

第 2 章 : ISaGRAF 程式設計基礎

2.1: 一個簡單的階梯圖程式 (LD)

本章使用簡單的例子來說明五種簡單但是有效率的 IEC61131-3 語言進行 I-8xx7 控制器的程式設計，初學者請按本章的說明操作練習。

若需要知道如何使用更多 ISaGRAF 的功能,可參閱附錄 E, 或 ISaGRAF CD_ROM 中“ISaGRAF USER’S GUIDE”手冊的電子檔“ISaGRAF.pdf”或是“ISaGRAF.doc”。

階梯圖語言程式設計基礎

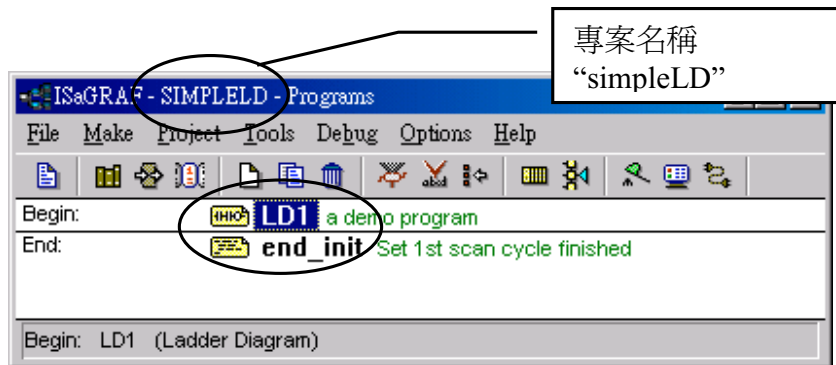
階梯圖程式語言是一種結合 contacts (輸入參數) 和 coils (輸出參數)的圖形化布林方程式語言，其與電機人員在工業控制系統上常用的電路圖表示法類似，階梯圖程式語言讓程式設計師將硬體設備的狀態及其改變情形以圖示法表示出來。

階梯圖語言程式設計範例

我們將依序引導您利用 I-8xx7 控制器提供的 ISaGRAF 作業平台建構一個階梯圖語言程式。

同時我們會另外建一個 結構化文字 語言程式 (ST), 如此在這個 ISaGRAF 專案範例中,我們有 2 個程式包函在裏面。

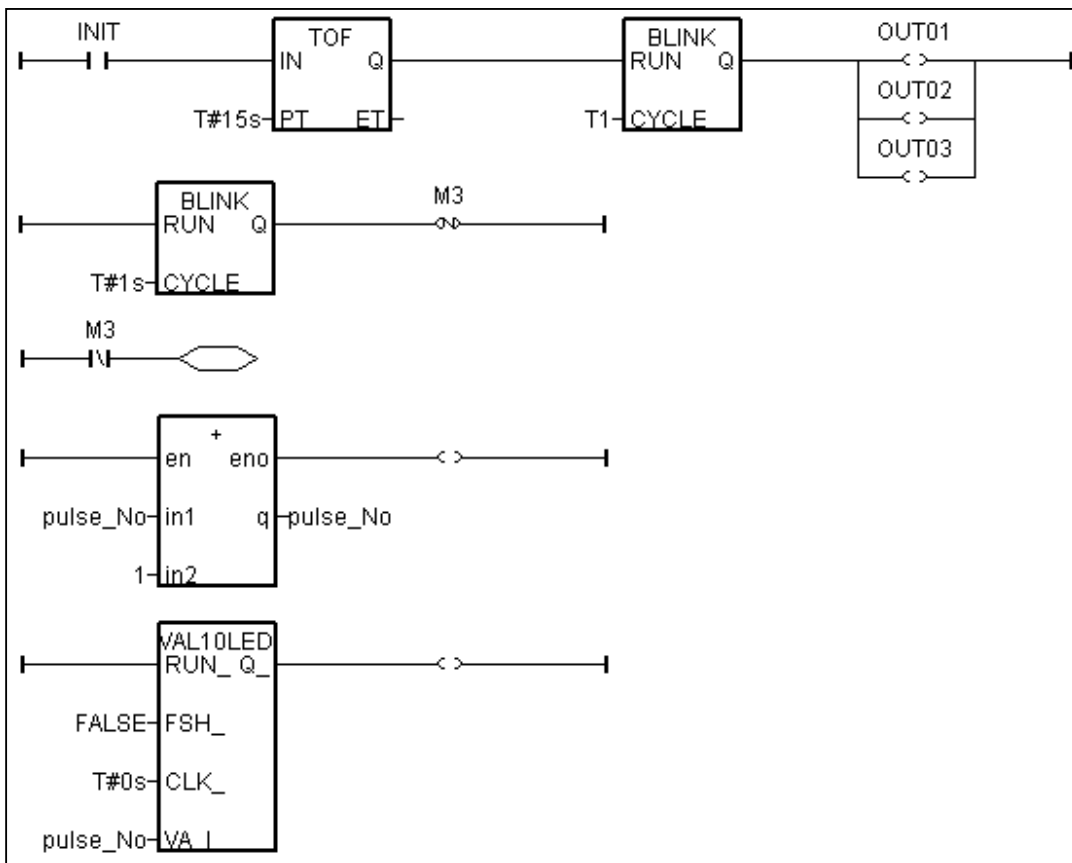
我們把這個專案取名為 “simpleLD”. 而此專案內的 LD 程式取名為 “LD1”, ST 程式取名為 ”end_init”



本範例內的變數宣告：

Name	Type	Attribute	Description
INIT	Boolean	Internal	初值設為 TRUE. TRUE 表示 第 1 個 Scan 周期
M3	Boolean	Internal	用來表示 pulse 有沒有產生
OUT01	Boolean	Output	Output 1
OUT02	Boolean	Output	Output 2
OUT03	Boolean	Output	Output 3
T1	Timer	Internal	閃爍周期時間, 初值設為 T#1s
Pulse_No	Integer	Internal	當 pulse 產生時, 加 1, 初值設為 0

階梯圖程式“LD1”全貌



結構化文字程式“end_init”全貌

```
INIT := FALSE ;
```

範例程式邏輯說明：

“LD1” 程式:

1. 在程式執行起來的前 15 秒, 閃爍輸出 Outputs 1, 2, & 3, 閃爍周期為 “T1”, “T1” 的初值為 1 秒. 15 秒過後, Outputs 1, 2, & 3 會設成 FALSE.
2. 每秒產生一個 Pulse 到布林內部變數 “M3” 內.
3. 每當 M3 – Pulse 產生時, 將整數變數 “pulse_No” 的值加 1.
4. 將 “pulse_No” 的值顯示到 I-8xx7 的 7 段式顯示 Led 上.

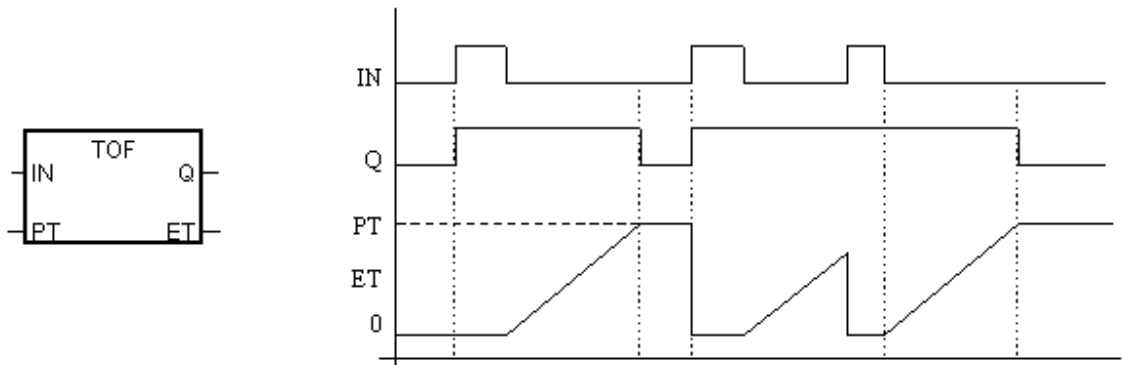
ST 程式 “end_init”:

1. 在本專案的末端, 將布林變數 “INIT” 的值設為 FALSE. 如此 INIT 只會在第 1 個 Scan 周期時為 TRUE .

方塊及一些 LD 元件的描素:

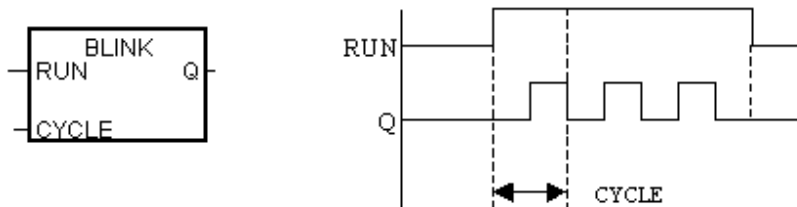
TOF: 延遲一個布林值一段時間後才設為 FALE.

- “IN” 參數為布林型態, 當它剛好從 TRUE 降為 FALSE 時. 此方塊開始從 0 秒計時到 “PT”
- “PT” 參數為計時器型態, 它指的是所要延遲的時間.
- “Q” 為此方塊的布林輸出. 當計時時間到達 “PT” 時, 就會被設成 FALSE.
- “ET” 為此方塊的計時器輸出. (本範例沒有使用它)



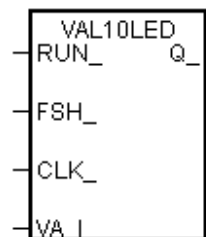
BLINK: 閃爍一個布林

- “RUN” 參數為布林型態, 當它為 TRUE 時, 會產生一個閃爍的輸出到 “Q”, 周期為 “CYCLE”.

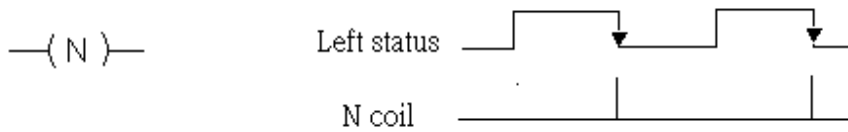


VAL10LED: 顯示 1 個整數值到 I-8xx7 的 7 段式顯示 Led 上.

- “RUN_” 參數為布林型態. TRUE 就顯示.
- “FSH_” 參數為布林型態. TRUE 會閃爍該顯示的數字.
- “CLK_” 參數為計時器型態. 為閃爍的周期時間.
- “VA_I_” 參數為整數型態, 指的是所要顯示的數字.



“N” coil : 當 coil 左端的狀態剛好由 TRUE 降為 FALSE 時, 該 coil 會產生一個 Pulse 輸出.



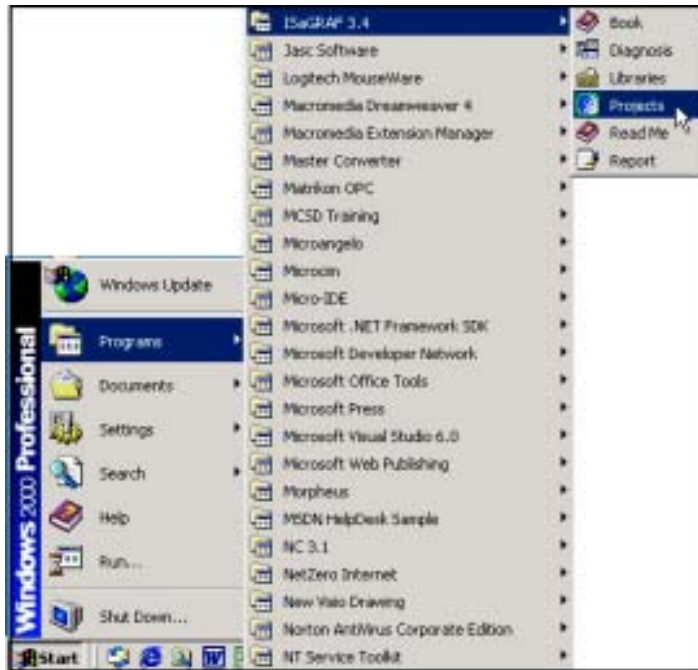
“Retrun” : 當 return 左端的狀態為 TRUE 時, 在此 return 以下的 LD 程式不會被執行.



2.1.1: 程式撰寫

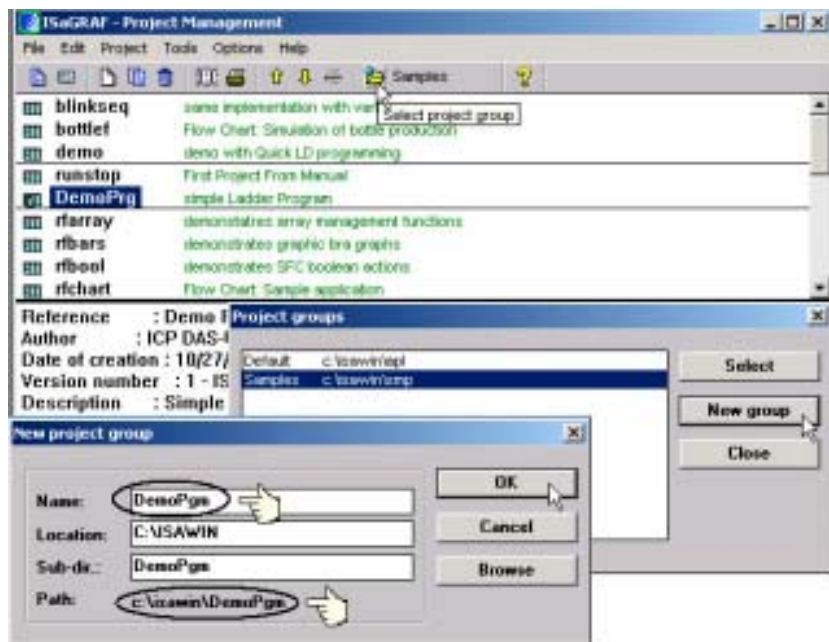
啓動 ISaGRAF 作業平台

如下圖所示，選擇左下方的[開始]->[程式集]->[ISaGRAF 3.4]->[Projects]



2.1.1.1: 建立 ISaGRAF 使用者群組

如下圖所示，選擇[Select Program Group]->[New Group]，輸入您想建立的群組名稱，按下"OK"鍵。

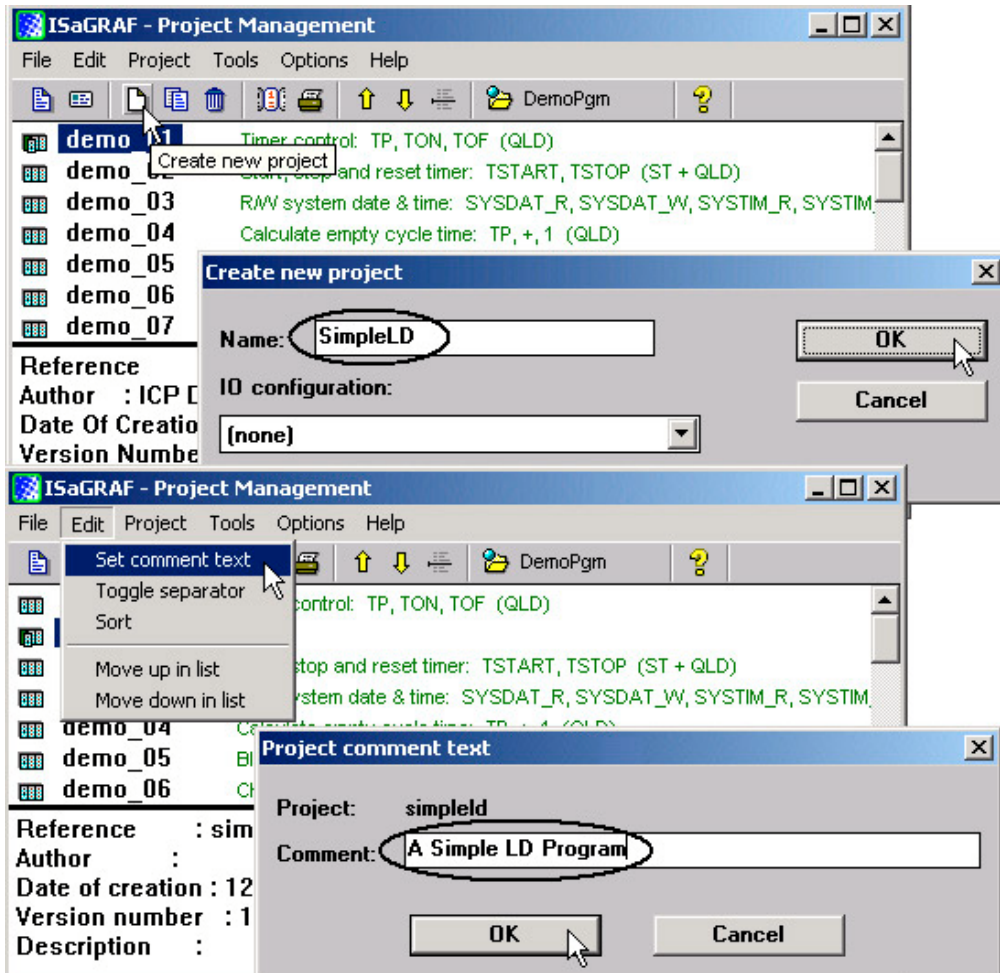


ISaGRAF 將依照您輸入的群組名稱在"New Project Group"加入一個新的資料夾，並且在 "c:\isawin"下亦同時產生一個對應的資料夾。

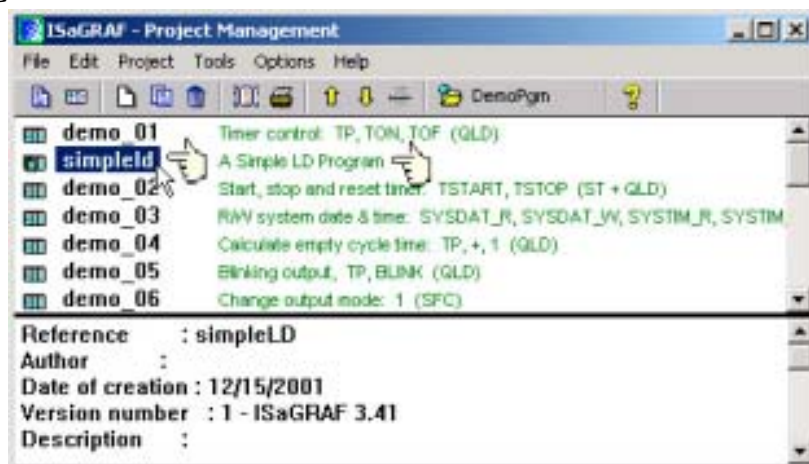
您可以雙擊這個新群組名稱以進入該新群組，或者選擇這個新群組的名稱(群組名稱會變亮)，然後按下"Select"鍵進入新群組。

