PLC Programmer for Linux Controller

編 輯 器 說 明 書

2012/07 Ver：V00.00.006

Leading Numerical Controller

LNC Technology Co., Ltd.

目錄

1 PLC PROGRAMMER FOR LINUX CONTROLLER 軟體簡介 6

2 PLC PROGRAMMER FOR LINUX CONTROLLER 功能架構 7

3 PLC PROGRAMMER FOR LINUX CONTROLLER 軟體操作 10

3.1 執行軟體 10
3.2 軟體畫面介紹 11
3.3 專案(Progect).....11
3.3.1 開新檔案(New) 11
3.3.2 開舊檔案(Open) 13
3.3.3 存檔(Save + Compile) 14
3.3.4 另存新檔(Save As + Compile).....16
3.3.5 關檔(Close) 17
3.3.6 模擬器(Simulator).....18
3.3.7 列印(Print) 19
3.3.8 結束(Exit).....22
3.4 編輯(Edit).....22
3.4.1 插入一行 (Ins One Line).....22
3.4.2 插入多行(Ins Multi Lines) 23
3.4.3 刪除(Delete) 25
3.4.4 複製(Copy)、貼上(Paste)、剪下(Cut).....26
3.4.5 檢查(Check) 26
3.5 搜尋(Search).....27
3.5.1 找尋(Find) 27
3.5.2 尋找下一個(Find Next) 29
3.5.3 跳躍到某一行(Go To Line Number).....29
3.5.4 取代(Replace) 29
3.6 設定(Setting).....30
3.6.1 提示編輯(Symbol) 30
3.6.2 專案註解(Annotation) 31
3.7 傳送(Transfer).....31
3.8 幫助(Help).....31
3.8.1 關於(About) 31

4 工具列功能說明 32

5 元件工具列說明 33

5.1 游標快捷鍵： 33

5.2 Contact(接點)快捷鍵 33

5.3 Coil 快捷鍵： 34

5.4 Timer 快捷鍵： 36

5.5 Counter 快捷鍵： 38

5.6 Arithmetic 快捷鍵： 39

5.7 Tool 快捷鍵： 44

5.8 Sequence 快捷鍵： 45

5.9 Horizontal Connect 快捷鍵： 46

5.10 Vertcal Connect 快捷鍵： 46

5.11 R-Contact 快捷鍵： 46

5.12 R-Coil 快捷鍵： 48

6 PLC 接點動作說明 50

6.1 常開接點 -| |- 50

6.2 常閉接點 -|/|- 50

6.3 常開線圈 ( ) (Normal Open Coil) 50

6.4 常閉線圈 (/) (Normal Closed Coil) 51

6.5 正緣觸發脈衝線圈 (↑) (Positive Edge Triggered Coil) 51

6.6 負緣觸發脈衝線圈 (↓) (Negative Edge Triggered Coil) 51

6.7 自保持線圈 (S) (Latch Coil) 52

6.8 復歸線圈 (R) (Unlatch Coil) 52

6.9 限時繼電器 (Timer Relay) 53

6.10 向上計數器 (Up_Counter) 53
6.11 向下計數器 (Down_Counter) 54
6.12 向上循環計數器 (Ring Up_Counter) 54
6.13 向下循環計數器 (Ring Down_Counter) 55
6.14 計數器復歸 (Reset Coounter) 56
6.15 加算指令 (Add) 56
6.16 減算指令 (Subtract) 56
6.17 乘法指令 (Multiply) 57
6.18 除法指令 (Divide) 57
6.19 邏輯 AND 指令 (And) 58
6.20 邏輯 OR 指令 (Or) 59
6.21 邏輯 XOR 指令 (Exclusive Or) 59
6.22 比較 CMP 指令(大於) (Compare，Larger Than) 60
6.23 比較 CMP 指令(小於) (Compare，Smaller Than) 61
6.24 比較 CMP 指令(等於) (Compare，Equal To) 61
6.25 資料搬移 MOVE 指令 (Move) 62
6.26 資料搬移 MOVE 指令 (由 Register Move to Register pointer) 62
6.27 資料搬移 MOVE 指令 (由 Register pointer Move to Register) 63
6.28 連續 R 資料暫存器設值指令 (MULRINI) 63
6.29 連續 R 資料暫存器拷背指令 (MULRCPY) 64
6.30 連續 I 接點映射指令(I/RMAP) 64
6.31 連續 O 接點映射指令(O/RMAP) 65
6.32 程式跳躍 JMP 指令 (Jump) 65
6.33 階梯圖副程式呼叫 JSR 指令 (Jump Subroutine) 66
6.34 階梯圖副程式呼叫結束 Return 指令 (Return Subroutine) 66
6.35 符號標記宣告 (Label) 66
6.36 階梯圖編輯結束 END 指令 (End) 66

7 PLC PROGRAMMER 搜尋操作說明 68

8 滑鼠按鍵功能說明 69

9 上傳、下載、重新載入使用說明 70

10 PLC 警報/警告字串編輯 76

11 專案註解編輯 80

12 PLC 編輯器版本資訊 81

13 模擬器使用說明 82

13.1 開啟程式 82
13.2 開檔(Open) 82
13.3 功能介紹 83
13.4 版本資訊 85

14 編程建議 86

14.1 特定元件重覆使用 86
14.2 往上區塊 86
14.3 操作資料暫存器指標指令使用 86

1 PLC Programmer for Linux Controller 軟體簡介

全系列產品提供內建 PLC 模組。並提供 Windows 作業系統上的階梯圖編輯軟體供客制化修改。PLC 模組支援輸出入點，除了可作 I/O 訊號的 AND、OR 等邏輯控制外，並提供了計時、計數、邏輯、算數和比較等運算能力，可用於各種時序控制的需求。

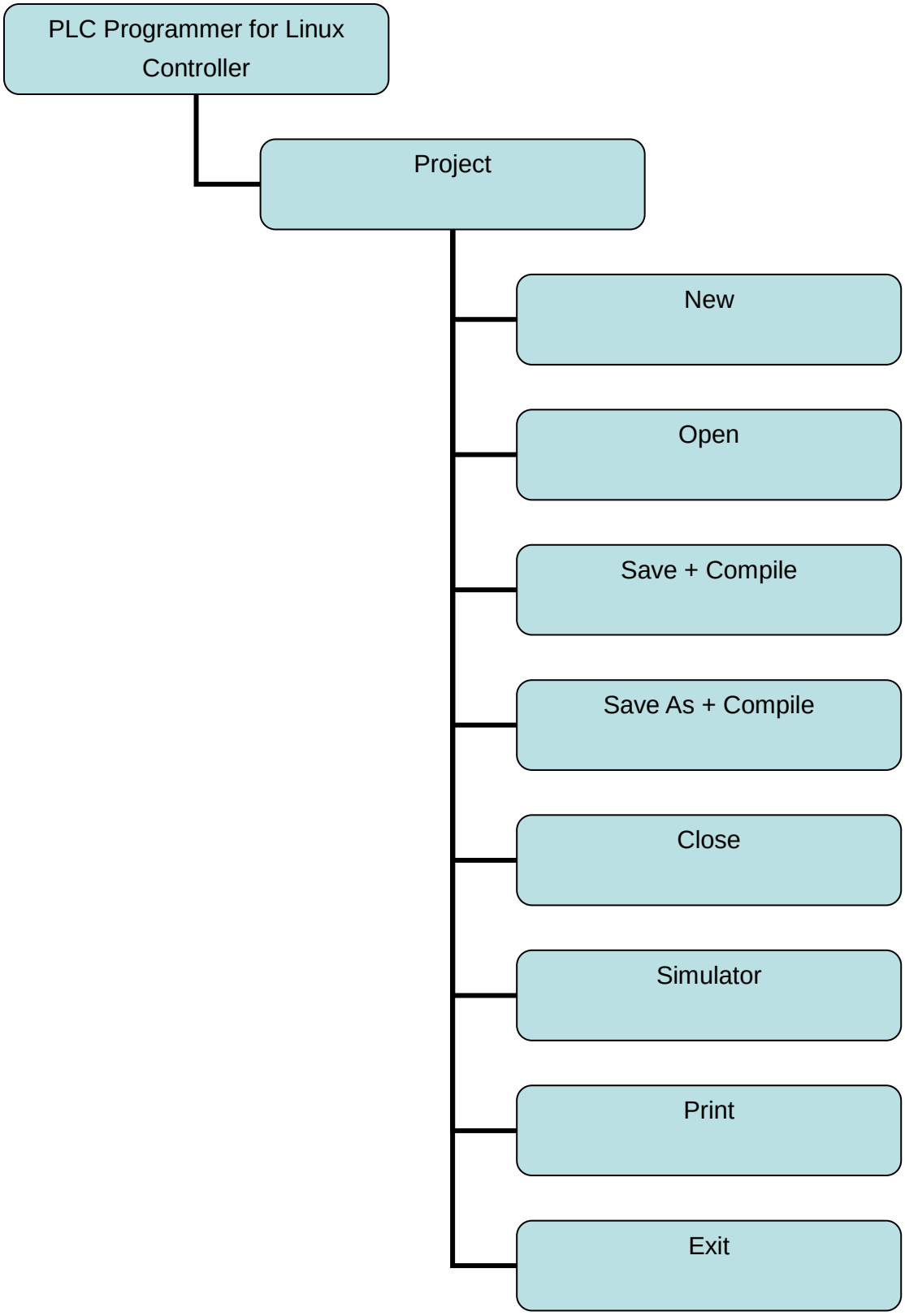
在本文中將會介紹如何使用階梯圖編輯軟體。

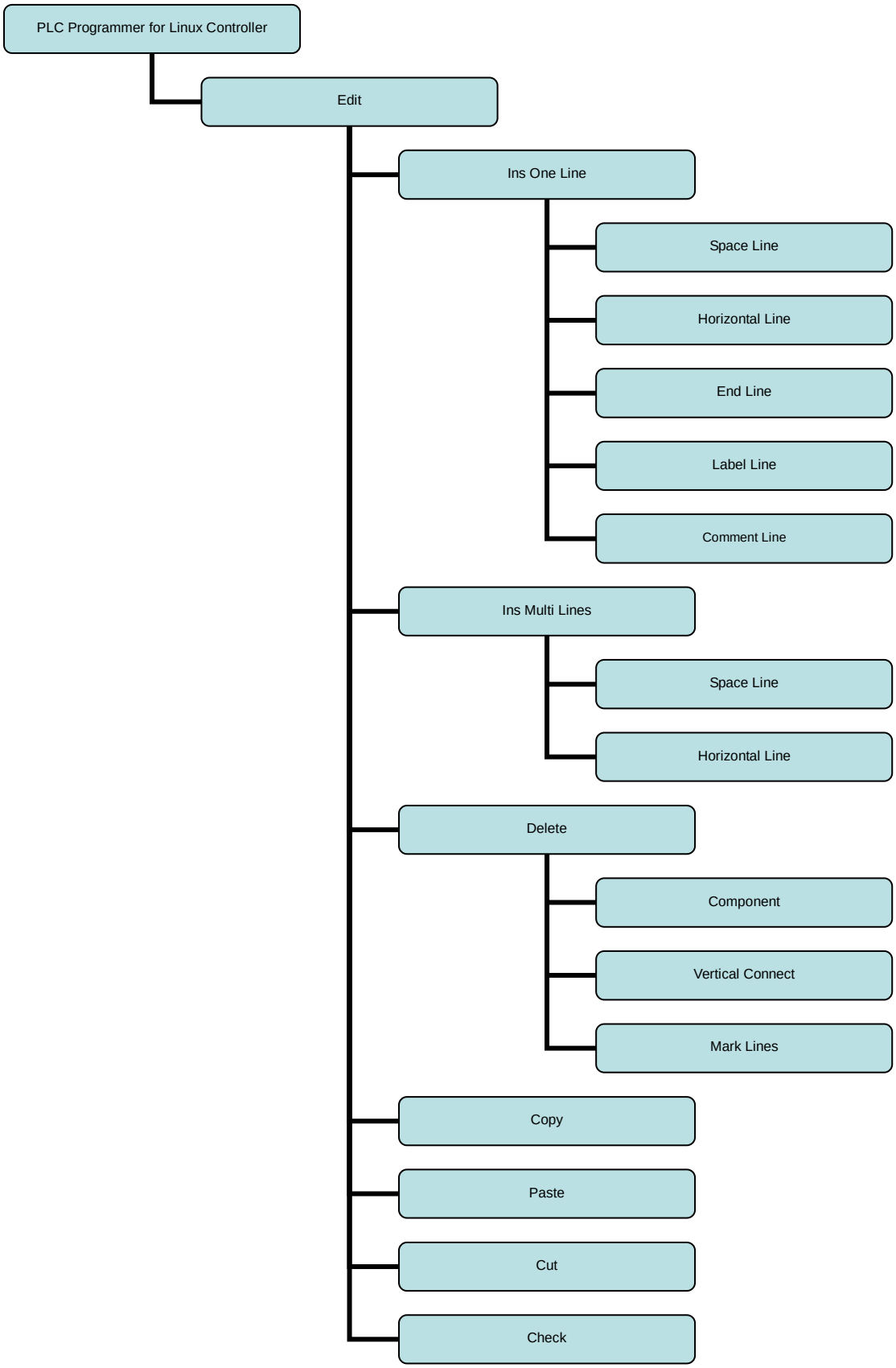
下表為 PLC 模組內之資料型態，用於編寫階梯圖之參考。

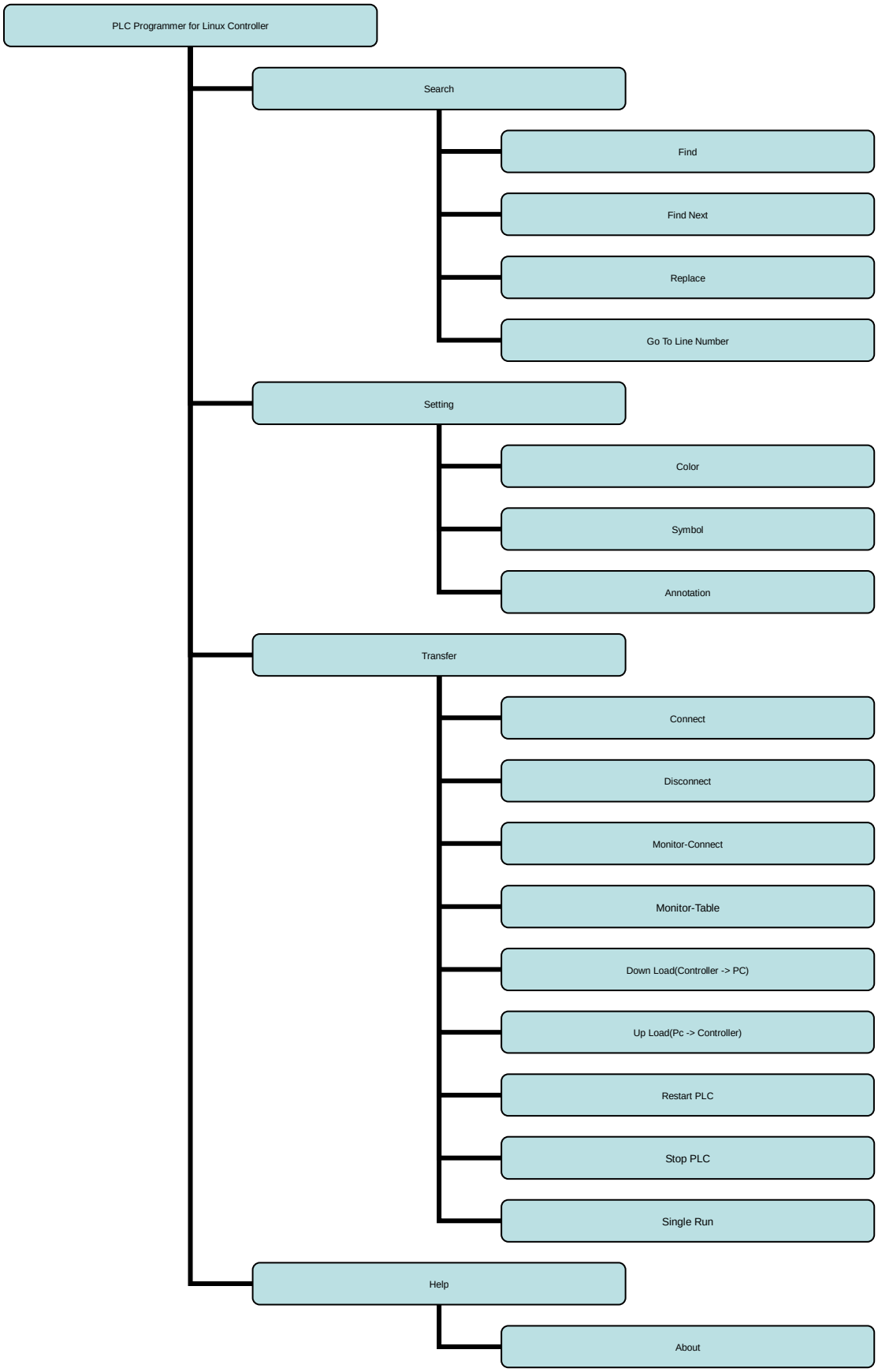
資料型態	名稱	數量	描述
布林(bool)	I	4096	數位輸入----------I0 ~ I4095
	O	4096	數位輸出----------O0 ~ O4095
	A	4096	內部接點 ----------A0 ~ A4095
	S	4096	內部接點 (狀態)-------S0 ~ S4095
	C	4096	內部接點 (命令)-------C0 ~C4095
數值(long)	TIMER (TIM)	256	計時器，保持計時器-------TM0 ~ TM255 計時器種類----------1毫秒，10毫秒，0.1秒，1秒
	計數器 (CNT)	256	計數器---------CT0 ~ CT255計數器種類 上數，下數，重置，循環上數，循環下數
	暫存器 (Register)	30000	暫存器---------- R0 ~ R29999 其中R8000~R9999會保持,其餘斷電即清除為0

表格 一 PLC 模組資料型態定義

2 PLC Programmer for Linux Controller 功能架構




PLC Programmer for Linux Controller
Edit
Ins One Line
Space Line
Horizontal Line
End Line
Label Line
Comment Line
Ins Multi Lines
Space Line
Horizontal Line
Delete
Component
Vertical Connect
Mark Lines
Copy
Paste
Cut
Check


PLC Programmer for Linux Controller
Search
Find
Find Next
Replace
Go To Line Number
Setting
Color
Symbol
Annotation
Transfer
Connect
Disconnect
Monitor-Connect
Monitor-Table
Down Load(Controller -> PC)
Up Load(Pc -> Controller)
Restart PLC
Stop PLC
Single Run
Help
About

3 PLC Programmer for Linux Controller 軟體操作

3.1 執行軟體

有兩種方法可以開啟 PLC Programmer for Linux Controller

方法一、

1. 至安裝PLC資料夾內，例如C:\Program Files\LNC\ (預設資料夾)

2.執行PLCProgrammer.exe檔，開啟PLC Programmer for Linux Controller畫面如圖 一所示。

方法二、

1.點選桌面工具列的開始→所有程式→ LNC

2.點選PLC Programmer捷徑

3.開啟PLC Programmer for Linux Controller畫面如圖 一所示。

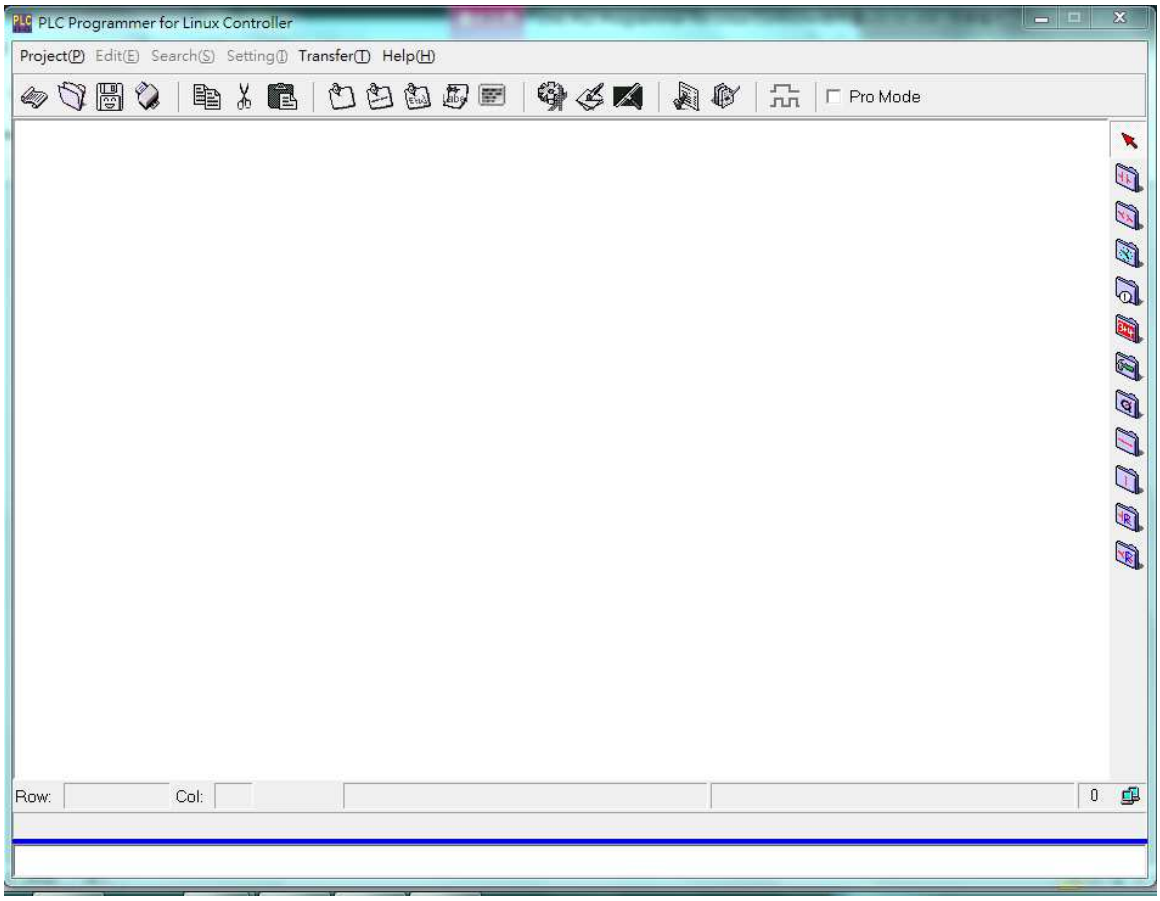


圖 一 PLC Programmer 軟體畫面

3.2 軟體畫面介紹

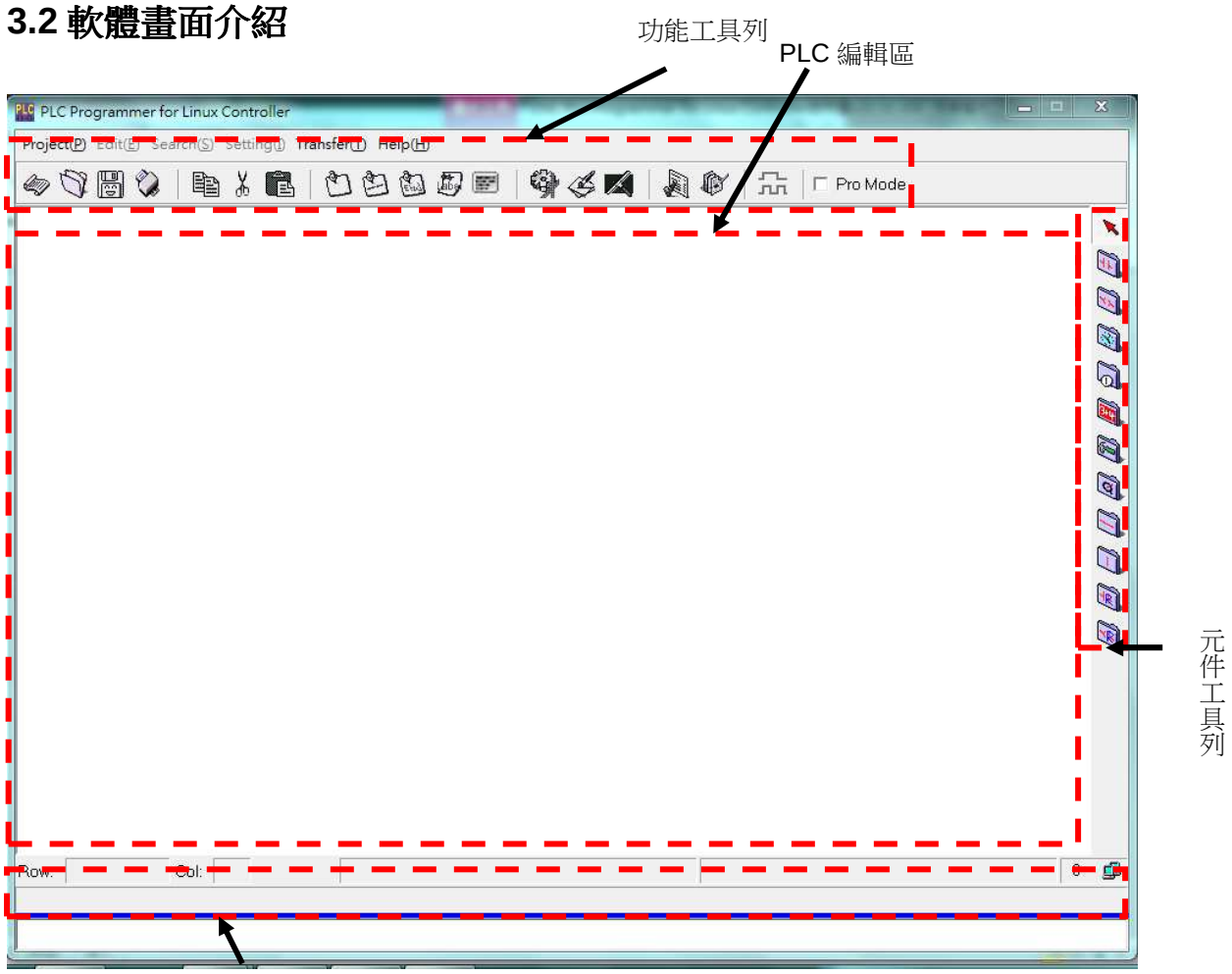


圖 二軟體畫面介紹

訊息區

- 功能工具列：依序為【開新檔、開舊檔、存檔、列印、複製、剪下、貼上、新增空白行、新增線段行、新增結束行、新增標記行、刪除元件、刪除垂直線、刪除選取線段、檢查、註解編輯】，後面章節會有詳細介紹個別功能。
- PLC 編輯區：顯示與編輯階梯圖區域。
- 元件工具列：點選元件後，當滑鼠移至 PLC 編輯區按下滑鼠左鍵後設入該元件於顯示區塊上，各元件功能後面章節會有詳細介紹。
- 訊息區：檢查或編譯有錯誤會顯示在該區塊內，後面章節會有詳細介紹。

3.3 專案(Progect)

3.3.1 開新檔案(New)

建立新專案流程如圖 三：

1. 點選**Project**。
2. 點選New。

3. Language Dialog視窗可選擇註解的語系。

4. 選定請點選OK開啟新專案。

5. 完成後如圖 四。

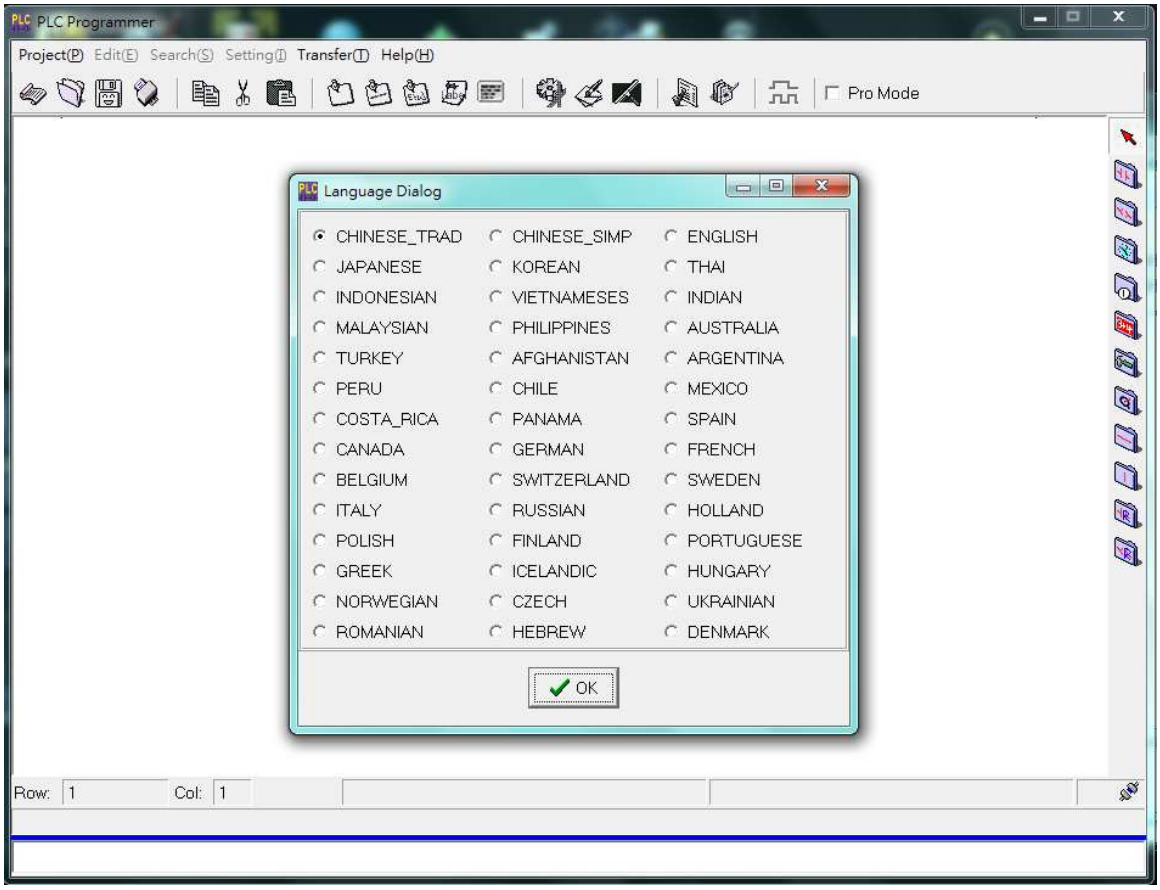


圖 三建立新專案流程

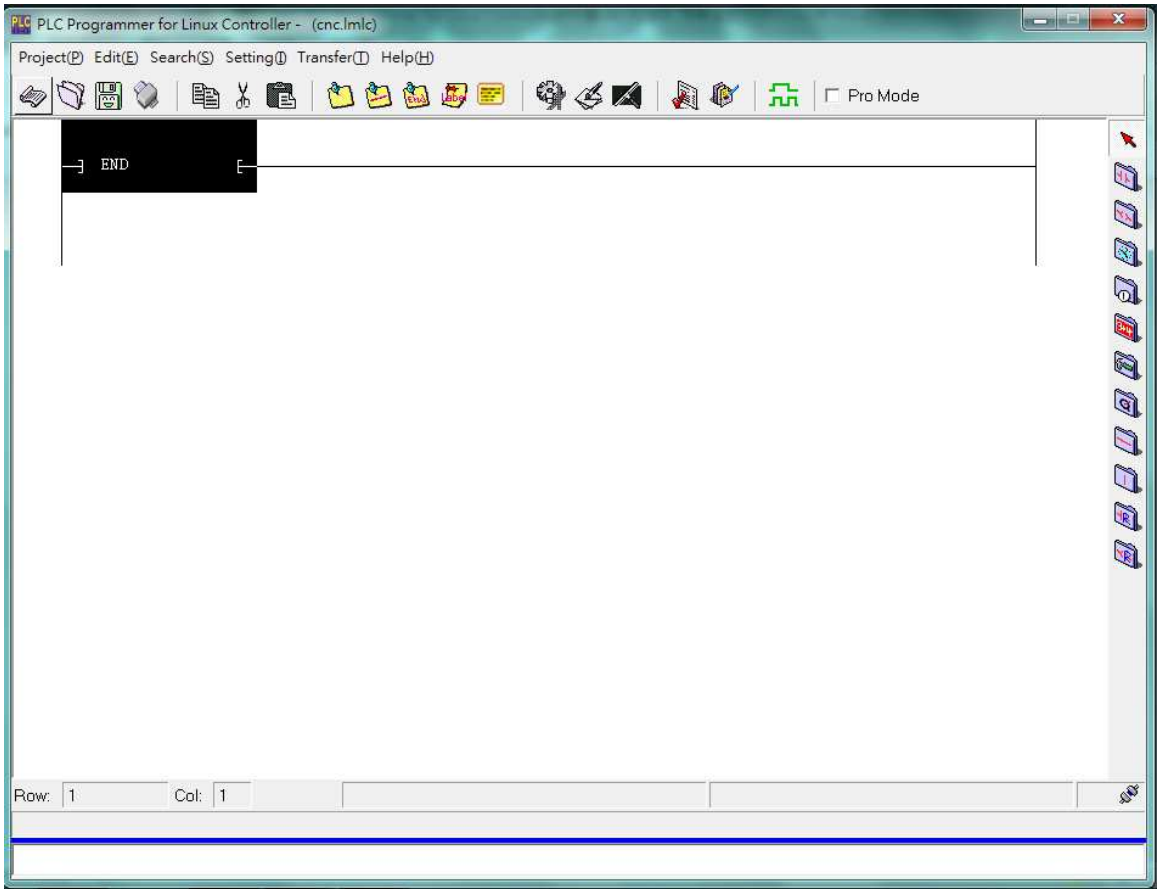


圖 四新增專案完成

3.3.2 開舊檔案(Open)

