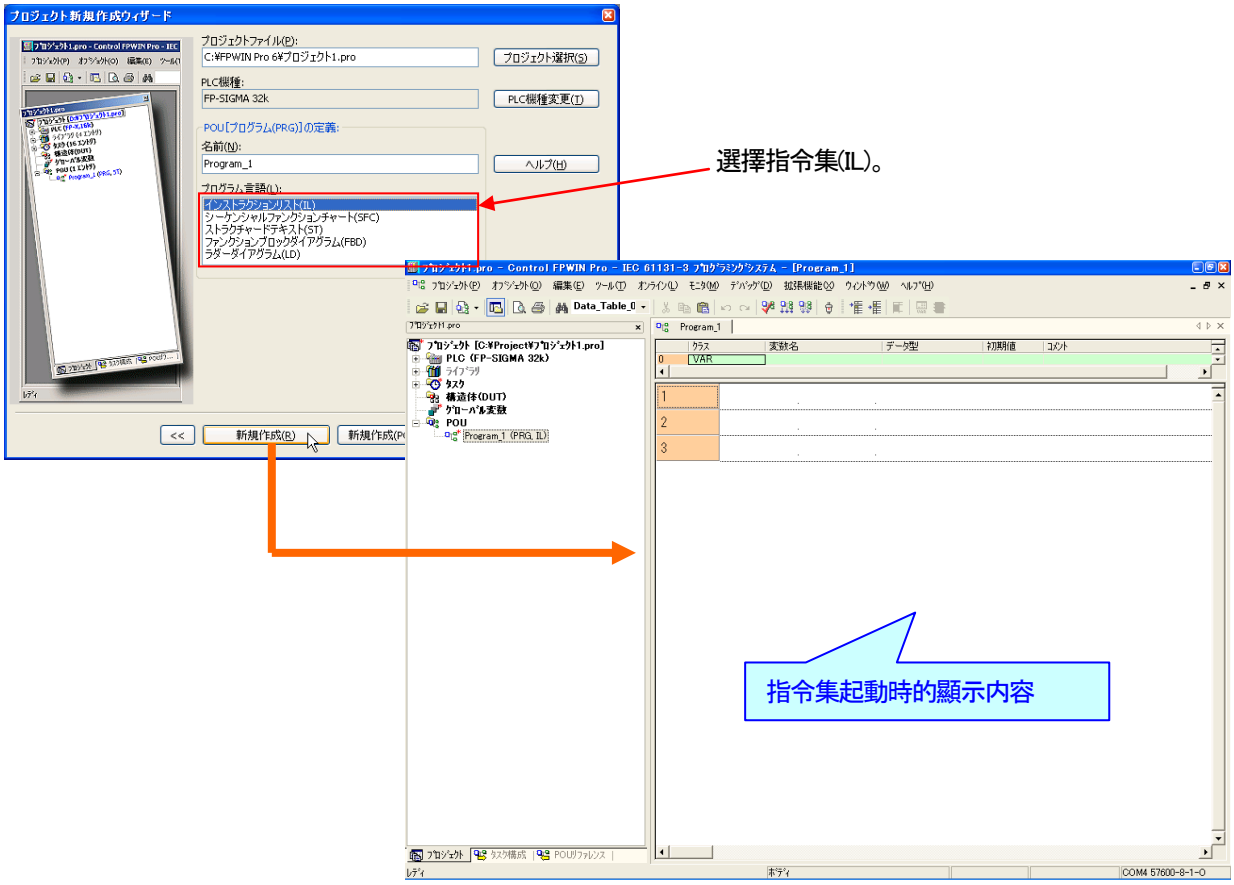
■ Function Block 圖(FBD)



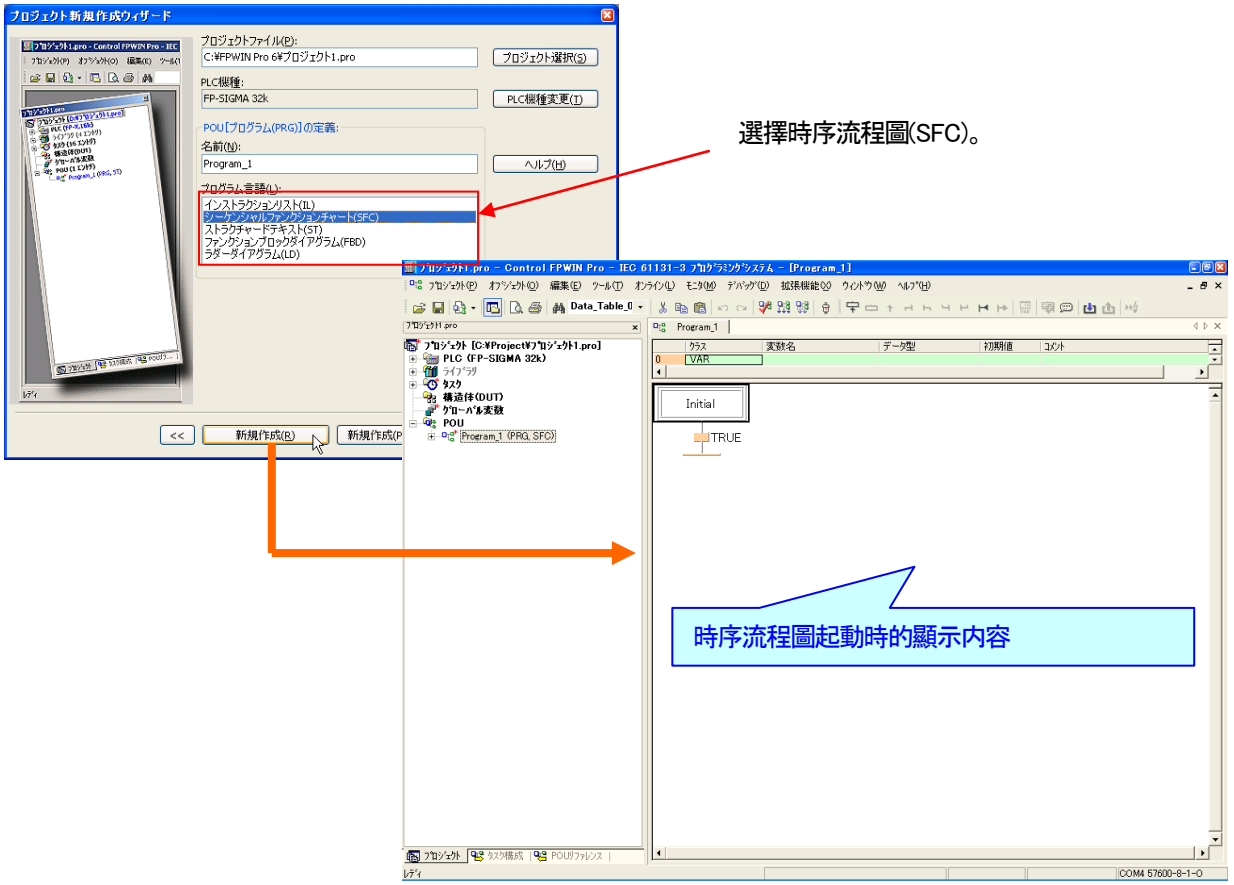
■ 結構化文字(ST)



■ 指令集(IL)



■ 時序流程圖 SFC)

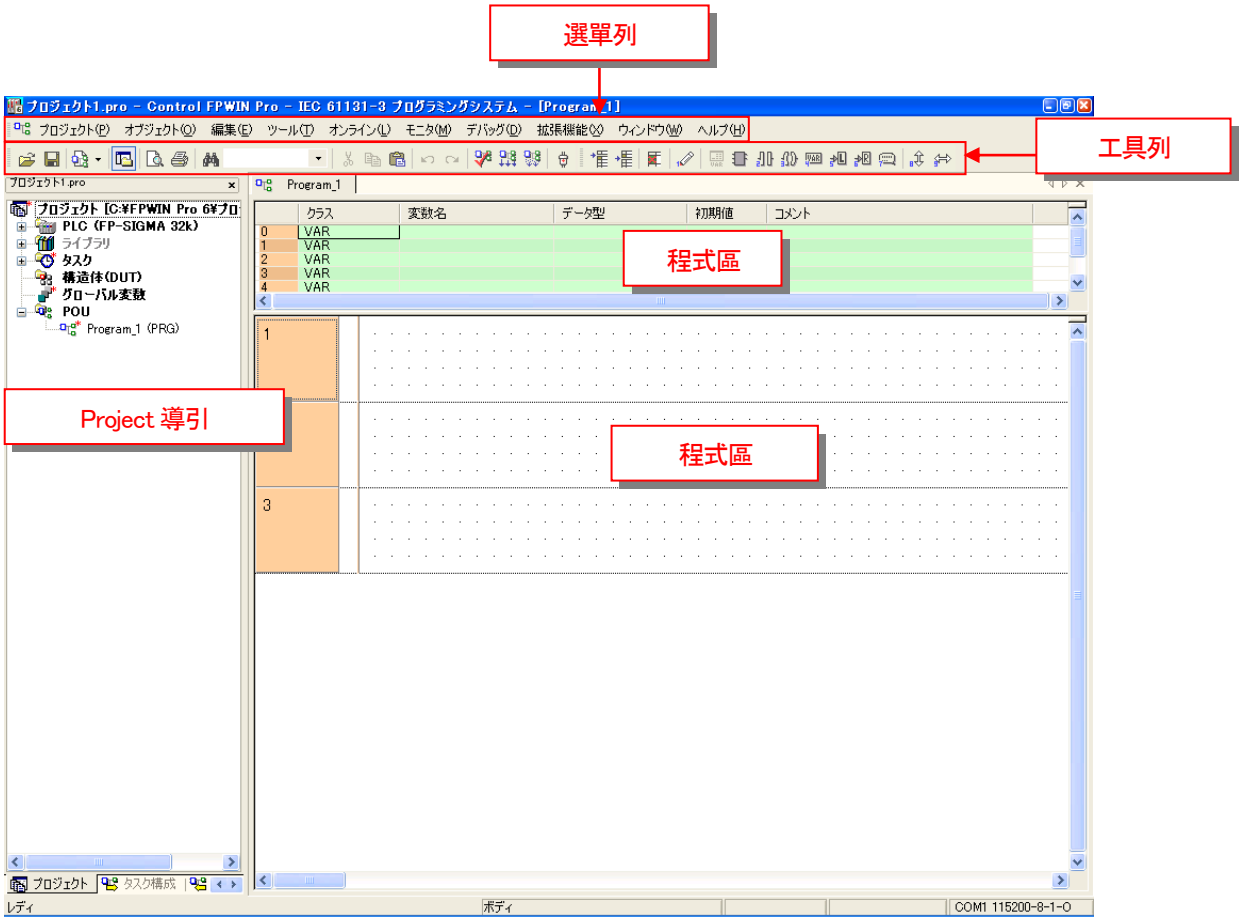


第3章

關於 Project

3-1 Project 的概念

在 FPWIN Pro、並非只是單單對程式進行管理、而是包含全部的相關資訊作為管理。
Project 是由、系統暫存器、可以使用的指令一覽(Library)、對程式的執行順序進行管理的Task、對已定義變數進行管理的資料表、實際的動作程式等所構成的。
這些構成及資訊在 FPWIN Pro 的「Project 導引」中顯示。

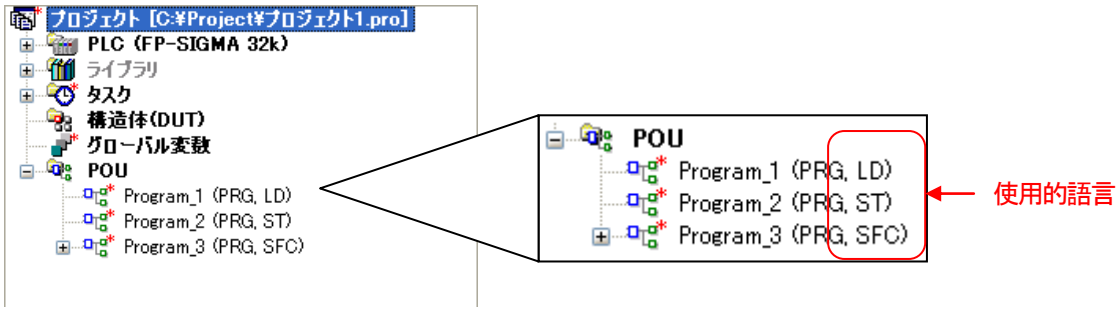


3-2 Project 操縱的構成

Project 導引是由以下元素所構成。

3-2-1 POU

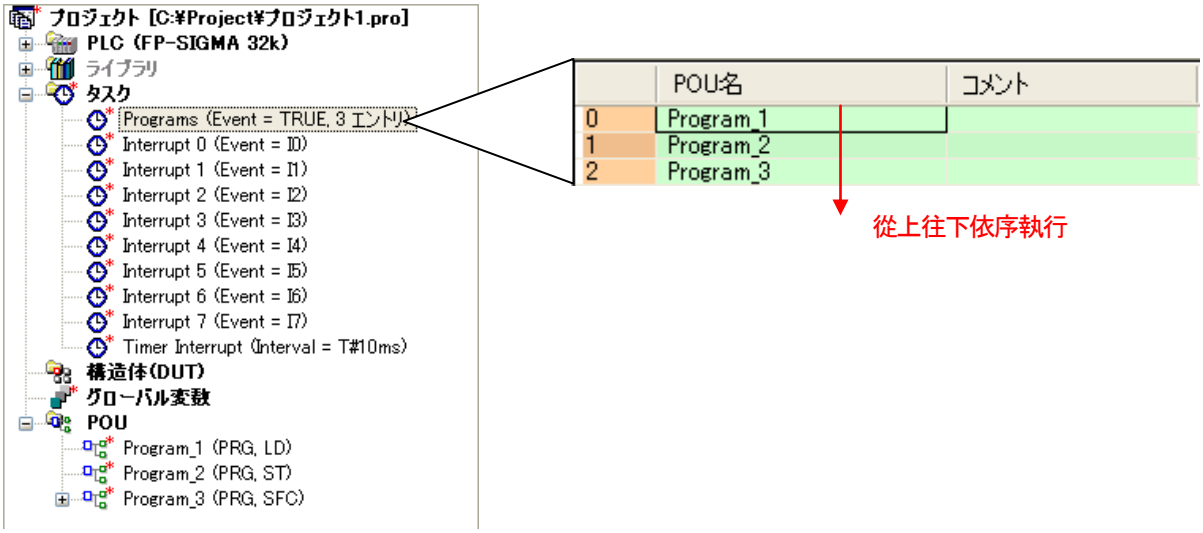
POU 是 Program Organization Unit 的簡稱、是程式的構成單位。
在此建立已完成的程式。



■POU 的概念
用 FPWIN Pro 所編寫的程式會全部由 POU 來管理。
新編寫的程式在 FPWIN Pro 稱為「新 POU」。
編寫新 POU 時、必須選擇名稱為 POU 型(Prg、Fun、FB)和程式語言(LD、FBD、SFC、ST、IL)。
程式編寫途中、無法變更其他語言。
能編寫數個 POU、並且可將全部的 POU 合併為一個程式、或是只選擇其中需要的 POU 合併。

3-2-2 Task

Task 建立各個 POU(PRG 型:POU)的執行順序。
在本教材裡、選出 Task 最重要的「Programs」做為解說。
在 Task 或是其他地方、Interrupt(中斷程式)等可以建立到 Task。
編寫好的 POU 建立到 Task、能使該 POU 成為變換對象。
編寫完成的 POU 若沒建立到 Task 裡的話、
該 POU 則無法新編輯完成的 POU 會自動建立到 Program Task 裡。
POU 是按照在 Task 中所建立的順序來執行的、但是這順序也可以在事後進行更改。

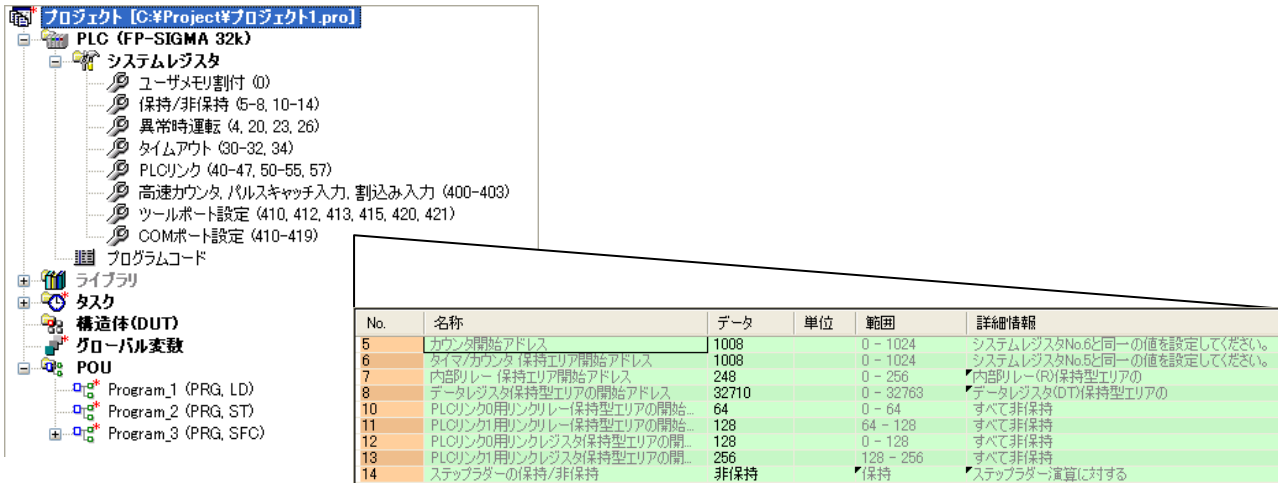


3-2-3 PLC

PLC 動作時的各種環境設定。

■系統暫存器

錯誤發生時的動作選擇和通信埠的設定、內部記憶體的保持區設定等、進行 PLC 動作用的各種設定。
系統暫存器會因 PLC 機種不同而設定內容有所差異。



3-2-4 資料結構(DUT)

外部機器的送收信資料、位置控制用的數據資料等將資料結合成資料組去使用。
在 FPWIN Pro 中、這種可以將資料組作為變數群組預先定義的方式稱為「資料結構」。

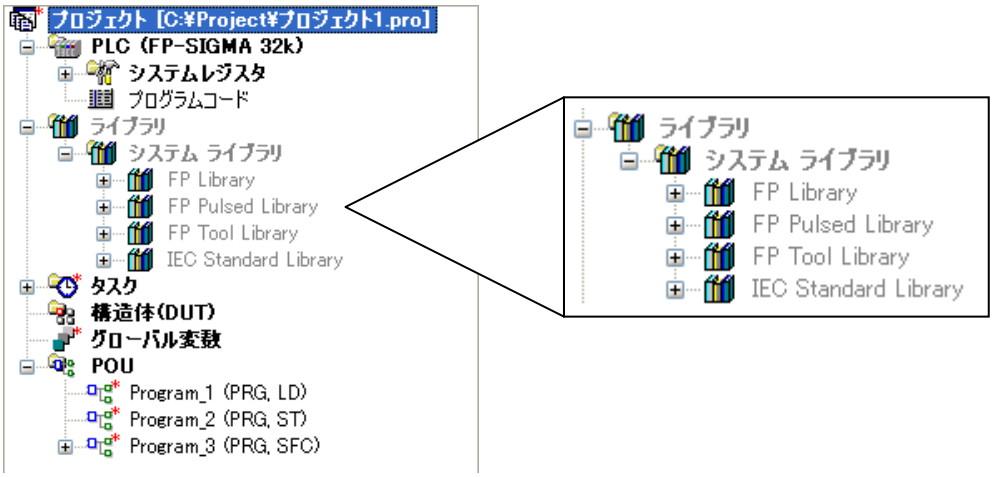
3-2-5 Library

所謂 Library、在 Project 內可以使用的應用指令一覽。
在 FPWIN Pro 中、除了支援以往的 FP 系列 PLC 的應用指令、也支援 IEC 等指令。

會因為 PLC 機型不同、可以使用的指令也不同、Library 的顯示也會有差異。
此外、也可以登録使用者所編寫完的 Library。
Library 有以下幾種型態。

FP Library: 登録有 FP 系列支援的應用指令。
FP Pulsed Library: 登録有 FP 系列應用指令中的微分執行型指令。
FP Tool Library: 在 FP 系列的應用指令中、有些需要在操作中指定位址或大小(字數)。
在 FP Tool Library 中、包含有在 FPWIN Pro 上能改變變數的函數。
IEC Standard Lib: IEC 標準定義的指令。

C



●備註

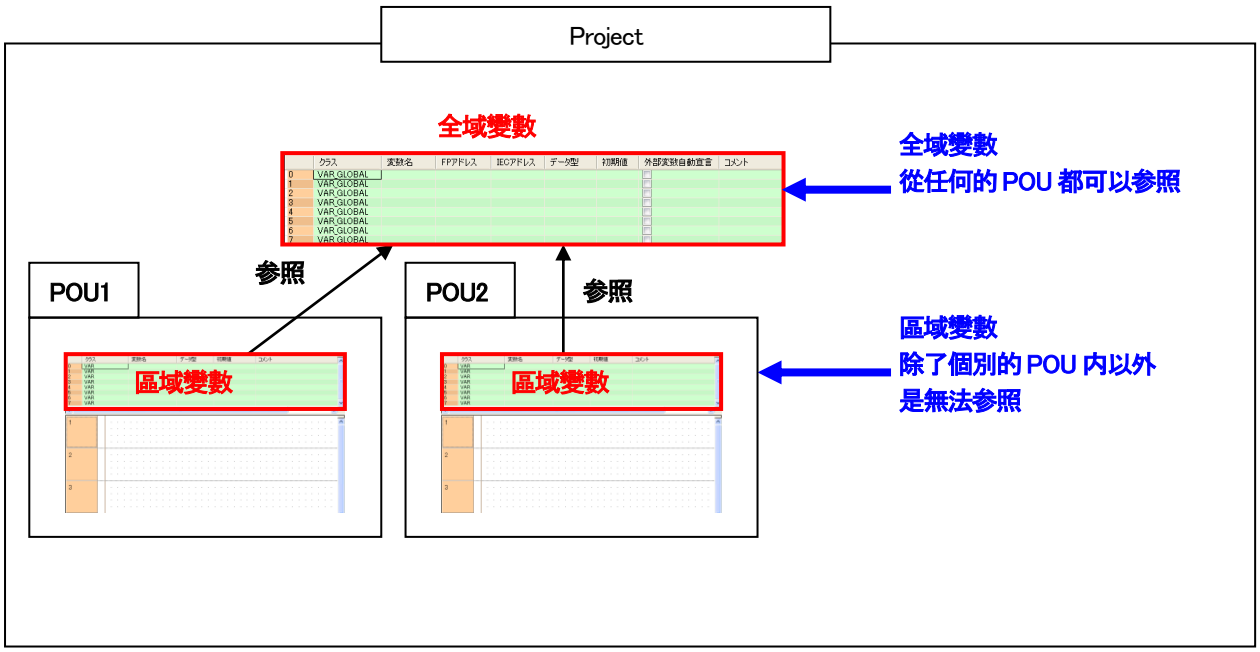
IEC 指令是根據 IEC 標準 61131-3 所定義的指令。在 FPWIN Pro 中、可轉換成 PLC 本體能使用的指令、
下載後可以使用這些指令。
相對的、IEC 指令本身並不直接支援。

3-3 全域變數和區域變數

變數是 PLC 內的輸出入(X、Y)和記憶區(WR、DT 等)能任意加上名字更換的東西。
變數可以自由宣告(登録)、並使用變數來編寫程式。

3-3-1 全域變數和區域變數的差異

全域變數是、從任何的 POU 都可以參照的變數。
在 Project 導引中、在 POU 的上位位置。用 Project 導引的「全域變數」來宣告(登録)。
PLC 的位址可以指定分配。
區域變數是、只能在其 POU 內、參照的變數。
在各個 POU 的 Header 進行宣告(登録)。PLC 的位址無法指定分配。



3-3-2 全域變數的定義

全域變數需要設定以下項目。

■必須設定項目

・Class: 通常的變數、數值保持的變數、數值固定的變數等。
(變數的 Class)
・變數名稱: 在程式內所使用的名稱。 變數名不可以用數字為開始。
・型態: 宣告變數的型態。(BOOL、INT、WORD 等)
・初始值: 設定變數的初始值。雖然會自動的設定、不過也可以變更。

■任意設定項目

FP 位址: 分配對應 PLC 設備區塊時設定。(EX:DT10、R10 等)
如果沒有設定時、FPWIN Pro 會自動分配。
IEC 位址: 程式內會自動設定位址。
使用者無須輸入。
外部變數自動登録: 選取此項之後、所設定的全域變數會各自自動登録到 POU 的 Header。
備註: 可以將備註寫入到變數。程式中不會顯示。

3-3-3 區域變數的定義

使用區域變數時需設定以下項目。

■必須設定項目

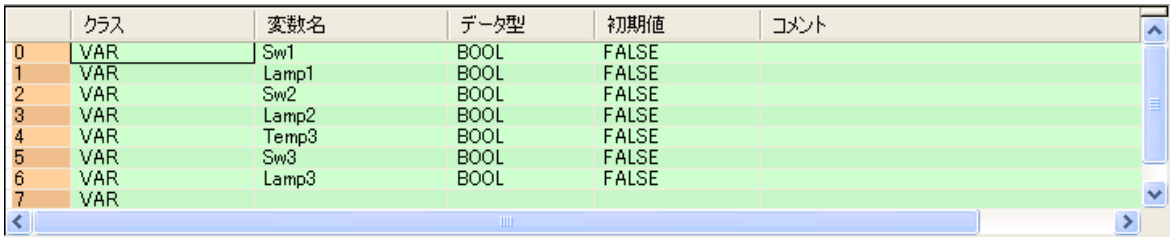
・Class: 一般型的變數、保持型的變數、數值型的變數等。
(變數的 Class)
・變數名稱: 在程式內所使用的名稱。 變數名不可以用數字為開始。
・型態: 宣告變數的型態。(BOOL、INT、WORD 等)
・初始值: 設定變數的初始值。雖然會自動的設定、不過也可以變更。

■任意設定項目

備註: 可以將備註寫入到變數。程式中不會顯示。

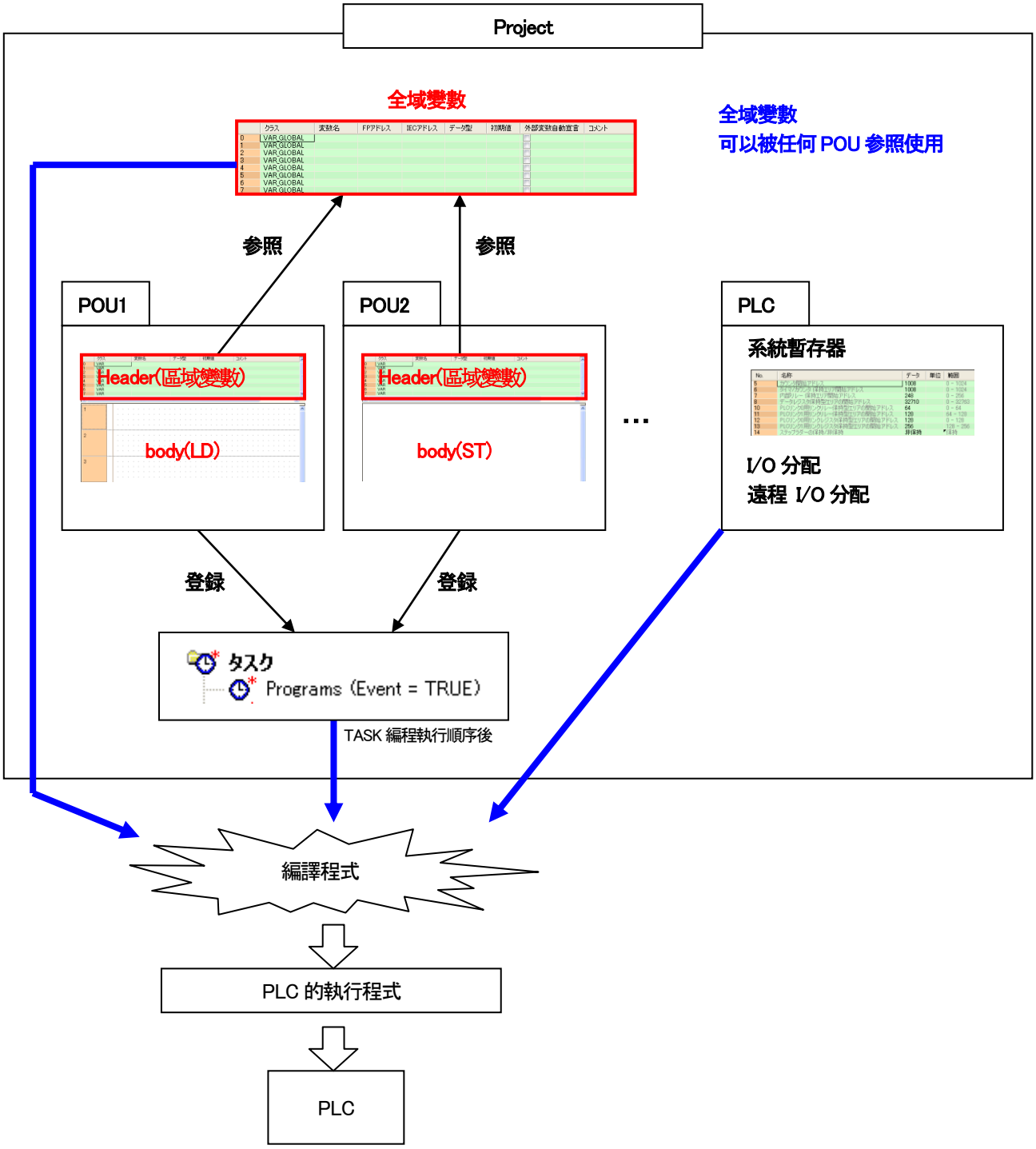
●注意事項

・在 POU 的 Header 中、必須對該 POU 內所使用的變數全部進行登録。
作為全域變數所宣告(登録)的變數、也要宣告(登録)到 POU 的 Header。
這種情況下、表示作為「全域變數」已經完成宣告(登録)。
以「外部變數(VAR_EXTERNAL)」登録到「POU Header」。
以「全域變數」、所登録的變數可以自動以「外部變數」登録到全部的 POU 的 Header 之中。
・在區域變數無法分配 PLC 位址。
如果需要預先分配 PLC 位址、請將其作為全域變數進行宣告(登録)。



	クラス	変数名	データ型	初期値	コメント
0	VAR	Sw1	BOOL	FALSE	
1	VAR	Lamp1	BOOL	FALSE	
2	VAR	Sw2	BOOL	FALSE	
3	VAR	Lamp2	BOOL	FALSE	
4	VAR	Temp3	BOOL	FALSE	
5	VAR	Sw3	BOOL	FALSE	
6	VAR	Lamp3	BOOL	FALSE	
7	VAR				

■ Project 的構成



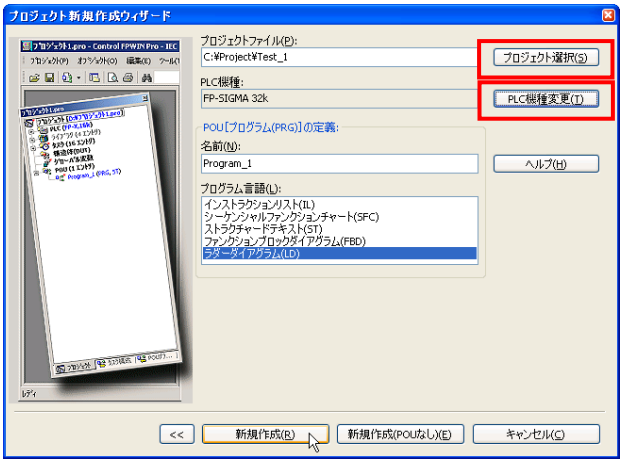
・PLC 的執行程式在 POU 的 body 部分中記述。
・在程式中如果使用變數、請參照定義變數的一覽。
・編寫好的 POU 登録到 Task、進行編譯(程式轉換)。
・全編譯時、程式和 PLC(系統暫存器)的資料會一起轉換、成為傳送到 PLC 的資料。

第4章

用階梯圖來製作程式

4-1 製作新的 Project

選擇製作新 Project 會顯示以下畫面。



請使用導引工具、設定以下內容。

■製作 Project 的位置和 Project 名稱

在「Project」的區域內、輸入 Project 名稱。

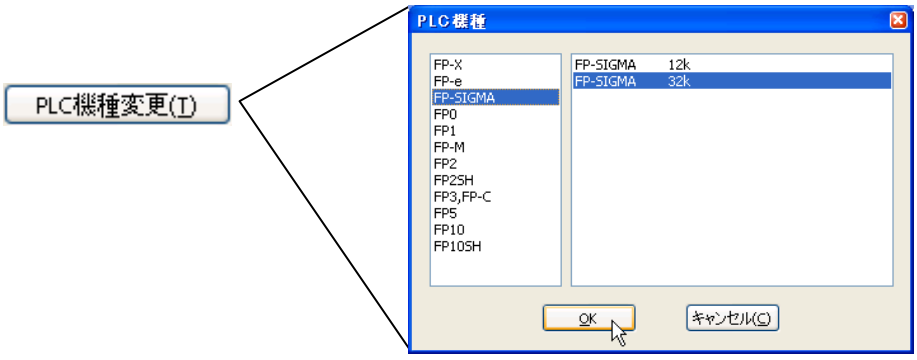
本次的例子之中、Driver : C 內稱為「Project」的文件夾已經事先建立、在其中設定「Test_1」的文件名。

■選擇所使用的 PLC 機種

「PLC 機種」的區域中、選擇所要使用的 PLC 機種。

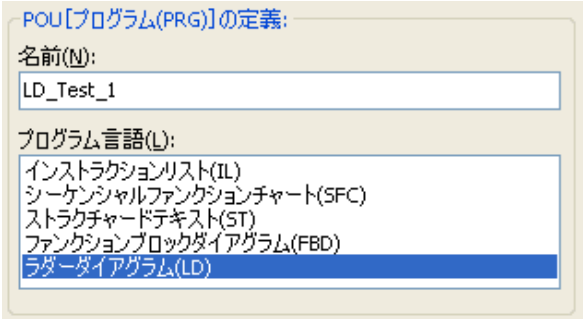
需要變更目前顯示的 PLC 機種時、按下「PLC 機種變更」按鍵後會顯示以下對話框、請在此變更機種。

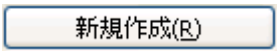
此外、在標寫程式的過程中也可以變更 PLC 機種。



■程式的定義(在 POU 所登録的名稱和使用的語言)

「程式(PRG)的定義」區域中、設定文件中所登録的 POU 名和使用的語言。
此處以「LD_Test_1」作為 POU 名稱、編輯語言選擇階梯圖程式(LD)。



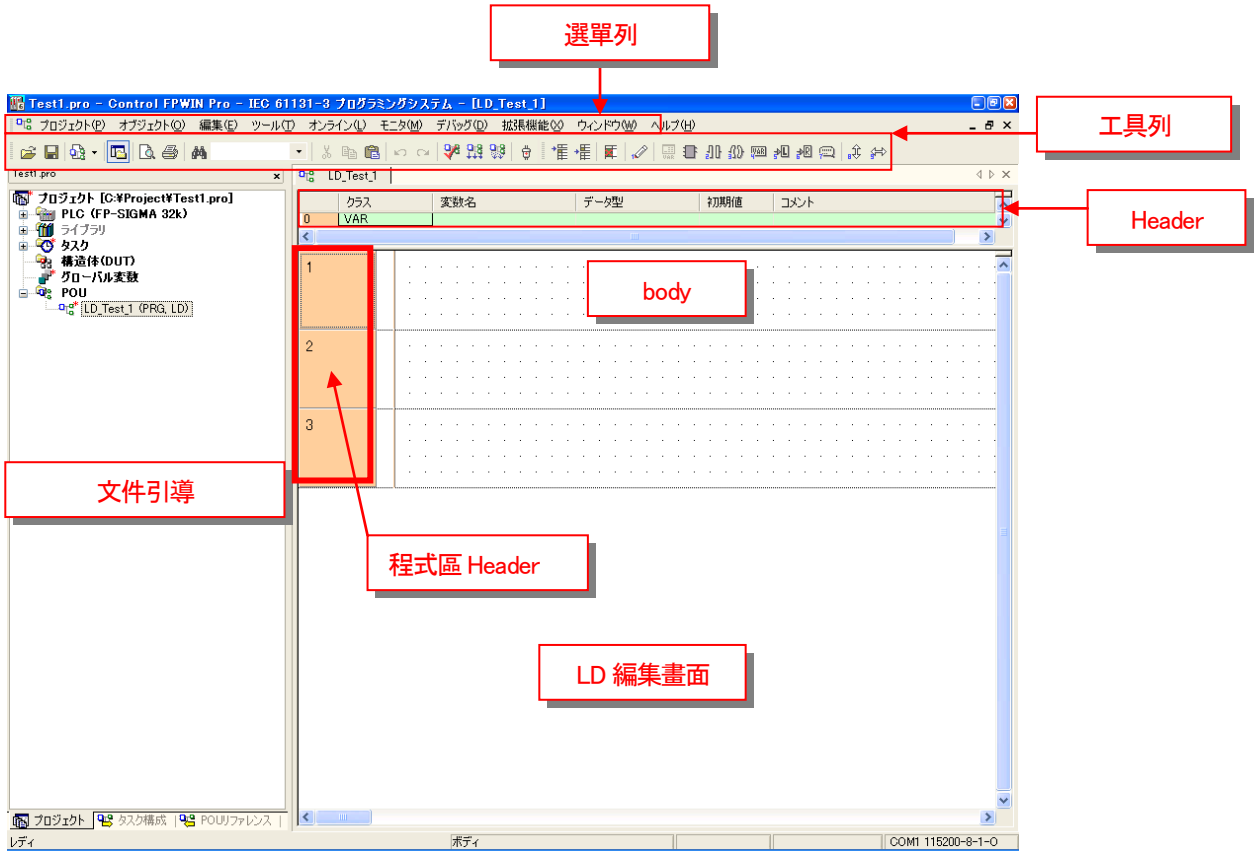
全部設定完成後、請按下  按鍵。

至此、準備已經結束。

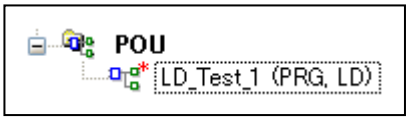
4-2 FPWIN Pro 的基本畫面

Project 新建之後、會顯示以下畫面。

■FPWIN Pro 起動時的畫面



請在此點選文件引導的「POU」中的「+」標記。
會有如以下的樹狀圖。



最初設定的 POU 名「LD_Test_1」會顯示出現。

4-3 圖繪階梯圖

4-3-1 LD 編集用工具列

為了編輯 LD(階梯圖程式)、需要使用以下的工具列中的圖標。

■繪圖用圖示


用於繪製接點和應用指令的連線。


使用此圖示選擇配置應用指令和 IEC 指令等。


配置接點(基本命令 ST、AN、OR 等)。


配置線圈(輸出指令 OT)。


配置輸出入變數應用指令的操作數)。


在編集畫面加入備註。

■編集補助功能用圖標

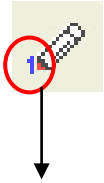

現在動作中的 Network 前面、插入新的 Network。


現在動作中的 Network 後面、加入新的 Network。


縱方向任意擴大。


橫方向任意擴大。

■備考


代替圖示、可以從鍵盤將圖示上面的數字輸入也能進行同樣的操作。

此時的操作為按下鍵盤“1”。

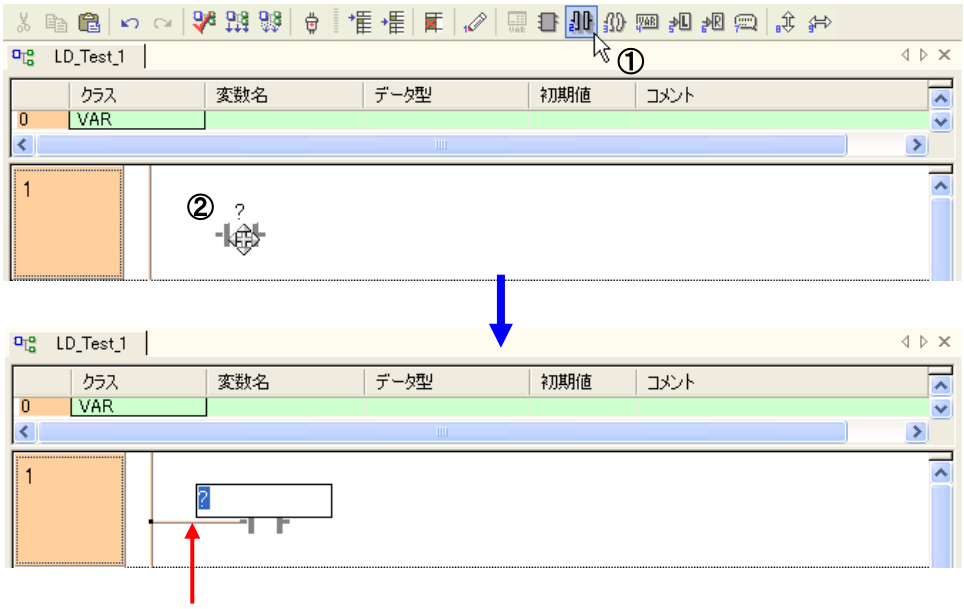
4-3-2 元件的配置

在 LD 編集中、點擊工具列上的元件(接點和線圈等)、在編集畫面上的任意位置再次按下滑鼠左鍵、就可以在該位置放置元件。
和「拖曳」所不同的是、點擊後無需要繼續按住滑鼠的按鍵。

畫出接點看看。
以下有 2 種方可以進行描繪。

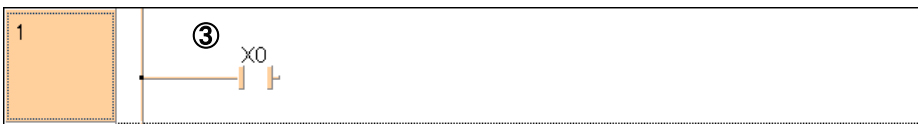
■操作順序

①點擊工具列上的接點 。
②在滑鼠的指標附有選擇好的元件、在 LD 編集畫面上的任意位置(想將元件放入的地方上)再次點擊。



會自動的和階梯圖母線進行連結。

③接點被配置的元件名會成為等待輸入的狀態、在此用鍵盤直接輸入元件名稱。



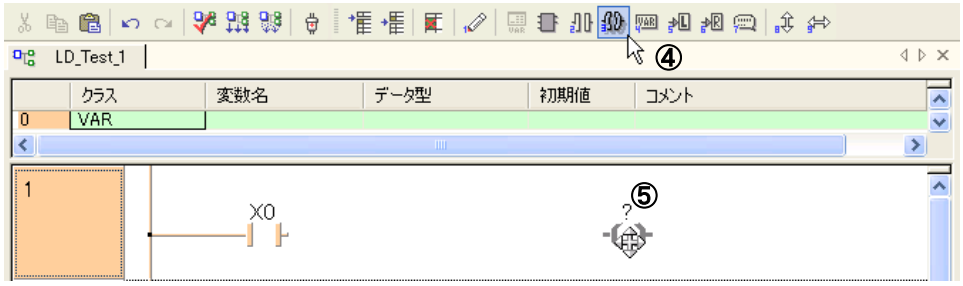
上圖為輸入“X0”的例子。

●注意事項

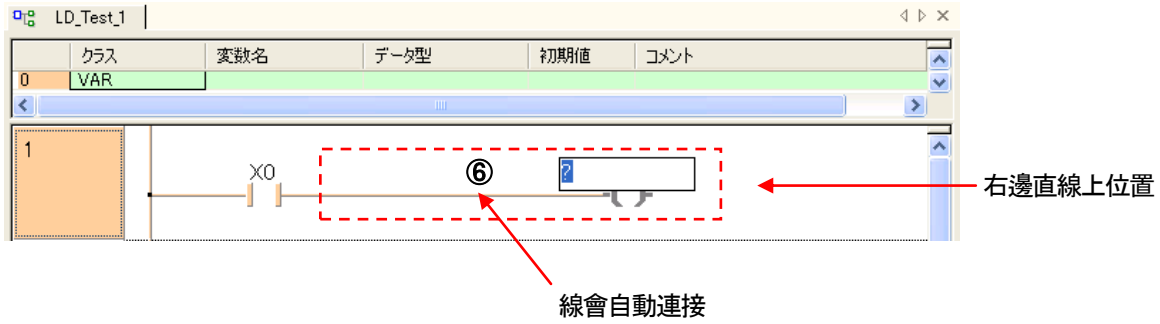
輸入名稱時英文請必須輸入半形文字。
關於元件的英文字母、不分大小寫。
例如、輸入「x0」會自動變換成「X0」。

用相同的順序畫出輸出(線圈)指令。

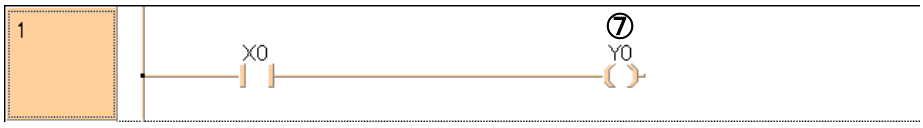
④點選工具列上的接點 。
⑤在 LD 編集畫面上的任意位置(想將元件放入的地方上)再次點擊。



⑥想連接的元件的右邊直線上配置輸出(線圈)、線會自動連接。



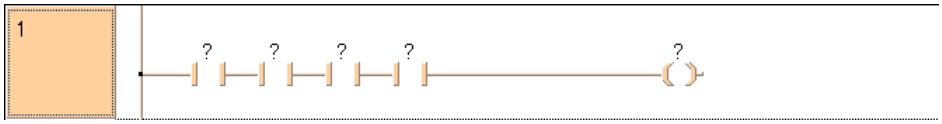
⑦輸出(線圈)所配置的元件名稱變成等待輸入狀態、在此用鍵盤直接輸入元件名稱。



上圖為輸入“Y0”的例子。

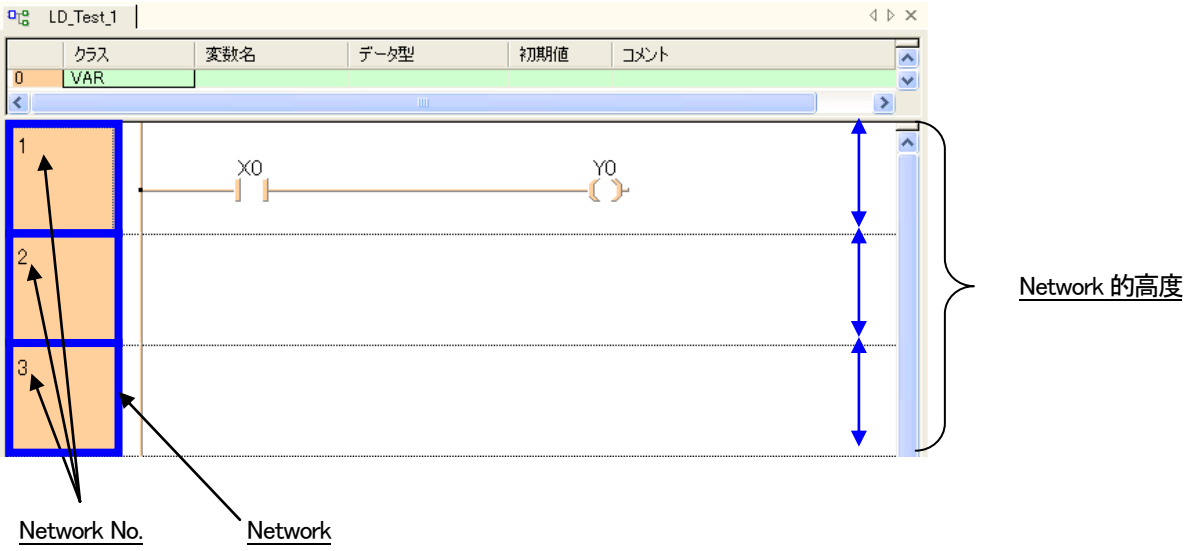
●備註

配置幾個接點、後面開始沒有輸入元件名稱也沒關係。



4-3-3 Network 區域(高度)的變更

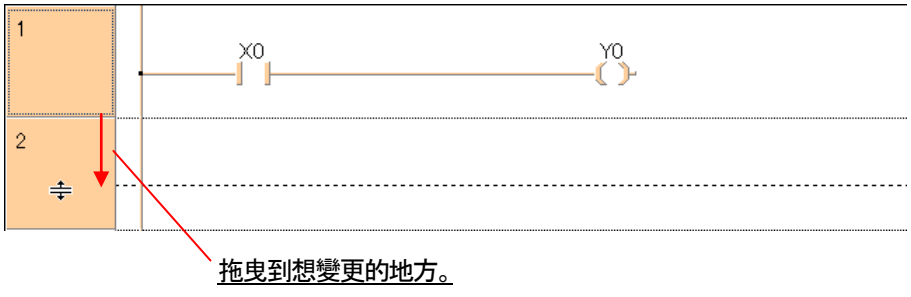
製作新的 POU 的時候、初期狀態如下圖般、所設定 Network 的高度為固定、用以下方法可以變更高度。