2.1.1.2: 建立新的 ISaGRAF 專案

您可以點選"Create New Project"圖示，並鍵入您要的專案名稱以建立您的 ISaGRAF 專案，如果您想要替您的專案加註解，請選擇[Edit]>[Set Comment Text]

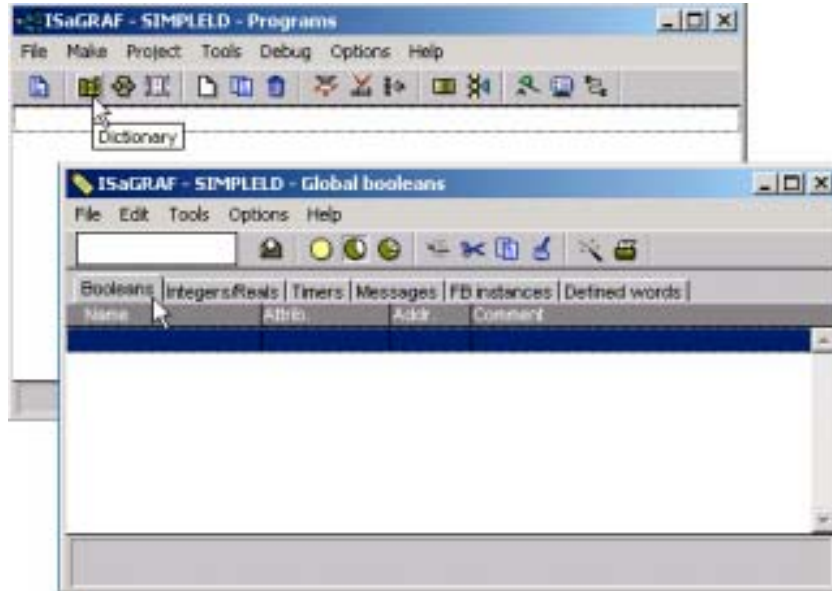


您現在在"Project Management"視窗可以見到您的專案名稱，您可以用滑鼠雙擊您的專案名稱來開啓它。

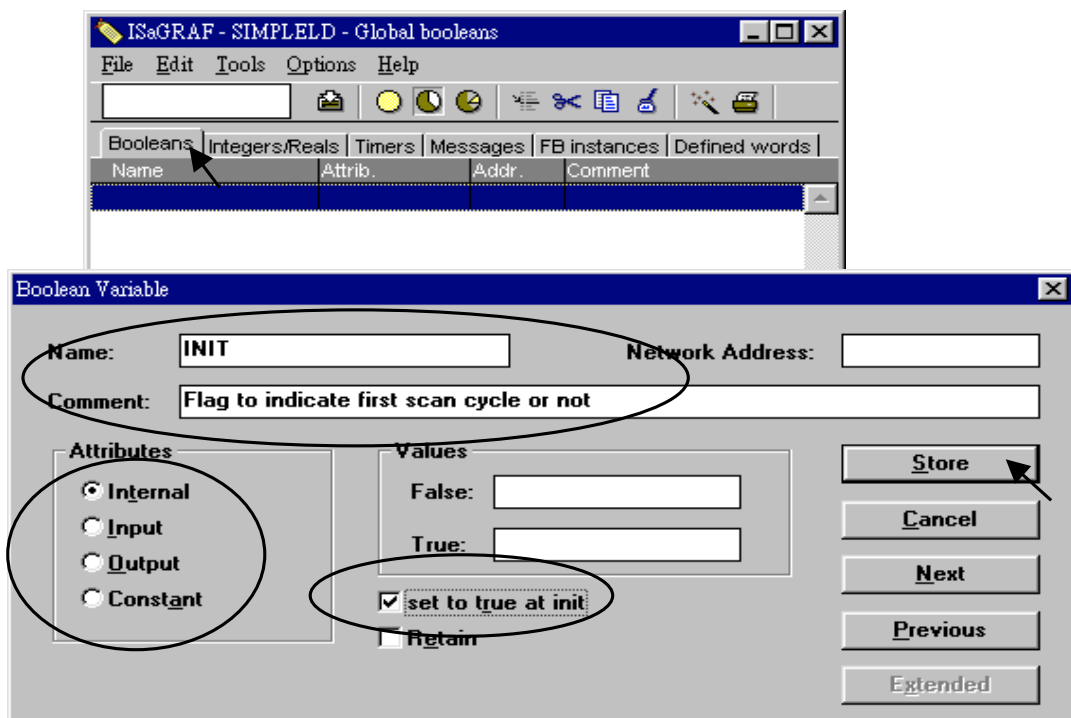


2.1.1.3: 宣告 ISaGRAF 專案變數

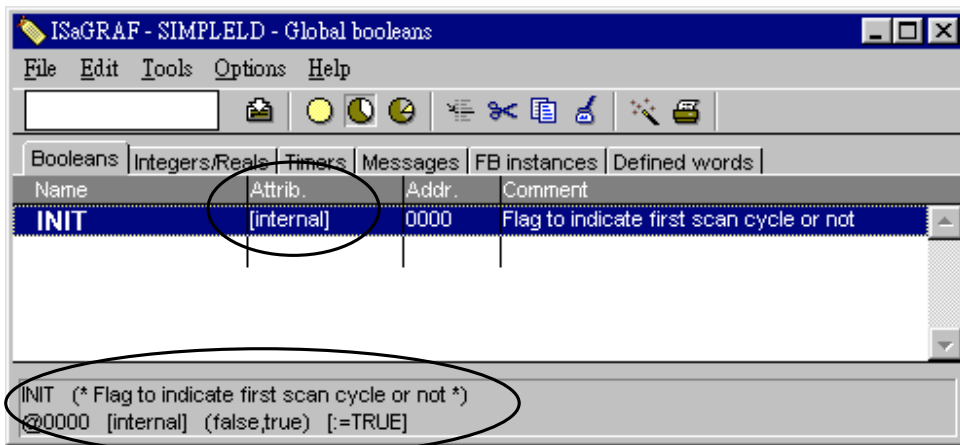
在您開始撰寫 ISaGRAF 程式前您必須先宣告在程式中的各個變數，首先，請點選"Dictionary"圖像，再點選"Boolean"標籤宣告在程式中會用到的布林變數。



您可以利用滑鼠雙擊"Boolean" 標籤下有顏色的區域開啓"Boolean Variable"視窗，並依序鍵入您需要的變數名稱，本範例中輸入布林參數名稱爲"INIT"，在"Comment"欄加入"Flag to indicate first scan cycle or not"註解，然後須選擇"INIT"變數的屬性，在此範例中"INIT"的屬性爲"Internal"，並將"set to true at init"打勾，完成上述步驟後，按下"Store" 鍵，儲存相關的設定。

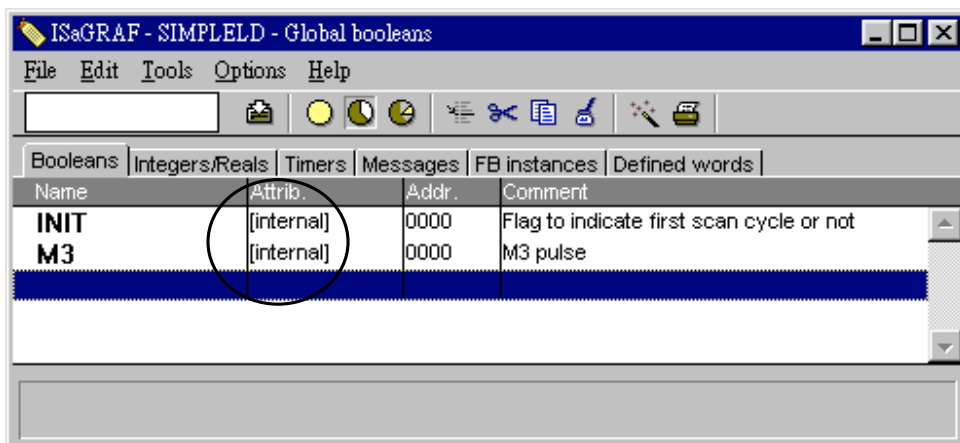


如此便宣告一個布林變數了，在變數後方各欄提供程式設計者詳盡的變數相關訊息。

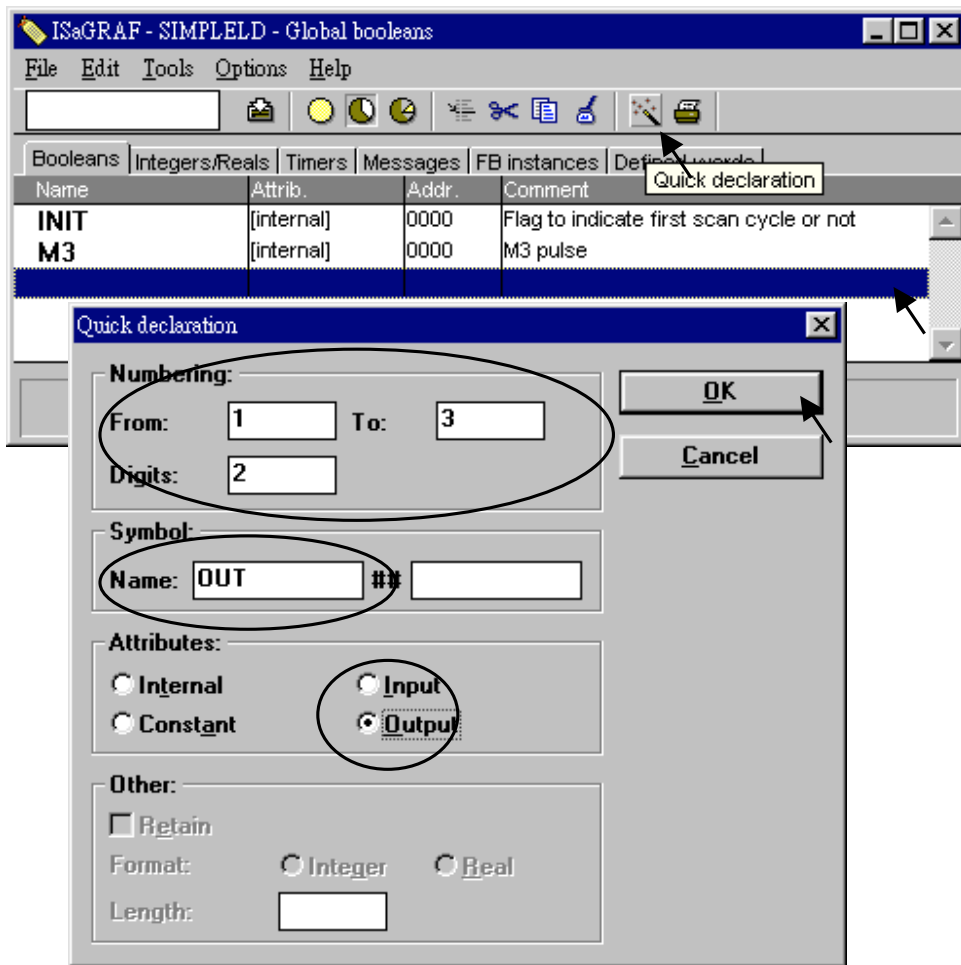


注意：您必須確認您欲宣告的變數有正確的屬性，如果您需要改變變數的屬性，您只需用滑鼠雙擊變數名稱，您便可重新設定變數的屬性。

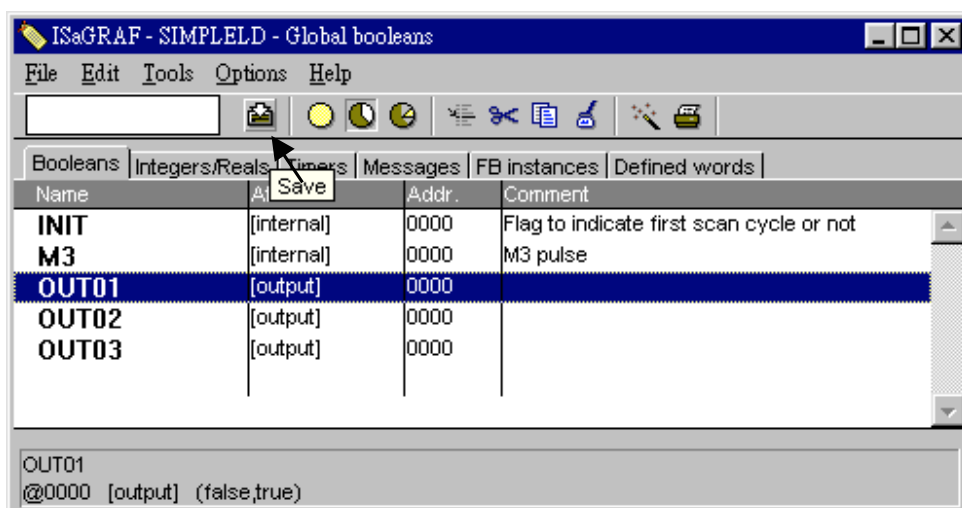
重複上述的步驟您可以完成"M3"的宣告, 結果會如下圖所示。



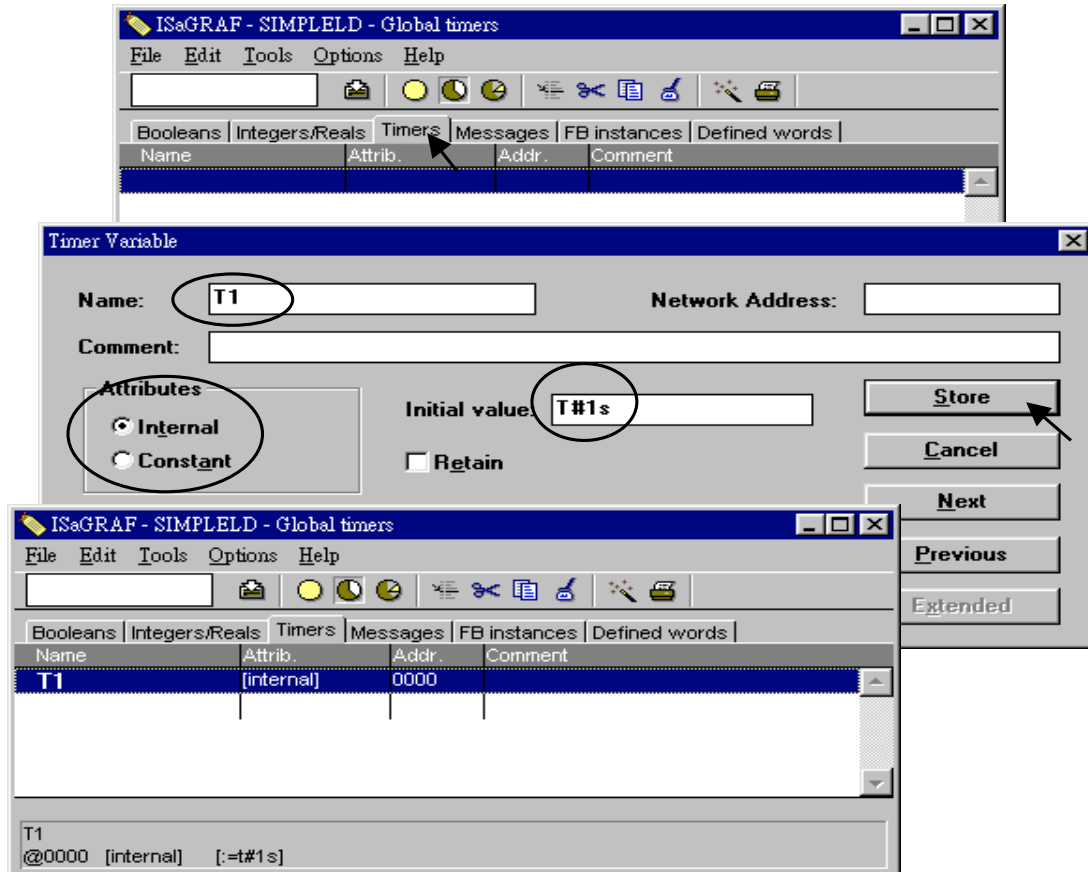
有 3 個輸出變數"OUT01, OUT02, 和 OUT03"還沒有宣告，ISaGRAF 提供一種簡便快速的方法宣告這一系列同性質的變數，您如果想使用這種方法只需要用滑鼠單擊"Quick Declaration"圖像，並在"NAME"欄輸入"OUT"，在"Numbering"內的"from"及"To"欄輸入序號數(此範例為 1 到 3)，最後選擇"Output"屬性，按下"OK"鍵