1. 點選**Project**。
2. 點選Open。
3. Language Dialog視窗可選擇已存在註解的語系。
4. 選定請點選OK開啟專案。
5. 完成後如圖 五。

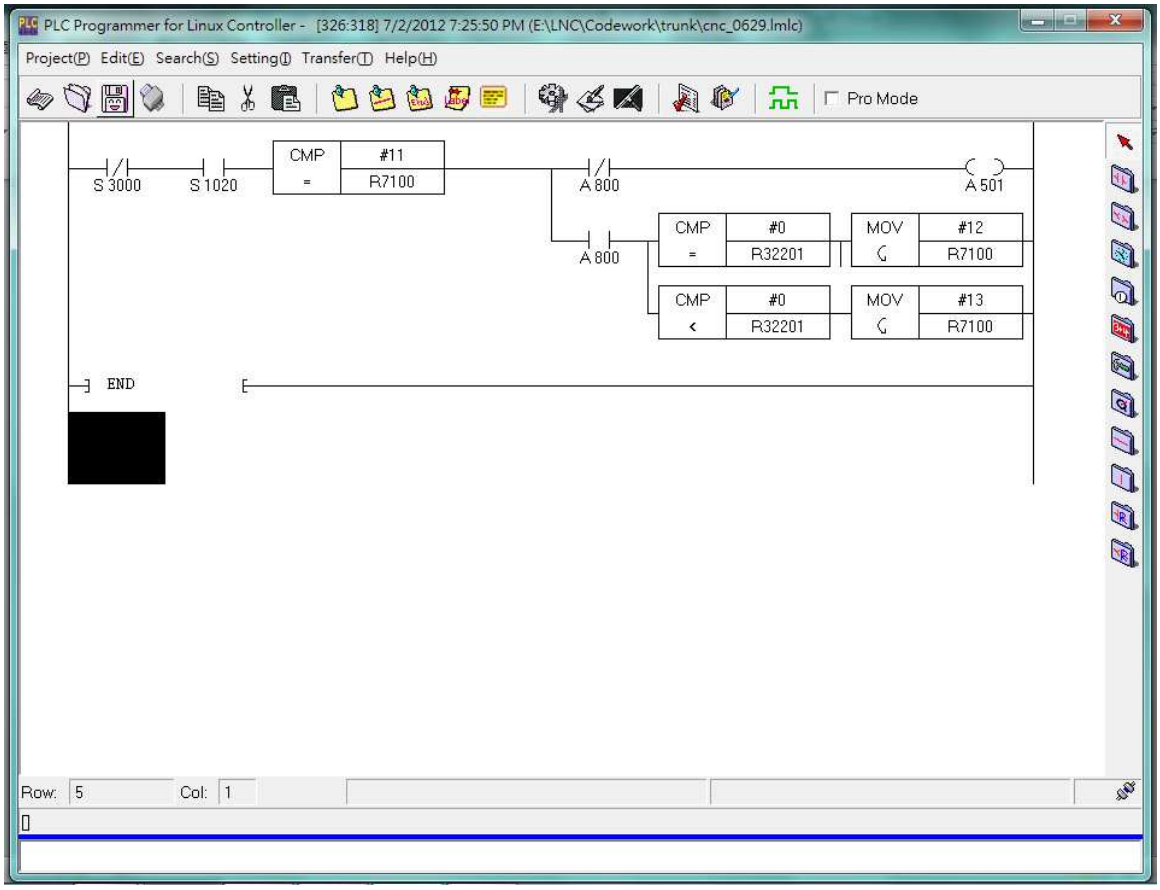


圖 五開啟舊專案完成

3.3.3 存檔(Save + Compile)

存檔同時會進行編譯動作，並產生以下控制器所需檔案。

- cnc.lmlc：階梯圖檔案
- cnc.lcod：階梯圖程式
- cnc.lpar：階梯圖初始值
- cnc_plc_xxxx.str：階梯圖字串檔，其中 xxxx 表示四位數字，意思如下

0000=繁體中文	0011=澳大利亞	0022=德文	0033=希臘文
0001=簡體中文	0012=土耳其	0023=法文	0034=冰島文
0002=英文	0013=阿富汗	0024=比利時	0035=匈牙利
0003=日文	0014=阿根廷	0025=瑞士	0036=挪威文
0004=韓文	0015=秘魯	0026=瑞典	0037=捷克文
0005=泰文	0016=智利	0027=義大利	0038=烏克蘭文
0006=印尼	0017=墨西哥	0028=俄文	0039=羅馬尼亞文
0007=越南	0018=哥斯大黎加	0029=荷蘭文	0040=希伯來文
0008=印度	0019=巴拿馬	0030=波蘭文	0041=丹麥
0009=馬來西亞	0020=西班牙	0031=芬蘭	
0010=菲律賓	0021=加拿大	0032=葡萄牙文	

表格 二 語系對應表

- 編譯錯誤訊息表示
  - -1:記憶體不足。
  - -2:編譯錯誤，如果是邏輯錯誤會顯示出行號來表示，如圖 六。
  - 成功，如圖 七。

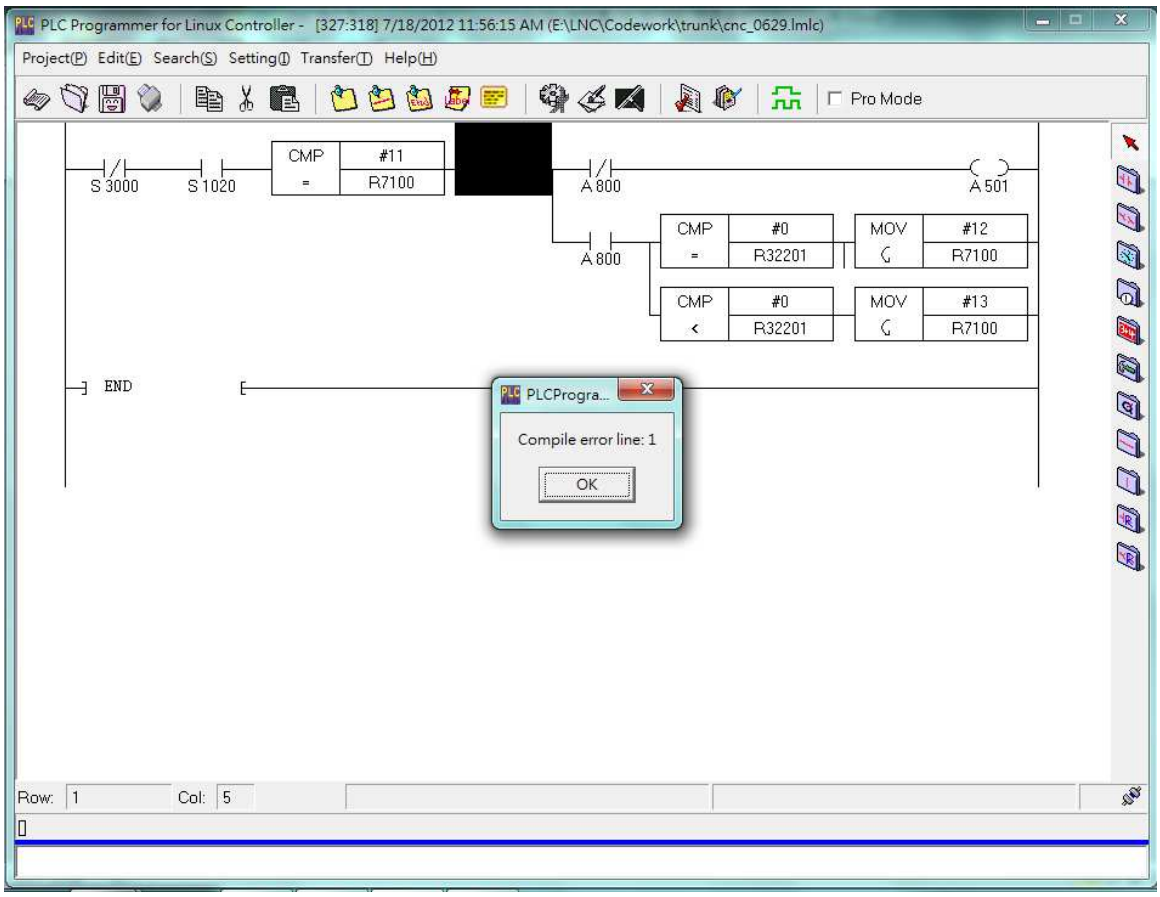


圖 六 編譯錯誤行號顯示圖

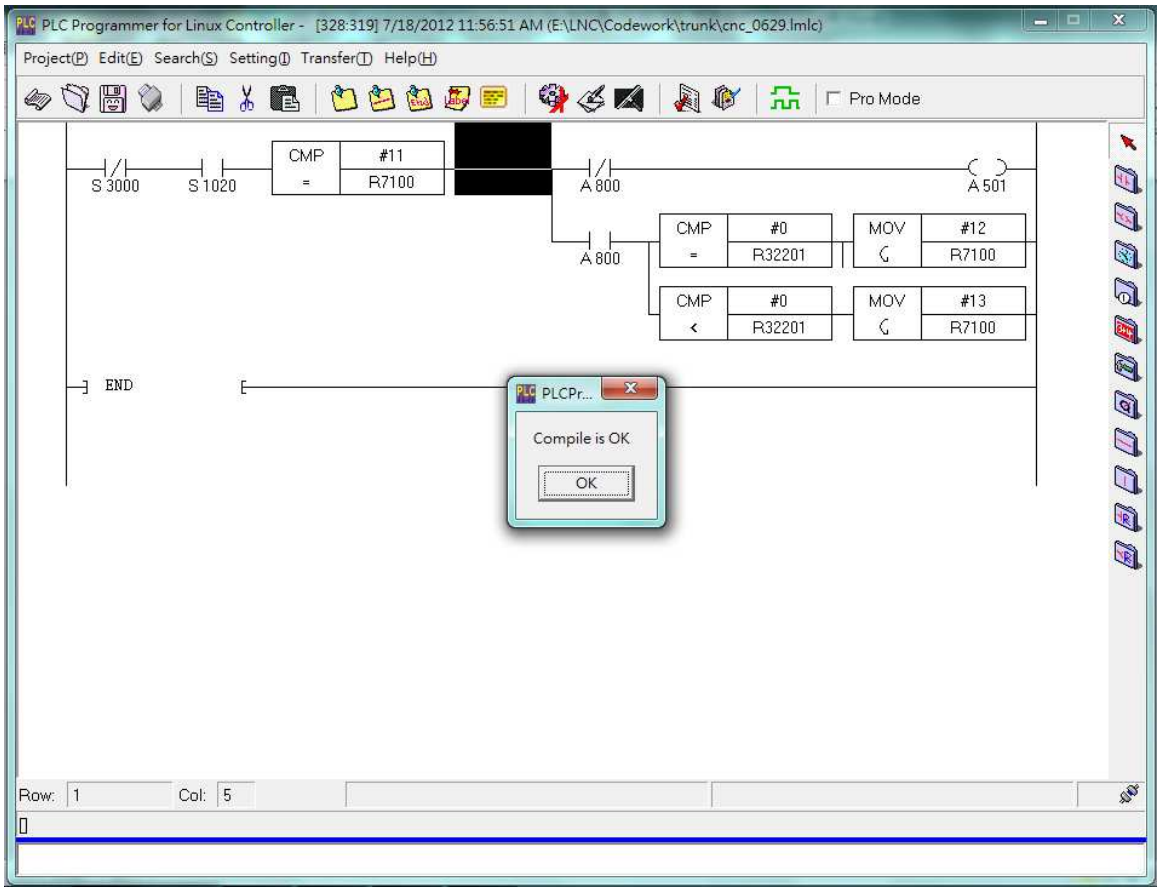


圖 七 編譯成功顯示圖

3.3.4 另存新檔(Save As + Compile)

選擇另存新檔會跳出儲存畫面，預設為【cnc.lmlc】，如圖 八，確定後同時會進行編譯動作。

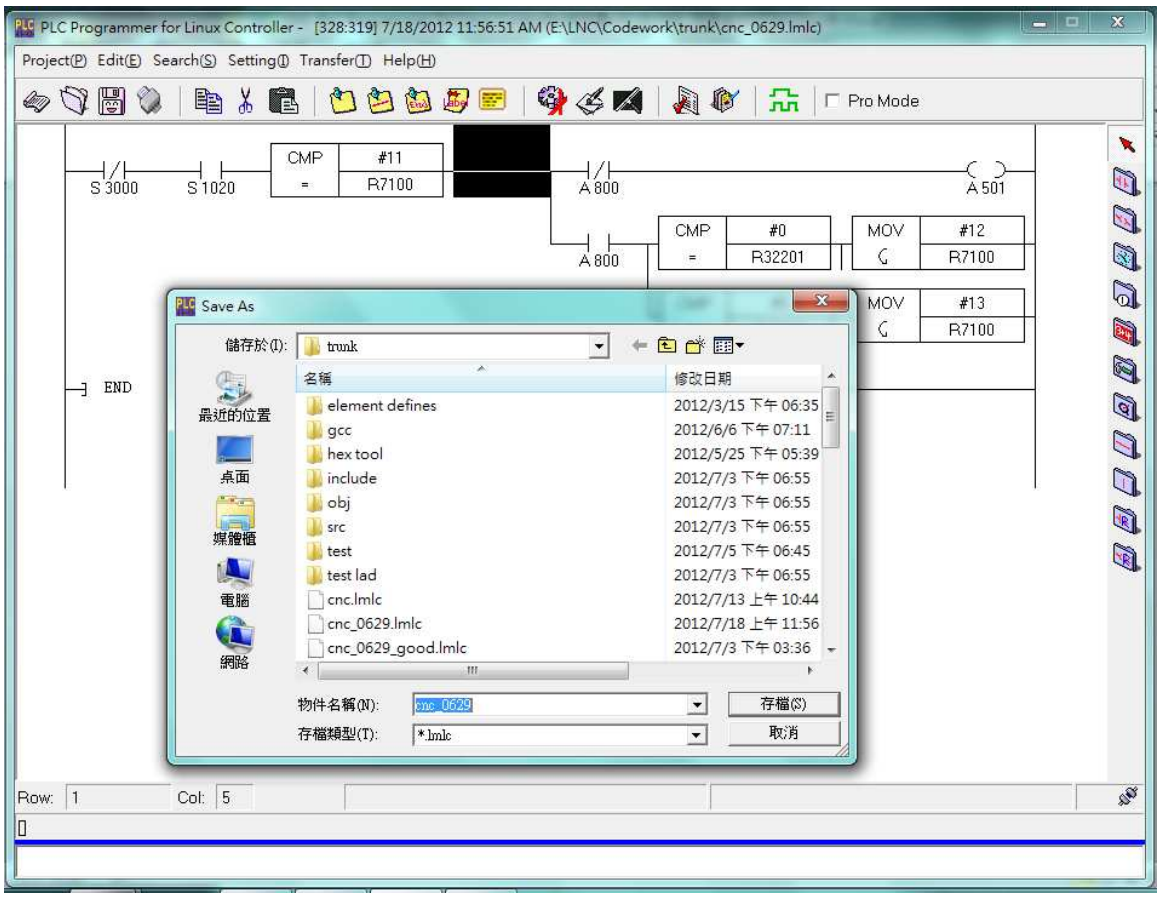


圖 八 另存新檔畫面

3.3.5 關檔(Close)

結束目前編輯的檔案，如果尚未存檔，將會跳出確認存檔的畫面，如圖 九。

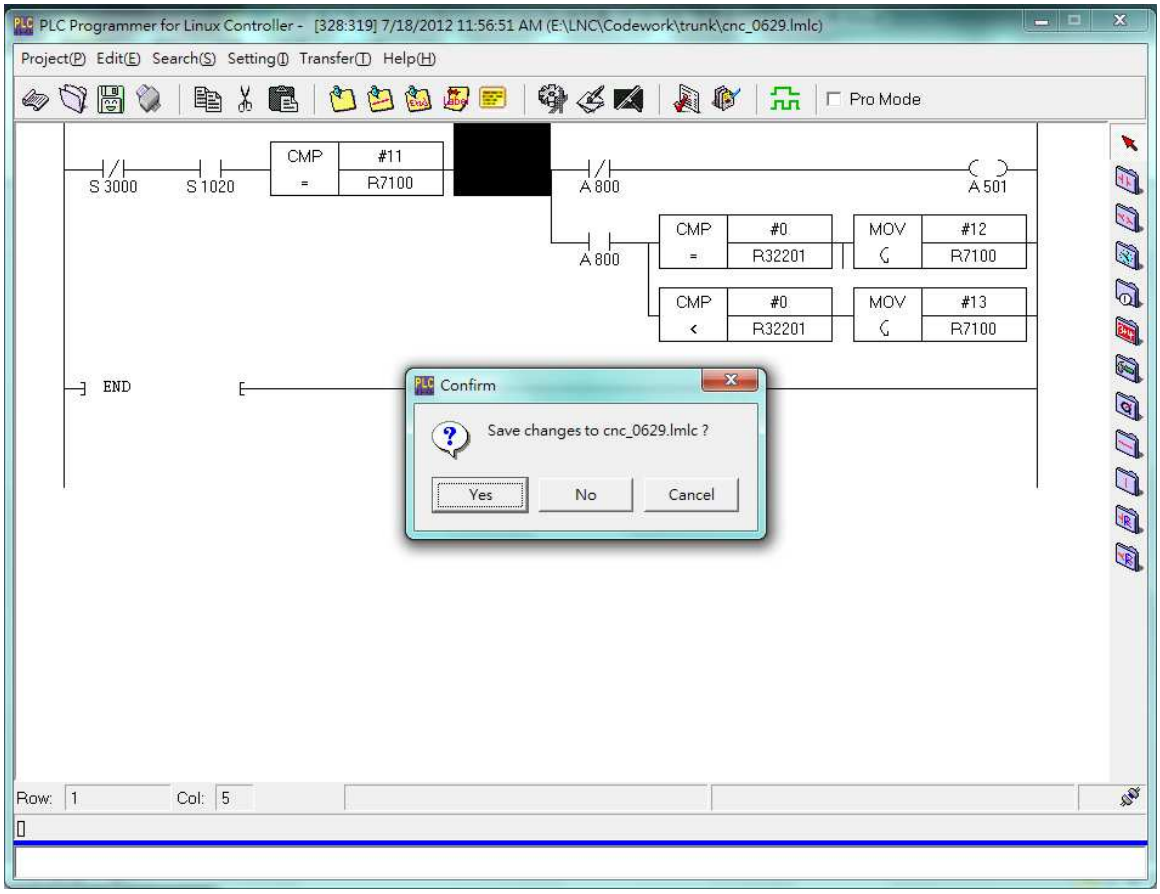


圖 九 確認存檔畫面

3.3.6 模擬器(Simulator)

啟動模擬器環境，如圖 一〇所示，可參考模擬器使用說明。

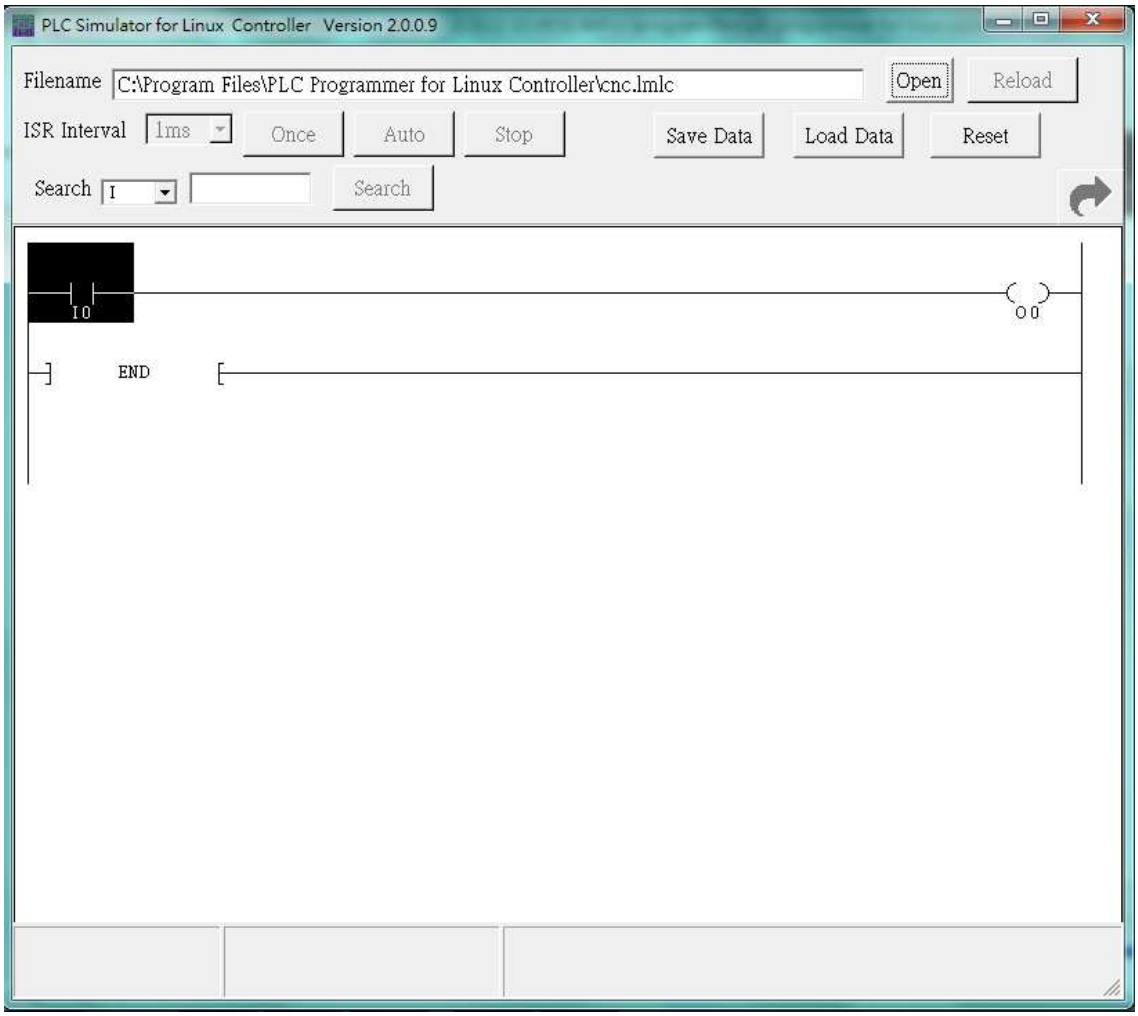


圖 一〇 模擬器環境

3.3.7 列印(Print)

1. 透過印表機列印出目前編輯之階梯圖，會跳出選單如圖 一一。

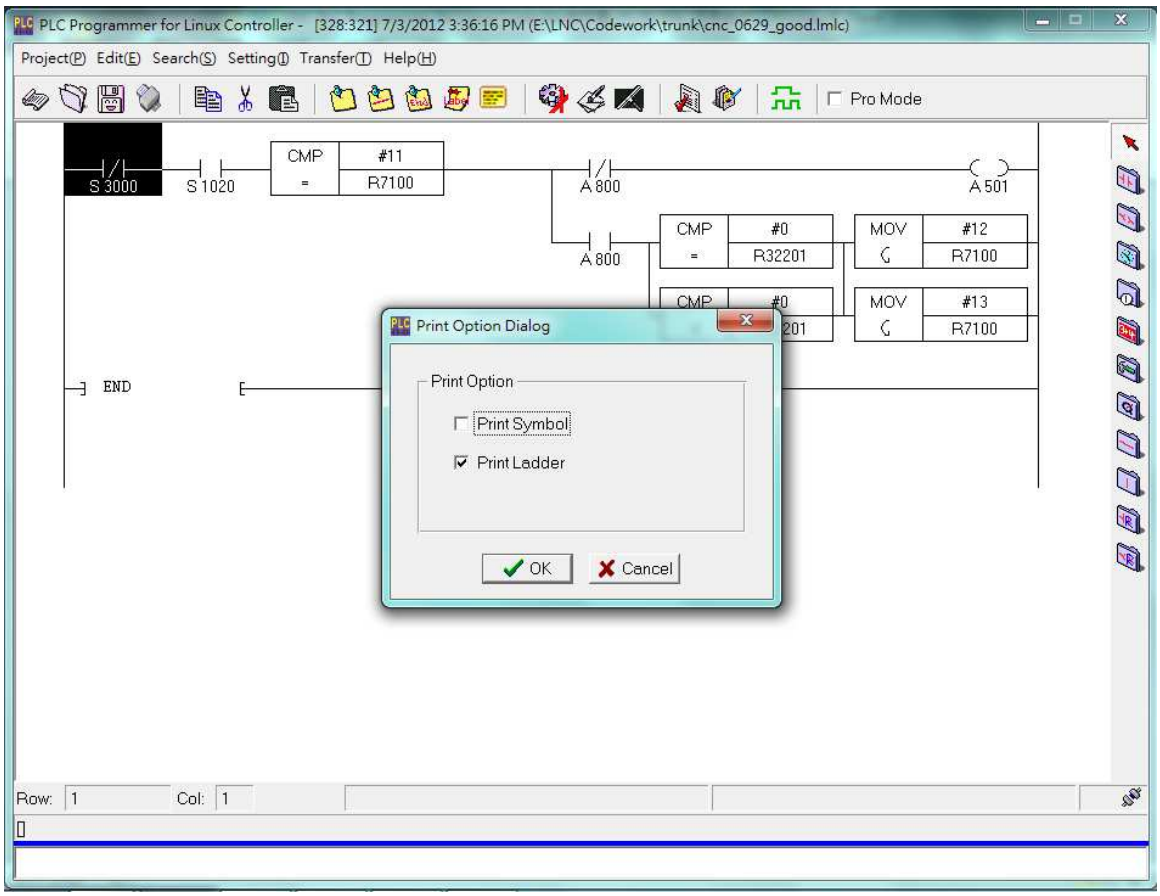


圖 一一 列印選項選單

2. 勾選是否列印提示與階梯圖。
3. 按下 OK 會跳出印表機選項設定如圖 一二。

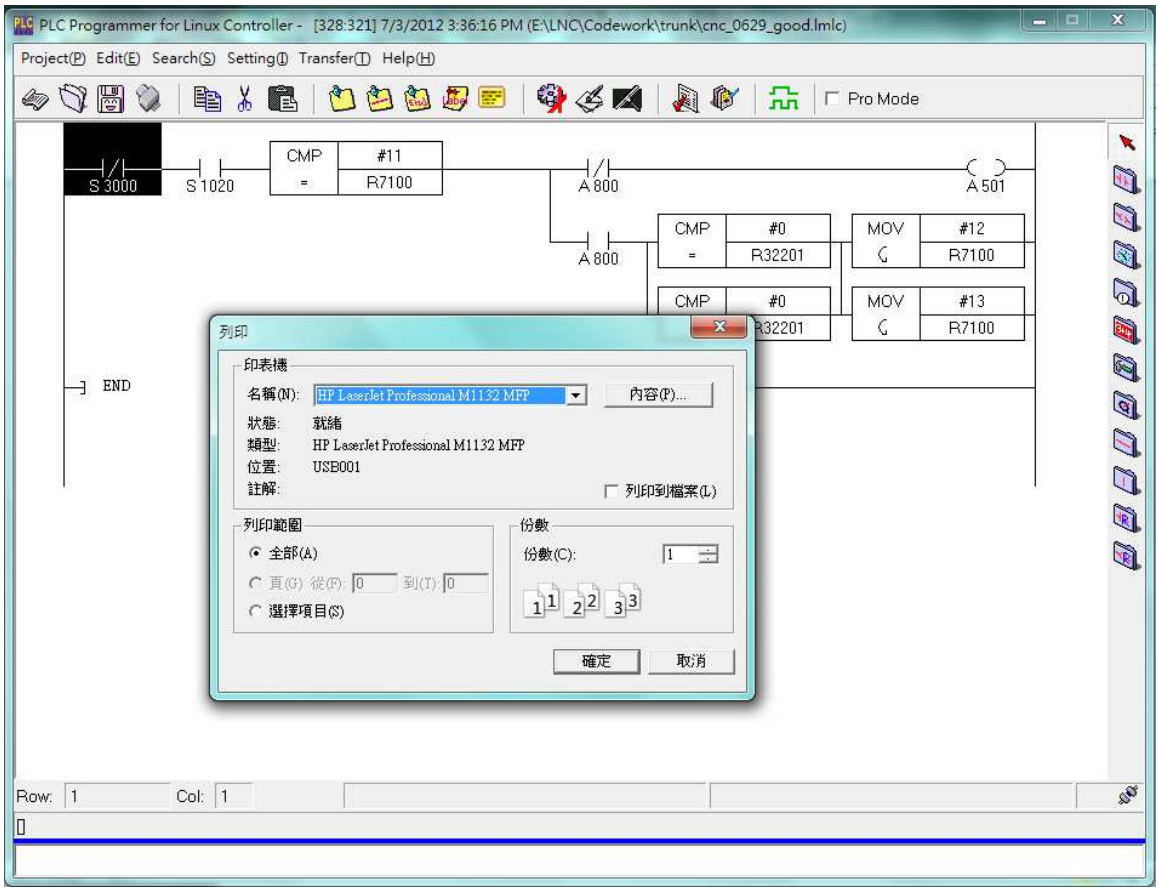


圖 一二 印表機選項

4. 按下確認後即可列印輸出。如圖 一三與圖 一四。

I Bit No	Symbol	Comment
0	緊停開關	緊停開關 外部信號 EMG SWITCH
1	重置開關 R	重置開關 RESET
2	程式啓動開	程式啓動開關 SYCLE START
3	程式停止開	程式停止開關 FEED HOLD
4	送線架警報	送線架警報信號 DECOILER ALARM
5	安全門信號	安全門信號 SECURITY DOOR
6	探針停止1	探針停止信號 TOUCH STOP
7	探針停止2	探針停止信號 1TOUCH STOP_1
8	探針警報信	探針警報信號 TOUCH ALARM
9	MPG盒使用	MPG盒使用開關 MPG BOX SWITCH
10	寸動正 JOG	寸動正 JOG +
11	寸動負 JOG	寸動負 JOG -
12	1倍	MPG / JOG X 1
13	10倍	MPG / JOG X 10
14	100倍	MPG / JOG X 100
15	X-W軸旋鈕1	X-W軸旋鈕
16	X-W軸旋鈕2	X-W軸旋鈕
17	X-W軸旋鈕3	X-W軸旋鈕
18	X-W軸旋鈕4	X-W軸旋鈕
19	X-W軸旋鈕5	X-W軸旋鈕
20	產品檢知信	產品檢知信號 PRODUCTS
21	檢長機 I	檢長機 I 點信號 GAGE_INPUT
22	X軸伺服警	X軸伺服警告信號 X_ALARM
23	Y軸伺服警	Y軸伺服警告信號 Y_ALARM
24	Z軸伺服警	Z軸伺服警告信號 Z_ALARM
25	A軸伺服警	A軸伺服警告信號 A_ALARM
26	B軸伺服警	B軸伺服警告信號 B_ALARM
27	C軸伺服警	C軸伺服警告信號 C_ALARM
28	O軸伺服警	O軸伺服警告信號 O_ALARM
29	P軸伺服警	P軸伺服警告信號 P_ALARM
30	Q軸伺服警	Q軸伺服警告信號 Q_ALARM
31	U軸伺服警	U軸伺服警告信號 U_ALARM
32	V軸伺服警	V軸伺服警告信號 V_ALARM
33	W軸伺服警	W軸伺服警告信號 W_ALARM
34	X軸原點復	X軸原點復歸信號 X_HOME
35	Y軸原點復	Y軸原點復歸信號 Y_HOME
36	Z軸原點復	Z軸原點復歸信號 Z_HOME
37	A軸原點復	A軸原點復歸信號 A_HOME
38	B軸原點復	B軸原點復歸信號 B_HOME
39	C軸原點復	C軸原點復歸信號 C_HOME
40	O軸原點復	O軸原點復歸信號 O_HOME
41	P軸原點復	P軸原點復歸信號 P_HOME
42	Q軸原點復	Q軸原點復歸信號 Q_HOME
43	U軸原點復	U軸原點復歸信號 U_HOME
44	V軸原點復	V軸原點復歸信號 V_HOME