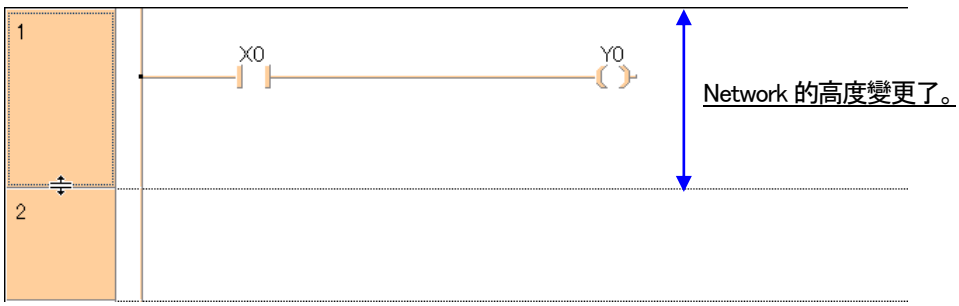
■操作順序

①變更 Network1 的高度。
滑鼠移動到以下位置、滑鼠指標的形狀會改變。

②將想變更地方拖曳出來。



③確定拖曳完成。



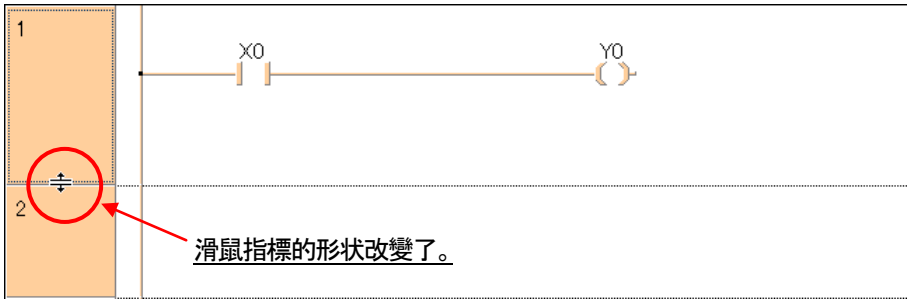
●備註

相反的、雖然可以用之前的方法把高度變窄、也有自動調整最適合的方法。

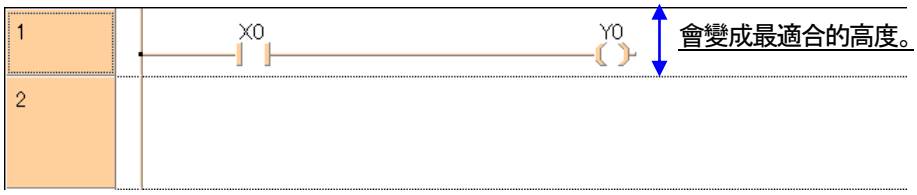
1 調整 Network 的適合高度

■操作順序

①Network1 的高度使其最適合化。
滑鼠移動到以下位置、滑鼠指標的形狀會改變。



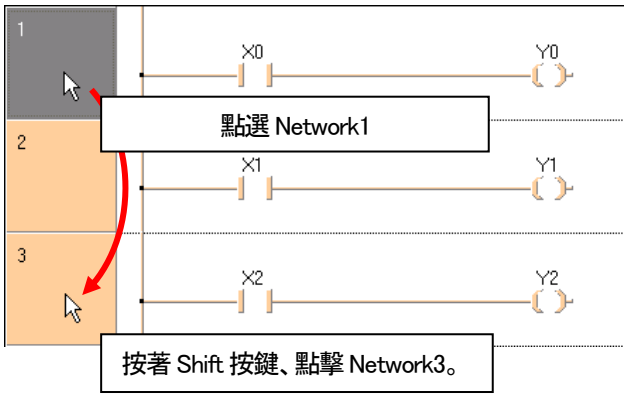
②點選左鍵兩次。
Network1 的高度使其最適合化。

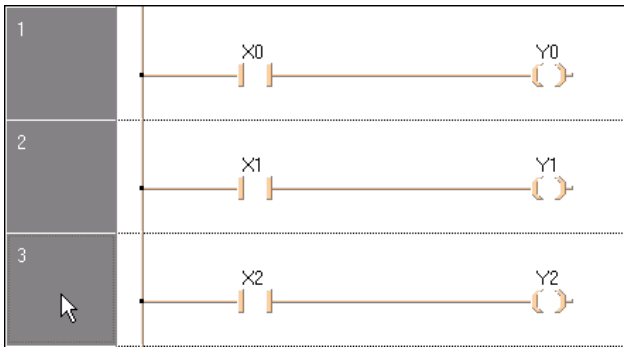


複數 Network1 的高度使其最適合化。

■操作順序

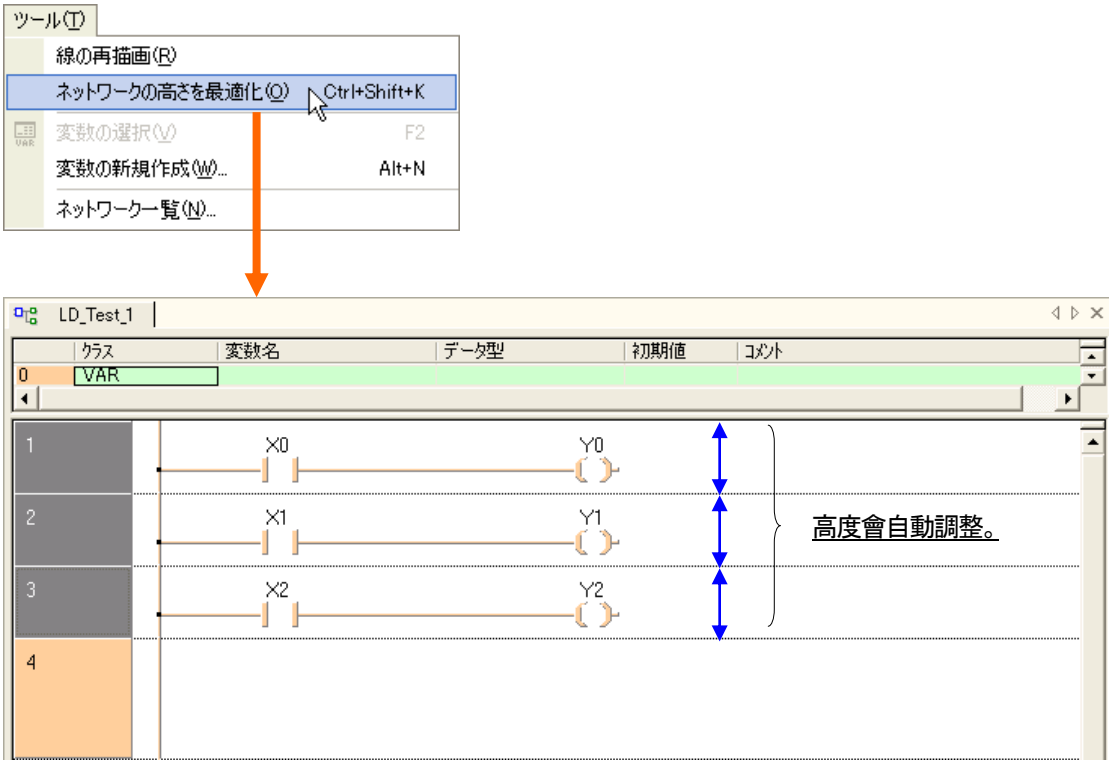
①如下圖的階梯圖選擇 Network1～Network3。





選擇到了 Network1～Network3。

②選擇**(選單)工具 → 最適合的 Network 高度**。



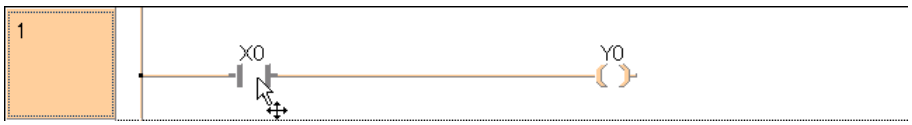
所選擇的 Network 高度會同時自動調整。

4-3-4 複製元件

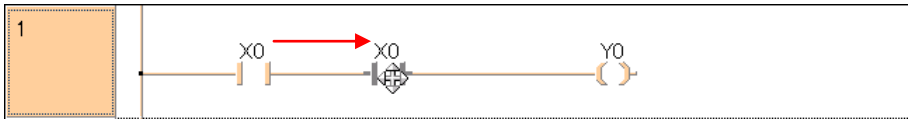
複製一個接點(元件)。

■操作順序

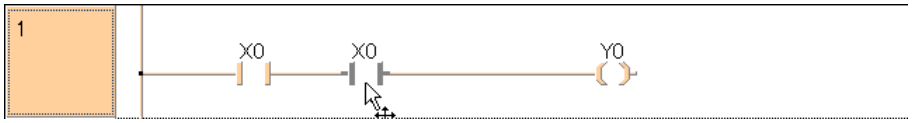
①點選想要複製的元件(接點)。



②按著 Ctrl 鍵、拖曳到目的位置。



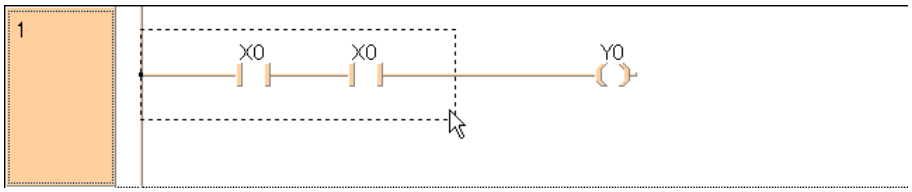
③拖曳結束就會複製。



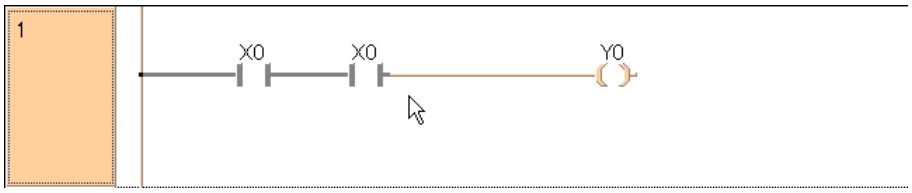
階梯回路固定就會複製。

■操作順序

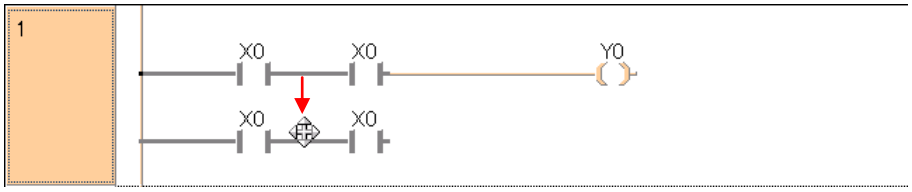
①用滑鼠圈選想複製的範圍。



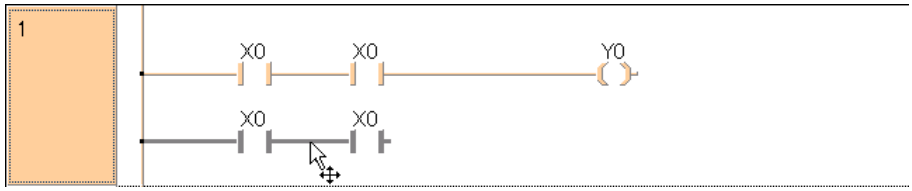
②所選擇的地方會變色。



③按著 Ctrl 鍵、拖曳到目的位置。



④拖曳結束就會複製。

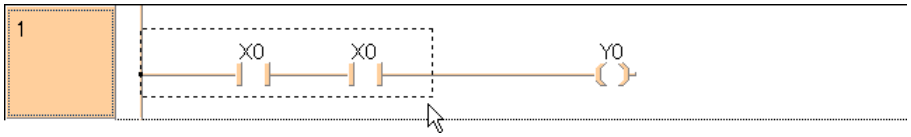


●備註

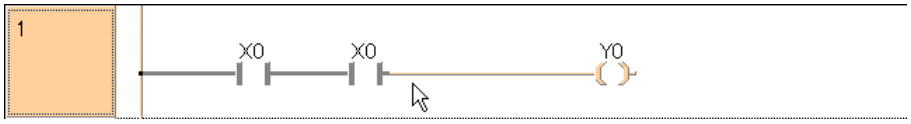
也可以跨越 Network 之間的複製。

■操作順序

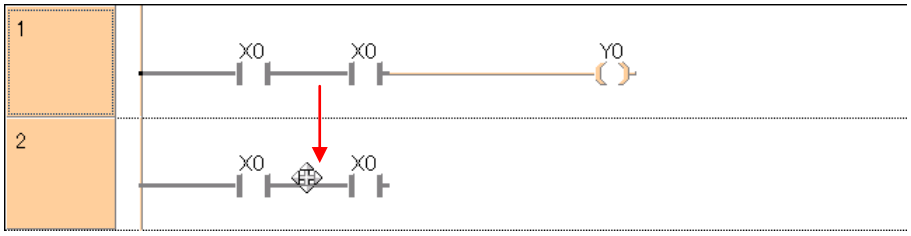
①用滑鼠圈選想複製的範圍。



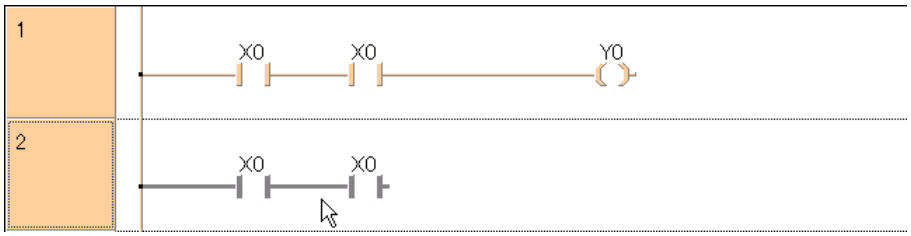
②所選擇的地方會變色。



③按著 Ctrl 鍵、拖曳到 Network2 目的位置。



④拖曳結束就會複製。

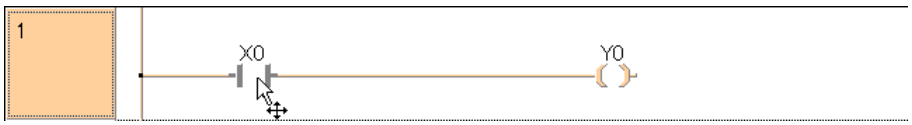


4-3-5 削除元件

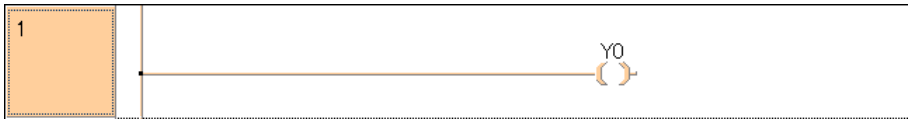
刪除一個接點(元件)。

■操作順序

①點選想刪除的元件(接點)。



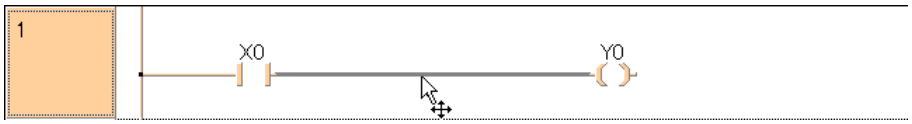
②按下 Del 鍵就會削除元件(接點)。



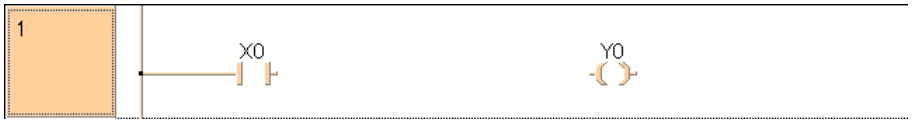
刪除接線。

■操作順序

①選擇想刪除的接線。



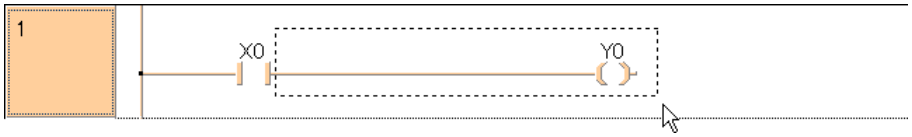
②按下 Del 鍵就會刪除所選擇的接線。



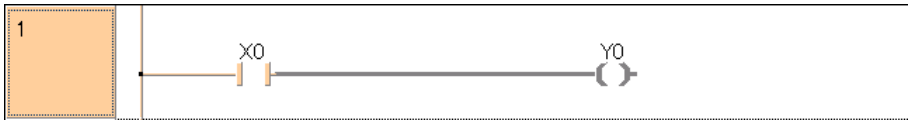
刪除固定接點(元件)。

■操作手順

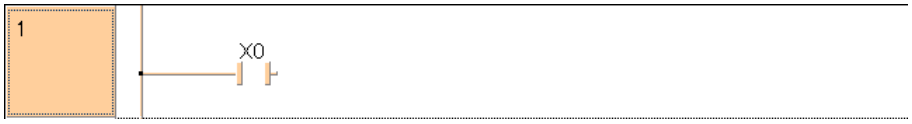
①用滑鼠拖曳圈選想削除的地方。



②所選擇的地方會變色。

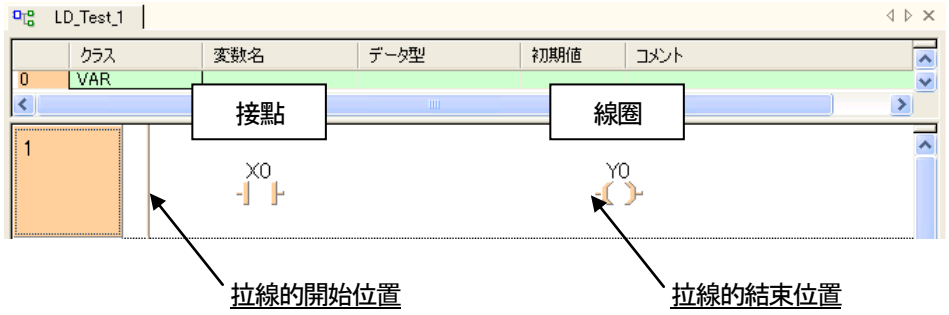


③按下 Del 鍵就會刪除選擇範圍。



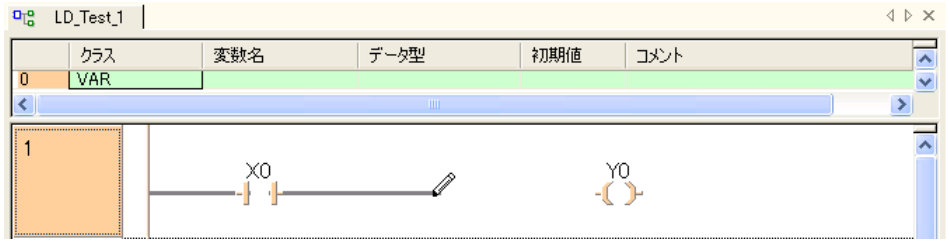
4-3-6 連接元件

將未連接的接點(元件)連接起來。

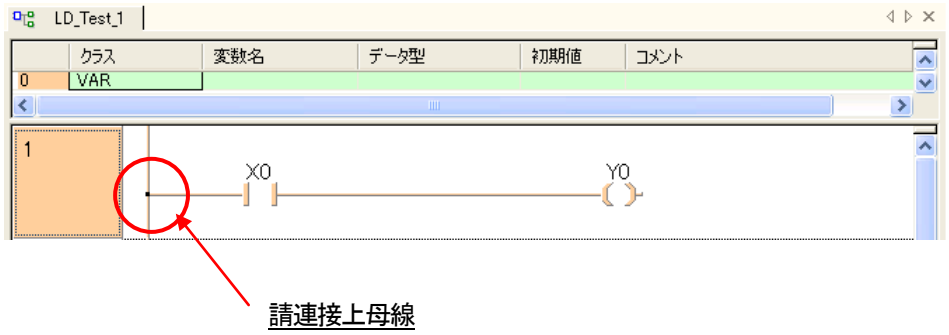


■操作順序

①點選畫線  圖示、從拉線的開始位置再次點擊。
按著將線拉到結束位置、放開滑鼠按鍵。

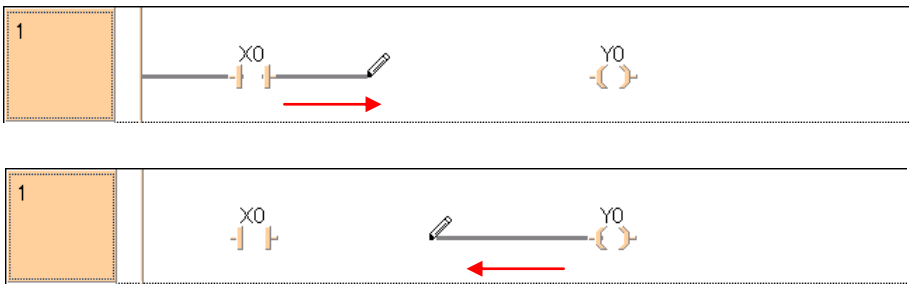


②拉線結束後、會變成以下的狀態。
程式內可以任意自由的畫線。



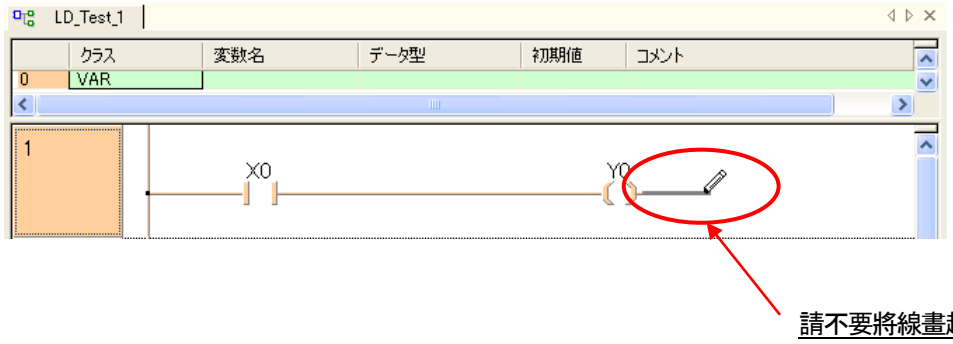
●備註

線可以從左從右都可以描畫。



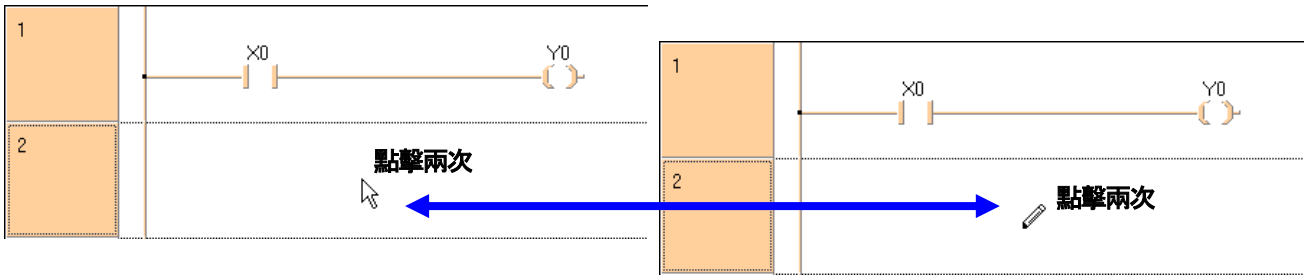
●注意事項

請不要將線畫超出輸出元件(線圈)。



●便利機能

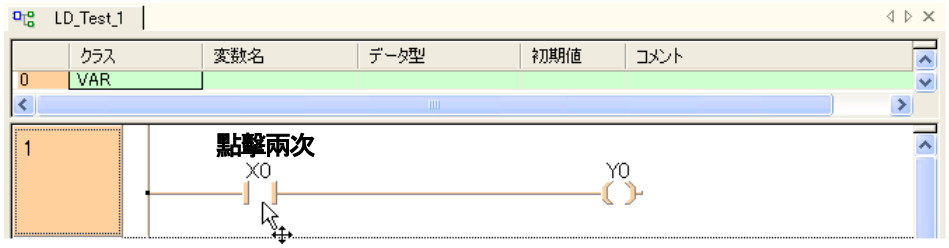
在程式編集區點選兩次、游標的方式會變為繪線用畫筆，即可使用。



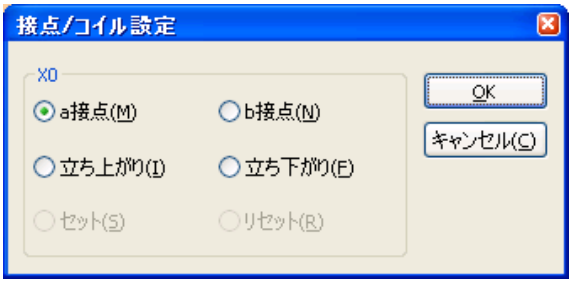
4-3-7 改變接點設定

■操作順序

①用之前畫好的程式、於接點 X0 用滑鼠標點擊兩次。

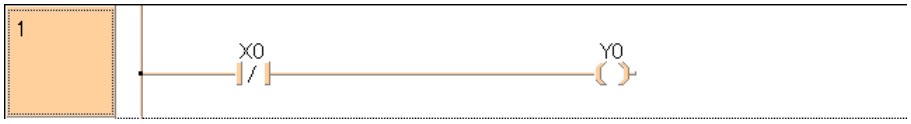


②顯示以下對話框。

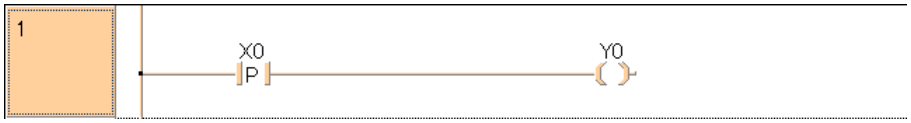


③在此、a 接點和 b 接點的設定、進行打開/關閉的設定。
初始值設定為「a 接點」。
以下顯示為各別選擇設定時的状態。

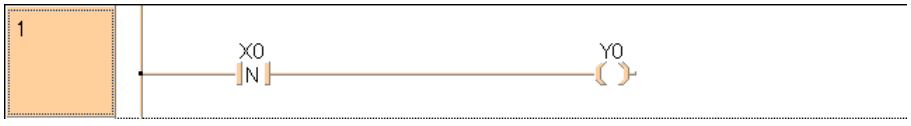
・b 接點



・上微分



・下微分



4-3-8 應用指令的輸入

①點擊工具列上的 。

會顯示以下指令選擇用的對話框。

在 FPWIN Pro 之中、接點以外的全部指令都由對話框輸入。

在此、位於左上的「Library」的欄位中、請點擊「FP Library」兩次。

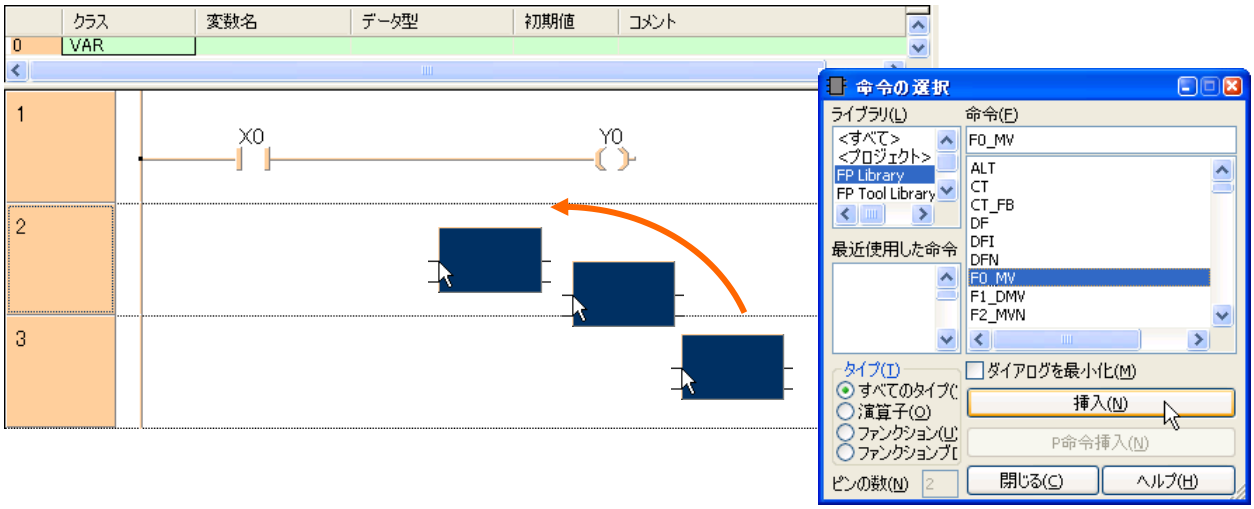
右邊的區域內、會顯示 PLC 的標準指令一覽。

②輸入資料傳送指令(F0 MV)。

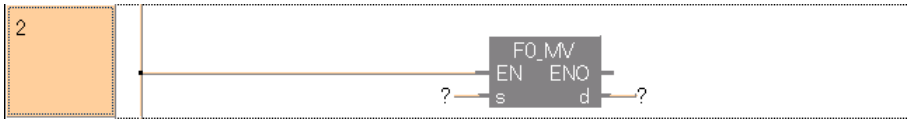
在 Library 中、將「F0_MV」的部分點擊兩次、按下「挿入」鍵。

如此、就選擇了 MV 指令。

將滑鼠移動到 LD 編集畫面上、會有以下的附上 F0_MV 指令的滑鼠指標。



在任意的位置按下滑鼠鍵、會如以下配置上 MV 指令。



各個 Pin 連接以下的內容。

EN: 連接這個指令的執行條件。

ENO: MV 指令執行後、別的指令就會起動、線圈會變成 ON 的時候使用。
(不使用時、不需要連接。)

s: 接收端

d: 發出端

③配置指令值。

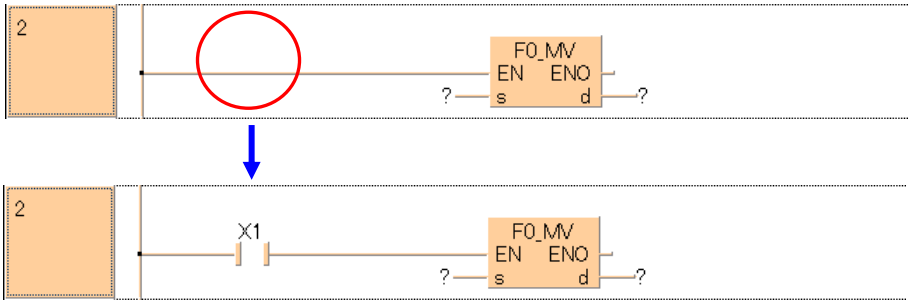
最後、對此輸入 MV 指令所必要指令值(EN、s、d)。(ENO 不使用時不需要連線。)

點擊"?"的地方、請輸入元件。

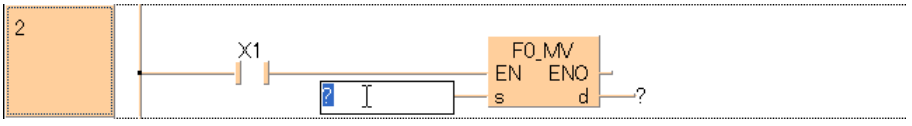
這次、輸入執行條件(EN)為 X1、輸入側(s)為 DT0、輸出側(d)DT1。

對執行條件(EN)的輸入

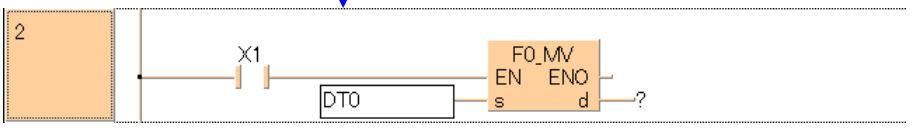
連接 EN 的線上、將 X1 的 a 接點貼上去。



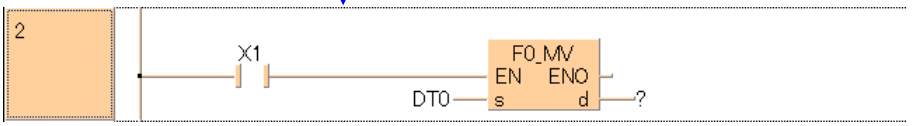
對輸入側(s)、輸出側(d)的輸入



點擊輸入側(s)的"?"部。

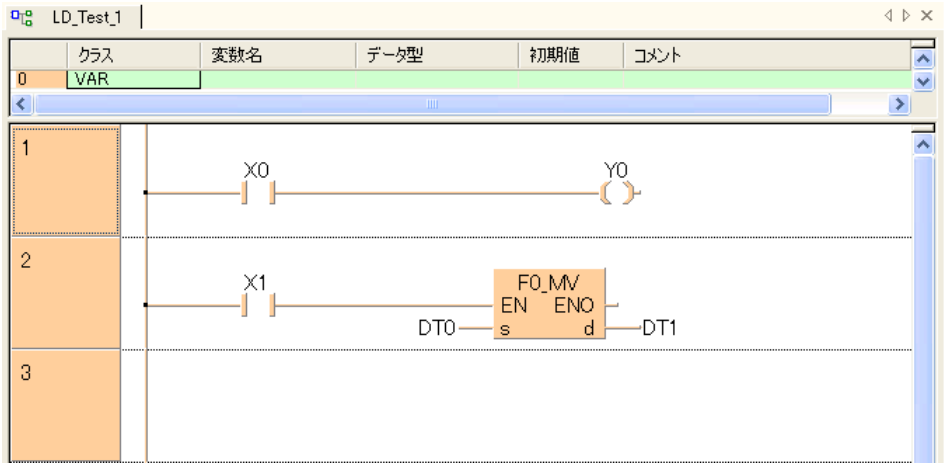


輸入 DT0。



按下 Enter 鍵確定。

用同樣方法、於輸出部(d)輸入 DT1 就會完成。



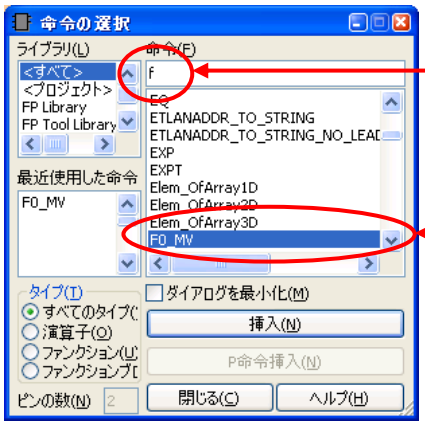
●備註

應用指令輸入時的指令搜索

如果知道預先輸入的指令名稱時、在顯示應用指令選擇用對話框、於選擇對話框輸入指令的字首、輸入的文字後就會顯示指令一覽。

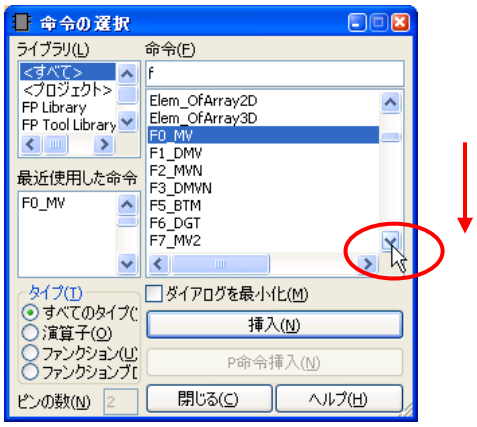
選擇對話框

輸入指令的字首。



輸入字首。
(不分大小寫)

例如左圖輸入“f”。
會跳到 PLC 的 Fun 指令的前頭(F0 MV)。



之後、Fun 指令會依字母順序排列、
搜尋目的 Fun 指令。
因為會跳到附近、
在搜尋指令時相當有效率。

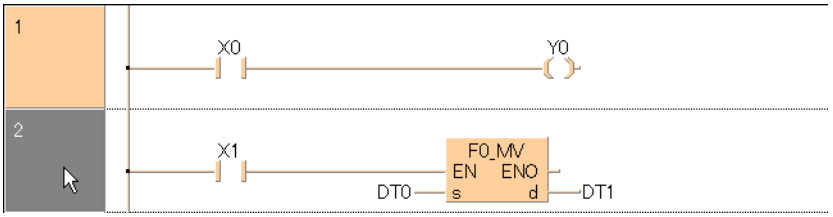
4-4 關於編集補助功能

4-4-1 Network 的追加･挿入･削除･複製･移動

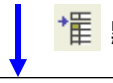
Network 的 1 和 2 之間插入新的 Network 時如以下順序。

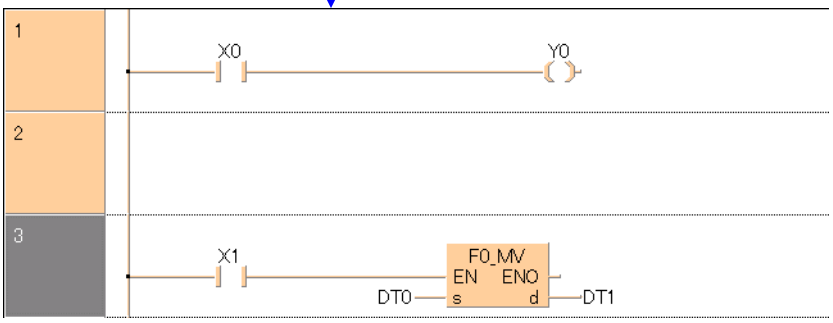
■操作順序(追加･挿入)

將選擇的 Network 上面插入新的 Network

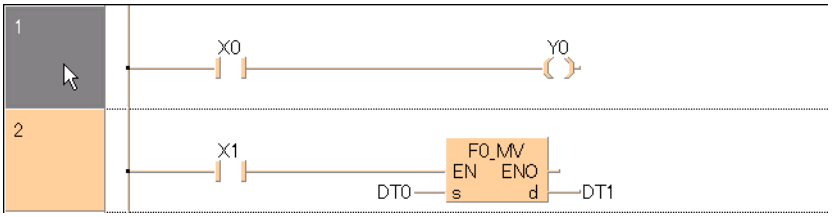


2 的 Network 設定為動作。(用滑鼠點擊)

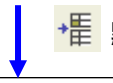


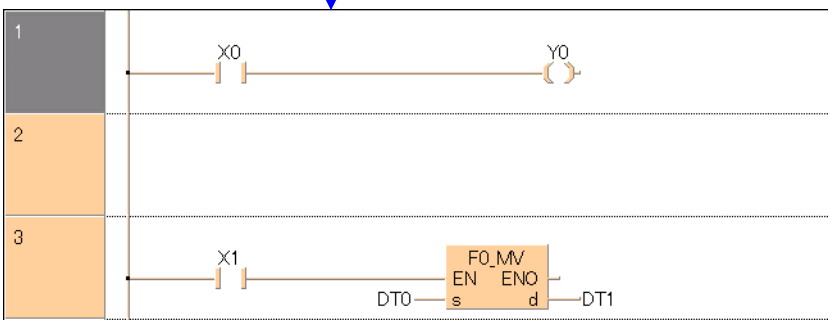


將選擇的 Networkg 下面插入新的 Network



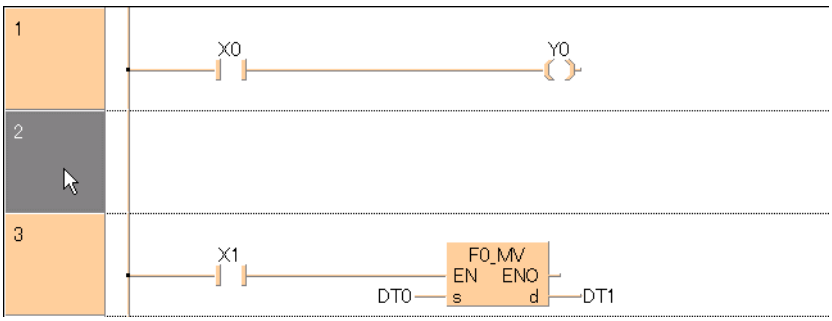
1 的 Network 設定為動作。(用滑鼠點擊)





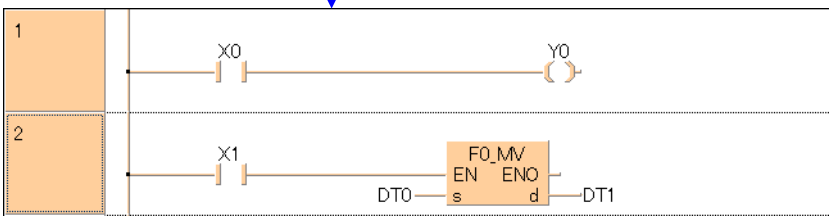
■操作順序(削除)

將剛才插入的 Network 削除看看。



想削除的 Network(本次為 Network2)使其動作。(點擊滑鼠)

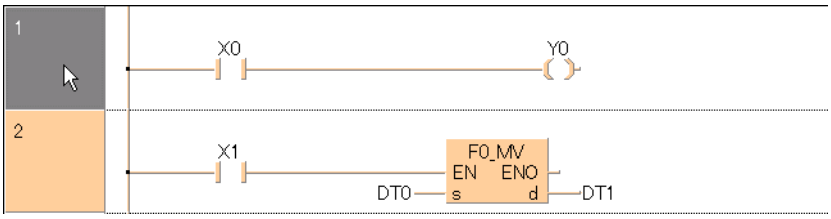
按下 Del 鍵。



選擇的 Network 將會被削除。

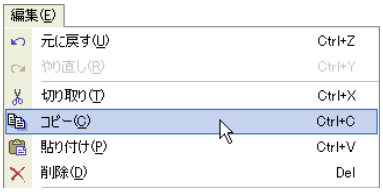
■操作順序(複製)

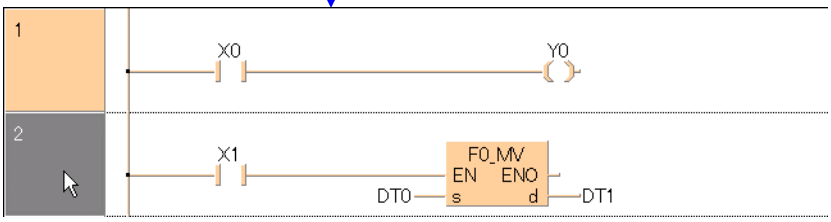
將 Network1 複製看看。



想複製的 Network(本次為 Network1)使其動作。(點擊滑鼠)

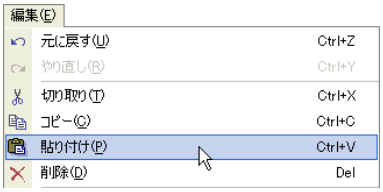
「Ctrl」+「C」鍵或是
請選擇**(選單)編集 → 複製** 。

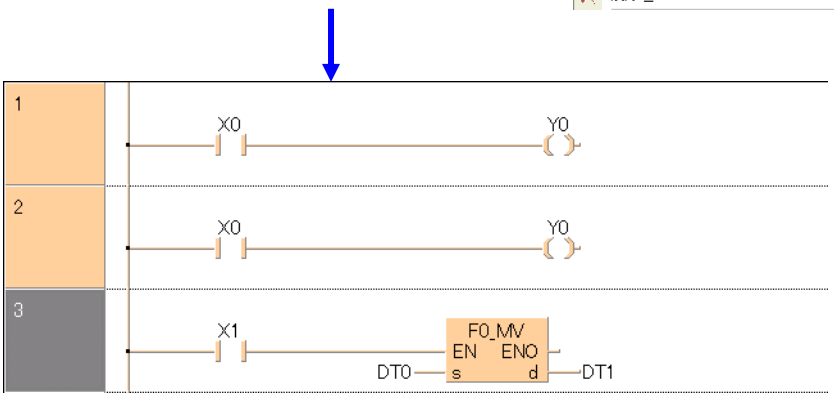




將 Network2 使其動作。(點擊滑鼠)

「Ctrl」+「V」鍵或是
請選擇**(選單)編集 → 貼上** 。



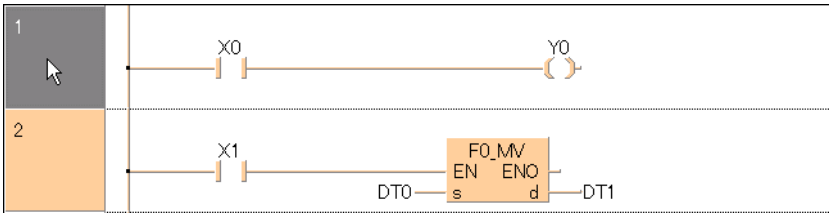


完成 Network 的複製。

貼上時會貼在選擇的 Network 上面。

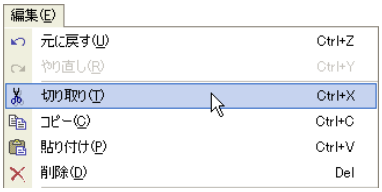
■操作手順(移動)

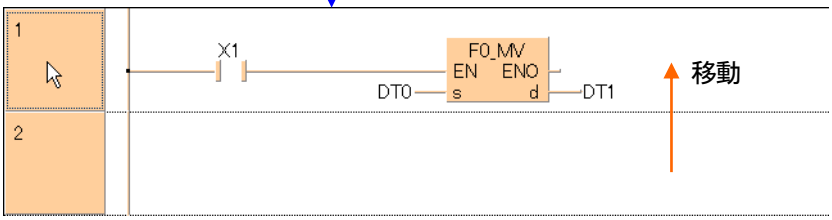
移動到 Network 看看。



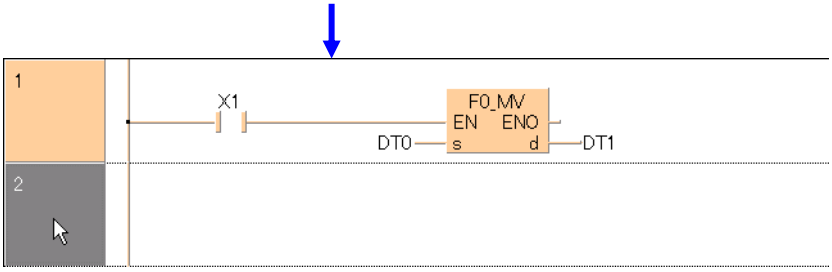
將想移動的 Network(本次為 Network1)使其動作。(點擊滑鼠)

「Ctrl」+「X」鍵或是
請選擇**(選單)編集 → 剪下**。



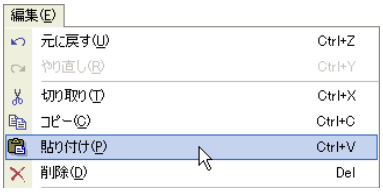


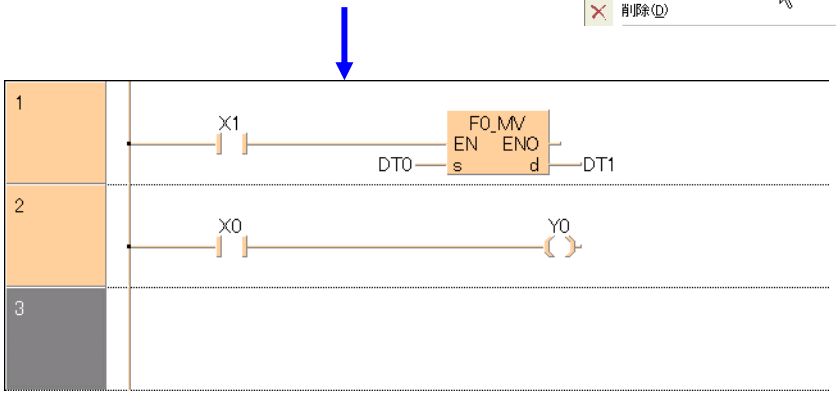
選擇的 Network 會消失、Network 會往上一段移動。



將想移動的 Network(本次為 Network2)使其動作。(點擊滑鼠)

「Ctrl」+「X」鍵或是
請選擇**(選單)編集 → 貼上**。





Network 移動成功。

移動時、貼到動作的 Network 上一段。

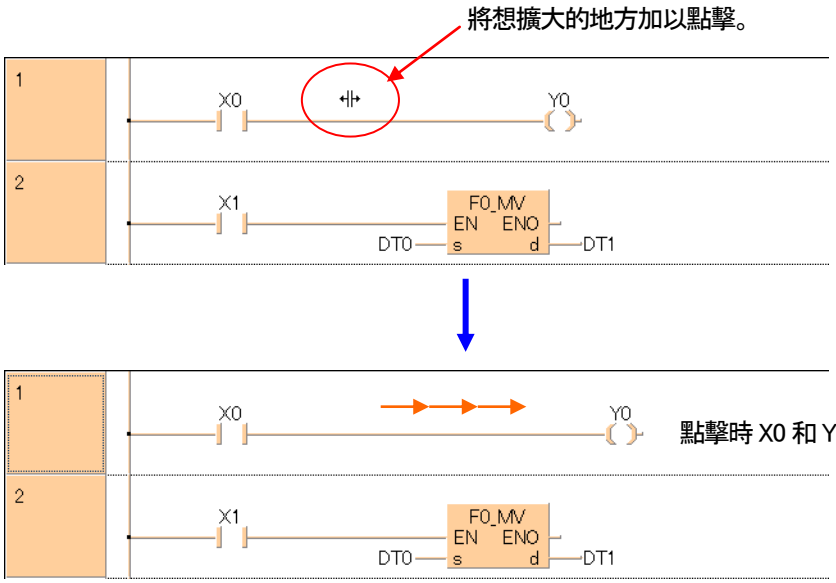
4-4-2 任意的地方使其擴大

X0 和 Y0 之間加以擴大、想放入新的元件時如以下的順序。

■操作順序

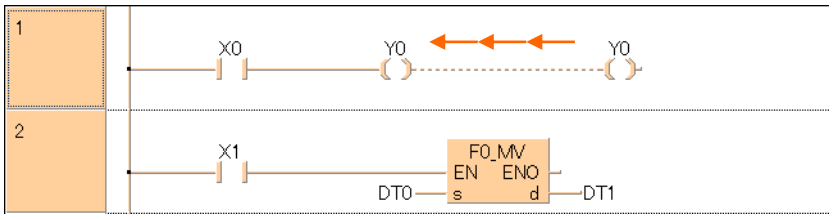
<u>橫向擴大</u>

①點擊工具列的  、將滑鼠移動到編集畫面上想擴大的地方。
②點擊滑鼠、將其擴大。會應映滑鼠點擊次數、擴大範圍會變大。



點擊時 X0 和 Y0 之間的寬度會變大。

寬度狹窄時



按著「Ctrl」鍵、點擊時會相反的寬度可以變窄。

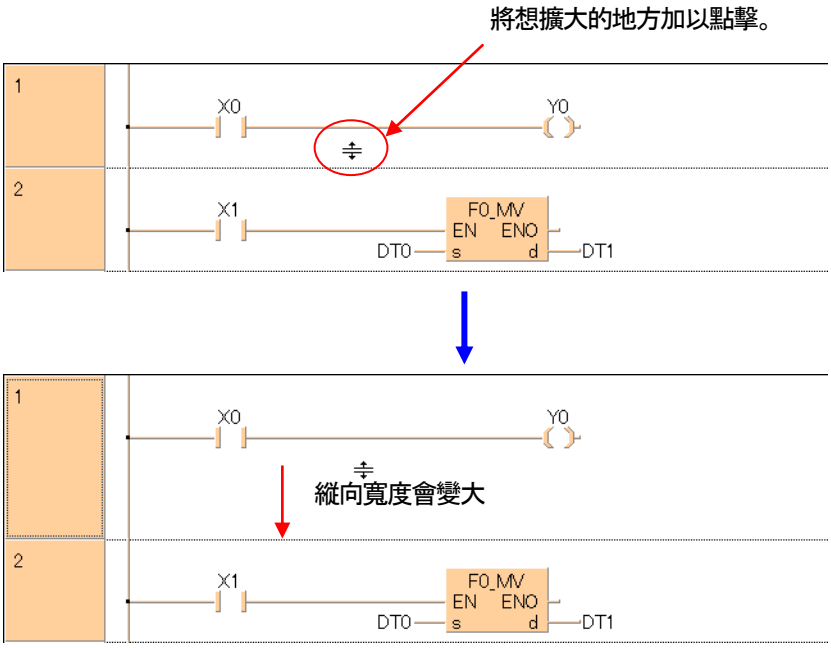
●備註

※擴大、縮小只會在同一個 Network 之內。不會影響其他 Network。
就上圖而言、Network 的寬度擴大或縮小、並不會影響到 Network2。

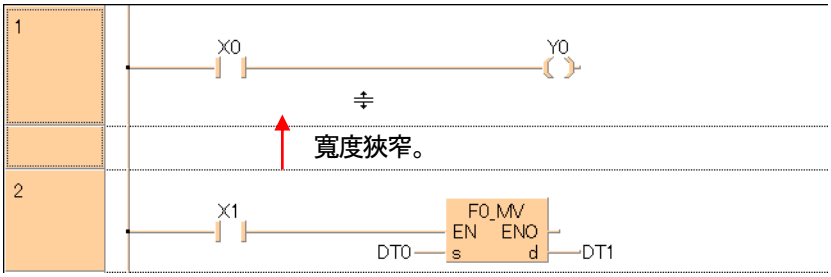
■操作順序

縱向擴大

①點擊工具列的  、將滑鼠移動到編集畫面上想擴大的地方。
②點擊滑鼠、將其擴大。會應映滑鼠點擊次數、擴大範圍會變大。



寬度狹窄時



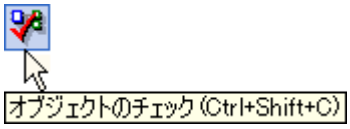
按著「Ctrl」鍵、點擊時會相反的寬度可以變窄。

4-5 執行編譯

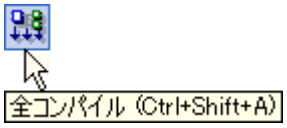
在此對剛才完成的階梯圖程式進行編譯。(相當於 FPWIN GR 的 PG 轉換。)
在 FPWIN Pro 中、此作業稱為「編譯」。

4-5-1 關於編譯

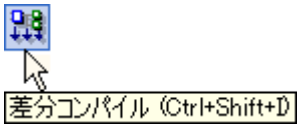
■在 FPWIN Pro 的編譯檢查、有以下 3 個。



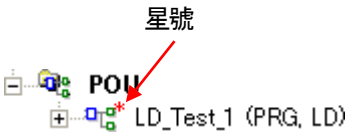
・Object 檢查
在此、只對現在動作的畫面進行檢查。
在系統暫存器的設定畫面動作的話、雖然會進行系統暫存器的檢查、
但是此時、不會檢查 LD 的編集畫面。
對於在現在編集中的程式的簡單文法檢查是有效的功能。



・全編譯
會全部進行檢查系統暫存器及程式等、
會產生出傳送到 PLC 的資料。
全編譯雖然包含有 Object 的檢查、
因此只有執行此功能也沒有關係。

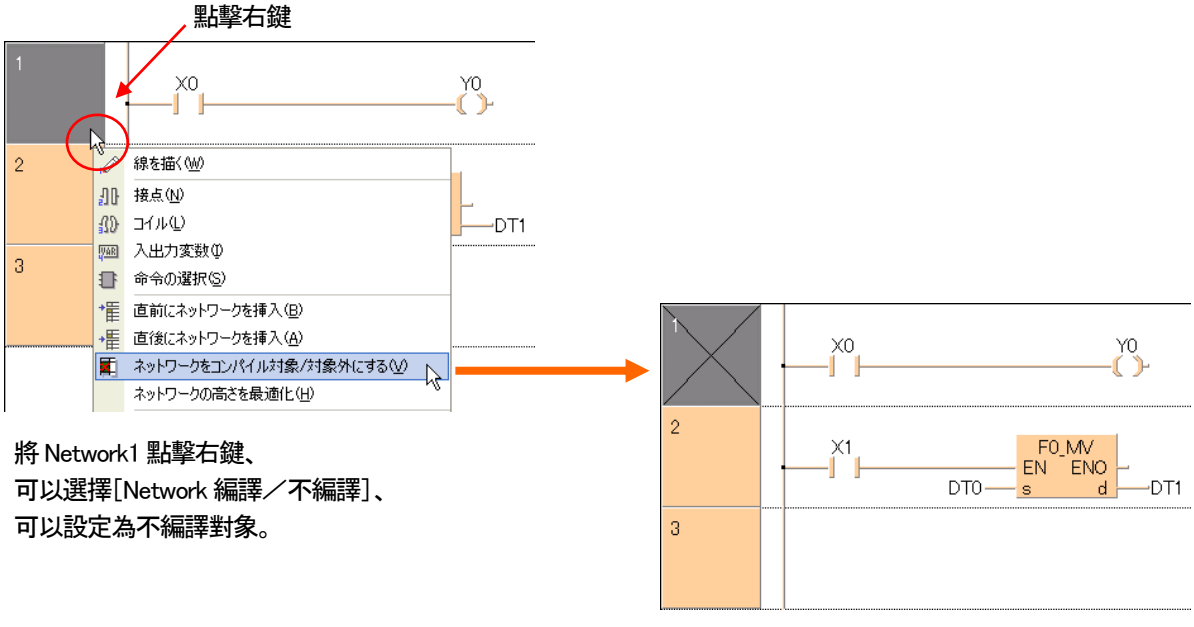


・部分編譯
變更編譯過的程式內容時、
只檢查變更的部分、進行編譯。



※文件引導附有星號(*)的文件是尚未編譯還在編集中的 Object。

■將 Network 作為不編譯對象

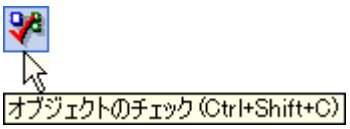


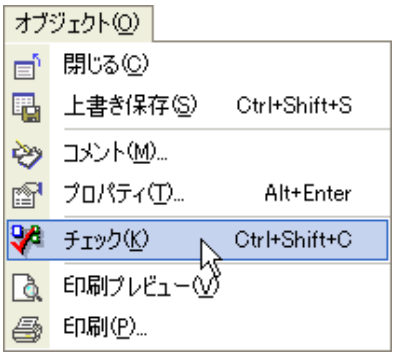
將 Network1 點擊右鍵、
可以選擇[Network 編譯／不編譯]、
可以設定為不編譯對象。

4-5-2 進行 Object 檢查

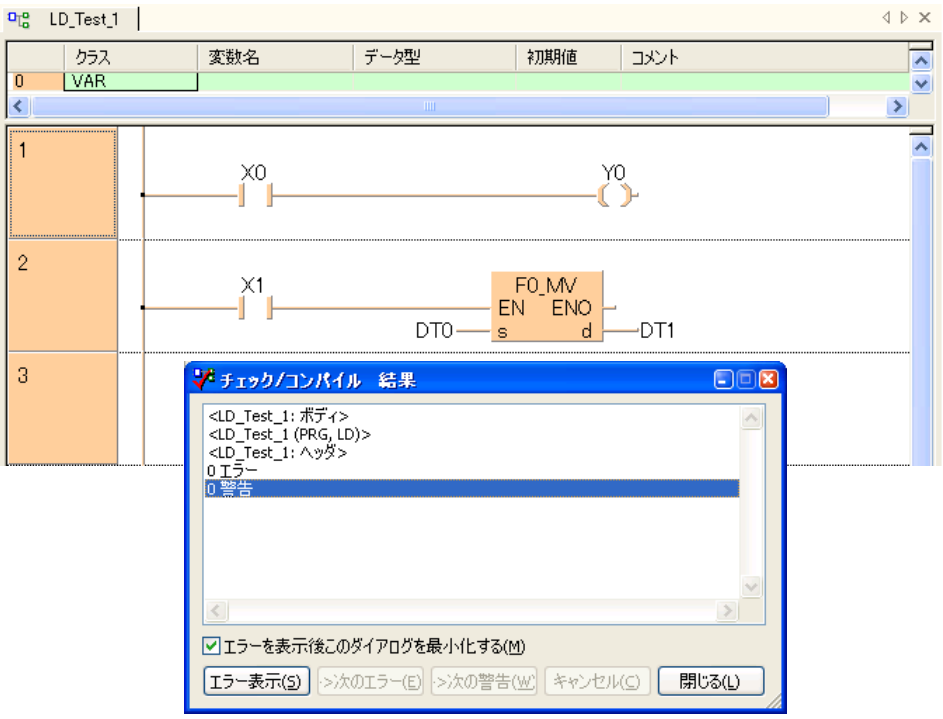
對於要進行 Object 檢查時、首先移至想進行檢查的編集畫面並點選、

 圖示、或是選擇(選單)元件 → 檢查。





LD 編輯畫面使其動作、執行 Object 檢查。會執行畫面如下。



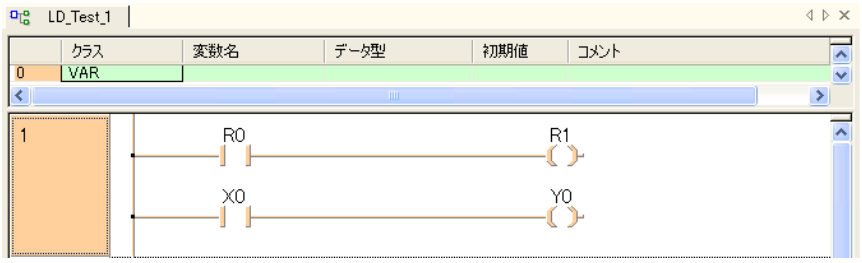
在上例中結束檢查時雖然沒有發生問題、但有時會出現錯誤或警告。

錯誤發生時必須進行處理、不然程式無法下載到 PLC、警告不處理也可以下載到 PLC。

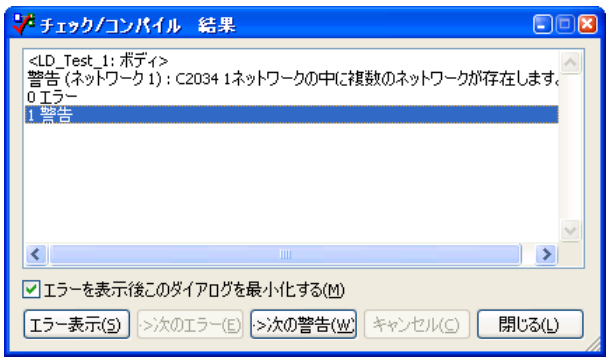
4-5-3 發生錯誤和警告的時候

■在 1 個 LD 編集畫面中、編寫 2 個 Block。

編集畫面的状態



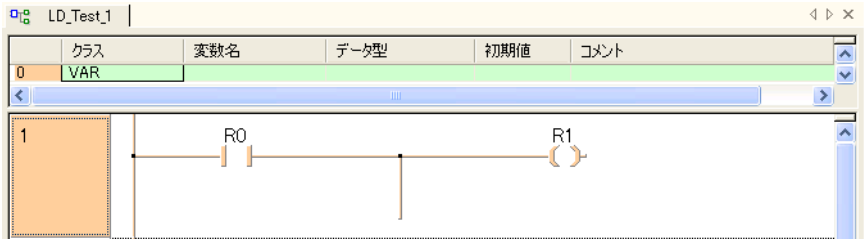
檢查的結果



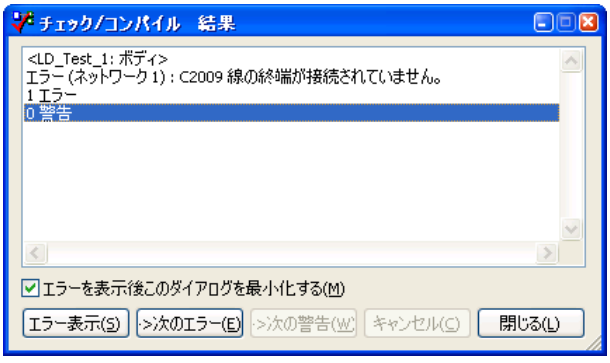
對策:第 2 個階梯圖 Block、請移動到其他的 Network。

■存在未連接的線。

編集畫面的状態



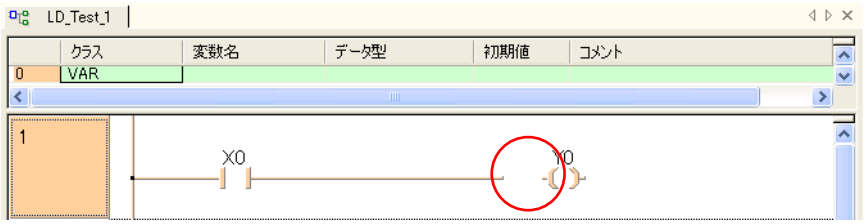
檢查的結果



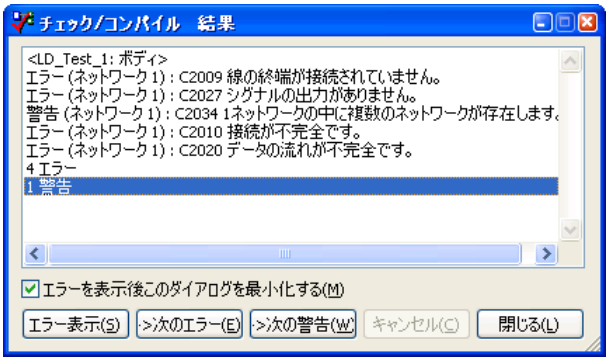
對策:請刪除未連接的線。

■未連接的線。

編集畫面的状態



檢查的結果



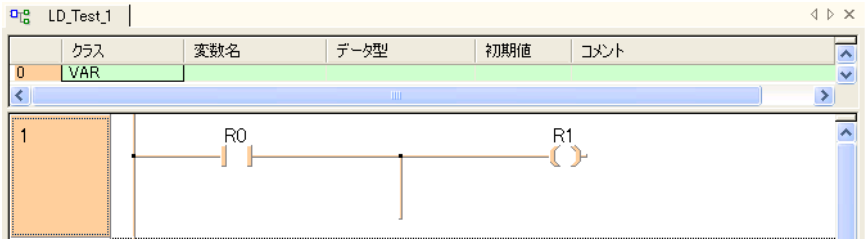
雖然顯示錯誤有 4 個警告有 1 個、
原因只有 1 個地方。

對策:請把線接起來。

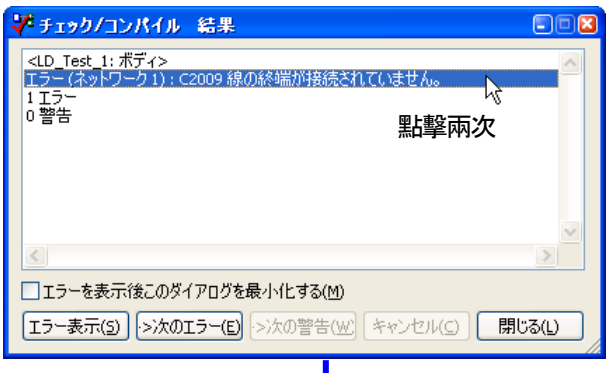
●備註

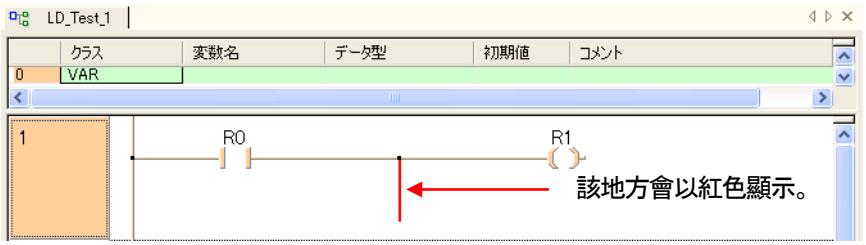
檢查結果顯示錯誤時、在錯誤的地方點擊兩次、可以檢索錯誤的位置。

編集畫面的状態



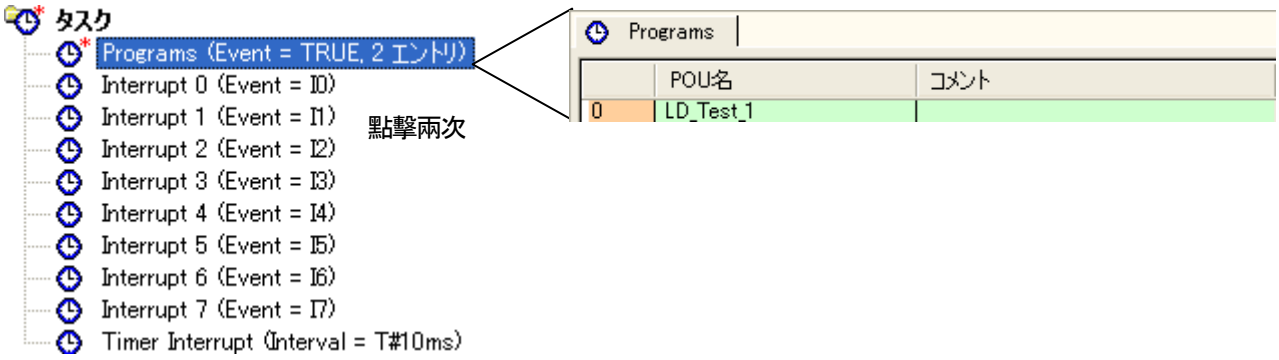
在檢查結果對話框、在錯誤的地方點擊兩次。





4-5-4 執行編譯

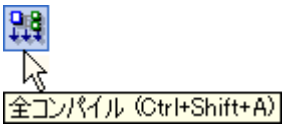
「Object 檢查」如果沒有問題就可以執行編譯。
編譯結束時沒有問題的話、就可以將 Object 下載到 PLC。
（即使不進行 Object 檢查、執行編譯也沒有關係。Object 檢查未執行時、在編譯時會自動執行。）
編譯執行前、Object 導引的「Task」裡面的「Programs (Event = TRUE)」點擊兩次、
請確認以下狀態。

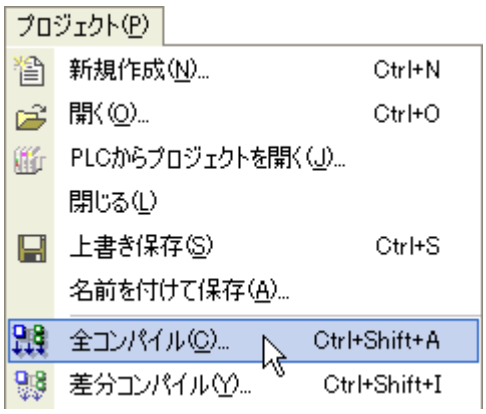


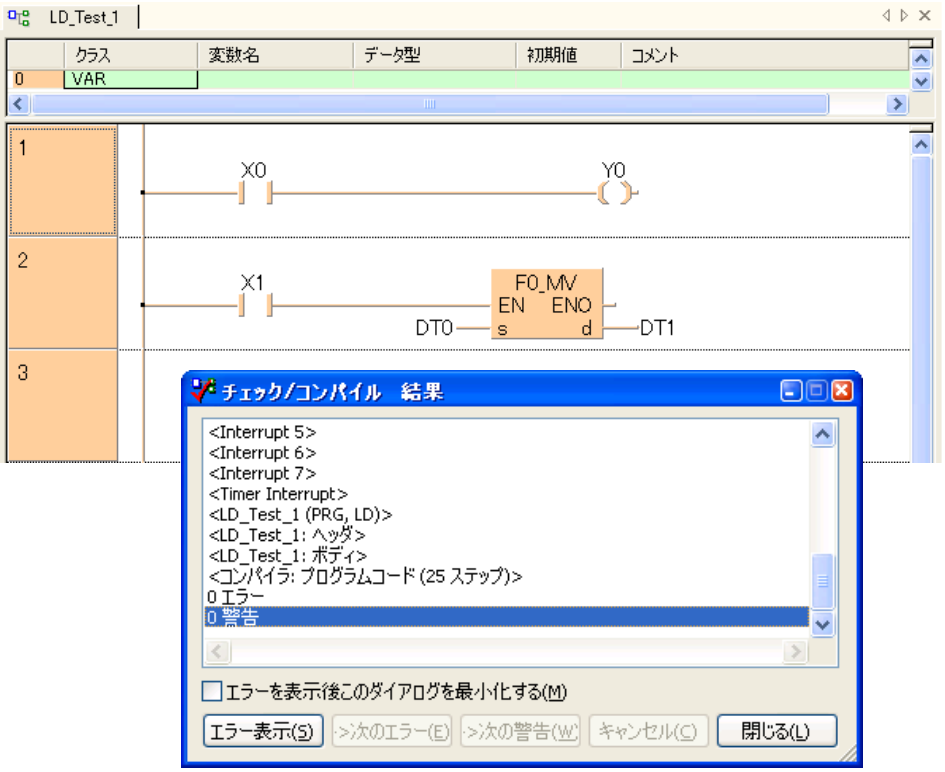
在此、可以確認 POU 的「LD_Test_1」是否已經登録。
FPWIN Pro 只有在 Task 所登録的 POU 執行編譯。
新製作的 POU、會自動登録到 Task、所以可以直接執行編譯。
有複數個 POU 登録時、從上往下依順執行編譯。