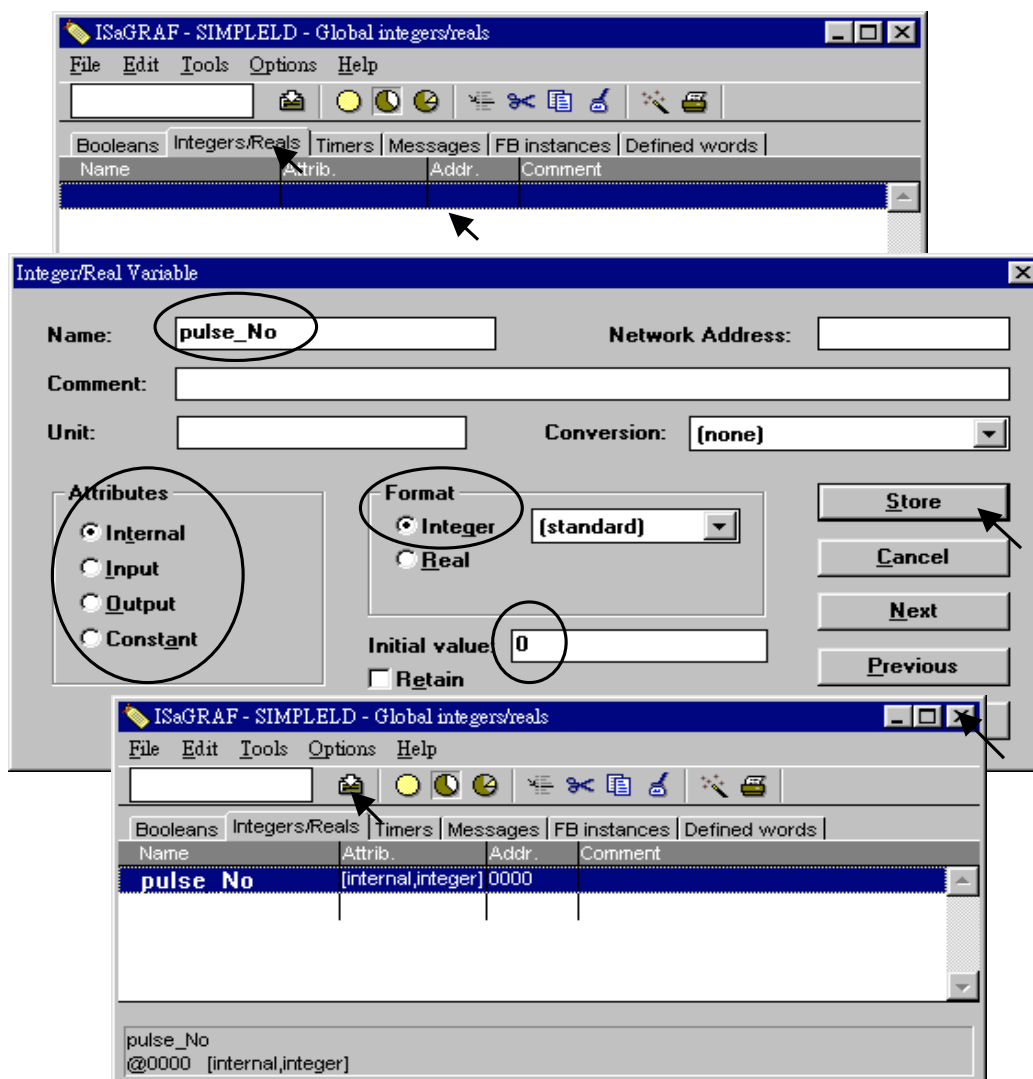
然後你會得到如下的視窗, 按一下“Save” 儲存所宣告的資料。



接下來宣告計時器變數(T1)，類似上述的步驟，用滑鼠單擊 Global 設定視窗上的"Timers"標籤並雙擊有顏色的區域開啓"Timer Variable"視窗，鍵入變數名稱"T1"，設定屬性爲"Internal"，初始值(Initial Value)設定爲"T#1s"，最後按"Store"鍵儲存設定，如下圖所示。



要宣告整數變數 (pulse_No)，用滑鼠單擊 Global 設定視窗上的" Integers/Reals "標籤，並雙擊有顏色的區域開啓" Integers/Reals Variable"視窗，鍵入變數名稱"pulse_No"，設定屬性爲"Internal"，格式爲 "Integer"，初始值(Initial Value)設定爲"0"，最後按"Store"鍵儲存設定，如下圖所示。

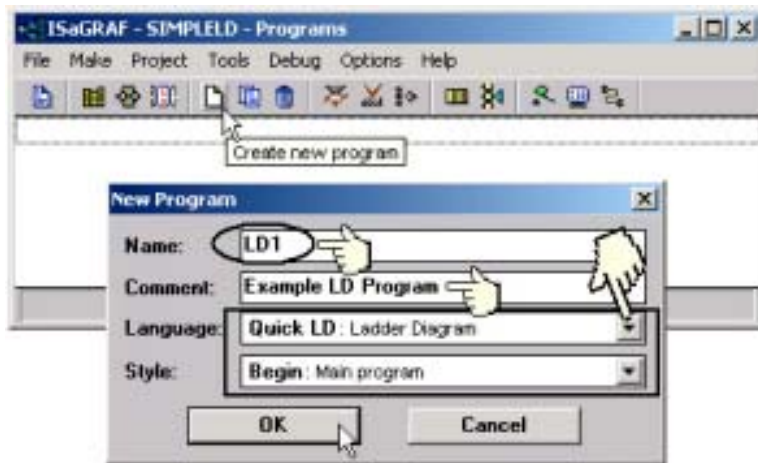


當您設定好全部的變數後您可以點選右上角"X"圖像來關閉視窗。

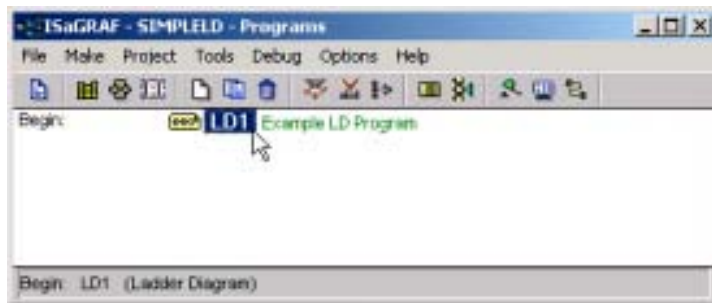
2.1.1.4: 建立 ISaGRAF 範例程式

當您完成所有的變數設定之後，現在可以開始建立 LD 範例程式了，請先點選"Create New Program"圖像，您會見到"New Program"視窗。

首先，在"Name"欄輸入"LD1"(本 LD 範例的名稱)，然後在"Language"欄內選擇"Quick LD: Ladder Diagram"，在"Style"欄內選擇"Begin: Main Program"，您可以在"Comment"內加入註解，不過這不是必須的。

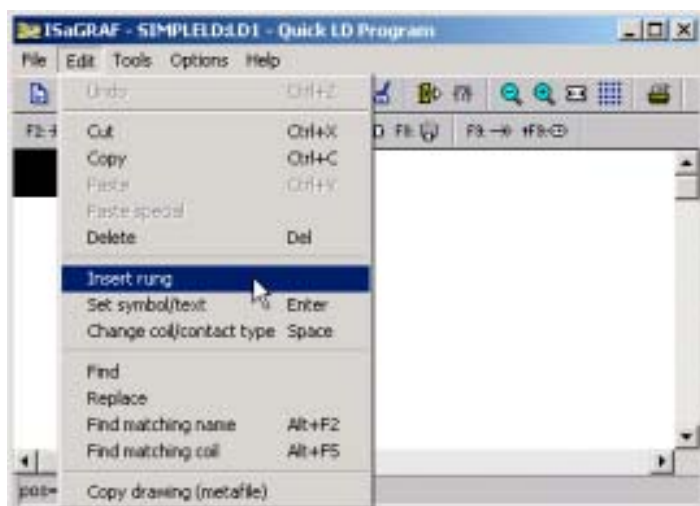


此時您已經建立了一個空的 LD 程式，您可以用滑鼠雙擊"LD1"名稱來開啓程式編輯器。

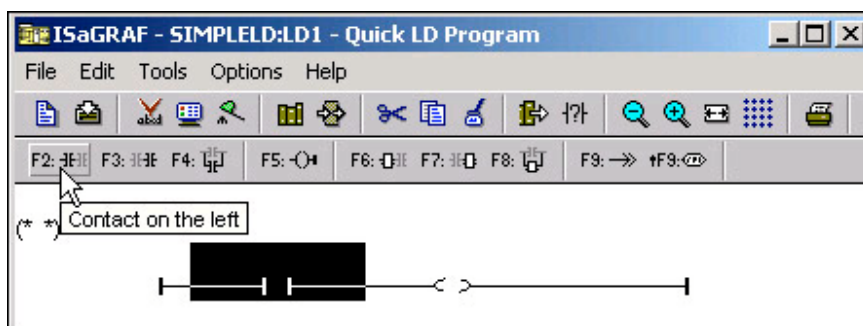


2.1.1.5: 編輯"LD1"範例程式

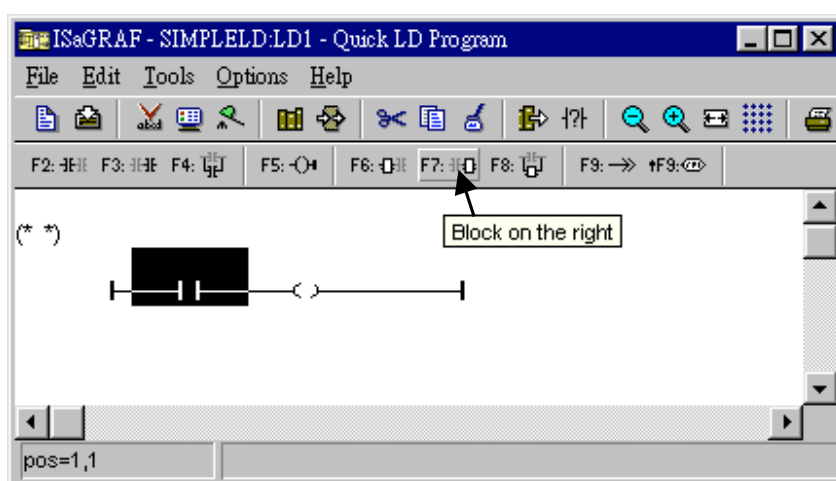
當您用滑鼠雙擊"LD1"後您會看到"Quick LD Program"視窗，如下圖所示，在"Edit"選單中點選"Insert Rung"



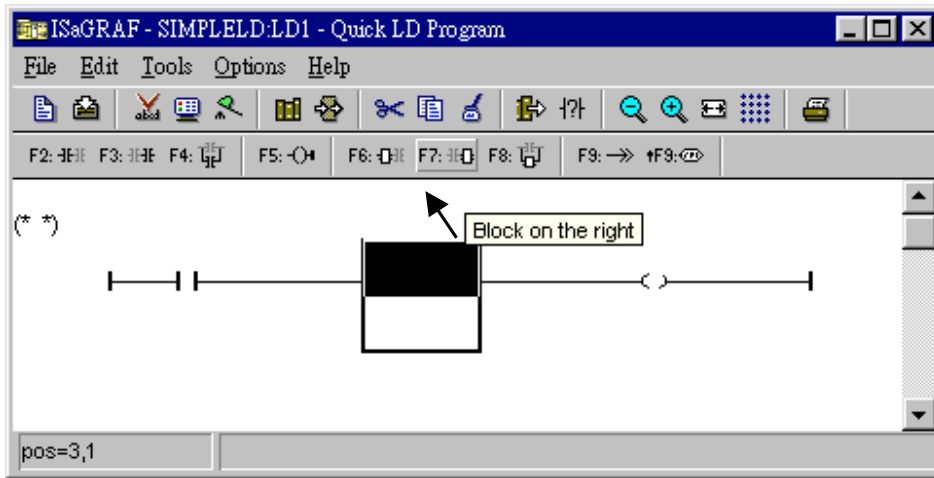
或者，您可以用滑鼠直接點選"F2 (Contact On The Left)"圖像，將會產生如下圖所示的"Quick LD Program"視窗。



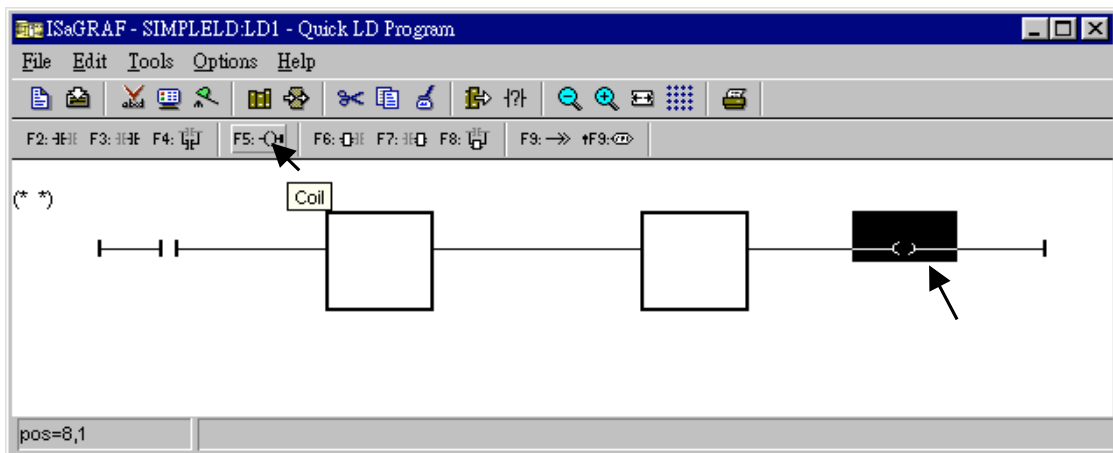
用滑鼠點選 "F7 (Block on the right)", 可以在郵標的右方加入 1 個空的方塊.