圖 一三 列印出之 symbol 圖

圖 一四 列印出之 Ladder 圖

3.3.8 結束(Exit)

結束 PLC Programmer for Linux Controller 程式，如果尚未存檔，則會跳出確認視窗。

3.4 編輯(Edit)

3.4.1 插入一行 (Ins One Line)

包含

- 空白行(Space Line)：讓使用者容易分段或新增元件。
- 水平線(Horizontal Line)：水平線讓使用者可以不用一段一段輸入。
- 結束行(End Line)：標記為程式結束。
- 標籤行(Label Line)：作為標示或提供跳躍點所用。
- 註解行(Comment Line) ：作為程式段落說明增加可讀性。

範例如圖 十五。

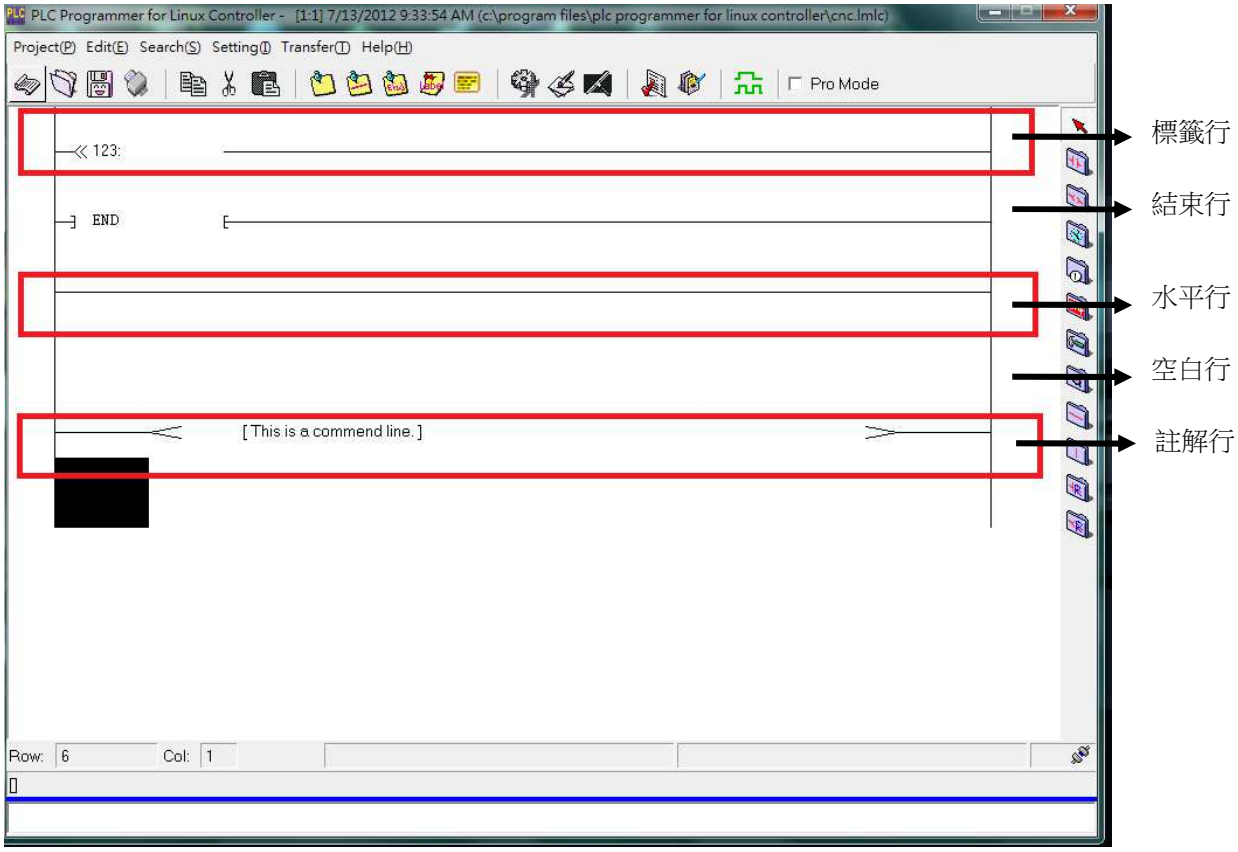


圖 十五【插入一行】範例說明

3.4.2 插入多行(Ins Multi Lines)

包含

- 空白行(Space Line)：讓使用者容易分段或新增元件。
- 水平線(Horizontal Line)：水平線讓使用者可以不用一段一段輸入。

範例如圖 一六與圖 十七。

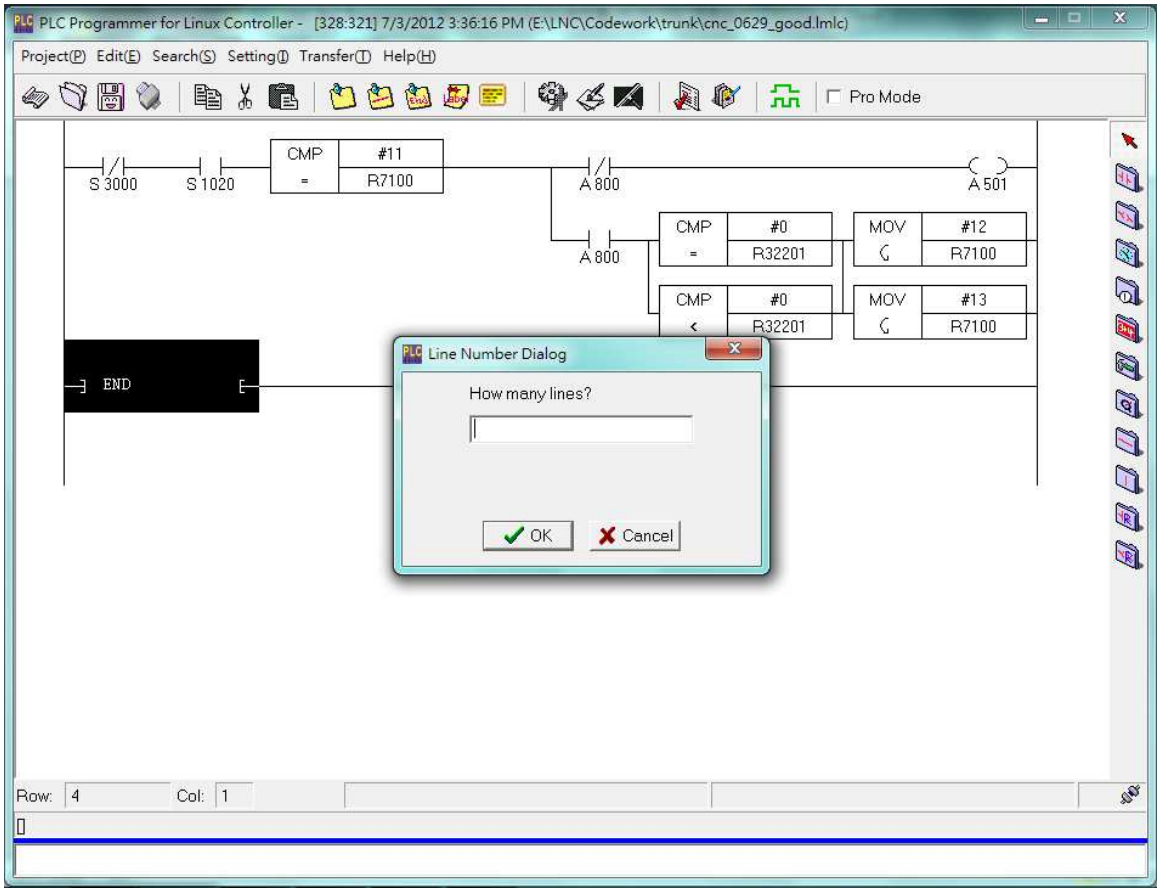

PLC Programmer for Linux Controller - [328:321] 7/3/2012 3:36:16 PM (E:\LNC\Codework\trunk\cnc_0629_good.lmlc)
Project(P) Edit(E) Search(S) Setting(I) Transfer(T) Help(H)
Pro Mode
S 3000
S 1020
CMP
#11
=
R7100
A 800
A 501
A 800
CMP
#0
=
R32201
MOV
#12
R7100
CMP
#0
<
R32201
MOV
#13
R7100
END
Line Number Dialog
How many lines?
OK
Cancel
Row: 4
Col: 1

圖 一六 插入多行詢問

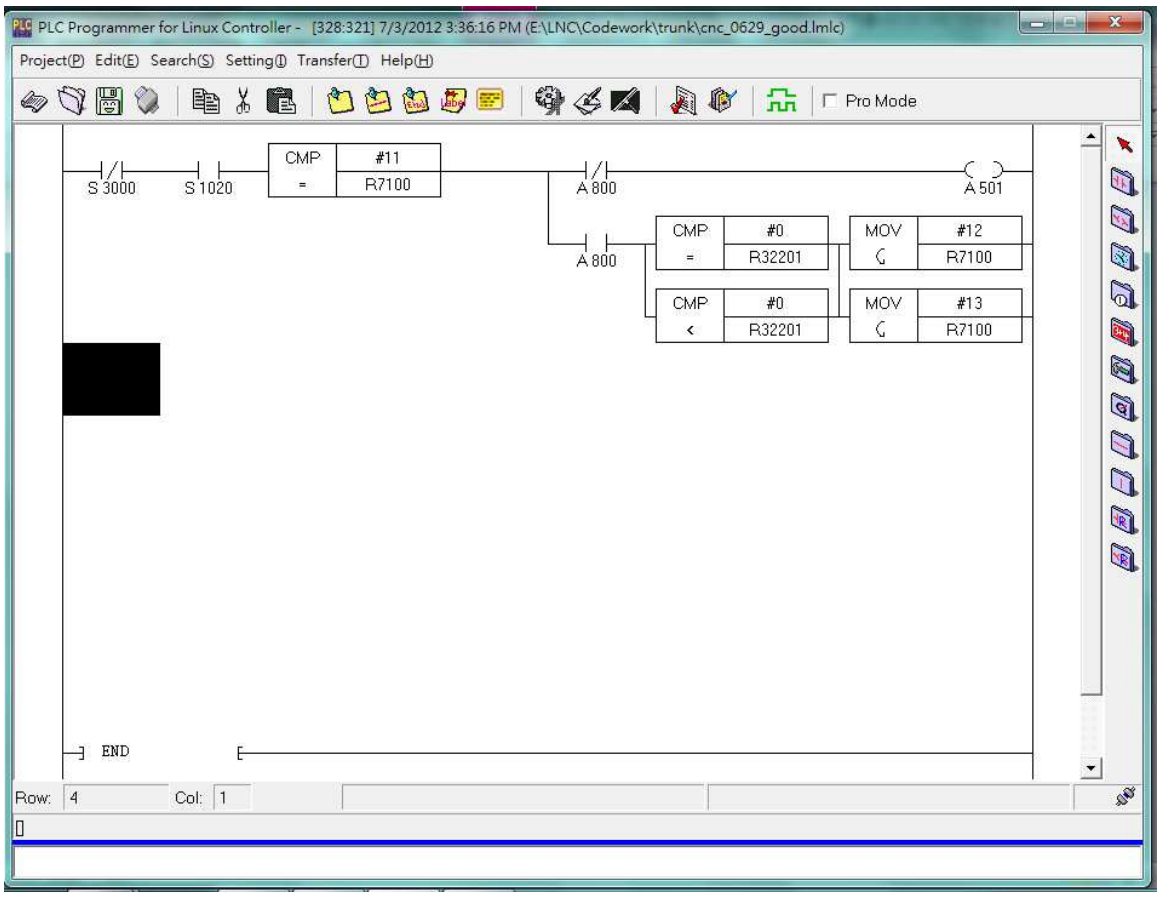


圖 十七 多行範例圖(5 行空白)

3.4.3 刪除(Delete)

包含

- 元件(Component)：包含單元件或雙元件與線段。
- 垂直線(Vertical Connect)：向上連結之線段。
- 標記列(Mark Lines)：標記起來之區塊。

詳見圖 十八之說明

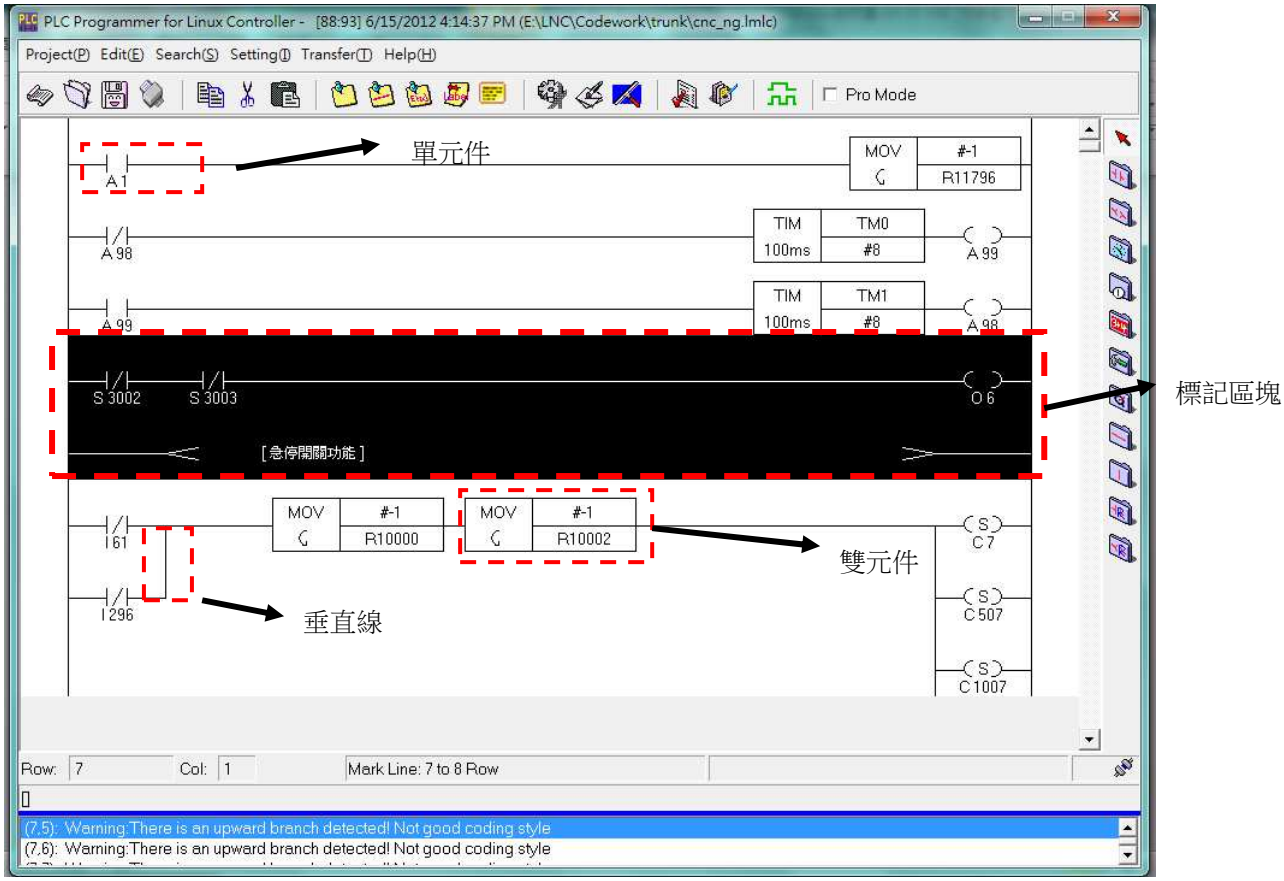


圖 十八 刪除介紹

3.4.4 複製(Copy)、貼上(Paste)、剪下(Cut)

提供單元件/雙元件及整數行區域複製編輯操作。

3.4.5 檢查(Check)

此功能最為檢查線段邏輯檢查。

如果有錯誤或警告會提示在訊息區內，如圖 十九。

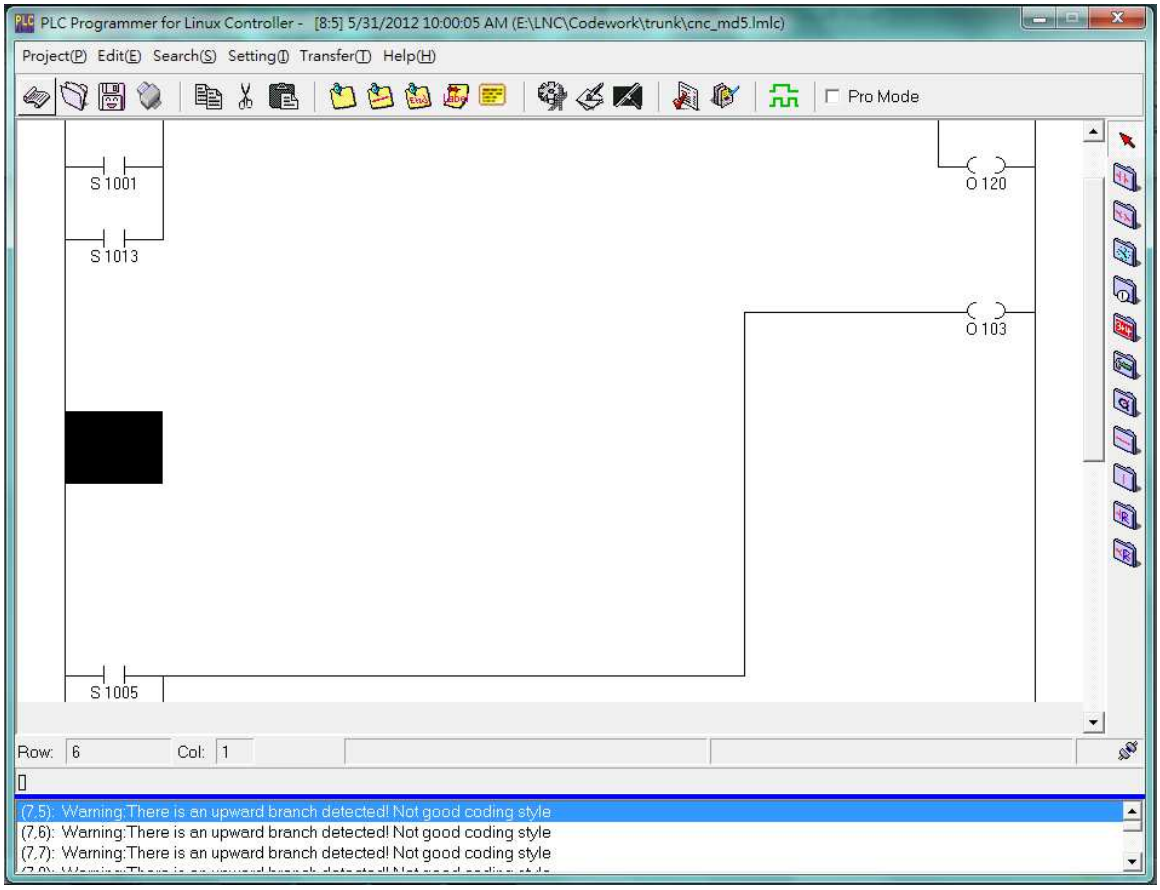


圖 十九 檢查範例

3.5 搜尋(Search)

3.5.1 找尋(Find)

分為找尋【元件位置】與【標記】兩種，點選後會跳出如圖 二〇與圖圖 二十一兩種。

收尋方式有三種【從頭找起】、【往下收尋】、【往上收尋】。

可參考 PLC Programmer 搜尋操作說明。

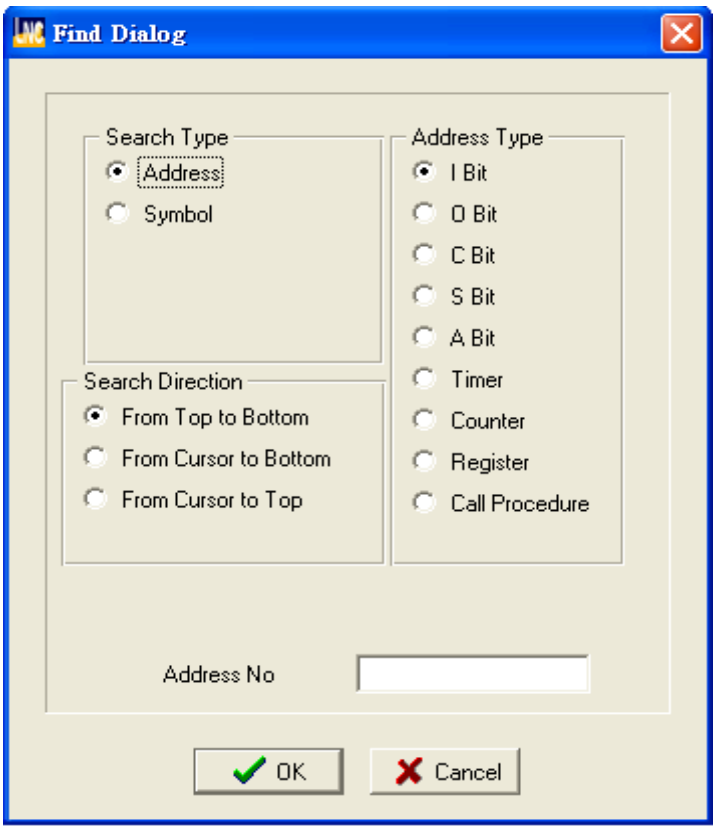

Find Dialog
Search Type
Address
Symbol
Address Type
I Bit
O Bit
C Bit
S Bit
A Bit
Timer
Counter
Register
Call Procedure
Search Direction
From Top to Bottom
From Cursor to Bottom
From Cursor to Top
Address No
OK
Cancel

圖 二〇找尋元件位置圖

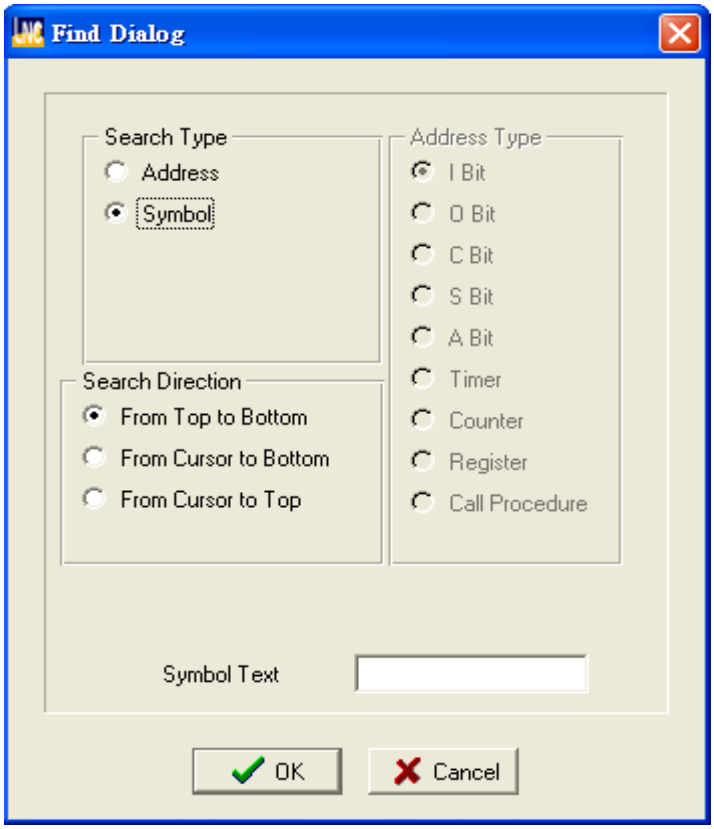

Find Dialog
Search Type
Address
Symbol
Address Type
I Bit
O Bit
C Bit
S Bit
A Bit
Timer
Counter
Register
Call Procedure
Search Direction
From Top to Bottom
From Cursor to Bottom
From Cursor to Top
Symbol Text
OK
Cancel

圖 二十一 找尋標記圖

3.5.2 尋找下一個(Find Next)

可以依照收尋的方式來找下一個符合的目標。

3.5.3 跳躍到某一行(Go To Line Number)

可以直接跳躍到某一行，輸入如圖 二十二。

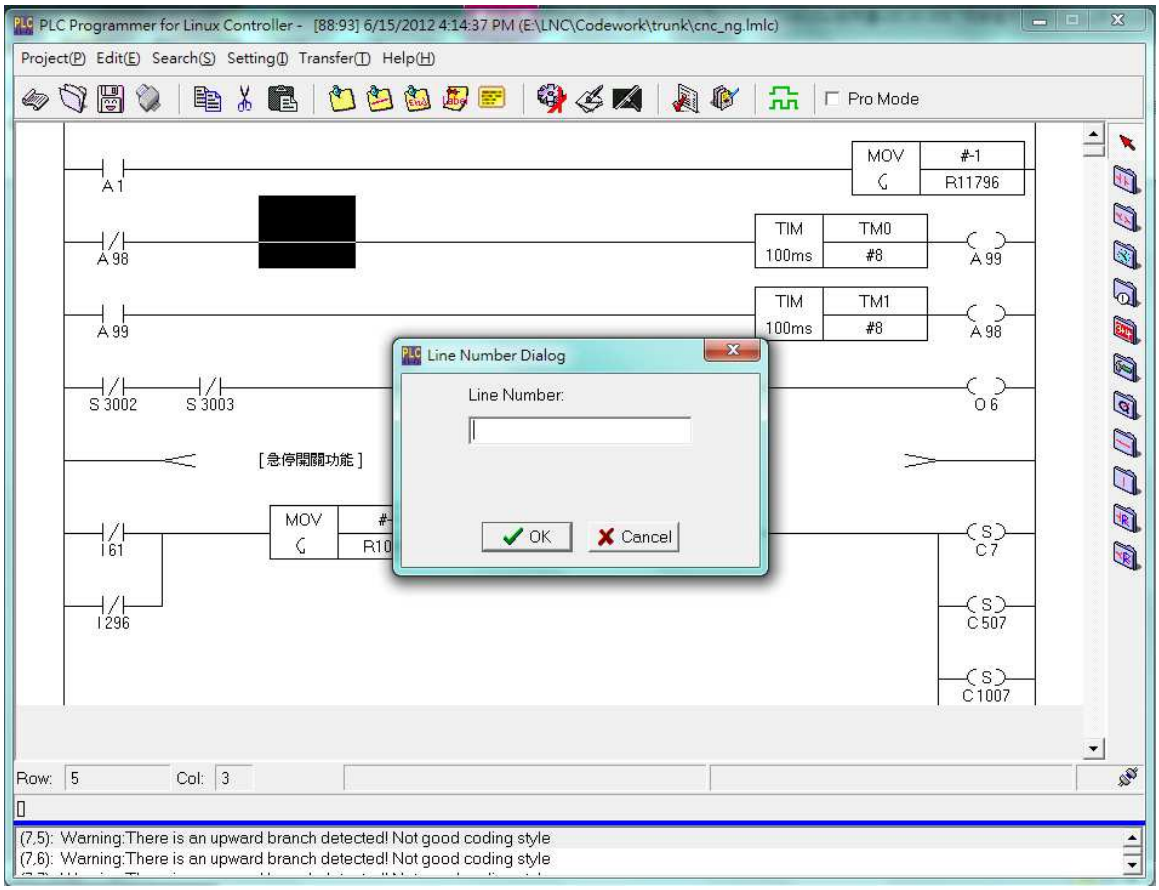


圖 二十二 跳躍至某行

3.5.4 取代(Replace)