全編譯會對文件的全部情報進行檢查後、進行編譯。

 的圖示可以點擊、或是選擇**(選單)文件 → 全編譯**。







如圖上所示、「0 錯誤、0 警告」的編譯為正常結束。
在此發生錯誤時、請在該訊息上點擊兩次、會自動移到錯誤地方再加以修正。

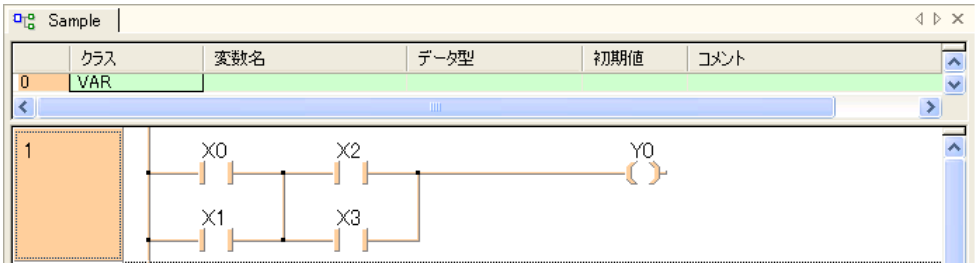
並且、此對話框即使打開、可以進行程式以及系統暫存器設定畫面的修正。

4-6 編譯的注意事項

在 FPWIN Pro 的編譯時、雖然和 FPWIN GR(或是 NPST-GR)的 PG 轉換是相同的動作、
但在實際的變換方法有部分的差異。
其中以下項目為相當重要、請確認使用 FPWIN GR 和 NPST-GR 有不同的地方。

4-6-1 生成的記憶指令不同

例如以下的編集階梯圖並轉換。



在 FPWIN GR 的場合、變換成以下編碼。

```
ST   X0
OR   X1
ST   X2
OR   X3
ANS  (And Stack)
OT   Y0
```

在 FPWIN Pro 的場合、變換成以下編碼。

```
ST   X0
OR   X1
OT   R240
ST   X2
OR   X3
AN   R240
OT   Y0
```

將改成階梯圖的形式、則變成以下內容。

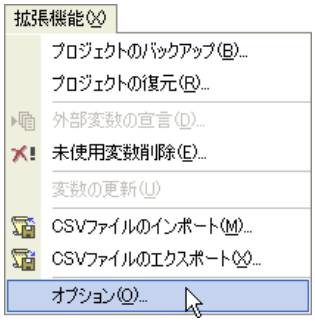
把這些內容進行比較、可以看到相同動作的程式被以不同的編碼呈現。
此外實際程式的 STEP 數目也會不同。(ANS 和 ORS、通常不會轉換。)
因此在 FPWIN Pro 進行編譯會產生和 FPWIN GR 不同的編碼。

4-6-2 PLC 本體演算用記憶體的使用

編譯執行時、使用到 PLC 內部記憶體(繼電器和暫存器等)的時候、
實際程式可使用的 PLC 內部記憶體會變少。

首先在前頁程式中、請注意有使用但沒有描繪的 R240。
這個意味著 FPWIN Pro 在執行編譯時、會使用 PLC 的內部記憶體。

請選擇**(選單)擴張功能 → OPTION → 編譯 OPTION → 位址範圍**。
顯示以下的對話框。



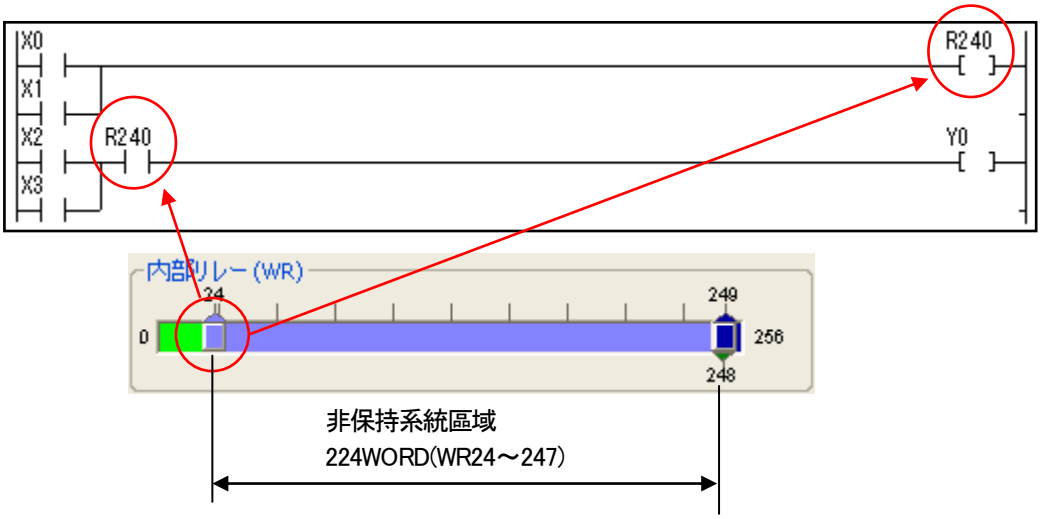
如上圖所示在 FP Σ(32K 型)的初期值、

內部繼電器(R) : WR24～247(224WORD 非保持)、WR249～255(7WORD 保持)
資料暫存器(DT) : DT3271～32709(29439WORD 非保持)、DT32716～32762(47WORD 保持)

分配於編集用、在實際程式是無法使用。

這些值雖然可以變更、但是過於太小、在編譯時無法轉換程式、有可能會發生錯誤(ERROR)。

用剛才的階梯圖為例

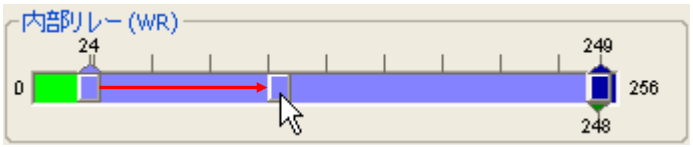


以 FP Σ (32K 型)的初期設定去編譯後、
非保持型的地方、從 R240 開始分配給編集用、
如上圖例在編譯後、會自動產生 R240。

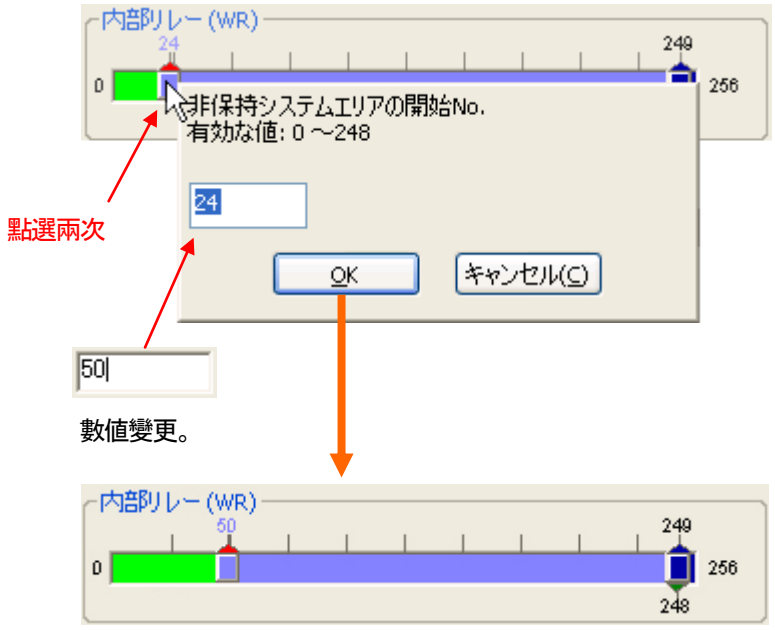
●備註

編譯領域的變更方法

編譯領域變更時、如下圖般用滑鼠拖曳就可以變更。



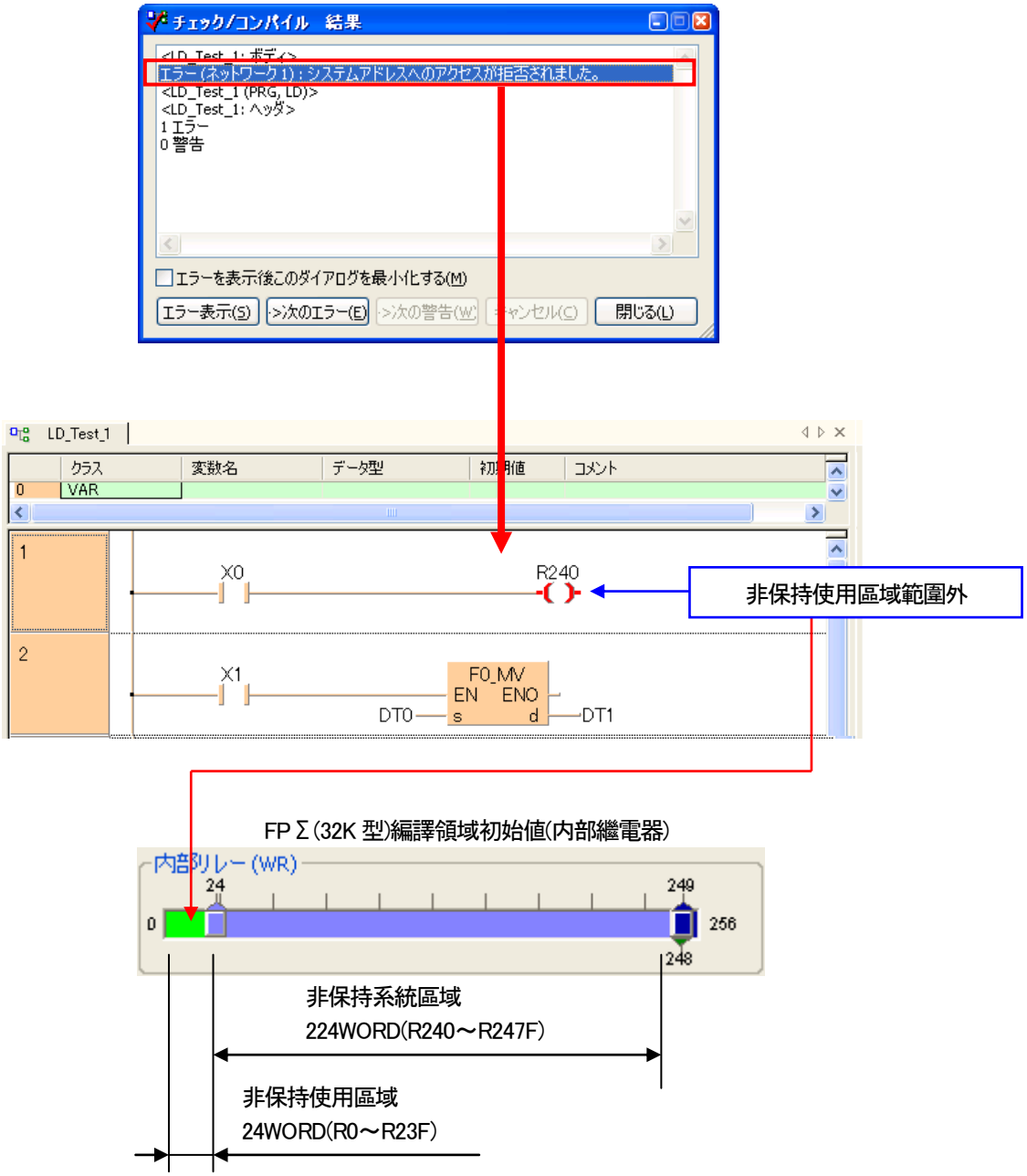
直接輸入數值變更時、
將想變更的地方點擊兩次。(下圖例的內部繼電器非保持區域)



可以確認編集領域已經變更。

編譯領域所導致的錯誤

編譯後、顯示以下的錯誤訊息時、其原因為可能為編譯領域錯誤發生。



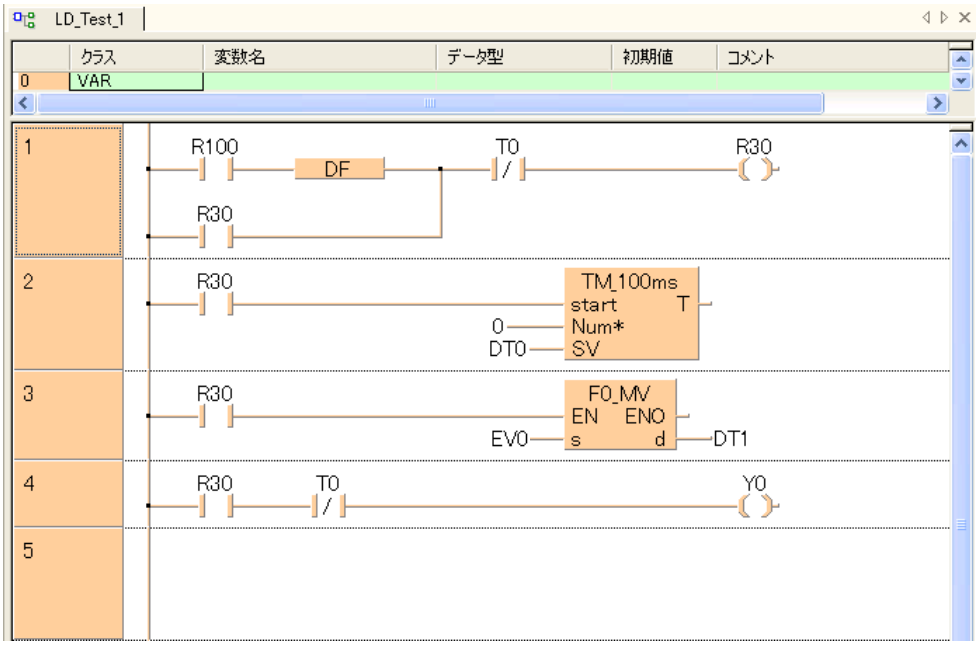
上圖例中、稱為「程式中使用了可以做為使用者區域的內部繼電器的範圍之外的部分」的錯誤。
請在程式中的 No.(R240)修改成範圍內的 No.(R0～23F)、以變更編譯領域、增加使用領域來避免以上的錯誤。

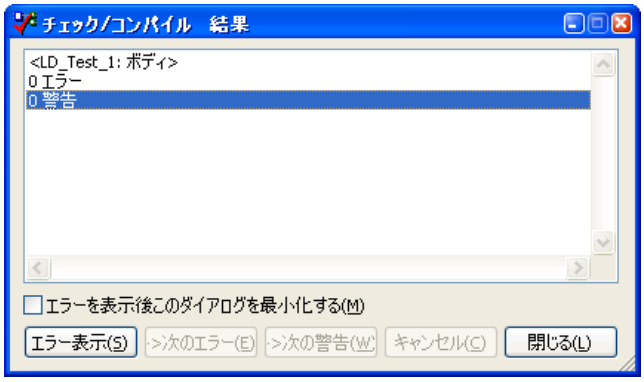
通常、在 FPWIN Pro 領域(在上例為 R240)輸入時、必須吻合編譯領域、
範圍外時會無法輸入。

上圖錯誤作為錯誤發生事例、應考量
- 「元件輸入後、變更編譯領域。」
- 「FPWIN GR(*.fp)文件導入。」等。

【 課題 】

請作成下圖的程式。





執行編譯、確認沒有錯誤。

第5章

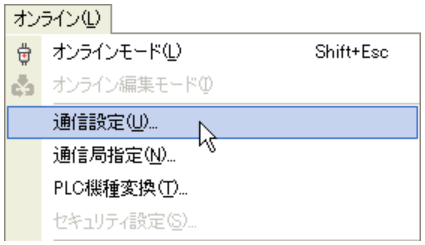
On line

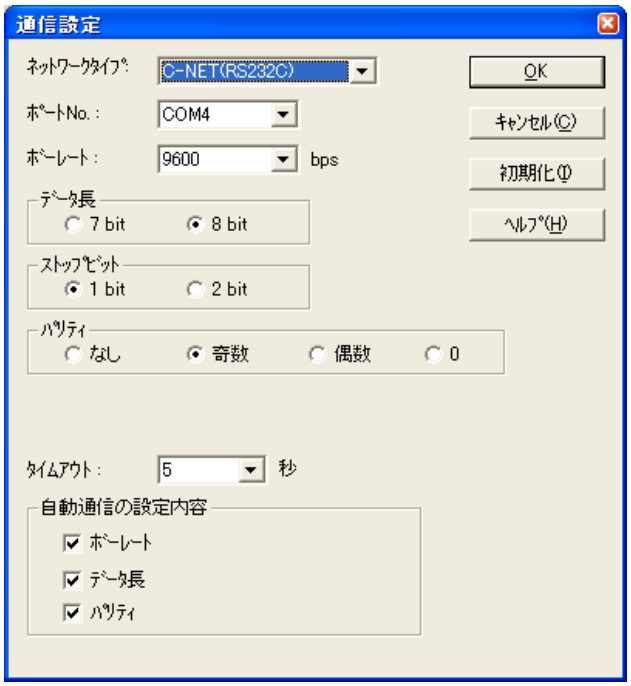
5-1 和 PLC 連線

用第 4 章的課題作成的程式下載到 PLC。

5-1-1 設定通信條件

首先、選擇**(選單)On line → 通信設定**、起動以下對話框。

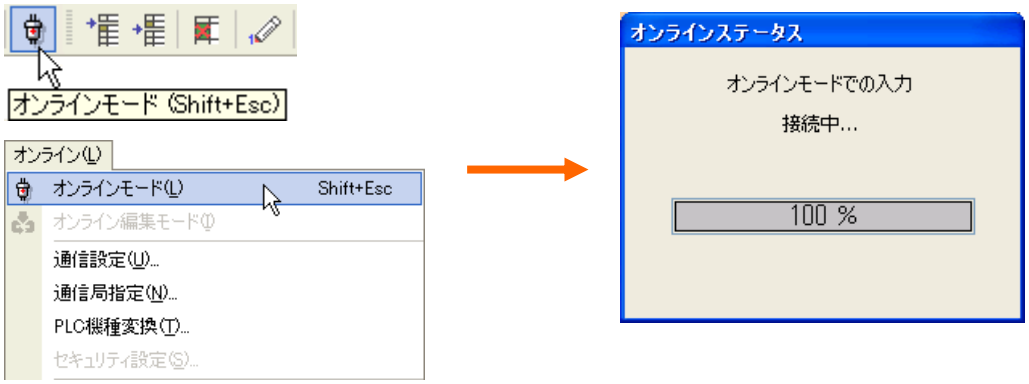




在此、設定所使用的通信速度等。(也請確認 PLC 方面的設定。)
按下「OK」、就會以現在設定保存。

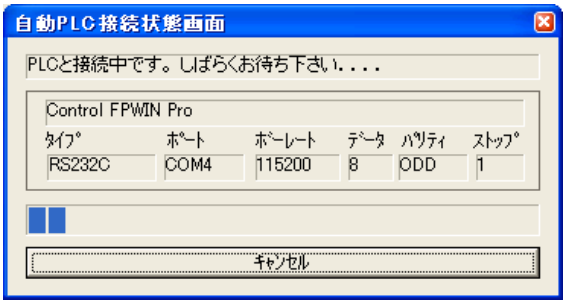
5-1-2 進行連線

點選(選單)On line → On line Mode、
選擇工具列的 FPWIN Pro 和 PLC 開始通信。

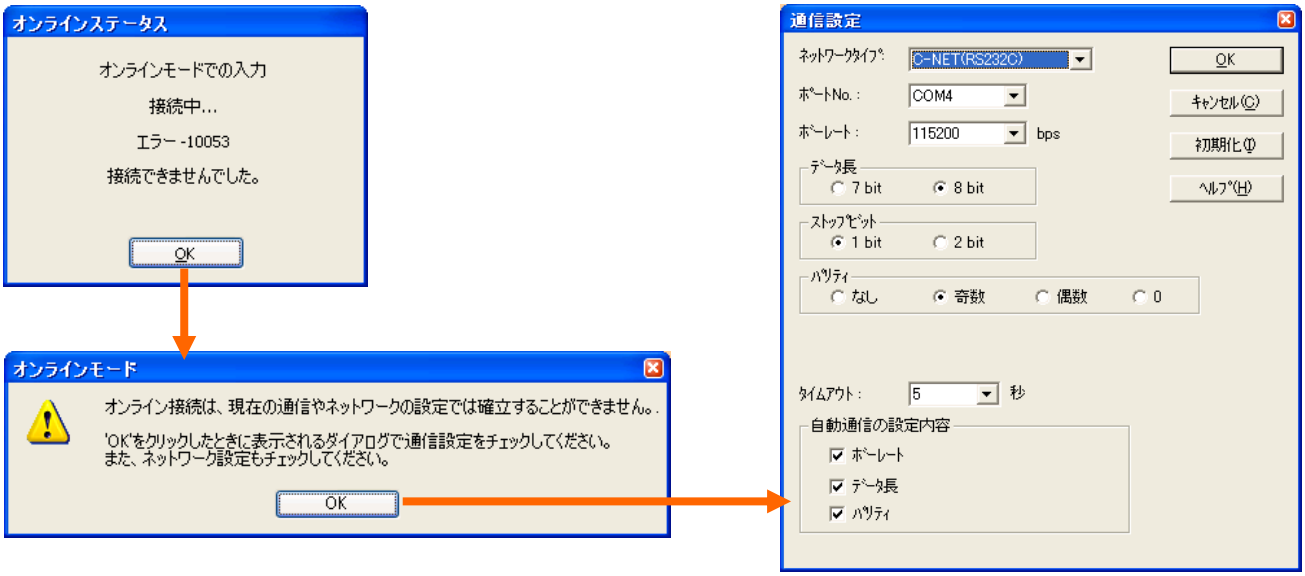


和 PLC 通信成功、就會顯示新的 On line 用的工具列。

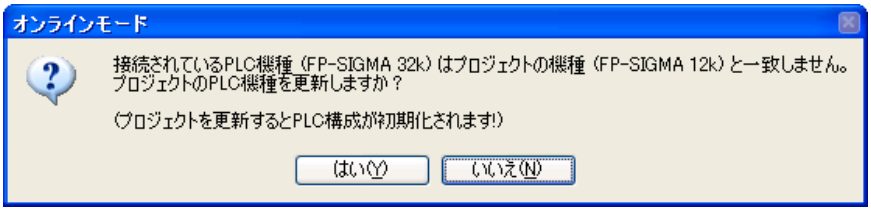
用在「通信設定」所設定的條件和 PLC 通信無法連接時、會自行啓動自動通信設定功能。
所點選項目的通信條件會一邊變更、一邊跟 PLC 的通信作確定。但是通信 port No.並不會自動變更、
如有不同時、請變更設定。



即使全部檢測也確定無法和 PLC 通信時會顯示錯誤訊息、
移往 On line 處理會中止。



並且、即使通信成立、在文件所設定 PLC 和實際所連接的 PLC 機種不同時、
會顯示以下訊息。



はい(Y) 選擇是、會遵從所連接的 PLC 機種進行系統暫存器的初始化等、
會移往 On line 模式。

いいえ(N) 選擇否、移往 On line 模式會停止。

■常使用的小圖示

在 Online 用的工具列中、以下的按鍵會經常使用、其功能請牢記。

 **下載程式到 PLC。**

 **Online 編集時、只下載程式變更部分。**

 **移動到 Online 編集模式。**

 **切換到程式監控模式。**

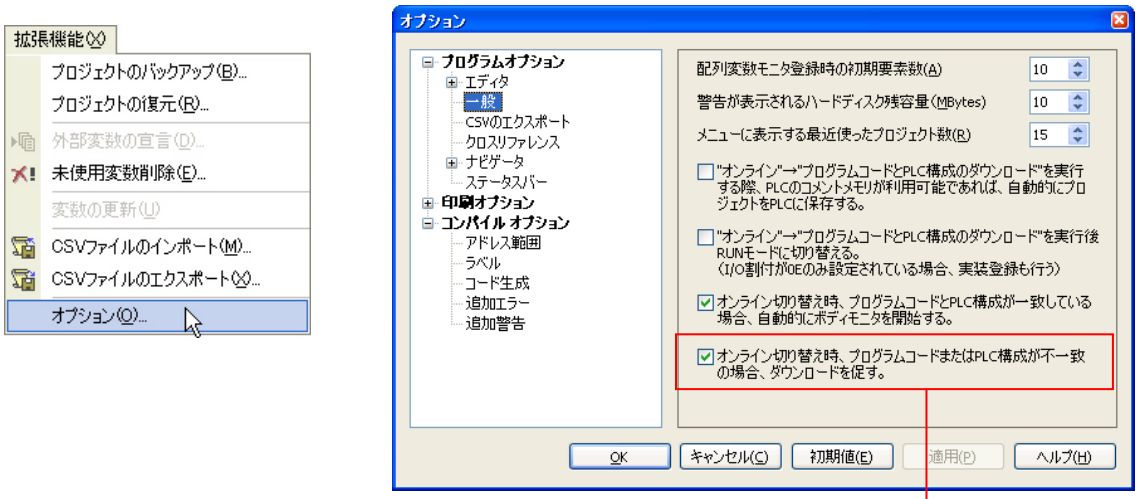
切換到 PLC 的動作模式。

5-2 執行程式下載

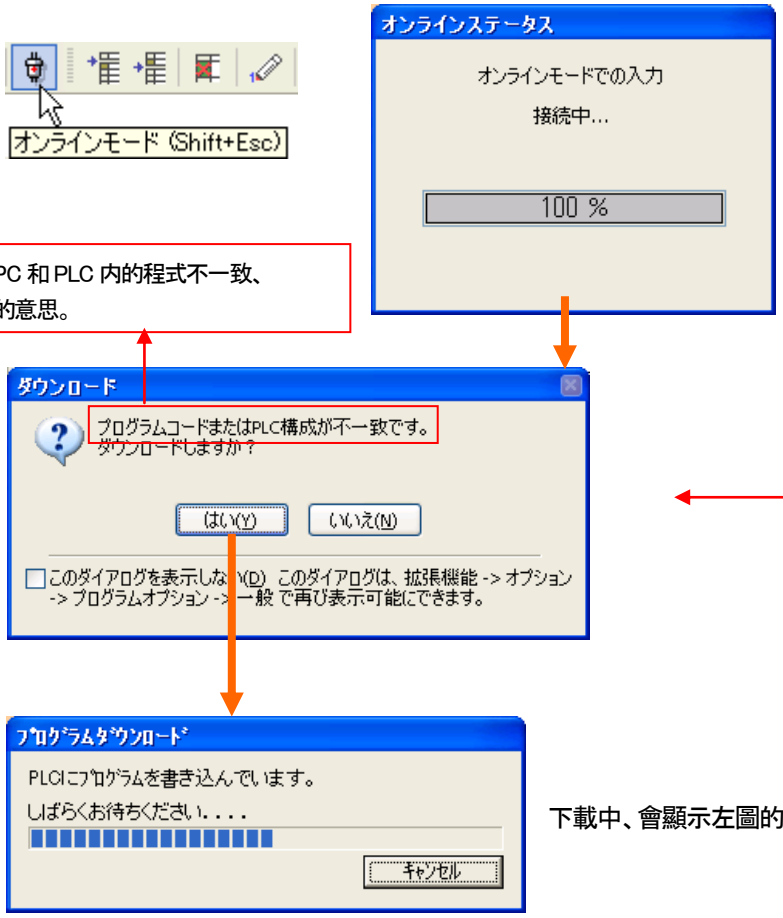
如果可以移動到 Online 模式的話、請用第 4 章的課題所作成的程式下載到 PLC。

5-2-1 移動到 online 時的下載順序

在 FPWIN Pro 的初期設定中、
設定為**「Project 不一至的話、移動到 Online 之後、會要求下載程式和系統暫存器」**。



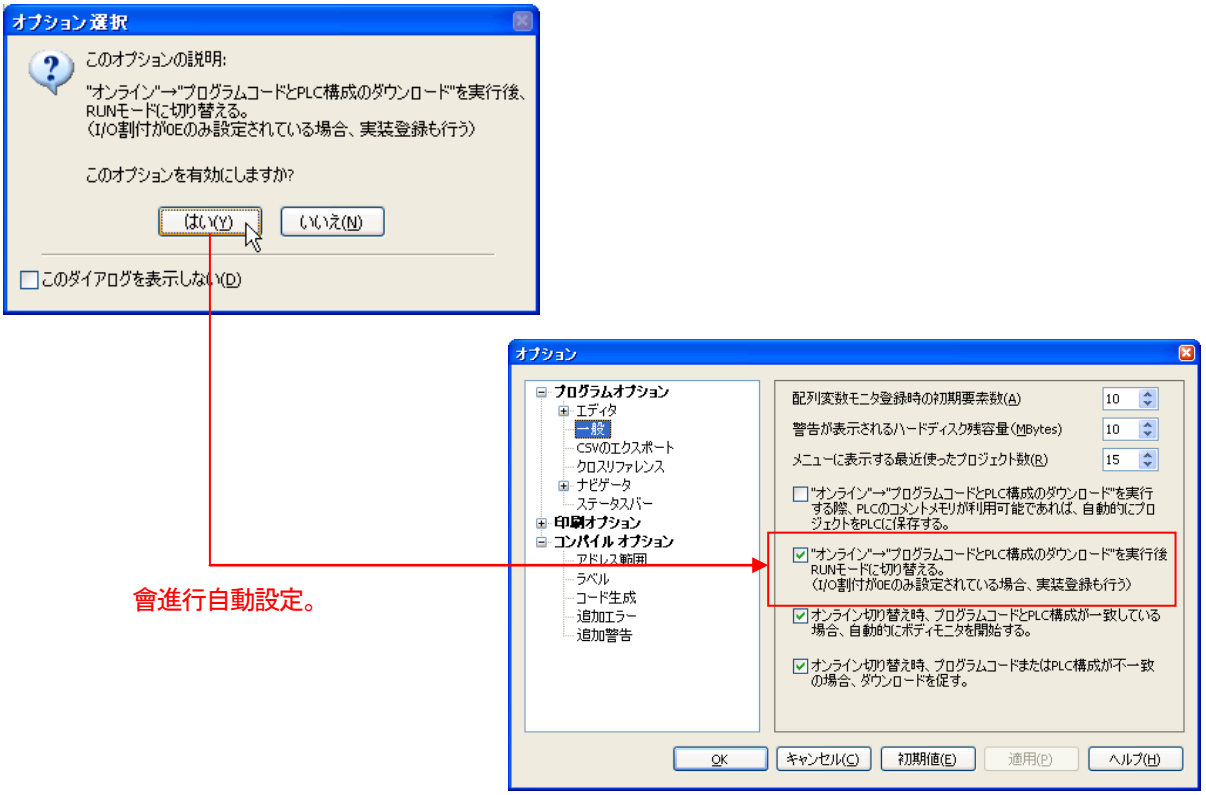
■操作順序



下載中、會顯示左圖的畫面。

●選項的選擇(PLC 模式(RUN/PROG.)自動切換有效)

下載時、會顯示如下圖的對話框、 選擇是 はい(Y) 的時候。



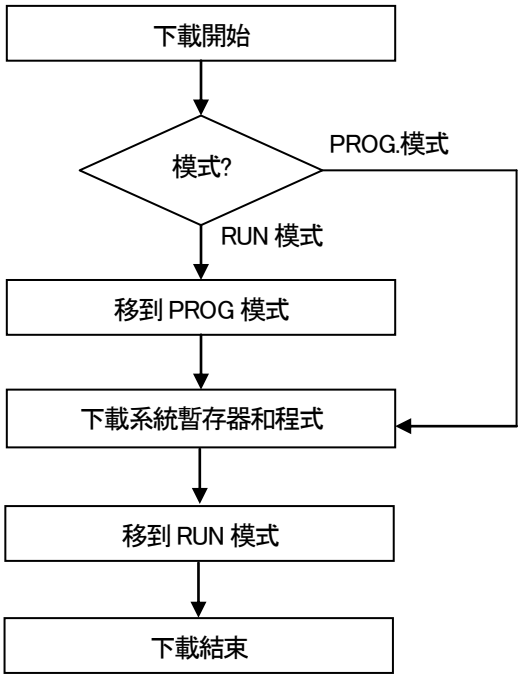
文件下載時、PLC 為 RUN 模式的時候

自動會變更為 PROG 模式後進行下載、下載結束之後、自動回到 RUN 模式。

文件下載時、PLC 為 PROG 模式的時候

進行下載、下載結束後、自動回到 RUN 模式。

自動切換模式有效時的流程圖



注意:PLC 必須為 REMOTE 模式。

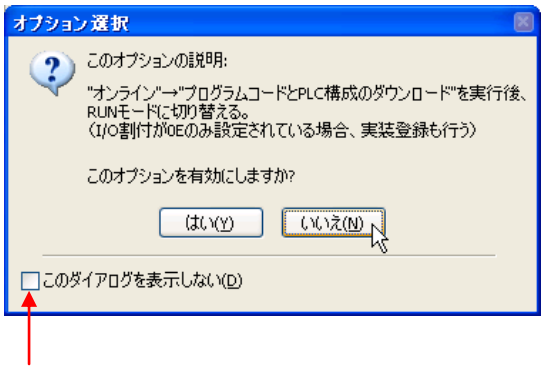
FP2SH、FP2 的場合

← DSW 要調到 REMOTE。

FP-X、FP Σ、FP0、FP-e 一般為 REMOTE 模式。

●選項的選擇(PLC 模式(RUN/PROG.)自動切換無效)

下載時、會顯示如下圖的對話框、選擇否 いいえ(N) 的時候。



之後不想進行自動切換時、請打勾。

程式下載時、PLC 為 RUN 模式時

進行選擇切換到 PROG.模式會顯示對話框、
切換到 PROG 模式後、進行下載。

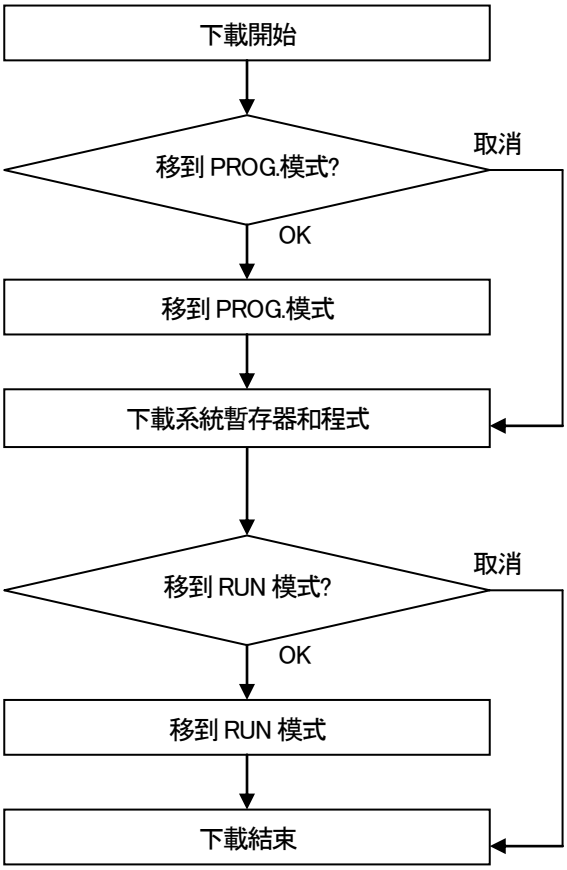
下載結束後、本次進行切換選擇到 RUN 模式時會顯示對話框。

Project 下載時、PLC 為 PROG 模式時

PROG 模式的話、進行下載。(不會移到 RUN 模式。)

模式自動切換無效時的流程圖

Project 下載時、PLC 為 RUN 模式時



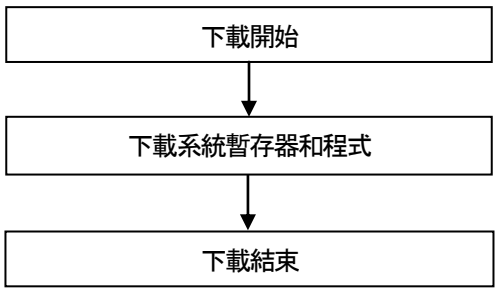
注意:PLC 必須為 REMOTE 模式。

FP2SH、FP2 的場合

← DSW 要調到 REMOTE。

FP-X、FP Σ、FP0、FP-e 經常為 REMOTE 模式。

文件下載時、PLC 為 PROG 模式時

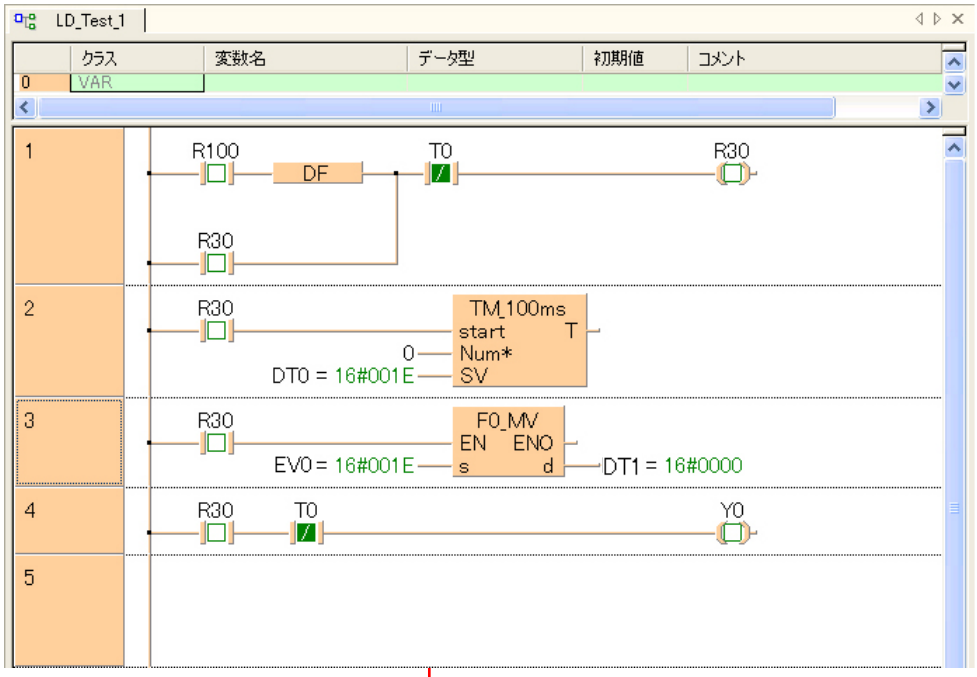


5-3 執行監控

如果下載正常結束、LD 畫面會移動監控模式。

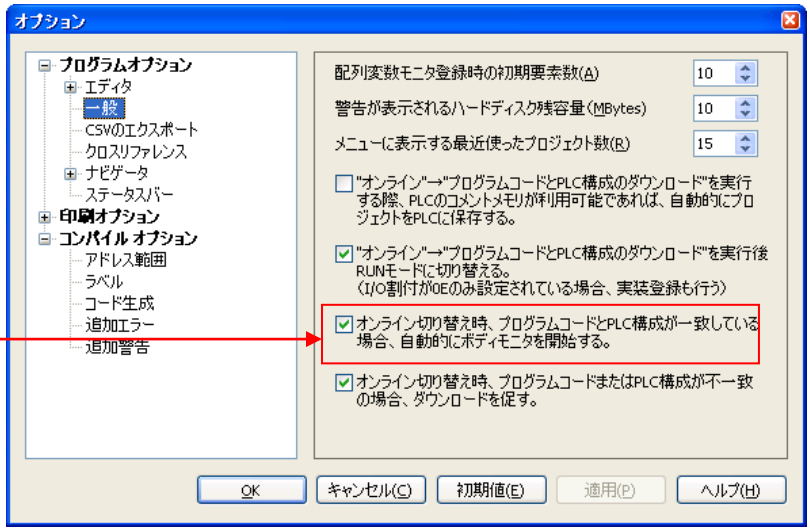
5-3-1 監控模式移動的順序

下載結束後、自動移到程式監視狀態。



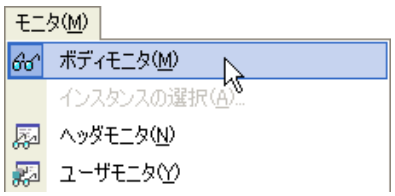
LD 畫面監控狀態

在 FPWIN Pro 的初期設定中、
設定為**「文件正確時、程式監控會自動開始」**。

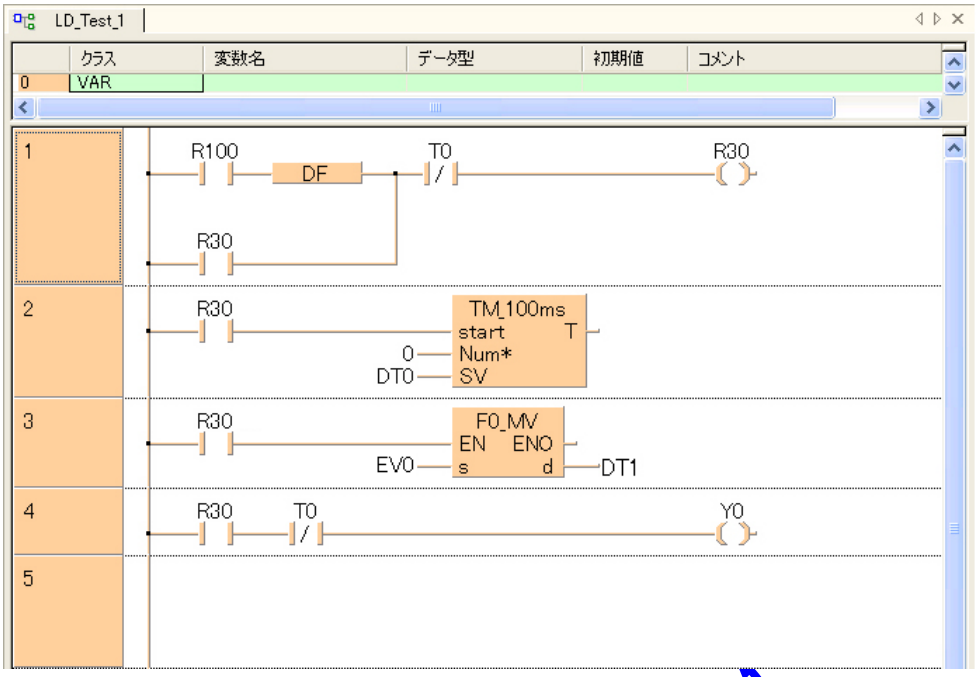


移到監控模式執行/停止狀態可選擇 **(選單)監控→ 程式監控**、

或是到工具列選擇 

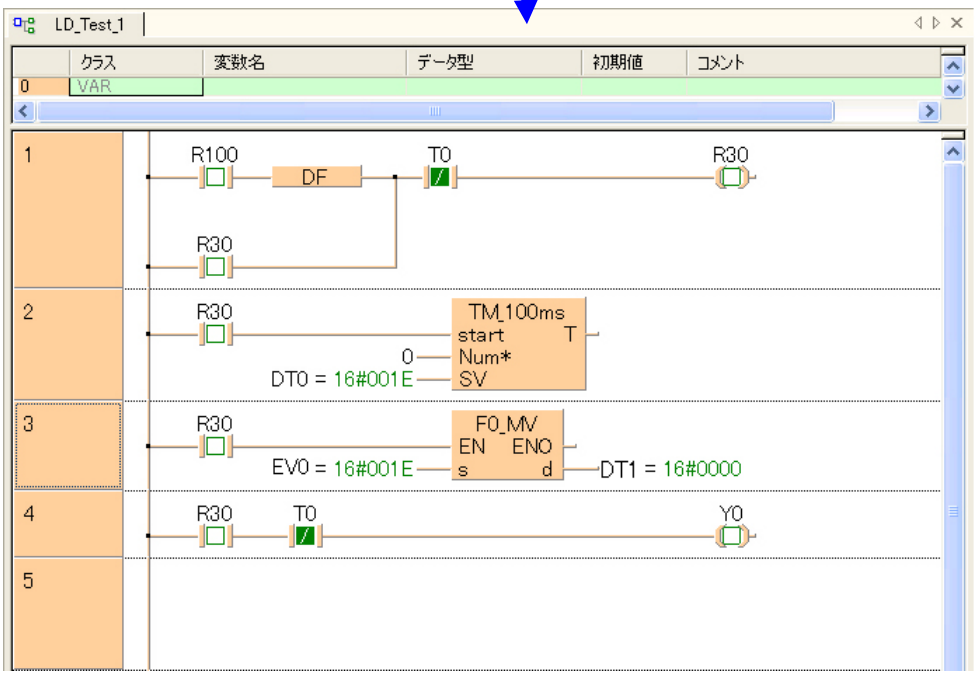


●LD 畫面監控停止狀態

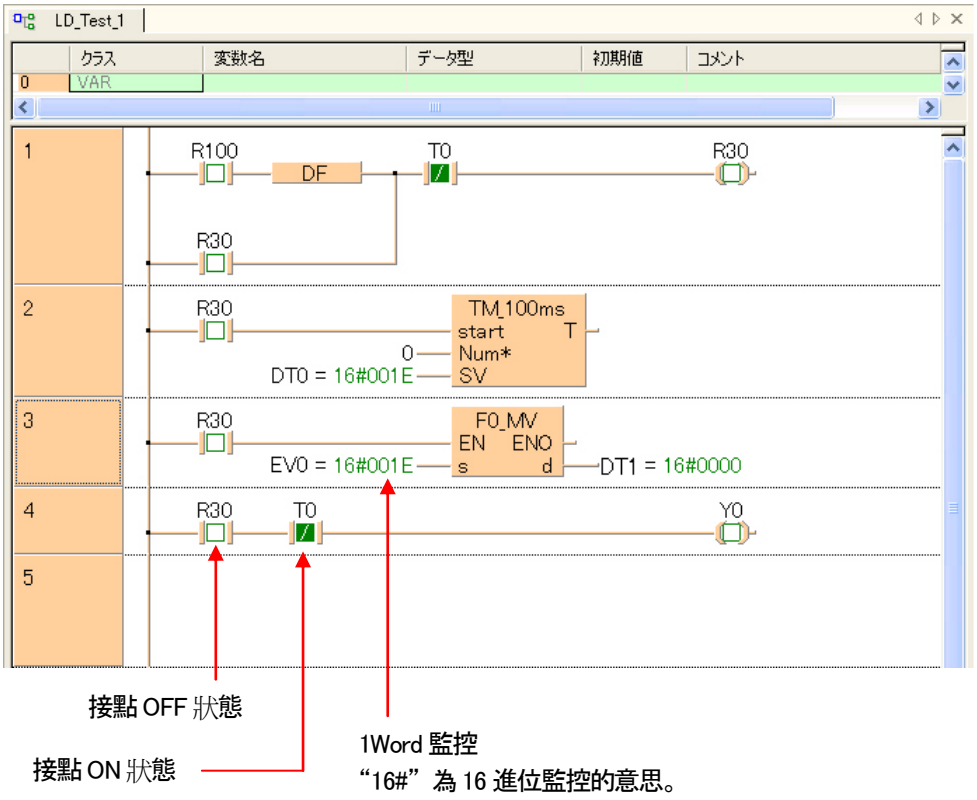


 的點擊可以切換。

●LD 畫面監控執行狀態



■數值的監控



接點的情況

顯示 ON 狀態為  、OFF 狀態為  。

WORD 元件的場合

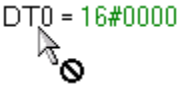
DT0 = 16#0000 如左、元件名稱右邊會有 16 進位顯示。

5-3-2 數值的變更

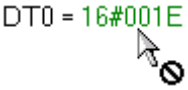
程式區的 WORD 元件點擊兩次就可以變更數值。

■操作順序

①將要變更數值的 WORD 元件點擊兩次。

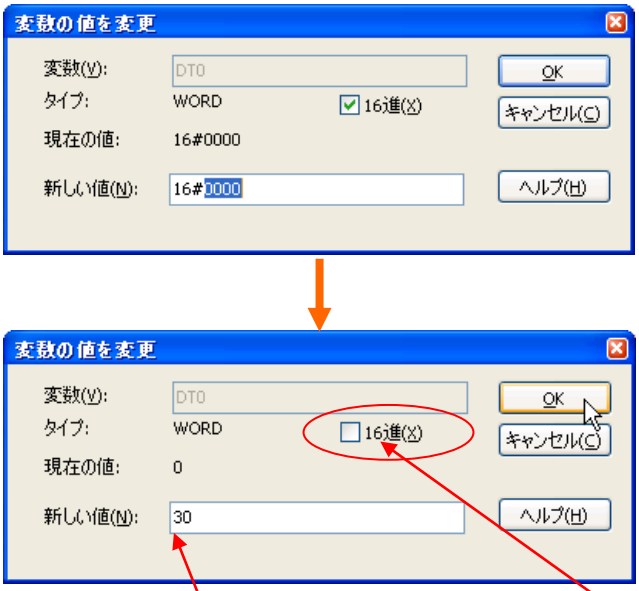


在此變更 DT0 的數值。



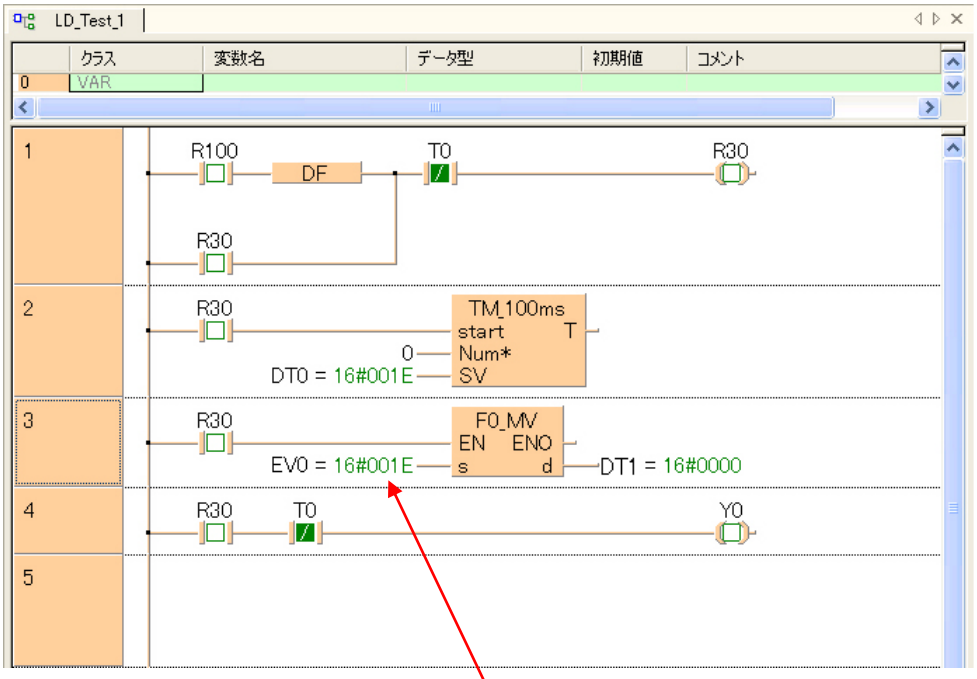
數值點兩次也可以。

②「變更變數的數值」因為會顯示對話框、數值輸入按下「OK」鍵。



在此輸入“30”(10 進位)。

打勾消除就成為「10 進位」。

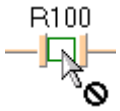


數值輸入。(10 進位:30、16 進位:1E)

點選程式區的接點兩次、可以變更 ON/OFF 的狀態。

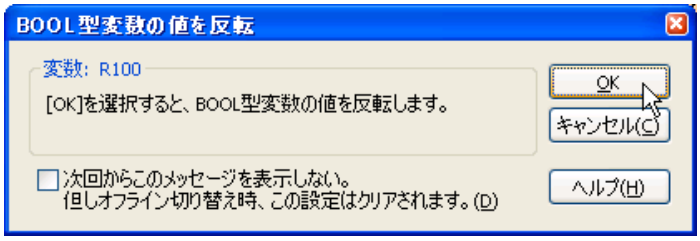
■操作順序

①點選要變更 ON/OFF 狀態的接點兩次。



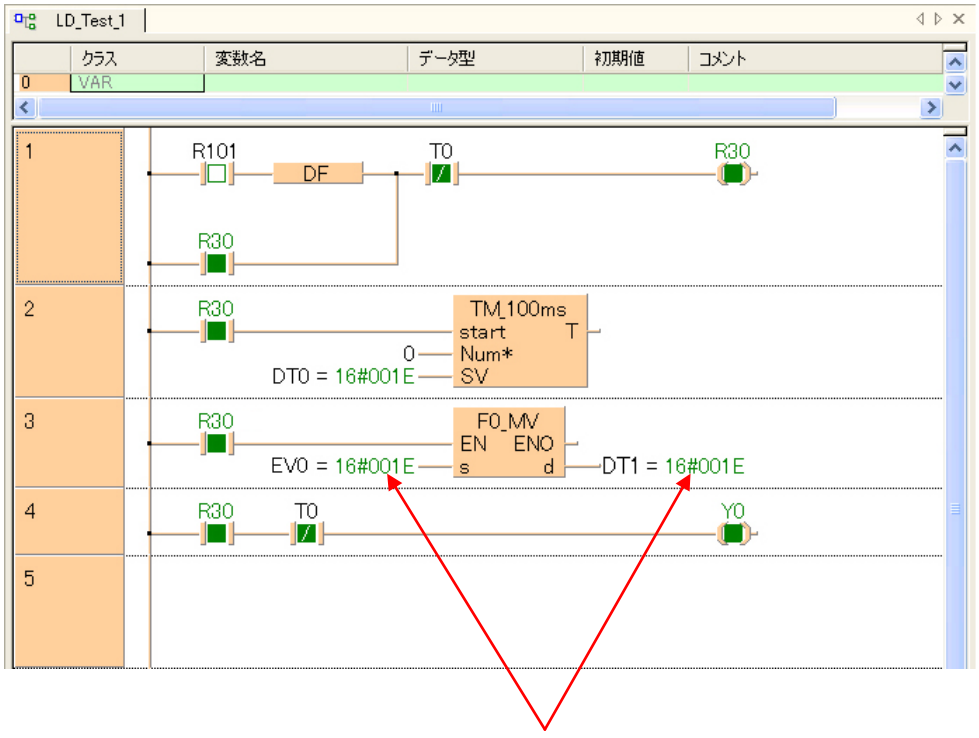
在此讓接點 R100 使其 ON。

②會顯示「反轉 BOOL 變數的數值」的對話框、按下「OK」鍵。



「下次不顯示此訊息」打勾、到 Online 模式解除前
此對話框不會再顯示。

③點選「OK」鍵、接點反轉(在此 R100 會 OFF→ON)。



計時器 0 的經過值會監控。

以上階梯圖、在 DT0 輸入“30”、
會成為「當 R100 ON 時、3 秒後 Y0 會 ON」的程式。

請確認 DT0 的數值何時會變。

5-4 進行 Online 編集

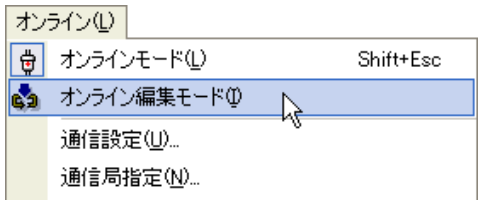
5-4-1 Online 編集時的限制事項

FPWIN Pro 和 PLC 在 Online 中、可以編集程式。
PLC 在 RUN 模式也可以。
此功能相當於 FPWIN GR 的「RUN 中的改寫」。

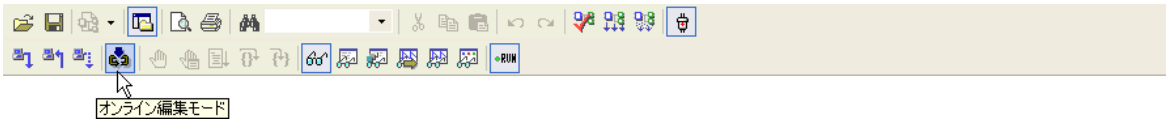
Online 模式時、一次可以下載程式到 PLC 本體最多可達 118 STEP。

5-4-2 Online 編集的順序

LD 編集畫面 Online 中、選擇**(選單)Online → Online 編集模式**、
選擇工具列的  按鍵。



移到 Online 編集模式、可以變更程式區上的各元件(接點和線圈等)。
並且、可以追加新的 Network、階梯圖。



點擊可以移到 Online 編集模式

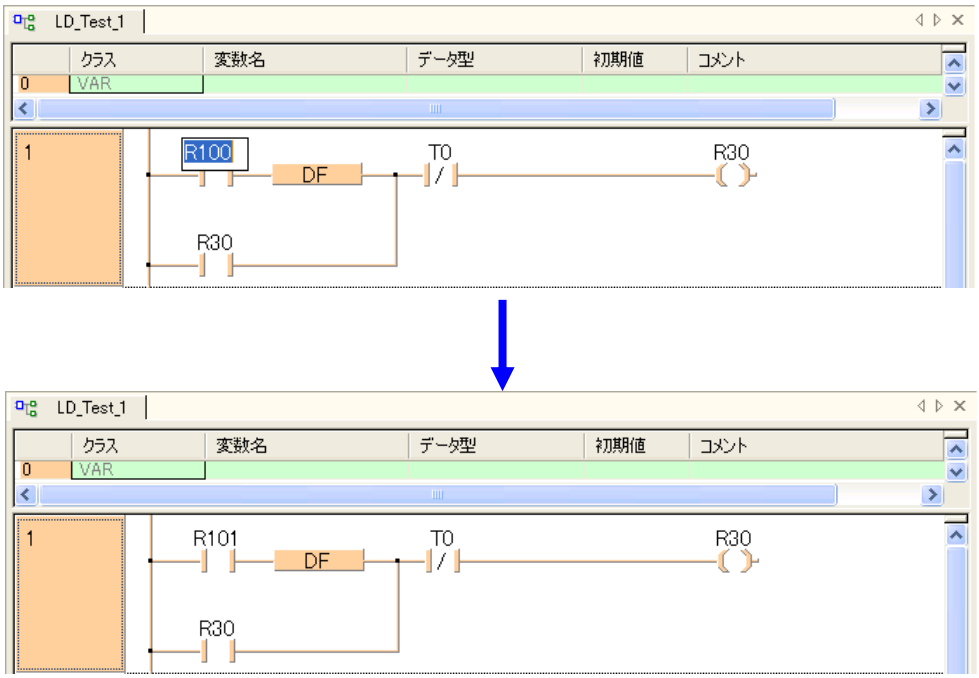
顯示變更各元件的小圖示。

編寫程式 Online 看看。

<例>
「R100」變更為「R101」的情況

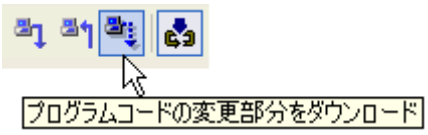
①  按鍵點擊一下、移到 Online 編集模式。

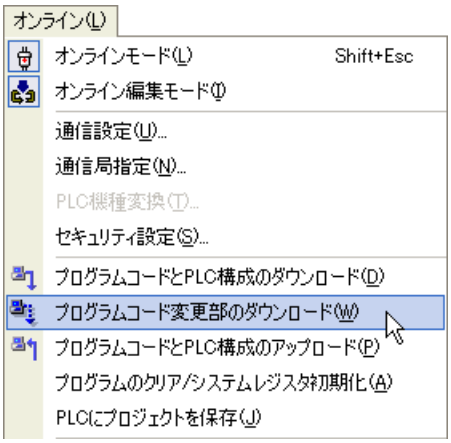
②R100 的部分用滑鼠點選、成為可變更的狀態、輸入 R101。



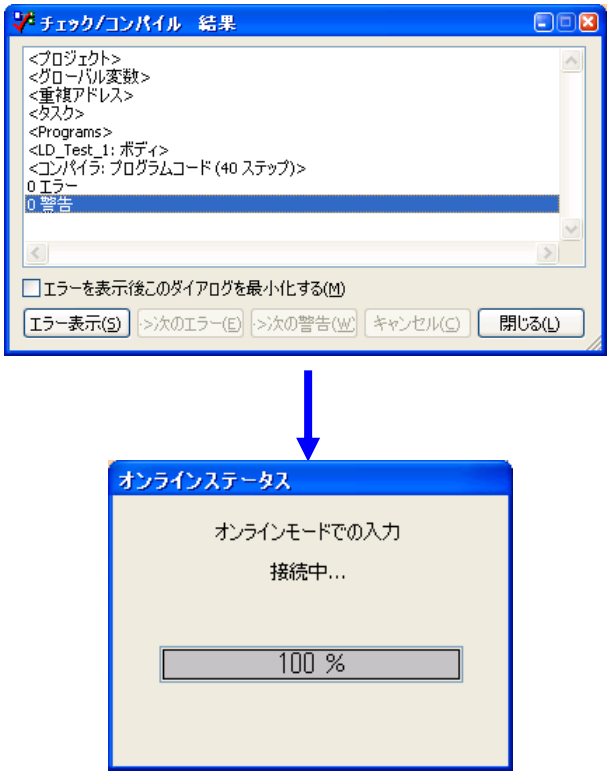
③變更結束後、選擇**(選單)Online → 下載變更地方**、

選擇工具列的  按鍵。

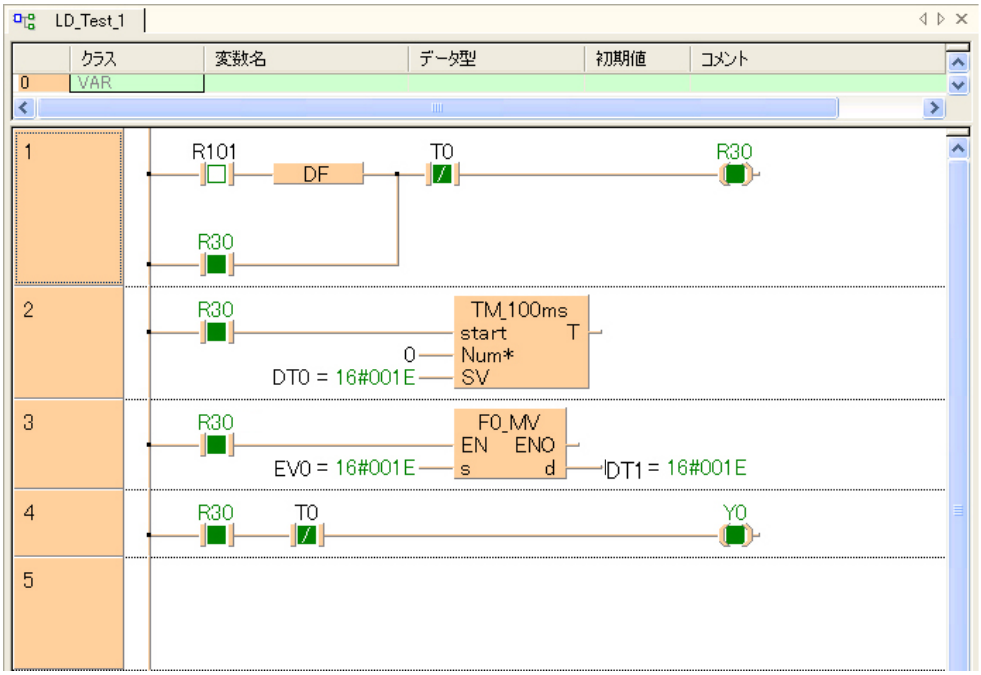




④FPWIN Pro 把變更的部分再編譯、下載到 PLC。



⑤下載結束後的畫面。



⑥解除 Online 編集模式時、必須再一次到選單的「Online 編集模式」

或是選擇工具列的  按鍵。

第6章

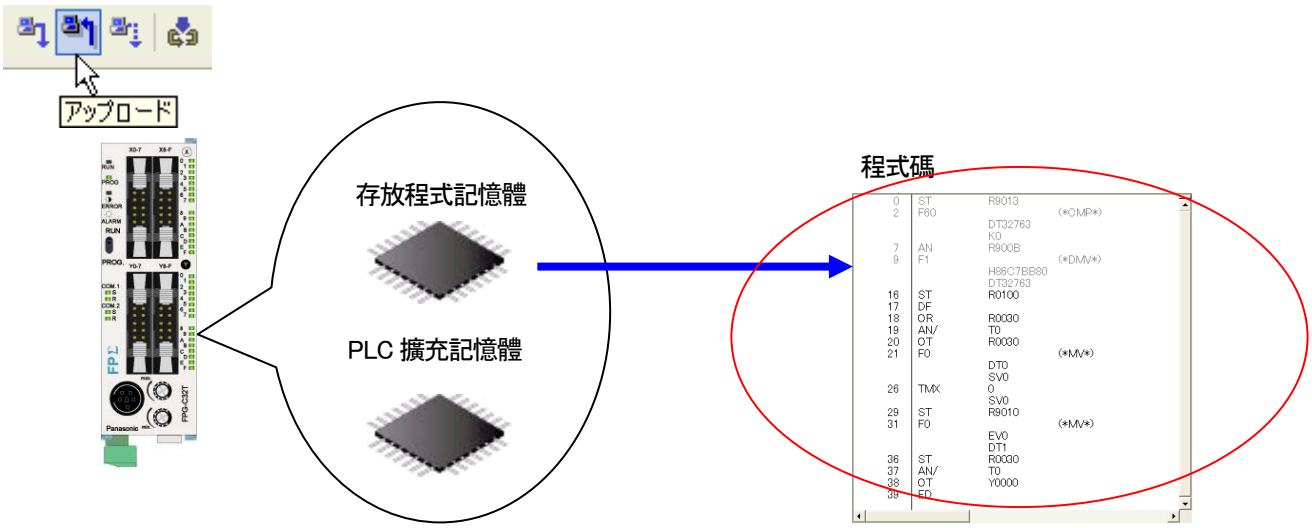
從 PLC 上傳程式

6-1 從 FPWIN Pro 上傳程式

用 FPWIN Pro 執行上傳程式時、有以下 2 個方法。

■程式碼的上傳

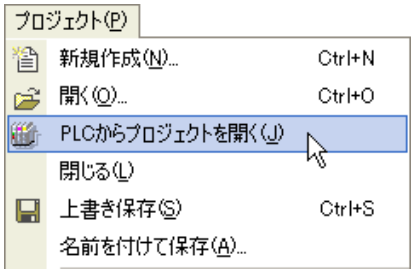
上傳現在 PLC 內的程式碼。
被上傳的命令碼、會顯示為機械語言而非階梯圖格式。
（無法變更為階梯圖以及其他格式。）
換句話說、即使上傳也不會重現 POU。
程式碼的上傳、支援全部的 PLC。