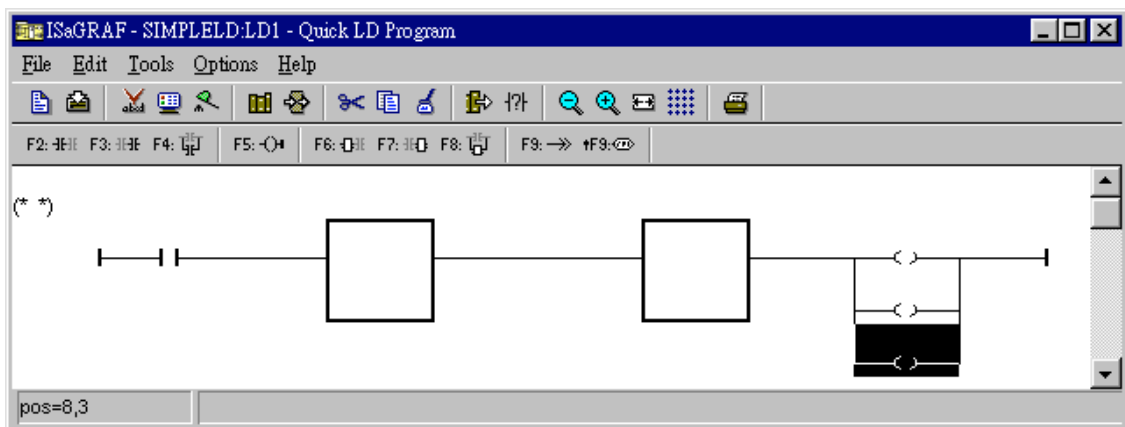
再用滑鼠點選 "F7 (Block on the right)"一次 ,在右方再加入 1 個空的方塊.



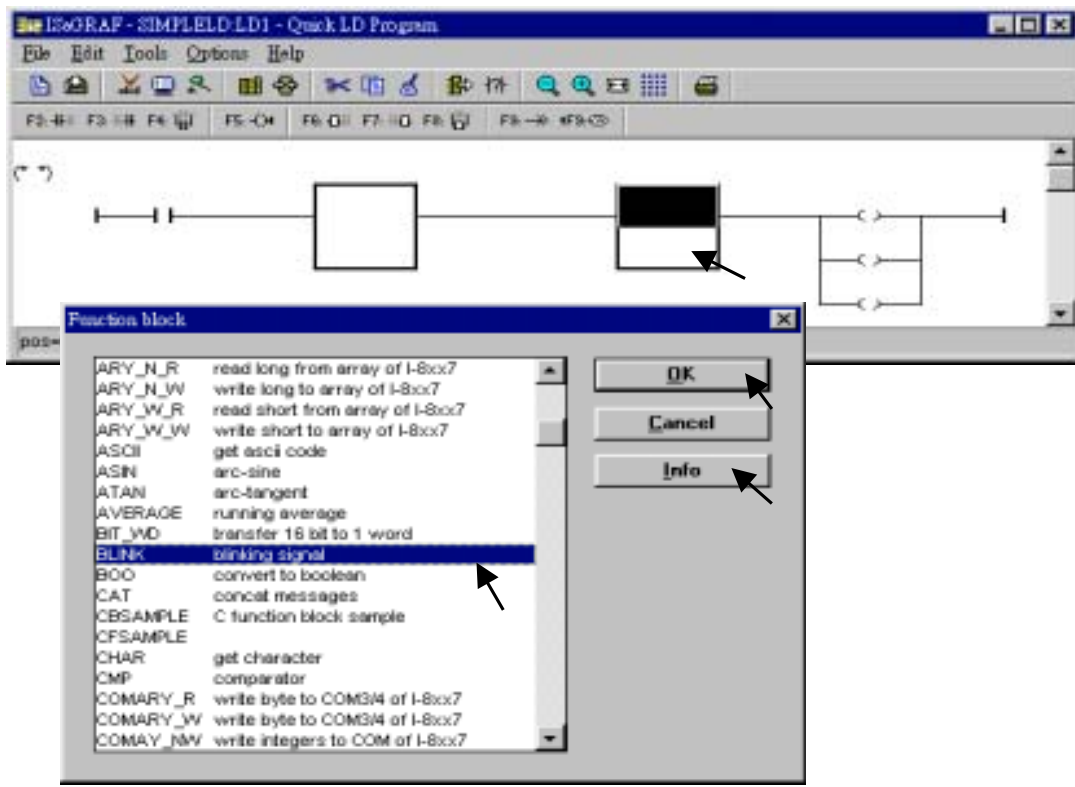
然後得到如下的視窗. 用滑鼠點選右方的 coil 將郵標移到右方. 然點選 "F5 (Coil)" 以增加 1 個 coil, 之後再點選 "F5 (Coil)" 1 次以增加第 3 個 coil.



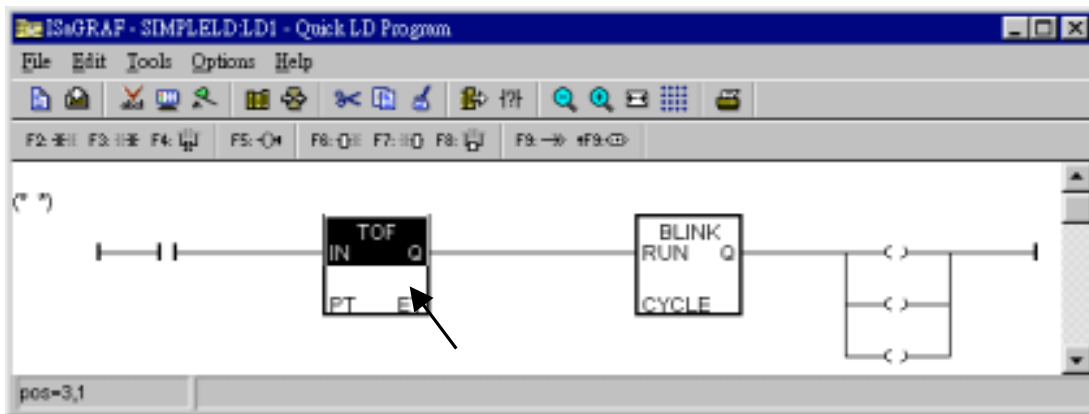
然後得到如下的視窗.



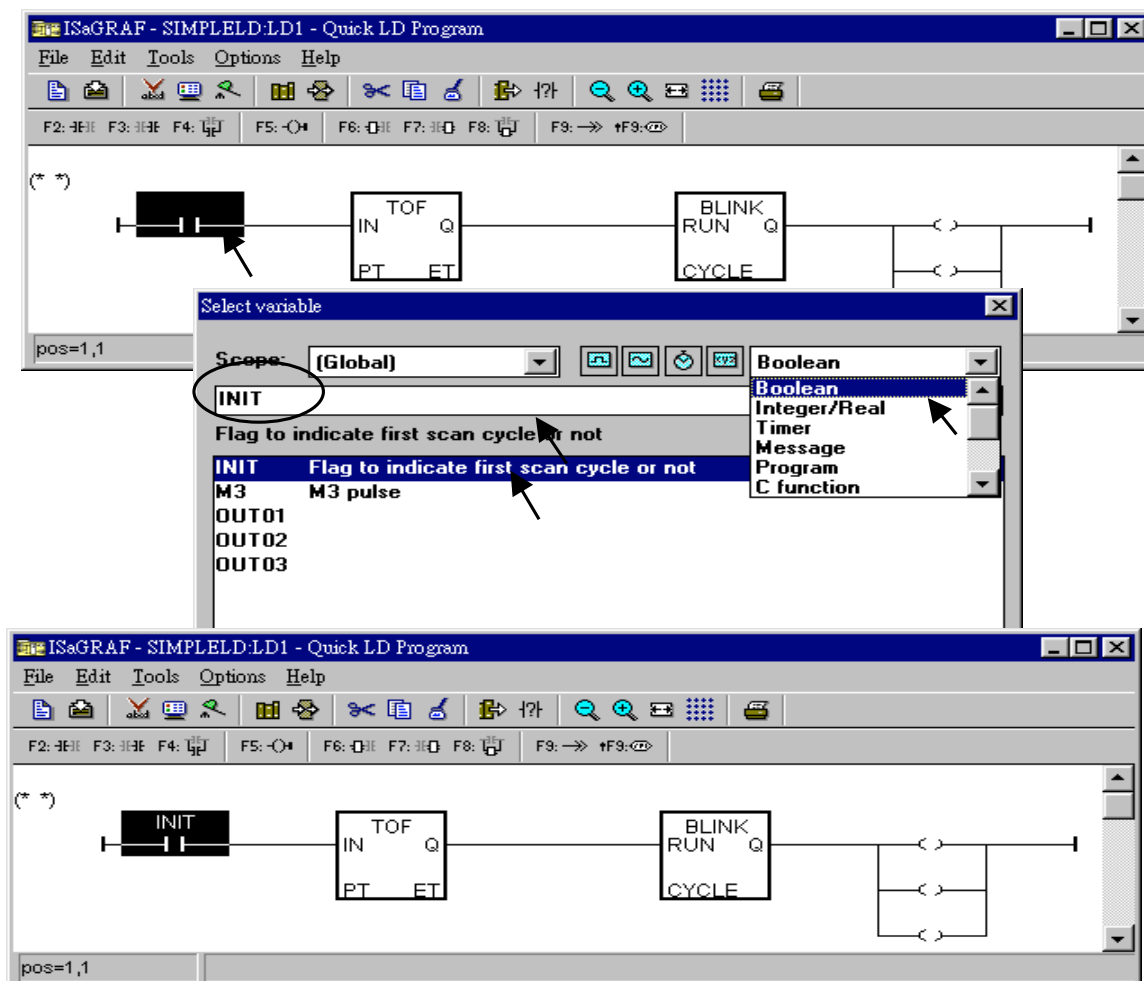
用滑鼠雙擊在右方的方塊, 之後會出現 "Function Block" 視窗. 選取"BLINK" 方塊, 之後再單擊 "OK". 若要查看該方塊的說明, 可以單擊 "Info".



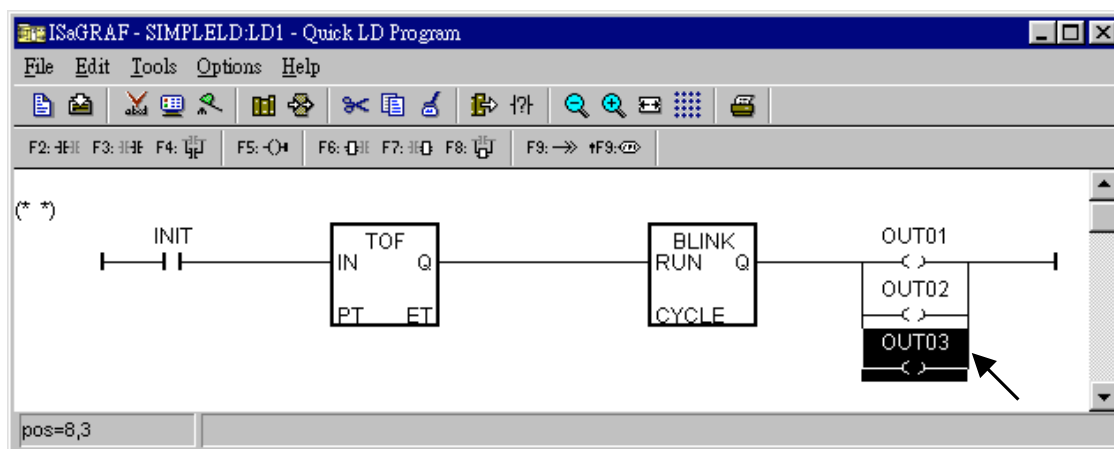
使用同樣的方法, 把 "TOF" 指定給左側的方塊, 如下圖所示.



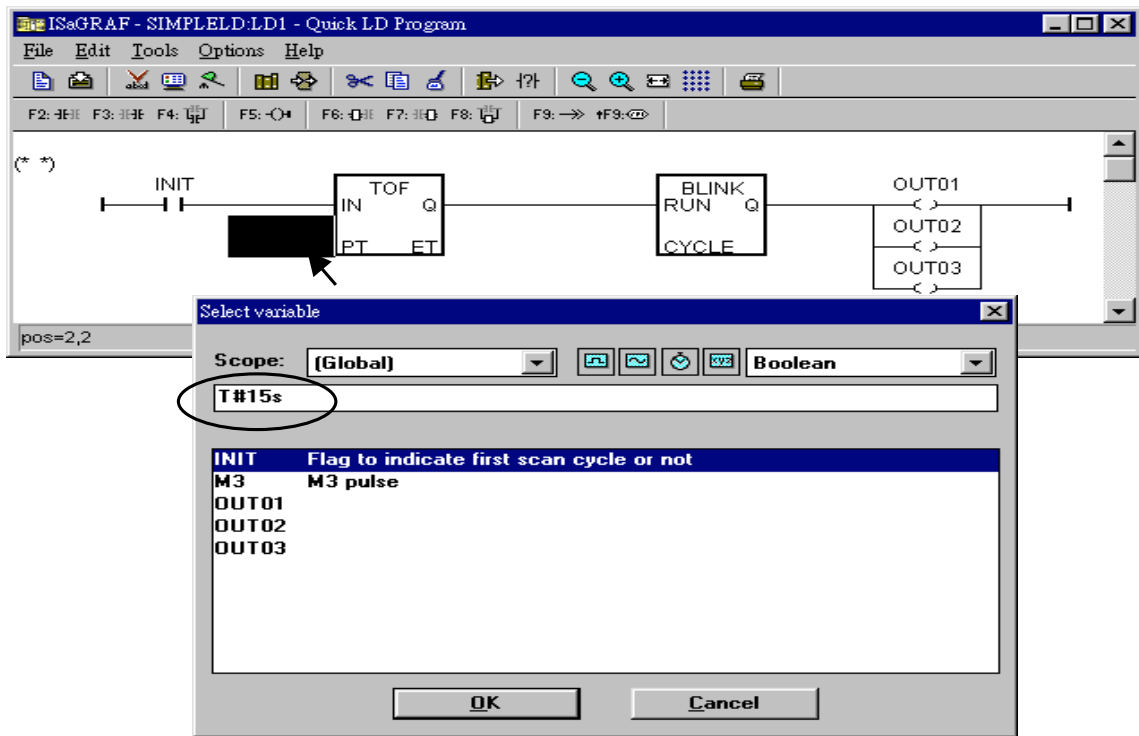
現在我們要開始指定變數或常數給每個元件. 用滑鼠雙擊左側的 Contact, 之後 “Select variable” 視窗會跑出來. 先選取 “Scope” 為 “(Global)”, 再選取 “Boolean”. 之後雙擊 “INIT” (你也可以用鍵盤輸入).



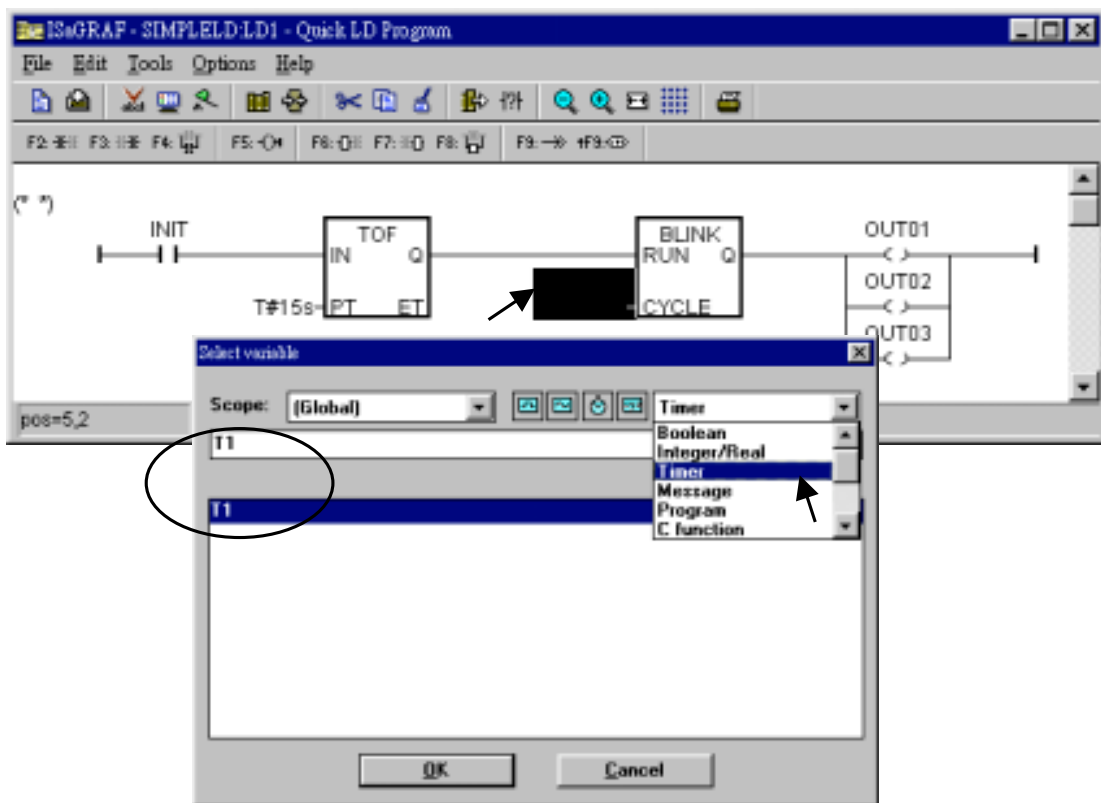
使用同樣的方法指定 OUT01~ OUT03 到個別對應的 Coil, 如下圖所示.