可以尋找並取代元件的資料型態及位址，畫面如圖 二三。

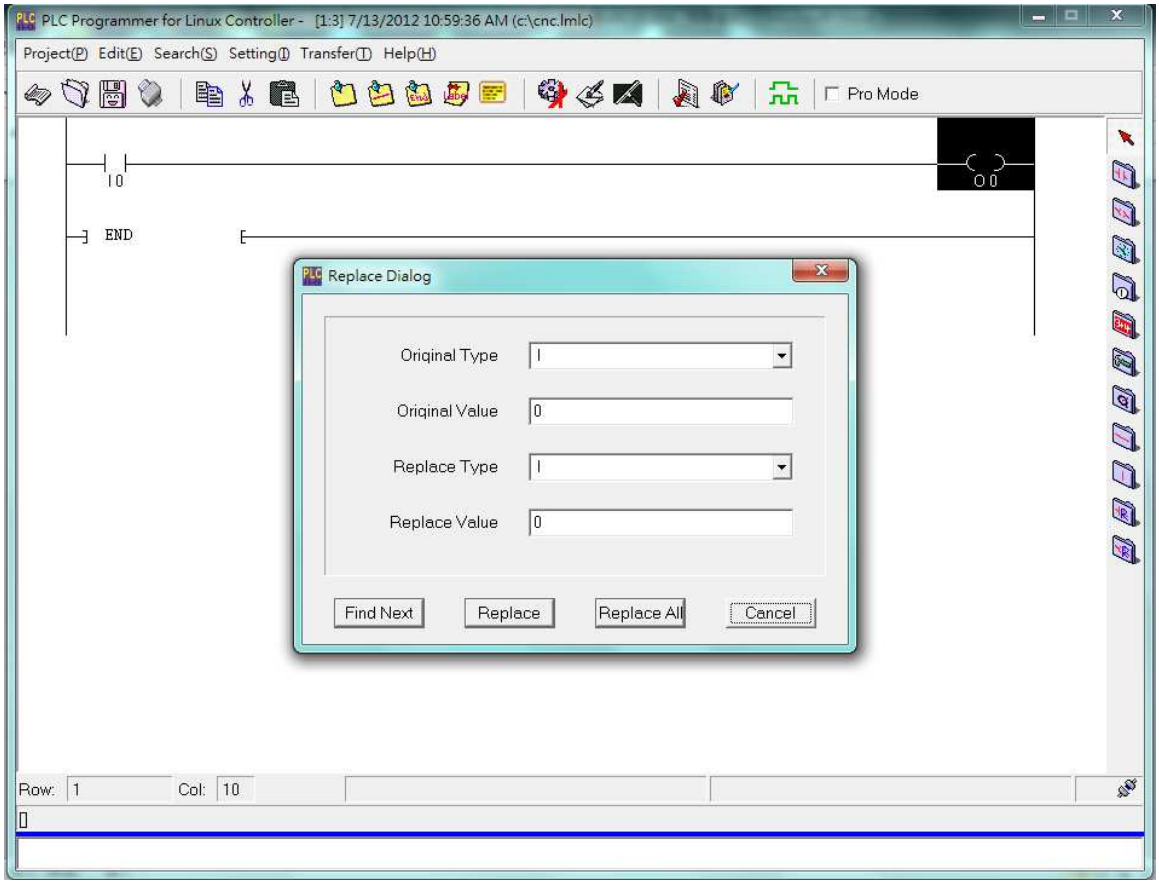


圖 二三 取代

3.6 設定(Setting)

3.6.1 提示編輯(Symbol)

編輯元件之提示與註解，如圖 二十四。

元件可分為 I Bit、O Bit、C Bit、S Bit、A Bit、Timer、Counter、Register、Register Bits。

使用請詳見 PLC 警報/警告字串編輯。

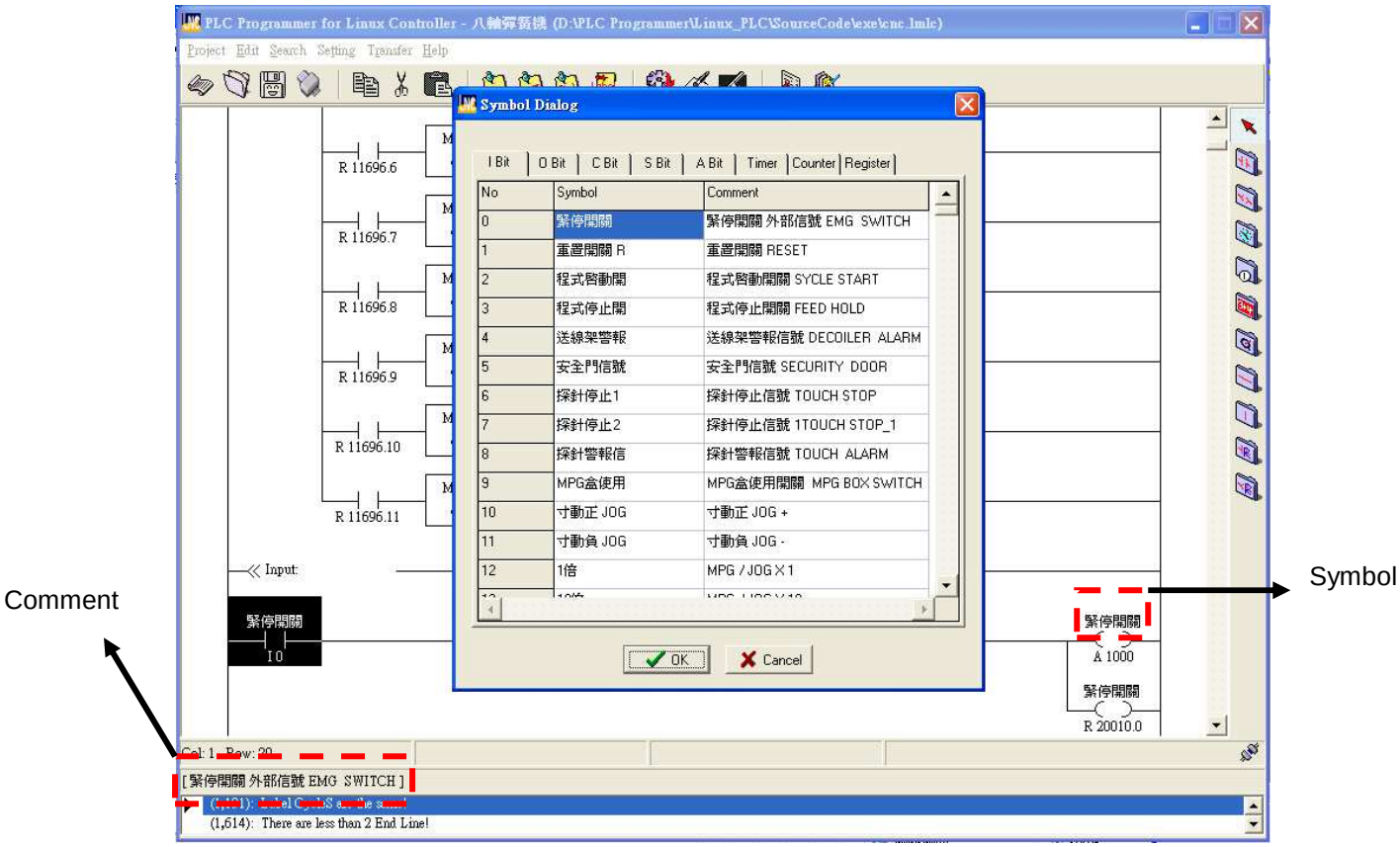


圖 二十四 提示與註解編輯

3.6.2 專案註解(Annotation)

可以針對專案加註解，詳見專案註解編輯。

3.7 傳送(Transfer)

此章節請見上傳、下載、重新載入使用說明。

3.8 幫助(Help)

3.8.1 關於(About)

顯示一些軟體資訊與公司訊息如圖 二五，詳細資訊參考 PLC 編輯器版本說明。

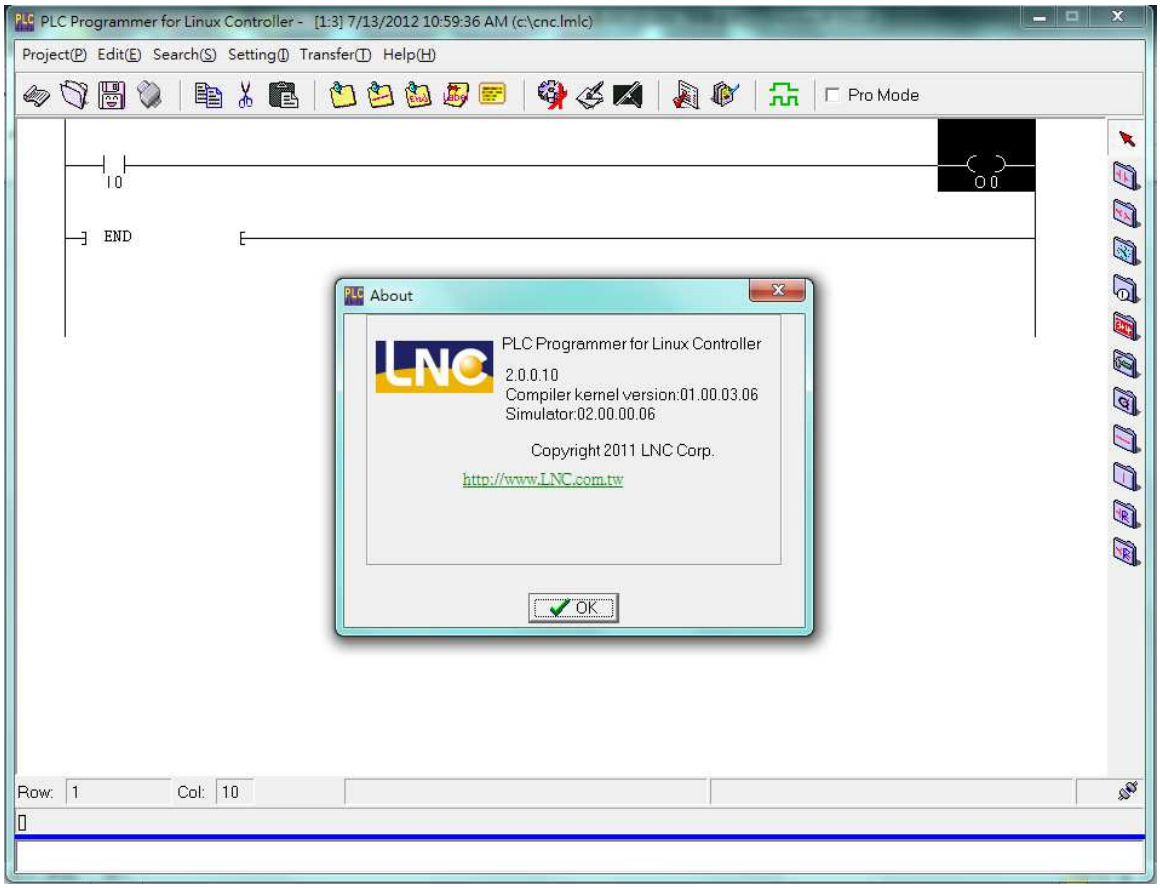


圖 二五 軟體版本資訊

4 工具列功能說明

◎ New 快捷鍵：建立新專案。

◎ Open 快捷鍵：開啟舊專案。

◎ Save 快捷鍵：儲存目前編輯專案。

◎ Print 快捷鍵：列印目前專案。

◎ Copy：複製一整列【無法複製單一元件】。

◎ Cut：剪下一整列【無法剪下單一元件】。

◎ Paste：貼上一整列【無法貼上單一元件】。

◎ Insert One Space Line：新增空白列(Shift + Ins)。

◎ Insert One Horizontal Line：新增水平線列(Ctrl + Ins)。

◎ Insert One End Line：新增結束列。

◎ Insert One Label Line：新增標籤列。

◎ Insert One Command Line：新增註解列。

◎ Delete Component：刪除單一元件(Del)。

◎ Delete Vertical Connect：刪除垂直線(Shift+Del)。

◎ Delete Mark Line：刪除一整列(Ctrl+Del)。

◎ Check：偵測PLC編輯語法正確與否。

◎ Symbol：編輯標示提示符號。

◎ Simulator：啟動模擬器。

◎ Pro mode：勾選後編譯產出cnc.lcco檔，目前供模擬器使用。

5 元件工具列說明

5.1 游標快捷鍵：

選取功能按住滑鼠的右鍵進行拖曳進行選取的動作。

5.2 Contact(接點)快捷鍵：

在新增接點處點左鍵，會出現接點屬性設定視窗圖 二六。

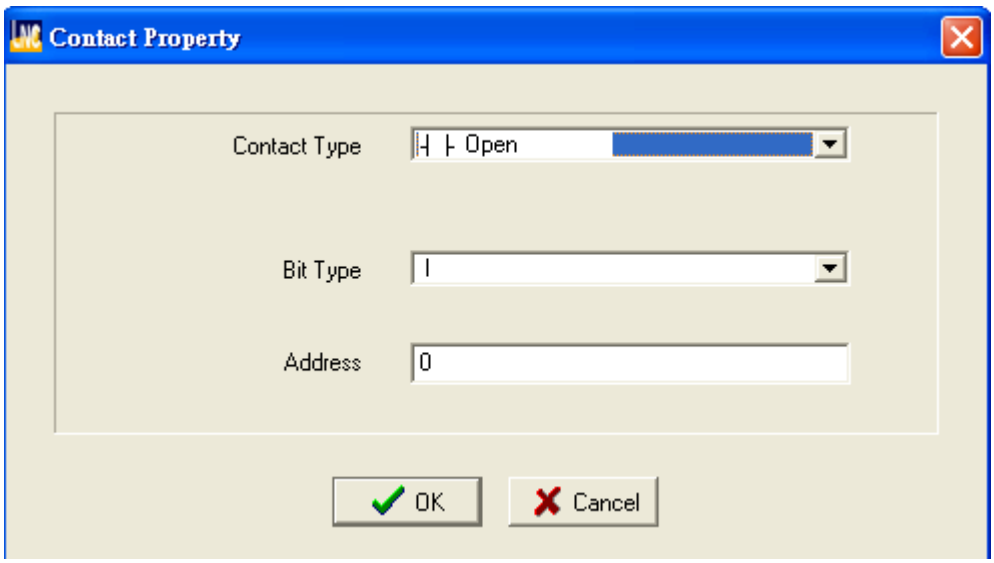


圖 二六 接點屬性設定

※ 接點型式選擇如圖 二七：

Open：常開接點。

Close：常閉接點。

Timer Open：計時器常開接點。

Timer Close：計時器常閉接點。

Counter Open：計數器常開接點。

Counter Close：計數器常閉接點。

※ 元件型式選擇(僅限Open和Close型式)如圖 二八：

I：外部元件輸入信號給PLC，讀取機械狀態。

O：PCL輸出信號給外部元件，要求機械作動。

C、S：NC與 PLC溝通界面。

A：PLC內部輔助接點。

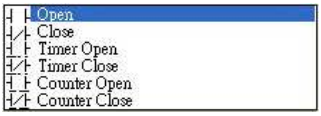


圖 二七 接點型式選擇

圖 二八 元件型式選擇

※ 輸入元件位址範圍：IOCSA點:0~4095。

名稱	說明
開接點 Normal Open	當接點通電時為True
常閉接點 Normal Close	當接點通電時為False
計時常開 Timer Open	當計時器有輸出通電時為True
保持計時常開 Latch Timer Open	當計時器有輸出通電時為True
計時常閉 Timer Close	當計時器有輸出通電時為False
保持計時常閉 Latch Timer Close	當計時器有輸出通電時為False
計數常開 Counter Open	當計數器有輸出通電時為True
計數常閉 Counter Close	當計數器有輸出通電時為False
水平連結 Horizontal Line	水平元件連接線段垂直元件連接線段
垂直連結 Vertical Line	垂直元件連接線段

表格 三 PLC 線圈功能方塊說明

5.3 Coil 快捷鍵：

在新增線圈處點左鍵，會出現線圈屬性設定視窗圖 二九。

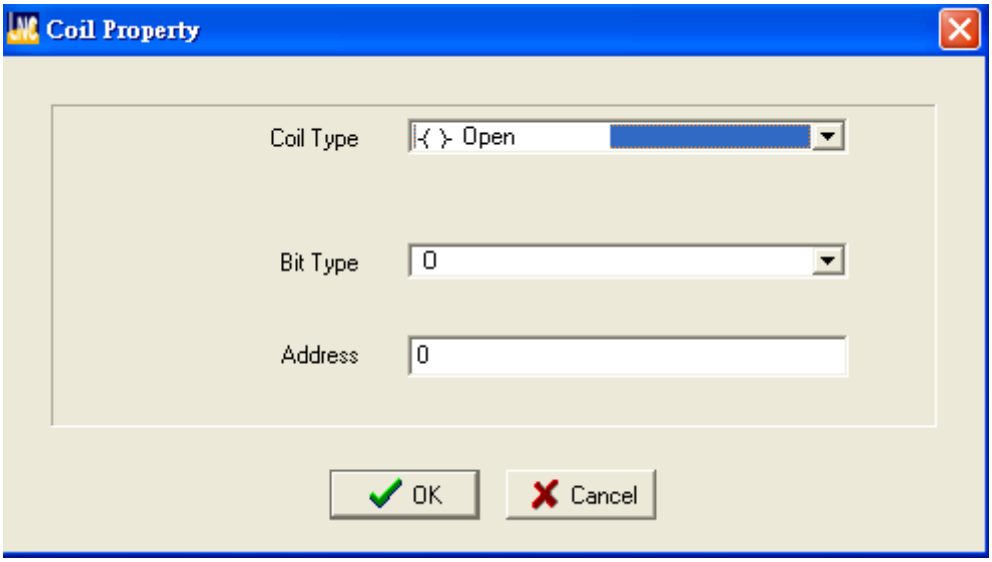


圖 二九 線圈屬性設定

※ 線圈型式選擇圖 三〇：

Open：常開接點線圈。

Close：常閉接點線圈。

Rising Edge：上緣觸發線圈。

Falling Edge：下緣觸發線圈。

Set ：自保持線圈。

Reset：復歸自保持線圈。

※ 元件型式選擇圖 三一：

O：PCL輸出信號給外部元件，要求機械作動。

C、S：NC與 PLC溝通界面。

A：PLC內部輔助接點。

a.在按下OK鍵後，若該列於游標焦點位置右方有任何元件(不包含HorConnect及VerConnect)，則會出現一個詢問使用者是否要覆蓋的對話框。

b.不論是否出現詢問使用者是否要覆蓋的對話框，在確定後，Coil元件會自動被放置於該列的最右邊位置，從原先游標焦點的位置到最右邊的位置間會自動填入HorConnect元件。

圖 三〇線圈型式選擇

圖 三一 元件型式選擇

名稱		說明
常開線圈	Normal Open	當True時線圈通電
常閉線圈	Normal Close	當False時線圈通電
正緣觸發線圈	Rising Triggered	正緣觸發線圈通電
負緣觸發線圈	Falling Triggered	負緣觸發線圈通電
正緣觸發保持	Latch	正緣觸發線圈通電並保持
負緣觸發覆歸	Unlatch	負緣觸發線圈覆歸

表格 四 PLC 線圈功能方塊說明

5.4 Timer 快捷鍵：

在新增計時器處點左鍵，會出現計時器屬性設定視窗圖 三二。

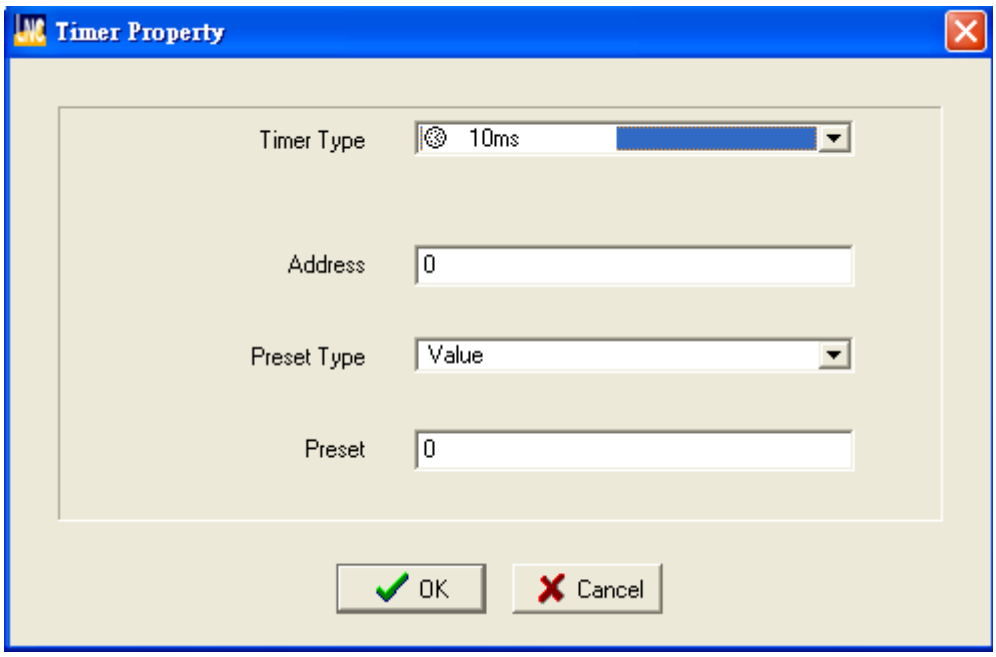


圖 三二計時器屬性設定

※ 計時器型式選擇如圖 三三：

1ms：以1毫秒為計時單位。ps.中斷必須設定為1ms才準確

10ms：以10毫秒為計時單位。

100ms：以100毫秒為計時單位。

1s：以1秒為計時單位。

※ 設定值型式選擇如圖 三四：

Value：常數型式(0 ~ 2147483647)。(此型式下設定值請輸入任意常數)

Register：暫存器型式(0~29999)。 (此型式下設定值會讀取暫存器內容數值)

圖 三三 計時器型式選擇

圖 三四 設定值型式選擇

※ 輸入元件位址範圍：0~255。

名稱	說明
1毫秒計時器 1ms Timer	以1毫秒為計時單位
10毫秒計時器 10ms Timer	以10毫秒為計時單位
100毫秒計時器 100ms Timer	以100毫秒為計時單位

1秒計時器	1s Timer	以1秒為計時單位

表格 五 PLC 計時器功能方塊說明

5.5 Counter 快捷鍵：

在新增計數器處點左鍵，會出現計數器屬性設定視窗圖 三五。

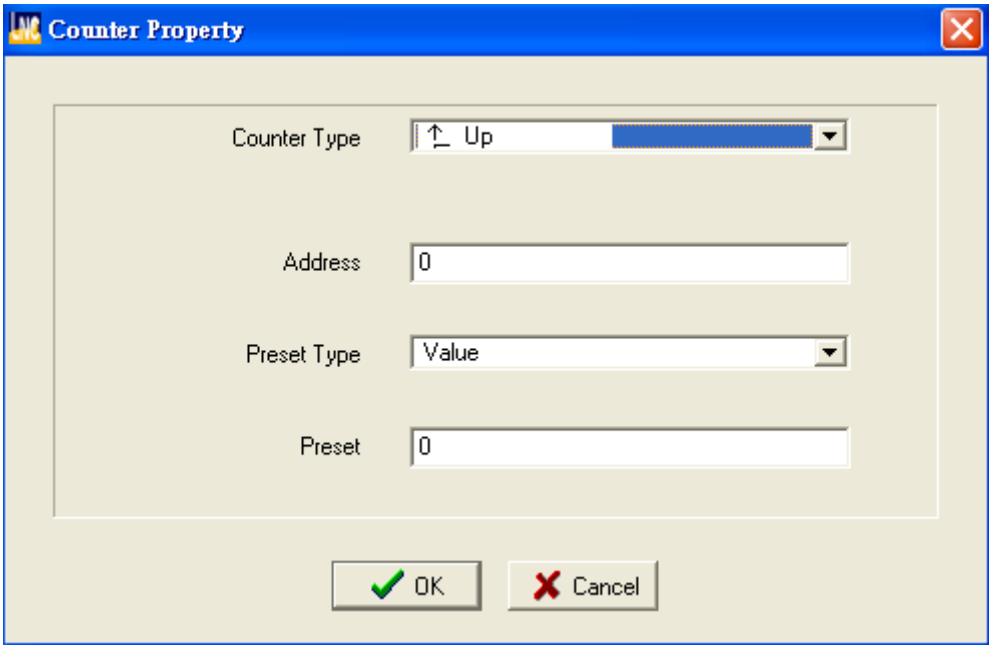


圖 三五 計數器屬性設定

※ 計數器型式選擇如圖 三六：

UP：由 0 遞增計數至計數設定值。

Down：由計數設定值遞減計數至 0。

Ring UP：由 0 遞增計數至計數設定值後再由 0 遞增計數至計數設定值。

Ring Down：由計數設定值遞減計數至 0 後再由計數設定值遞減計數至 0。

※ 設定值型式選擇如圖 三七：

Value：常數型式(0 ~ 2147483647)。(此型式下設定值請輸入任意常數)

Register：暫存器型式(0~29999)。 (此型式下設定值會讀取暫存器內容數值)

※ 當Counter型式是Reset時，Preset型式及Preset無效。

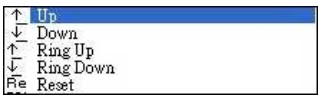


圖 三六 計數器型式選擇

圖 三七 設定值型式選擇

※ 輸入元件位址範圍：0~255。

名稱		說明
上計數器	Up Counter	由0遞增計數至計數設定值
下計數器	Down Counter	由計數設定值遞減計數至0
循環上計數器	Ring Up Counter	由0遞增計數至計數設定值後再由0開始遞增計數至計數設定值
循環下計數器	Ring Down Counter	由計數設定值遞減計數至0後再由設定值開始遞增計數至0
重置計數值	Reset Counter	重置計數值為設定值

表格 六 PLC 計數器功能方塊說明

5.6 Arithmetic 快捷鍵：

在新增運算元件處點左鍵，會出現運算元件屬性設定視窗圖 三十八。

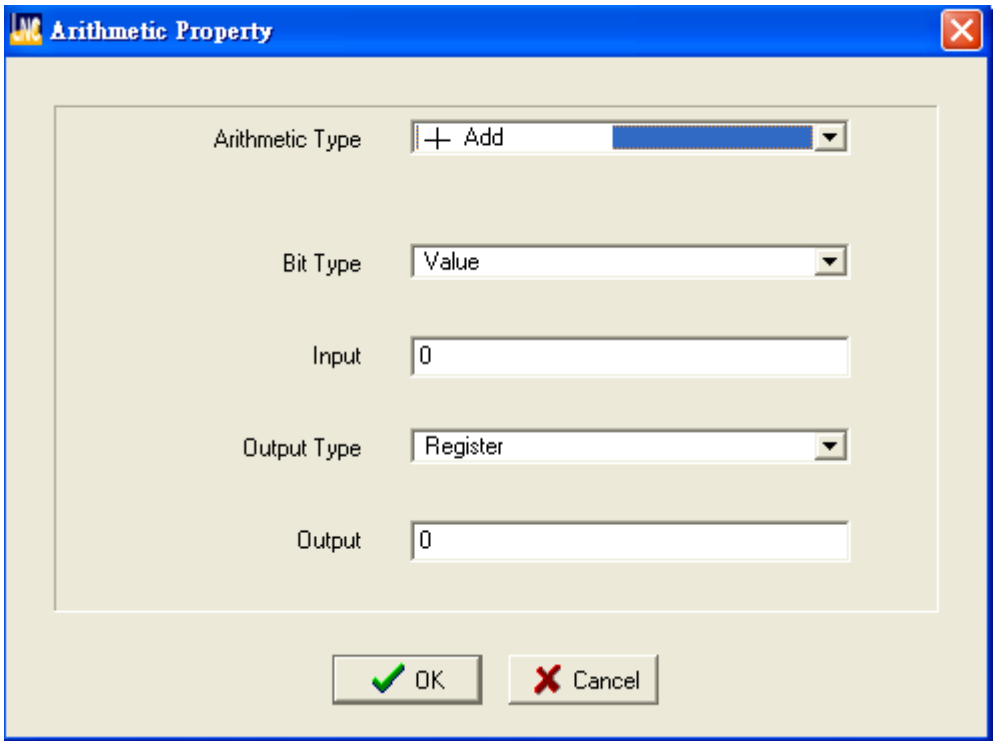


圖 三十八 運算元件屬性設定

※ 運算元件型式選擇 圖 三十九：

+Add：加法元件。

-Sub：減法元件。

×Mul：乘法元件。

÷Div：除法元件。

↶Move：搬移元件。

>More than：大於元件。

<Less than：小於元件。

=Equal to：比較元件。

&And：邏輯and元件。

|Or：邏輯or元件。

∧Xor：邏輯xor元件。

Mulrini：對多個R一次設值。

Mulrcpy：將一連續區域R值拷背至另一連續區域。

I/Rmap：將連續32個I值映射至指定R。

O/Rmap：將連續32個O值映射至指定R。

※ 輸入型式選擇 圖 四〇：

Value：常數型式(±2147483647)。

Register：暫存器型式(0~29999)。

Note: I/Rmap及O/Rmap輸入分別為I，O接點型式(0~4095)。

※ 輸出型式選擇 圖 四一：

Register：暫存器型式(0~29999)。

※ add元件當導通時，在導通期間數值會不斷累加，直到截止時數值才會停止累加。

※ Mulrini及Mulrcpy有三個輸入區域如圖 四二及圖 四三所示。

- Mulrini第一個輸入為起始暫存器，第二個輸入為要由起始暫存器初使化多少暫存器，第三個輸入為設定的初使值。
- Mulrcpy第一個輸入為來源暫存器起始位置，第二個輸入為目標暫存器起始位置，第三個輸入為拷背數目 (來源及目標區域不可重疊)。