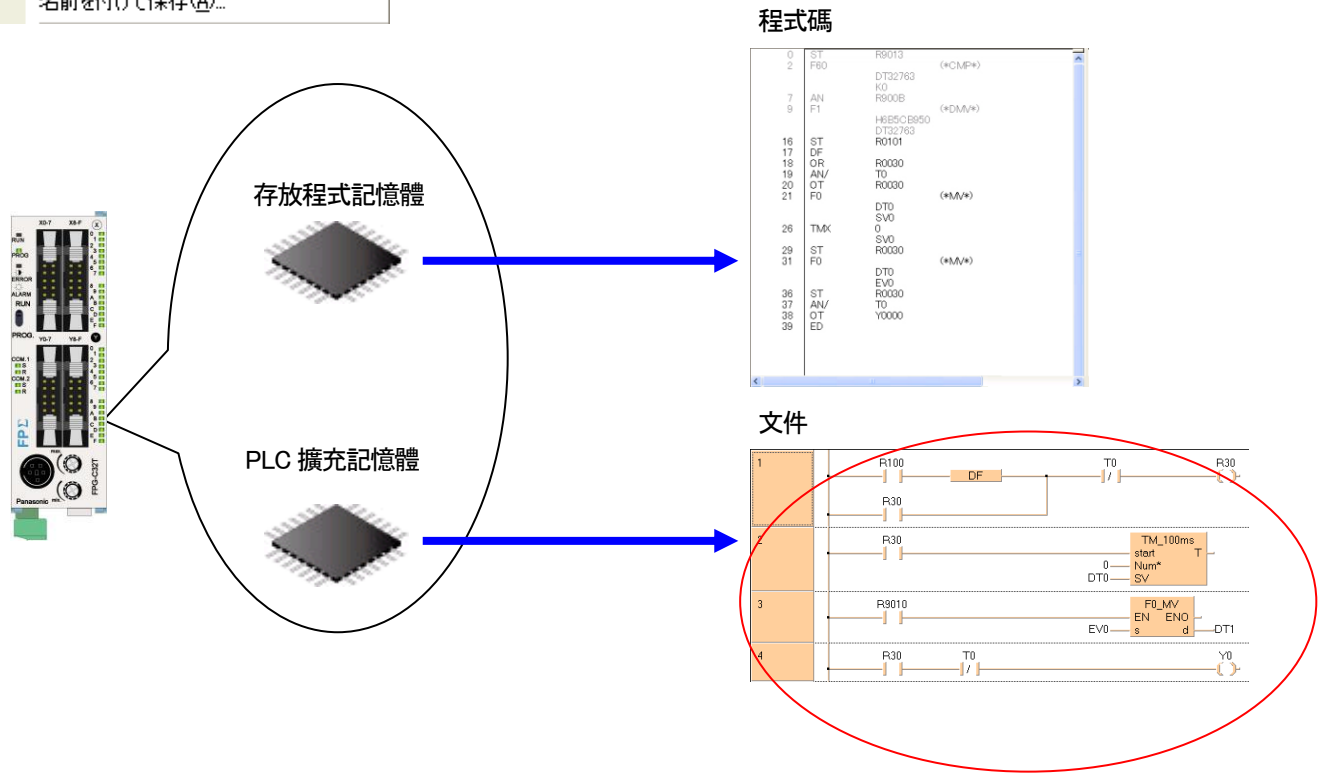


■Project 的上傳

上傳現在 PLC 內所存的程式碼和存在 PLC 的擴充記憶體的 Project 以及變數的資訊等。
此時和程式碼的上傳有所不同、全部的 POU（階梯圖等）會重現。
可以上傳 Project 的 PLC 為以下類型。

- FP Σ
- FP-X
- FP2（有安裝選配的註解儲存器“AFP2201, AFP2202, AFP2203”時）
- FP2SH





6-2 執行上傳程式

6-2-1 上傳程式

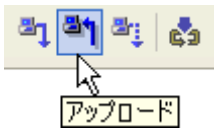
程式上傳雖然為 PLC 的機械碼、

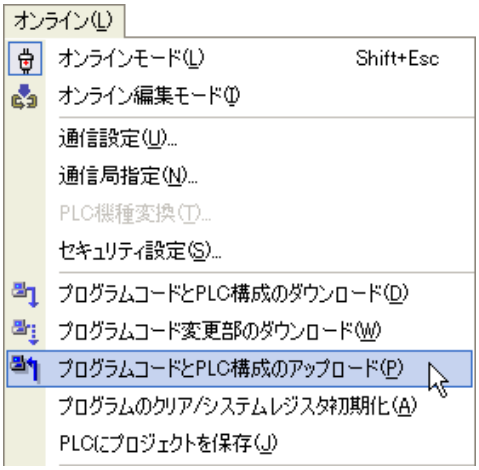
①程式碼的上傳(只有機械碼上傳)
②程式碼的上傳和轉換(上傳機械碼、變換為 LD 語言畫面)

可以選擇以上 2 種。

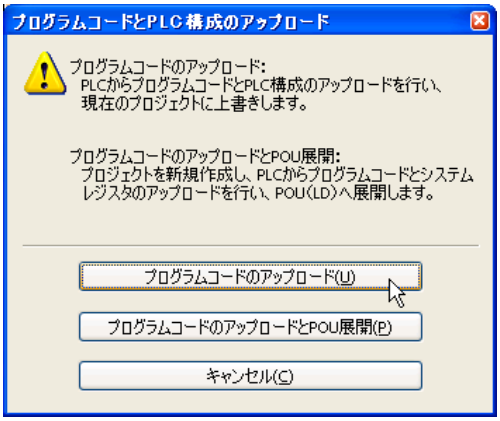
程式的上傳在 FPWIN Pro 和 PLC 的 Online 狀態下
選擇(**選擇)Online → 程式碼的上傳**、

選擇工具列的  執行。

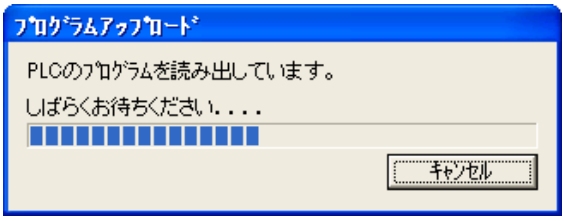




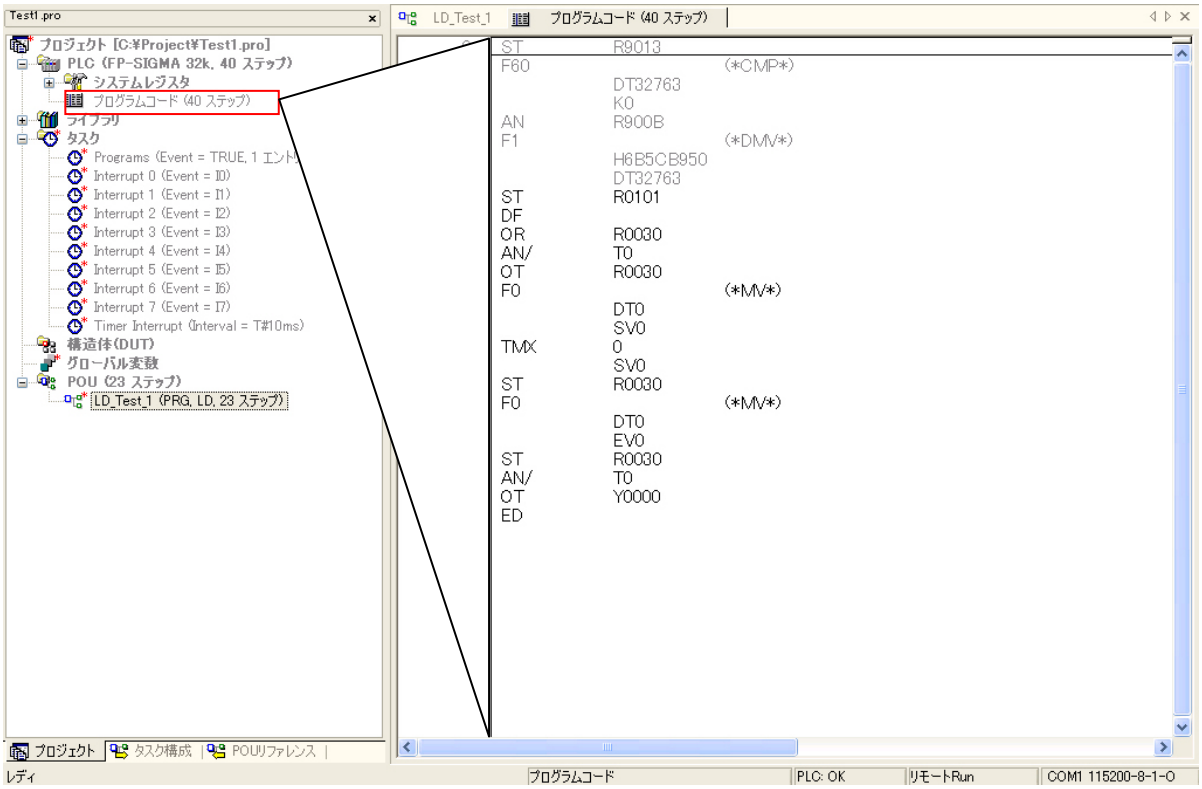
①程式的上傳



開始上傳。



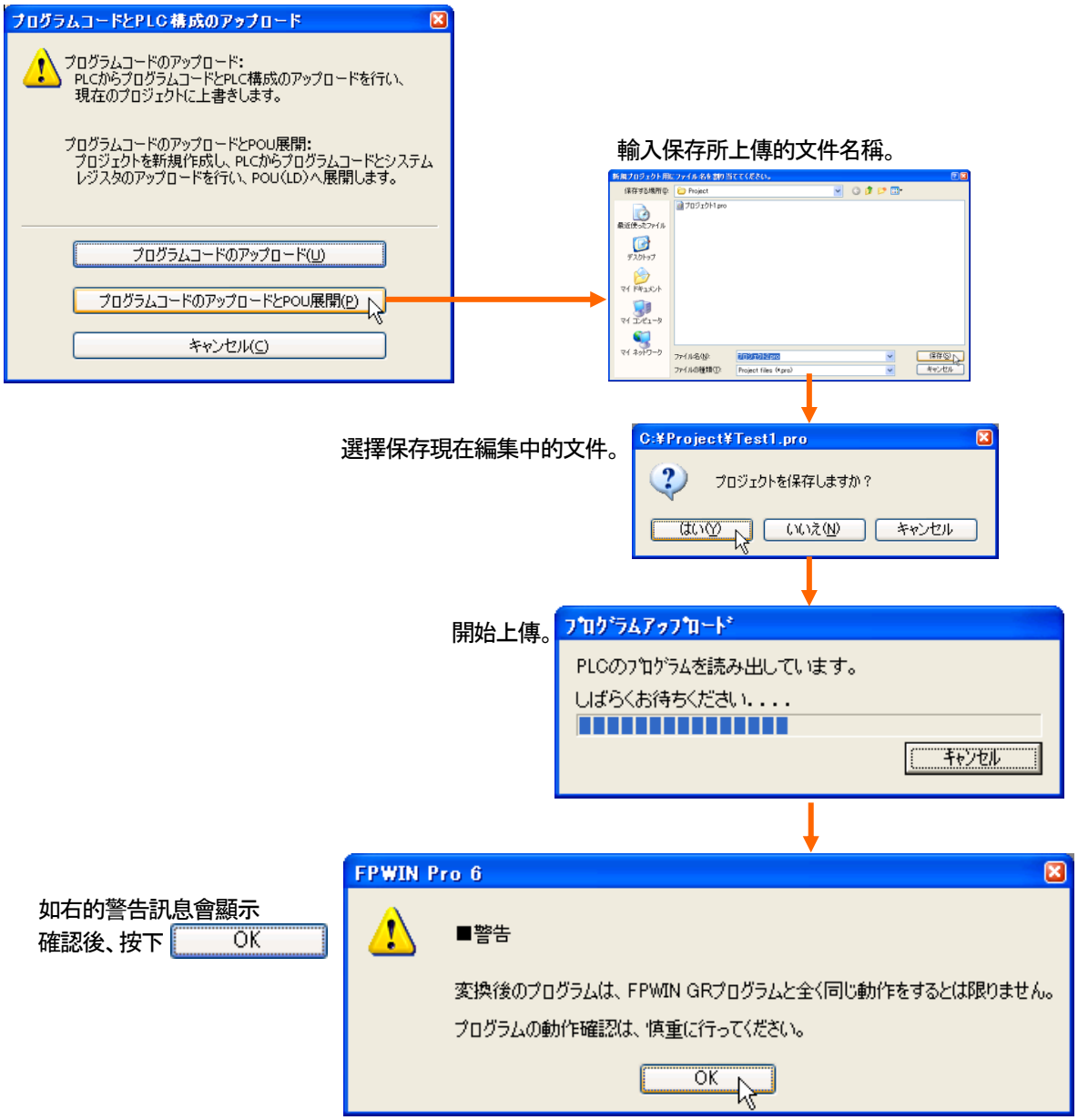
上傳結束後、可以確認所上傳的程式為機械碼而非階梯圖格式。



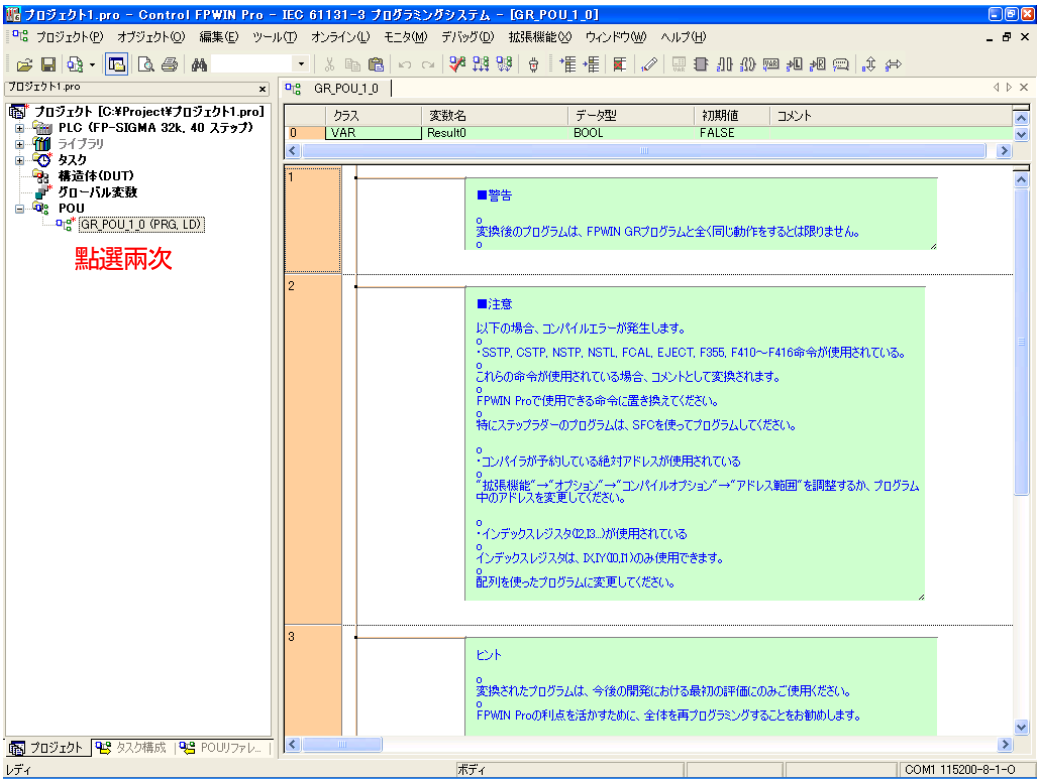
●注意

所上傳的內容、只有在這裡會反映出現。
不會反映到 POU 的程式區。

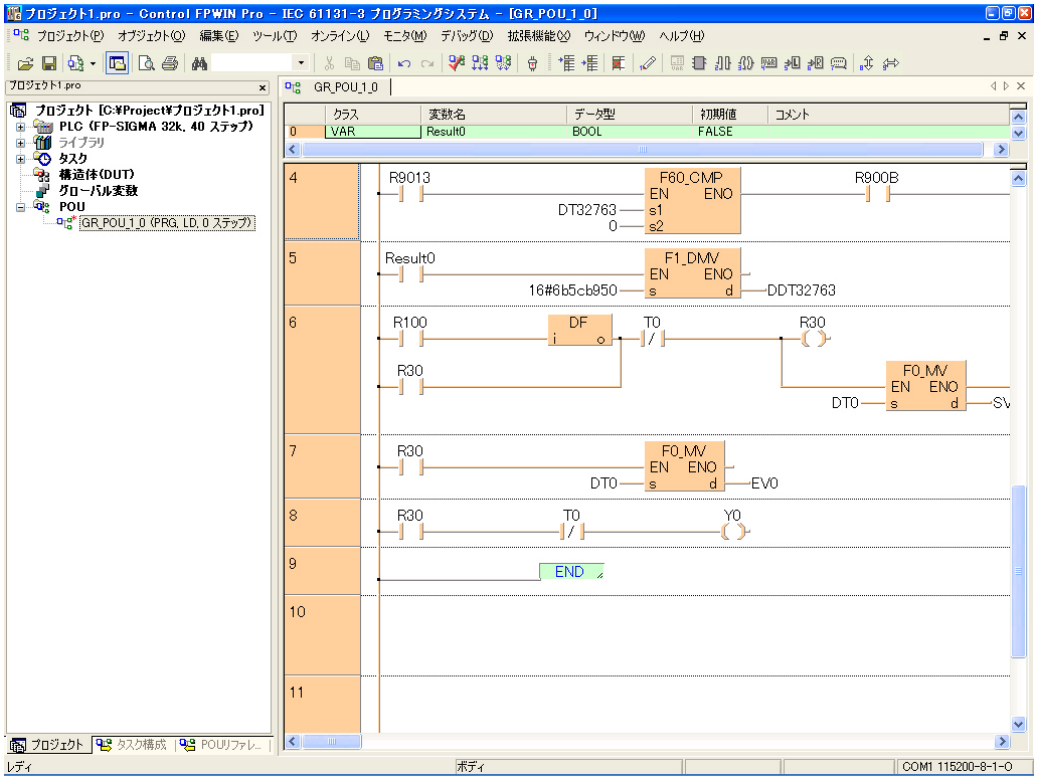
②程式碼上傳和轉換


プログラムコードとPLC構成のアップロード
プログラムコードのアップロード:
PLCからプログラムコードとPLC構成のアップロードを行い、現在のプロジェクトに上書きします。
プログラムコードのアップロードとPOU展開:
プロジェクトを新規作成し、PLCからプログラムコードとシステムレジスタのアップロードを行い、POU(LD)へ展開します。
プログラムコードのアップロード(U)
プログラムコードのアップロードとPOU展開(P)
キャンセル(C)
輸入保存所上傳的文件名稱。
選擇保存現在編集中的文件。
C:¥Project¥Test1.pro
プロジェクトを保存しますか？
はい(Y)
いいえ(N)
キャンセル
開始上傳。
ﾌﾟﾛｸﾞﾗﾑｱｯﾌﾟﾛｰﾄﾞ
PLCのﾌﾟﾛｸﾞﾗﾑを読み出しています。
しばらくお待ちください. . . .
キャンセル
如右的警告訊息會顯示
確認後、按下
OK
FPWIN Pro 6
■警告
変換後のプログラムは、FPWIN GRプログラムと全く同じ動作をするとは限りません。
プログラムの動作確認は、慎重に行ってください。
OK

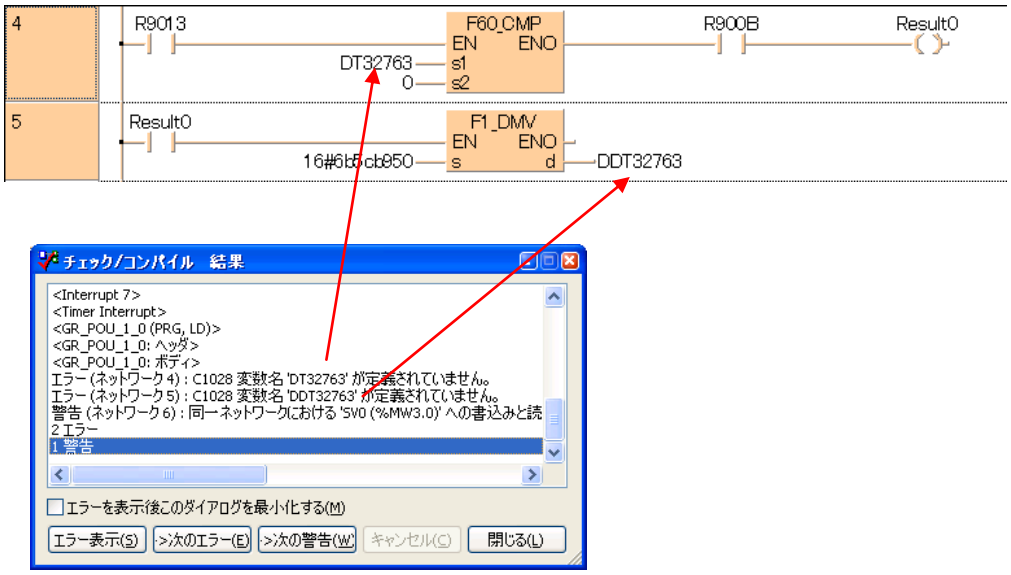
上傳作業結束後、點擊 POU 的程式兩次會顯示出程式。
（在程式的開頭部分會自動附上警告、注意的文字。）



拉動視窗右邊的拉 BAR。

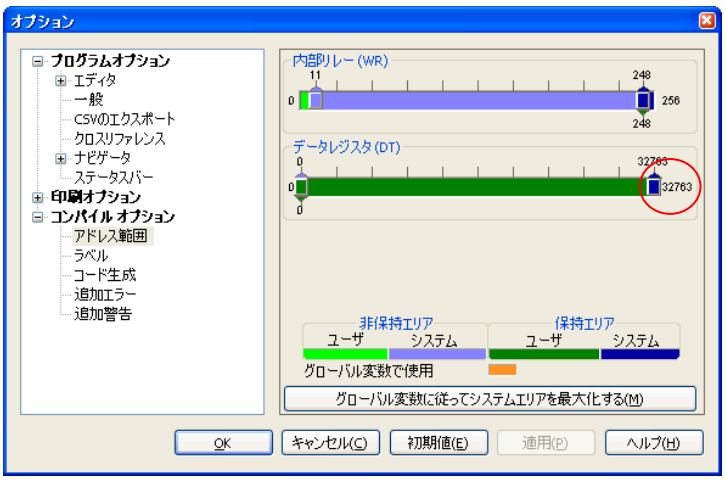


在此編譯上傳的 Project 後、會如下圖發生錯誤。



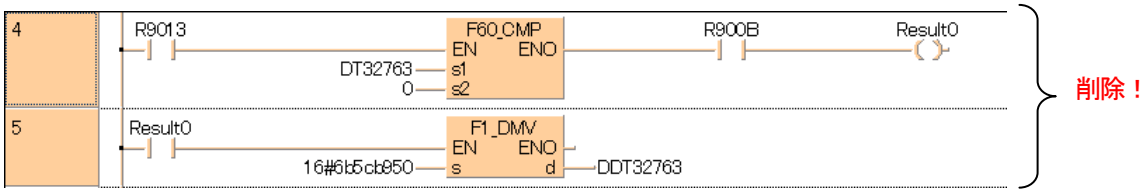
發現錯誤的地方、會知道在“DT32763”和“DDT32763”的元件發生錯誤。

這個“DT32763”和“DDT32763”在 FPWIN Pro 為使用者區域的未開放的區域。



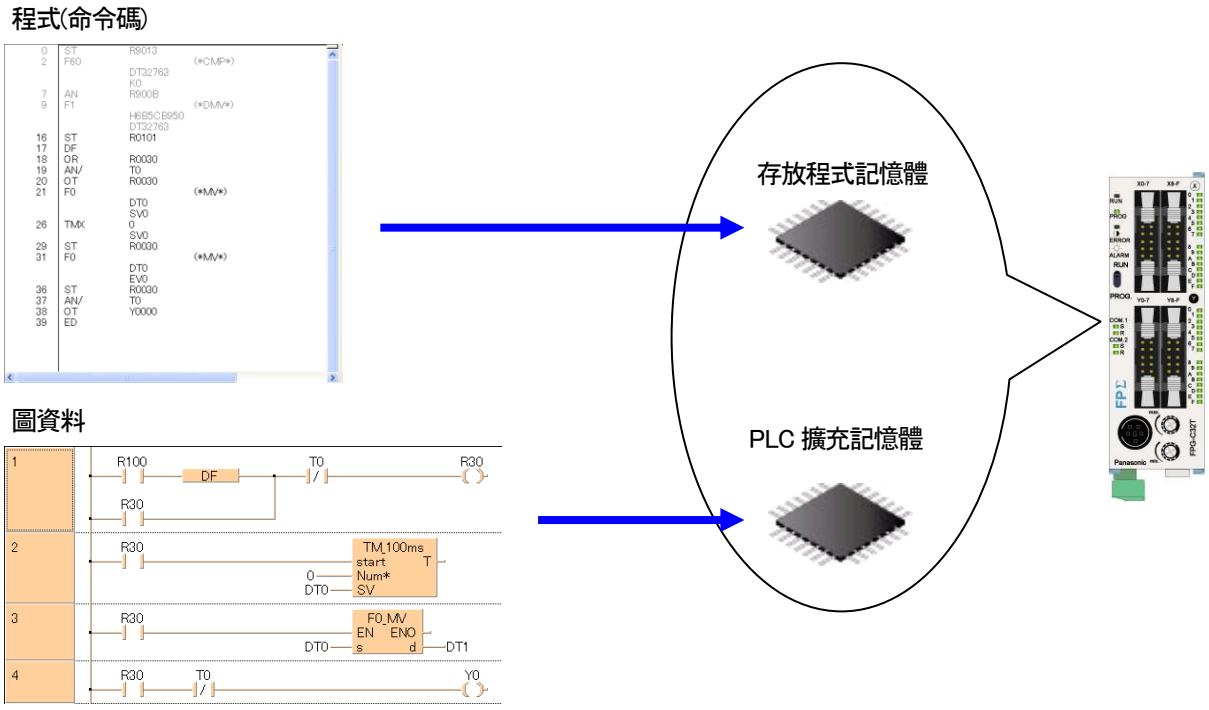
用 FPWIN Pro 所下載的程式、編輯器會自動使用“DT32763”。
其程式上傳會轉換時捉取到無法使用的“DT32763”。
當然這次編譯時、就會在“DT32763”發生編譯錯誤。

追加程式等、再次下載時、請削除這 2 個 Network。
編輯器在編輯時會再次產生這 2 個 Network。



6-2-2 上傳 Project

上傳 Project 時、現在 PLC 內的程式(命令碼)和 PLC 擴充記憶體裡的階梯圖的圖資料等必須先下載。

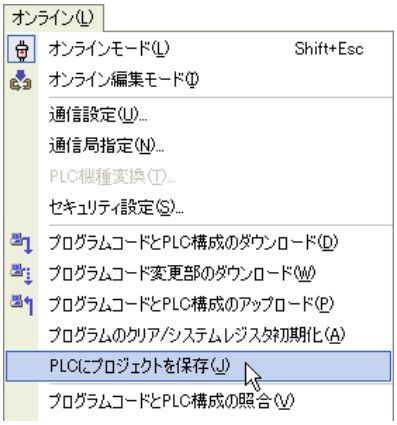


文件可以上傳的 PLC 有以下類型。

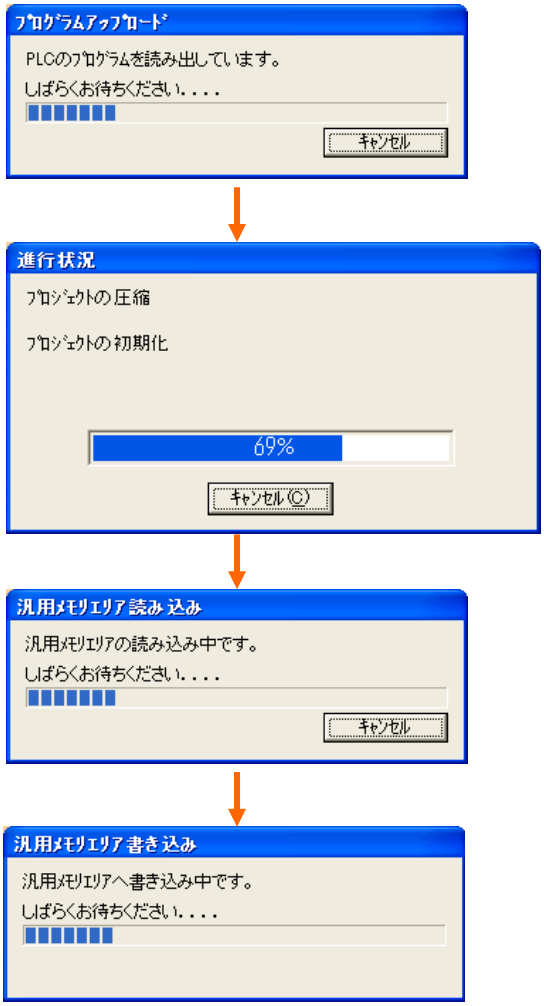
- FP Σ
- FP-X
- FP2(有安裝選配的註解儲存器"AFP2201, AFP2202, AFP2203"時)
- FP2SH

■Project 下載

Project 下載要在 Online 狀態下選擇執行 **(選單)Online → 保存文件到 PLC**。



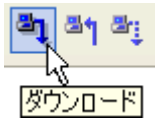
下圖對話況會連續顯示、進行文件下載。



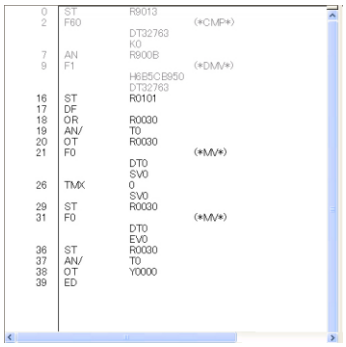
●備考(下載整理)

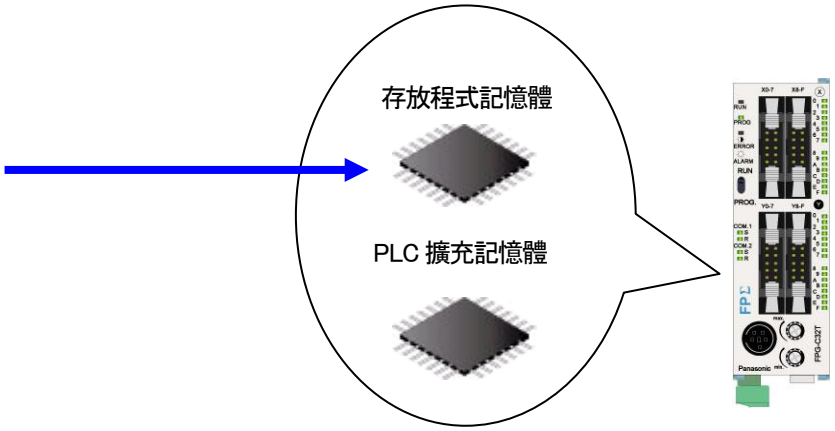
下載(FPWIN Pro 初期設定時)

只有下在程式碼。



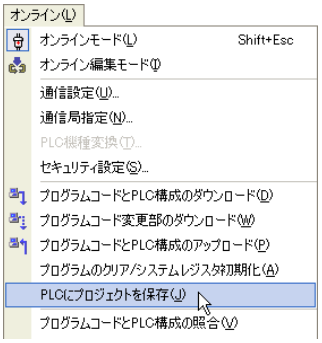
程式碼



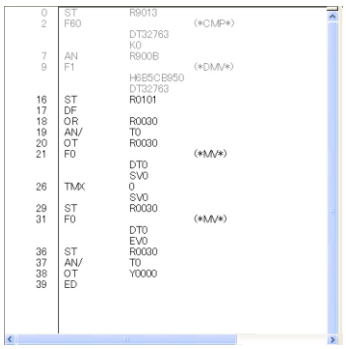


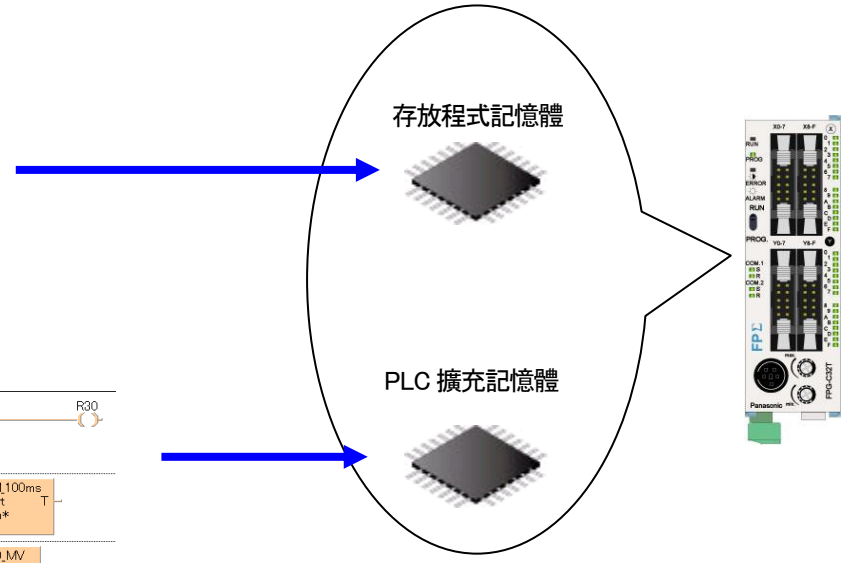
下載(Project 保存時)

下載程式碼和 Project。

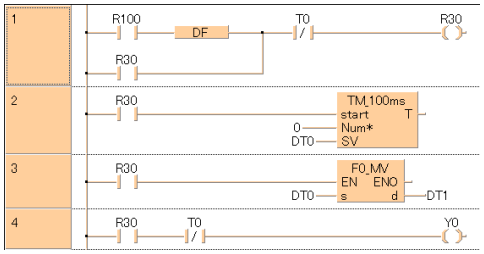


程式碼





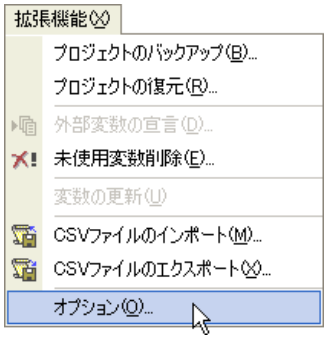
Project

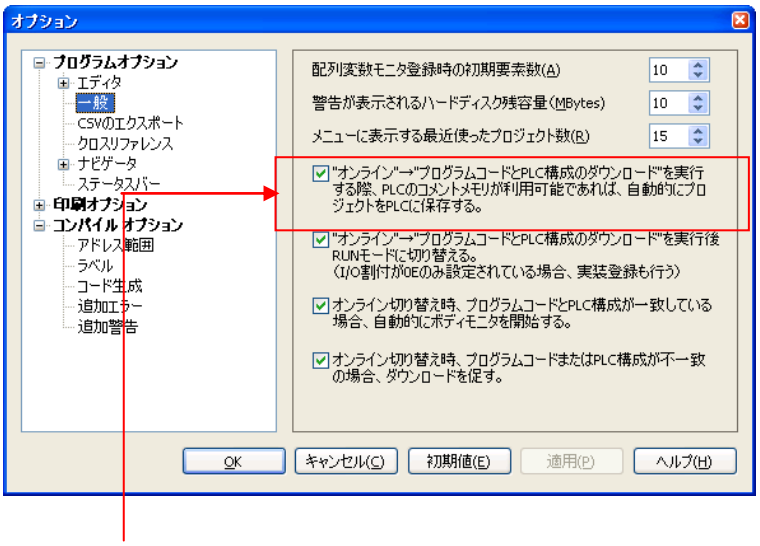


下載(FPWIN Pro 初始設定時)

可以預先設定 Project 下載。

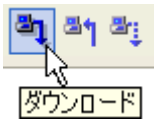
選擇執行**(選單)擴充功能 → 選項**。



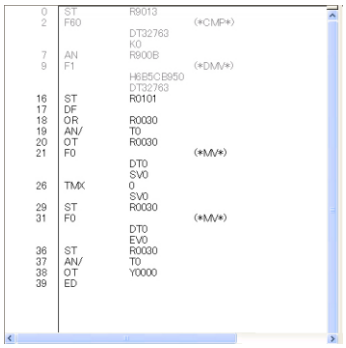


這部分打勾。

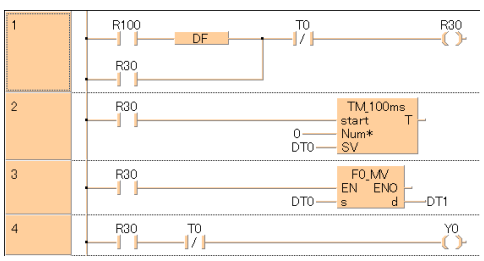
在此點選  的小圖示、PLC 內的程式碼會同時下載和 PLC 擴充記憶體的 Project。

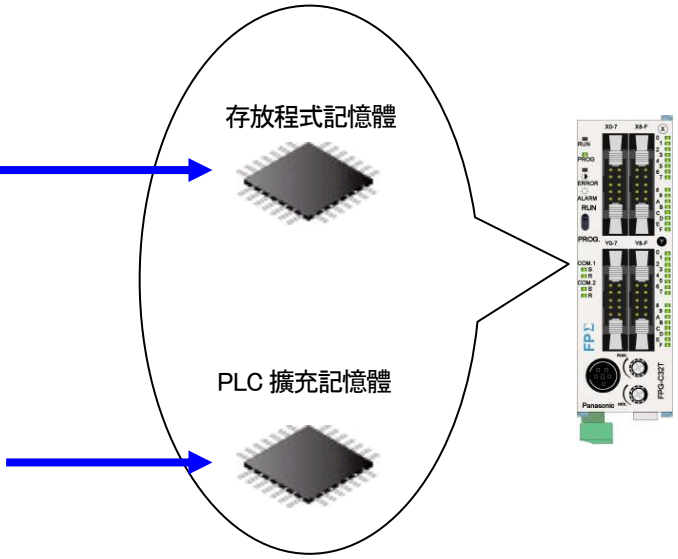


程式碼



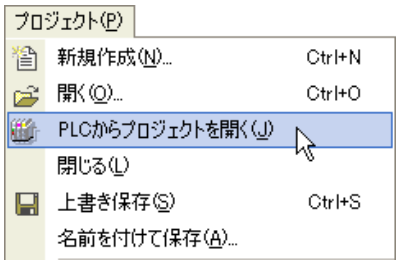
文件



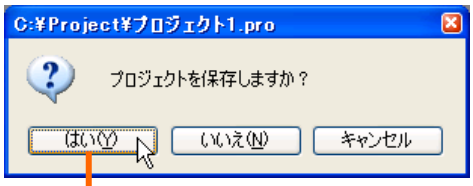


■上傳 Project

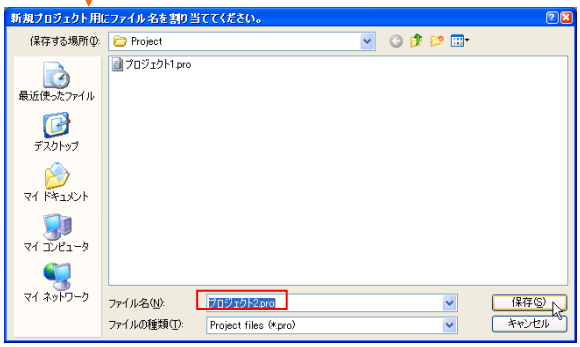
Project 上傳時、請打開選擇**(選單)文件 → 執行從 PLC 開啟文件**。



會顯示出現在編集中的 Project 是否保存、選擇「是」。

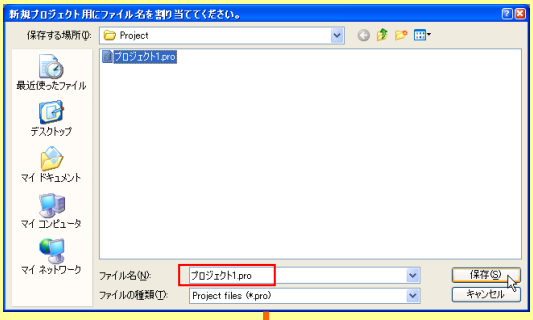


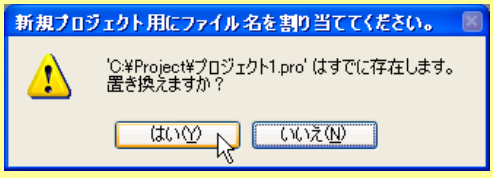
將現在編集中的文件保存到其他地方時、請輸入¥Project 名稱。



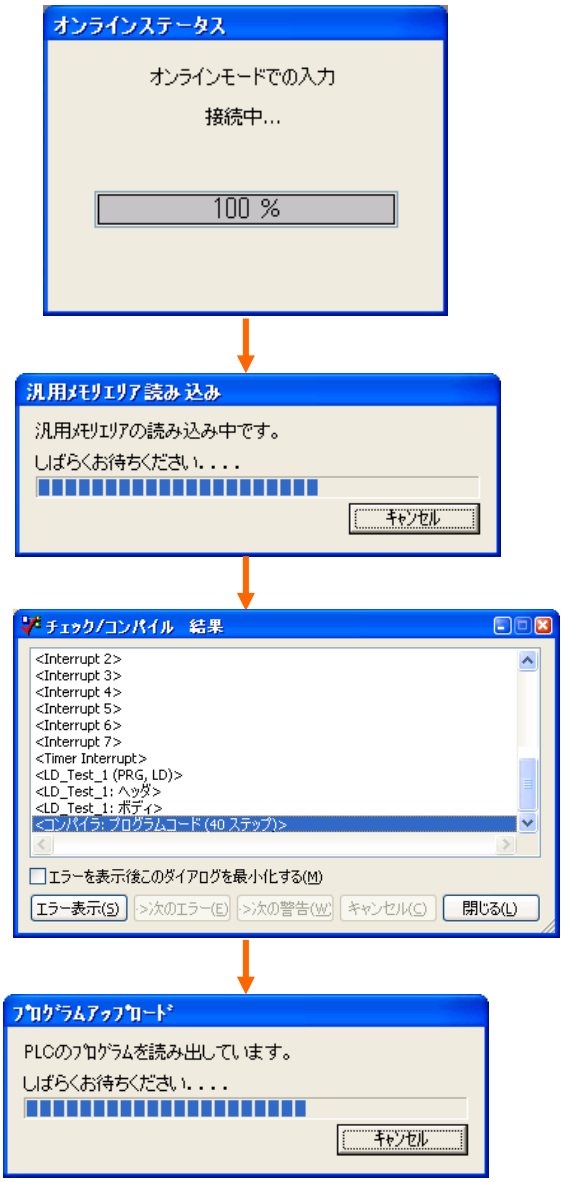
※本次請輸入其他名稱。

將覆蓋現在編集中的 Project、
編集中的 Project 名和舊 Project 名稱輸入相同名稱。

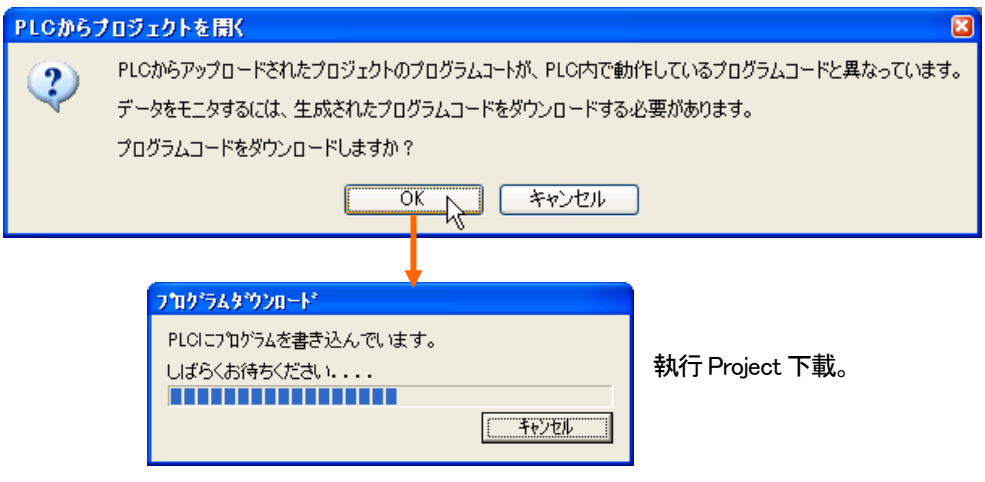




下圖對話框會連續顯示、進行 Project 上傳。

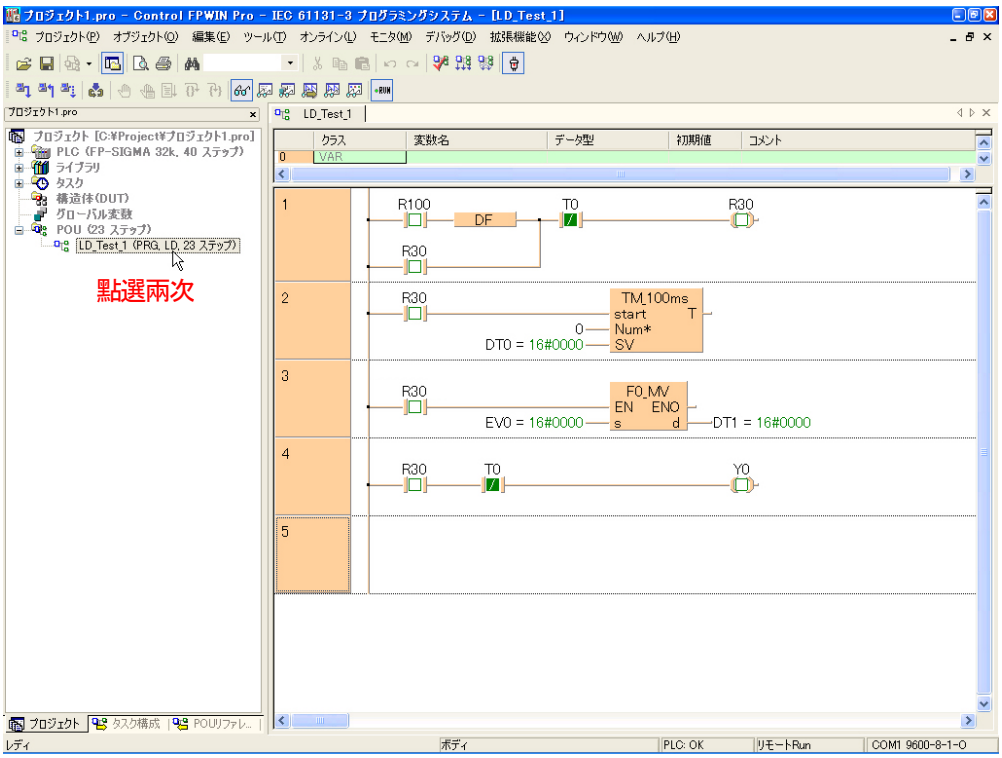


從 PLC 上傳的程式和現在 PLC 內的程式不同時、會顯示下圖的視窗。
(包含系統暫存器)
產生的程式下載到 PLC 時,按下「OK」按鍵。



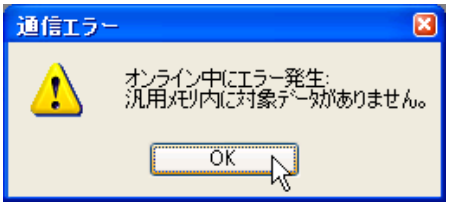
執行 Project 下載。

上傳作業結束後、POU 的程式點擊兩次、會顯示程式。



●備註

Project 未保存在 PLC 的状態、就進行 Project 上傳。



上傳中會顯示如上圖的對話框 時、會中止上傳。

第7章

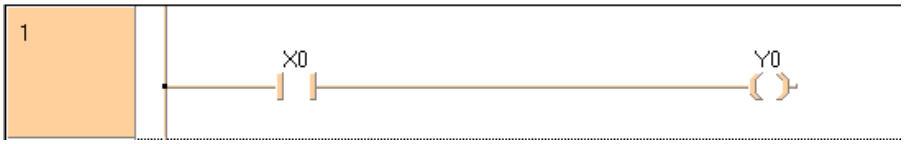
使用變數撰寫程式

7-1 概要

■何謂變數

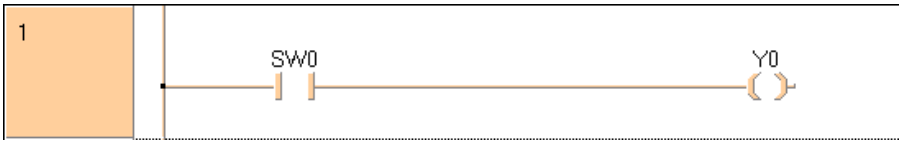
所謂變數、就是為了識別在 PLC 內部的輸出入、PLC 內部的元件以及號碼(DT 和 WR 等)的名稱。
變數可在程式中使用。

例如、使用 No.0 確認開關是否有輸入。以下的記述為從輸入 No.0 讀取數值時的
LD(階梯圖)。



不過此記述方法卻有以下 2 個缺點。
・程式會越變越複雜、個別的輸出入功能以及內部記憶體的分配內容越難以記住。
・位址的變更以及 PLC 機種變更時、程式必須大範圍的進行修改。

如果使用 Control FPWIN Pro 就不會發生這樣的問題。
例如、宣告變數的 SW0 分配到 X0 時合、程式如以下所記述。



■全域變數和區域變數

全域變數和區域變數的差異如下。

全域變數: 全域變數、在以下時使用。
・分配變數到外部輸出入時
・用 Project 內的複數 POU 對共通使用的變數時
・和外部機器的通信用、將變數分配到固定 PLC 內部記憶體位址時

區域變數: 區域變數、只有在 1 個 POU 內有效。
對於變數內部記憶體的分配、編譯時會自動執行。
區域變數、在各 POU 的 Header 進行宣告。

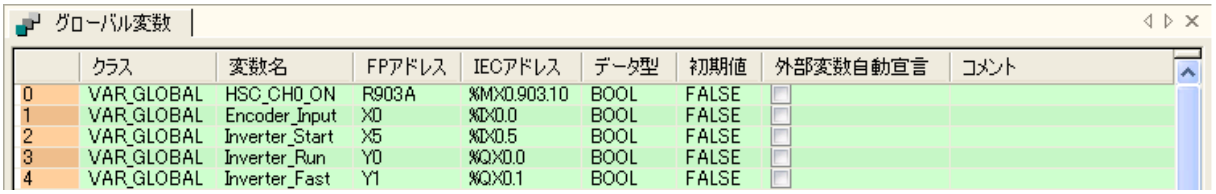
為了掌握變數全體的動作、在 cross-reference 裡, 變數的宣告和屬性的全部可以一覽顯示。

7-2 全域變數

全域變數在 Project 引導的全域變數明細表中宣告。
可以分配外部輸出入或是 PLC 內記憶體位址的是全域變數。
區域變數是無法指定位址。

7-2-1 全域變數的宣告內容

進行變數宣告之前、簡單說明全域變數明細表上的各個區域。


グローバル変数

	クラス	変数名	FPアドレス	IECアドレス	データ型	初期値	外部変数自動宣言	コメント
0	VAR_GLOBAL	HSC_CH0_ON	R903A	%MX0.903.10	BOOL	FALSE		
1	VAR_GLOBAL	Encoder_Input	X0	%IX0.0	BOOL	FALSE		
2	VAR_GLOBAL	Inverter_Start	X5	%IX0.5	BOOL	FALSE		
3	VAR_GLOBAL	Inverter_Run	Y0	%QX0.0	BOOL	FALSE		
4	VAR_GLOBAL	Inverter_Fast	Y1	%QX0.1	BOOL	FALSE		

■用全域變數明細表可以指定的內容

等級

全域變數分為 3 個變數種類。

VAR_GLOBAL: **非保持型的全域變數。**
電源 OFF 時或是 PROG 模式變更時無法保持數值。
RUN 模式切換時、[初期值]會被設定成所指定的內容。

VAR_GLOBAL_RETAIN: **保持型的全域變數。**
電源 OFF 時或是 PROG 模式變更時會保持數值。

VAR_GLOBAL_CONSTANT: **作為定數所使用的全域變數。**
VAR_GLOBAL_CONSTANT 無法指定位址。

變數名

程式內所使用的簡稱。變數名稱不可用數字為字首。

FP 位址以及 IEC 位址

FP 位址為變數所分配到 PLC 的元件以及號碼(X0, Y0, DT0 等)。
位址、作為 PLC 的外部輸出入用、也就是說指定資料暫存器時必須要登録。
請勿登錄不必要的位址登錄。
IEC 位址、會從 PLC 位址自動被算出、使用者不需要輸入。

資料型

登録位址後顯示為預設資料型(例: 對於輸入/輸出用「BOOL」)。
也可以選擇其他的資料類型。

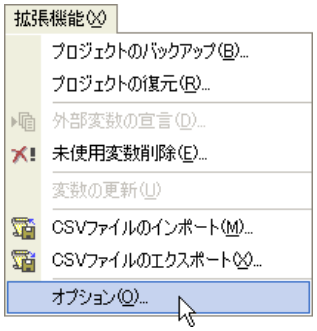
省略文字	涵義	範圍	資料長
BOOL	布林	0(FALSE)或是 1(TRUE)	1Bit
INT	16bit 整數	-32,768～32,767	16Bit
DINT	32bit 整數	-2,147,483,648～2,147,483,647	32Bit
WORD	16bit WORD	0～FFFF(H)	16Bit
DWORD	32bit WORD	0～FFFFFFFF(H)	32Bit
REAL	浮點數	-1.175494×10^{-38}～-3.402823×10^{38} と 1.175494×10^{-38}～3.402823×10^{38}	32Bit
STRING	字串	1～255 文字(ASCII)	1～255Byte
TIME	16bit 時間(間隔)	T#0.01S～T#327.67S	16Bit
TIME	32bit 時間(間隔)	T#0.01～2,147,483,647S	32Bit

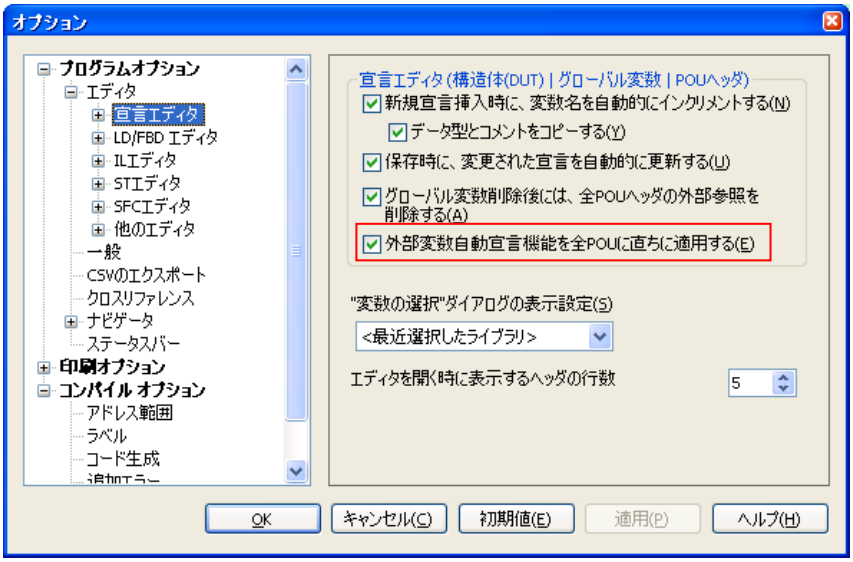
初始值

對應所選択的資料類型會自動顯示預設的初始值。
這數值在 PLC 起動時自動分配到變數的數值。必要時可以變更初始值。

外部變數自動宣告

此區域打勾的話、全域變數會對以後製作成全部的 POU - Header、作為外部變數(VAR_EXTERNAL)自動插入。
點擊(選單)擴充功能 → 選項 → 程式選項 → Edit → 宣告 Edit、
「外部變數自動登録功能對全 POU 都立即適用」的地方打勾、
全域變數登録時、全部的 POU - Header 會自動以外部變數插入。