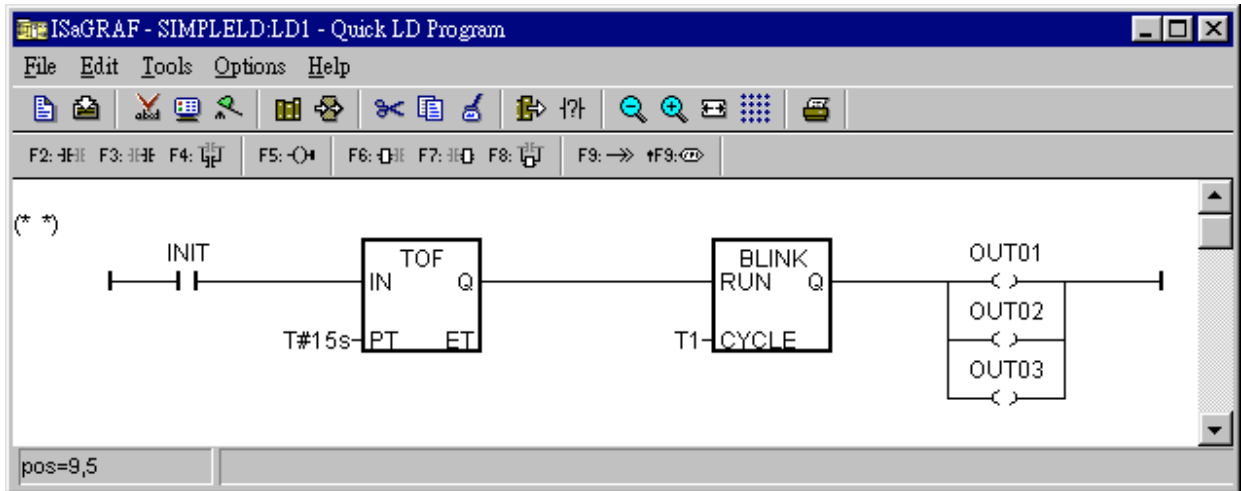
用滑鼠移動郵標到 ”TOF” 方塊 ”PT” 參數的左邊一格, 雙擊該位置, 用鍵盤輸入 T#15s (表示 15 秒), 然後按下 “OK”。



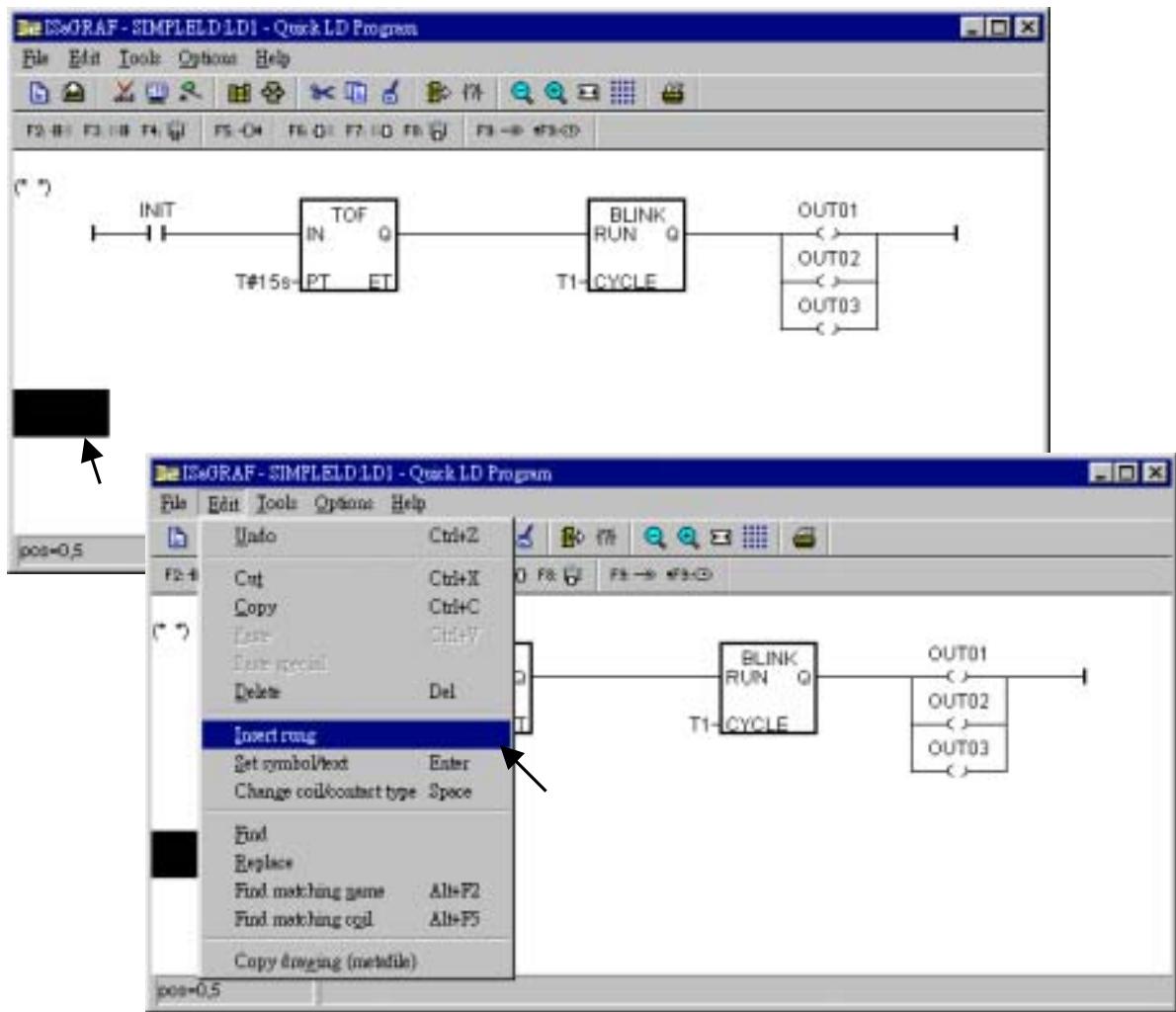
以同樣的方法指定 T1 給 “BLINK” 方塊的 “CYCLE” 參數。



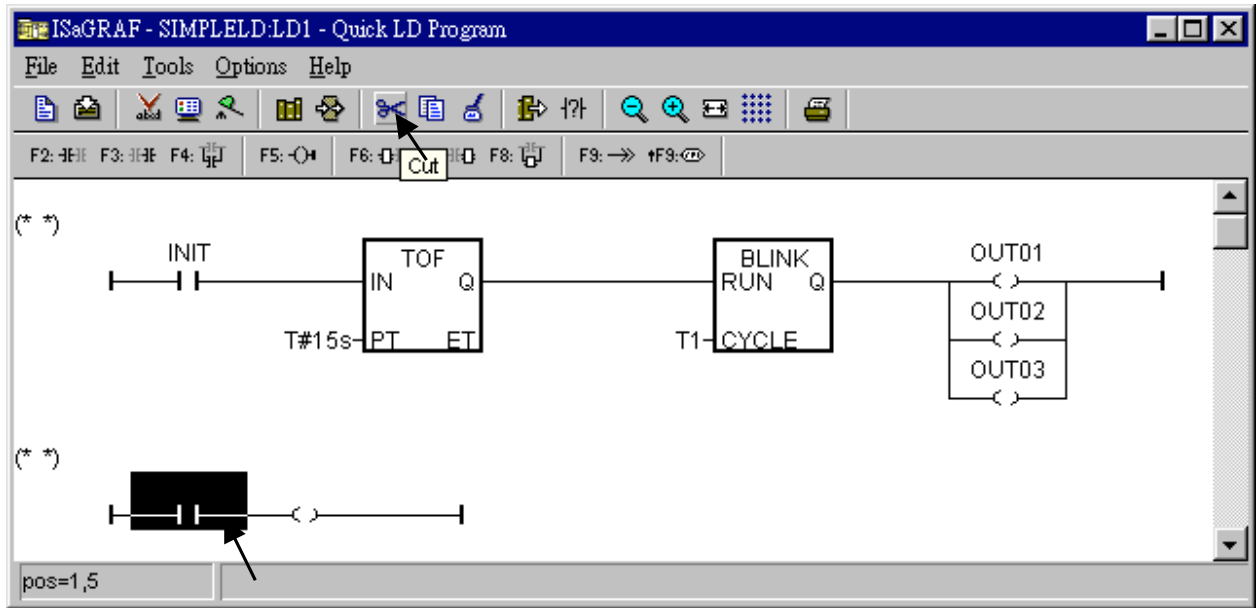
此時視窗會如下所示。



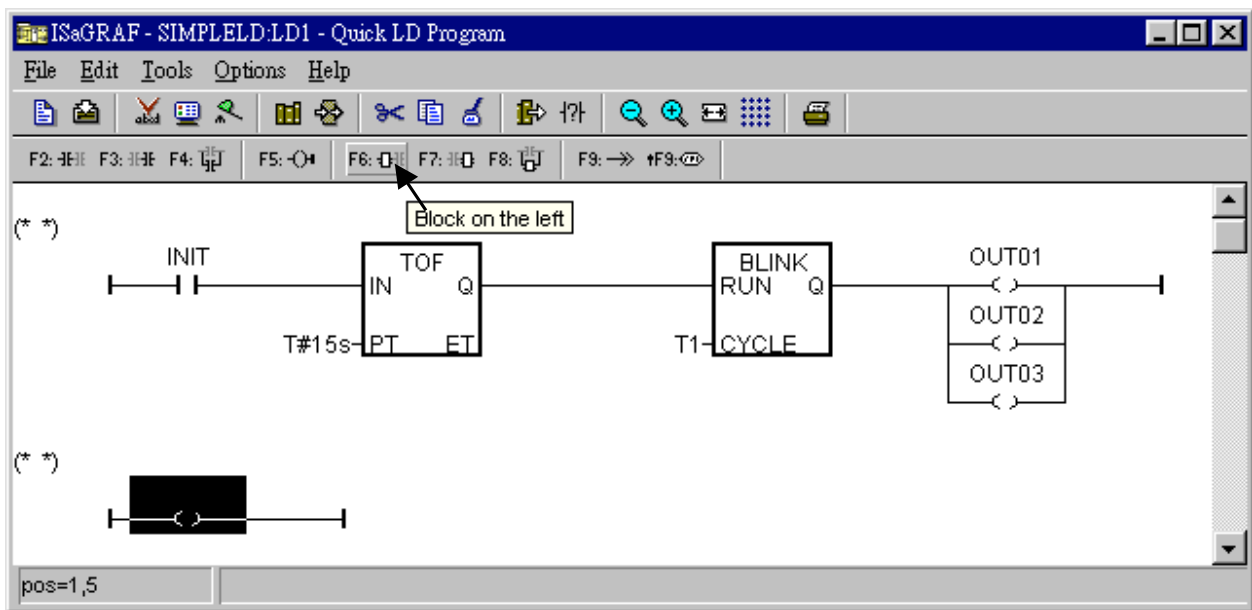
接下來要在下方新增一行, 先用滑鼠移動郵標到下方的適當位置, 之後點選“Edit – Insert rung”



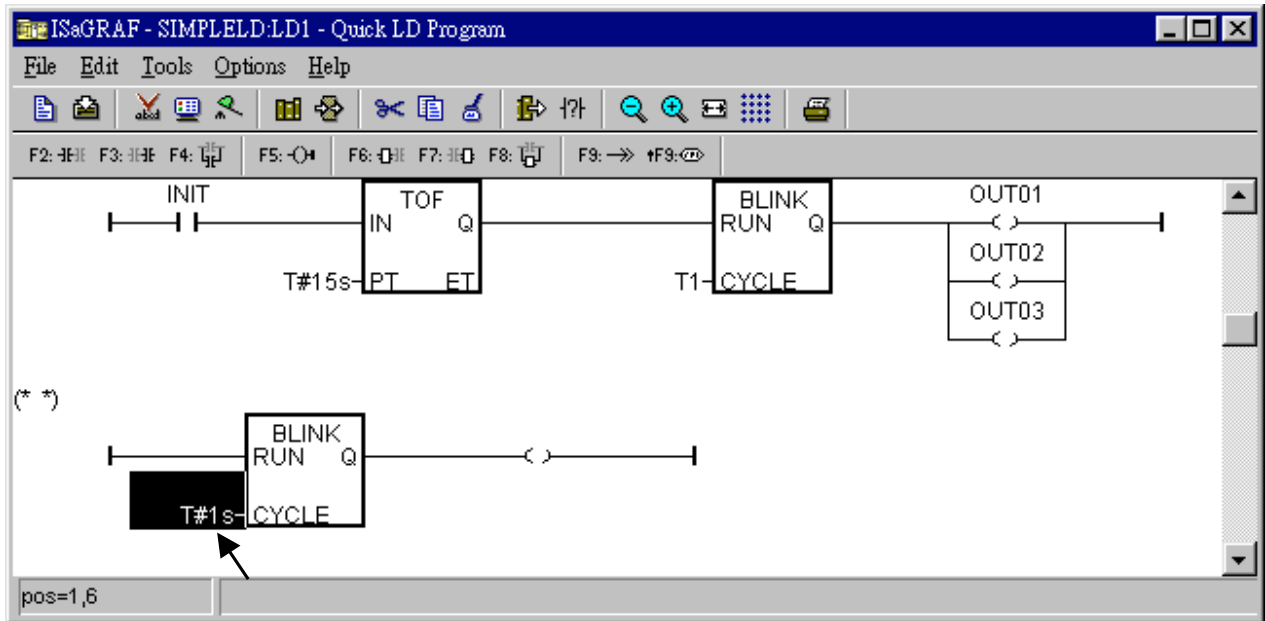
我們在這個範例內不須要左側的 Contact, 先用滑鼠點選它, 然後再點選 “Cut” 把它刪掉.



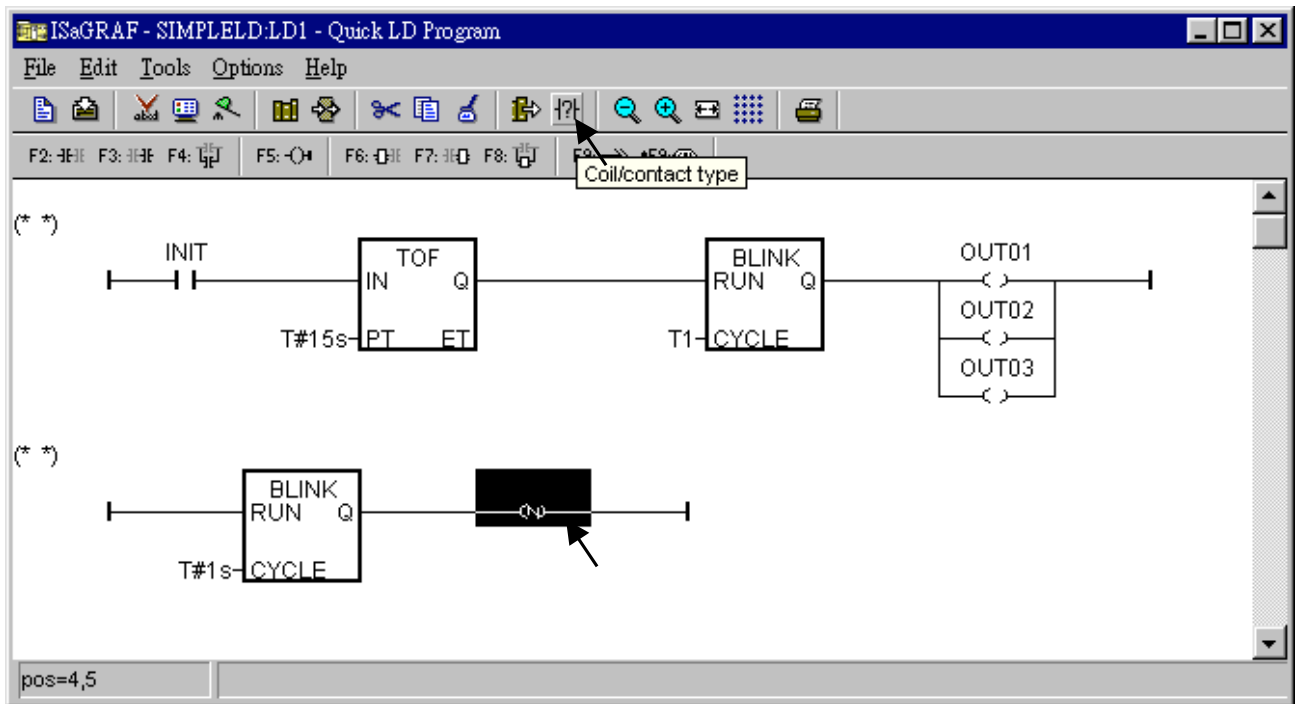
點選 “F6 (Block on the left)”, 之後雙擊新增的方塊, 並指定 “BLINK” 給該方塊.