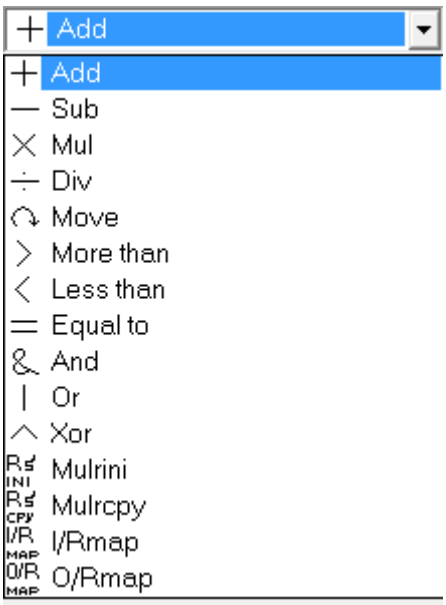


圖 三十九 運算元件型式選擇

圖 四〇 輸入型式選擇

圖 四一 輸出型式選擇

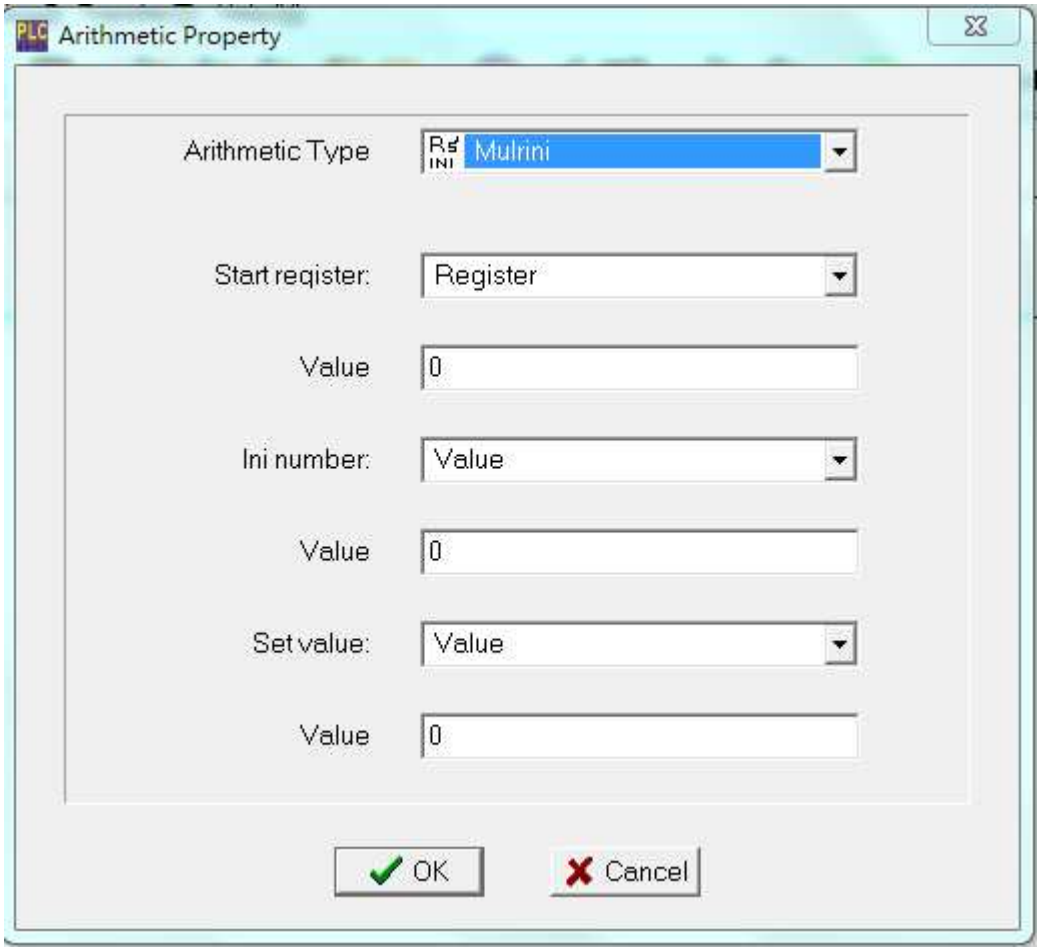


圖 四二 Mulrini 設定

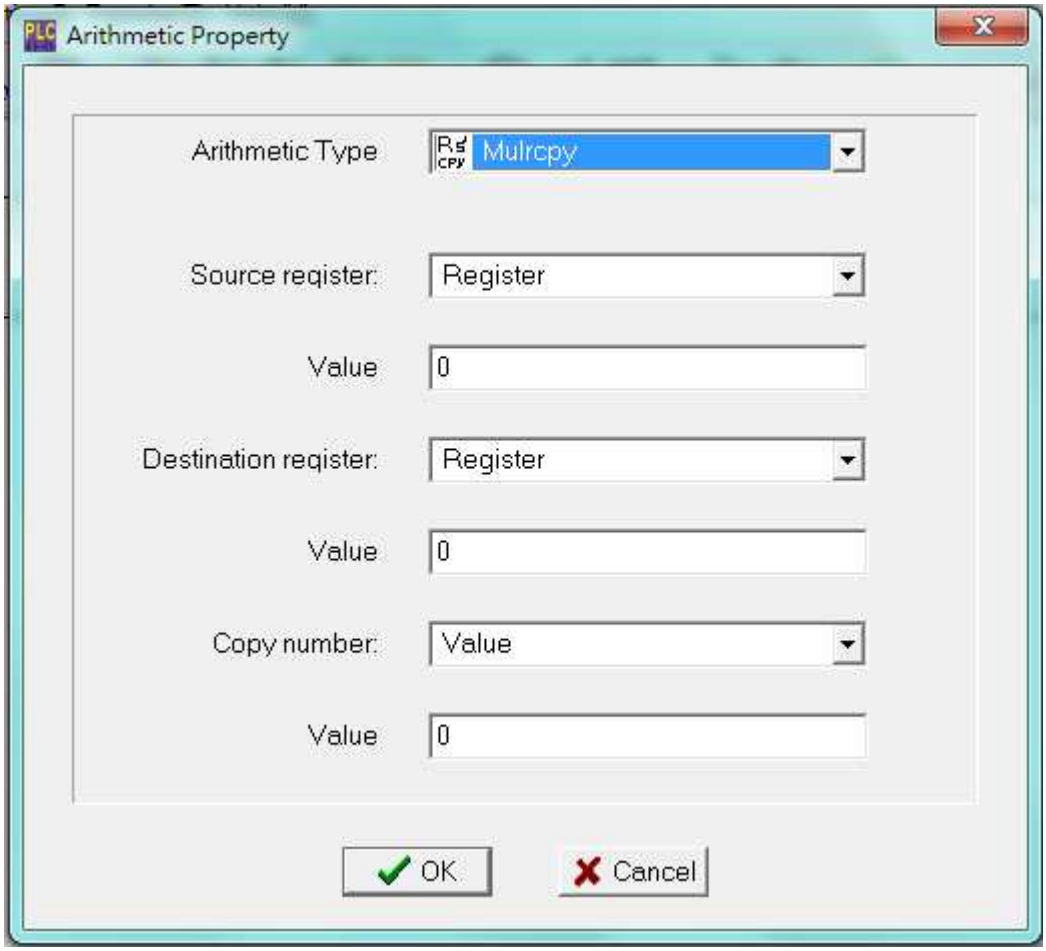


圖 四三 Mulrcpy 設定

名稱	說明
加Add	將數值或暫存器中數值相加放到暫存器中
減Subtract	將數值或暫存器中數值相減放到暫存器中
乘Multiply	將數值或暫存器中數值相乘放到暫存器中
除Divide	將數值或暫存器中數值相除放到暫存器中
搬移Move	將數值或暫存器中數值搬移放到暫存器中
大於Compare Large	比較數值或暫存器中數值是否大於
小於Compare Less	比較數值或暫存器中數值是否小於
等於Compare Equal	比較數值或暫存器中數值是否等於
且AND	邏輯且運算
或OR	邏輯或運算
互斥或Exclusive OR	邏輯互斥或運算
Mulrini	對多個R一次設值
Mulrcpy	將一連續區域R值拷背至另一連續區域

I/Rmap	將連續32個I值映射至指定R
O/Rmap	將連續32個O值映射至指定R

表格 七 PLC 數學運算功能方塊說明

5.7 Tool 快捷鍵：

在新增工具元件處點左鍵，會出現運算元件屬性設定視窗圖 四十四。

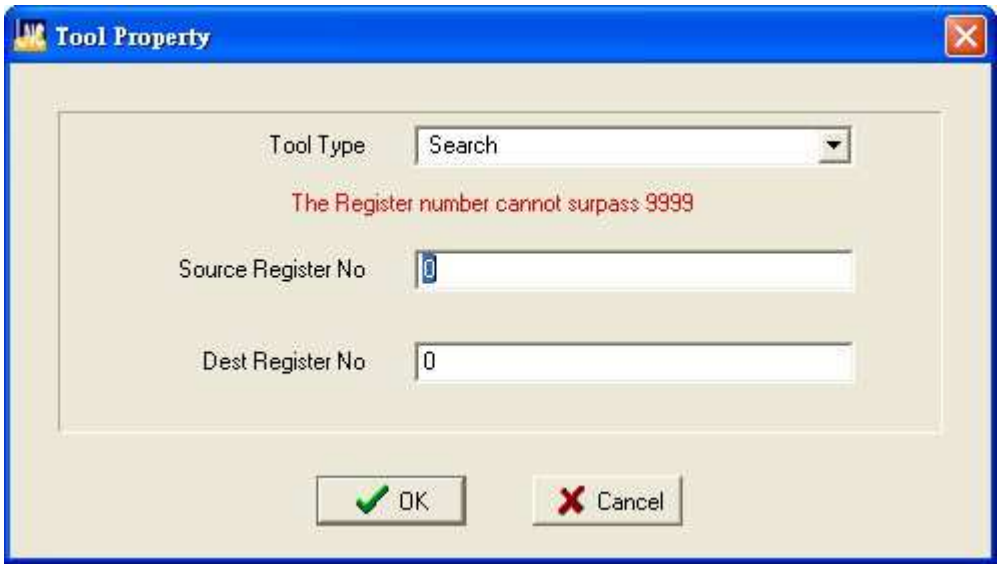


圖 四十四運算元件屬性設定

※ 工具元件型式選擇圖 四十五：

Search：搜尋元件。

Rotate：計算旋轉元件。

※ 輸入型式：

Register：暫存器型式(0~9999)。

※ 輸出型式：

Register：暫存器型式(0~9999)。

※ 元件當導通時，在導通期間會不斷作用，直到不導通才會停止作用。

圖 四十五 運算元件型式選擇

名稱	說明
Search	收尋刀庫中的刀號所在刀套編號
Rotate	計算目前刀套編號到目標刀套的移動旋轉次數

表格 八 PLC 工具功能方塊說明

5.8 Sequence 快捷鍵：

在新增程式流程處點左鍵，會出現程式流程屬性設定視窗圖 四十六。

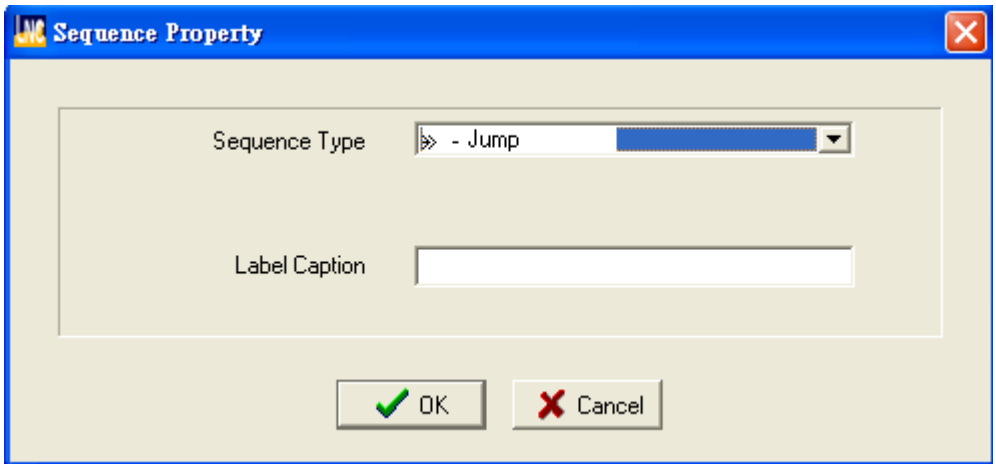


圖 四十六程式流程屬性設定

※ Sequence元件型式選擇圖 四十七

Jump：當執行程式跳躍指令時，程式跳到與符號標記相同的地方，再繼續往下執行程式。

Jump Sub Routine：執行副程式呼叫指令，程式跳與符號標記相同的地方開始執行，執行結束則返回執行下一個元件指令 。

Return：副程式結束指令，此指令只可以搭配副程式成對使用，若無成對使用或巢狀副程式將會在程式檢查時發生錯誤。

a.標記：可輸入超過六個英文數字，但Ladder在顯示與儲存時最多只能有六個英文數字。

b.當Sequence 型式是Return時，標記無效。

c.當Sequence 型式選擇Jump或Return時，在按下OK鍵後，若該列於游標焦點位置右方有任何元件(不包含HorConnect及VerConnect)，則會出現一個詢問使用者是否要覆蓋的對話框。

d.不論是否出現詢問使用者是否要覆蓋的對話框，在確定後，Sequence元件會自動被放置於該列的最右邊位置，從原先游標焦點的位置到最右邊的位置間會自動填入HorConnect元件。

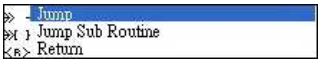


圖 四十七 程式流程型式選擇

名稱	說明
跳躍JUMP	跳躍到指定的標籤列執行
呼叫副程式Call Subroutine	呼叫副程式執行
標籤Label	程式列行號標籤或副程式名稱宣告
返回主程序Return	副程式執行結束返回主程序
區塊結束End	程序區塊結束標籤

表格 九 PLC 程式流程功能方塊說明

5.9 Horizontal Connect 快捷鍵：

在新增水平線處點左鍵，可以建立新的水平線也可按Shift + H組合鍵。

5.10 Vertcal Connect 快捷鍵：

在新增垂直線處點左鍵，可以建立新的垂直線也可按Shift + V組合鍵。

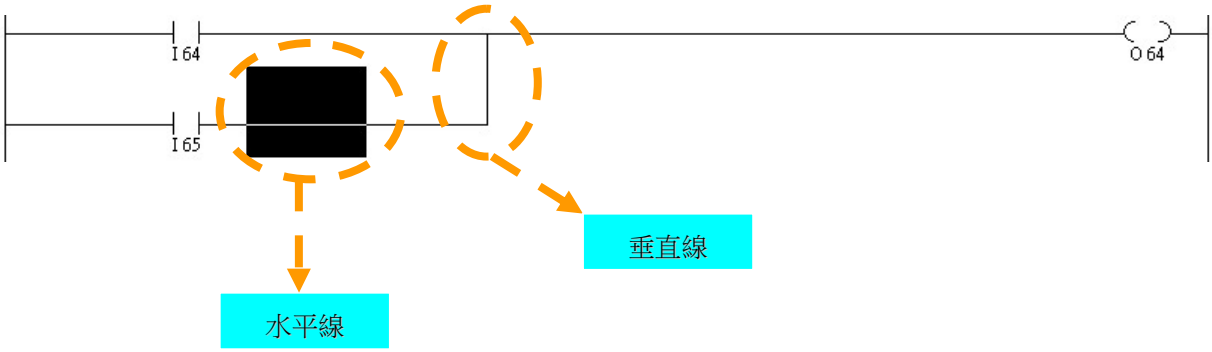


圖 四八連結線

名稱	說明
水平連結 Horizontal Line	水平元件連接線段垂直元件連接線段
垂直連結 Vertical Line	垂直元件連接線段

表格 十 PLC 連結功能方塊說明

5.11 R-Contact 快捷鍵：

在新增接點處點左鍵，會出現 R 接點屬性設定視窗圖 四九。

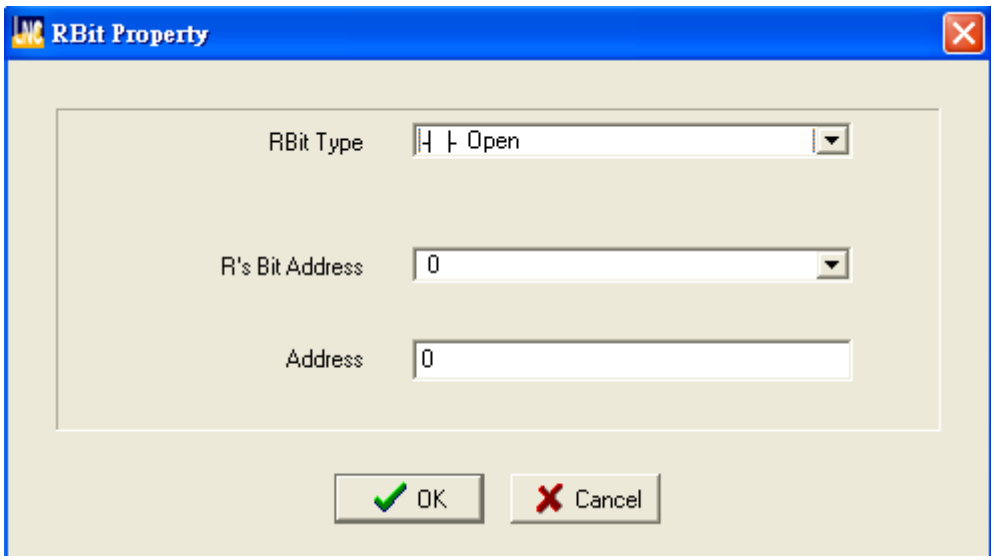


圖 四九 R 接點屬性設定

※ 接點型式選擇如圖 五〇：

Open：常開接點。

Close：常閉接點。

※ R-bit選擇如圖 五一：

Bit編號：0~31。

圖 五〇 接點型式選擇

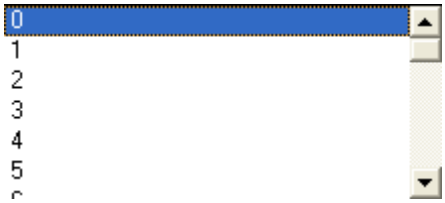


圖 五一 R-bit 選擇

名稱	說明
開接點 Normal Open	當接點通電時為True
常閉接點 Normal Close	當接點通電時為False

表格 十一 PLC 線圈功能方塊說明

5.12 R-Coil 快捷鍵：

在新增線圈處點左鍵，會出現線圈屬性設定視窗圖 五二。

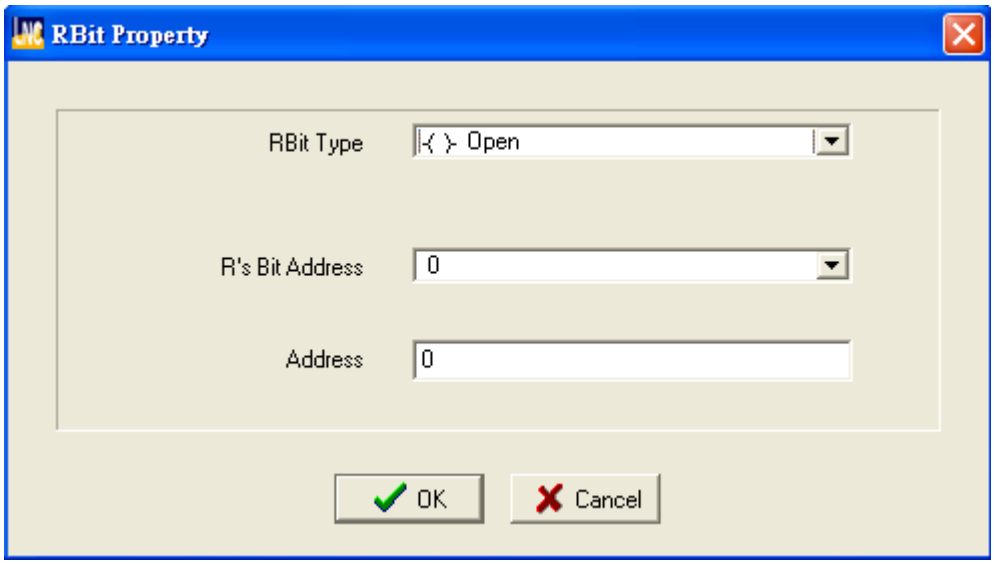


圖 五二 線圈屬性設定

※ 線圈型式選擇如圖 五三：

Open：常開接點線圈。

Close：常閉接點線圈。

※ R-bit選擇如圖 五四：

Bit編號：0~31。

a.在按下OK鍵後，若該列於游標焦點位置右方有任何元件(不包含HorConnect及VerConnect)，則會出現一個詢問使用者是否要覆蓋的對話框。

b.不論是否出現詢問使用者是否要覆蓋的對話框，在確定後，Coil元件會自動被放置於該列的最右邊位置，從原先游標焦點的位置到最右邊的位置間會自動填入HorConnect元件。

圖 五三 線圈型式選擇

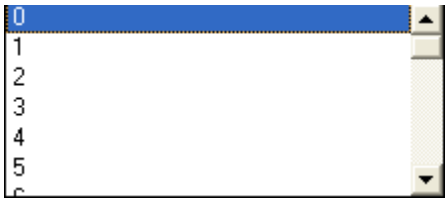


圖 五四 R-bit 選擇

名稱		說明
常開線圈	Normal Open	當True時線圈通電
常閉線圈	Normal Close	當False時線圈通電

表格 十二 PLC 線圈功能方塊說明

6 PLC 接點動作說明

6.1 常開接點 -| |-

輸出信號端 B 的狀態為輸入信號端 A 的狀態 AND 接點 I0 的狀態。接點形式可有計時器與計數器的狀態接點可供選擇，當計時或計數到達設定值時接點便會作動如圖 五十五。

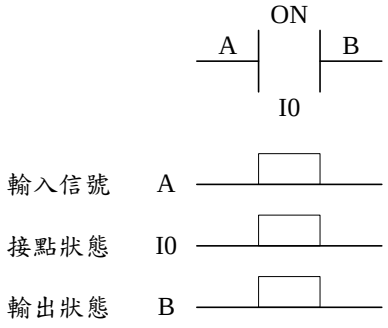


圖 五十五 常開接點

6.2 常閉接點 -|/|-

輸出信號端 B 的狀態為輸入信號端 A 的狀態 OR 接點 I0 的狀態。接點形式可有計時器與計數器的狀態接點可供選擇，當計時或計數到達設定值時接點便會作動如圖 五十六。

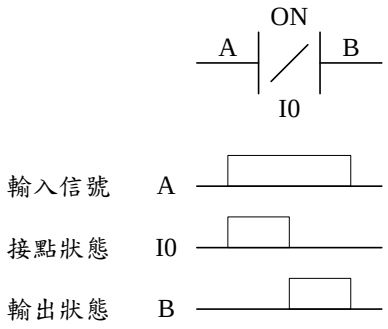


圖 五十六 常閉接點

6.3 常開線圈 ( ) (Normal Open Coil)

當 I001 與 I002 ON 之後，在 ON 之同時，輸出 RELAY O005 信號為 ON 如圖 五十七。

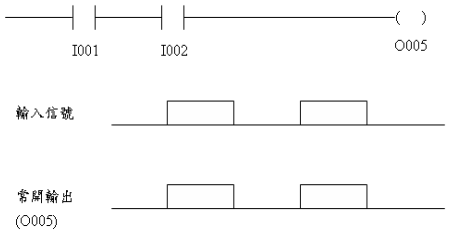


圖 五十七 常開線圈

6.4 常閉線圈（ / ） (Normal Closed Coil)

當 I001 與 I002 ON 之後，在 ON 之同時，輸出 RELAY O005 信號為 OFF 如圖 五八。

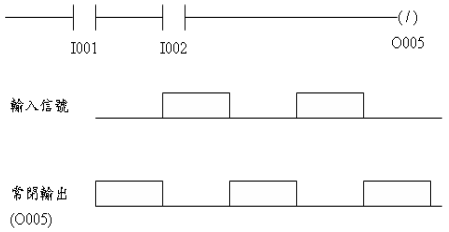


圖 五八 常閉線圈

6.5 正緣觸發脈衝線圈（ ↑ ） (Positive Edge Triggered Coil)

a. 當 I001 與 I002 ON 之後，在 ON 之同時，輸出 RELAY O005 輸出一個脈沖信號。
b. 一個脈衝信號之週期時間，代表 PC 之軟體程式執行一個循環所需之掃描時間。

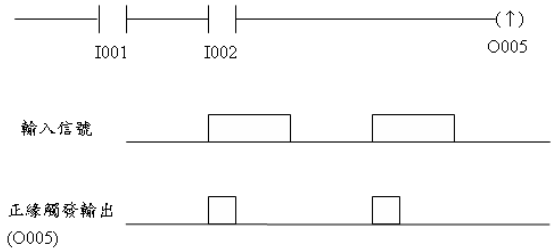


圖 五十九 正緣觸發脈衝線圈

6.6 負緣觸發脈衝線圈（ ↓ ） (Negative Edge Triggered Coil)

a. 當 I001 與 I002 OFF 之後，在 OFF 之同時，輸出 RELAY O005 輸出一個脈衝信號。

b. 一個脈衝信號之周期時間，代表 PC 之軟體程式執行一個循環所需之掃描時間。

※ 備註：正緣觸發及負緣觸發所能使用的 RELAY 數目，共256個，當超過256個時，指定無效。

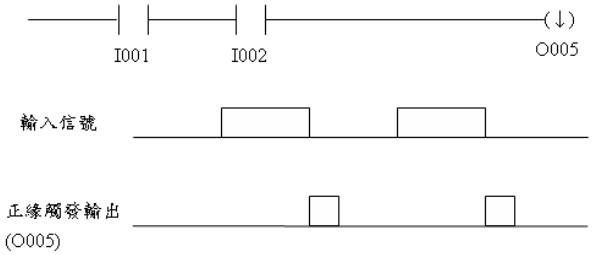


圖 六十 負緣觸發脈衝線圈

6.7 自保持線圈 (S) (Latch Coil)

a. 當 I001 與 I002 ON 之後，在 ON 之同時，RELAY O005 信號變成 ON 且一直保持 ON。

b. 輸入信號為 PULSE (1 SCAN)或不為 PULSE 均可

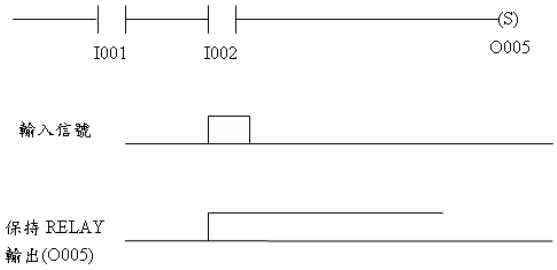


圖 六一 自保持線圈

6.8 復歸線圈 (R) (Unlatch Coil)

a. 當 I003 與 I004 ON 之後，在 ON 之同時，RELAY O005 信號由 ON 變 OFF。

b. 輸入信號為 PULSE(1 SCAN)或不為 PULSE 均可。

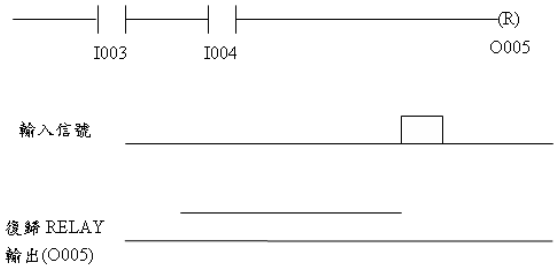


圖 六二 自保持線圈

6.9 限時繼電器 (Timer Relay)

a. 當 I001 與 I002 由 LOW 變 ON 之後，且一直 ON，1 號計時器開始以時基為 0.1 秒計時。TM*

b. 當計時器到達預設值時(1 秒)，輸出 RELAY(O005)信號為 ON。

c. 計時器的總數目為 256，每一個計時器均有一個相對的接點信號，當計時器到達預設值時每一個計時器相對的接點為 ON，每一個計時器相對的接點可以運用作 A、B 接點，使用次數無限。

d. 當輸入信號為 LOW(即當 I001 與 I002 由 ON 變 LOW 之後)計時器停止計時。

e. 計時器的時基為 0.01 秒，0.1 秒，1 秒。

f. 計時器的預設值可以來自暫存器資料。

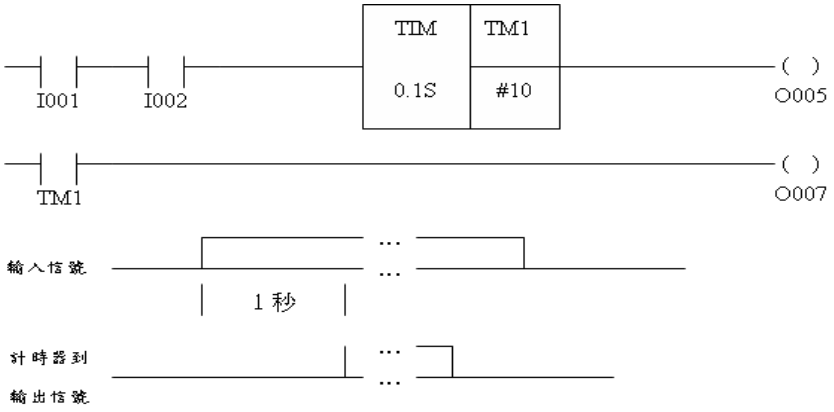


圖 六三 限時繼電器

6.10 向上計數器 (Up_Counter)

a. 當 I001 與 I002 由 LOW 變 ON 之後，在 ON 之同時，1 號計數器向上計數一次，只要輸入信號由 LOW 變 ON 則 1 號計數器向上計數一次，直到計數器的值為 10 才停止計數。

b. 當計數器到達預設值時，輸出 RELAY(O005)信號為 ON。

c. 計數器的總數目為 256，每一個計數器均有一個相對的接點信號

d. 當計數器到達預設值時每一個計數器相對的接點為 ON CT※※※，每一個計數器相對的接點可以運用作 A、B 接點，使用次數無限。

e. 輸入信號為 PULSE (1 SCAN)或不為 PULSE 均可。

f. 計數器的預設值可以來自暫存器資料。

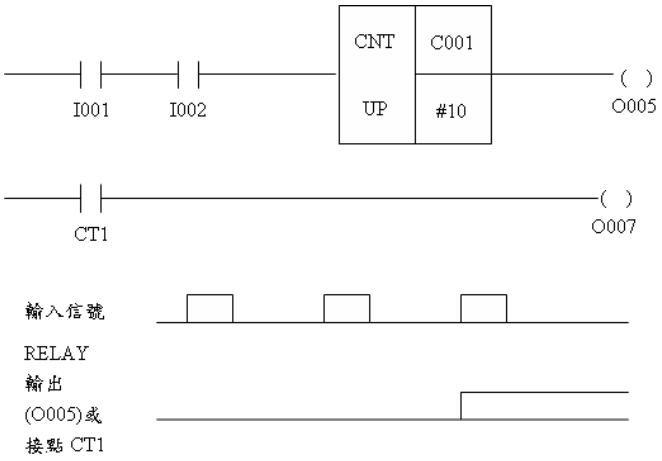


圖 六四 向上計數器

6.11 向下計數器 (Down_Counter)

a. 當 I001 與 I002 由 LOW 變 ON 之後，在 ON 之同時，1 號計數器向下計數一次，只要輸入信號由 LOW 變 ON 則 1 號計數器向下計數一次，直到計數器的值為 0 才停止計數。

b. 當計數器到達預設值時每一個計數器相對的接點為 ON CT※※※，每一個計數器相對的接點可以運用作 A、B 接點，使用次數無限。

c. 輸入信號為 PULSE(1 SCAN)或不為 PULSE 均可。

d. 計數器的預設值可以來自暫存器資料。

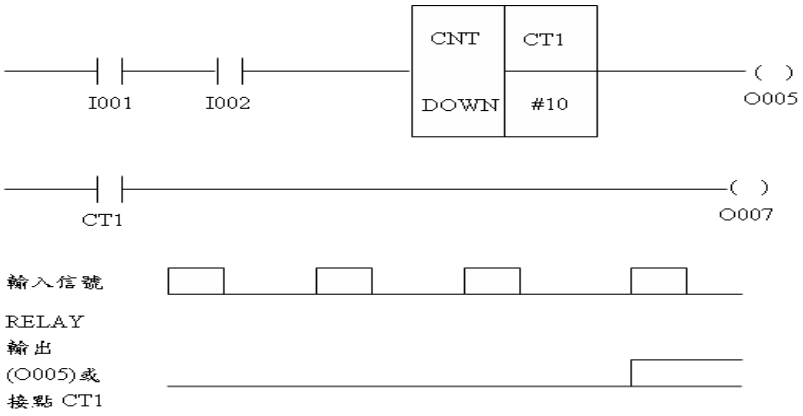


圖 六五 向下計數器

6.12 向上循環計數器 (Ring Up_Counter)

a. 當 I001 與 I002 由 LOW 變 ON 之後，在 ON 之同時，1 號計數器向上計數一次，只要輸入

信號由 LOW 變 ON 則 1 號計數器向上計數一次，當計數器的值為 10 則重新由 0 開始計數。

b. 當計數器到達預設值時，輸出 RELAY(O005)信號為 ON。

c. 計數器的總數目為 256，每一個計數器均有一個相對的接點信號，當計數器到達預設值時每一個計數器相對的接點為 ON，每一個計數器相對的接點可以運用作 A、B 接點，使用次數無限。

d. 輸入信號為 PULSE(1 SCAN)或不為 PULSE 均可。

e. 計數器的預設值可以來自暫存器資料。

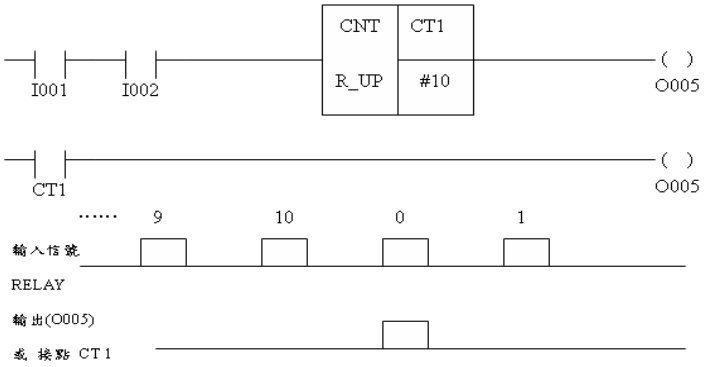


圖 六六 向上循環計數器

6.13 向下循環計數器 (Ring Down_Counter)

a. 當 I001 與 I002 由 LOW 變 ON 之後，在 ON 之同時，1 號計數器向上計數一次，只要輸入信號由 LOW 變 ON 則 1 號計數器向上計數一次，直到計數器的值為 0 則重新由 10 開始計數。

b. 當計數器到達預設值時，輸出 RELAY(O005)信號為 ON。

c. 計數器的總數目為 256，每一個計數器均有一個相對的接點信號，當計數器到達預設值時每一個計數器相對的接點為 ON，每一個計數器相對的接點可以運用作 A、B 接點，使用次數無限。

d. 輸入信號為 PULSE(1 SCAN)或不為 PULSE 均可。

e. 計數器的預設值可以來自暫存器資料。

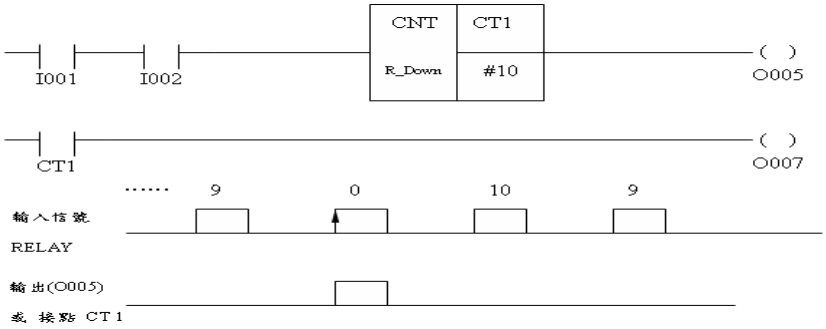


圖 六七 向下循環計數器

6.14 計數器復歸 (Reset Coounter)

a. 當 I001 與 I002 由 LOW 變 ON 之後，在 ON 之同時，將 1 號計數器復歸。
b. 當要復歸向上計數器時，其復歸值為 0。
c. 當要復歸向下計數器時，其復歸值為預設值。
d. 當復歸計數器時，輸出 RELAY(O005)信號變為 HIGH，而相對的接點信號，變為 LOW。