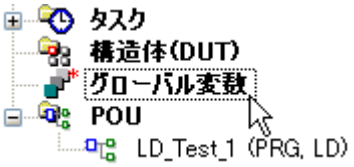
註解

填寫關於變數的註解説明。

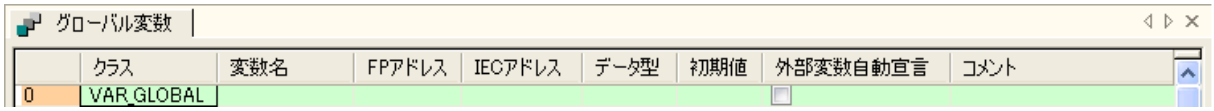
7-2-2 宣告全域變數

■操作手順

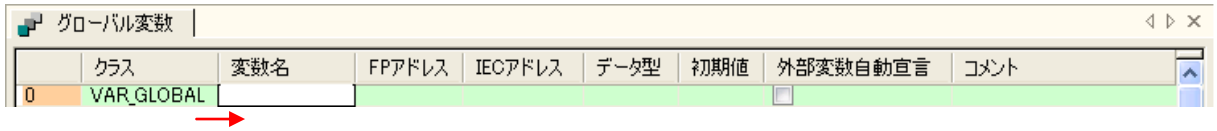
1. 點擊兩次文件導引的全域變數。



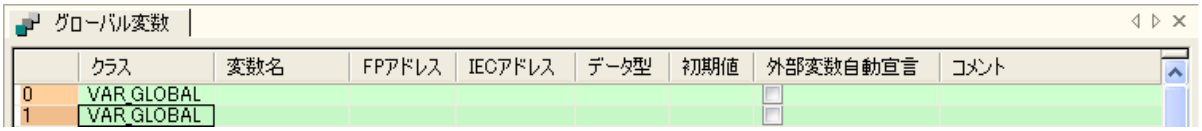
2.在全域變數明細表中將必要事項登録到各區域。



按下<Tab>鍵、移動到下個區域。



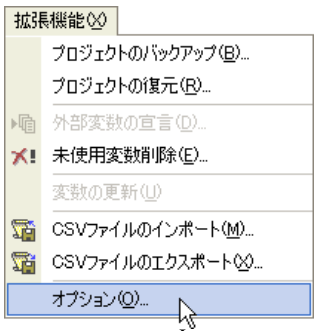
3.點擊  。

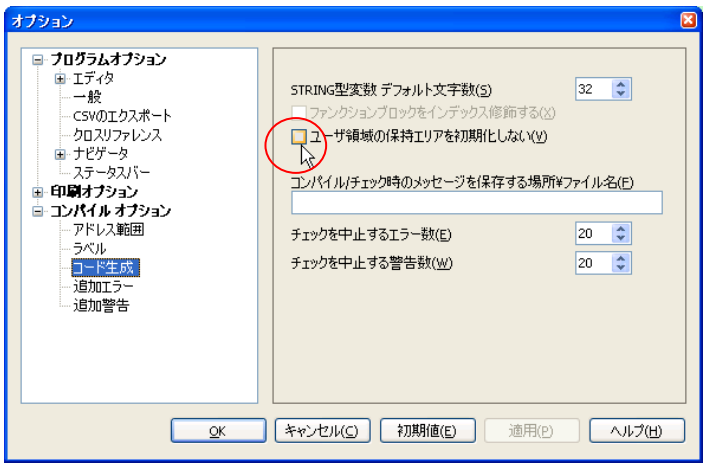


游標行之下出現新的宣告。

●注意

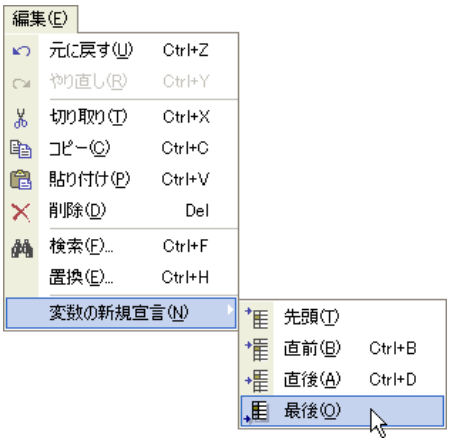
・變數名稱的前頭不可使用數字。
・PLC 的物理位址(X0, Y0 等)、不可作為變數名稱使用。
・VAR_GLOBAL 的初始值、PLC 從 PROG 模式切換到 RUN 模式時設定。
・VAR_GOLBAL_RETAIN 的初始值是程式下載到 PLC、
且只有最初為 RUN 模式時才會分配到變數。
點擊**(選單)擴充功能 → 選項** 打開對話框的
「使用者領域的保持型變數不初始化」勾為有效、位址所分配到的變數就不會再被初始化。





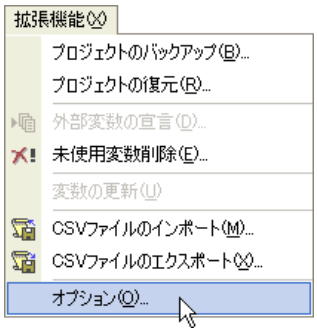
●重點

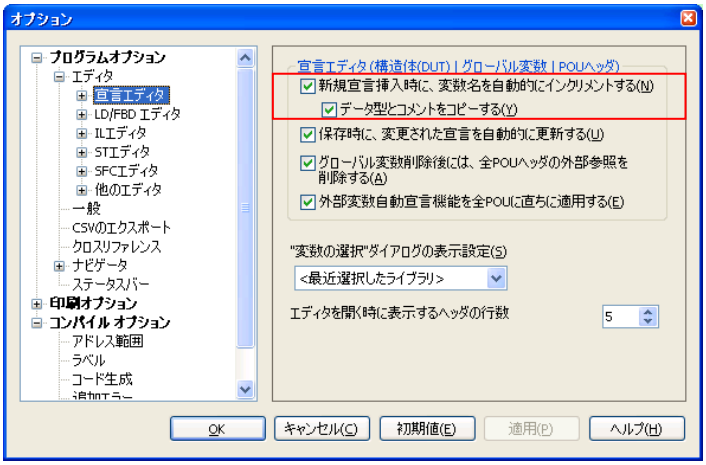
・點擊**(選單)編集 → 宣告新變數 → 前面/上一列/下一列/最後**、
在全域變數明細表的任意位置可以插入新的宣告行。



・**(選單)擴充功能 → 選項 → 程式選項 → Edit → 宣告 Edit** 的
「插入新的宣告時、變數名稱自動增量」為有效的話、
追加新宣告時、變數名稱和位址會被複製。變數名會附加上數字 1。
變數名稱的最後已經是數字時、其數字會加 1。變數位址也會加 1。
只有「新宣告被插入時、變數名稱會自動增量」為有效時、
「複製註解和類型情報」可以設定為有效。

「複製註解和類型情報」先設為有效的話、
點擊**(選單)編集 → 變數的新作成 → 下一列** 插入新的宣告行時、
就會複製現在的宣告行內容。



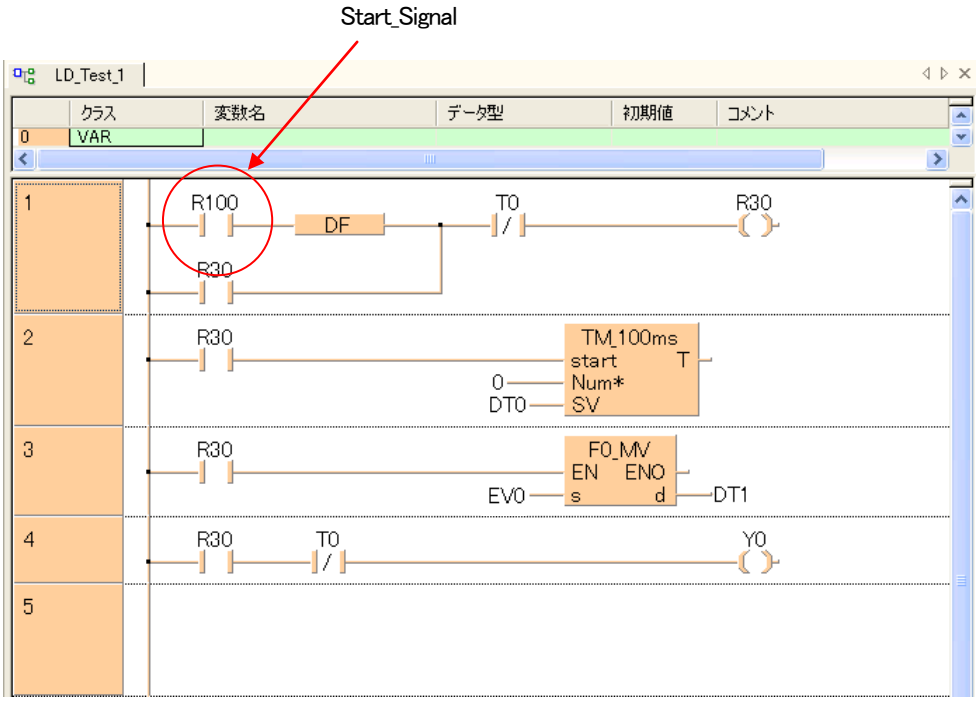


・點擊各個最上段的「class」、「變數名稱」、「資料類型」、「初始值」、「註解」
即可對各項目的字首進行排列。

7-2-3 配置全域變數

那麼實際登録全域變數、到前項為止所寫好的 LD 程式修改看看。

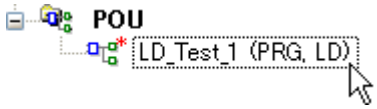
將開始信號的 R100 改寫成稱為 “Start_Signal”的全域變數。



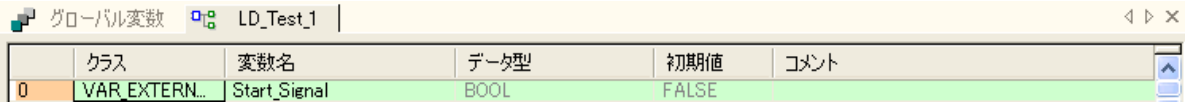
順序① 顯示全域變數明細表。

將變數名稱 “Start_Signal”寫入到 FP 位址的“R100”。
IEC 位址、資料類型、初始值會自動設定。
將外部變數自動宣告打勾。(作為外部變數(VER_EXTERNAL)會自動插入到 POU – Header)

順序② 使其顯示 POU(LD_Test_1)。



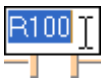
POU(LD_Test_1)點擊兩次。



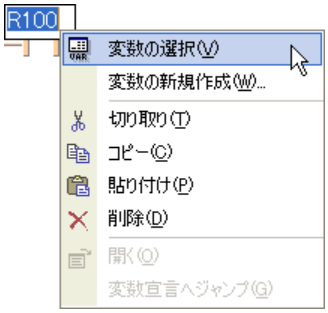
在全域變數明細表登録的 “Start_Signal”會以外部變數(VER_EXTERNAL)、自動登録。

順序③ 將“R100”變更為“Start_Signal”。

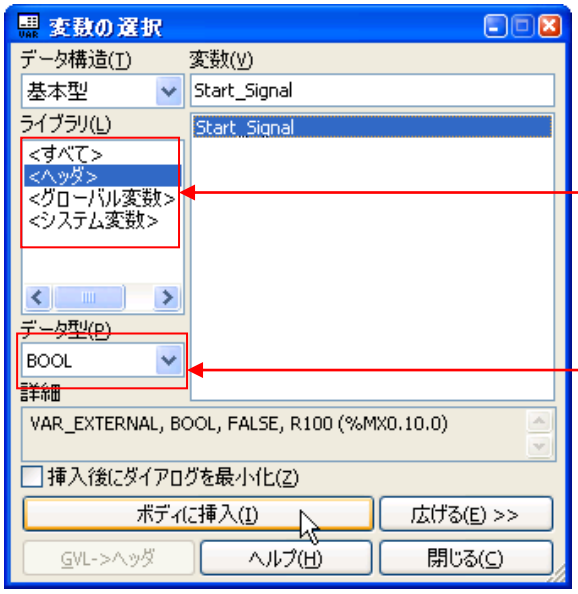
“R100”的接點上點擊滑鼠左鍵。



如上圖的顯示狀態下、點擊滑鼠右鍵。
再請點擊「變數的選擇」。



會顯示出變數的選擇對話框。
選擇“Start_Signal”。(本次只有 1 個變數、立刻可以找到！)

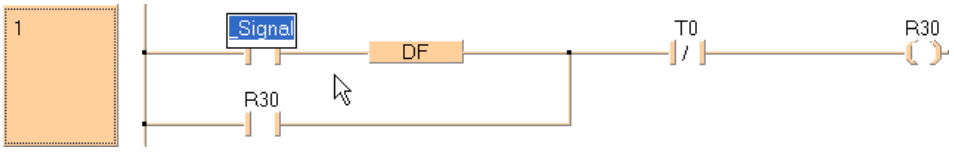


以某個 Library 選擇變數來設定。
本次從 Header 選擇。

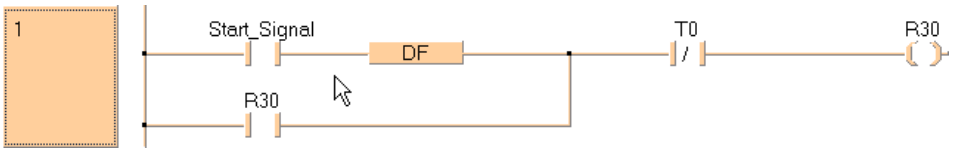
以某個資料類型選擇變數來設定。
本次從 BOOL 型選擇。

請點擊 ボディに挿入(I) 按鍵。

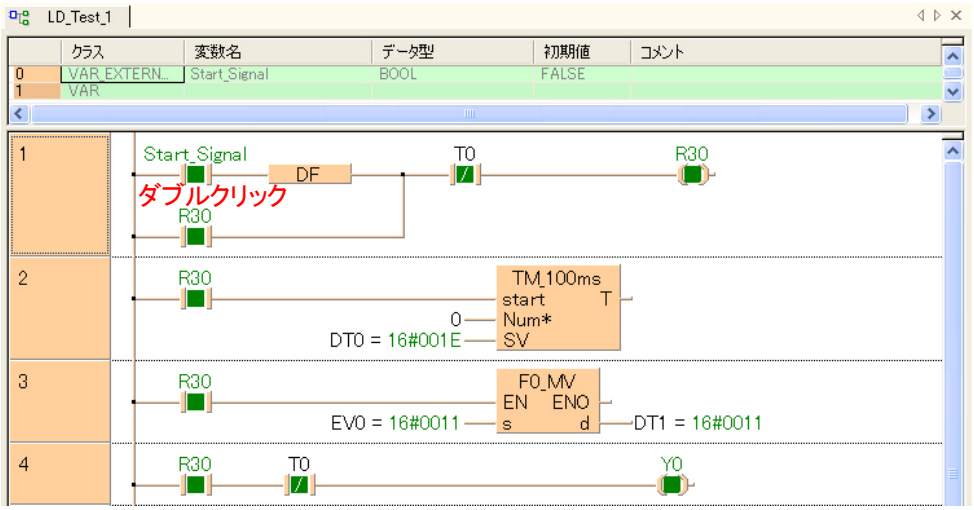
在程式區空白位置點擊滑鼠左鍵。



會輸入“Start_Signal”。



那麼開始進行編譯、實際將“Start_Signal”(R100)使其 ON 來確認動作看看。



●備註

當然、“R100”可手動(鍵盤)輸入“Start_Signal”、
不過遇到長的變數名稱太過麻煩、而且容易輸入錯誤。
已經宣告過的變數、推薦用上述方法進行輸入。

7-3 區域變數

區域變數會被 POU(程式構成單位)Header 宣告、只有對應的 POU 程式區才能使用。
其他的 POU - Header 用同樣的變數名宣告時、會作為其他的變數來使用。
POU - Header 之中、包含從全域變數明細表所被插入的變數和用 POU - Header 所宣告的區域變數。

7-3-1 從全域變數中能適用的插入變數種類

從全域變數明細表中所插入 POU - Header 的變數、適用於 3 種類型的種類。

VAR_EXTERNAL

指在全域變數明細表所宣告的 VAR_GLOBAL 變數。
PLC 從 PRG 模式切換到 RUN 模式時、電源 ON 時、會分配 VAR_GLOBAL 初始值進行分配。

VAR_EXTERNAL_CONSTANT

意指在全域變數明細表所宣告的 VAR_GLOBAL_CONSTANT 變數。
此 class 的變數、雖然在 PLC 的記憶體不會分配到位址、但在程式碼之中會插入定數。

VAR_EXTERNAL_RETAIN

指在全域變數明細表所宣告的 VAR_GLOBAL_RETAIN 變數。
點擊(選單)文件 → 編譯選項 → 產生編碼 、
「使用者領區的保持型變數不初始化」為有效時、所分配到保持型領域的變數不會再初始化。

7-3-2 區域變數的宣告內容

區域變數在 POU – Header 宣告。
區域變數的位址會由編譯時自動來分配。
區域變數適用於 7 種變數種類。
可選擇的變數種類會因為 POU 的類別(PRG, FUN, FB)有所差異。

VAR
作為演算處理的中途結果的保存等用途、於各 POU 可以宣告的變數。
VAR 的數值從 1 個處理執行到下個處理執行為止會保留而不會有任何變化。
PLC 從 PROG 模式切換到 RUN 模式時、在電源 ON 時、VAR 會回到初始值。

VAR_CONSTANT
在各 POU 可以宣告的常數變數。
VAR_CONSTANT 裡面雖然沒有分配位址、但常數會被插入到程式碼之中。

VAR_RETAIN
保持型的區域變數。
電源 OFF 時或是 PROG 模式時、數值也會被保持住。

VAR_INPUT
在 Function 以及 FunctionBlock 需要輸入必要的參數時所使用的變數。
被呼叫的 POU 會傳送變數值到 Function 以及 FunctionBlock(PRG 除外)。
VAR_INPUT 為、Function 以及 FunctionBlock 所對應的 Header 中進行宣告。
輸入變數的值雖然可以讀出、但是無法寫入。(強制輸出入除外)

VAR_OUTPUT
只有功能、功能區塊所使用的輸出變數。
PLC 從 PROG 模式切換到 RUN 模式時、電源 ON 時、VAR_OUTPUT 會被設定為初始值。

VAR_OUTPUT_RETAIN
只有 FunctionBlock 可以使用的保持型輸出變數。
電源 OFF 時或是 PROG 模式時、數值會被保持。
點選(選單)文件 → 編譯選項 → 編碼產生、
「使用者區域的保持型變數不初始化」為 “有效”時、在保持型領域所分配的變數不會被初始化。

VAR_IN_OUT
輸出入變數、擔任輸入變數和輸出變數的兩邊工作、
只有用 FunctionBlock 才能使用。

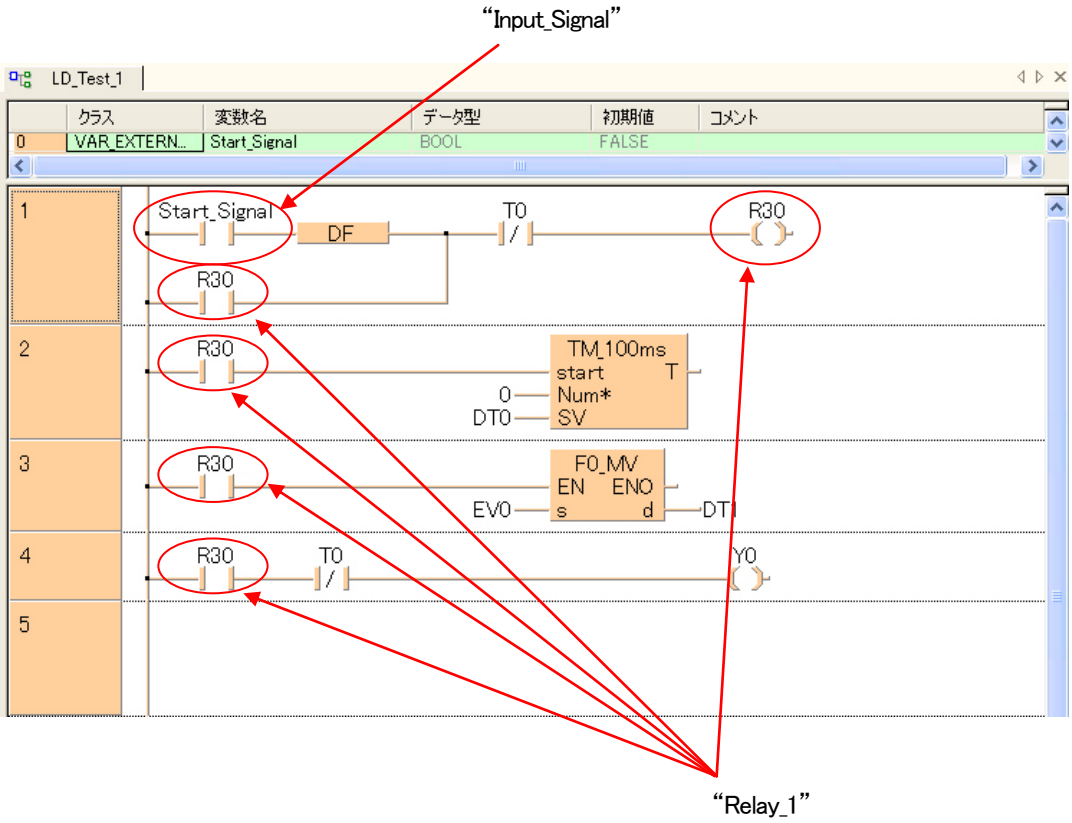
7-3-3 配置區域變數到程式上

那麼實際登録區域變数、把前項為止所作好的 LD 程式改寫看看。

首先

“Start_Signal” → “Input_Signal”
“R30” → “Relay_1”

和配置 2 個區域變数。



順序①宣告區域變数。

在 POU – Header 請輸入如下圖的區域變数。

LD_Test_1

	クラス	変数名	データ型	初期値	コメント
0	VAR_EXTERN...	Start_Signal	BOOL	FALSE	
1	VAR	Input_Signal	BOOL	FALSE	
2	VAR	Relay_1	BOOL	FALSE	
3	VAR				

順序② 刪除全域變数宣告。

選擇想刪除的行。

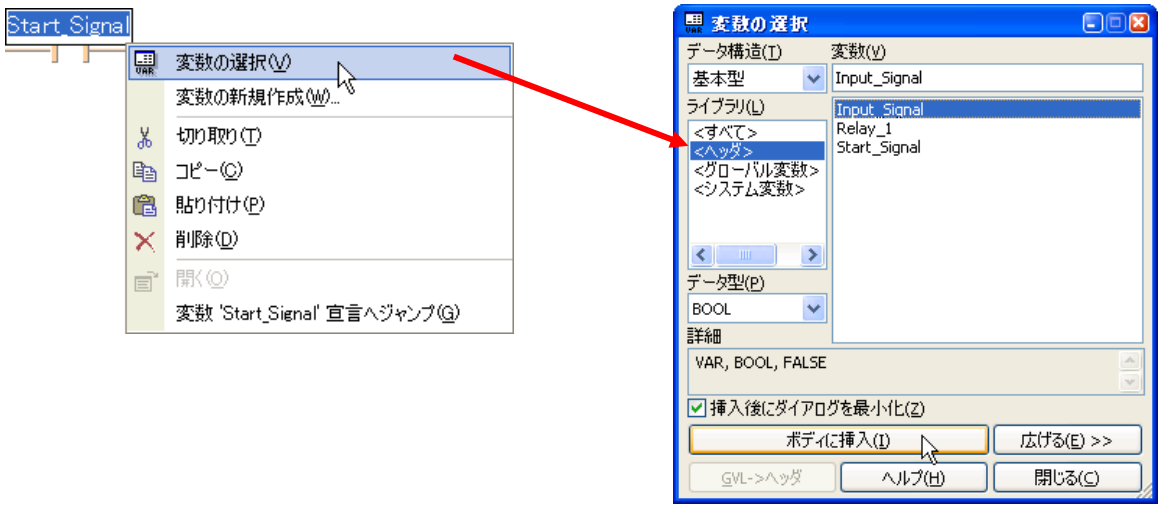
LD_Test_1

	クラス	変数名	データ型	初期値	コメント
0	VAR_EXTERN...	Start_Signal	BOOL	FALSE	
1	VAR	Input_Signal	BOOL	FALSE	
2	VAR	Relay_1	BOOL	FALSE	
3	VAR				

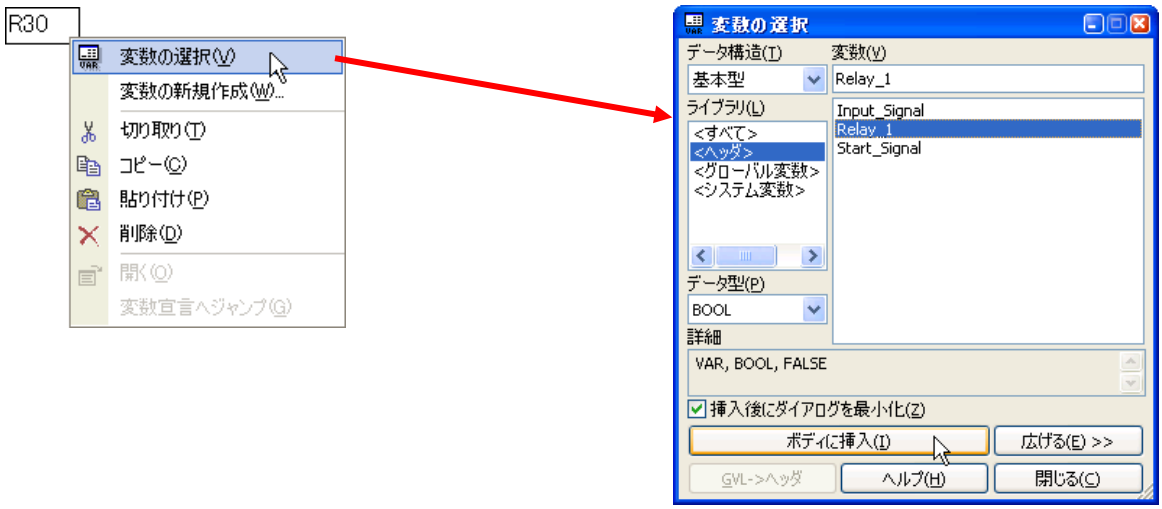
在上圖的状態下按下「Del」鍵就可以刪除。(※ 即使不刪除也沒有問題)

順序③ 配置變數。

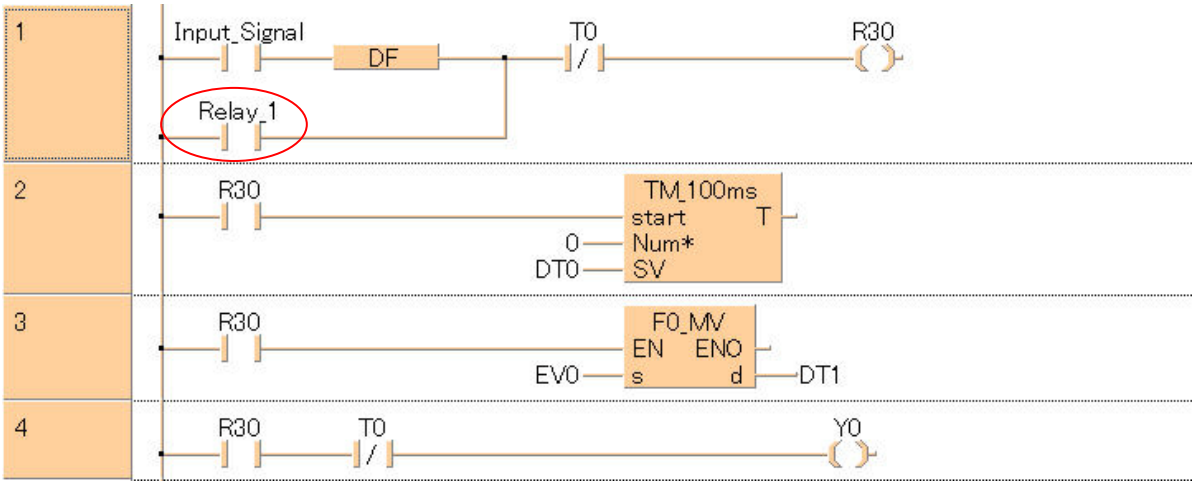
“Start_Signal”變更為“Input_Signal”。



“R30” 變更為“Relay_1”。



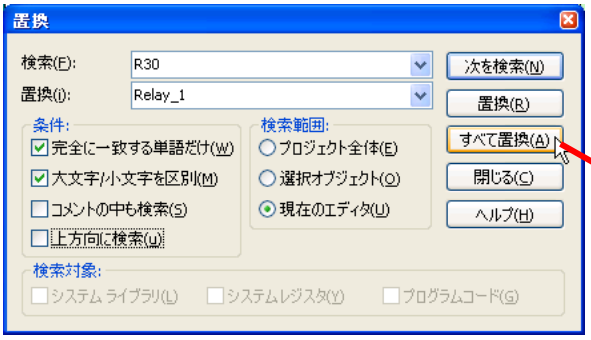
1 個地方被更改了。



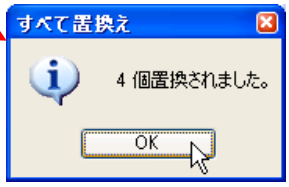
雖然可以對剩餘的“R30”做同樣的操作、但是是相當瑣的作業。

使用取代來進行變數的配置

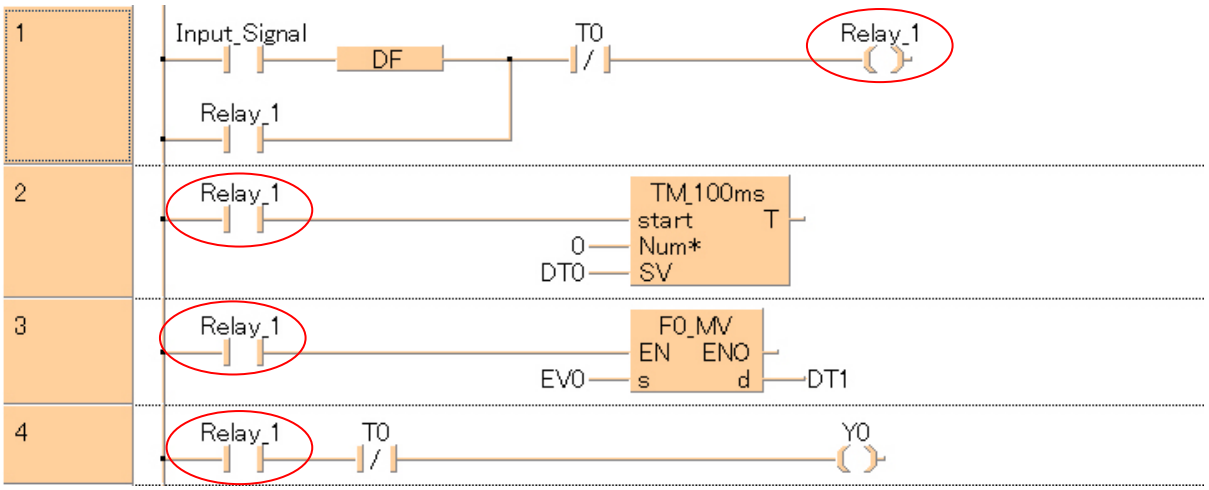
點擊**(選單)編集 → 置換**、會顯示下圖的對話框。



以下輸入
尋找 → “R30”
取代 → “Relay_1”。



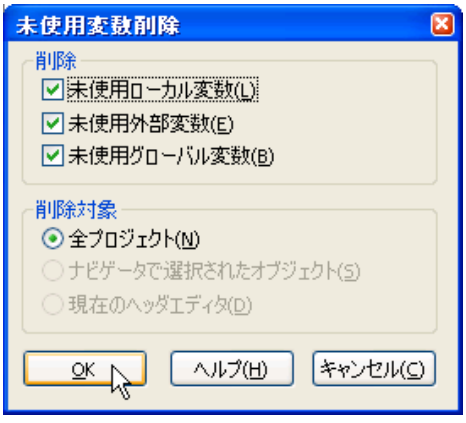
其餘的變數會被配置。



●備註　從 POU－Header 刪除全域變數

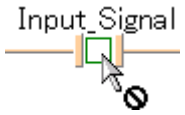
點選**(選單)擴充功能 → 未使用變數刪除**、在 POU 功能區未使用的外部變數(VAR_EXTERNAL)可以從其 POU－Header 刪除。

①點選**(選單)擴充功能 → 未使用變數刪除**、會顯示以下的對話框。

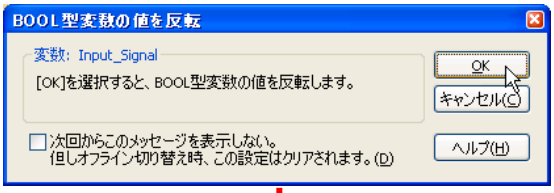


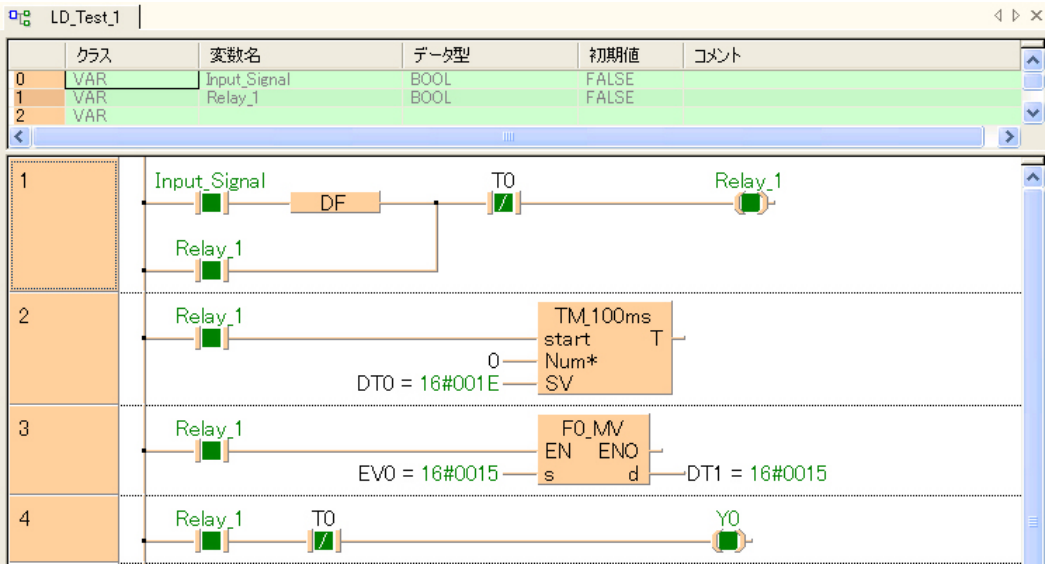
②選擇刪除欄內的「未使用外部變數」。
③選擇「刪除對象」的下面選項。
④點選「OK」。

執行編譯器、實際讓“Input_Signal”ON 來確認動作看看。



點選“Input_Signal”兩次可以反轉接點的状態。





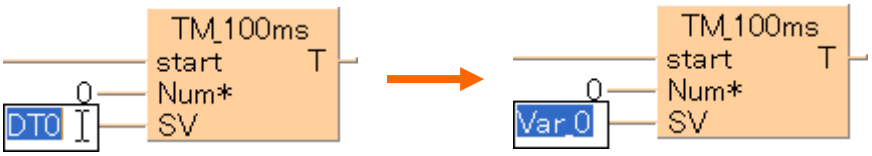
接下來

“DT0” → “Var_0”
“DT1” → “Var_1”

配置 2 個區域變數。

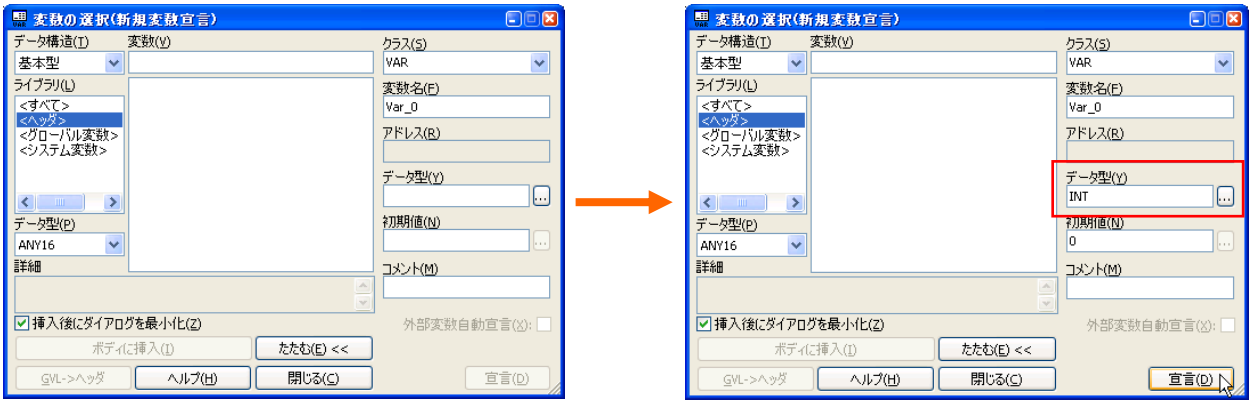
此時不用重新宣告變數、而是以每次都用宣告的方式宣告變數。

順序① “DT0”變更為“Var_0”。

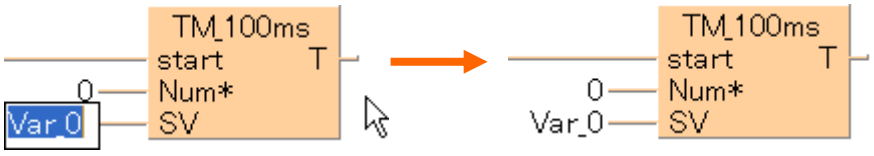


可以輸入“Var_0” 的話、請按 Enter 鍵。

顯示「變數的選擇」對話框。



資料類型選擇 “INT”、請點選 宣言(D) 鍵。



請點擊程式區適當的空白地方。　　確定為“Var_0”。

順序② 確認 Header 的變數宣告

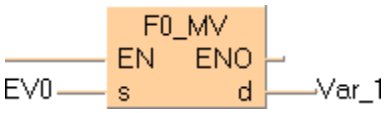
在順序①點擊 宣言(D) 鍵時被登録。

如此、不用重新登録變數也可以每次宣告變數。
但是這時候、變數會被追加登録到 Header 的最後一行、
因此需要將變數排列順序時等、必須調整位置。

疾著點擊 Header 的最上段的「等級」、「變數名稱」、「資料類型」等、
對於各項目可以照字母排列順序。

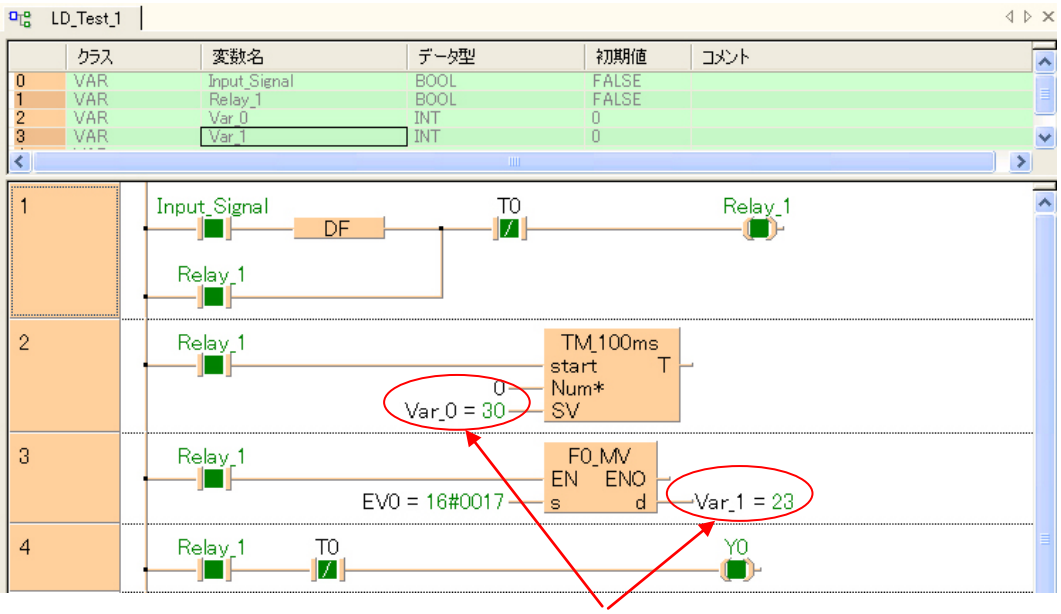
順序③ “DT1”變更為“Var_1”。

用同樣的步驟、”DT1”變更為”Var_1”。



順序④ 動作確認

那麼執行編譯、實際讓“Input_Signal”ON 確認動作看看。



10 進位

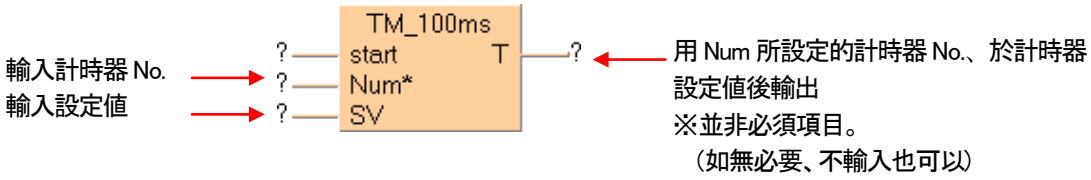
可以監控方式看到 Var_0 和 Var_1 的數值以 10 進位顯示。
用 INT、DINT 型宣告、也可以已 10 進位來監控。

那麼計時器 T0(TM_100ms)會如何？
計時器 T0 等的計時器號碼無需固定、只須替換成使用變數的計時器。

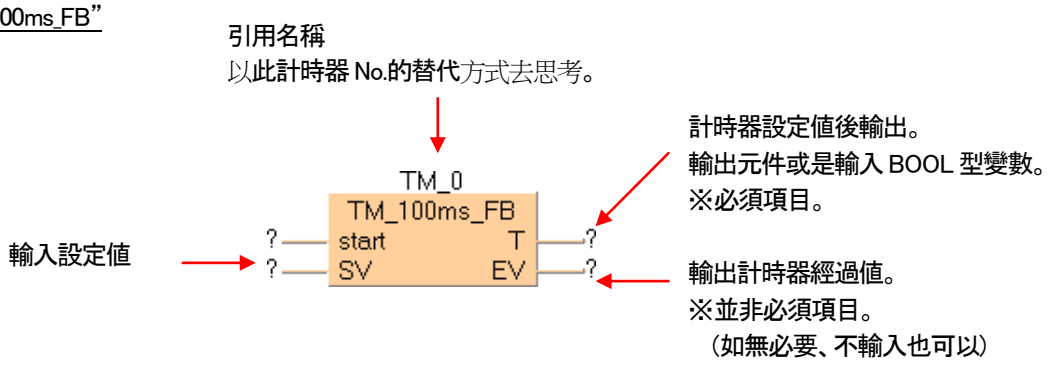
要將“TM_100ms” 指令替換成使用變數的計時器時、選擇“TM_100_ms_FB”。

●“TM_100ms”和“TM_100ms_FB”的差異

“TM_100ms”



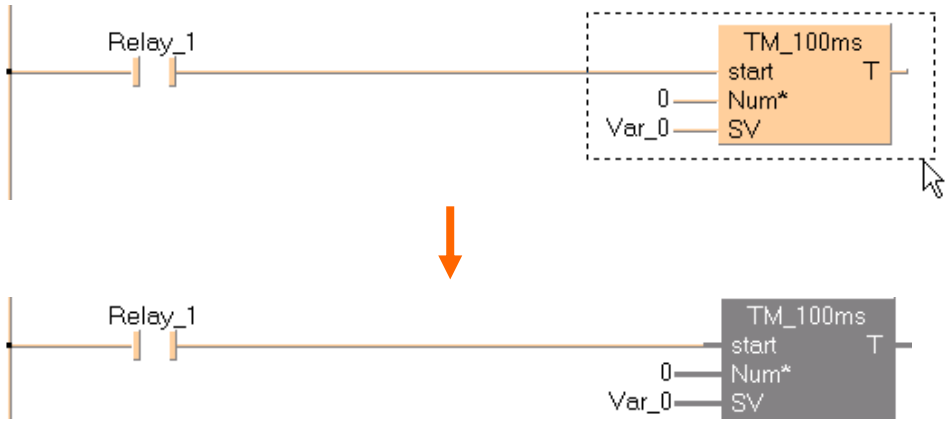
“TM_100ms_FB”



以下順序進行。

順序① 刪除“TM_100ms”。

選擇“TM_100ms”。
如下圖選擇每個變數。



“TM_100ms”以及變數在被選擇的狀態下、
按下 DEL 鍵。

順序② 選擇“TM_100ms_FB”。

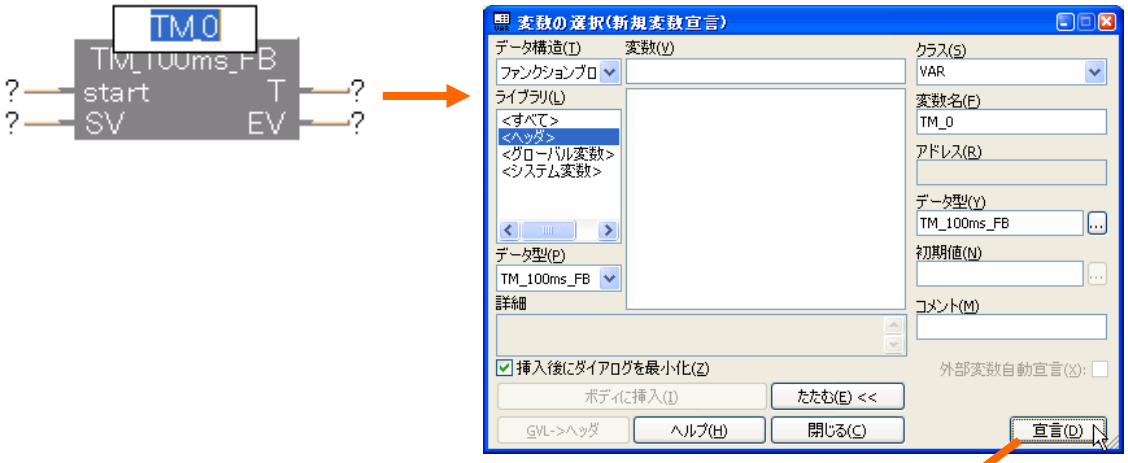
順序③ 輸入引用名稱。

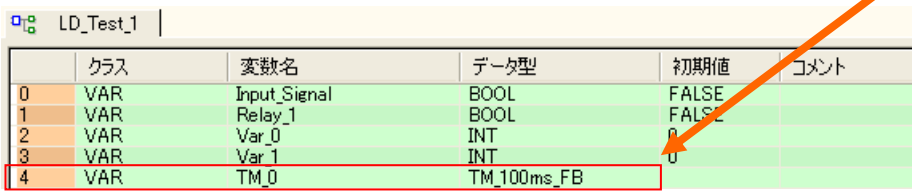
“TM_100_ms_FB”貼到程式區上、會如上圖般催促輸入“引用名稱”。

例如、使用複數個“TM_100_ms_FB”時、必須要知道使用那個計時器。
因此、利用附上引用名稱可以來區別。

在此輸入“TM_0”作為引用名稱。

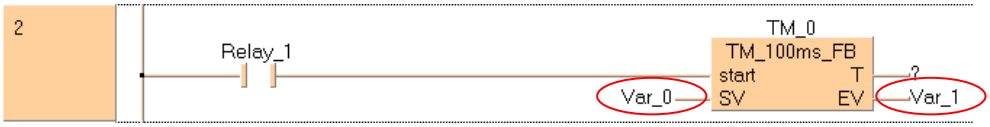
輸入確定的同時、在 Header 中會登録引用名稱。



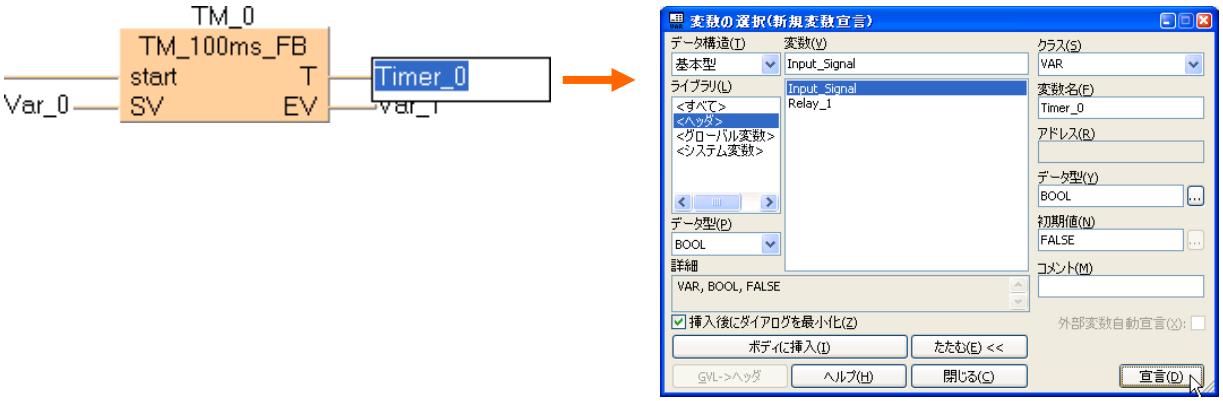


順序④ 輸入變數。

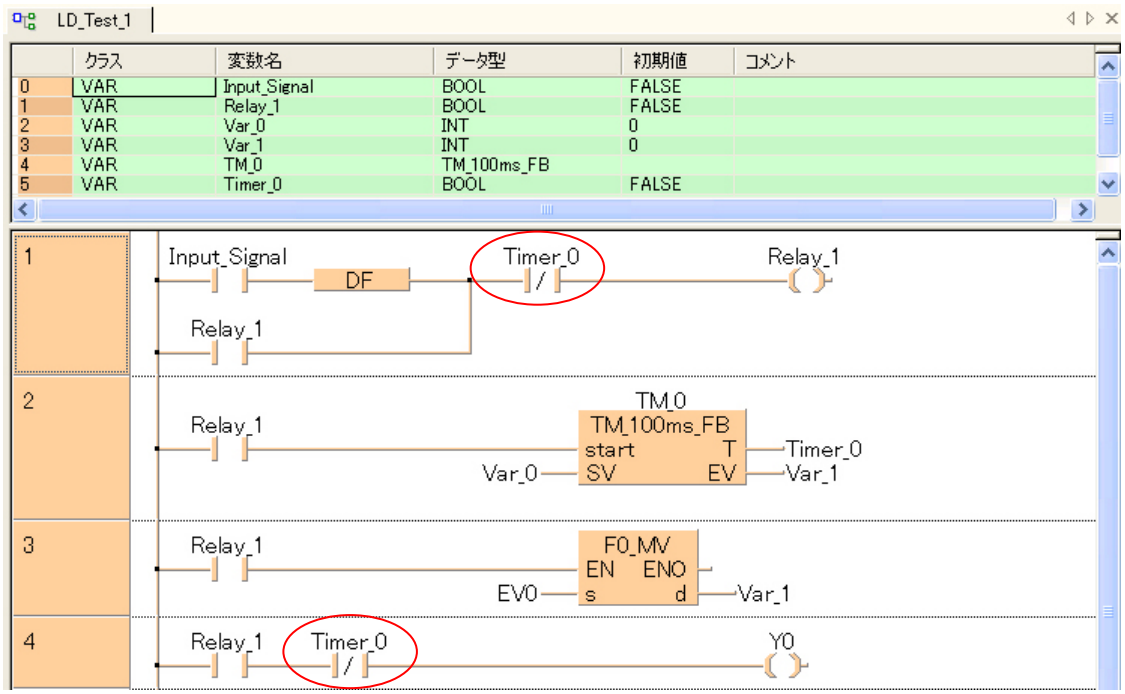
已宣告的 Var_0 輸入到設定值(SV)、輸入 Var_1 到經過值(EV)。



宣告計時器輸出部分的輸入變數(BOOL 型)。
在此、以“Timer_0”的變數名稱作宣告。



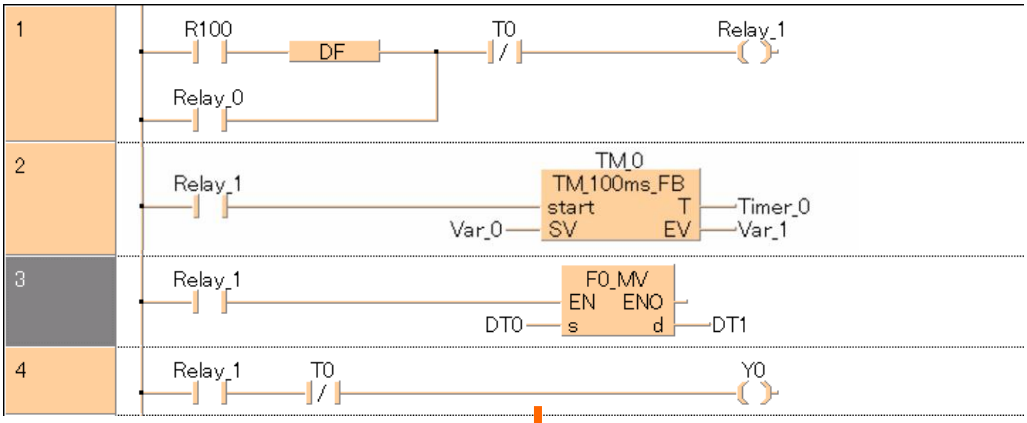
請將其餘的“T0”變更為“Timer_0”。



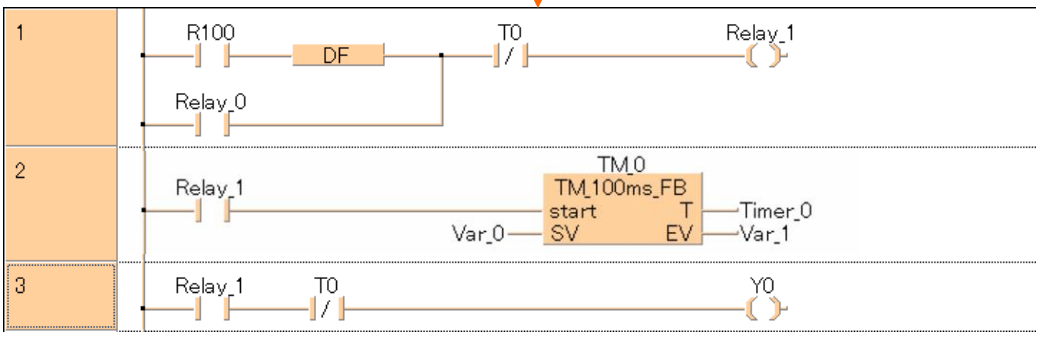
順序⑤ 刪除不要的 Network。

經由使用“TM_100ms_FB”、會讓經過值輸出到“Var_0”、
程式區的 Network3 就不需要了。

選擇 Network3



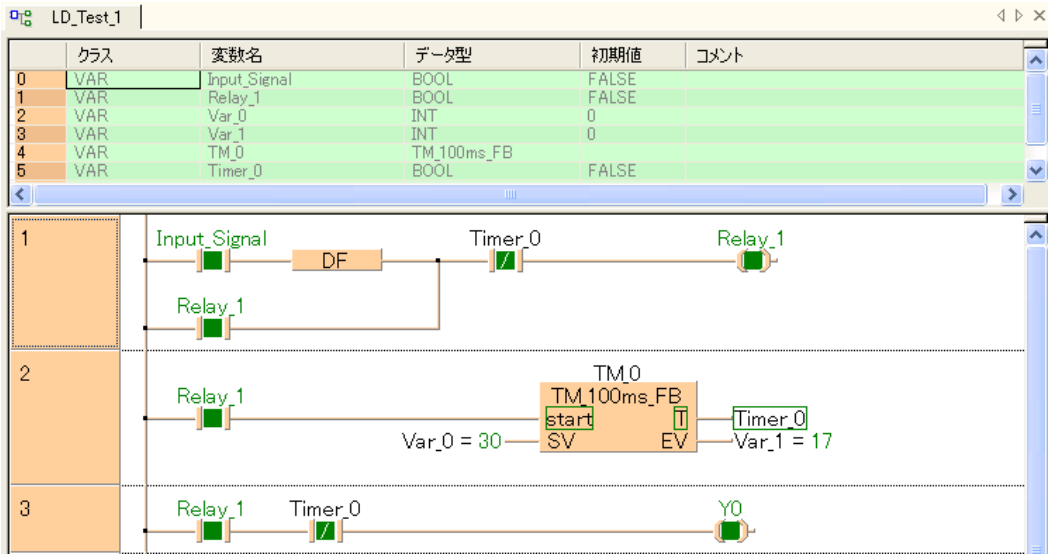
按下 DEL 鍵刪除。



以上的元件全部置換成變數。

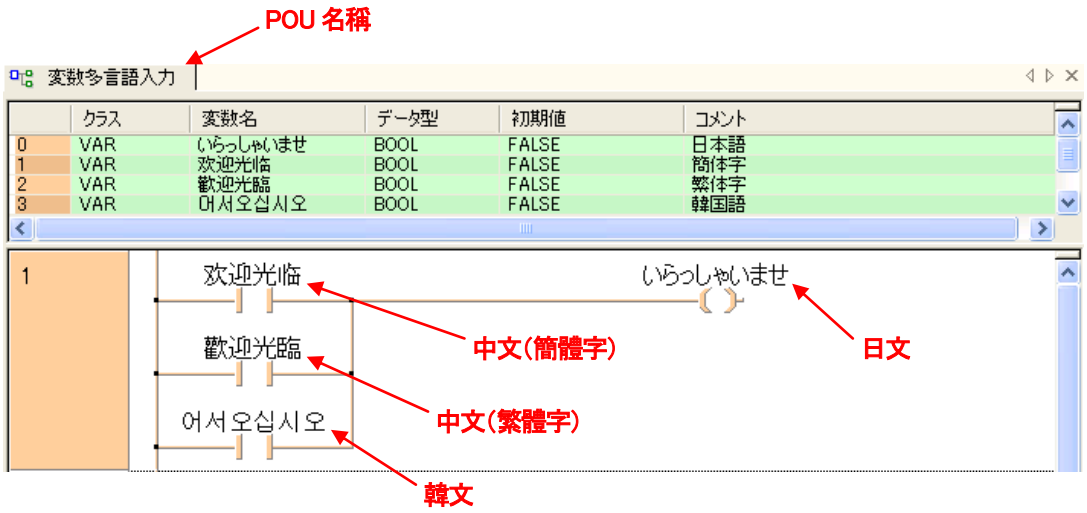
順序⑥ 確認動作。

在此執行編譯、實際讓“Input_Signal” ON 來確認動作看看。



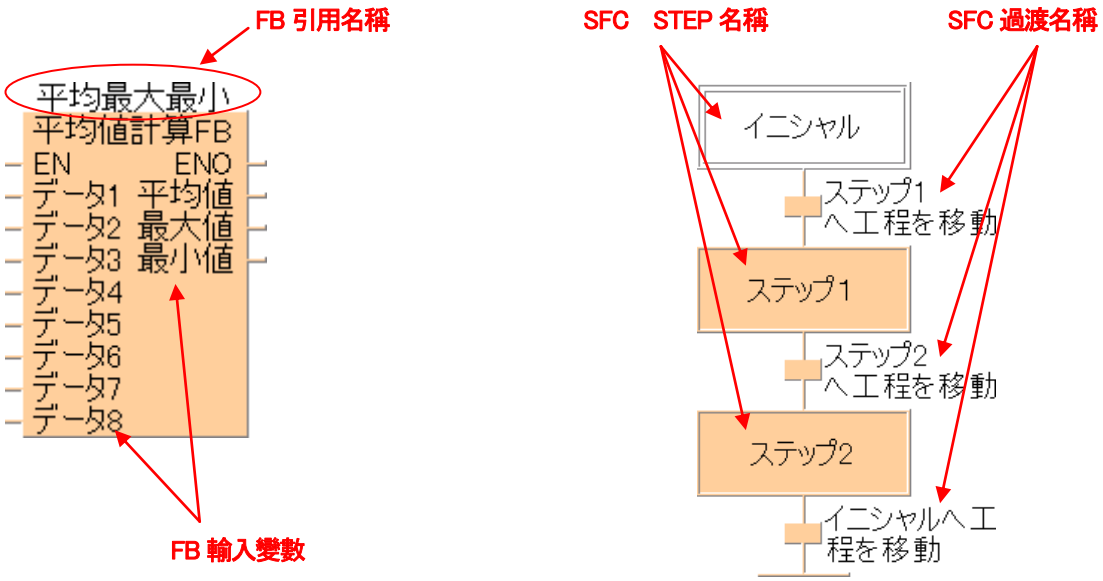
●備註　關於輸入語言

從 FPWIN Pro Ver6 以後、有對應 Unicode、
變數以及 FB 引用名稱等、可以輸入多國語言。



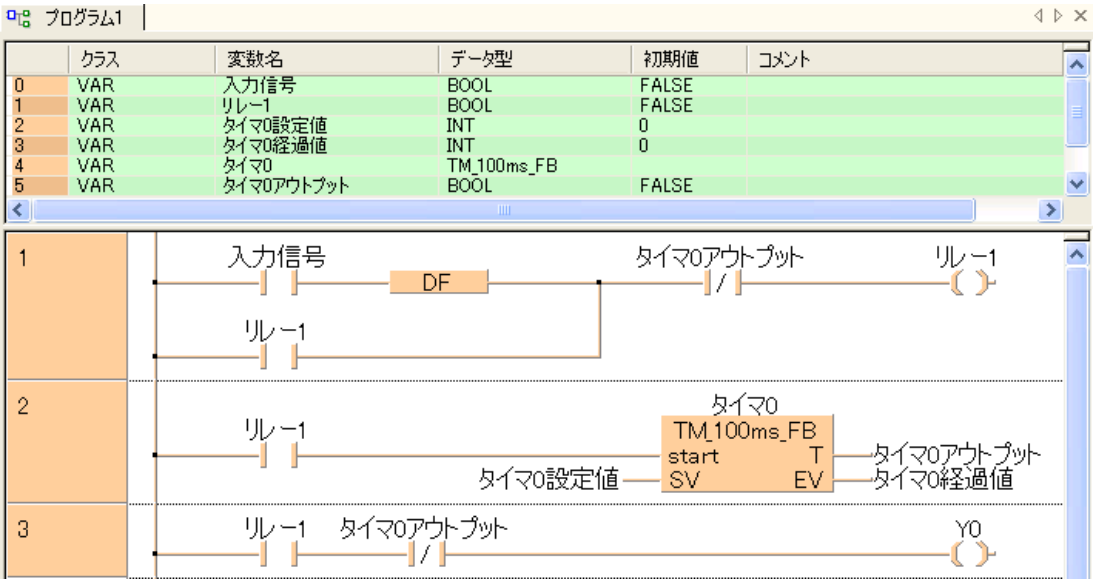
如上圖般可以將各國語言混合使用。

還有 FB(Function Block)和 SFC(連續功能流程圖)也因為 Unicode 可以輸入多國語言
(下圖例為日文)。



※關於 FB(功能區塊)在第 8 章有做解說。
※關於 SFC(連續功能流程圖)在第 12 章有做解說。

到目前為止所做的程式變更為日文變數的例子。



7-4 變更變數

變數名稱、資料類型等的參數變更、全域變數在全域變數明細表中、區域變數在各個 POU - Header 執行。
變數的變更內容、會反映在目前正打開文件的 POU - Header 和程式區。