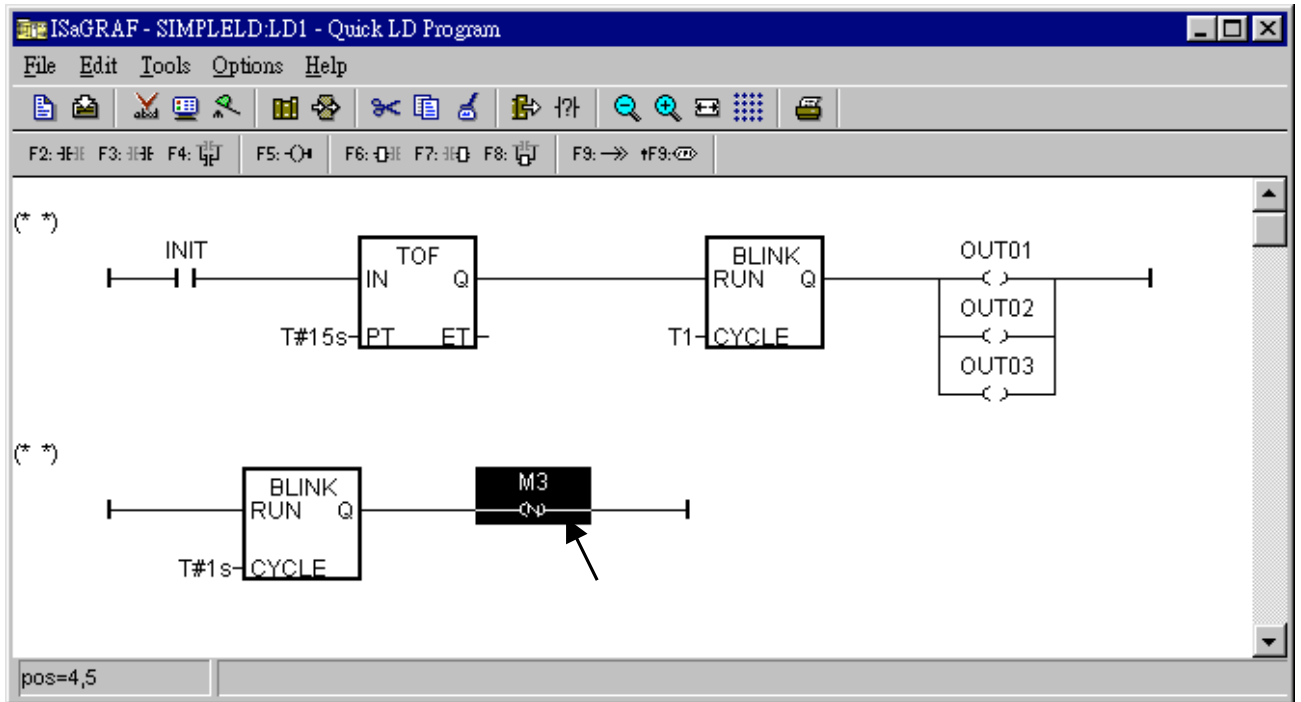
指定 T#1s 給“BLINK”方塊的“CYCLE”參數,之後會得到如下的視窗.



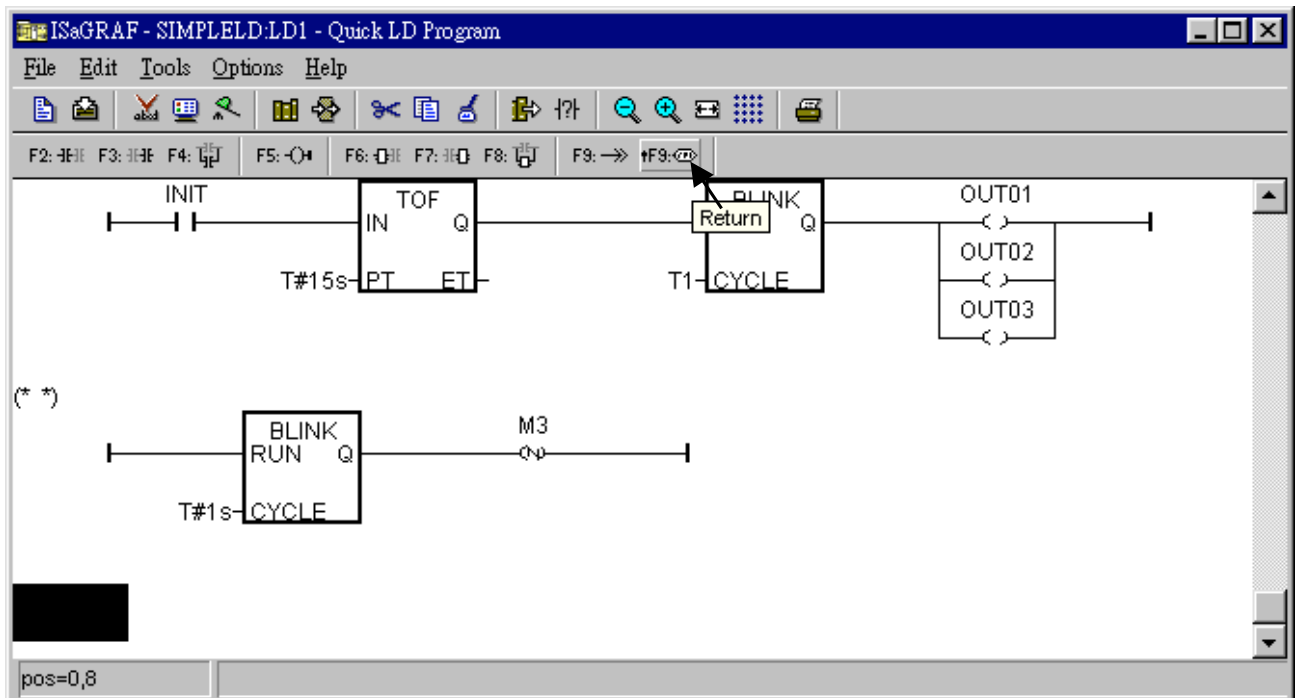
移動游標到右方的 Coil, 然後點選“Coil/contact type”幾次來切換 Coil 到 型態“N”.



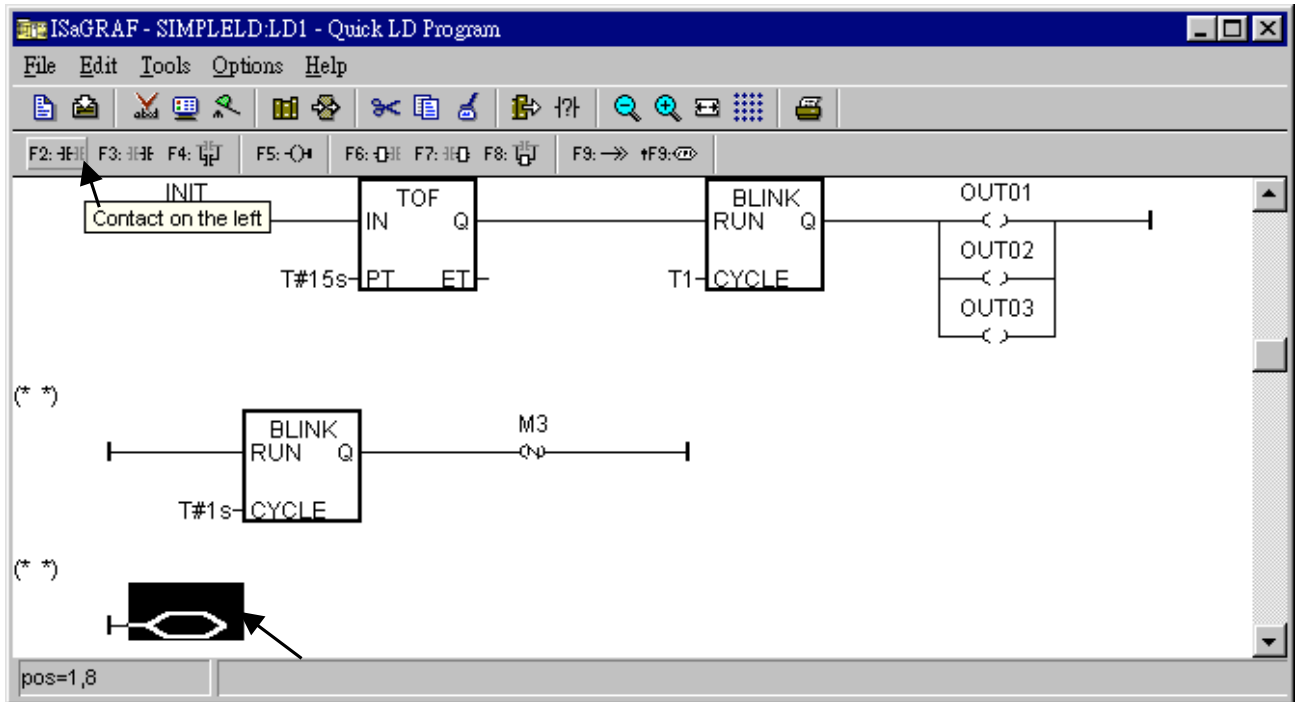
用滑鼠雙擊“N” Coil 以指定“M3” 給它。



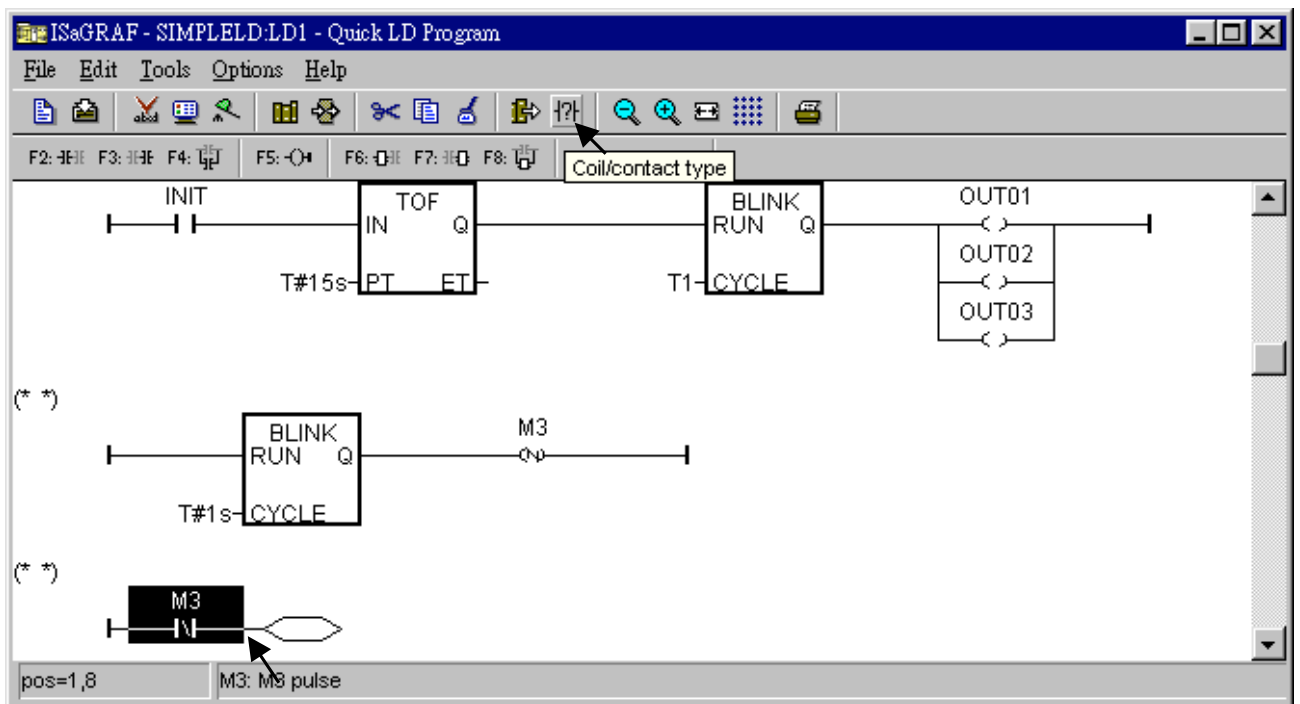
現在, 我們還要新增另一行 . 用滑鼠移動郵標到下方的適當位置, 然後點選“F9 (Return)”。



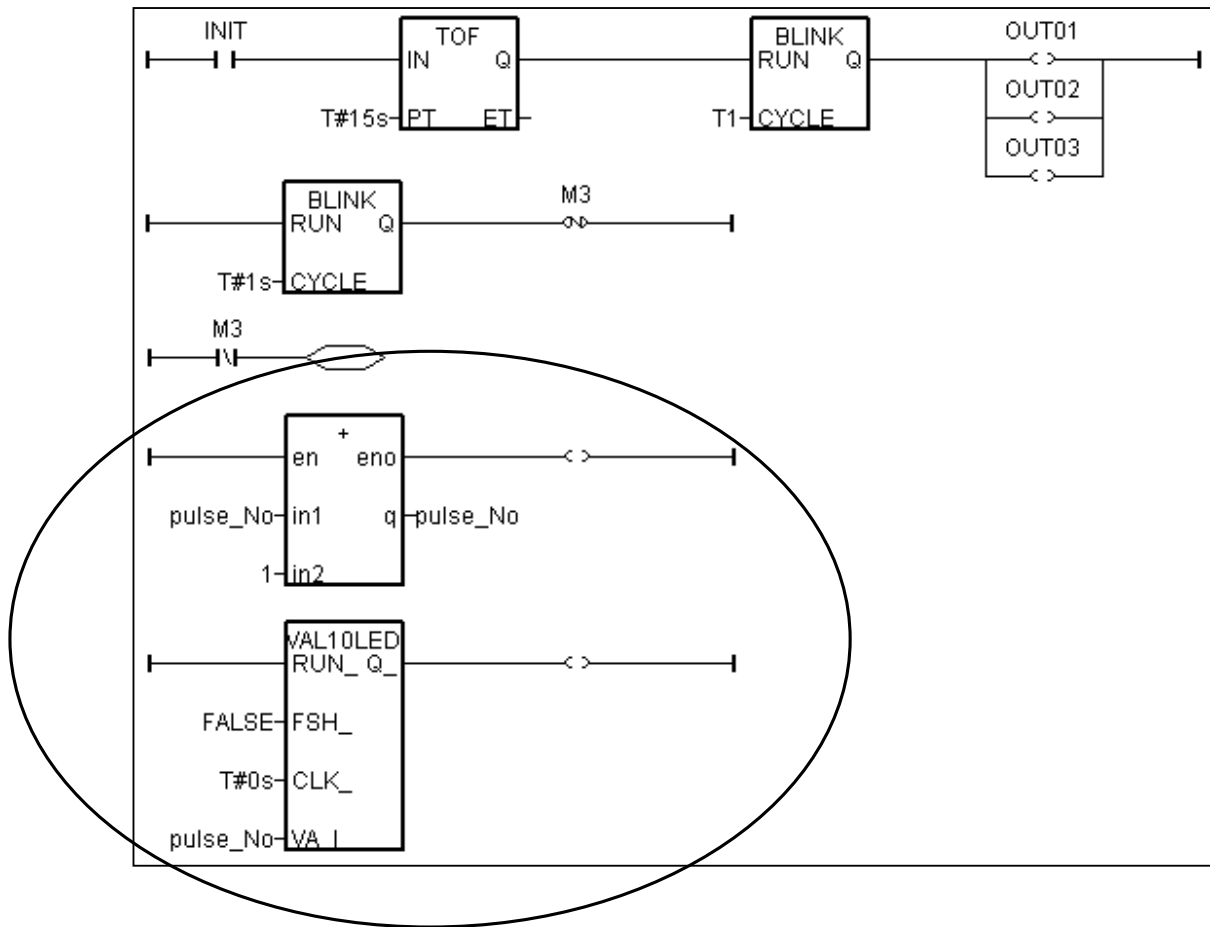
移動游標到“return”，之後點選“F2 (Contact on the left)” 以在左邊增加一個 Contact.