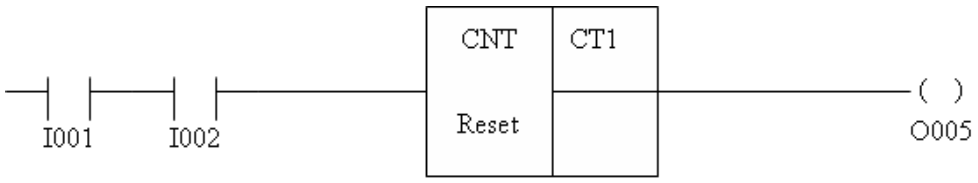


圖 六十八 計數器復歸

6.15 加算指令 (Add)

a. 當 I001 與 I002 由 LOW 變 ON 之後，執行加法運算，被加數的位置在第 R002 號資料暫存器，加數的位置在第 R001 號資料暫存器，運算結果的位置在第 R002 號資料暫存器，每一個資料暫存器內的值均可為長整數形式，每一個數均為 4 Bytes。(-2147483648<=X<=+2147483647)
b. 運算式：R002<-R002+R001。
c. 加數的值可以直接為長整數值。
d. 當輸入為 LOW 時輸出繼電器(O005)為 LOW。當輸入為 ON 時輸出也為 ON。
e. 當輸入為 ON 時 ADD 指令在 PC 內部每次掃描一次時，均被執行一次。

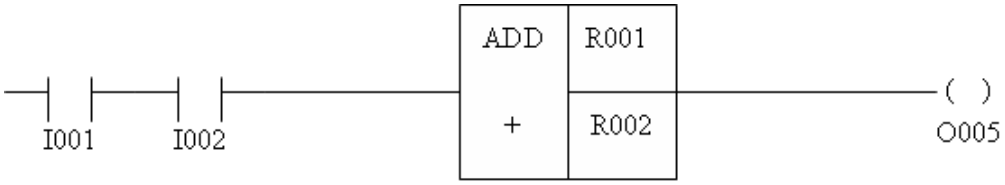


圖 六九 加算指令

6.16 減算指令 (Subtract)

a. 當 I001 與 I002 由 LOW 變 ON 之後，執行減法運算，被減數的位置在第 R002 號資料暫存器，減數的位置在第 R001 號資料暫存器，運算結果的位置在第 R002 號資料暫存器，每一

個資料暫存器內的值均可為整數形式，每一個數均為 4 Bytes。

b. 運算式：R002<-R002-R001

c. 減數的值可以直接為長整數值。

d. 當輸入為 LOW 時輸出繼電器(O005)為 LOW。當輸入為 ON 時輸出也為 ON。

e. 當輸入為 ON 時 SUB 指令在 PC 內部每次掃描一次時，均被執行一次。

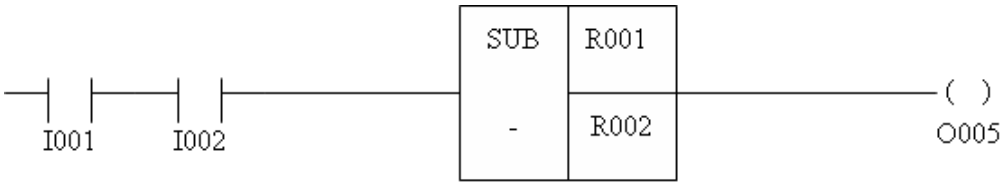


圖 七〇 減算指令

6.17 乘法指令 (Multiply)

a. 當 I001 與 I002 由 LOW 變 ON 之後，執行乘法運算，被乘數的位置在第 R002 號資料暫存器，乘數的位置在第 R001 號資料暫存器，運算結果的位置在第 R002 號資料暫存器，每一個資料暫存器內的值均可為整數形式，每一個數均為 4 Bytes。

b. 運算式：R002<-R002xR001

c. 乘數的值可以直接為長整數值。

d. 當輸入為 LOW 時輸出繼電器(O005)為 LOW。當輸入為 ON 時輸出也為 ON。

e. 當輸入為 ON 時 MUL 指令在 PC 內部每次掃描一次時，均被執行一次。

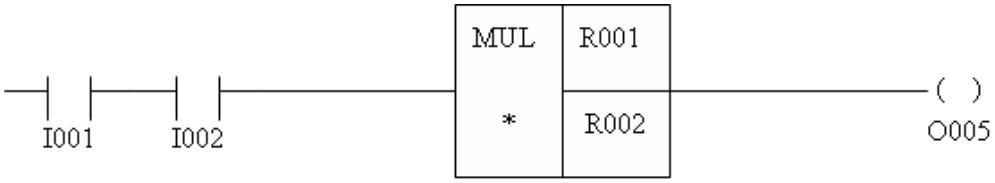


圖 七一 乘法指令

6.18 除法指令 (Divide)

a. 當 I001 與 I002 由 LOW 變 ON 之後，執行除法運算，被除數的位置在第 R002 號資料暫存器，除數的位置在第 R001 號資料暫存器，運算結果的位置在第 R002 號資料暫存器，每一個資料暫存器內的值均可為整數形式，每一個數均為 4 Bytes。

b. 運算式：R002<-R001/R002

c. 除數的值可以直接為長整數值。

d. 當除數為零或輸入為 LOW 時輸出繼電器(O005)為 LOW。當輸入為 ON 時輸出也為 ON。

e. 當輸入為 ON 時 DIV 指令在 PC 內部每次掃描一次時，均被執行一次。

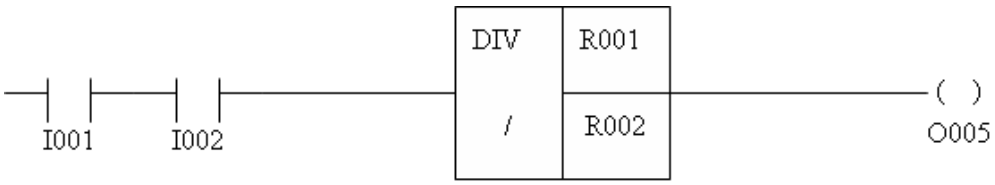


圖 七二 除法指令

6.19 邏輯 AND 指令 (And)

a. 當 I001 與 I002 由 LOW 變 ON 之後，執行邏輯 AND 運算，被 AND 的邏輯狀態的位置在第 R002 號資料暫存器，要 AND 的邏輯狀態的位置在第 R001 號資料暫存器，運算結果的邏輯狀態位置在第 R002 號資料暫存器，每一個資料暫存器內的值均為 32 Bits 的形式，每一個暫存器均為 32 Bits 的資料。

b. 運算式：R002<-R002&R001

c. &：位元與位元的 AND 運算。

d. 要 AND 的數可以直接為常數值-2147483648<=X<=+2147483647。

e. 當輸入為 LOW 時輸出繼電器(O005)為 LOW。當輸入為 ON 時輸出也為 ON。

f. 當輸入為 ON 時 AND 指令在 PC 內部每次掃描一次時，均被執行一次。

g. AND 邏輯表如圖 七三

X	Y	Z
1	1	1
1	0	0
0	1	0
0	0	0

圖 七三 AND 邏輯表

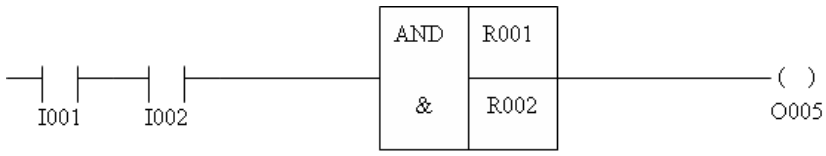


圖 七十四 邏輯 AND 指令

6.20 邏輯 OR 指令 (Or)

a. 當 I001 與 I002 由 LOW 變 ON 之後，執行邏輯 OR 運算，被 OR 的邏輯狀態的位置在第 R002 號資料暫存器，要 OR 的邏輯狀態的位置在第 R001 號資料暫存器，運算結果的邏輯狀態位置在第 R002 號資料暫存器，每一個資料暫存器內的值均為 32 Bits 的形式，每一個暫存器均為 32 Bits 的資料。

b. 運算式：R002<-R002|R001

c. |：位元與位元的 OR 運算。

d. 要 OR 的數可以直接為常數值-2147483648<=X<=+2147483647。

e. 當輸入為 LOW 時輸出繼電器(O005)為 LOW。當輸入為 ON 時輸出也為 ON。

f. 當輸入為 ON 時 OR 指令在 PC 內部每次掃描一次時，均被執行一次。

g. OR 邏輯表如圖 七五

X	Y	Z
1	1	1
1	0	1
0	1	1
0	0	0

圖 七五 OR 邏輯表

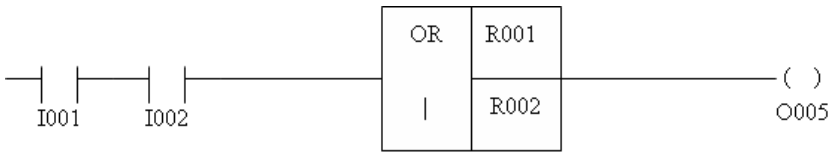


圖 七六 邏輯 OR 指令

6.21 邏輯 XOR 指令 (Exclusive Or)

a. 當 I001 與 I002 由 LOW 變 ON 之後，執行邏輯 XOR 運算，被 XOR 的邏輯狀態的位置在第 R002 號資料暫存器，要 XOR 的邏輯狀態的位置在第 R001 號資料暫存器，運算結果的邏輯狀態位置在第 R002 號資料暫存器，每一個資料暫存器內的值均為 32 Bits 的形式，每一個暫存器均為 32 Bits 的資料。

b. 運算式：R002<-R002^R001

c. ^：位元與位元的 XOR 運算。

d. 要 XOR 的數可以直接為常數值-2147483648<=X<=+2147483647。

e. 當輸入為 LOW 時輸出繼電器(O005)為 LOW。當輸入為 ON 時輸出也為 ON。

f. 當輸入為 ON 時 XOR 指令在 PC 內部每次掃描一次時，均被執行一次。

g. XOR 邏輯表 如圖 七七

X	Y	Z
1	1	0
1	0	1
0	1	1
0	0	0

圖 七七 XOR 邏輯表

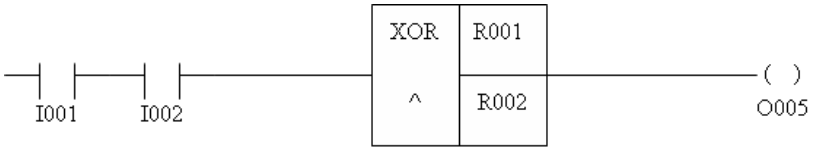


圖 七十八 邏輯 XOR 指令

6.22 比較 CMP 指令(大於) (Compare，Larger Than)

a. 當 I001 與 I002 由 LOW 變 ON 之後，執行比較 CMP 運算，被比較的數值的位置在第 R002 號資料暫存器，要比較的數值的位置在第 R001 號資料暫存器，比較結果為真時輸出繼電器為 HIGH。

b. 運算式：TRUE<-R002>R001

c. 要比較的數可以直接為十進位值。

d. 當比較結果為假時或輸入為 LOW 時輸出繼電器(O005)為 LOW。

e. 當輸入為 ON 時 CMP 指令在 PC 內部每次掃描一次皆執行。

f. 比較 CMP 指令(小於) (Compare，Smaller Than)

g. 當 I001 與 I002 由 LOW 變 ON 之後，執行比較 CMP 運算，被比較的數值的位置在第 R002 號資料暫存器，要比較的數值的位置在第 R001 號資料暫存器，比較結果為真時輸出繼電器為 HIGH。

h. 運算式：TRUE<-R002<R001

i. 要比較的數可以直接為十進位值。

j. 當比較結果為假時或輸入為 LOW 時輸出繼電器(O005)為 LOW。

k. 當輸入為 ON 時 CMP 指令在 PC 內部每次掃描一次時，均被執行一次。

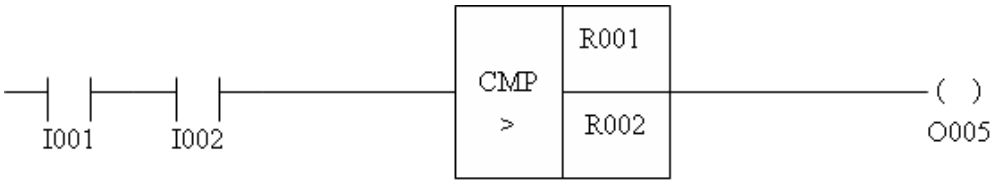


圖 七十九 比較 CMP 指令

6.23 比較 CMP 指令(小於) (Compare，Smaller Than)

a. 當 I001 與 I002 由 LOW 變 ON 之後，執行比較 CMP 運算，被比較的數值的位置在第 R002 號資料暫存器，要比較的數值的位置在第 R001 號資料暫存器，比較結果為真時輸出繼電器為 HIGH。
b. 運算式：TRUE<-R002<R001
c. 要比較的數可以直接為十進位值。
d. 當比較結果為假時或輸入為 LOW 時輸出繼電器(O005)為 LOW。
e. 當輸入為 ON 時 CMP 指令在 PC 內部每次掃描一次時，均被執行一次。

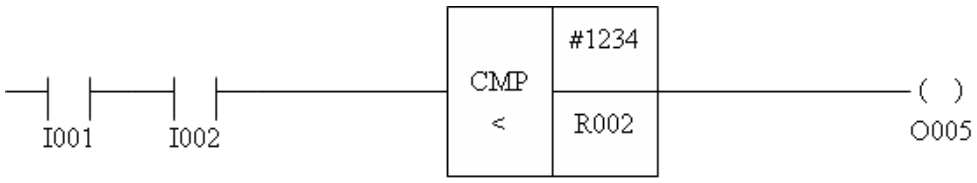


圖 八十 比較 CMP 指令

6.24 比較 CMP 指令(等於) (Compare，Equal To)

a. 當I001與I002由LOW變ON之後，執行比較CMP運算，被比較的數值的位置在第R002號資料暫存器，要比較的數值的位置在第R001號資料暫存器，比較結果為真時輸出繼電器為HIGH。
b. 運算式：TRUE<-R002=R001
c. 要比較的數可以直接為十進位值。
d. 當比較結果為假時或輸入為LOW時輸出繼電器(O005)為LOW。
e. 當輸入為ON時CMP指令在PC內部每次掃描一次時，均被執行一次。

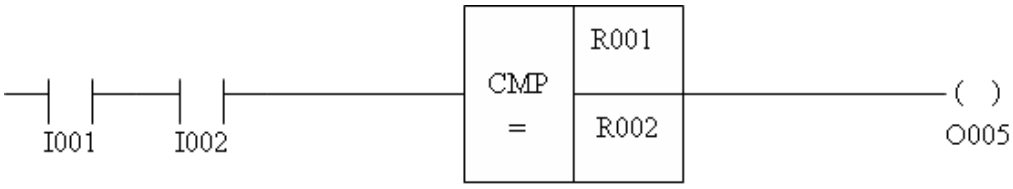


圖 八一 比較 CMP 指令

6.25 資料搬移 MOVE 指令 (Move)

a. 當 I001 與 I002 由 LOW 變 ON 之後，執行搬移 MOV 指令，被搬移的數值的位置在第 R001 號資料暫存器，要置放的位置在第 R002 號資料暫存器，搬移結束時輸出繼電器(O005)為 HIGH。
b. 運算式：R002=R001
c. 被搬移的數可以直接為長整數值。
d. 輸入為 LOW 時輸出繼電器(O005)為 LOW。
e. 當 I001 與 I002 由 LOW 變 ON 之後，執行搬移 MOV 指令，被搬移的數值為十進位數值 1234，要置放的位置在第 R002 號資料暫存器，搬移結束時輸出繼電器(O005)為 HIGH。
f. 當輸入為 ON 時 MOV 指令在 PC 內部每次掃描一次時，均被執行一次。

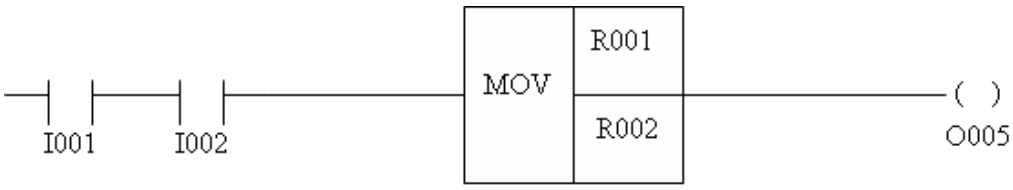


圖 八十二 資料搬移 MOVE 指令

6.26 資料搬移 MOVE 指令 (由 Register Move to Register pointer)

a. 當 I001 與 I002 由 LOW 變 ON 之後，執行搬移 MOV 指令，若由資料暫存器搬移至資料暫存器指標，被搬移的數值的位置在第 R001 號資料暫存器，要置放的位置在第 R002 號資料暫存器內指定的資料暫存器，如 R002 內存之值為 3，搬移結束時輸出繼電器(O005)為 HIGH，R003=R001。
b. 運算式：*R002=R001
c. 被搬移的數可以直接為長整數值。
d. 輸入為 LOW 時輸出繼電器(O005)為 LOW。
e. 當 I001 與 I002 由 LOW 變 ON 之後，執行搬移 MOV 指令，被搬移的值位於 R001 號資料暫存器，要置放的位置在第 R002 號資料暫存器所指定的資料暫存器，搬移結束時輸出繼電

器(O005)為 HIGH。

f. 當輸入為 ON 時 MOV 指令在 PC 內部每次掃描一次時，均被執行一次。

g. 暫存器指標所指向的暫存器若超過系統允許最大值(目前為 R186449)，此指令不會被執行並輸出 LOW。

圖 八十三 資料搬移 MOVE 指令(由暫存器至暫存器指標)

6.27 資料搬移 MOVE 指令 (由 Register pointer Move to Register)

a. 當 I001 與 I002 由 LOW 變 ON 之後，執行搬移 MOV 指令，若由資料暫存器指標搬移至資料暫存器，被搬移的數值的位置在第 R001 號資料暫存器指定的資料暫存器，要置放的位置在第 R002 號資料暫存器，若 R001 內存之值為 4，搬移結束時輸出繼電器(O005)為 HIGH，R002=R004。

b. 運算式：R002=*R001

c. 輸入為 LOW 時輸出繼電器(O005)為 LOW。

d. 當 I001 與 I002 由 LOW 變 ON 之後，執行搬移 MOV 指令，被搬移的值位於 R001 號資料暫存器所指定的暫存器，要置放的位置在第 R002 號資料暫存器，搬移結束時輸出繼電器(O005)為 HIGH。

e. 當輸入為 ON 時 MOV 指令在 PC 內部每次掃描一次時，均被執行一次。

f. 暫存器指標所指向的暫存器若超過系統允許最大值(目前為 R186449)，此指令不會被執行並輸出 LOW。

圖 八十四 資料搬移 MOVE 指令(由暫存器指標至暫存器)

6.28 連續 R 資料暫存器設值指令 (MULRINI)

a. 當 I001 與 I002 由 LOW 變 ON 之後，執行搬移 MULRINI 指令，由 R000 開始至 R009 十個資料暫存器皆設值為 1，搬移結束時輸出繼電器(O005)為 HIGH。

b. 運算式：[R000..R009]=1

c. 被設定的數可以直接為長整數值。

d. 輸入為 LOW 時輸出繼電器(O005)為 LOW。

e. 當 I001 與 I002 由 LOW 變 ON 之後，執行搬移 MULRINI 指令，被設定的數值為十進位數值 1，要設定的資料暫存器位置為 R000 至 R009 十個，設定結束時輸出繼電器(O005)為 HIGH。

f. 當輸入為 ON 時 MULRINI 指令在 PC 內部每次掃描一次時，均被執行一次。

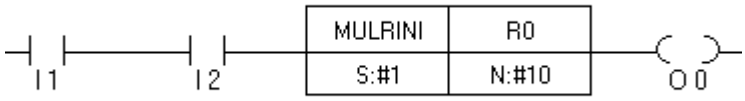


圖 八十五 MULRINI 指令

6.29 連續 R 資料暫存器拷背指令 (MULRCPY)

a. 當 I001 與 I002 由 LOW 變 ON 之後，執行搬移 MULRCPY 指令，由 R000 開始至 R009 十個資料暫存器皆設值為一對一拷背到 R020 至 R029 區域 (R000 拷至 R020，R001 拷至 R021，…，R009 拷至 R029)，拷背結束時輸出繼電器(O005)為 HIGH。

b. 運算式：[R000..R009]=[R020..R029]

c. 輸入為 LOW 時輸出繼電器(O005)為 LOW。

d. 當 I001 與 I002 由 LOW 變 ON 之後，執行搬移 MULRCPY 指令，要拷背的區域為 R000 至 R009 十個，目的為 R020 至 R029，拷背結束時輸出繼電器(O005)為 HIGH。

e. 當輸入為 ON 時 MULRCPY 指令在 PC 內部每次掃描一次時，均被執行一次。

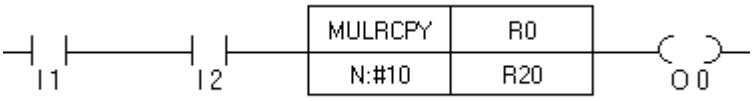


圖 八六 MULRCPY 指令

6.30 連續 I 接點映射指令(I/RMAP)

a. 當 I001 與 I002 由 LOW 變 ON 之後，執行搬移 I/RMAP 指令，由 I000 開始至 I031 32 個接點值映射至 R001，I000 值為最低位元，I031 值為最高位元，映設結束時輸出繼電器(O005)為 HIGH。

b. 運算式：R001=I031<<(31) OR (I030)<<(30) OR …OR (I000)

c. 輸入為 LOW 時輸出繼電器(O005)為 LOW。

d. 當 I001 與 I002 由 LOW 變 ON 之後，執行搬移 I/RMAP 指令，要映射的區域為 I000 至 I031 32 個，目的為 R001，映射結束時輸出繼電器(O005)為 HIGH。

e. 當輸入為 ON 時 I/RMAP 指令在 PC 內部每次掃描一次時，均被執行一次。

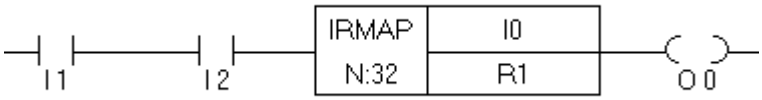


圖 八七 I/RMAP 指令

6.31 連續 O 接點映射指令(O/RMAP)

a. 當 I001 與 I002 由 LOW 變 ON 之後，執行搬移 I/RMAP 指令，由 O000 開始至 O031 32 個接點值映射至 R001，O000 值為最低位元，O031 值為最高位元，映設結束時輸出繼電器(O005)為 HIGH。
b. 運算式：R001=O031<<(31) OR (O030)<<(30) OR …OR (O000)
c. 輸入為 LOW 時輸出繼電器(O005)為 LOW。
d. 當 I001 與 I002 由 LOW 變 ON 之後，執行搬移 O/RMAP 指令，要映射的區域為 O000 至 O031 32 個，目的為 R001，映射結束時輸出繼電器(O005)為 HIGH。
e. 當輸入為 ON 時 O/RMAP 指令在 PC 內部每次掃描一次時，均被執行一次。

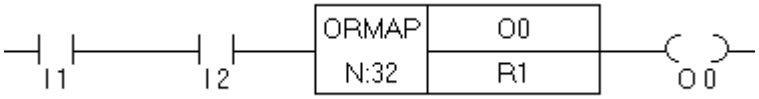


圖 八八 O/RMAP 指令

6.32 程式跳躍 JMP 指令 (Jump)

a. JMP 指令可以在 Ladder 的執行 Level 1 中呼叫，並且跳躍的標記必須與 JMP 指令在同一個 Level 中否則語法檢查會不通過。
b. 當 I1 與 I2 由 OFF 變 ON 之後，執行程式跳躍指令，程式跳到符號標記為 EMG 的地方開始執行。
c. 若 I1 與 I2 信號為 OFF，則 JMP 指令不執行。

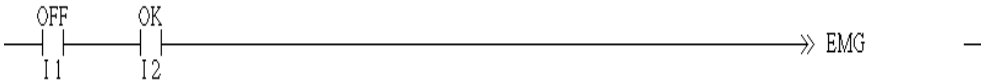


圖 八十九 程式跳躍 JMP 指令

6.33 階梯圖副程式呼叫 JSR 指令 (Jump Subroutine)

a. 副程式只可以在階梯圖的副程式呼叫區中設計，故 JSR 的符號標記只可以跳到副程式區中，若跳到非副程式符號標記則於程式檢查時會發生錯誤。

b. 當 I1 與 I2 由 OFF 變 ON 之後，執行階梯圖程式呼叫指令，程式跳到符號標記為 STOP 的地方執行，執行結束則返回執行下一個元件指令。

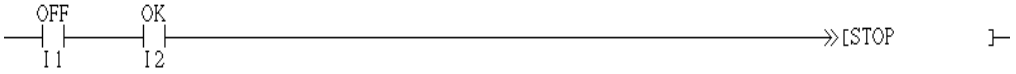


圖 九十 階梯圖副程式呼叫 JSR 指令

6.34 階梯圖副程式呼叫結束 Return 指令 (Return Subroutine)

a. Return 指令只可以搭配副程式 Label 成對使用，若無成對使用或巢狀副程式將會在程式檢查時發生錯誤。

b. 此為階梯圖副程式結束指令。

圖 九十一 階梯圖副程式呼叫結束 Return 指令

6.35 符號標記宣告 (Label)

a. 符號標記宣告******以被 JMP 或 JSR 使用。

b. 符號標記可輸入 6 個文字符號

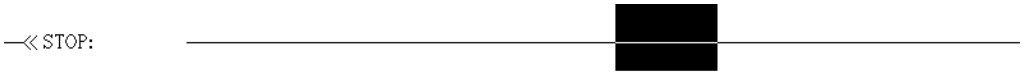


圖 九十二 符號標記宣告

6.36 階梯圖編輯結束 END 指令 (End)