從使用「變數的選擇」對話框的 POU 程式區可以直接將全域變數及區域變數的特定參數做變更。

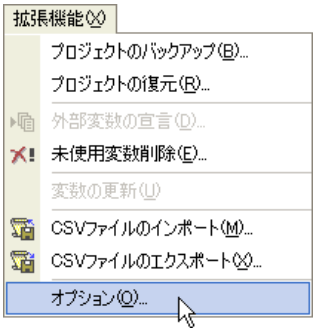
7-4-1 變更 POU - Header 及全域變數明細表的變數

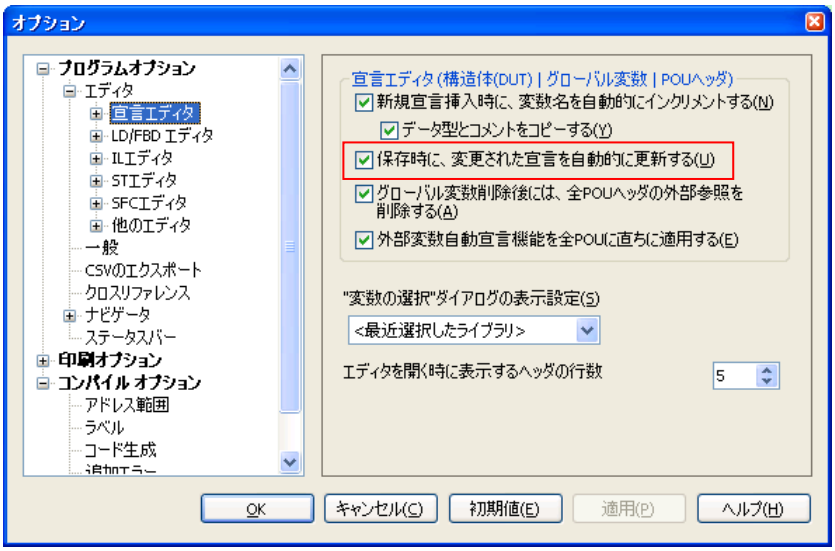
在 POU 裡讓全域變數和區域變數的變更內容為有效的方法有以下 2 種。

・全 POU - Header/程式區自動更新。
・全 POU - Header/程式區逐項更新。

■在 POU - Header 及全域變數明細的變數變更

在**(選單)擴充功能 → 選項 → 程式選項 → Edit → 宣告 Edit**、
「保存時所變更的宣告自動更新」為有效時、在全域變數明細表及 POU - Header 的變數的全部變更內容、
目前打開的 Project 中、正在使用的 POU - Header/程式區的該當變數的全部都會被更新。





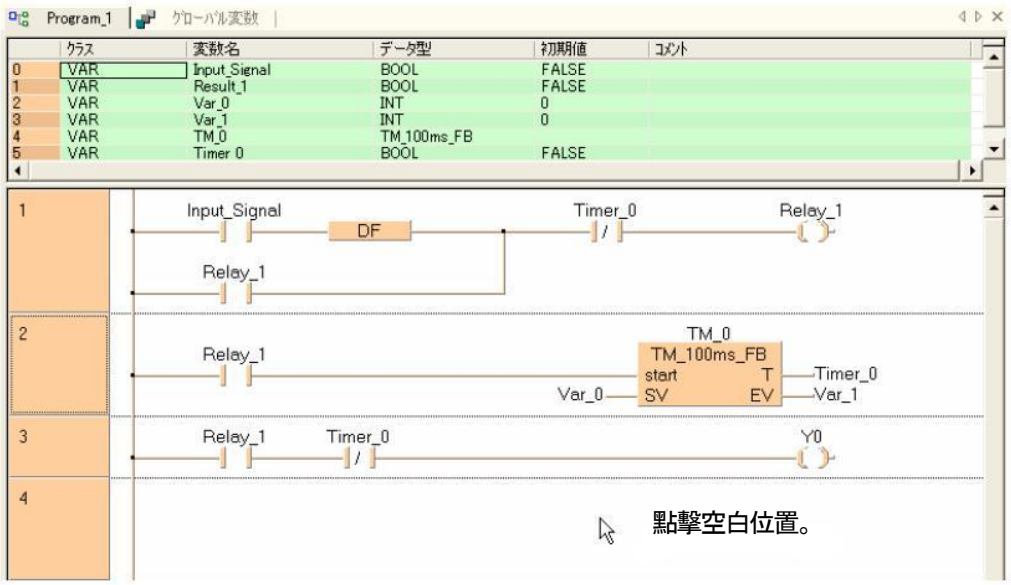
■POU – Header 的變更內容反映到程式。

現在正登録中的變數名稱想變更時、在程式區很多地方都都有記述、如果一個個變更是相當麻煩的。

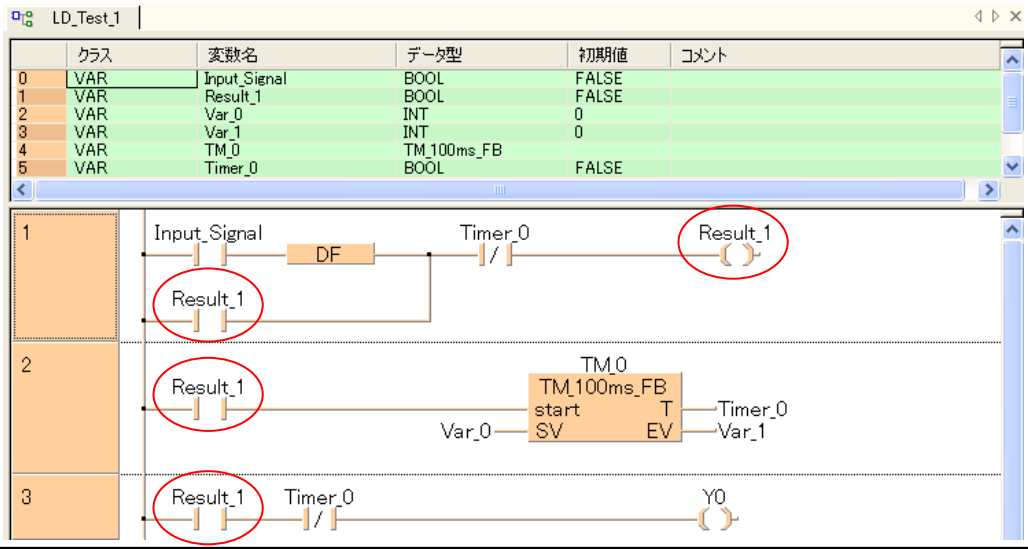
將現在程式區上所正使用的變數“Relay_1”全部變更為“Result_1”。

從 POU – Header 變更目的的變數名稱。

請在程式區上的適當空白位置點擊。



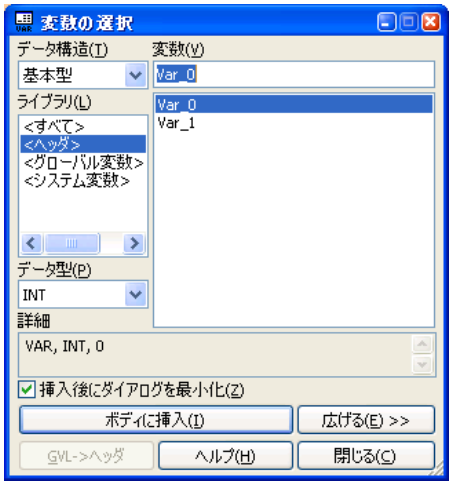
全部的“Relay_1”變更為“Result_1”。



7-4-2 在「變數的選擇」對話框、變更變數

■操作順序

①點擊變數“Var_0”、打開「變數的選擇」對話框。



②點選「放大」鍵。

「變數的選擇」對話框會往右邊擴大、會顯示所選擇變數的參數。

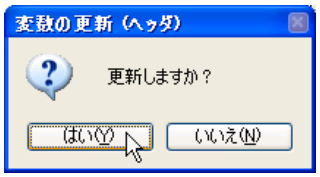
③變更任意的變數參數內容。

在此對話框、無法變更「變數名稱」。

輸入新的變數名稱、點選「更新」鍵會變成「宣告」鍵、可以宣告新的變數。

④點擊更新鍵。

會顯示變數的更新確認訊息。



⑤點擊「是」的按鍵。

變數已經變更。在對話框左邊的「詳細」裡面、顯示出所變更的變數參數。

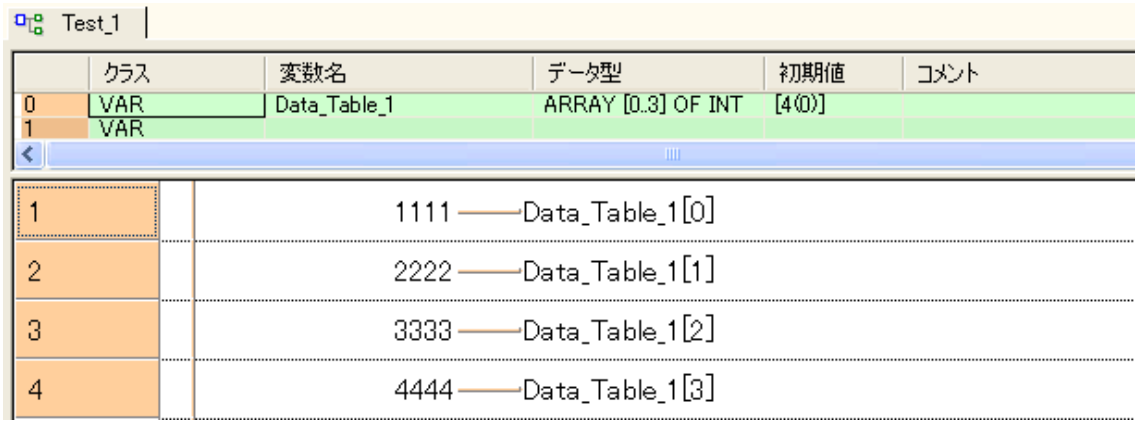
對其他變數變更、從「變數」的下面一覽選擇任意的變數、對話框右側的區域會顯示參數。

登録關於變數正在顯示中的「變數」、「Library」、「資料結構」、「資料類型」的內容。

7-5 陣列變數(ARRAY)

陣列為變數的群組。其中的全部變數、都擁有相同的基本資料類型、
對連續的資料區塊可以依照順序良好的分配為群組化的東西。
此變數群組自身為1個變數。
因此需要宣告。在程式中、可以使用陣列全體、或是個別陣列。

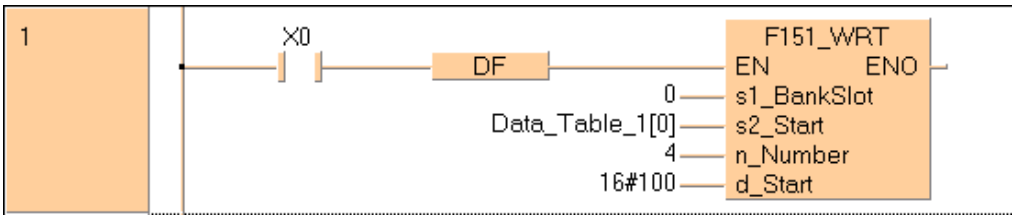
輸入陣列名稱[Var_1]則可以使用 1 次元陣列要素。
・陣列名(顯示陣列名稱)
・Var_1 在陣列宣告所設定範圍內的數值的 INT 型變數或常數。



上述為執行 INT 型變數用 4 個陣列變數宣告的例子。
並在全域變數中宣告 Data Table1 起始位值為 DT1000 後、

Data_Table_1[0] → DT1000(1111)
Data_Table_1[1] → DT1001(2222)
Data_Table_1[2] → DT1002(3333)
Data_Table_1[3] → DT1003(4444)

分配到連續的記憶體之中。



如上圖、必須要為連續的資料時是相當有用的。

而且、陣列可以用 1 次元、2 次元、3 次元的宣告、

1 次元陣列要素 ： Data_Table_1[0]、Data_Table_1[1]・・・
2 次元陣列要素 ： Data_Table_1[0,0]、Data_Table_1[0,1]・・・
3 次元陣列要素 ： Data_Table_1[0,0,0]、Data_Table_1[0,0,1]・・・

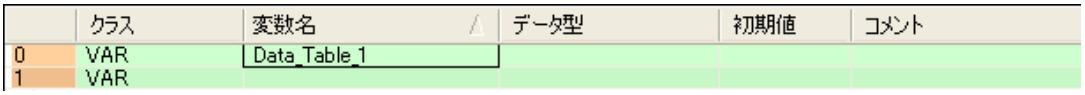
可用逗點區分指定。

在複數的 POU 使用等、陣列在 Project 全體使用時、
在全域變數明細表內必須要宣告為全域變數。

■陣列宣告的操作順序

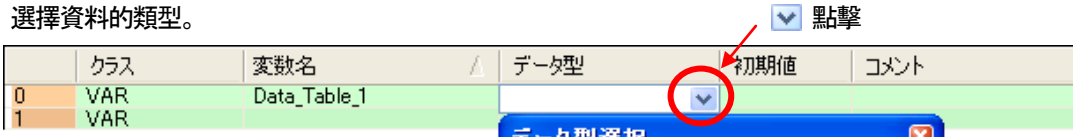
1. 輸入變數名稱

在此輸入“Data_Table_1”。

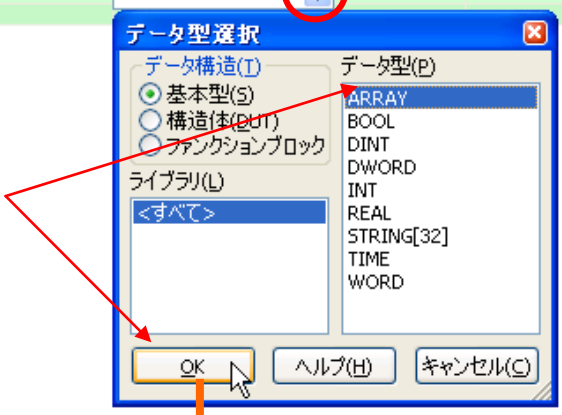


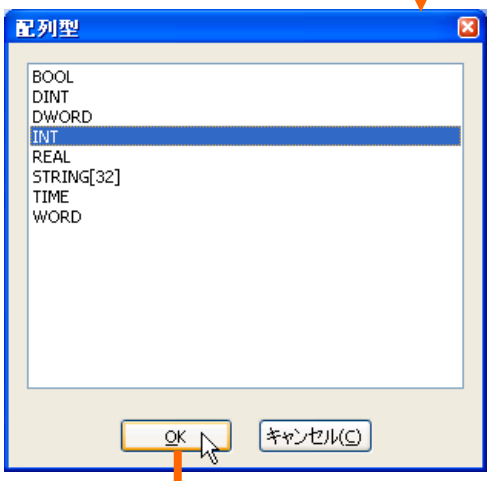
2. 輸入資料類型

選擇資料的類型。



在“ARRAY”按下[OK]鍵。

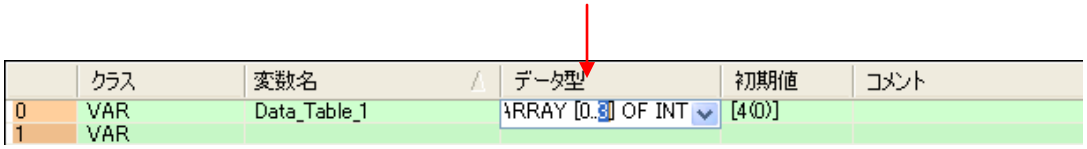




在此選擇“INT 型”、按下[OK]鍵。

■備註

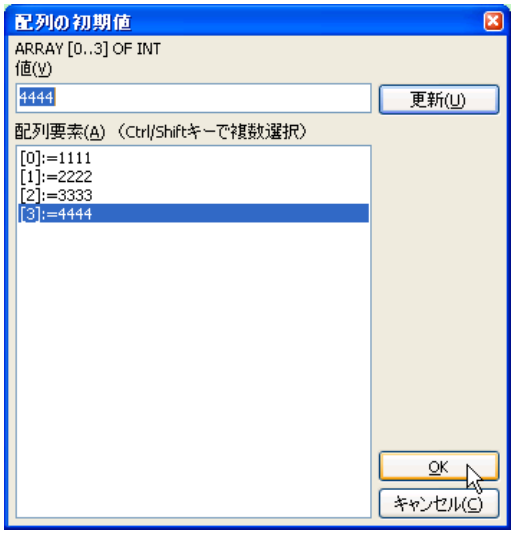
依據所需的陣列數、請輸入陣列數。



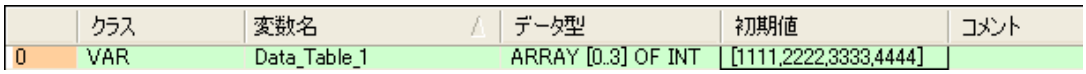
也可以作為初始值輸入資料。

1. **初始值的輸入欄用滑鼠點選。**

2. **輸入資料。**



個別輸入資料、按下 OK 鍵就結束。



7-6 資料結構(DUT)

在資料結構(Data Unit Type: DUT)中、可以定義多個不同的資料類型來組成的資料結構。
DUT 定義之後、全域變數明細表及 POU – Header 內的標準資料類型(BOOL 型、INT 型等)都相同的可以使用。

DUT 例

Data_Table_DUT [構造体(DUT)]

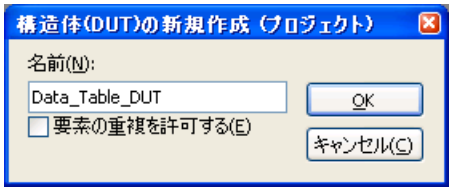
	変数名	データ型	初期値	コメント
0	Data_1	WORD	16#112	
1	Data_2	INT	100	
2	Data_3	INT	1000	
3	Data_4	INT	30	
4	Data_5	DINT	50000	
5	Data_6	INT	0	

複数の異なるデータ型で構成

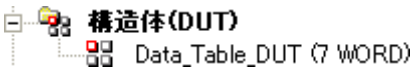
■操作方法

1. **工具列的「資料結構(DUT)」或是、**
   在 Project 導引的 DUT 點擊右鍵從選單選擇「製作新資料結構(DUT)」。

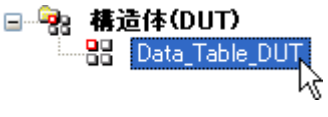
在此附上"Data_Table_DUT"的名稱。



在 Project 導引作為 DUT 顯示出"Data_Table_DUT"。



2. **DUT 的變數設定**
   "Data_Table_DUT"點選兩次。



點選兩次

Data_Table_DUT [構造体(DUT)]

	変数名	データ型	初期値	コメント
0				

顯示 DUT Header。

輸入變數名稱和資料類型、初始值。

Data_Table_DUT [構造体(DUT)]

	変数名	データ型	初期値	コメント
0	Data_1	WORD	16#112	
1	Data_2	INT	100	
2	Data_3	INT	1000	
3	Data_4	INT	30	
4	Data_5	DINT	50000	
5	Data_6	INT	0	

3. 在使用的 POU 中宣告。

在此用“Test_1”(PRG)宣告。

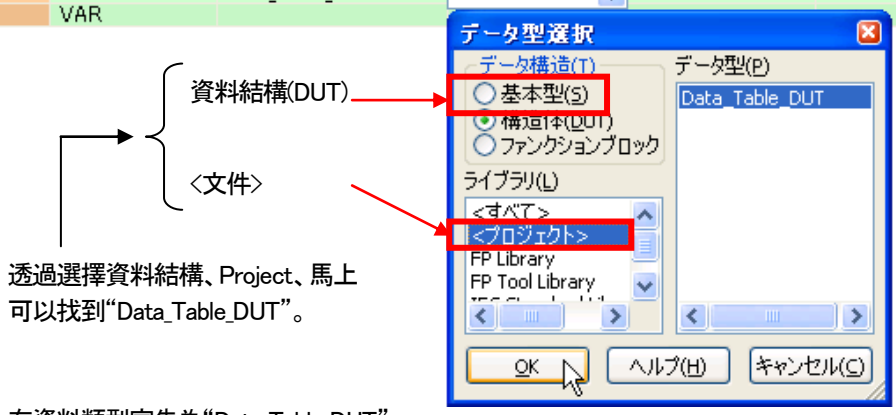
變數名稱為“Data_Table_1”。

Test_1

	クラス	変数名	データ型	初期値	コメント
0	VAR	Data_Table_1			

Test_1

	クラス	変数名	データ型	初期値	コメント
0	VAR	Data_Table_1			
1	VAR				



透過選擇資料結構、Project、馬上可以找到“Data_Table_DUT”。

在資料類型宣告為“Data_Table_DUT”。

Test_1

	クラス	変数名	データ型	初期値	コメント
0	VAR	Data_Table_1	Data_Table_DUT		

4. 變更 DUT 變數的值

DUT 變數作為可變資料使用時、在 DUT 名稱用點為分割符號指定目標變數名稱。

Test_1

	クラス	変数名	データ型	初期値	コメント
0	VAR	Data_Table_1	Data_Table_DUT		
1			10000 ——Data_Table_1.Data_5		

Data_Table_1.Data_5

點分割符號

7-7 取得正在使用變數的 PLC 位址

使用區域變數撰寫程式時、實際其變數所分配到的 PLC 位址要編譯後才會知道。
對於變數、作為編譯後所分配到的 PLC 位址的取得方法
在 FP Tool Library 有準備以下的函數。

- Adr_Of_Var_I : 取得使用輸入變數的 PLC 位址的起始位址。
- Adr_Of_Var_O : 取得使用輸出變數的 PLC 位址的起始位址。
- AdrLast_Of_Var_I : 取得使用輸入變數的 PLC 位址的終點位址。
- AdrLast_Of_Var_O : 取得使用輸出變數的 PLC 位址的終點位址。
- Adr_Of_VarOffs_I : 取得使用輸入變數的 PLC 位址(附偏移量)。
- Adr_Of_VarOffs_O : 取得使用輸出變數的 PLC 位址(附偏移量)。
- AdrDT_Of_Offs_I : 取得使用輸入變數的 DT(資料暫存器)的位址。
- AdrDT_Of_Offs_O : 取得使用輸出變數的 DT(資料暫存器)的位址。
- AdrFL_Of_Offs_I : 取得使用輸入變數的 FL(檔案暫存器)的位址。
- AdrFL_Of_Offs_O : 取得使用輸出變數的 FL(檔案暫存器)的位址。

Adr_Of_Var_I 的使用範例

FP0 F168(定位控制指令)作成時

DUT

Data_Table_DUT [構造体(DUT)]

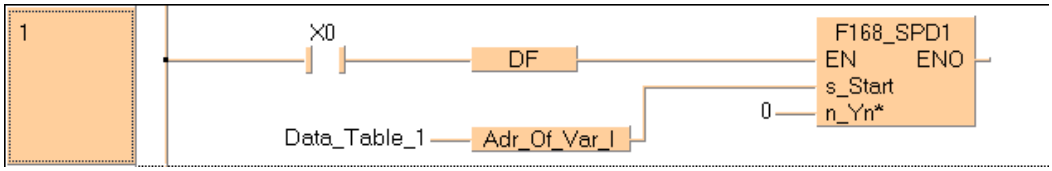
	変数名	データ型	初期値	コメント
0	Data_1	WORD	16#112	
1	Data_2	INT	100	
2	Data_3	INT	1000	
3	Data_4	INT	30	
4	Data_5	DINT	50000	
5	Data_6	INT	0	

POU Header

Test_1

	クラス	変数名	データ型	初期値	コメント
0	VAR	Data_Table_1	Data_Table_DUT		

POU 程式區



※ 在上例中、透過使用"Adr_Of_Var_I"、在 DUT 所宣告變數在編譯後、取得分配到 PLC 的起始位址。

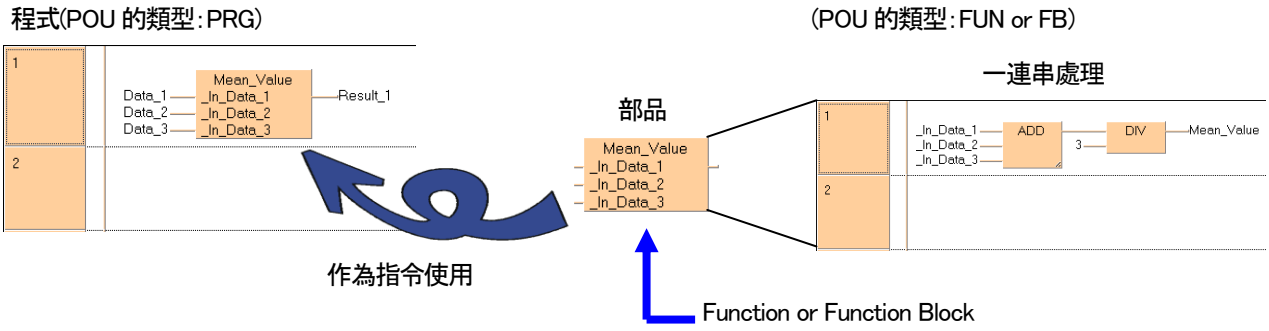
第8章

製作 Function(FUN)/Function Block(FB)

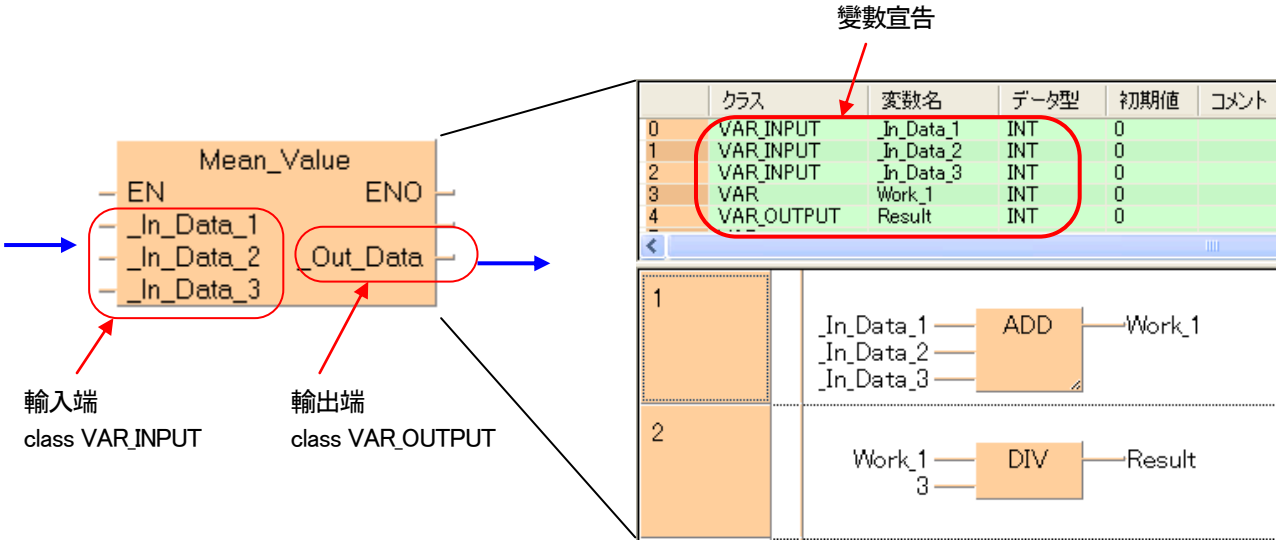
8-1 概要

■所謂 Function(FUN)/Function Block(FB)

Function／Function Block 是透過一連串處理、以函數化登録、使其可以作為一個指令使用的功能。



●對 Function 和 Function Block 的輸入和輸出



對於 Function 和 Function Block 的輸入和輸出需要使用變數。
如上圖、雖然在 Function 和 Function Block 內的 Header 宣告、
變數的 class、分別為
輸入→**VAR_INPUT**
輸出→**VAR_OUTPUT**　　進行宣告。

VAR_INPUT

在 Function 和 Function Block 內輸入必要參數所使用的變數。被呼叫的 POU 會將變數值傳送到 Function 和 Function Block(PRG 除外)。VAR_INPUT 在 Function 或是 Function Block 所對應的 Hader 中被宣告。輸入變數的值雖然可以讀出、但是無法寫入。
(強制輸出入除外)

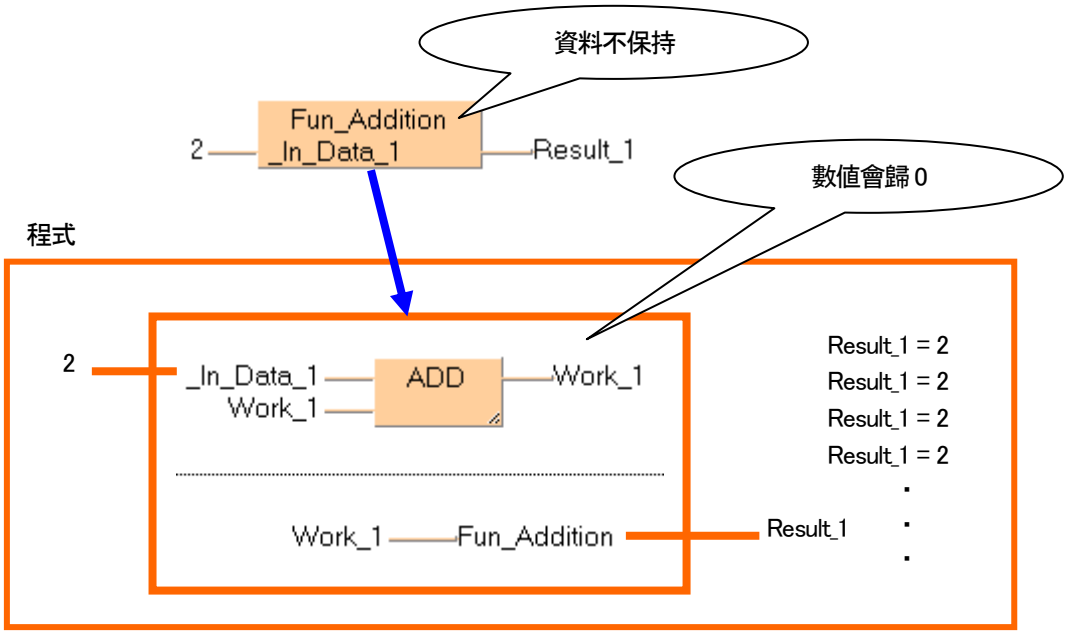
VAR_OUTPUT

只有在 Function Block 內所使用的輸出變數。PLC 從 PROG 模式切換到 RUN 模式時、電源 ON 時、VAR_OUTPUT 的初始值會被設定。

●Function 和 Function Block 的差異

Function(FUN)

程式中被呼叫後、執行一連串的處理並將結果傳回。 在 Function 中、對於輸入的處理一下子就決定了、所以不會佔用到 Function 內部的記憶體。



上圖為 In_Data_1 + Work_1 = Work_1 的計算 Function 所作成的範例、
因 Function 雖然不會用到記憶體、所以進行加算的 Work_1 每次都會歸“0”。

當輸入資料(_In_Data_1)為“2”的時候、

“2” “0” “2”

_In_Data_1 + Work_1 = Work_1

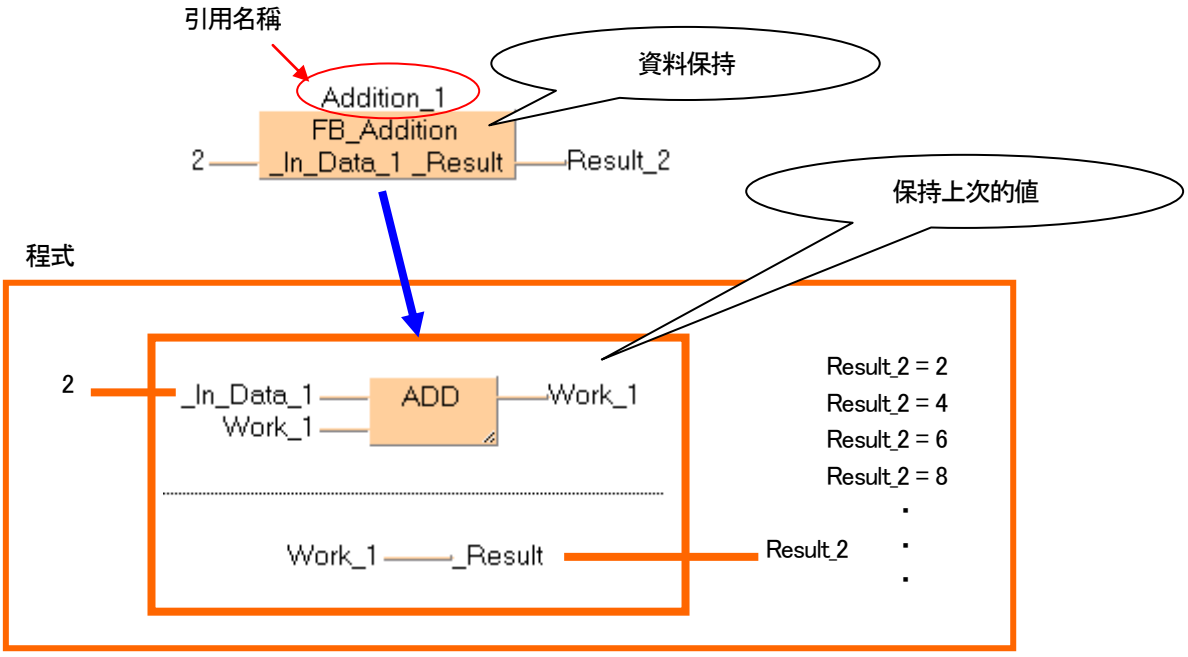
結果每次都會變成“2”。

換言之、因為 Function 不會用到記憶體的關係、對於需要使用計時器/計數器、定位的經過值等、需保持資料的情形是不適合使用的。(計時器/計數器使用、會在編譯時會發生錯誤)

而且、在 Function 內部不可以使用全域變數。

Function Block(FB)

雖然和 Function 同樣進行一連串處理、但 Function Block 自己本身有記憶數值的記憶體區域。因此、對相同的輸入值會產生不同結果。而且程式中、相同的 Function Block 可重複使用。 此時、呼叫各個 Function Blocki 時附上名稱(引用名稱)來區別。程式中可以配置若干、利用附上號碼來區別計時器指令是相同的概念。 呼叫 Function Block 的次數會受到 PLC 的 SUB 指令的數量有所限制。



上圖為 In_Data_1 + Work_1 = Work_1 的計算式是用 Function Block 所作成的範例、
因為 Function Block 為資料保持、所以會對每次累加計算的 Work_1 都會反映出上次的演算結果。

因此輸入資料(_In_Data_1)為 “2”的時候、

	_In_Data_1	+ Work_1	= Work_1
第 1 次	“2”	“0”	“2”
第 2 次	“2”	“2”	“4”
第 3 次	“2”	“4”	“6”
第 4 次	“2”	“6”	“8”
・		・	
・		・	
・		・	結果中可以反映出上次計算結果

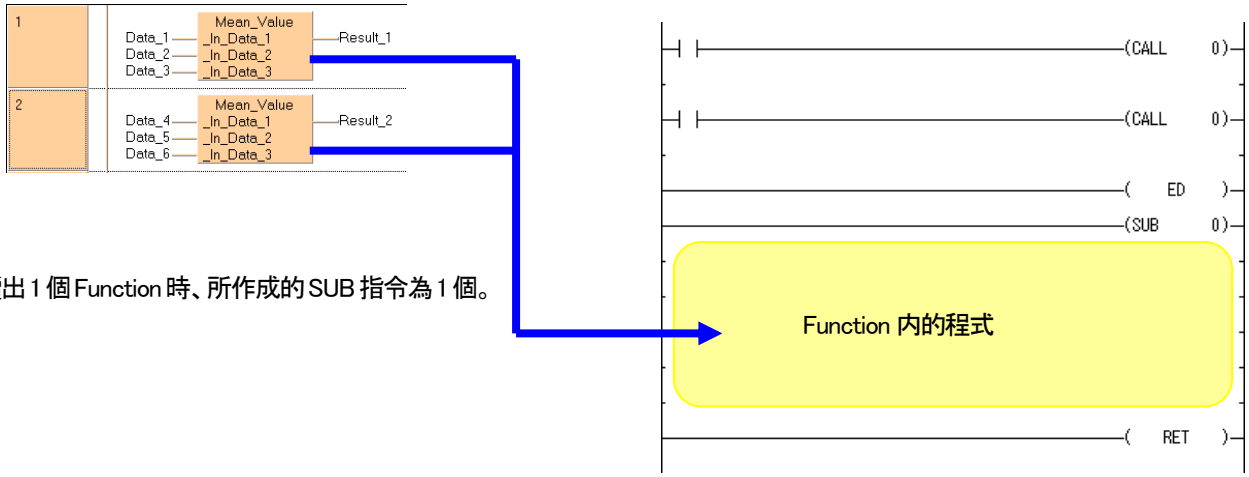
換言之、如計時器/計數器、定位經過值、需要資料保持等時、需使用功能區塊。

■Function(FUN)/Function Block(FB)的讀出次數

從程式可以讀取 Function 和 Function Block 的次數、依照 PLC 的機種來決定。

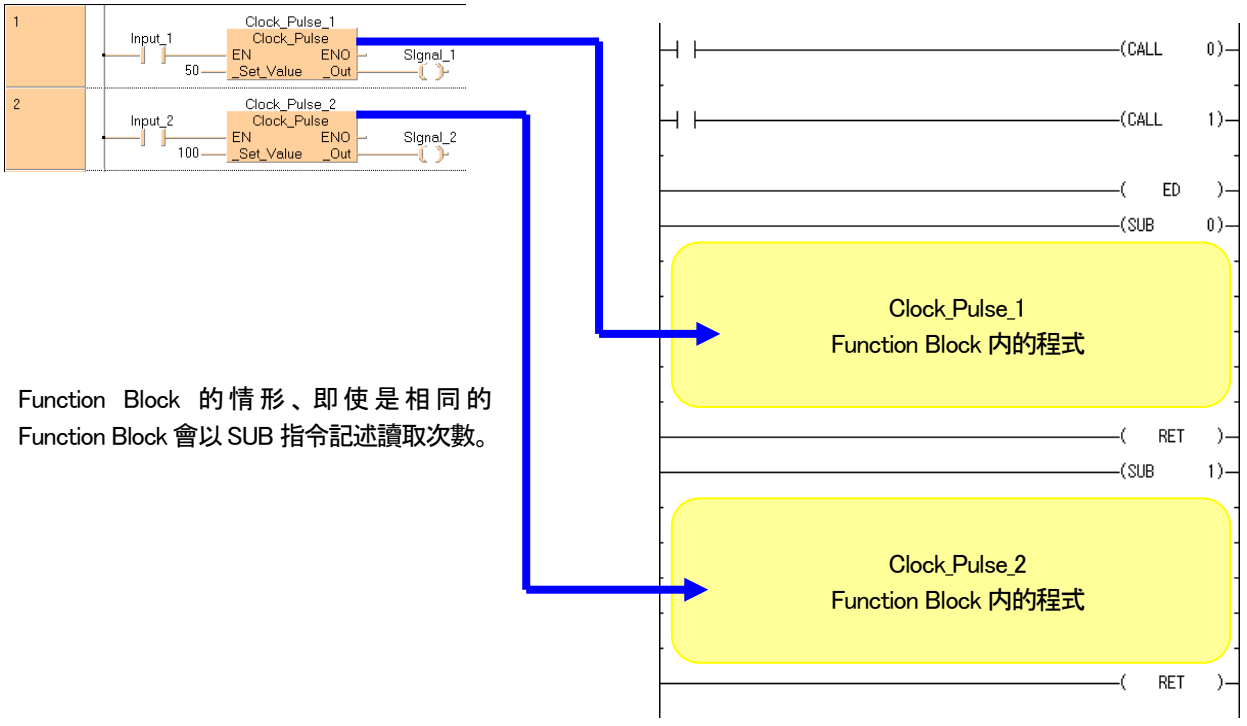
Function 和 Function Block 編譯後、為 SUB(副程式)指令來記述。
、此 SUB 所可以記述的次數就是讀取次數的限制。

●Function 的情形



讀出 1 個 Function 時、所作成的 SUB 指令為 1 個。

●Function Block



Function Block 的情形、即使是相同的 Function Block 會以 SUB 指令記述讀取次數。

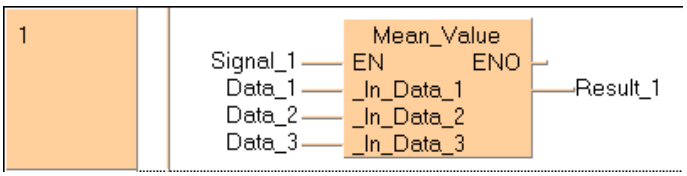
●主要 PLC 機種的副程式數量

	副程式數量
FP Σ 32K 型	500
FPX 全機種	500
FP2、FP2SH	100

8-2 製作 Function(FUN)

Function 的製作是以製作新的 Function（FUN）型的 POU 開始。其於的 Header、程式區都和之前方式編集。

程式(POU 型：PRG)的程式區畫面



如上圖「輸入 3 個數值(Data_1、Data_2、Data_3)後就會回傳平均值(Result_1)」的 Function 依照以下順序、使用階梯圖(LD)作成。

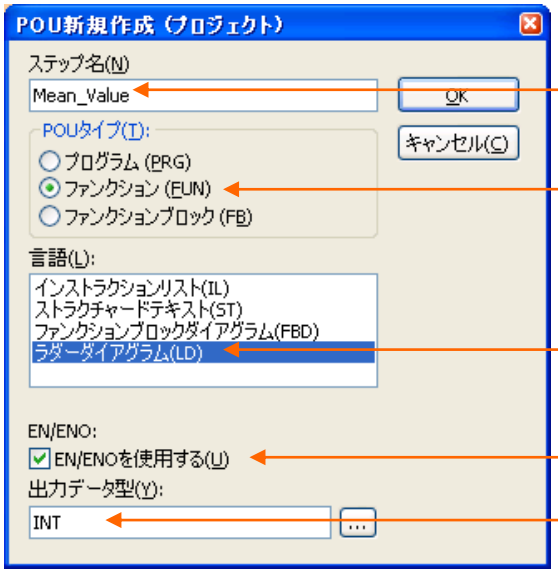
■操作順序

1. 製作新的 Function（FUN）型的 POU。

選擇 Function（FUN）作為 POU 型。

點選選單列  小圖示。

製作新 POU 的對話框。



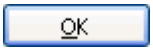
輸入 Function 名稱。

選擇 Function(FUN)。

選擇階梯圖(LD)。

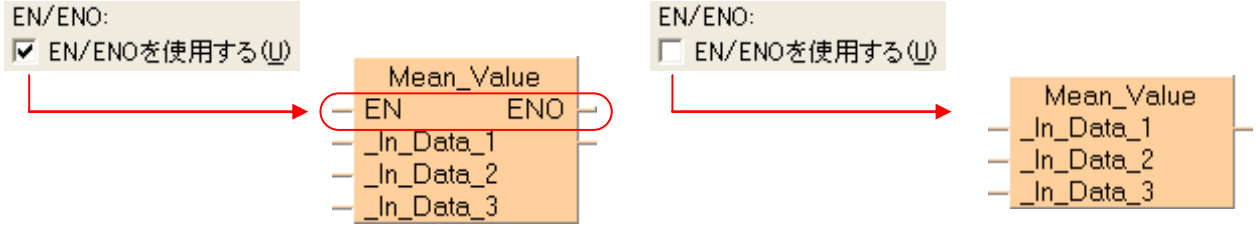
使用 EN/ENO 打勾。

指定 Function 的輸出資料型態為 INT。

設定好之後、請按下  按鍵。

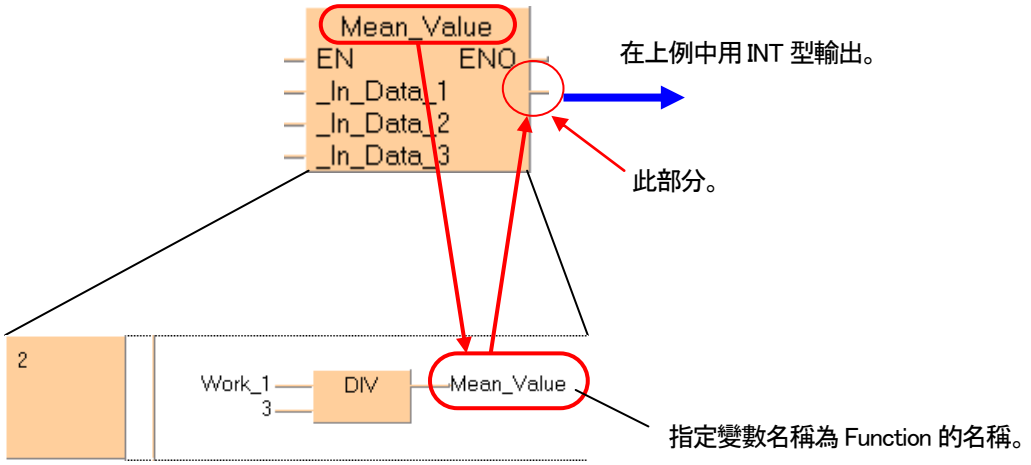
●關於 EN/ENO

指定「使用 EN/ENO」、會產生附有 EN/ENO 端子的功能。EN 是用於執行功能的觸發輸入、而 ENO 是用於輸出到下個 Function 的觸發。

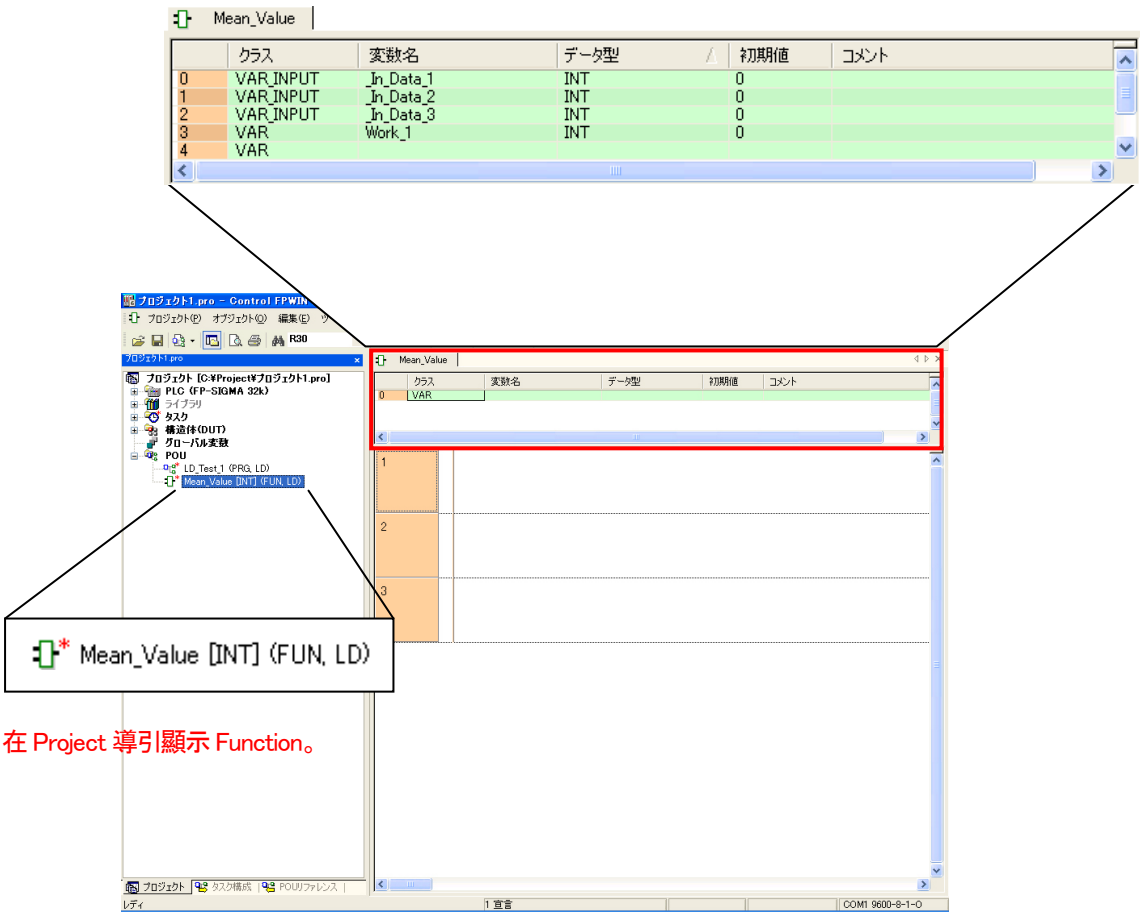


●輸出的資料型態

在 Function 中通過輸出的型態的指定、在 class 中也可以宣告 VAR_OUTPUT 型態的變數。

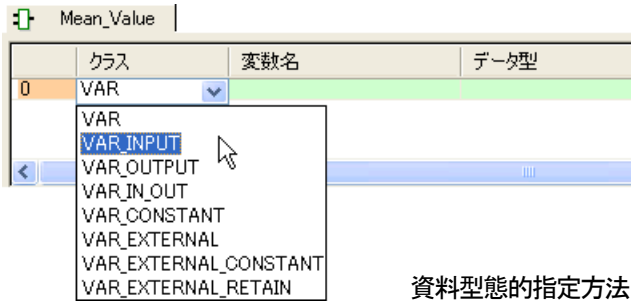


2. 如下圖在 Mean_Value(Function)的 POU – Header 中宣告任意變數。

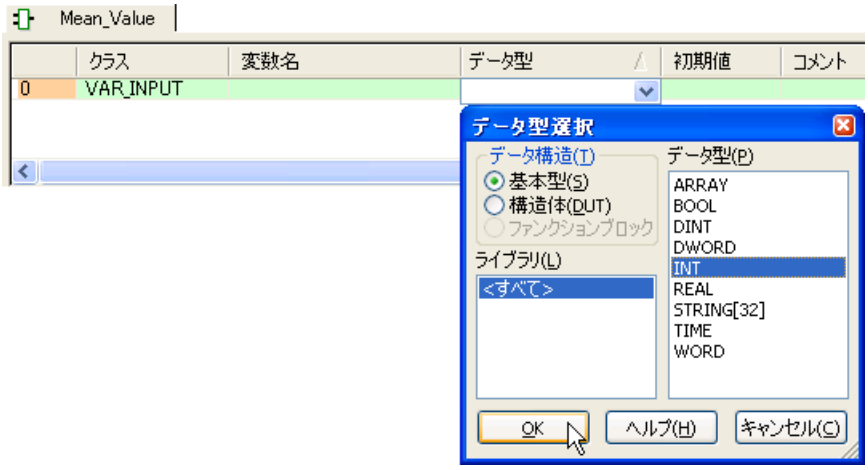


變數的等 class 和資料型態可以從下拉式選單中進行選擇。

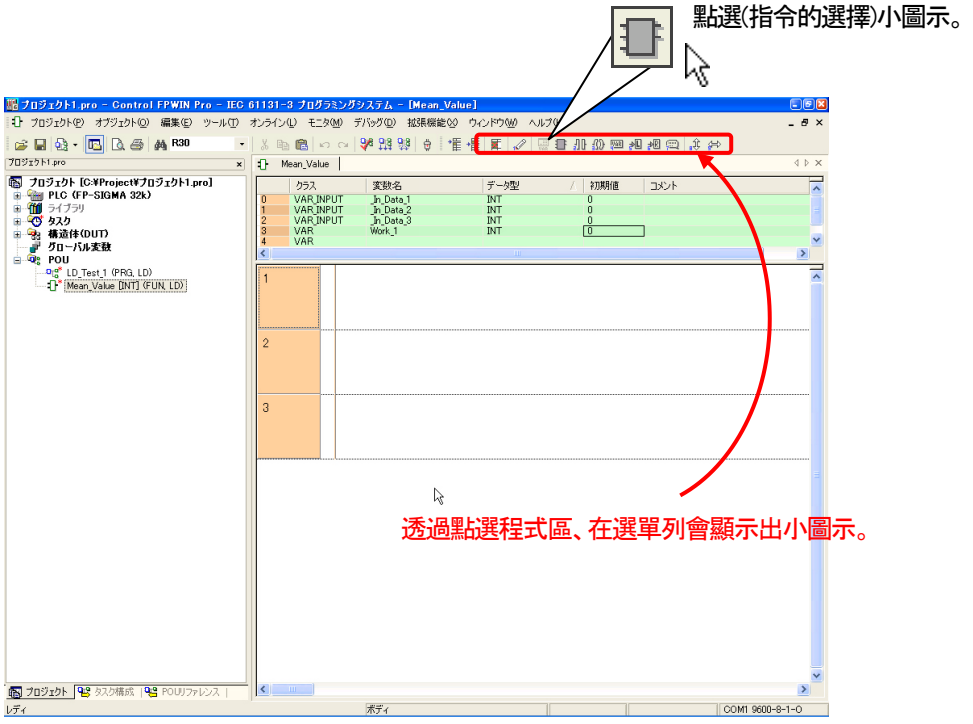
class 的指定方法



資料型態的指定方法



3. 點選程式區、後點選工具列上的 小圖示。



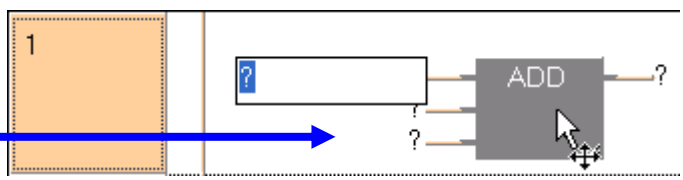
4. 顯示以下的對話框。

首先製作加總 3 個輸入值的 block、
選擇加總指令的 ADD。

5. 將「Pin 的數量」設定為「3」、按下 挿入(N) 鍵。

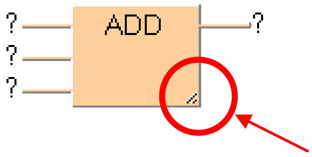
（Pin 的數量、配置到程式區後也可以變更）

配置到程式區。

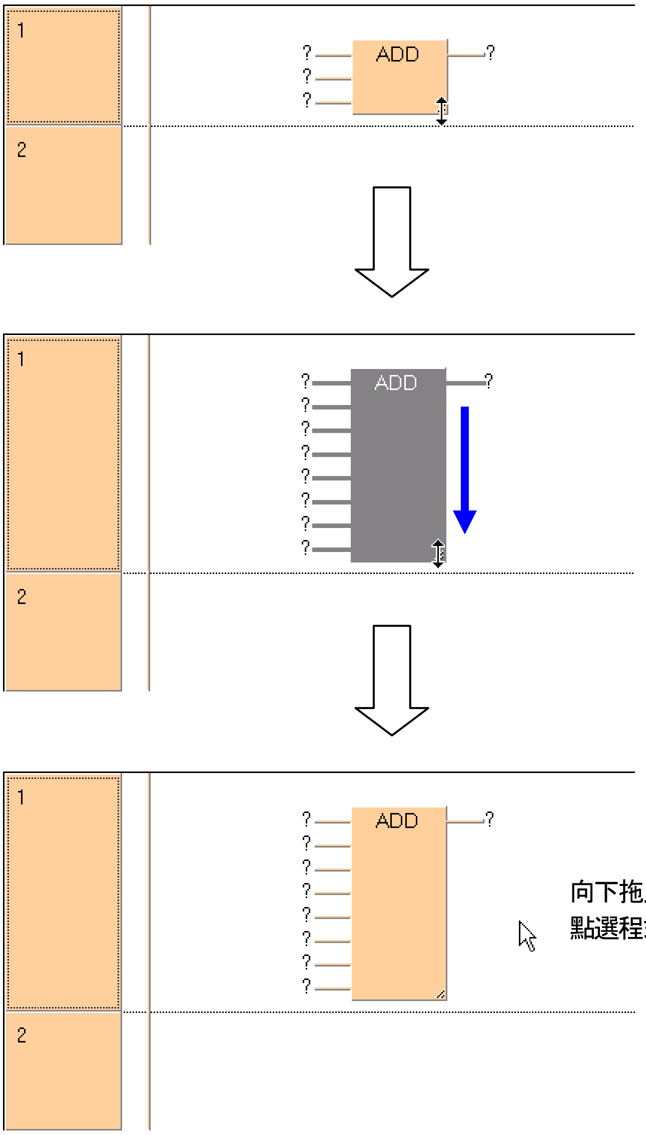


●参考

Pin 的數量、配置到程式區後也可以變更。

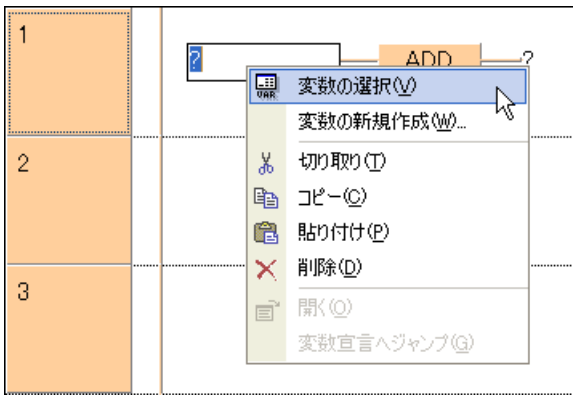


右下角的部分點選滑鼠並向下拖曳。

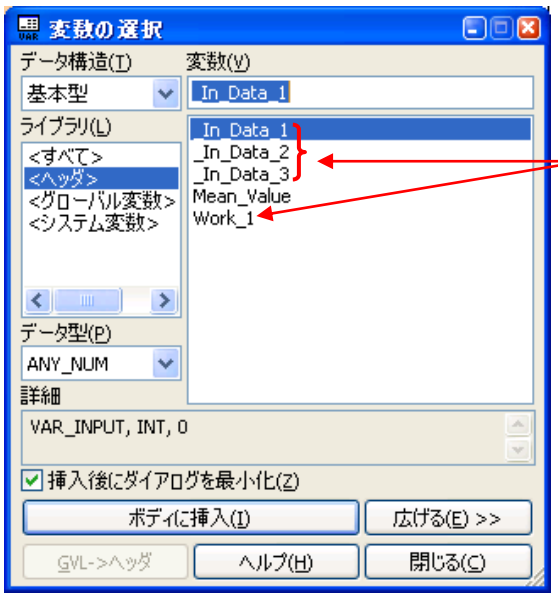


向下拖曳到目標 Pin 的數量為止、
點選程式區的空白部分。

5. 變數框用滑鼠點選使其中的"?"為反白狀態後、並按右鍵打開選單、點選「選擇變數」。

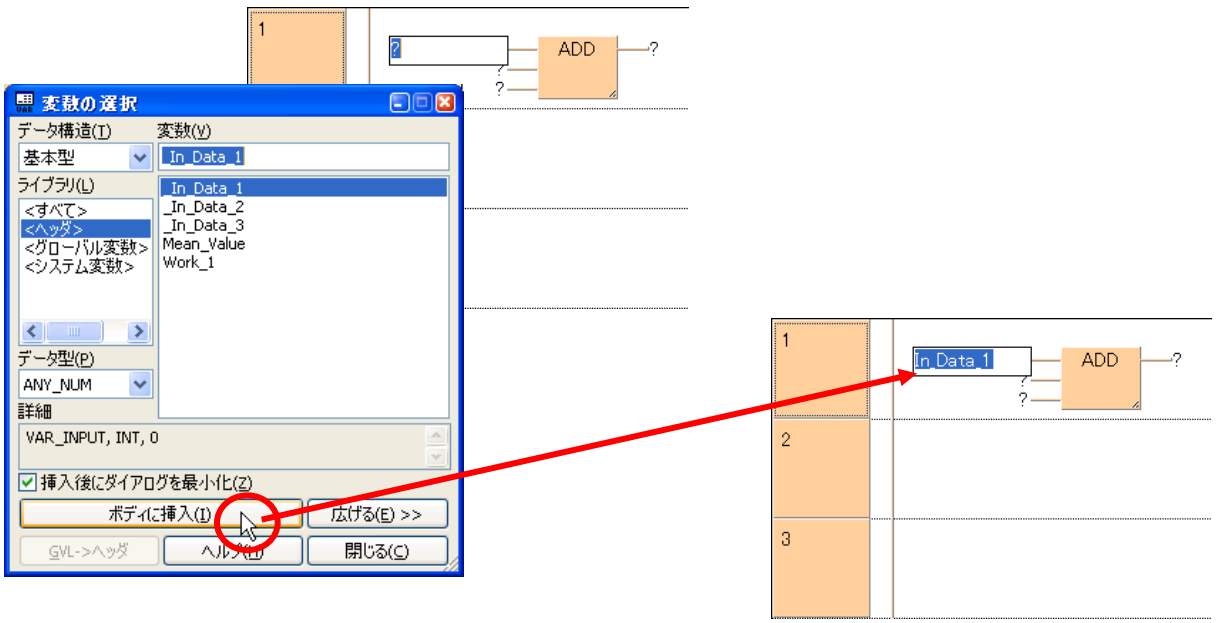


顯示以下的對話框。

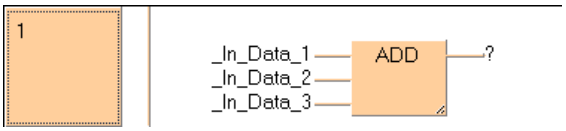


在 Header 會顯示出已經登録的變數。

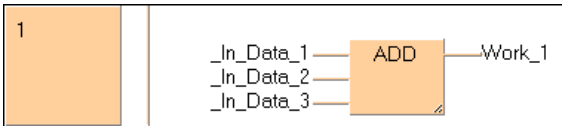
選擇"_In_Data_1" 的變數狀態下按 ボディに挿入(I) 鍵。



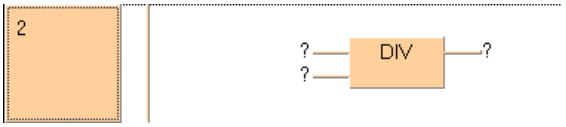
6. 用相同的步驟、將其餘的 2 個輸入值也插入變數。



7. 在此、從“_In_Data_1” 到 “_In_Data_3” 在輸入側、選擇_Work_1 作為 3 個輸入值總合的保存區域。



8. 接下來製作求平均值的 Function。跟 4. 相同的操作後出現對話框、
選擇功能「DIV」、插入到程式區。(請插入到下個區塊)