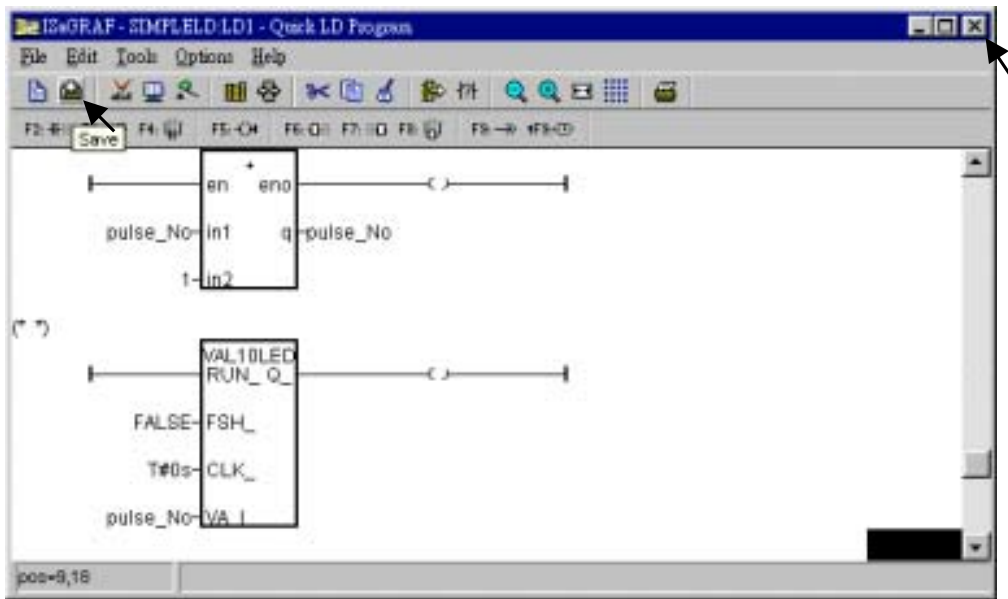
用滑鼠雙擊該 Contact, 之後指定“M3” 給它. 並更改它的型態為“\” (反相 Contact).



接下請自行依照先前描素的方法新增第 4 及第 5 行, 完成後的 LD 程式需如下所示.

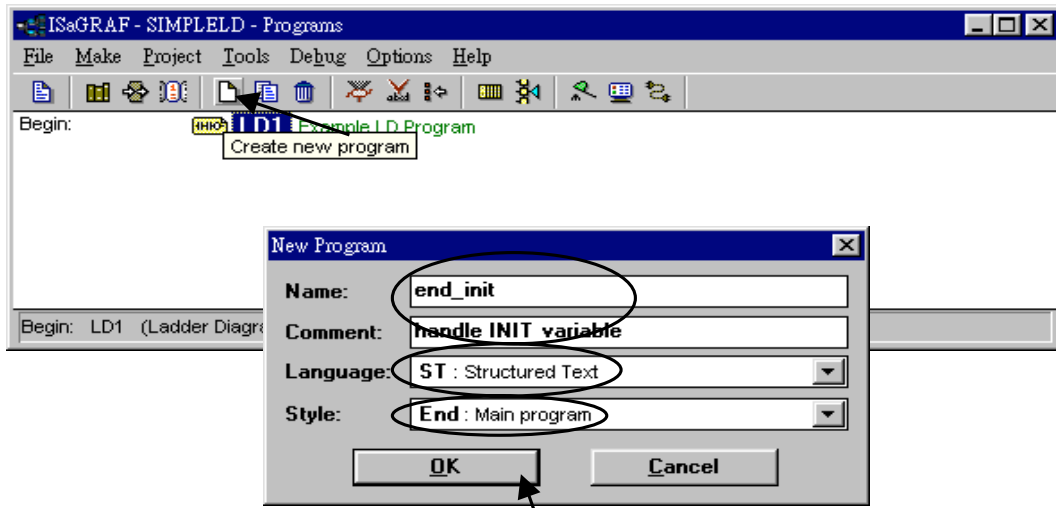


儲存這個 LD 程式, 之後關閉它.

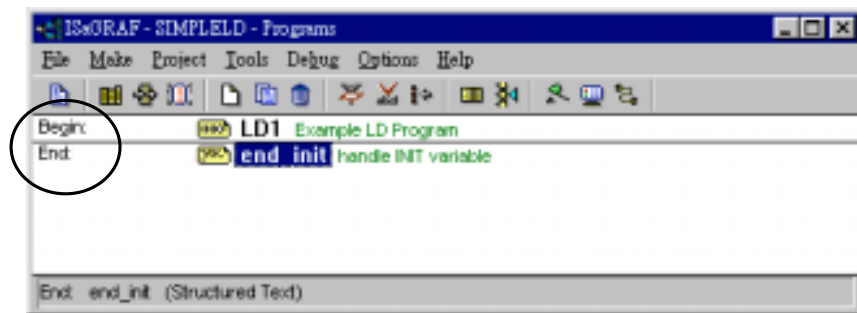


2.1.1.6: 建立 ST 程式 "end_init"

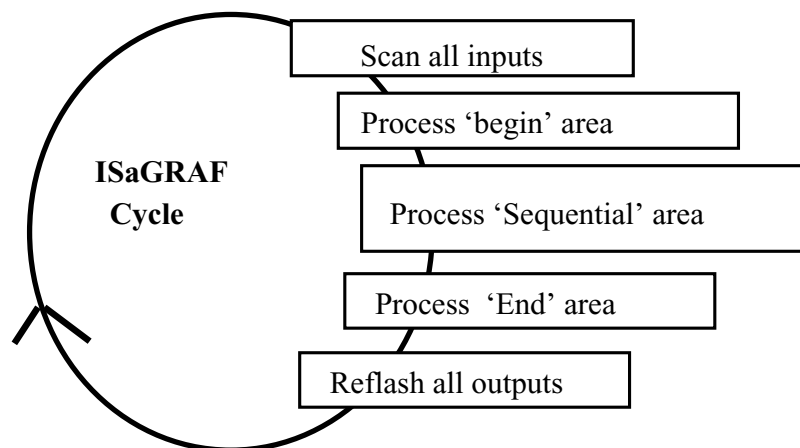
在這個專案內,我們需要一個 ST 程式來處理 INIT 變數,以滑鼠點選“Create new program”以建立一個新程式. 輸入程式名稱爲“end_init”, 注解爲“Handle INIT variable”, Language 爲“ST: structured Text”, Style 爲“End: Main program”. 之後點選“OK”.



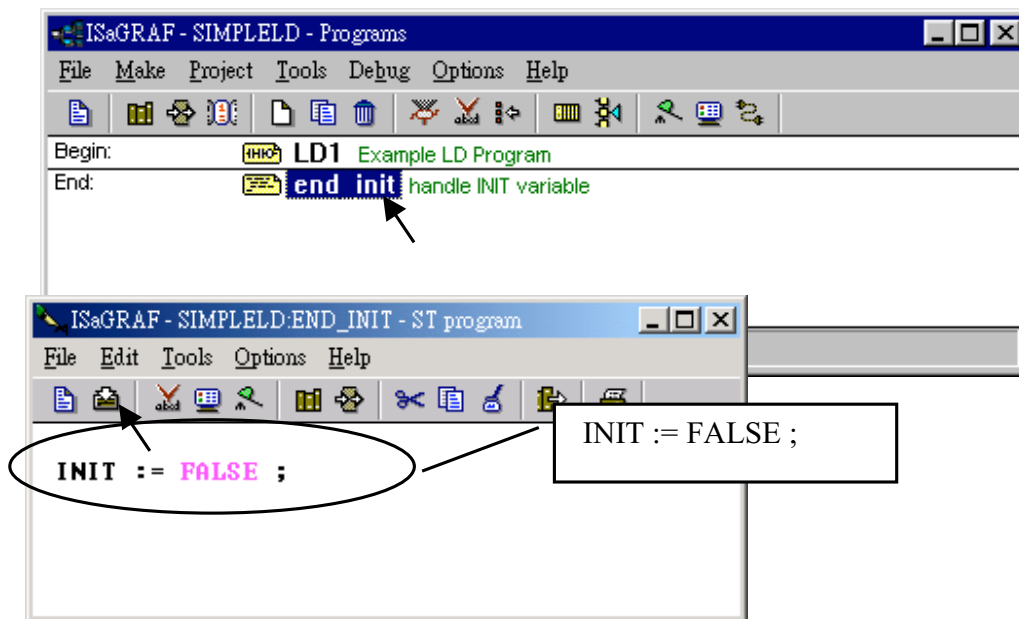
然後我們可以看到有 2 個程式在這個專案內.



ISaGRAF 每個 Scan 周期都會執行這兩個程式各 1 次. 其運作原則爲先執行“begin”區內的程式, 之後執行“Sequential”區, 最後再執行“End”區. 一個 ISaGRAF Scan 周期會如下圖的方式運作.



用滑鼠雙擊“end_init”程式. 用鍵盤編好 ST 程式後, 點選“save” 然後關閉它. 任何出現在(* 與 *) 內的文字為註解.

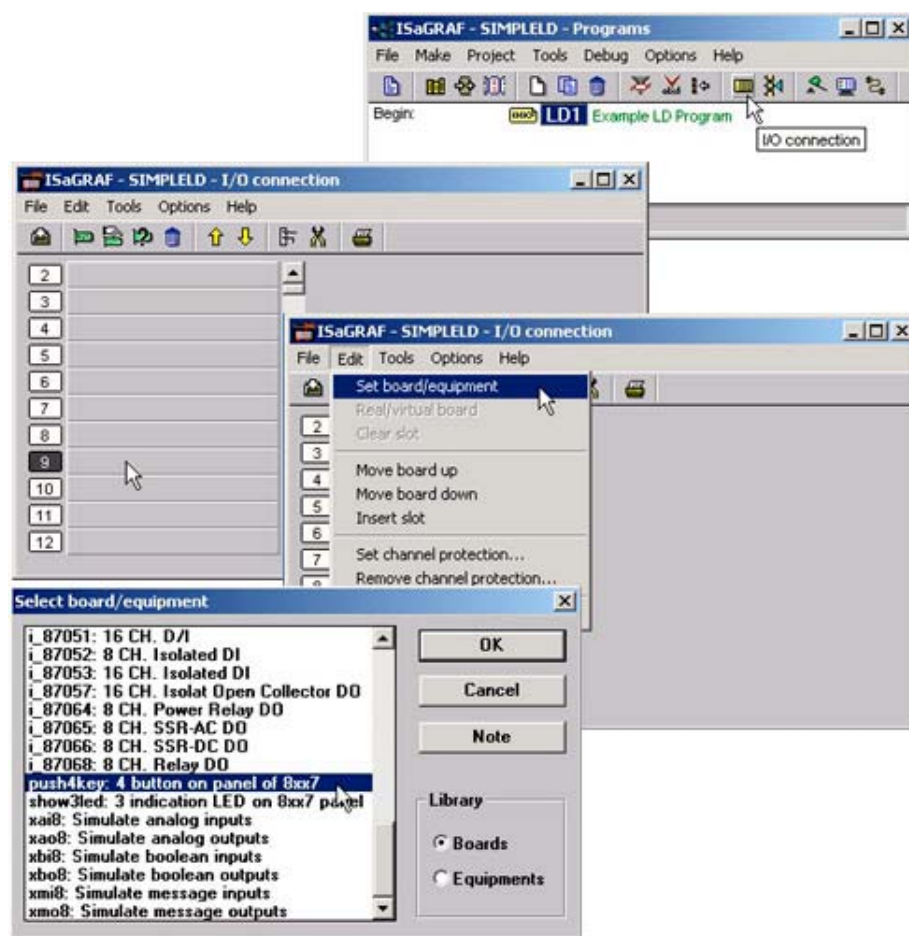


因為“INIT”被宣告成初值為“TRUE”, 這個 ST 程式會在第 1 個周期的尾端把“INIT”設為 FALSE 之後它的值就一直是 FALSE. 也就是說,“INIT”可用來判定目前程式是不是才剛開始執行 (TRUE: 第 1 個 Scan 周期, FALSE: 其它周期).

2.1.2: 連接輸入輸出(I/O)設備

由於 ISaGRAF 是一個開放式的軟體，很多控制器及 PLC 皆支援 ISaGRAF，然而，爲了要讓這些 PLC 控制器能夠使用 ISaGRAF 程式，首先這些 PLC 控制器的硬體製造商必須將 ISaGRAF 的驅動程式嵌入控制器中，ICP DAS I-8xx7 全系列的控制器皆將 ISaGRAF 驅動程式嵌入，使其成爲強大的、具高可塑性的工業用控制器。

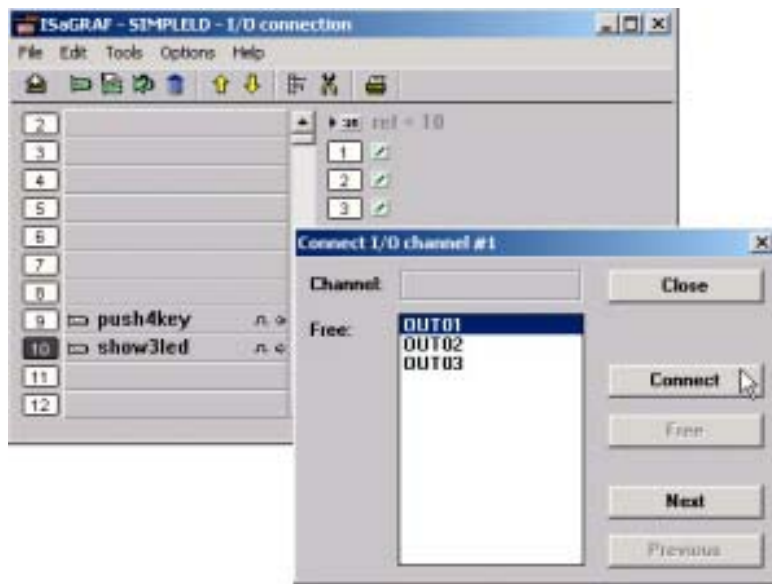
當您完成您的 ISaGRAF 程式後您必須將這個範例程式與 I-8xx7 I/O 控制系統連接，而 ICP DAS I-8xx7 控制系統有一項方便好用的特色可以讓您不需要實際插上 I/O 模板便可進行測試，I-8xx7 控制器上面板的 4 個按鍵可以作爲 4 個數位輸入，面板上方的 3 個 LED 燈號可作爲輸出的訊號。



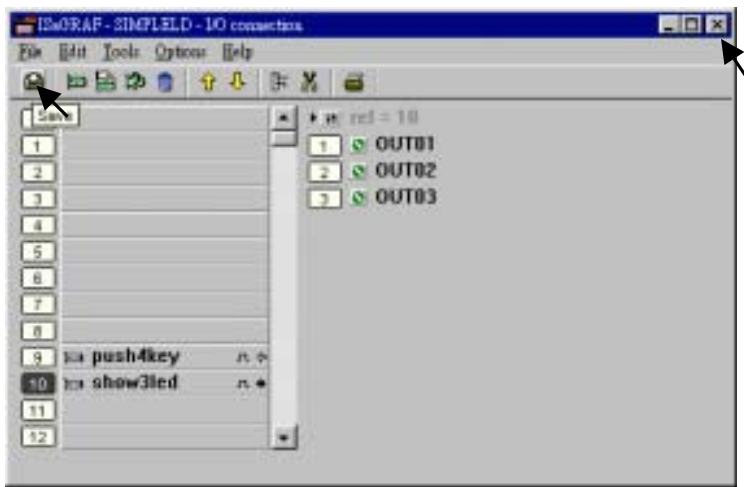
當您用滑鼠單擊"I/O Connection"圖像之後便會出現"I/O Connection"視窗，以本範例爲例，您可以滑鼠雙擊 9 號插槽，或者單擊 9 號插槽然後單擊[Edit]->[Set Board/Equipment]，也可以叫出"I/O Connection"視窗，點選"push4key:4 button on panel of 8xx7"連結以控制器面板上的 4 個按鍵作爲輸入的 I/O Board (本範例並未用到這 4 個按鍵)。