a. 此為階梯圖區塊編輯結束指令，用以區分階梯圖的 Level 1 / 副程式區的區塊間隔。

圖 九十三 階梯圖編輯結束 END 指令

7 PLC Programmer 搜尋操作說明

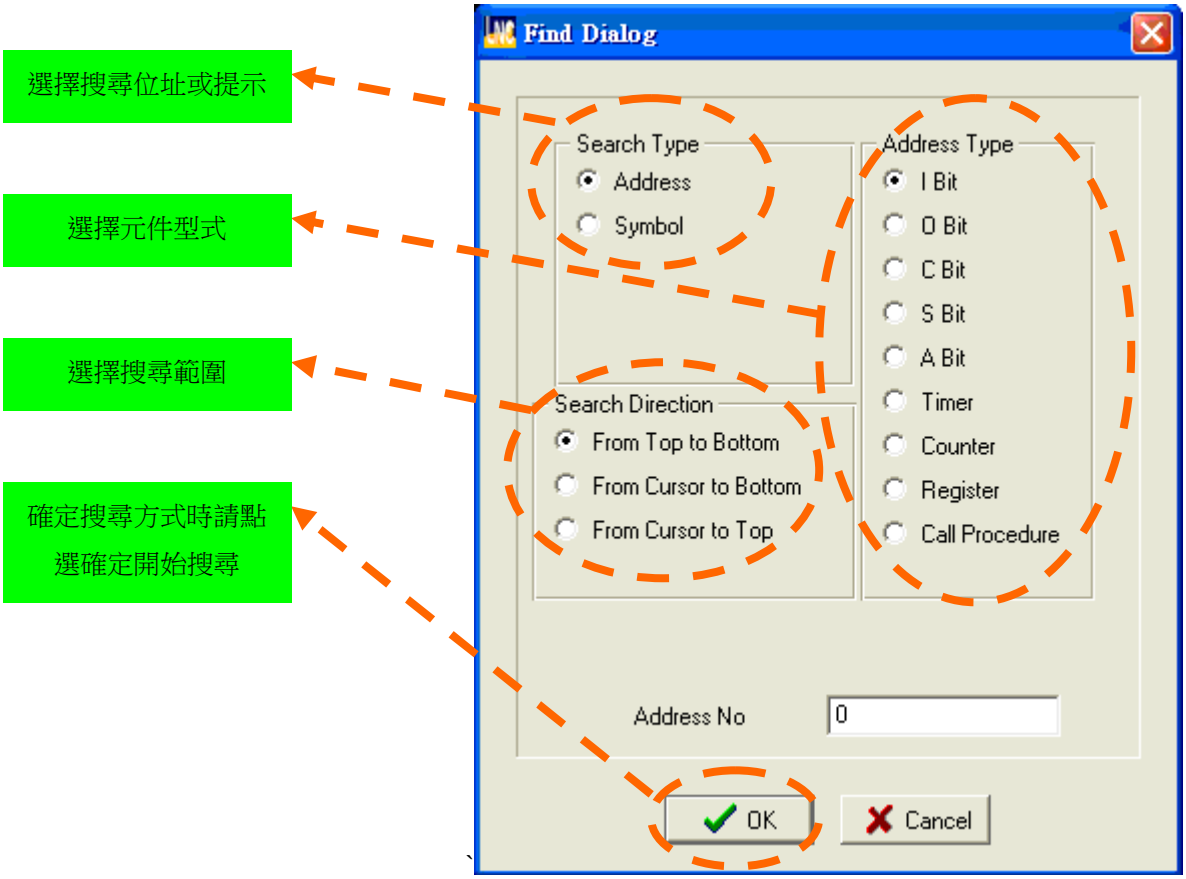


圖 九四 搜尋視窗

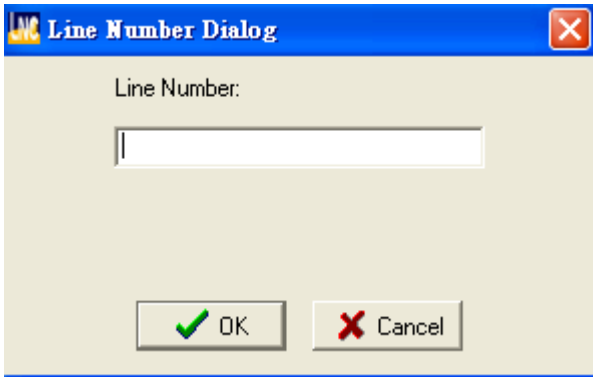


圖 九五 列號搜尋視窗

※ 搜尋：開啟搜尋視窗使用搜尋功能 圖 九四(Ctrl+F)。

※ 尋找下一個：搜尋下一個相同設定的元件或按快速鍵F3。

※ 到指定行：輸入列號直接跳至該列 圖 九五(Ctrl+G)。

8 滑鼠按鍵功能說明

滑鼠左鍵功能：

※ 選取功能按住滑鼠的右鍵進行拖曳進行選取的動作。

滑鼠右鍵功能：

※ 編輯工具列

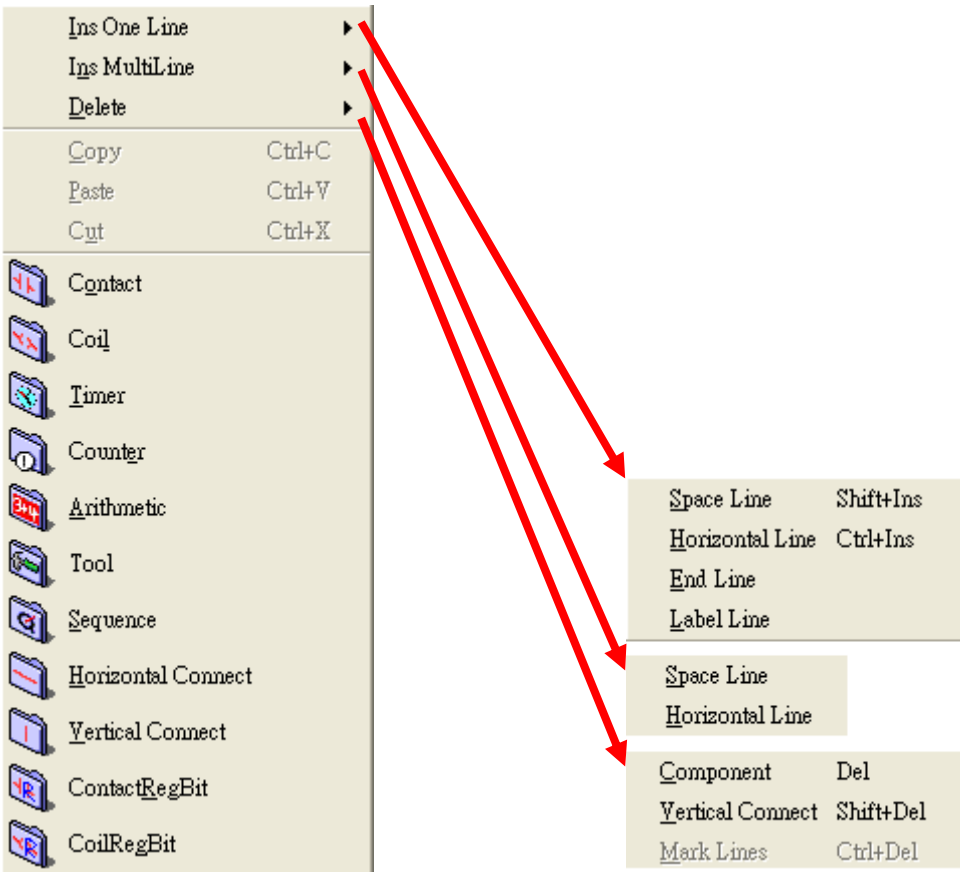


圖 九六 滑鼠功能介紹

9 上傳、下載、重新載入使用說明

PLC Programmer 上傳/下載等連線功能需搭配控制器 ReCON Server 版本，若連線失敗先確認 PLC Programmer 及控制器 ReCON Server 版次配對。

a. 選擇 Transfer 中的 Connect 如圖 九七

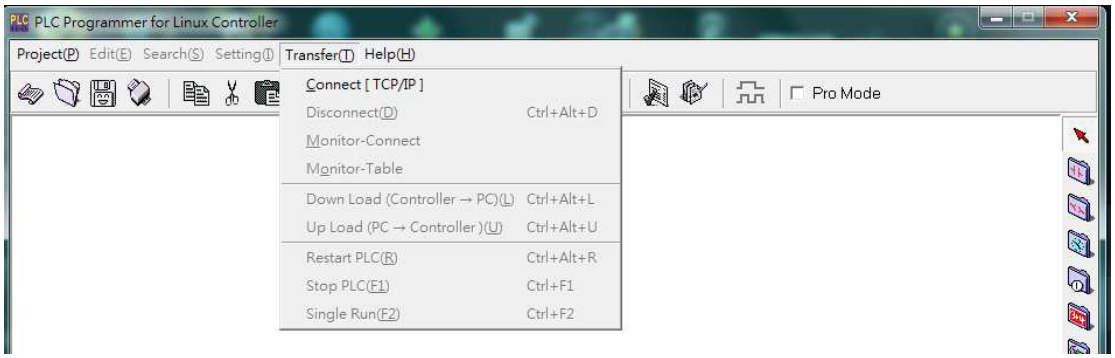


圖 九七 與 Controller 連線

b. 跳出輸入 IP address 的圖示，並輸入想連接的 Controller 之 IP 如圖 九八

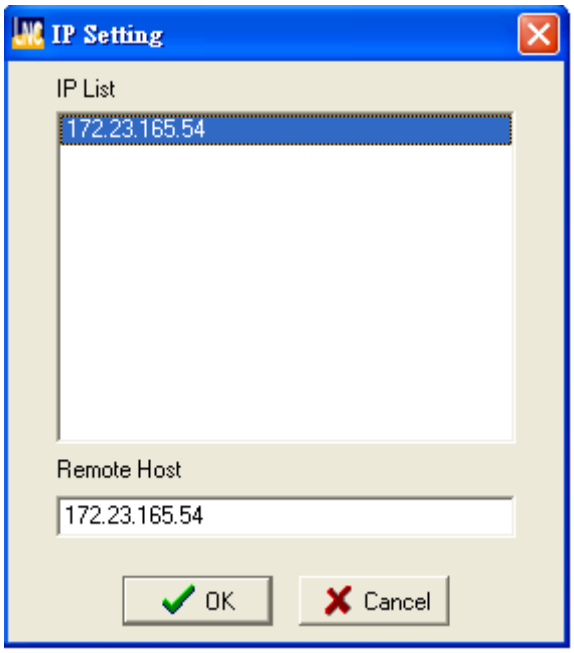


圖 九八 IP 輸入畫面

c. 連線成功畫面如圖 九九

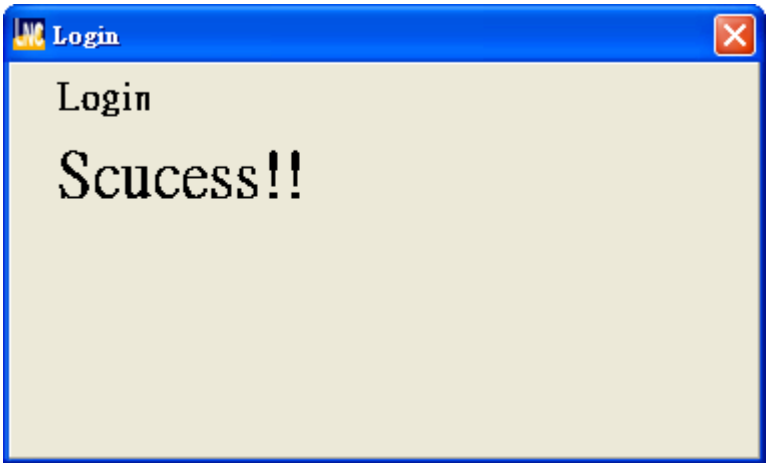


圖 九九 連線成功

d. 連線失敗畫面如圖 一〇〇

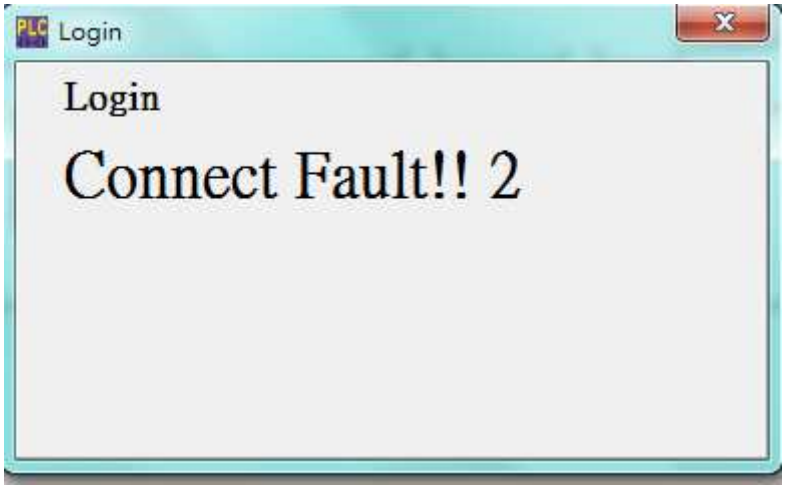


圖 一〇〇 連線失敗

e. 下載選擇如圖 一〇一

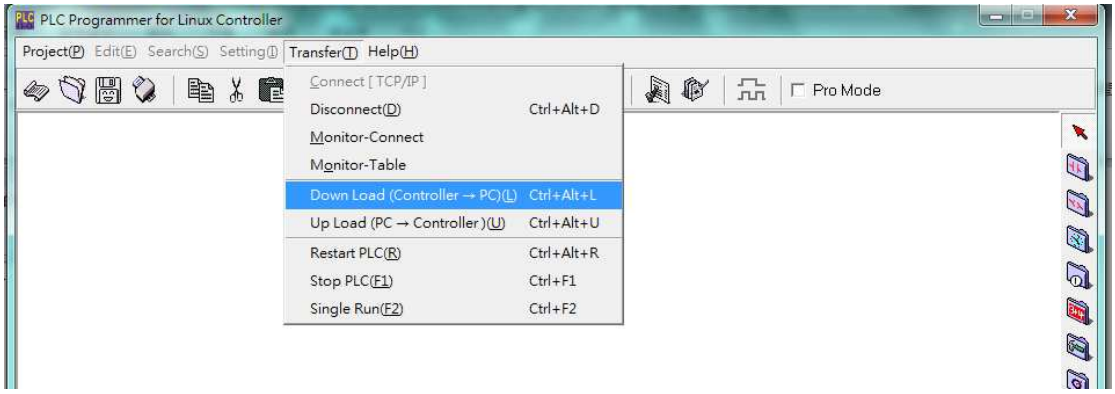


圖 一〇二 下載選擇

f. 上傳選擇如

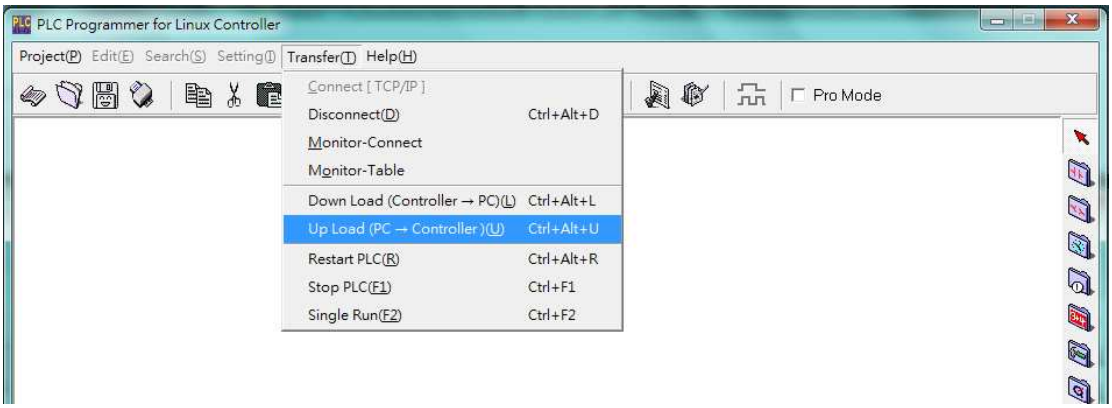


圖 一〇三 上傳選擇

g. 上傳或下載時，詢問是否確定覆蓋目標目錄內已存在之檔案畫面如

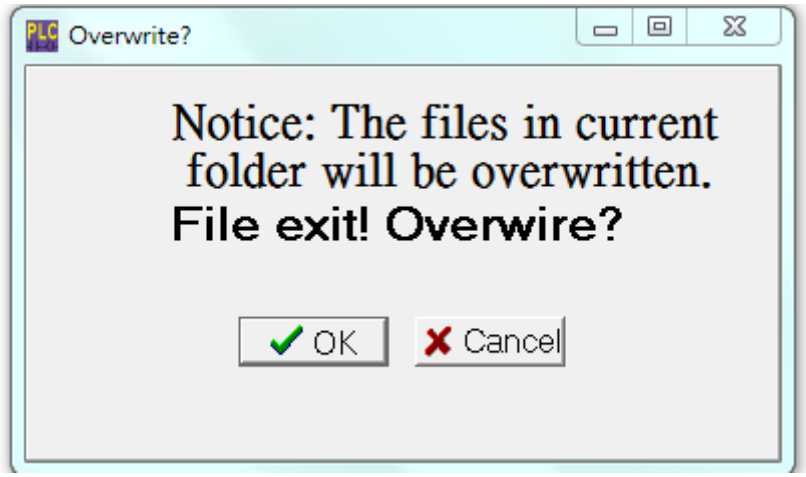


圖 一〇四 檔案存在是否覆蓋畫面

Ps.不論上傳與下載都是會則工作資料夾內的 cnc*.*檔案作處理

h. Monitor-Connect，以類即時顯示目前控制器上 PLC 梯型圖運行狀態。

Ps.PC 及控制器內梯型圖必需先同步成同一份。

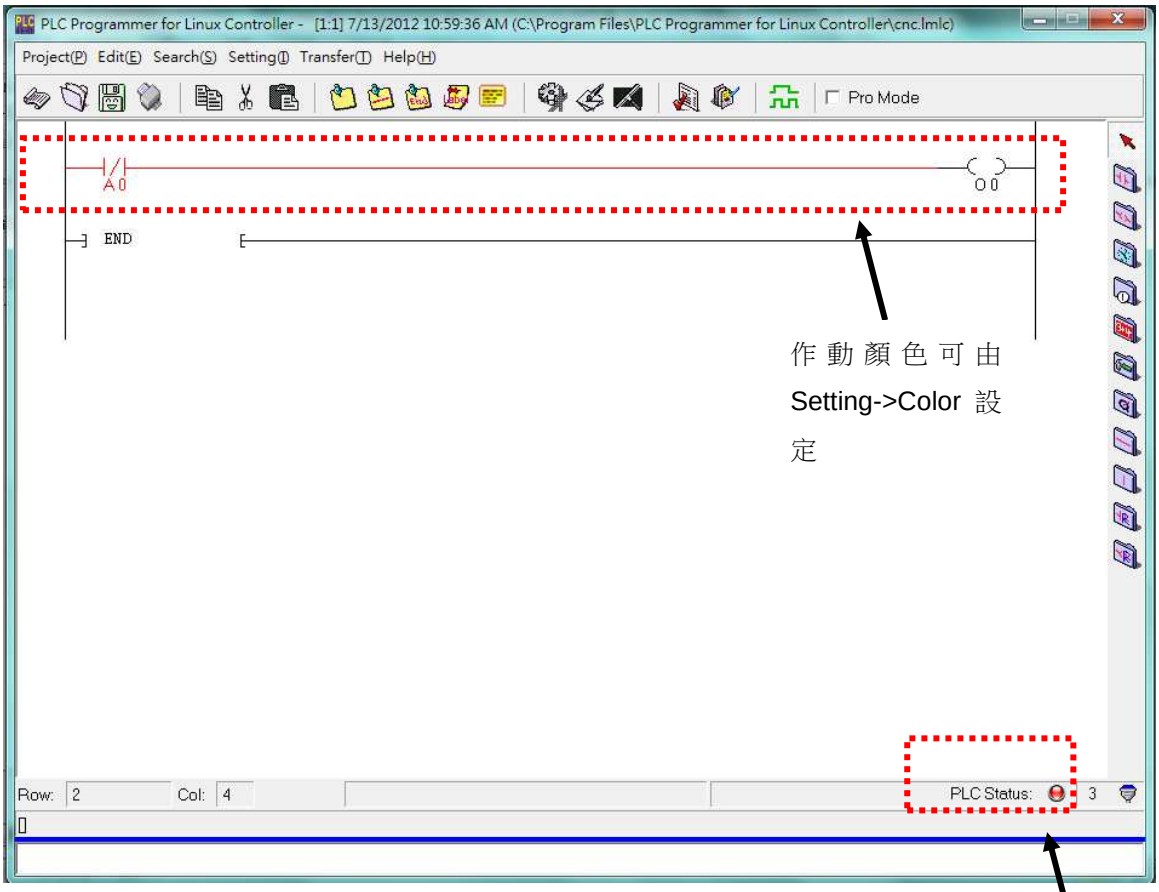


圖 一〇五

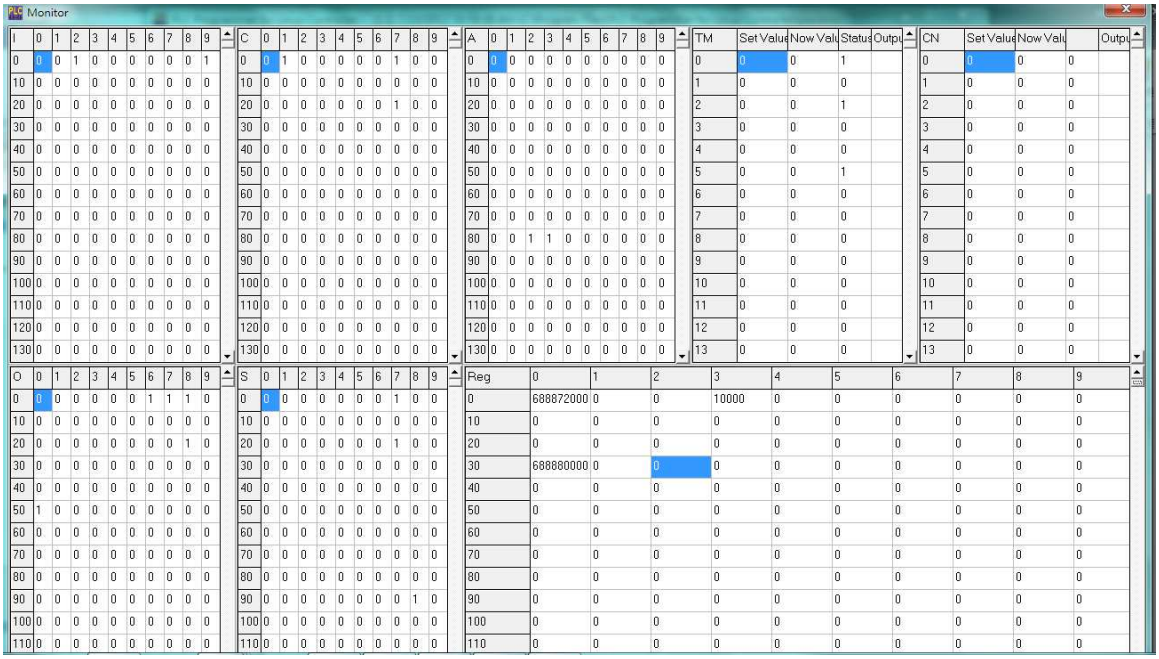
控制器 PLC 運作狀態

: 執行中

: 停止

: 等待

i. Monitor-Table，類即時顯示控制器內 IOCSAR。



j. PLC 重新啟動

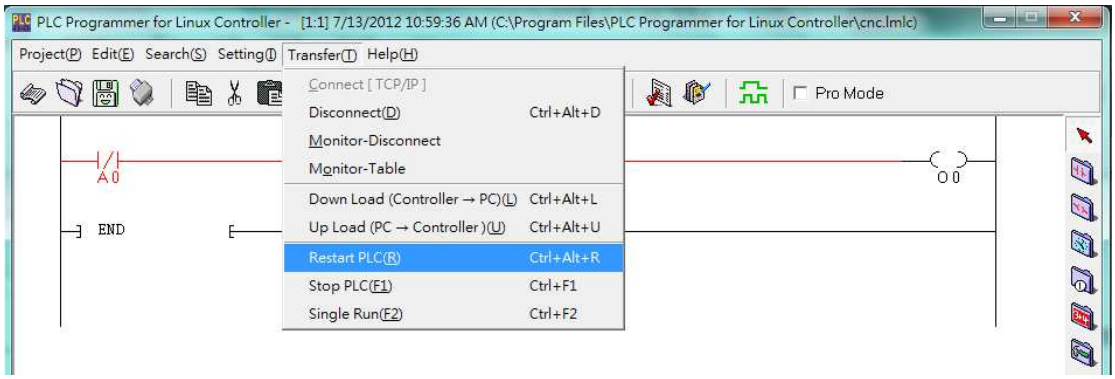


圖 一〇六 PLC 重新啟動功能

k. 重新載入成功

圖 一〇七 PLC 重新啟動成功

l. 重新載入失敗如

圖 一〇八 PLC 重新啟動失敗

Ps.重新載入失敗可能是因為 ReCON 功能沒有開啟所造成，或者使用權限為唯讀。

m. PLC Stop，停止 PLC 執行

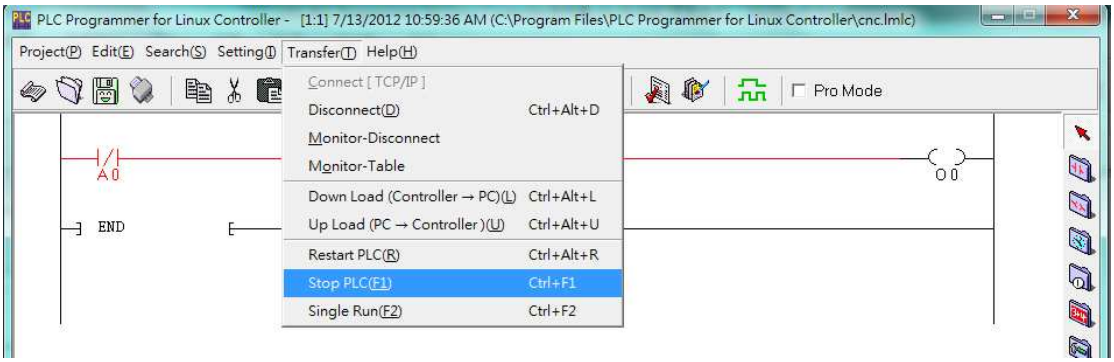


圖 一〇九

n. Single Run，PLC 執行一個 cycle 後停止。

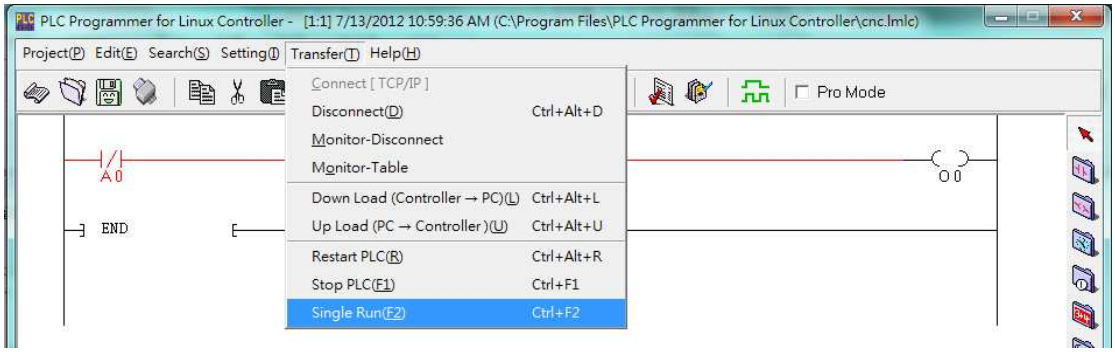

PLC Programmer for Linux Controller - [1:1] 7/13/2012 10:59:36 AM (C:\Program Files\PLC Programmer for Linux Controller\cnc.lmlc)
Project(P) Edit(E) Search(S) Setting(I) Transfer(T) Help(H)
Connect [TCP/IP]
Disconnect(D) Ctrl+Alt+D
Monitor-Disconnect
Monitor-Table
Down Load (Controller → PC)(L) Ctrl+Alt+L
Up Load (PC → Controller)(U) Ctrl+Alt+U
Restart PLC(R) Ctrl+Alt+R
Stop PLC(F1) Ctrl+F1
Single Run(F2) Ctrl+F2
Pro Mode
A 0
END
O 0

圖 一一○

10 PLC 警報/警告字串編輯

目前針對每個 Register 的 bit 都可以寫註解，如 R0.0=”人機 Reset”