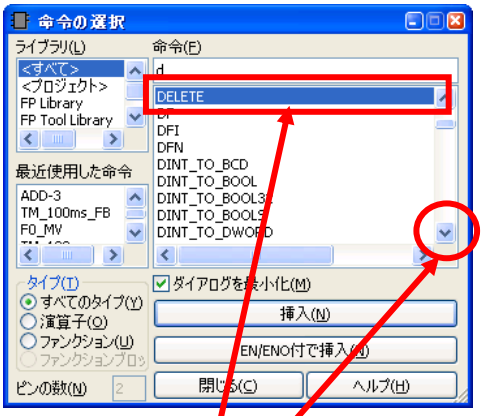


●參考

Function 的檢索方法

Function 會依照字母排序、在知道目標 Function 的情況下輸入 Function 的字首的字母、游標會移動到附近的 Function、可以有效率的選擇所要的 Function。
例如想選擇如上「DIV」的時候、

輸入字母的 “d”。
(不論大小寫)

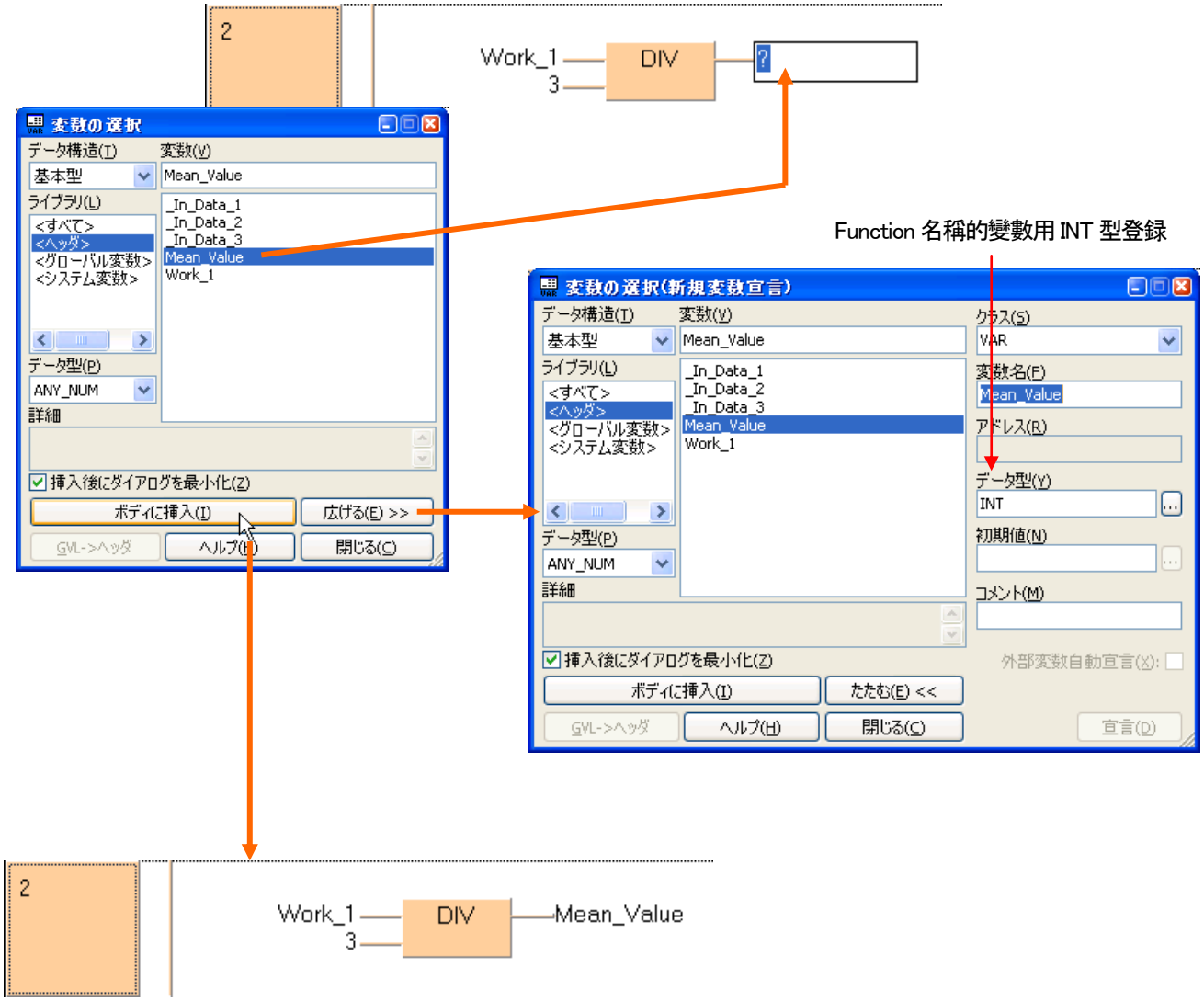


往

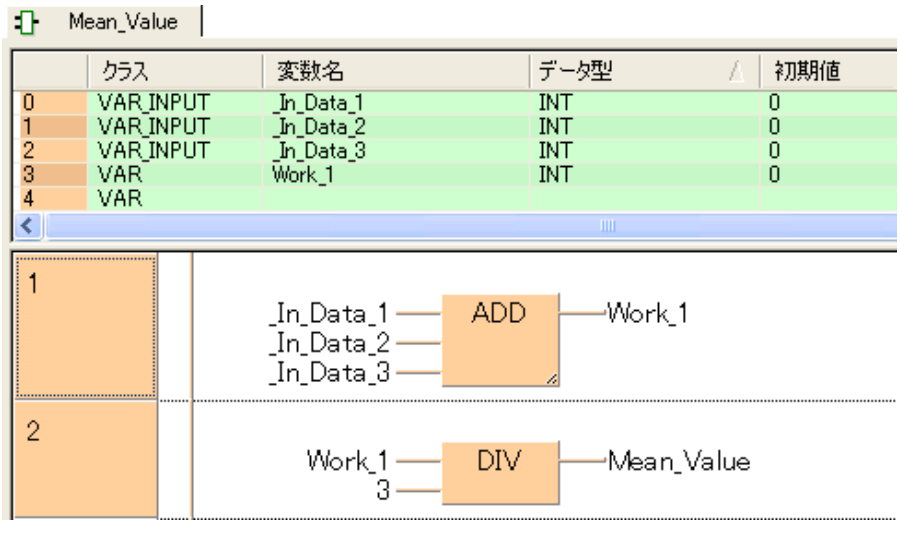
游標移動到從“D”開始的功能前頭“DELETE”。
之後用滾動軸往下移動就可以馬上找到 “DIV”。

9. 請依下圖方式配置變數到各個 Pin。

請將 Function 的輸出、請指定為 Function 名稱。輸出變數不必需在 Header 中定義。



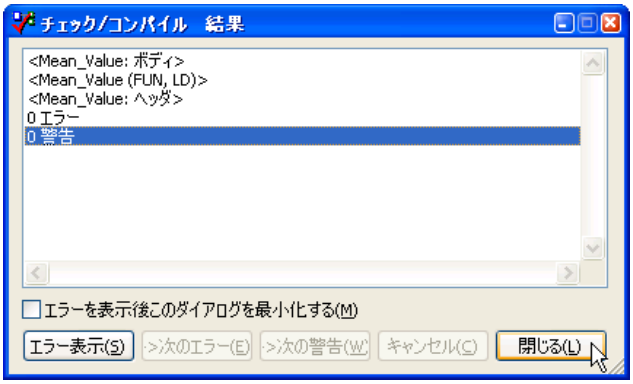
10. 以上就完成 Function 的程式。



11. 進行 Object 的檢查。

點選工具列上的「檢查 Object」小圖示來執行。

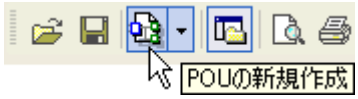
檢查結果對話框

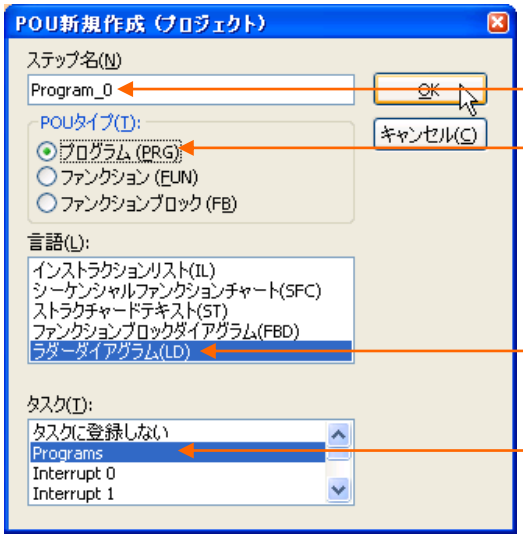


如果顯示“0 錯誤”的話、Function 的編輯工作結束。

12. 讀取程式(POU 類型:PRG)

那麼將編輯完成的 Function 讀取到程式中。
首先請用 “製作新 POU ”來製作程式(程式語言 LD)。





輸入 Project(程式)名稱。

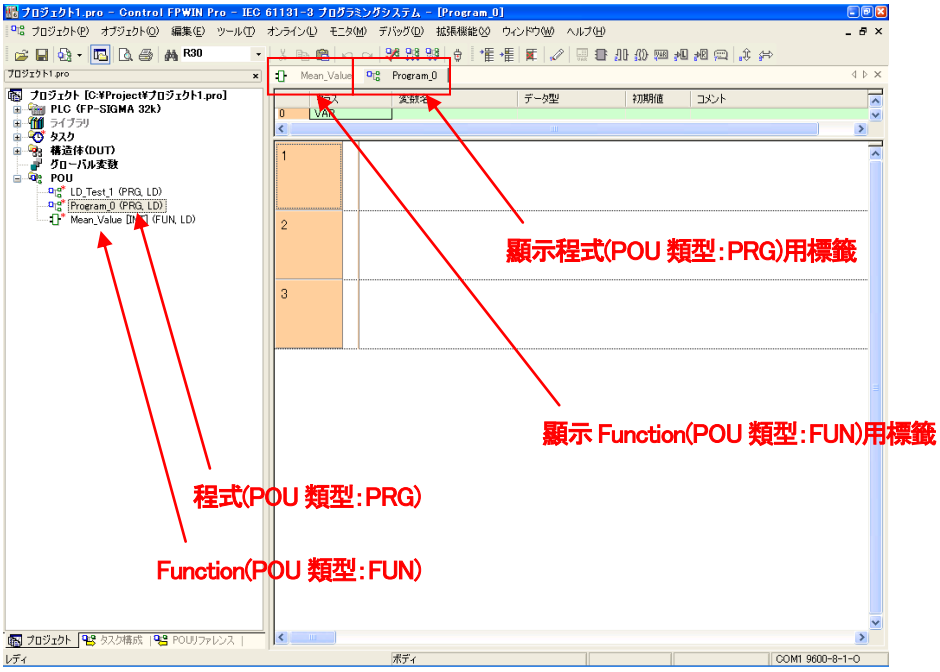
POU 類型設定為程式(PRG)。

程式語言設定為階梯語言(LD)。

Task 為 Programs。

輸入和設定完成後、請按下 OK 鍵。

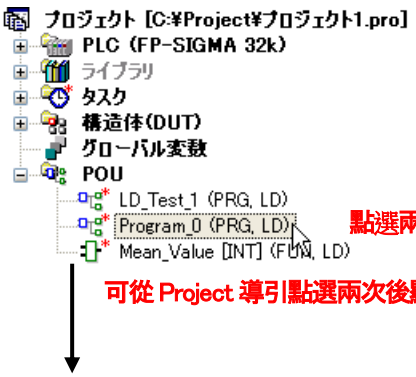
13. 從程式(POU 類型:PRG)讀出 Function(POU 類型:FUN)。



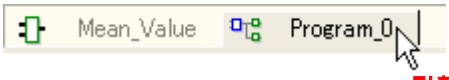
從程式(POU 類型:PRG)讀出 Function(POU 類型:FUN)。

從 Project 導引點選兩次程式(POU 類型:PRG)、或是點選程式(POU 類型:PRG)標籤後會顯示出程式區。

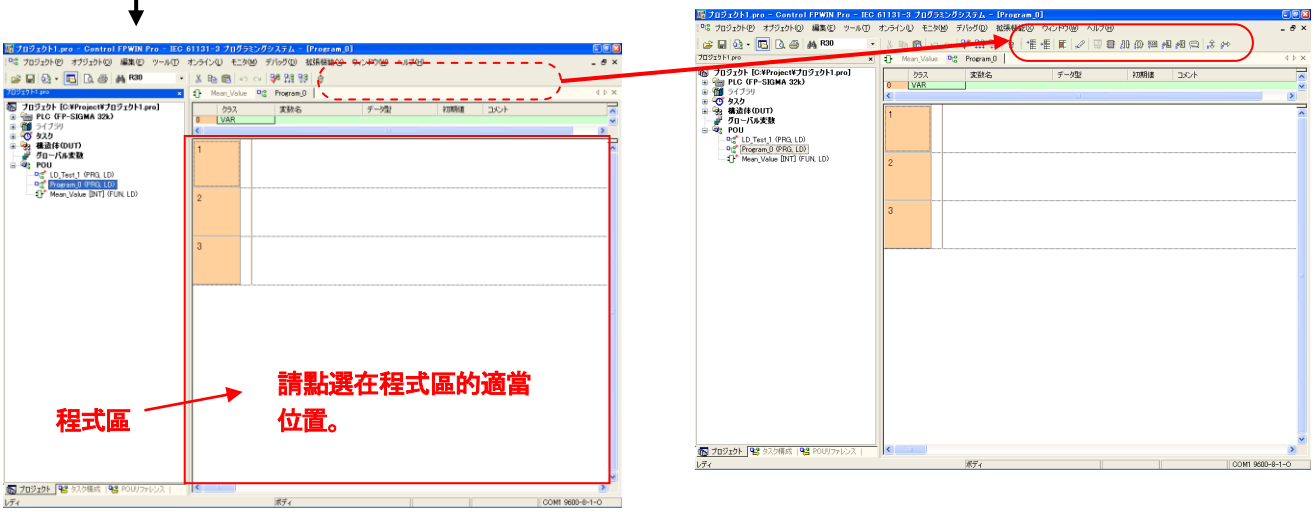
Project 導引



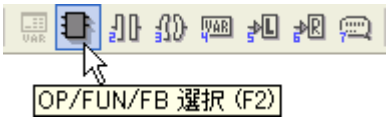
程式(POU 類型:PRG)標籤



顯示小圖示

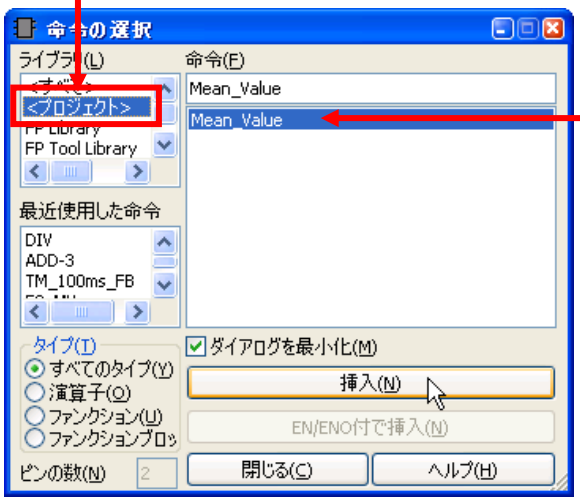


點選工具列的「指令的選擇」小圖示。



畫面將顯示 Function 選擇的對話框、選擇剛才完成的 Function。

Library：選擇<Project>

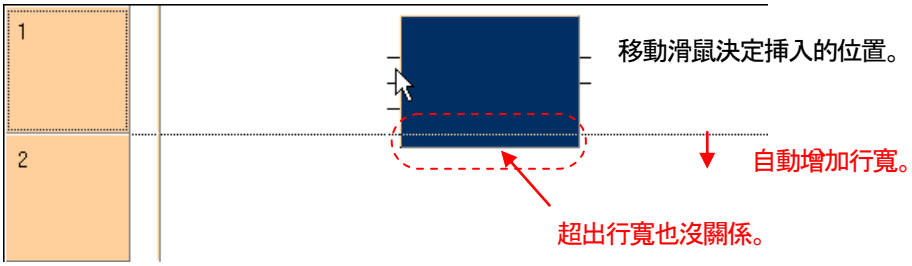


會顯示製成的 Function。

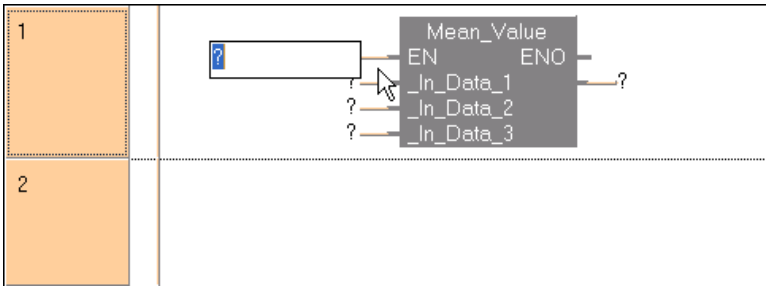
選擇完成後按下 

鍵。

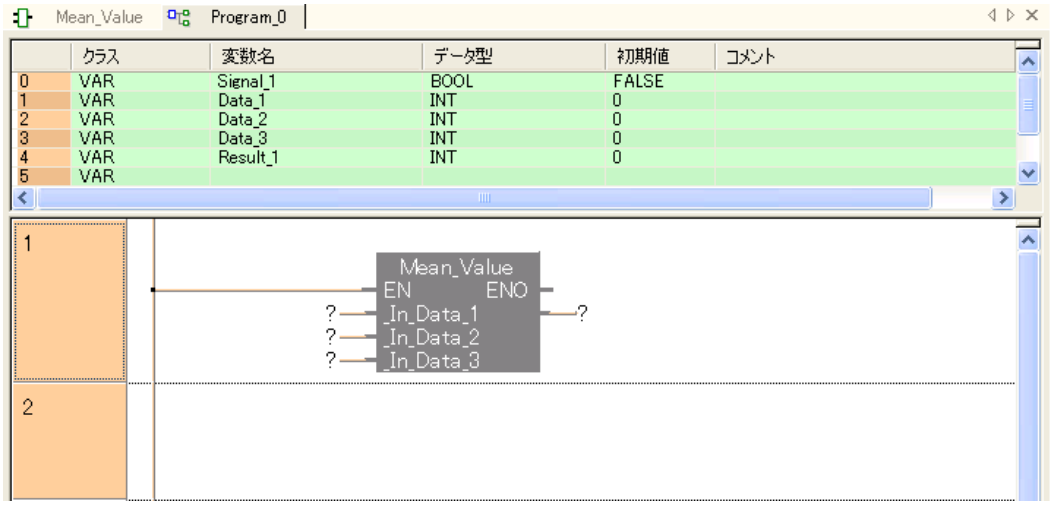
把功能貼到 Network1 的適當位置。



決定位置後、請按左鍵確定。如以下圖示。



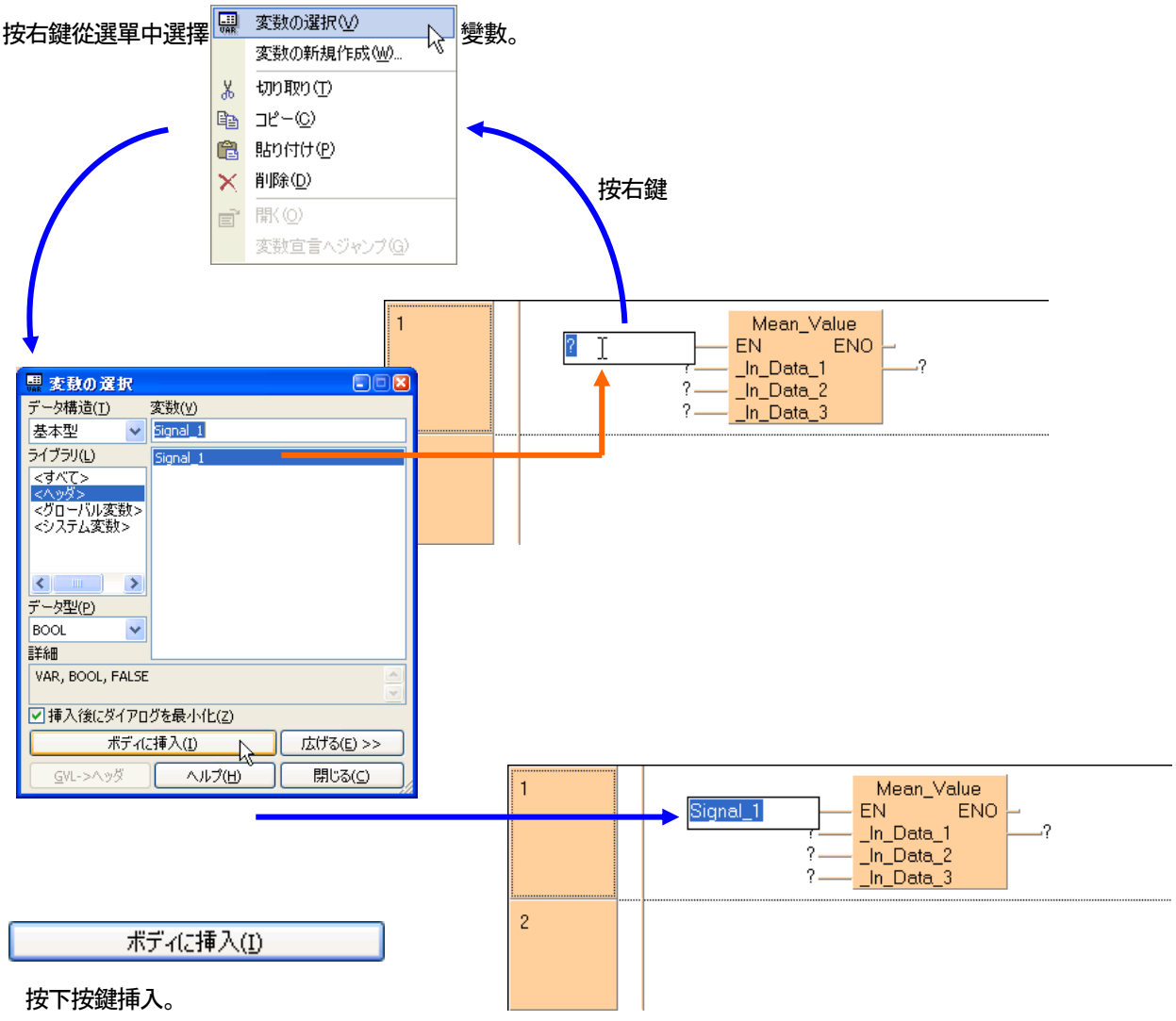
14. 如下圖所示在程式的 POU – Header 中宣告任意變數。



15. 配置輸入用的變數和輸出用的變數。

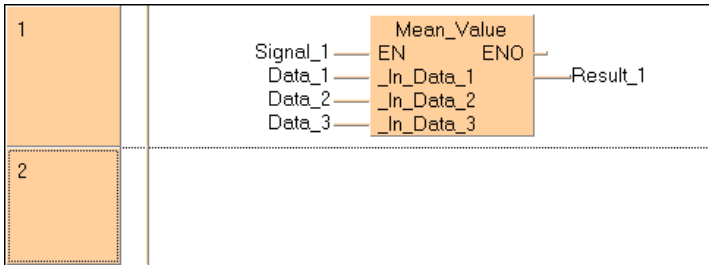
變數框按左鍵在 [?] 的狀態下、

按右鍵從選單中選擇 變數。



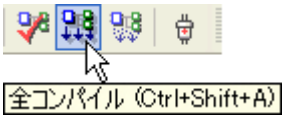
按下按鍵插入。

16. 以下相同方式插入其餘變數。



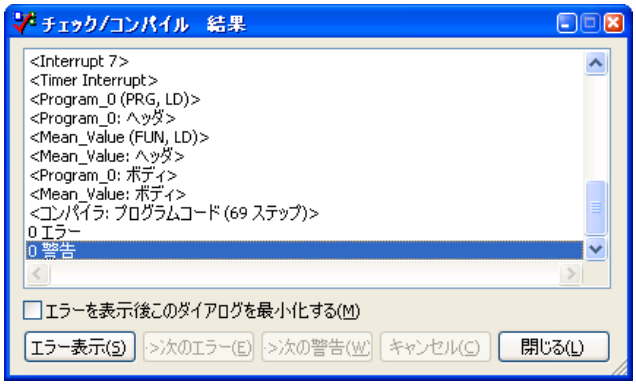
到此、插入到程式(POU 類型:PRG)的 FUnction(POU 類型:FUN)已經結束。

17. 進行 Project 全體編譯。



點選工具列的全編譯小圖示執行。

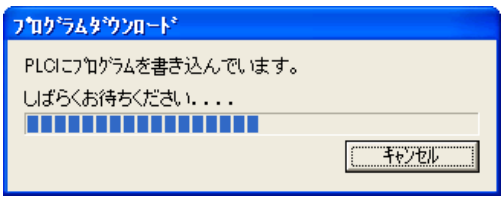
全編譯結果對話框



顯示出"0 錯誤"的話、Project 的編譯工作結束。

■動作確認

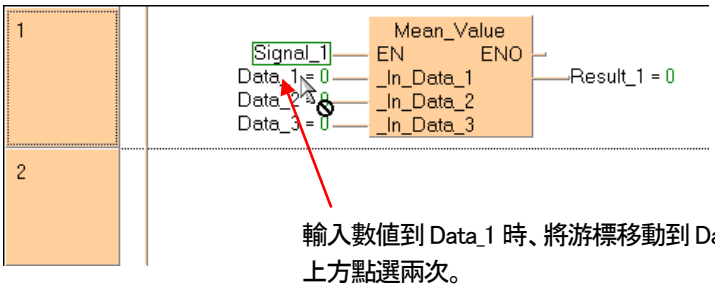
1. 下載 Project。



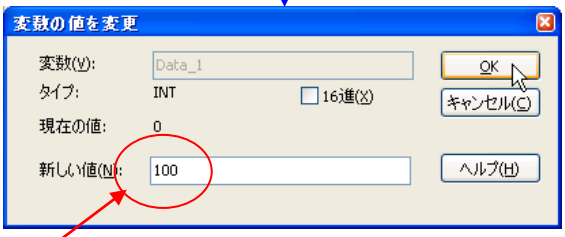
2. 確認 Function 動作。

請確認為變成監視狀態。

輸入數值到 Data_1、Data_2、Data_3。

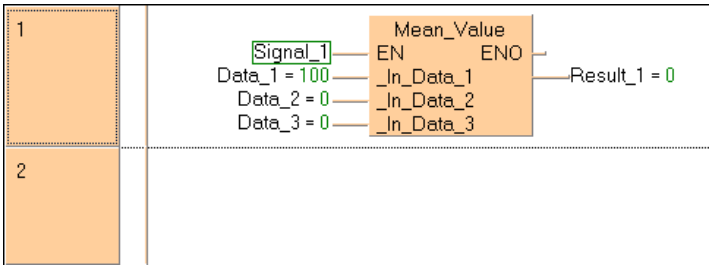


輸入數值到 Data_1 時、將游標移動到 Data_1 的上方點選兩次。

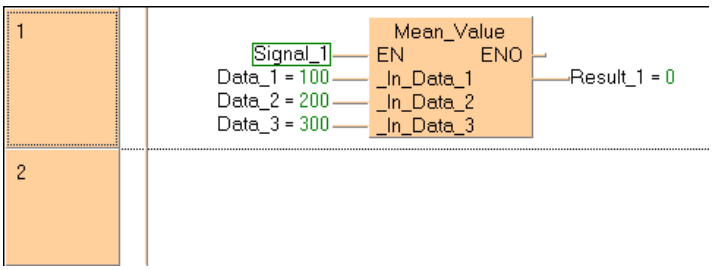


會顯示出資料輸入對話框、請輸入數值(上例為“100”)。

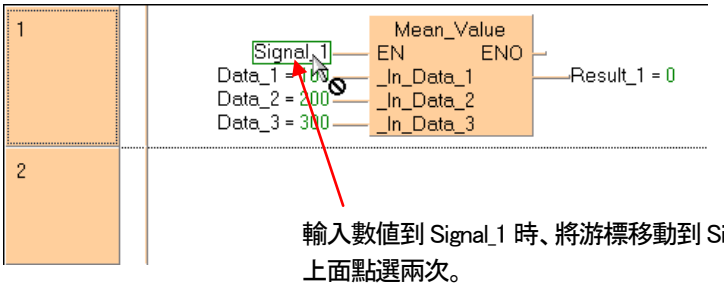
按下 OK 鍵確定。



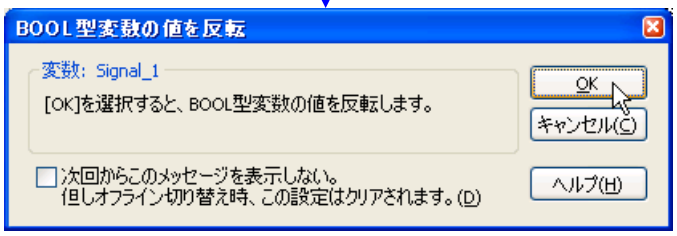
用相同步驟、輸入值到 Data_2 和 Data_3。

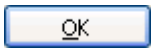


將執行條件 ON。

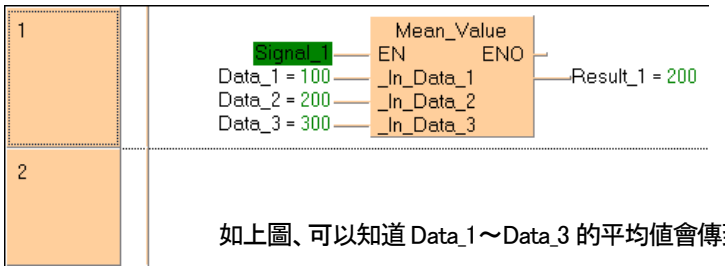


輸入數值到 Signal_1 時、將游標移動到 Signal_1 的上面點選兩次。



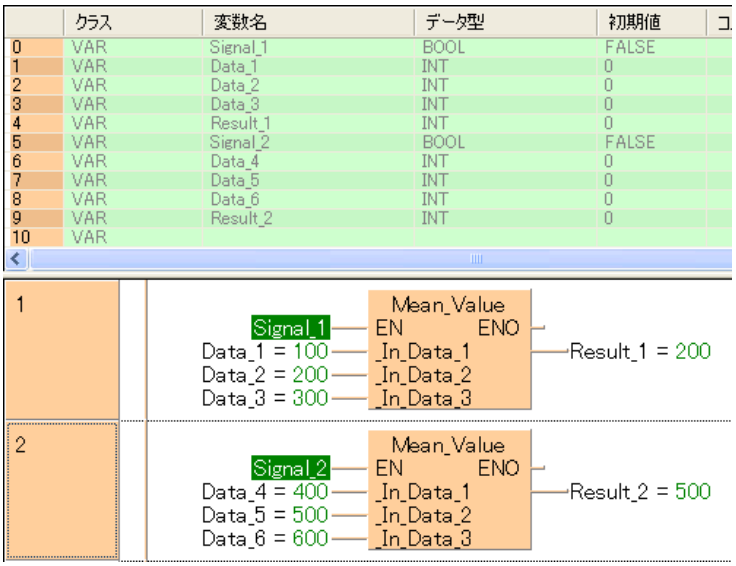
按下 

鍵。



如上圖、可以知道 Data_1～Data_3 的平均值會傳到 Result_1。

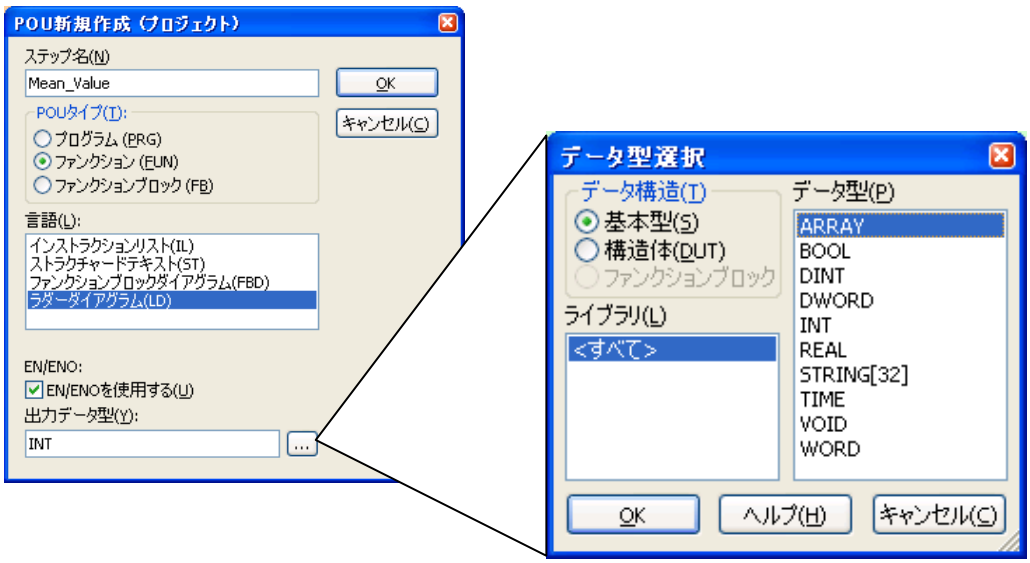
●備註

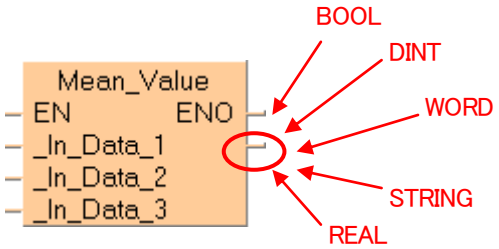


如上圖、使用相同的 2 個功能、對於不同的輸入、可以得到不同的結果。
像這樣內部沒有記憶體、對輸入進行演算再將結果輸出時會、使用 Function。
雖然 Function Block 也可以做到、但在使用 SUB 指令的同時會增加程式的 STEP 數量。

VOID 型的 Function(FUN)

雖然前述製作「輸出資料類型」以 INT 型態的 Function 製作成、
當然可以指定「輸出資料類型」INT 型態以外的型態輸出。





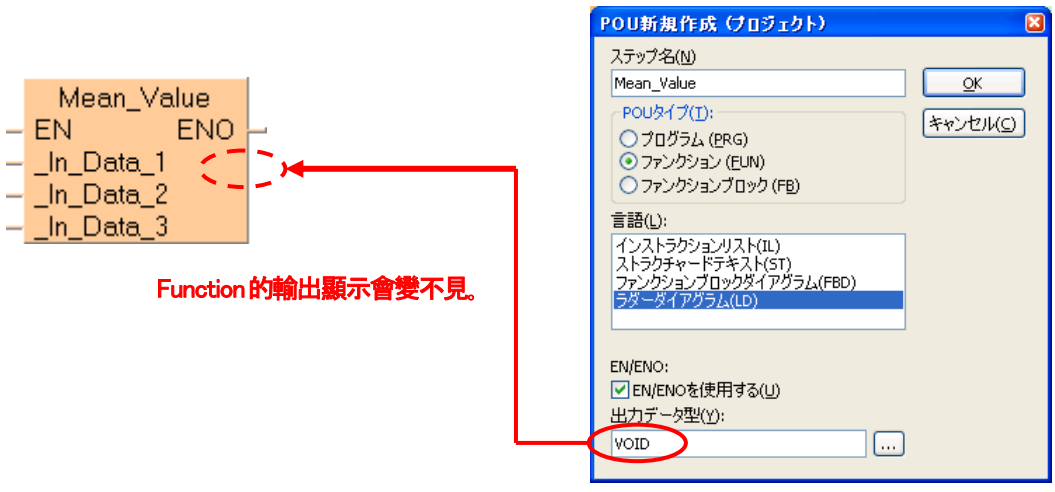
※也可以指定資料結構(DUT)、陣列

那麼此 Function 的輸出一定要使用嗎・・・

①不需要輸出。
②輸出想附上名稱。
③輸出有 2 個以上時。

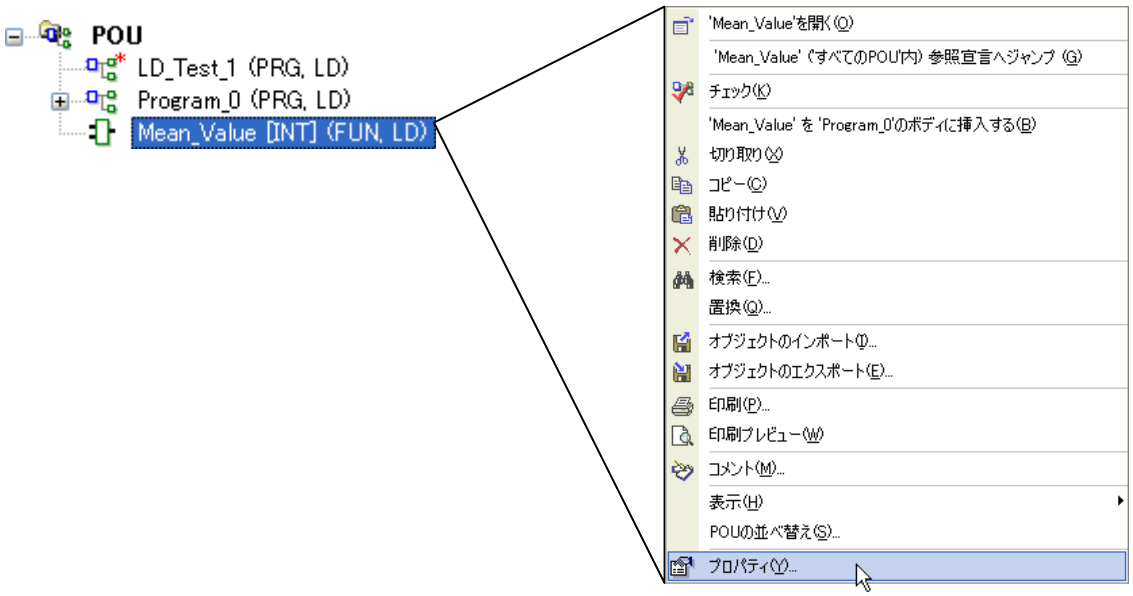
有等等理由、有時候想把此 Function 的輸出刪除時。

Function 的輸出、將「輸出資料類型」指定成 VOID 即可不使用。



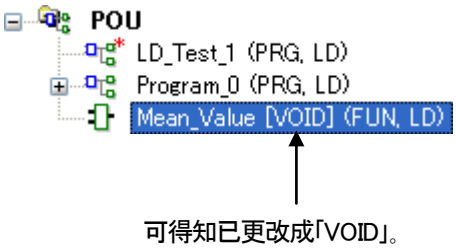
那麼將「輸出資料類型」使用 VOID、型態並修改成新 Function“Mean_Value”看看。

選擇 Project 引導的 Function“Mean_Value”、並具在上方按右鍵從選單選擇內容。



顯示「POU 內容」的對話框。

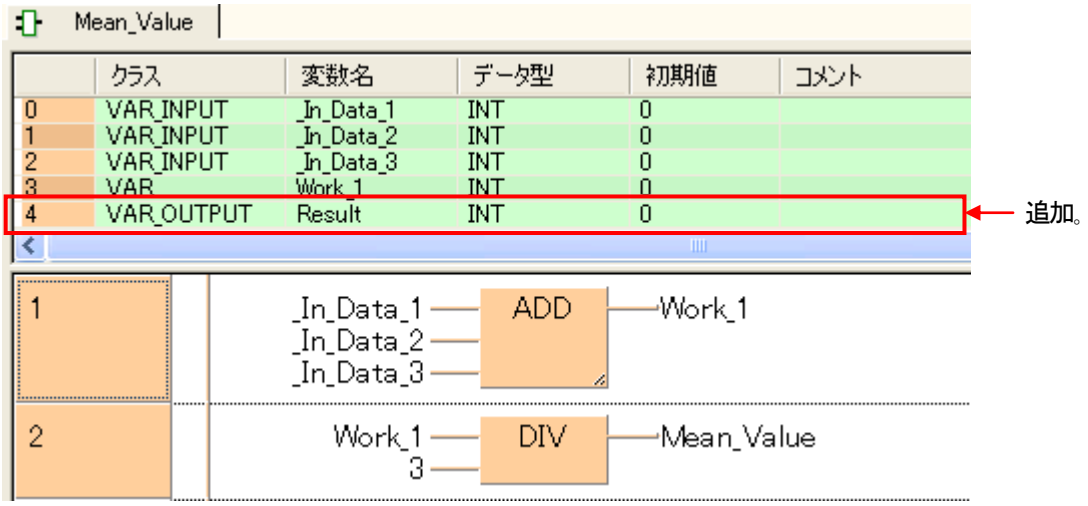
請用 Project 導引來確認「輸出資料類型」是否變更為 VOID。



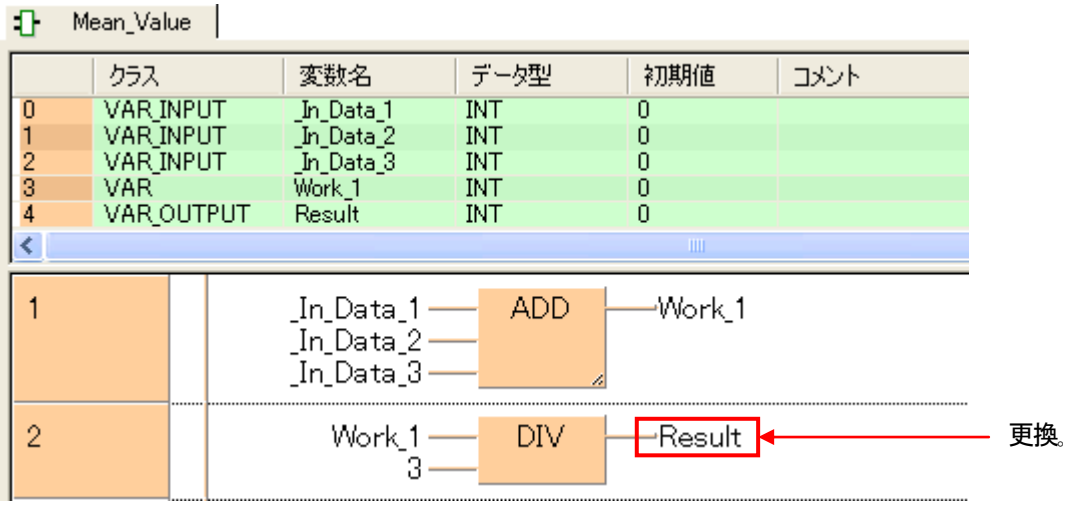
可得知已更改成「VOID」。

接著更改 Function“Mean_Value”。

請追加變數“VAR_OUTPUT”到 Header。

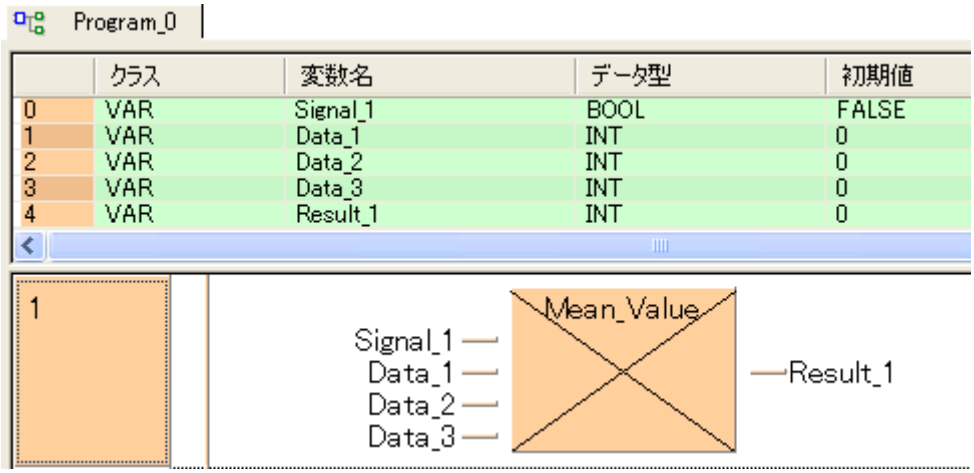


將結果的“Mean_Value”更改成為追加變數的“Result”。



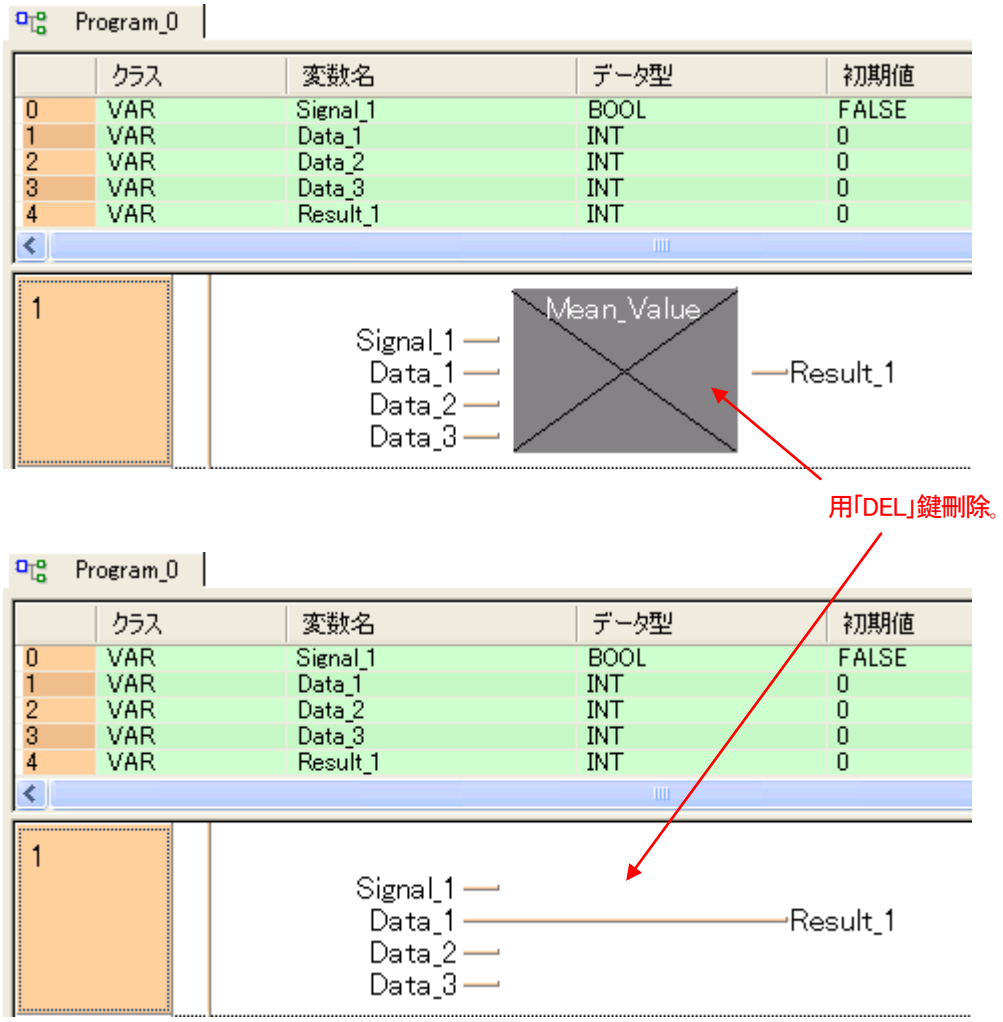
那麼來看看插入到 Function“Mean_Value”的 POU。

“Mean_Value”的輸出變數因為變更、所以 Function 的形狀也此改變、
在 Function 上面會有”X”印、顯示無法使用。

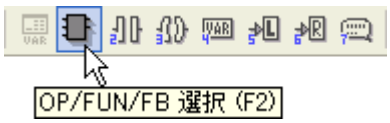


重新插入 Function“Mean_Value”。

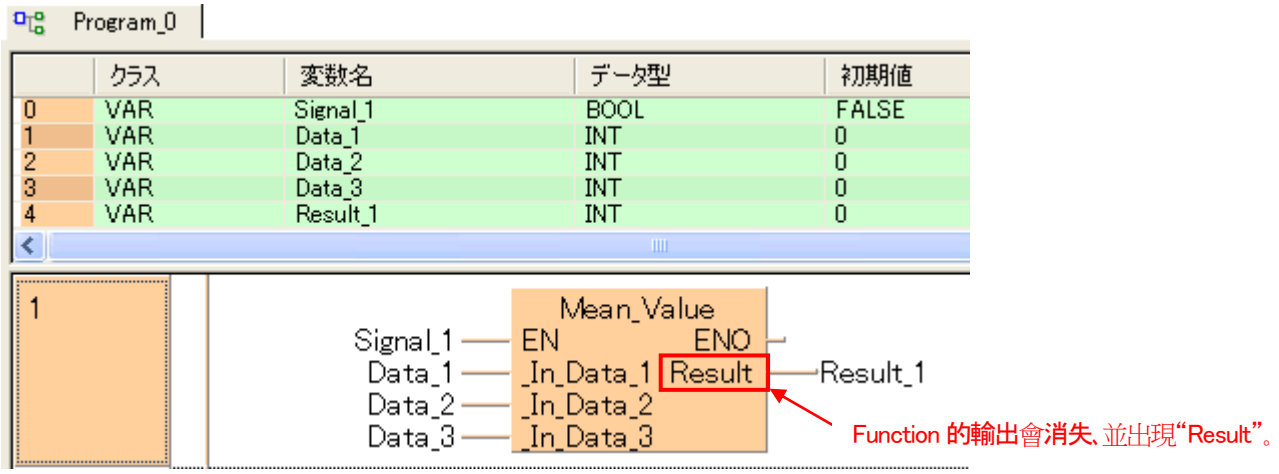
選擇附上有”X”印的 Function“Mean_Value”、用「DEL」鍵刪除。



再次插入 Function”Mean_Value”。

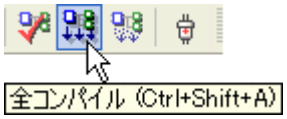


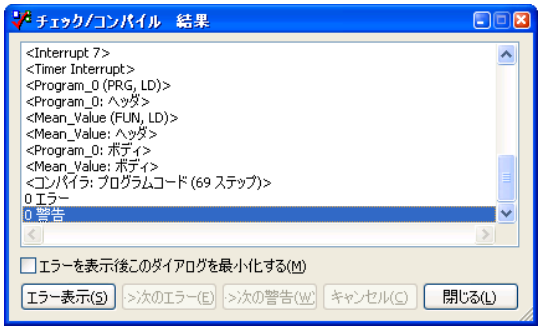
可以發現 Function 的輸出消失、並且“VAR_OUTPUT”會變成變數“Result”。

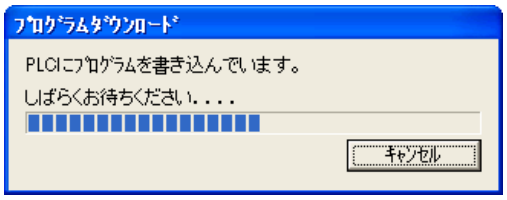


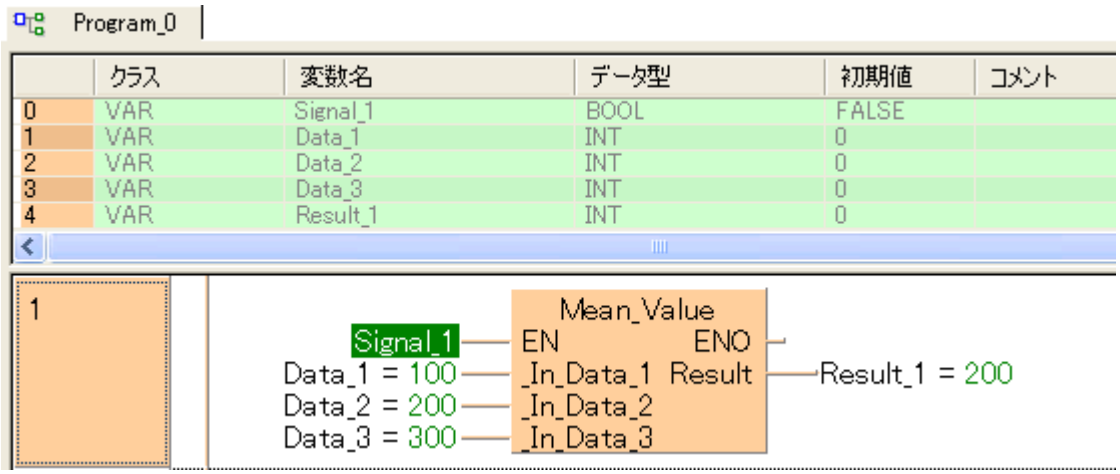
實際確認動作看看。

Function 編譯後、移動到 Online 下載到 PLC。







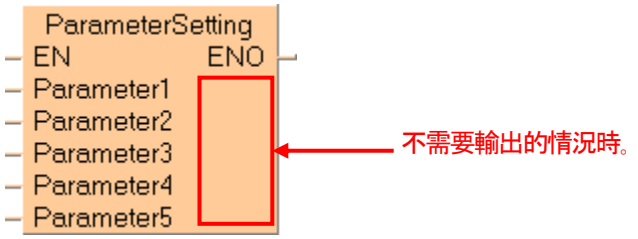


可以得知得到相同的結果。

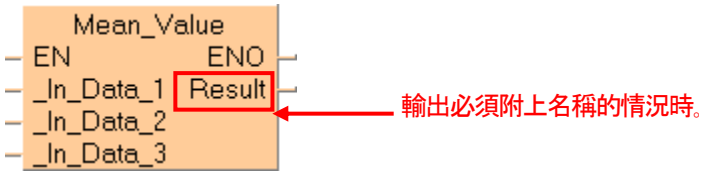
●備註

透過將「輸出資料類型」指定成 VOID 型態、

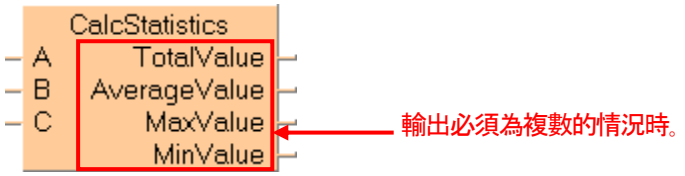
①不需要輸出。



②輸出想附上名稱時。



③輸出有 2 個以上。

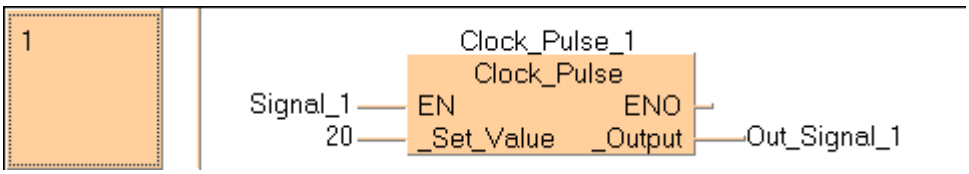


變得可以對應以上情況。

8-3 製作 Function Block(FB)

Function Block 的製作方法先從 Function Block(FB)型的 POU 新製作開始。而 Header、程式區都和之前是相同方式編集。

程式(POU 類型:PRG)的程式區畫面



使用以下的順序用階梯圖程式(LD)來製作如上圖「Signal_1 的信號 ON 時、在_Set_Value 所設定的時間(單位 0.1 秒)間隔、Out_Signal 會成重複 ON/OFF」的 Function Block。

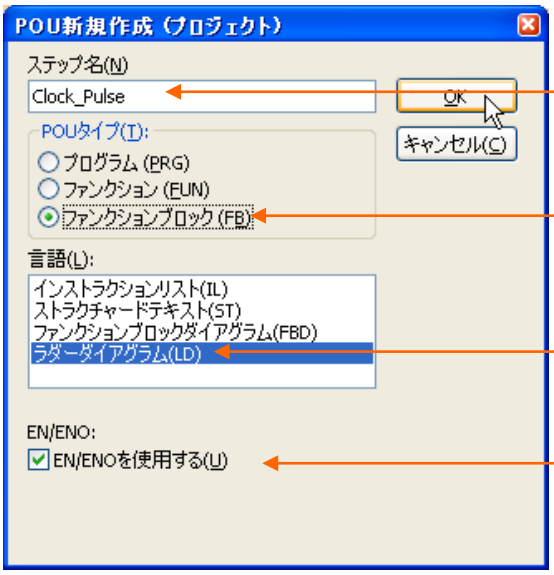
■操作順序

1. 製作新 Function Block(FB)型的 POU。

選擇 Function Block(FB)作為 POU 型態。

點選選單列上  小圖示。

製作新規 POU 的對話框。

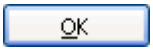


輸入 Function Block 名稱。

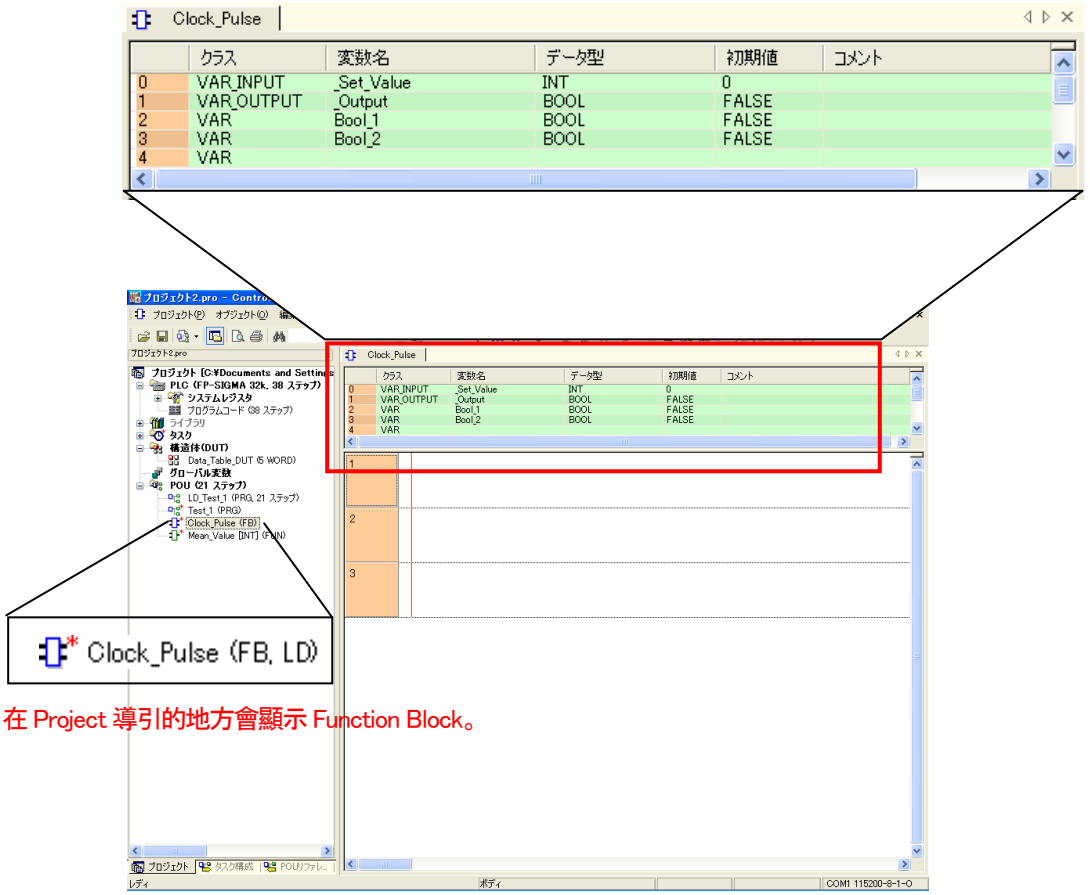
選擇 Function Block (FB)。

選擇階梯圖程式(LD)。

將使用 EN/ENO 打勾。

設定結束後、請按下  鍵。

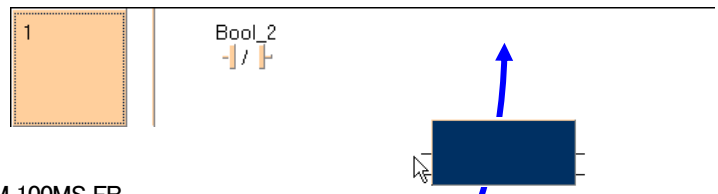
2. 如下圖在 Clock_Pulse(FB)POU 的 Header 中宣告任意變數。



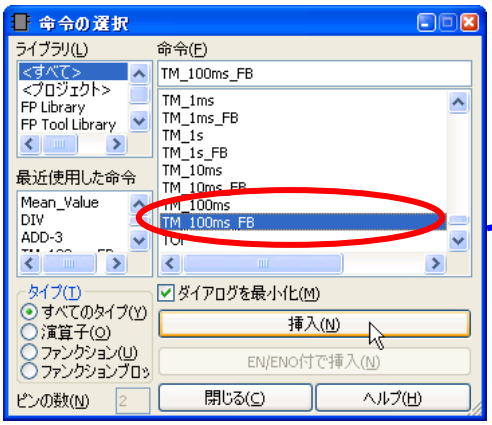
在 Project 導引的地方會顯示 Function Block。

3. 輸入程式。

下圖在輸入接點的狀態下、按下工具列上的 (指令的選擇)小圖示。

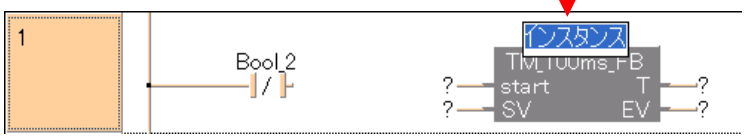


選擇 0.1 秒計時器的 TM_100MS_FB。

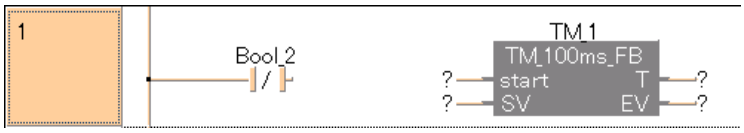


按下插入按鍵、貼到程式區。

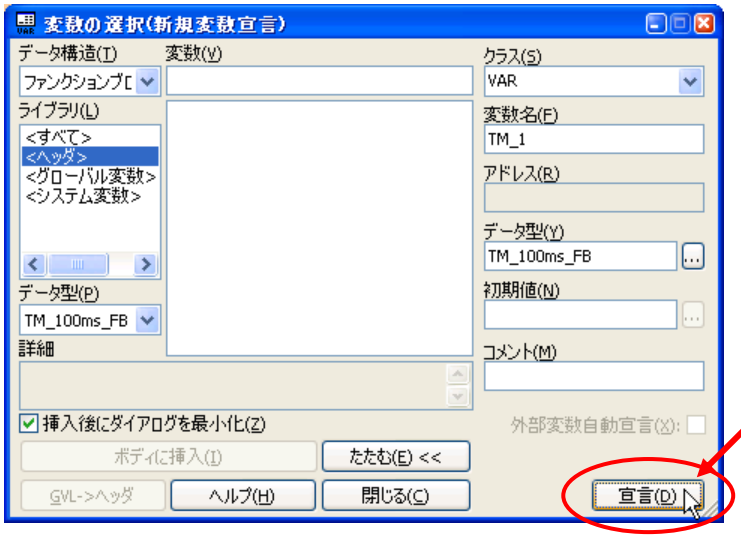
FB 貼入之後會要求輸入引用名稱。



輸入引用名稱。
在此輸入為 TM_1。



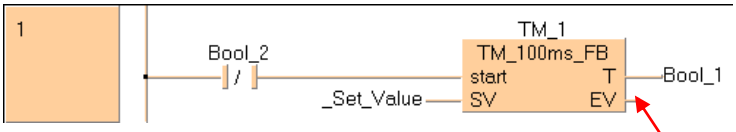
引用名稱輸入結束的同時、會顯示登錄變數對話框。



就這樣直接按下宣告鍵。

輸入的引用名稱「TM_1」會以變數會被自動登録在 Header 中。

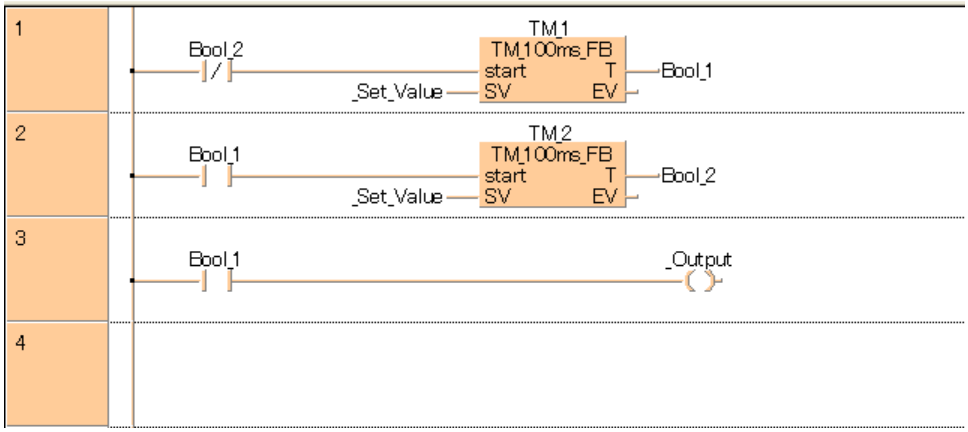
成功插入計時器後的話、請完成下圖的迴路。



輸出經過值。
因為本次沒有使用、請刪除變數框。
錯誤不會發生。

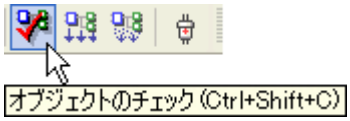
請同樣的完成下圖迴路。

	クラス	変数名	データ型	初期値	コメント
0	VAR_INPUT	_Set_Value	INT	0	
1	VAR_OUTPUT	_Output	BOOL	FALSE	
2	VAR	Bool_1	BOOL	FALSE	
3	VAR	Bool_2	BOOL	FALSE	
4	VAR	TM_1	TM_100ms_FB		
5	VAR	TM_2	TM_100ms_FB		
6	VAR				



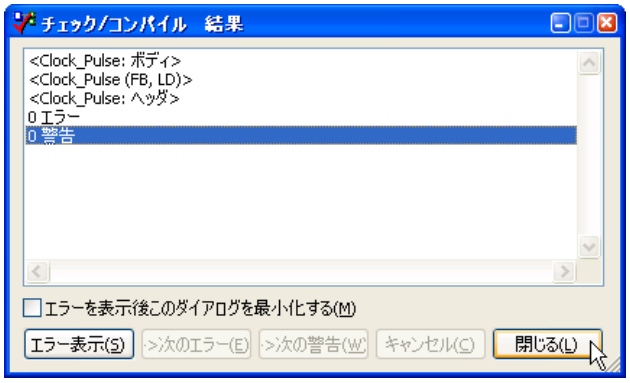
以上完成 Function Block(FB)的程式。

4. 進行 Project 的檢查。



點選工具列上的「Project 完成的檢查」的小圖示來執行。

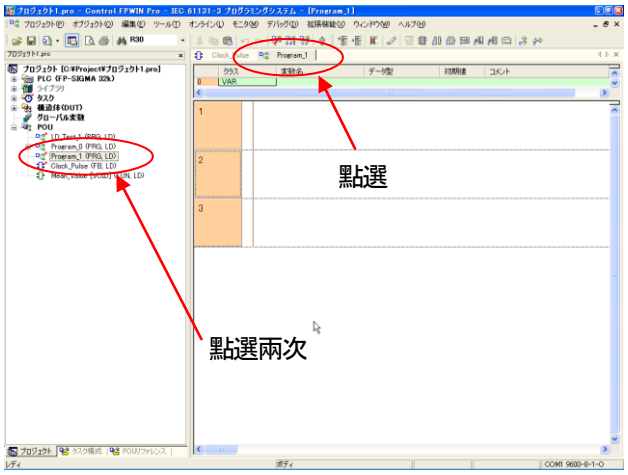
檢查結果對話框



如果顯示為“0 錯誤”、結束 Function Block 的編譯工作。

5. 讀取到程式(POU 類型:PRG)

打開程式(POU 類型:PRG)。

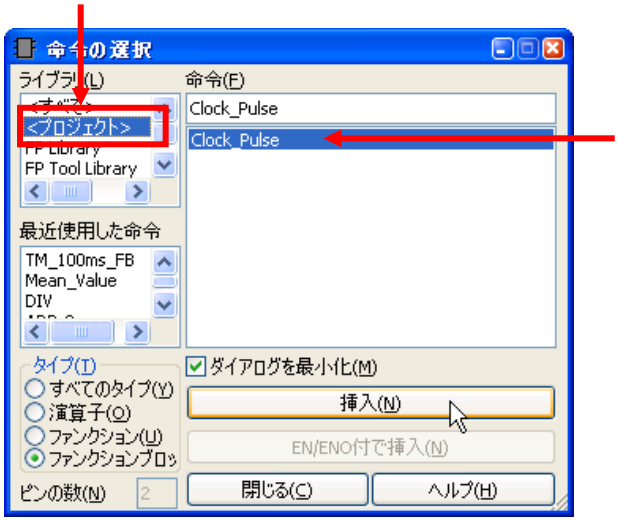


從 Project 導引點選兩次程式(POU 類型:PRG)、或是點選程式(POU 類型:PRG)的標籤、會顯示程式區。

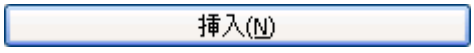
點選工具列的「指令選擇」小圖示。

有顯示 Function Block 的選擇對話框、選擇之前完成的 Function Block。

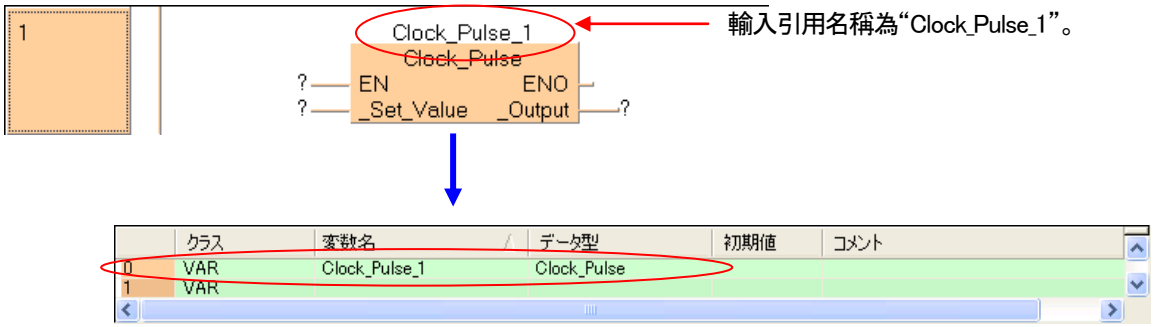
Library:選擇<Project>



顯示完成的 Function Block。

選擇後 

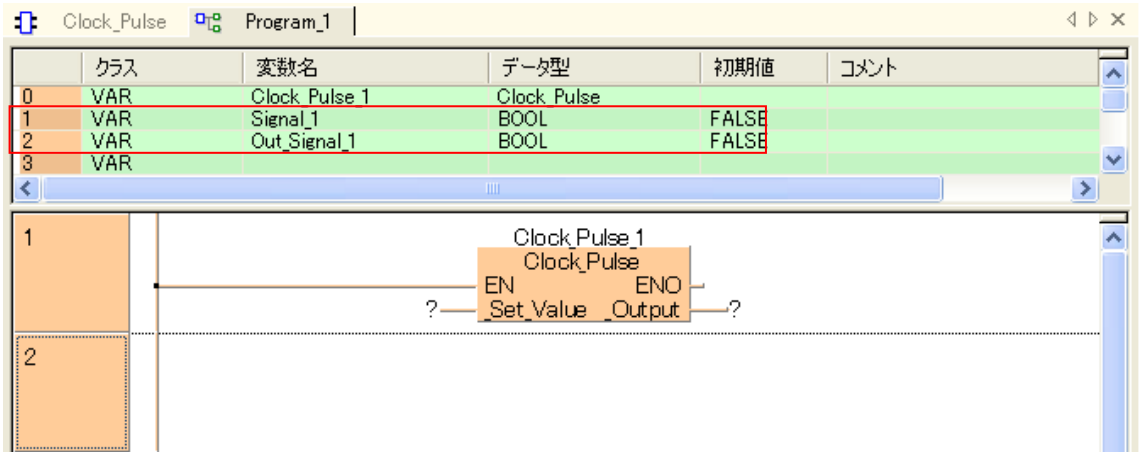
把 Function Block 貼到 Network1 的適當位置。



輸入引用名稱為“Clock_Pulse_1”。

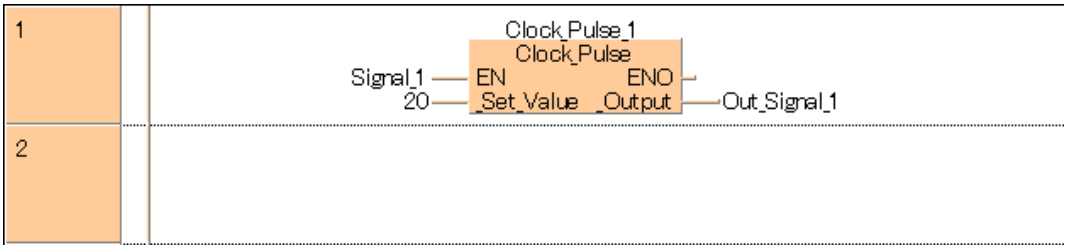
在 Header 裡 Function Block 的引用名稱會自動以變數登録。

6. 如下在程式的 Header 裡、宣告 2 個變數。



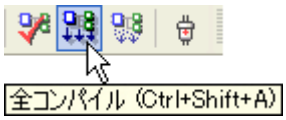
7. 分配登録過的變數到 Function Block 的變數框。

在此將 Clock 時間設定為 20(間隔 2 秒)。



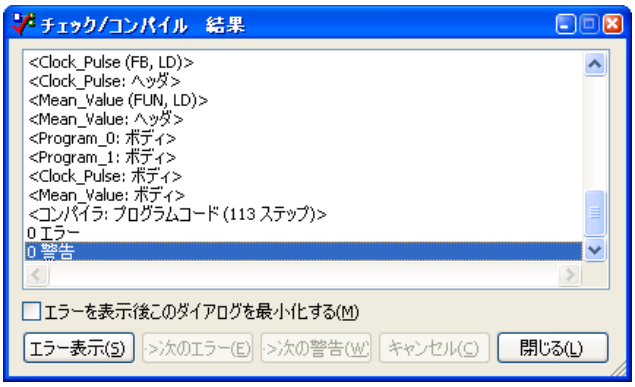
到此、已完成程式中插入 Function Block(POU 類型:FB)。

8. Project 全體進行編譯。



點選工具列上的全編譯小圖示執行。

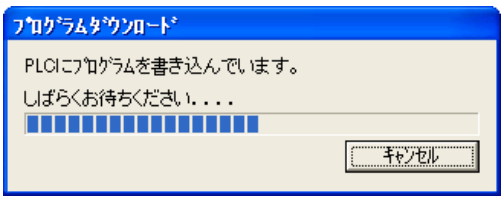
全編譯結果對話框



如果顯示"0 錯誤"、則 Project 的編譯工作結束。

■確認動作

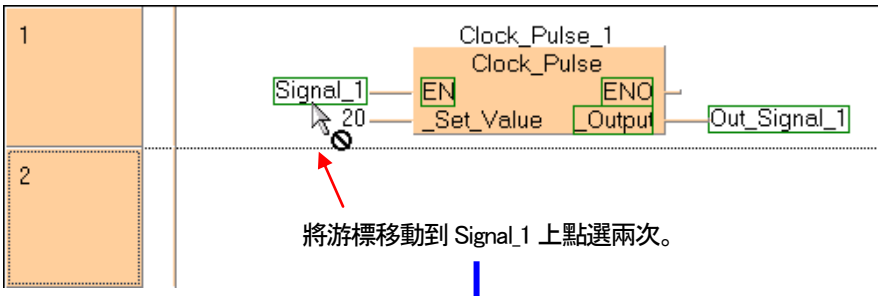
1. **下載 Project。**



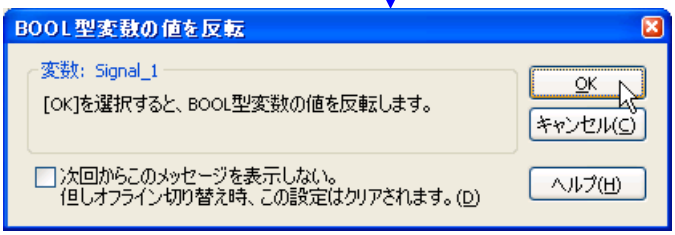
2. **確認 Function Block 的動作。**

請在監控狀態下確認。

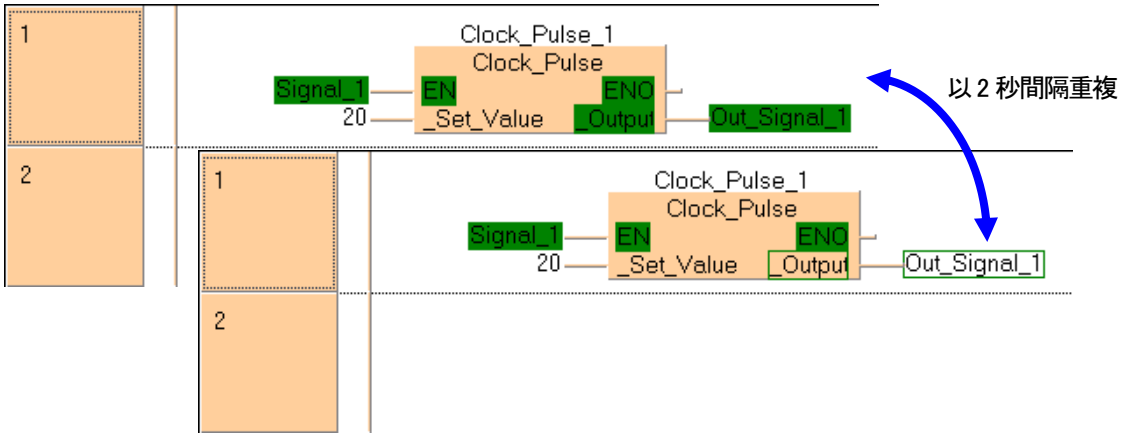
將 Signal_1 條件 ON。



將游標移動到 Signal_1 上點選兩次。



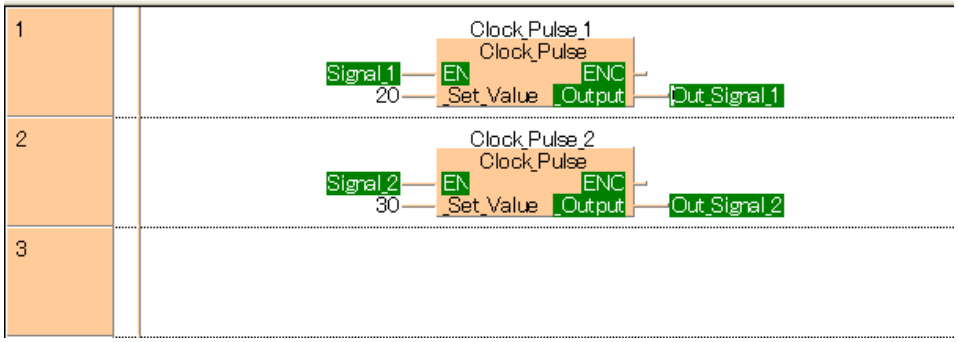
按下 OK 鍵。



如上圖示、可以知道 Out_Signal_1 每間隔 2 秒會切換 ON/OFF。

●備註

	クラス	変数名	データ型	初期値	コメント
0	VAR	Clock_Pulse_1	Clock_Pulse		
1	VAR	Signal_1	BOOL	FALSE	
2	VAR	Out_Signal_1	BOOL	FALSE	
3	VAR	Clock_Pulse_2	Clock_Pulse		
4	VAR	Signal_2	BOOL	FALSE	
5	VAR	Out_Signal_2	BOOL	FALSE	



在上圖、Function Block (FB)在程式中、兩個分別以不同時間的 Clock 計時繼電器的使用範例。
將 Clock 計數器回路 Library 化後、“程式中想重覆使用”時、在 POU 類型裡使用 Function Block (FB)。
像計時器內部需要記憶體時、對於 POU 類型不能使用功能(FUN)。

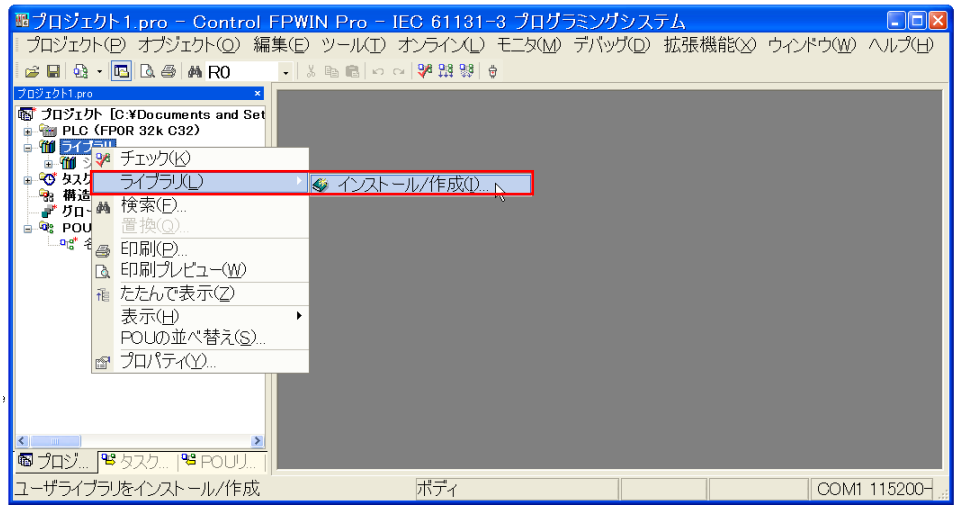
8-4 FunctionLibrary

FunctionLibrary 是為了減少程式開發的工時、以及通過程式標準化讓品質向上為目的、免費提供的 Function。
在制御機器 Web 網頁裡面有提供類比通訊和脈衝輸出、高機能的(FunctionLibrary)。
所下載的 FunctionLibrary 安裝到 FPWIN Pro 的使用者 Library 裡後、使用者的程式可直接組合使用。

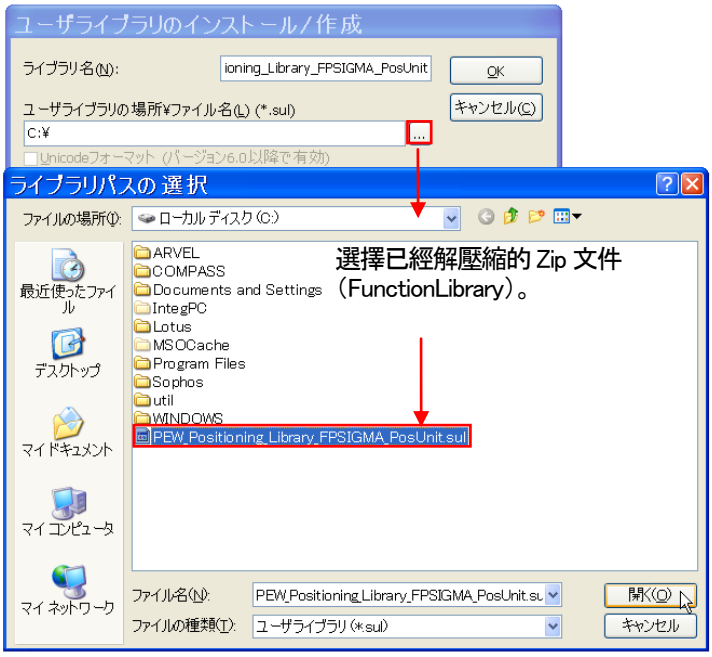
FunctionLibrary 的提供網頁(必須要登録會員才能下載。)
(http://panasonic-denko.co.jp/ac/j/dl/software_info/plc/fpwinpro/fb/index.jsp)

■安裝 FunctionLibrary 到 FPWIN Pro 的 Library

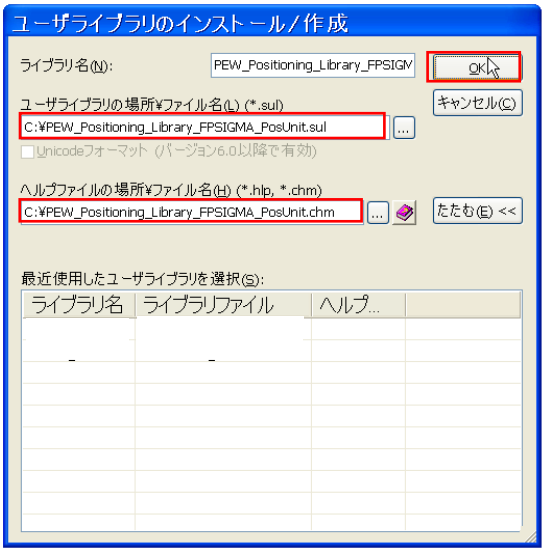
1. 利用敝公司 Web 上提供的 FunctionLibrary、下載 FunctionLibrary。
2. 將下載的 Zip 檔解壓縮。
3. 選擇 Project 導引、Library、安裝/作成。



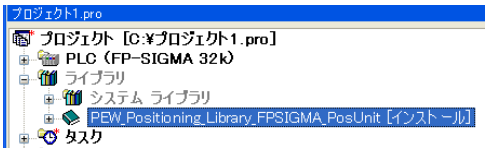
4. 按下使用者 Library/路徑檔案名稱旁的按鍵、選擇解壓縮後的檔案。
(這次用 FP Σ 定位模組的 Library 作為範例來介紹。)



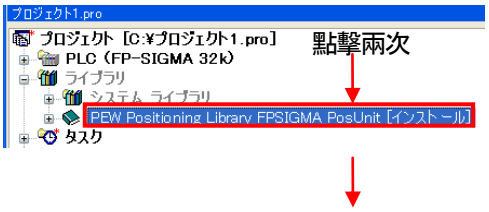
5. 將下載的 Function Library 設定到使用者 Library 路徑/檔案名稱之後按 OK。
（説明文件的路徑/文件名稱路徑會由使用者 Library 所設定的 Function Library 的説明文件自動設定。）



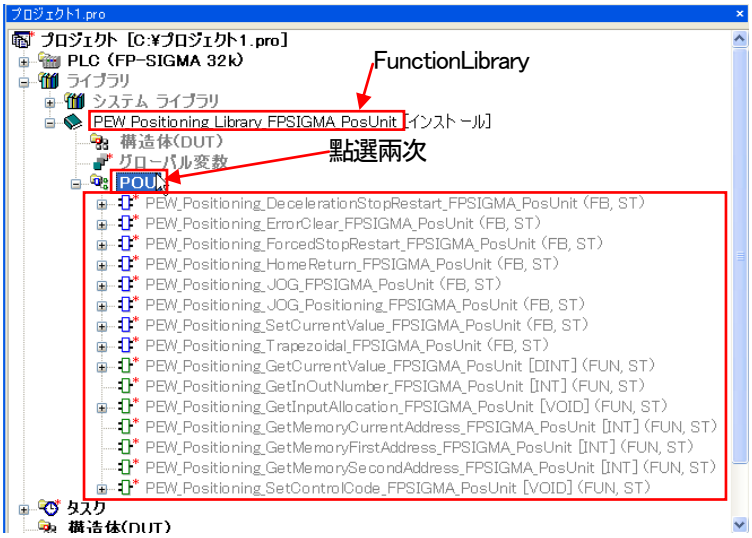
6. 已下載的 FunctionLibrary 會安裝到 Project 導引的 Library 裡。



7. 確認安裝好的 FunctionLibrary 內容。

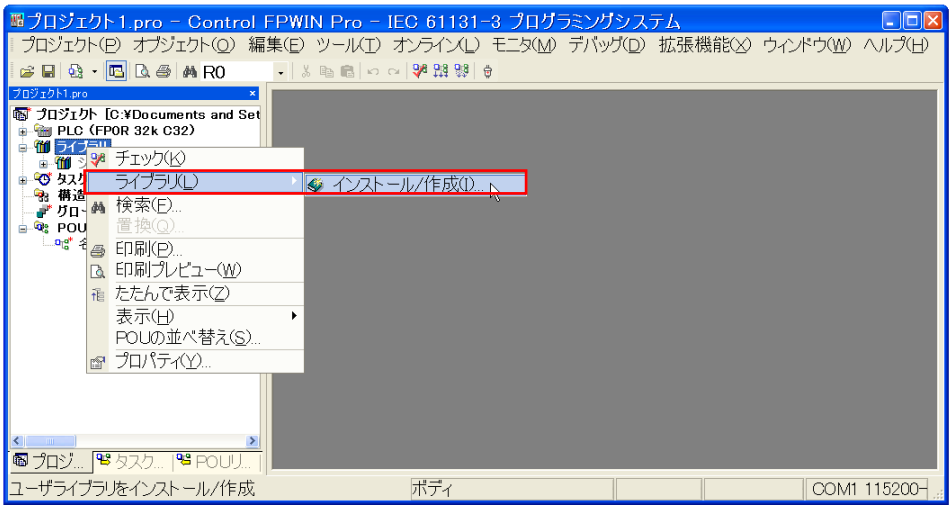


8. POU 正下方有已登録好下載的 Function Library 的 Function (FUN)、Function Block (FB)。

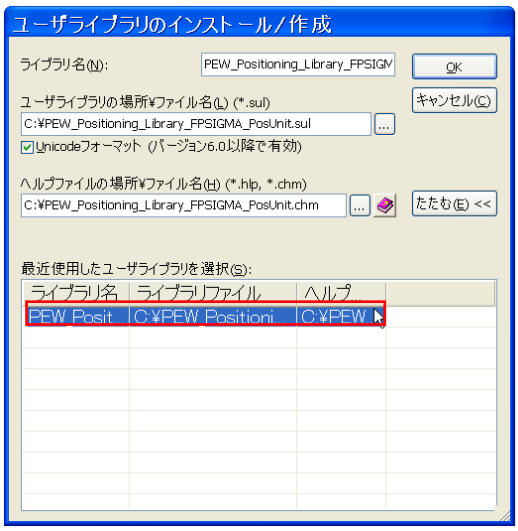


■FunctionLibrary 説明的確認方法

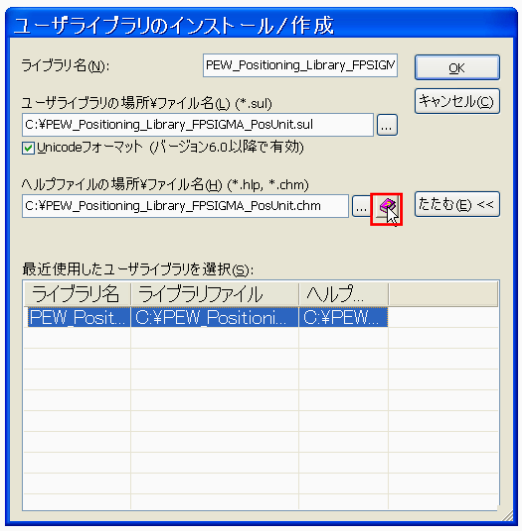
1. 選擇 Project 導引、Library、安装/作成。



2. 選擇所下載的 FunctionLibrary 名稱。
（本次用 FP Σ 位置定位模組的 Library 做範例介紹。）



3.點選以下標記。



4. 可以確認 Function Library（FPΣ定位模組用）的 Function(FUN)、
Function Block (FB)的內容。

PEW_Positioning_Library_FPSIGMA_PosUnit

Control FPWIN Pro Ver6.1 以上所使用的 Library

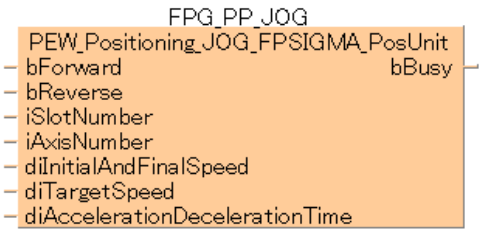
可以使用的機種：FPSIGMA

FPSIGMA 定位控制模組用的 Library		
PEW_Positioning_HomeReturn_FPSIGMA_PosUnit	FB	在 FP Σ 定位模組中，進行原點復歸的 FB
PEW_Positioning_JOG_FPSIGMA_PosUnit	FB	在 FP Σ 定位模組中，進行 JOG 運轉的 FB
PEW_Positioning_JOG_Positioning_FPSIGMA_PosUnit	FB	在 FP Σ 定位模組中，進行 JOB 定位運轉的 FB
PEW_Positioning_SetControlCode_FPSIGMA_PosUnit	FUN	在 FP Σ 定位模組中，設定控制碼的 FUN
PEW_Positioning_Trapezoidal_FPSIGMA_PosUnit	FB	在 FP Σ 定位模組中，進行 E 點控制的 FB
PEW_Positioning_GetCurrentValue_FPSIGMA_PosUnit	FUN	利用 FP Σ 定位模組，讀出經過值的 FUN
PEW_Positioning_SetCurrentValue_FPSIGMA_PosUnit	FB	變更 FP Σ 定位模組的經過值的 FB
PEW_Positioning_ForcedStopRestart_FPSIGMA_PosUnit	FB	強制停止 FP Σ 定位模組的脈衝輸出和解除的 FB
PEW_Positioning_ErrorClear_FPSIGMA_PosUnit	FB	清除 FP Σ 定位模組錯誤的 FB
PEW_Positioning_DecelerationStopRestart_FPSIGMA_PosUnit	FB	減速停止 FP Σ 定位模組的脈衝輸出和解除的 FB
PEW_Positioning_GetInputAllocation_FPSIGMA_PosUnit	FUN	確認 FP Σ 定位模組狀態的 FUN

點選 Library 的項目（這次為 PEW_Positioning_JOG _FPSIGMA_PosUnit）可以確認以下的詳細內容。

可以使用的機種：FPSIGMA

在 FP Σ 定位模組中，進行 JOG 運轉的 FB



變數：

參數	資料型態	內容
bForward	BOOL	JOG 正轉信號 範圍：TRUE or FALSE(ON or OFF)
bReverse	BOOL	JOG 逆轉信號 範圍：TRUE or FALSE(ON or OFF)
iSlotNumber	INT	SLOT No. 範圍：0～3
iAxisNumber	INT	軸 No. 範圍：1 or 2
diInitialAndFinalSpeed	DINT	起動速度[pps] 範圍：0～4,000,000
diTargetSpeed	DINT	目標速度[pps] 範圍：1～4,000,000
diAccelerationDeceleration Time	DINT	加減速時間[ms] 範圍：0～32,767
bBusy	BOOL	BUSY 信號 範圍：TRUE or FALSE(ON or OFF)

動作說明：
輸入「Slot No.」、「軸 No.」、「起動速度」、「目標速度」、「加減速時間」。
JOG 正轉信號(bFoward)或是 JOB 逆轉信號(bReverse)在 TRUE 之間，因應設定好的參數進行 JOG 運轉。
※在脈衝輸出中，緊急停止状態，減速停止状態之下，此 FB 無法起動。
細節請參考 FP Σ 定位控制模組操作手冊。

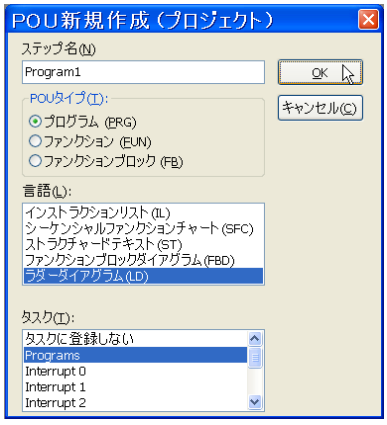
Example：
Slot No.0、軸 No.1 的 JOG 運轉中的範例。

「初速度」=500Hz
「最高速度」=1,000Hz
「加減速速度」=100ms
的條件下，進行 JOG 運轉。
並且關於控制碼如以下設定。細節請參考控制碼的 FUN
請看(PEW_Positioning_SetControlCode_FPSIGMA_PosUnit)的說明

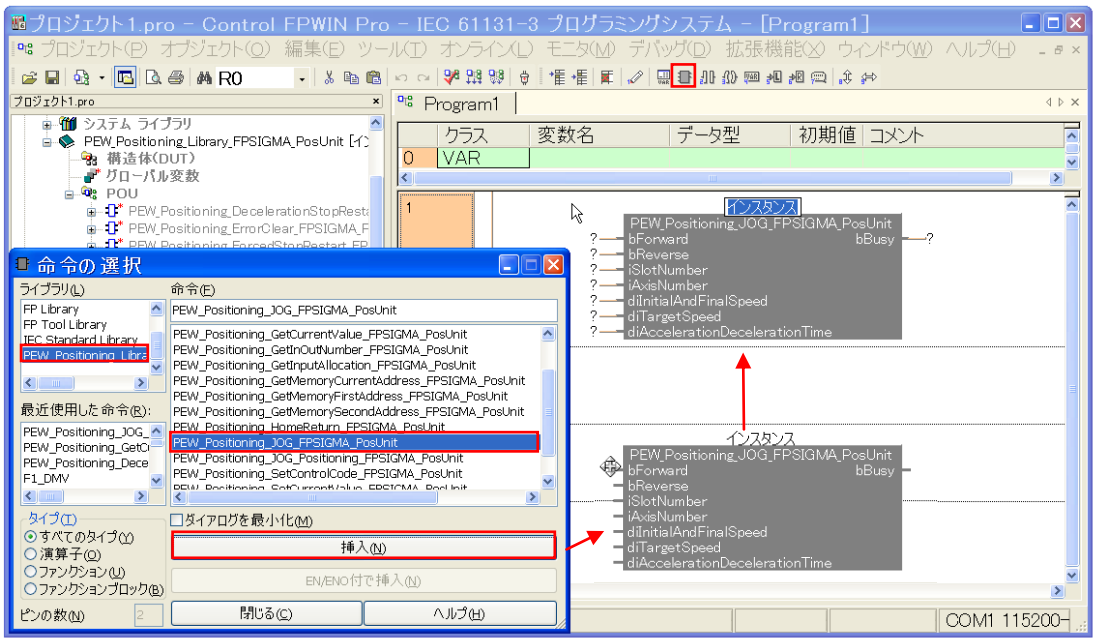
「控制方式」=絕對值
「加減速方式」=直線加減速
「原點復歸方向」=經過值-方向
「起動速度」=0.02ms
「原點輸入理論」=通電時輸入有效
「原點近傍理論」=通電時輸入有效
「原點搜尋」=有效
「極限輸入理論」=通電時輸入有效
「S 字型態」=Sin 曲線
「迴轉方向」=正轉
「輸出模式」=CW/CCW
「偏差計數清除時間」=1ms

■將 FunctionLibrary 讀出到 POU 看看。

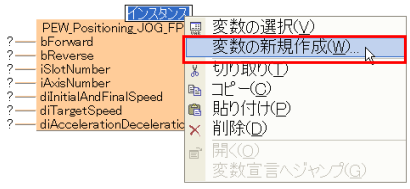
1. 用 POU 類型 PRG 型、LD 語法製做新的 POU。

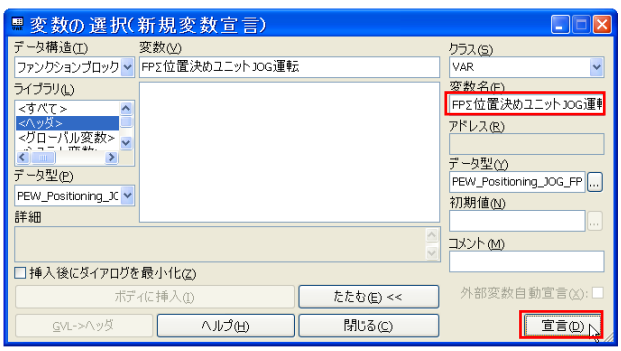


2. 按下 OP/FUN/FB 選擇按鍵。選擇已經下載的 FunctionLibrary 的“PEW_Positioning_JOG _FPSIGMA_PosUnit”（JOG 運轉用 FB）、按下插入按鍵就可以配置到 Network1 的程式區中。

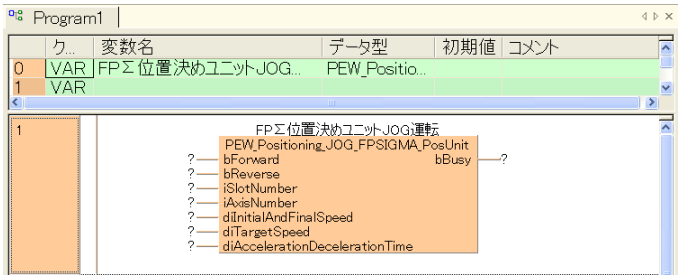


3. 引用名稱上按右鍵選擇製作新的變數。
將「FPΣ定位模組 JOG 運轉」輸入到變數名稱、（變數名稱可以輸入任意文字。）
按下宣告按鍵。

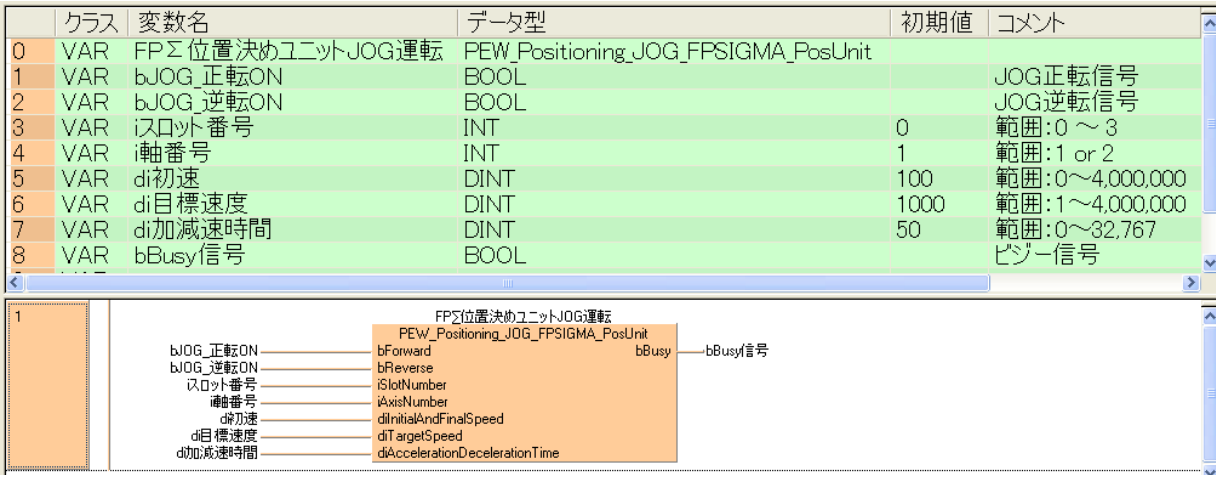




4. PEW_Positioning_JOG _FPSIGMA_PosUnit 作為變數被登録到 Header。



5. 參考下記的 Header 登錄變數並從 bForward 到 bBusy 配置各個 Pin。

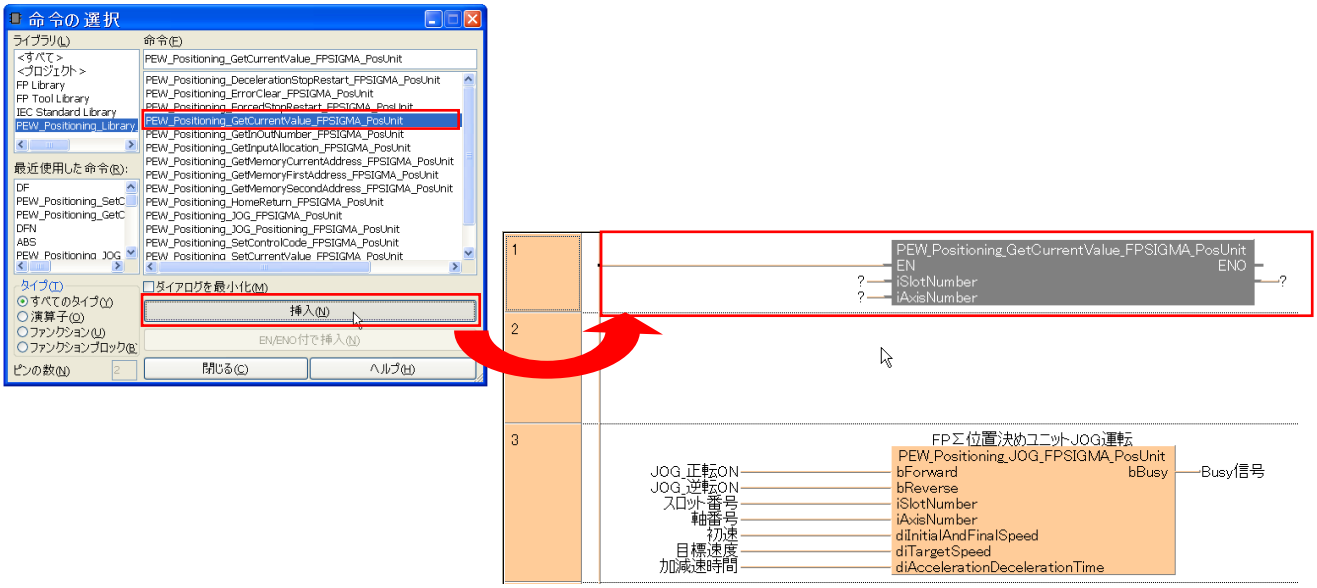


變數：

參數	資料型態	內容
iSlotNumber	INT	SLOT No. 範圍:0～3
iAxisNumber	INT	軸 No. 範圍:1 or 2

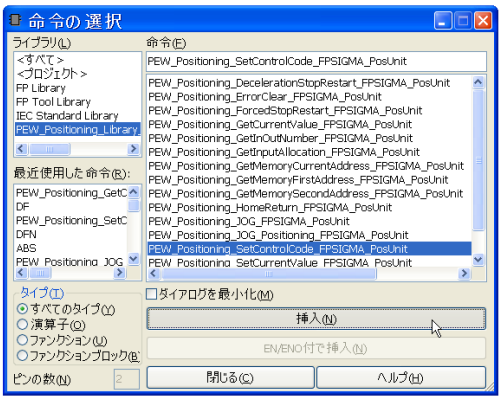
■課題 1

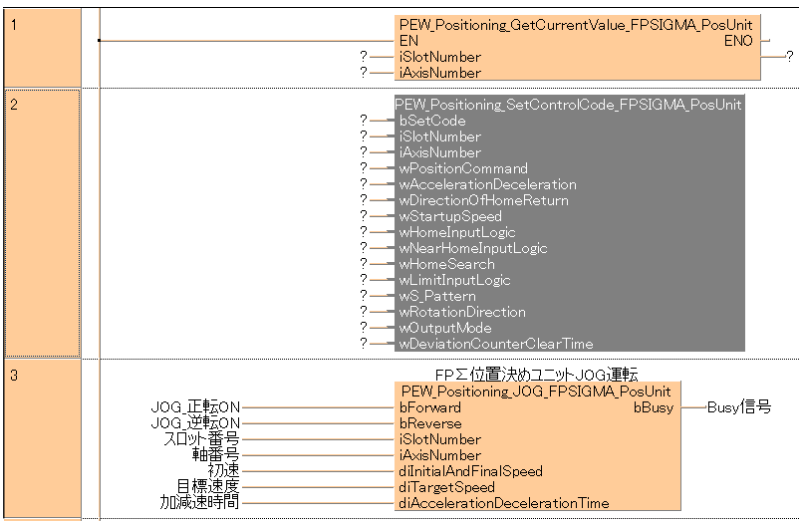
將 PEW_Positioning_GetCurrentValue_FPSIGMA_PosUnit（用 FPSIGMA 定位模組讀出經過值的 FUN）讀出到 POU、並登録變数看看。



■課題 2

將 PEW_Positioning_SetControlCode_FPSIGMA_PosUnit（設定 FPSIGMA 定位模組控制碼的 FUN）讀出到 POU、並登錄變數看看。



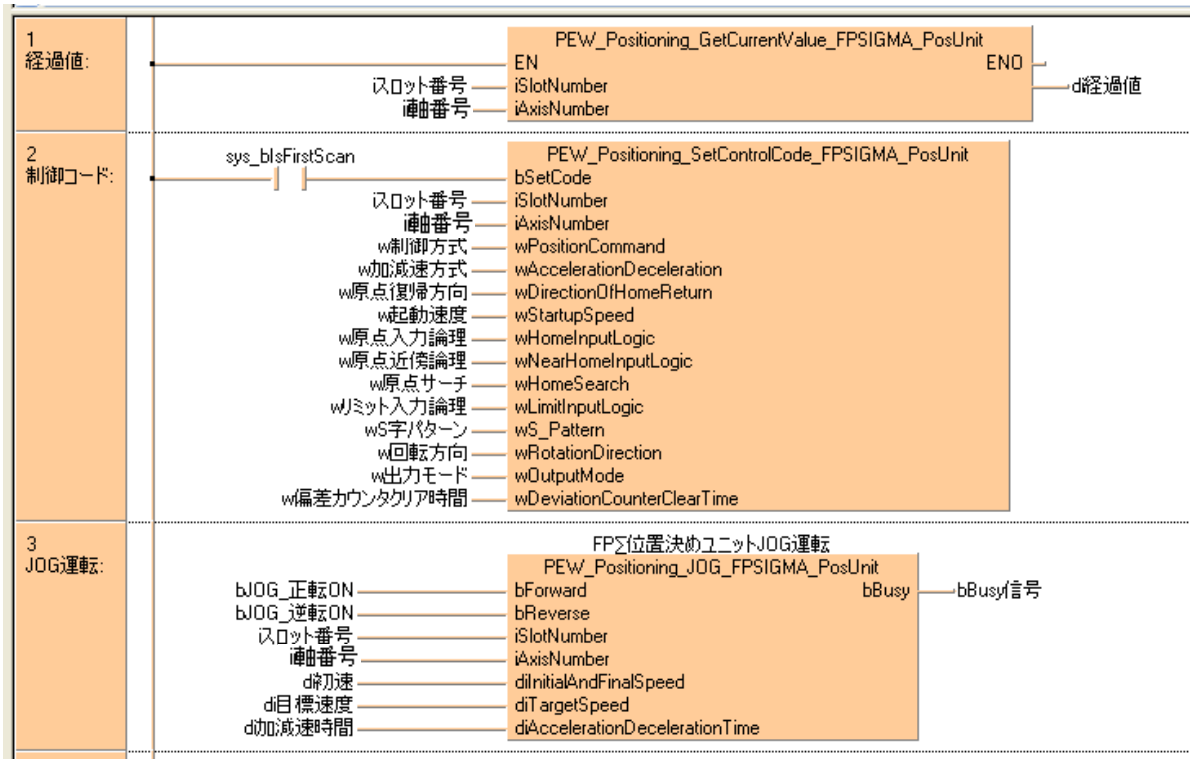


變數：

參數	資料型態	內容
bSetCode	BOOL	控制碼設定信號 範圍:TRUE or FALSE(ON or OFF)
iSlotNumber	INT	SLOT No. 範圍:0～3
iAxisNumber	INT	軸 No. 範圍:1 or 2
WPositionCommand	WORD	控制方式 範圍: 16#0000(增量),16#0001(絕對值)
wAcceleration Deceieration	WORD	加減速方式 範圍: 16#0000(直線加減速),16#0001(S 字加減速)
wDirectionOfHomeReturn	WORD	原點復歸方向 範圍: 16#0000(經過值-方向),16#0004(經過值+方向)
wStartupSpeed	WORD	起動速度 範圍: 16#0000(0.02ms),16#0008(0.005ms)
wHomeInputLogic	WORD	原點輸入理論 範圍: 16#0000(非通電時輸入有效),16#0010(通電時輸入有效)
wNearHomeInputLogic	WORD	原點近傍理論 範圍: 16#0000(通電時輸入有效),16#0020(非通電時輸入有效)
wHoneSearch	WORD	原點搜尋 範圍: 16#0000(無效),16#0040(有效)
wLimitInputLogic	WORD	極限輸入理論 範圍: 16#0000(非通電時輸入有效),16#0080(通電時輸入有效)
wS_Pattern	WORD	S 字型態 16#0000(Sin 曲線) 16#1000(2 次曲線) 16#2000(圓形曲線) 16#3000(3 次曲線)
wRotationDirection	WORD	迴轉方向 範圍: 16#0000(正轉),16#0100(反轉)
wOutputMode	WORD	輸出模式 範圍: 16#0000(Pulse/Sign),16#0200(CW/CCW)
wDeviationCounterClearTome	WORD	偏差計數清除時間 範圍: 16#0000(1ms),16#8000(10ms)

■課題 1、2 解答的設定例

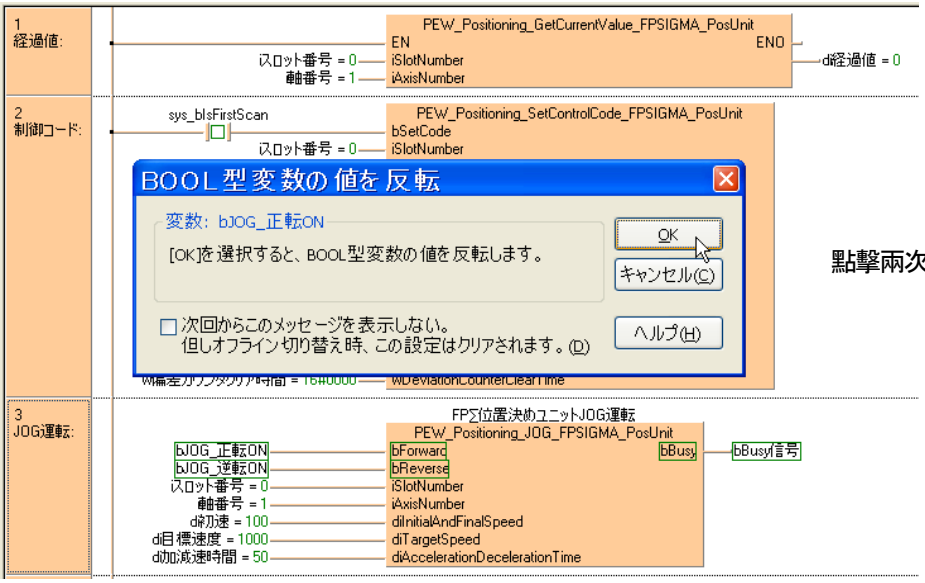
	クラス	変数名	データ型	初期値	コメント
0	VAR	iスロット番号	INT	0	範囲:0 ～ 3
1	VAR	i軸番号	INT	1	範囲:1 or 2
2	VAR	di経過値	DINT	0	-2,147,483,648 +2,147,483,647
3	VAR	w制御方式	WORD	16#0000	0:インクリメント 1:アブソリュート
4	VAR	w加減速方式	WORD	16#0000	0:直線加減速 1:S字加減速
5	VAR	w原点復帰方向	WORD	16#0000	0:マイナス方向 1:プラス方向
6	VAR	w起動速度	WORD	16#0000	0:0.02ms 8:0.005ms
7	VAR	w原点入力論理	WORD	16#0000	0:非通電時有効 10:通電時有
8	VAR	w原点近傍論理	WORD	16#0000	0:通電時有効 20:非通電時有
9	VAR	w原点サーチ	WORD	16#0040	0:無効 40:有効
10	VAR	wリミット入力論理	WORD	16#0000	0:非通電時有効) 80:通電時有
11	VAR	wS字パターン	WORD	16#0000	0:Sin曲線1000:2次曲線2000:サ
12	VAR	w回転方向	WORD	16#0000	0:正転 100:逆転
13	VAR	w出力モード	WORD	16#0200	0:Pulse/Sign 200:CW/CCW
14	VAR	w偏差カウンタクリア時間	WORD	16#0000	0:1ms 8000:10ms
15	VAR	FPΣ位置決めユニットJOG運転	PEW_Positioning_JOG_FPSIGMA_PosUnit		
16	VAR	bJOG_正転ON	BOOL		JOG正転信号
17	VAR	bJOG_逆転ON	BOOL		JOG逆転信号
18	VAR	di初速	DINT	100	範囲:0～4,000,000
19	VAR	di目標速度	DINT	1000	範囲:1～4,000,000
20	VAR	di加減速時間	DINT	50	範囲:0～32,767
21	VAR	bBusy信号	BOOL		ビジー信号
22	VAR	エラークリア	BOOL	FALSE	



■課題 3

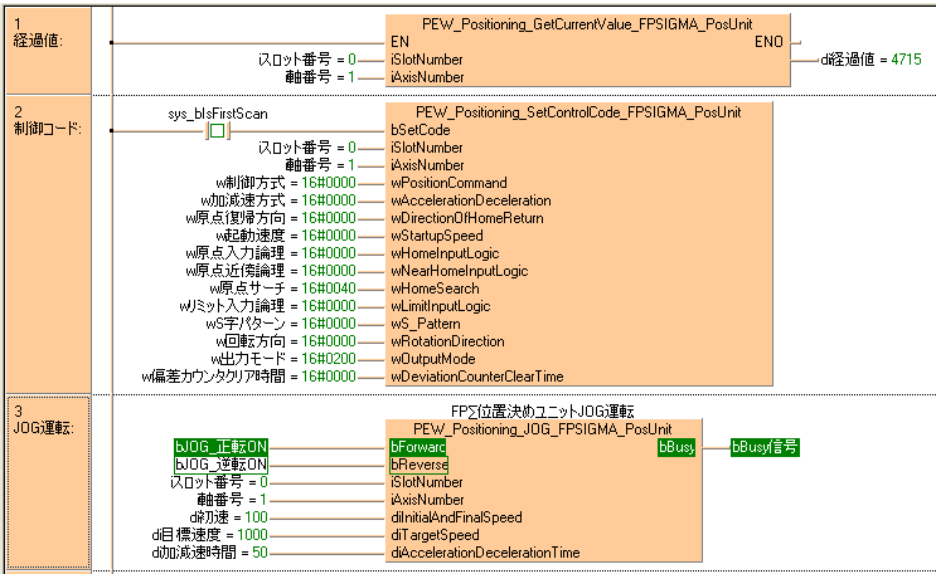
將 FPΣ定位模組裝到 FPΣ控制器模組、並確認課題 1、2 的程式動作。

將 JOG 正轉 ON、進行 JOG 正轉動作(往正方向計數)、將 JOG 逆轉 ON 進行 JOG 逆轉動作(往負方向計算)。



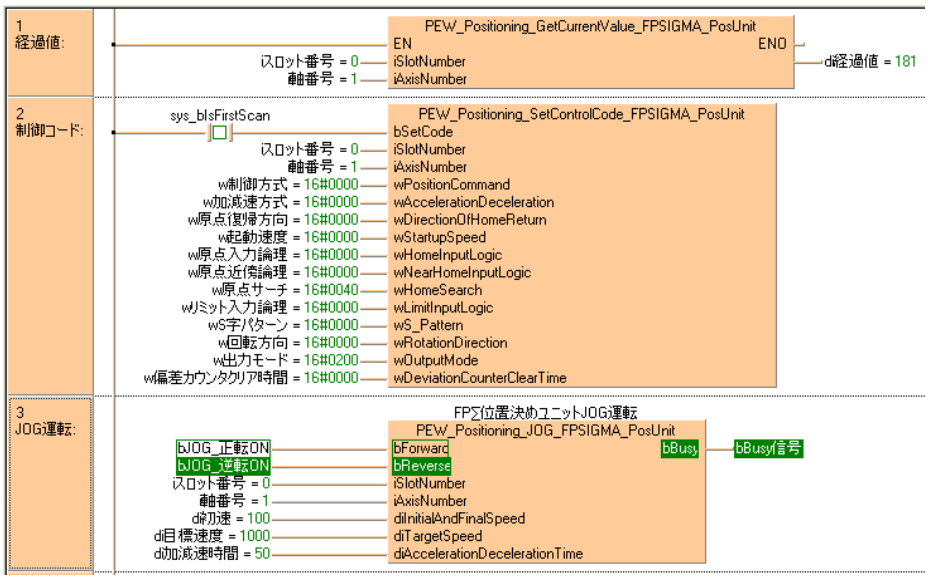
點擊兩次選擇 OK。

JOG 正轉 ON



往正方向計數

JOG 逆轉 ON



往負方向計數