呼叫出註解檔方式有兩個：

一. 選擇工具列 Setting→Symbol 如圖 一一一

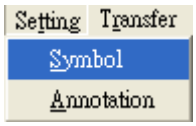


圖 一一一 呼叫註解檔

二. 按下快速鍵最後一個 Symbol 如圖 一一二

圖 一一二 呼叫註解檔

畫面會出現註解編輯畫面如圖 一一三

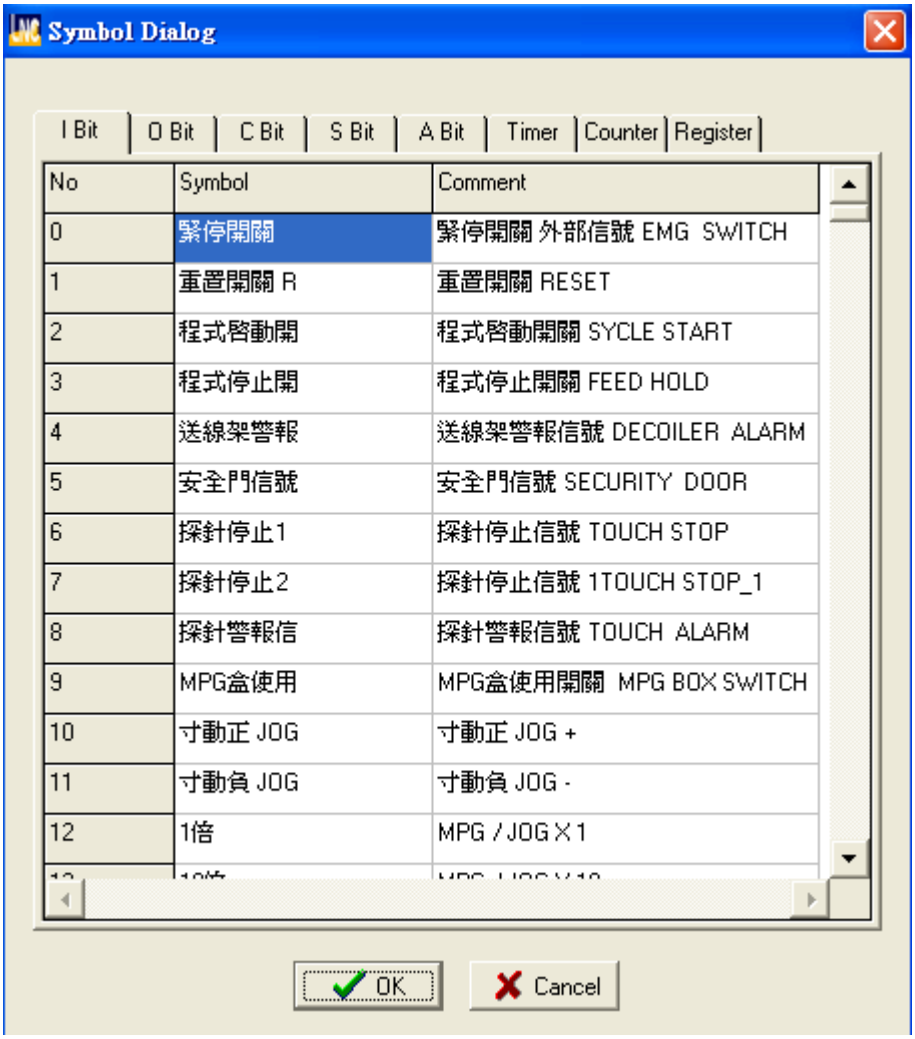


圖 一一三 註解編輯畫面

可以選擇 I bit、O bit、C bit、S bit、A bit、Timer、Counter、Register 頁面去做編輯

選擇 Register 頁面如圖 一一四

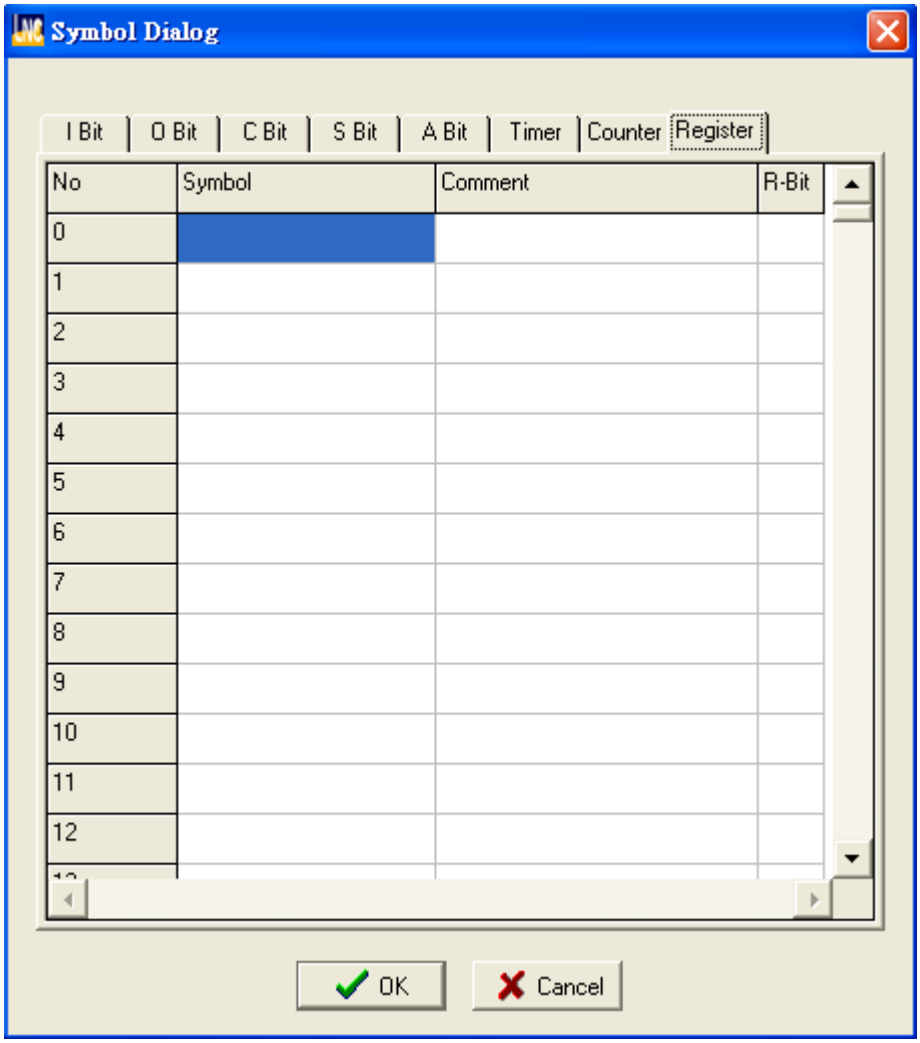


圖 一一四 選擇 Register 編輯

如果 Register 的 bit 有註解會秀於右邊有個 Click 字樣如圖 一一五

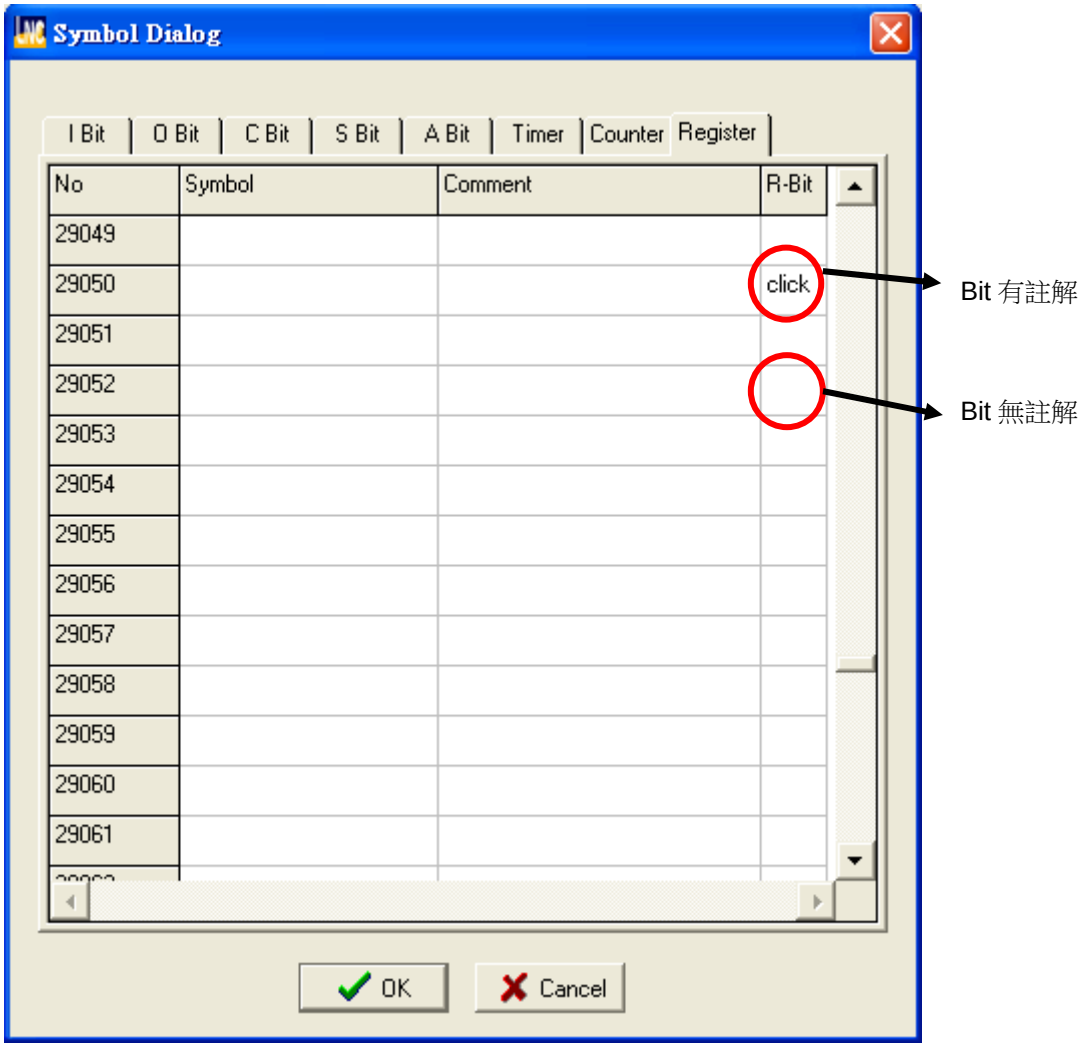


圖 一一五 Register 註解編輯畫面

不論有無註解皆可點選 R-Bit 欄位去編輯 Bit 註解如圖 一一六

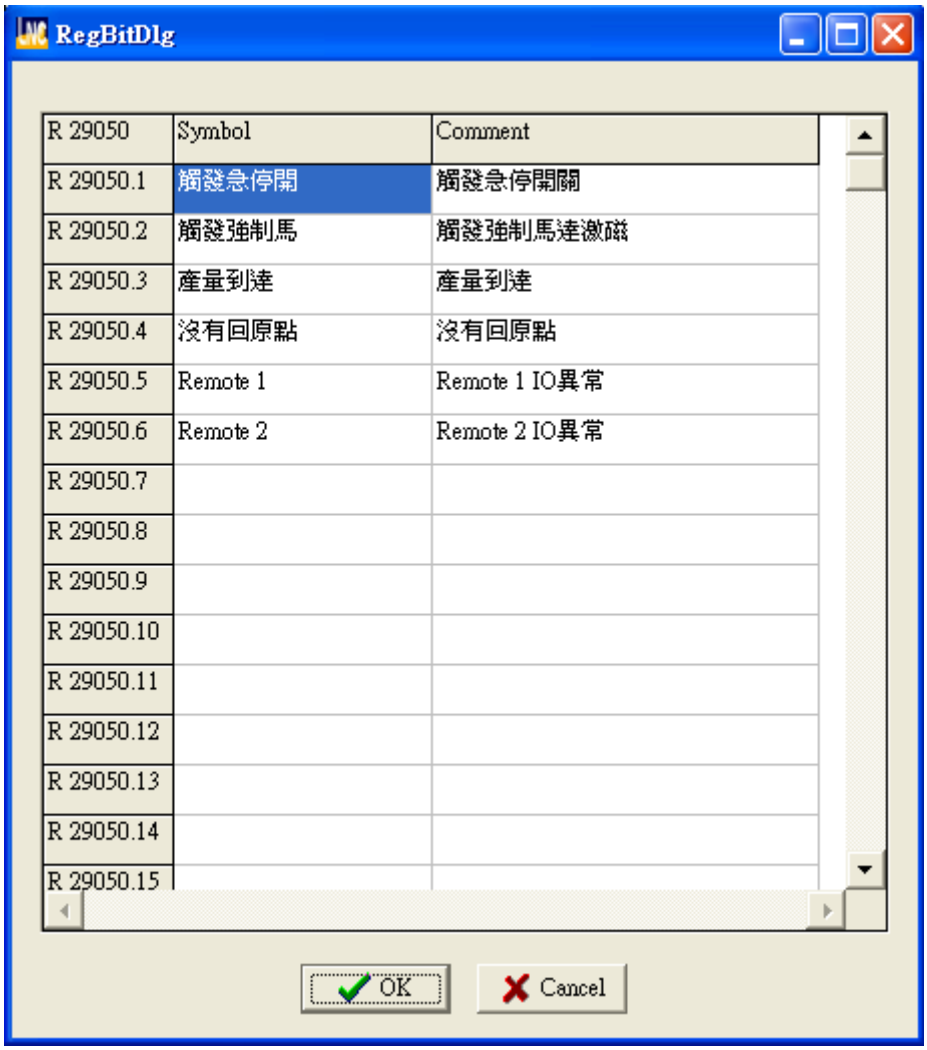


圖 一一六 註解編輯畫面

點選兩下即可編輯，編輯完畢按下 OK 即會儲存並跳回 Register 註解編輯畫面，完成編輯後按下 OK 就會更新註解，並跳回主畫面，同時更新畫面上對應的註解。

當使用 Reg-Contact 與 Reg-Coil 時，如果對應的 Register 之 Bit 無註解，會尋找 Register 之註解。

因為 PLC Alarm 對應到 R29000~R29049；PLC Warning 對應到 R29050~R29099，因此編輯對應的 Register 之 Bit 即可以秀出對應的 PLC 警報/警告字串。

11 專案註解編輯

此版提供可加註正個 PLC 的註解，點選工具列 Setting→Annotation 如圖 一一七

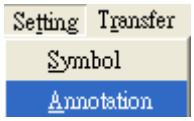


圖 一一七 PLC 註解

跳出 PLC 註解編輯畫面如圖 一一八

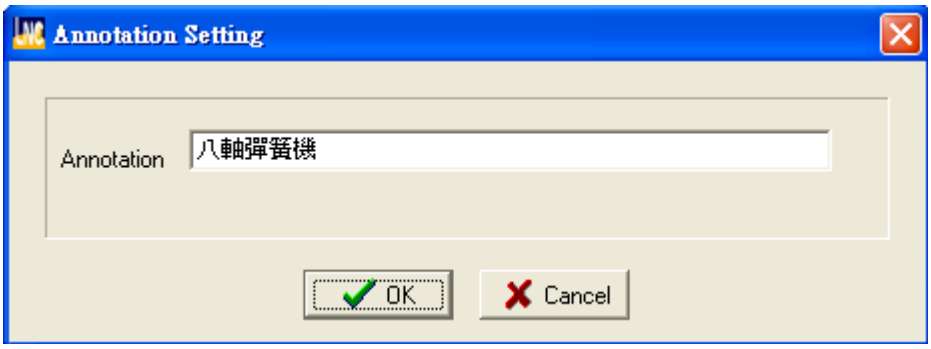


圖 一一八 PLC 註解畫面

編輯完畢按下 OK 後，抬頭會顯示出註解資訊如圖 一一九

圖 一一九 抬頭顯示註解畫面

12 PLC 編輯器版本資訊

點選工具列 Help→About 如圖 一二〇

圖 一二〇 編輯器版本資訊

跳出版權宣告與版本資訊畫面如圖 一二一，回報錯誤時請含 Editor 及 Compiler 版本。

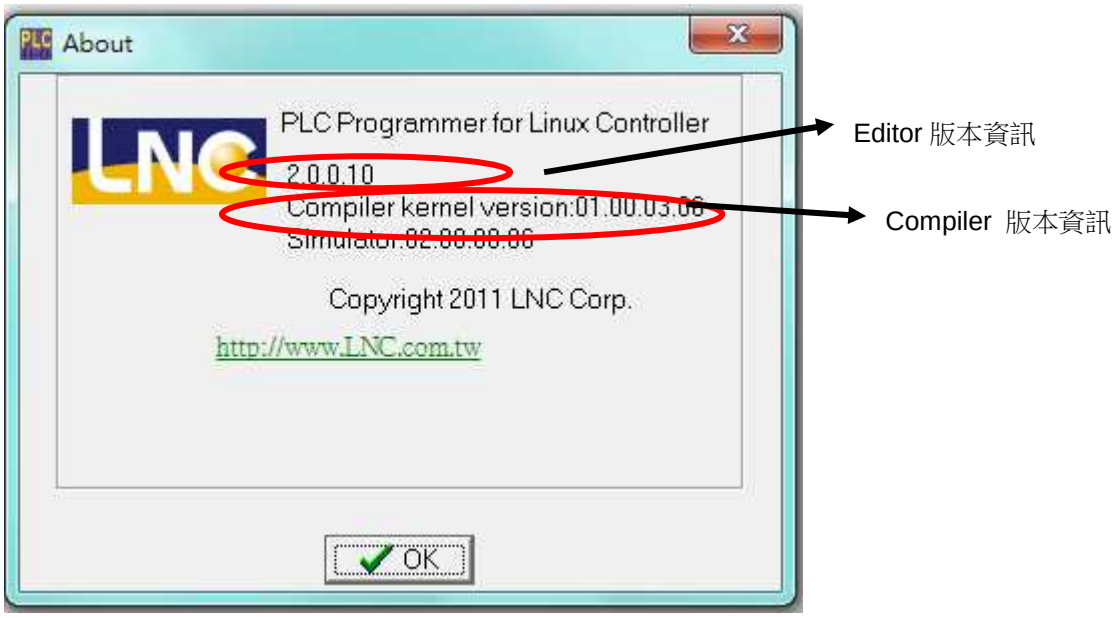


圖 一二一 版權宣告與版本資訊畫面

13 模擬器使用說明

13.1 開啟程式

有兩種方法可以開啟 PLC Simulator for Linux Controller

方法一、

1. 至安裝PLC資料夾內，例如C:\Program Files\LNC\ (預設資料夾)

2.執行PLCTester.exe檔，開啟PLC Simulator for Linux Controller畫面如<圖一零四>所示。

方法二、

1.點選桌面工具列的開始→所有程式→ LNC

2.點選PLCTester捷徑

3.開啟PLC Simulator for Linux Controller畫面如圖 一二二所示。

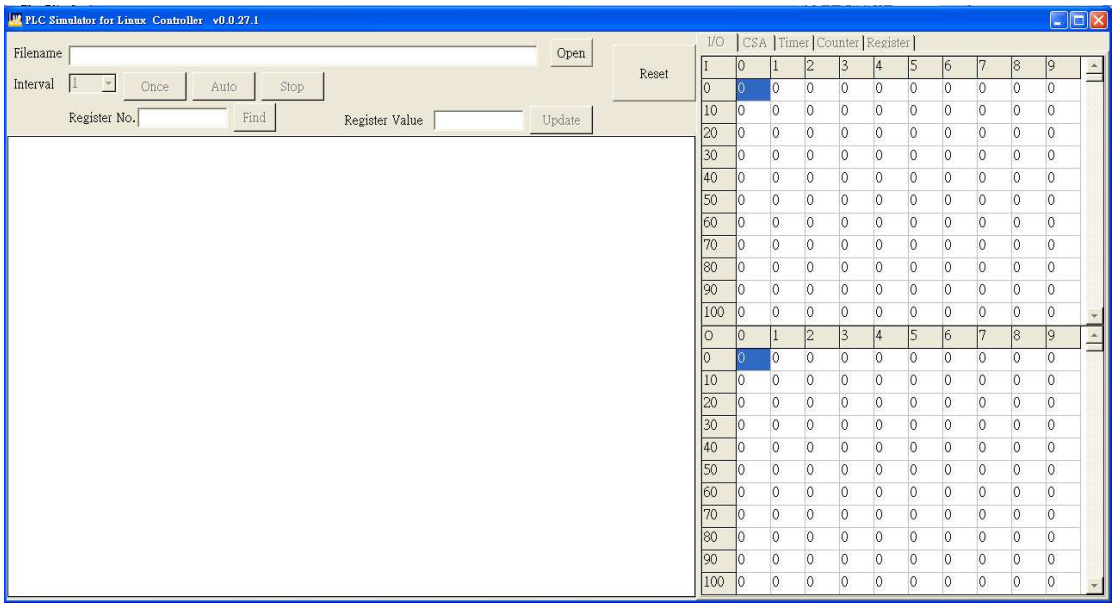


圖 一二二 PLC 模擬器

13.2 開檔(Open)

選擇 Open 按鈕，跳出開檔對話框，選擇欲開啟模擬之 PLC 檔案(cnc.lmlc)，結果如圖 一二三。

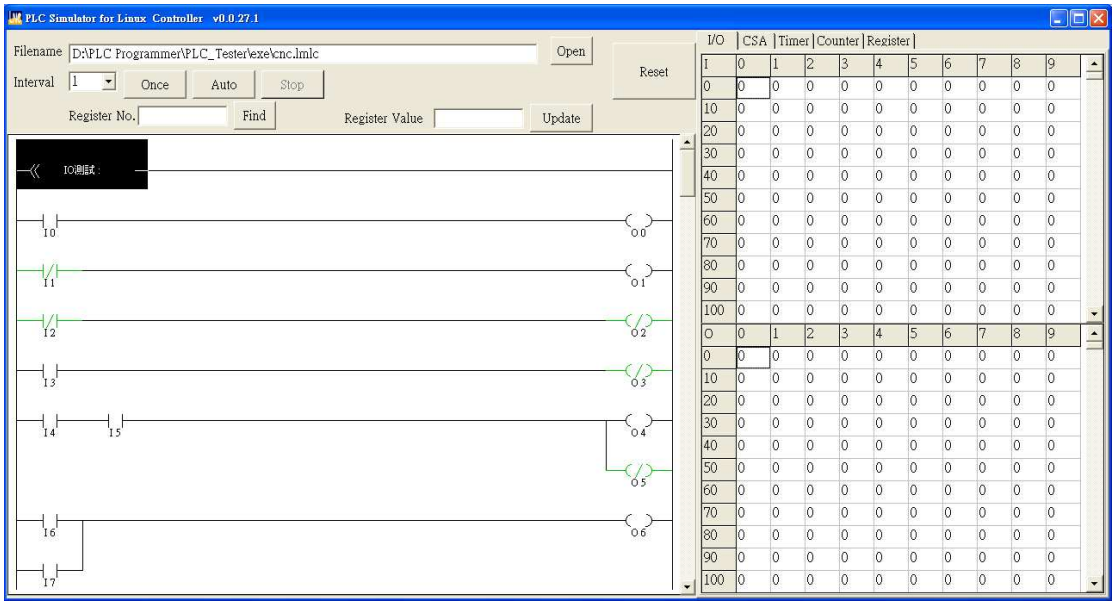


圖 一二三 開檔

13.3 功能介紹

圖 一二四

目前開啟檔名：顯示目前模擬的階梯圖。

開啟檔案：執行開檔動作。

重置狀態：將所有的 IOCSA、Timer、Counter、Register 接會清為 0。

圖 一二五

中斷週期：設定多久執行一次中斷，會影響 Timer 的結果。

執行命令：分為

- Once：每按一下只執行一次中斷。
- Auto：每次循環執行一次中斷。
- Stop：搭配 Auto 停止繼續執行中斷。

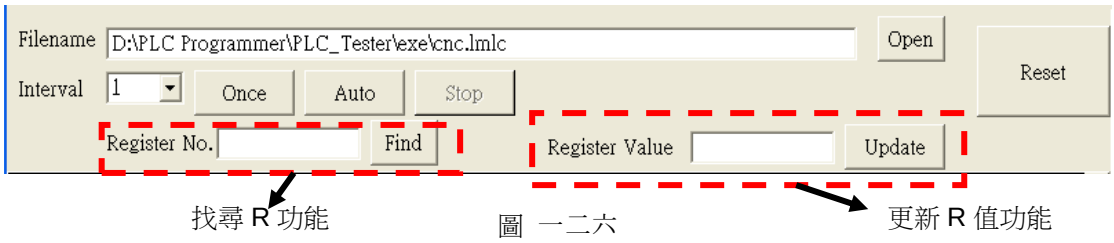


圖 一二六

找尋 R 功能：可以直接輸入欲尋找之 Reg 編號，即會跳至 Reg 畫面且標示在尋找的 Reg 編號。

更新 R 值功能：可以輸入更改目前標示 Reg 編號之值。

範例如圖 一二七。

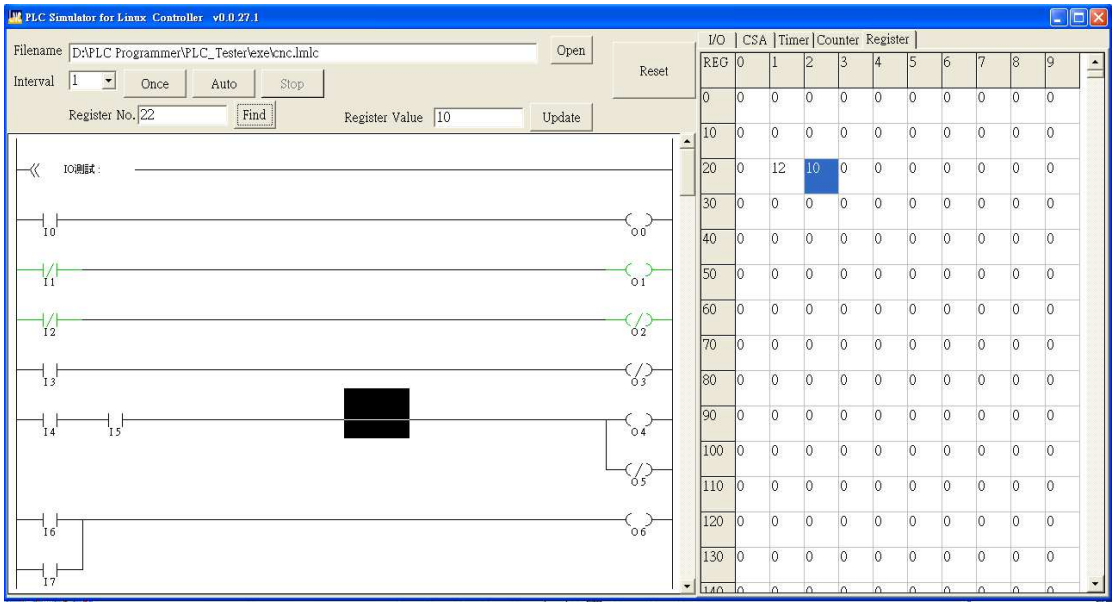


圖 一二七 尋找與設定 Register

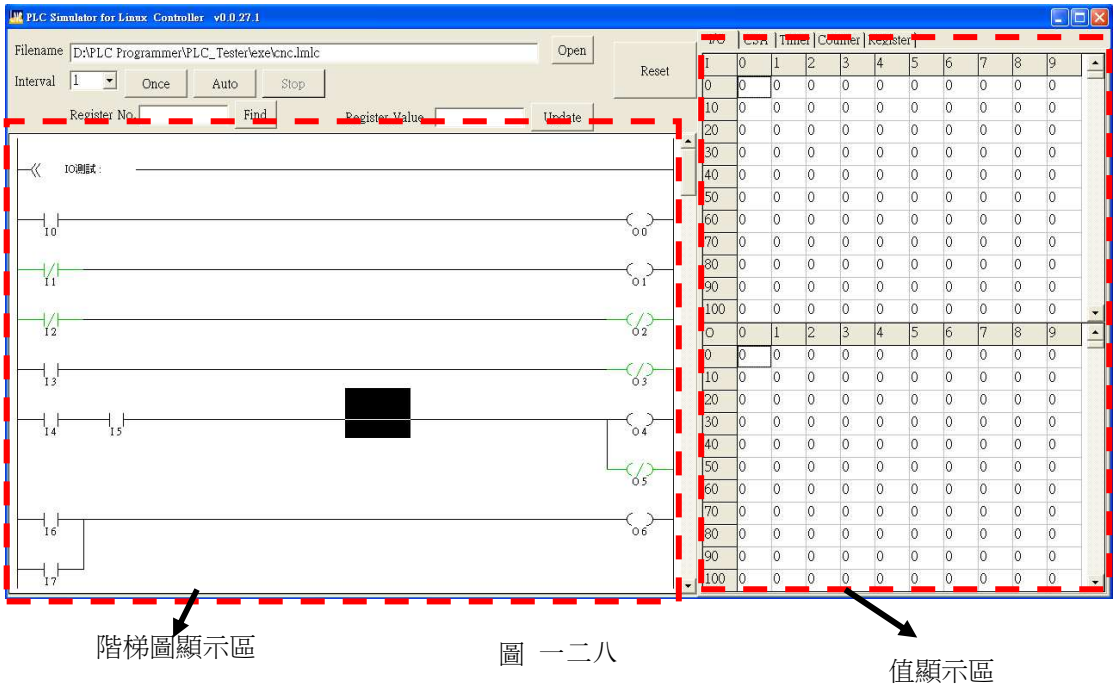


圖 一二八

階梯圖顯示區：顯示目前開啟檔之階梯圖形與狀態。

值顯示區：顯示 I、O、C、S、A、Timer、Counter、Register 值，其中 IOCSA 點兩下會改變狀態，如 1→0 或 0→1，Timer 與 Counter 不能修改，Register 必須透過上面功能才能設定。

13.4 版本資訊

顯示於左上角。　　版本資訊

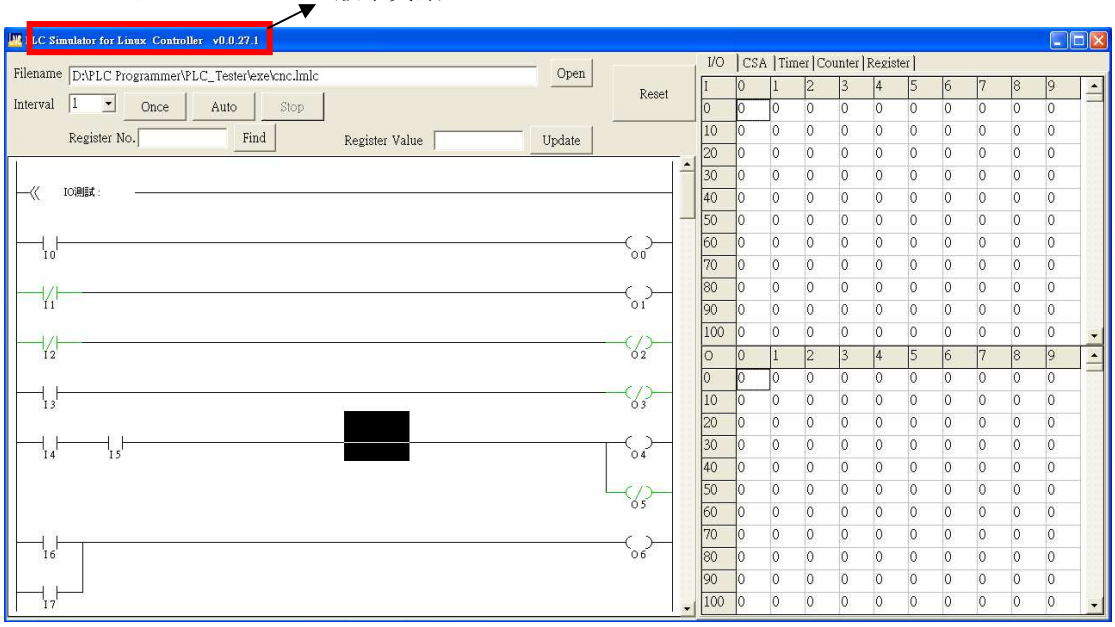


圖 一二九 版本資訊

14 編程建議

除了會造成編譯錯誤的梯型圖編程方式外，建議避免以下編程方式。

14.1 特定元件重覆使用

避免在同一個梯型圖內重覆輸出至同一編號 IOCSA 及 R bit，或使用同一編號 Timer 及 Counter。

14.2 往上區塊

避免使用往上區塊如下圖紅線標註區塊。

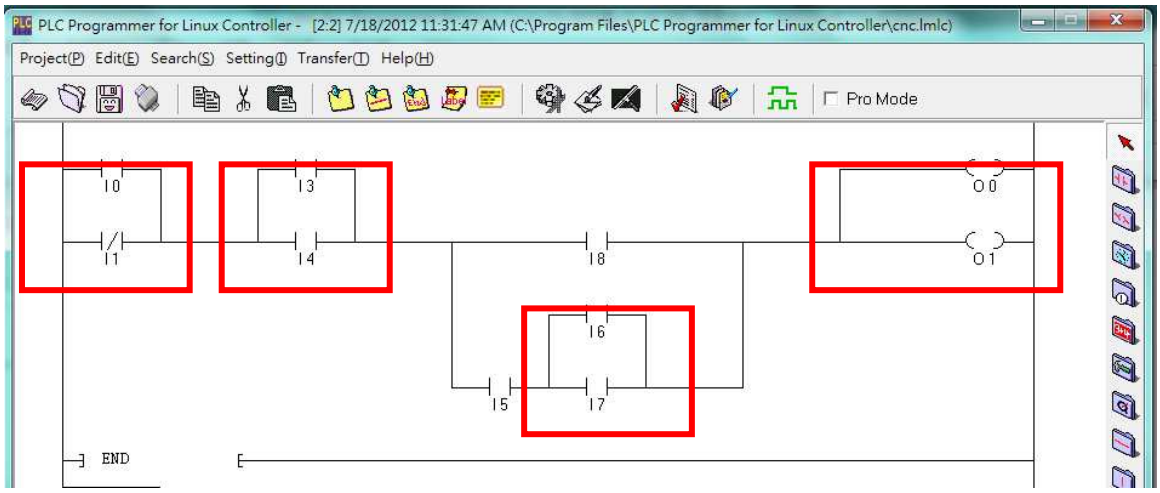


圖 一三〇往上區塊

14.3 操作資料暫存器指標指令使用

謹慎使用資料暫存器指標，避免在梯型圖內操作當做指標的資料暫存器內存值，以免更改到不欲更改區域。

如下圖，如 I000 被啟動後關閉，R 000 指向的資料暫存器會隨每個 cycle 變化，由 R001，R002，R003 等持續下去且都被設值為 3 直到系統允許指標操作最大值，若操作 R 000 值的區域離使用 R 000 做為指標的區域很遠，容易導致意外的結果。

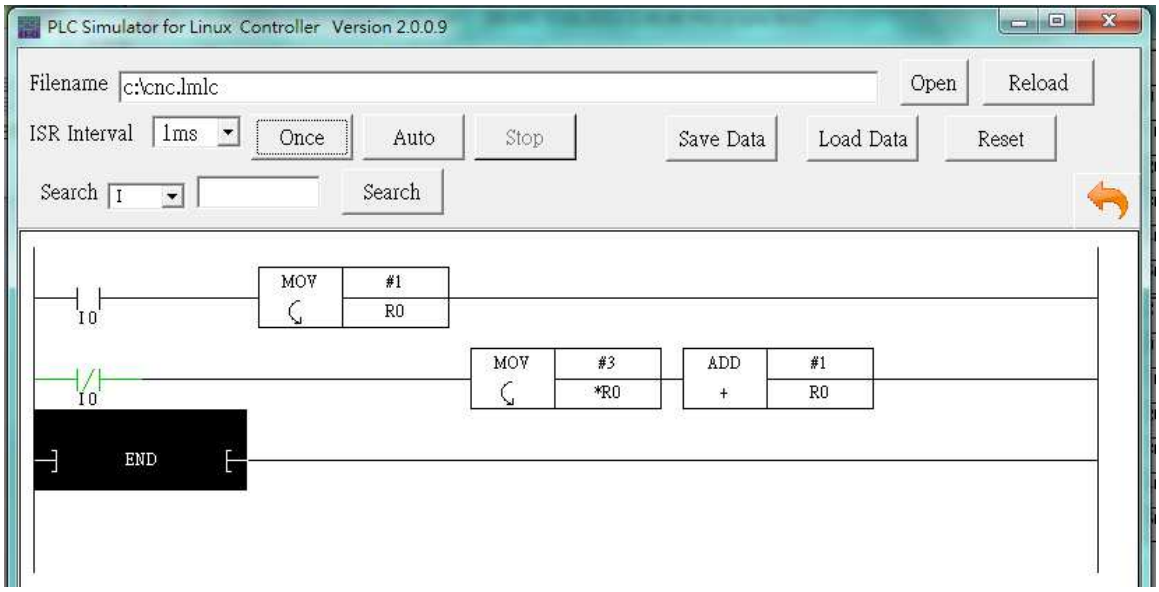


圖 一三一 操作資料暫存器指標