注意：編號 0 到 7 的 I/O 插槽是保留給實際的 I-8xx7 控制器 I/O 模板用的，您可以使用編號 8 以後的插槽連結虛擬 I/O 板。

爲了建立 I/O 連結您可以用滑鼠雙擊 10 號插槽並點選"Show3led: 3 indication LED on 8xx7 panel" 連結控制面板上的 3 個 LED 作爲輸出，此時您的"I/O Connection"視窗應如下圖所示。



當您完成 I/O 連結後記得點選"SAVE"來儲存。



注意:爲了能夠成功地編譯，所有輸入、輸出屬性的變數必須經由上述的方法進行 I/O 連結，而且，也只有輸出，輸入變數才會出現在"I/O Connections"視窗。

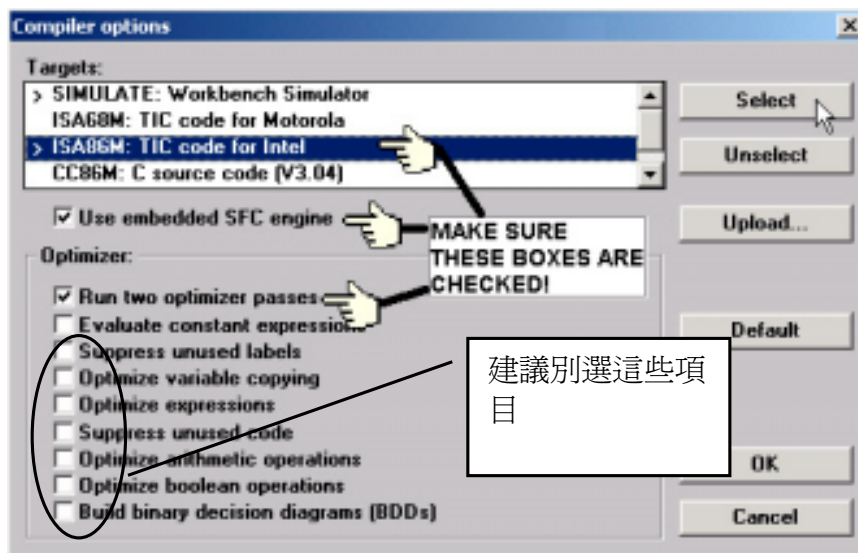
2.1.3: 編譯 LD 範例程式

爲了讓所有的 ISaGRAF 程式都能夠在 I-8xx7 控制器上正常的運作，程式設計師必須選擇正確的"Compiler Options"編譯選項，在 I-8xx7 控制器上，您必須選擇"ISA86ASM: TIC Code For Intel"選項，詳細步驟如下所示。

首先，用滑鼠在選單上點選"MAKE"選項，然後如下所示點選"Compiler Options"。



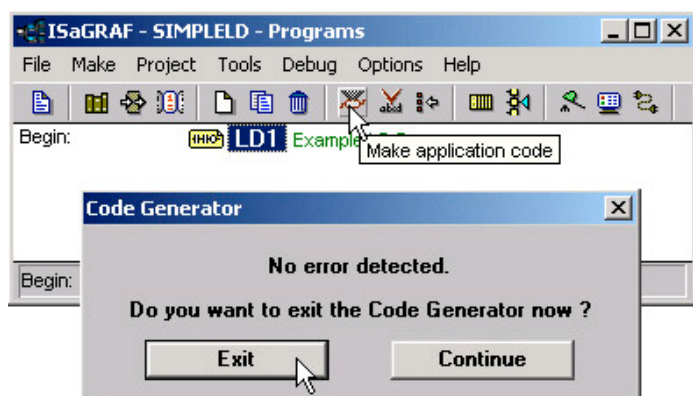
此時您會見到如下圖所示，請確認您正確地選擇如下圖所示的選項，並按下"OK"鍵完成所有的選擇。



編譯時的訊息處理!

當您點選"Make Application Code"圖示後便開始編譯此 LD 範例程式，如果編譯過程沒有出現任何錯誤訊息，恭喜您，您已經成功地完成了一個 LD 範例程式。

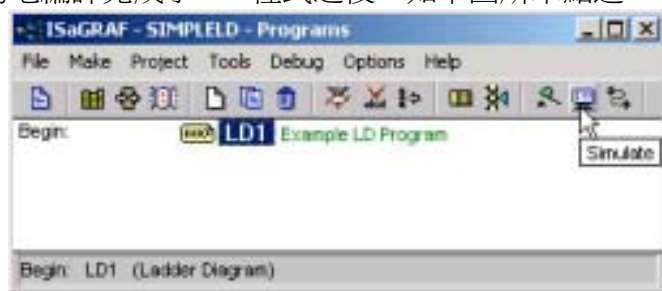
如果您在編譯過程中程式出現錯誤訊息，請點選"CONTINUE"按鍵檢視錯誤訊息代表的意義，並回到專案編輯畫面更正錯誤，並重新編譯。



2.1.4: 模擬 LD 專案

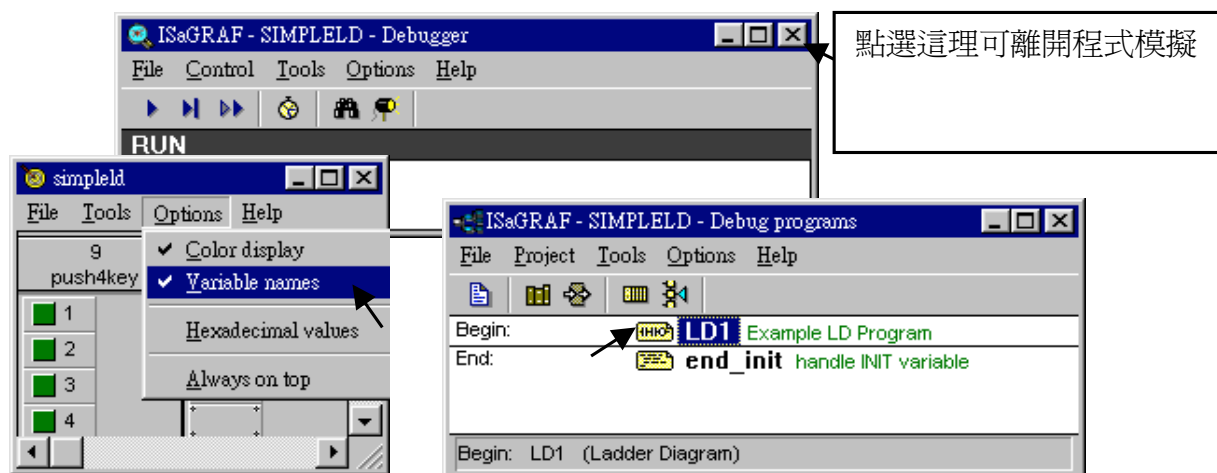
ISaGRAF 其中一項強大的功能在於它強大的除錯功能，在您還沒實際將專案下載至 I-8xx7 控制系統中，您可以利用"SIMULATE"(模擬)的功能進行除錯。

在成功地編譯完成了 LD 程式之後，如下圖所示點選"SIMULATE"圖示。



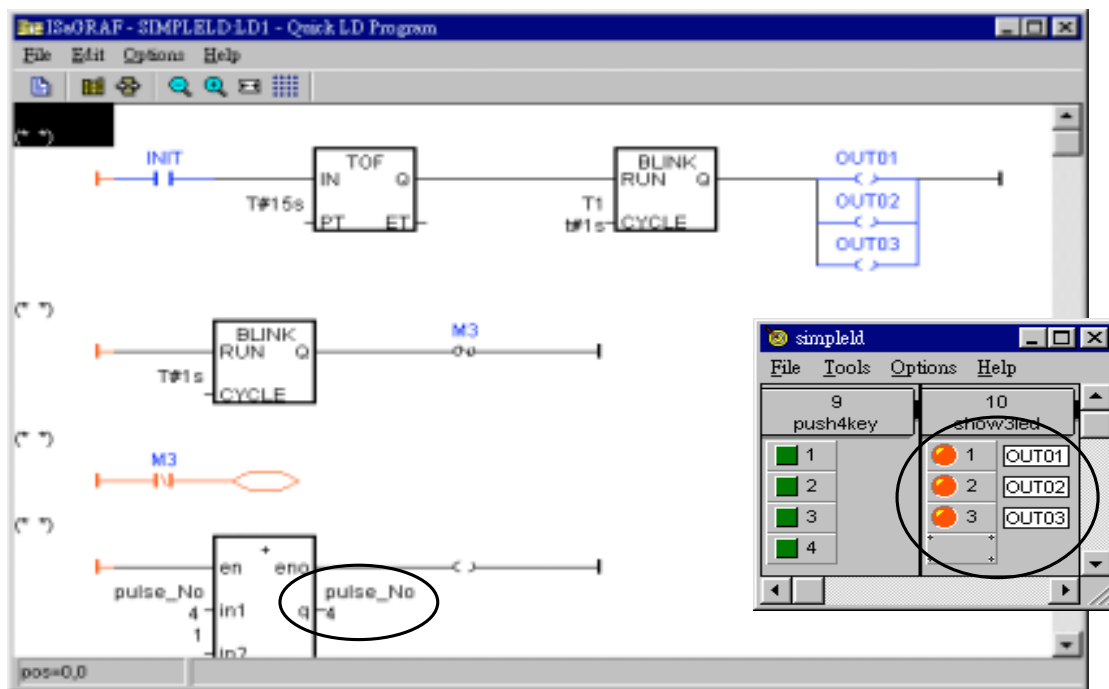
當您點選"Simulate"圖示後會出現 3 個視窗，分別是"ISaGRAF Debugger"、"ISaGRAF Debug Programs"和"I/O Simulator"三個視窗，如果您建立的 I/O 變數名稱沒有出現在"I/O Simulator"視窗，只需要點選"Options"和"Variable Names"選項便可以在"I/O Simulator"視窗中顯示變數名稱。

如下圖所示，在"ISaGRAF Debug Program"視窗中雙擊"LD1"，可以打開 ISaGRAF 的"Quick LD Program"視窗，並看到您先前所設計的程式。



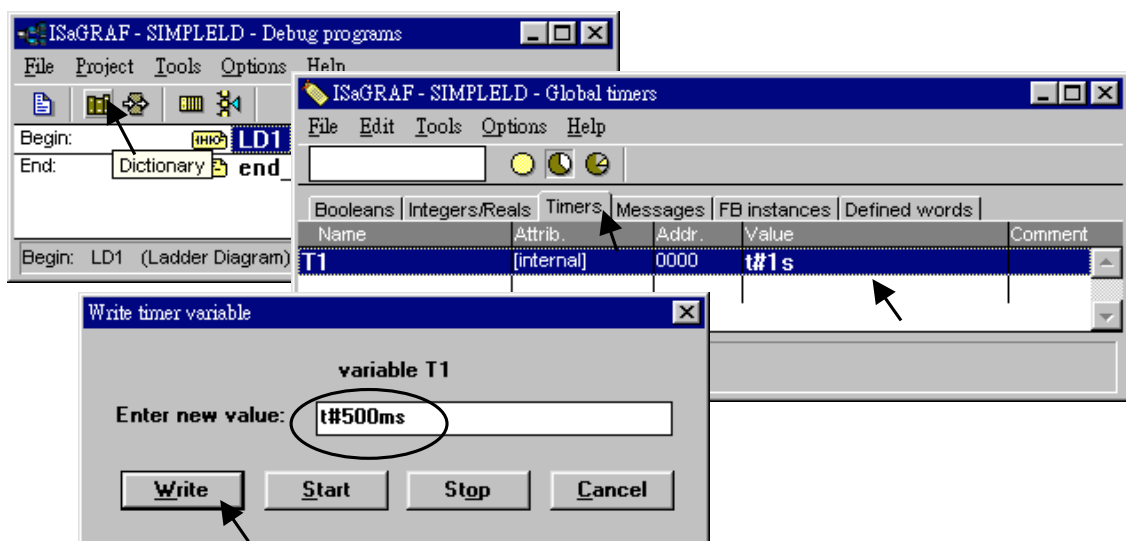
執行模擬程式

當您在"ISaGRAF Debug Programs"視窗中用滑鼠雙擊"LD1"後會見到如下畫面。



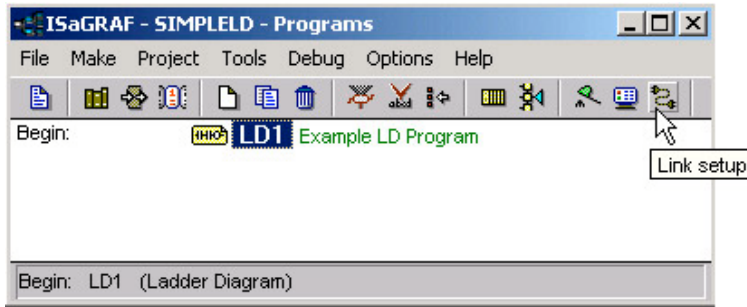
你會看到 OUT01 ~ OUT03 在程式開始執行的前 15 秒會以 1 秒為周期閃爍. 而且 pulse_No 的值會每秒加 1.

您可以在程式進行時調整 T1 計時器變數的值，首先，在 ISaGRAF 視窗中點選"Dictionary"圖示在"ISaGRAF Global Variables"視窗中點選"Timers"標籤，然後以滑鼠雙擊"T1"變數名稱，此時會開啓"Write Timer Variable"視窗，您可以在"Enter New Value:"欄位中將"t#1s"改成"t#500ms"並按下"Write"鈕。

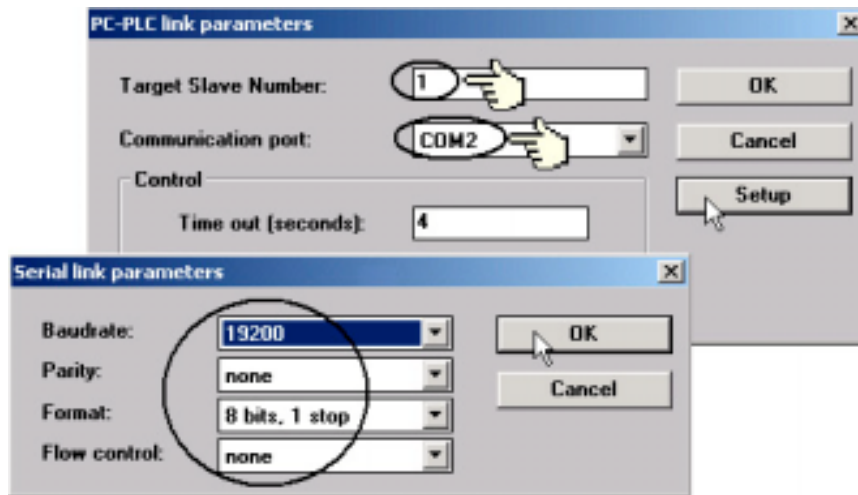


2.1.5: 下載及除錯 LD 範例程式

最終，您可將此 LD 範例程式下載至 I-8xx7 控制系統上運作，在下載之前您必須在 I-8xx7 與 PC 間建立通訊。



首先，請您用滑鼠在"ISaGRAF Programs"視窗中點選"Link Setup"圖示，當您點選"Link Setup"圖示後如下圖所示的視窗將會出現。



其中"Target Slave Number"是第 1 章 1.3.1 節 I-8xx7 控制系統的 Net-ID，Net-ID 的位置在 I-8xx7 控制器的右下方。Net-ID 是設定您控制器的編號，本範例您的控制器是設為 1，而“Communication Port”是指通訊線連結到你電腦所連接的通訊埠，通常不是 Com1 就是 Com2。

I-8xx7 的通訊參數和您電腦上的通訊參數必需設定為一樣的。在 I-8xx7 控制器的通訊埠 Com1(RS232)及 Com2(RS485)的預設參數如下：

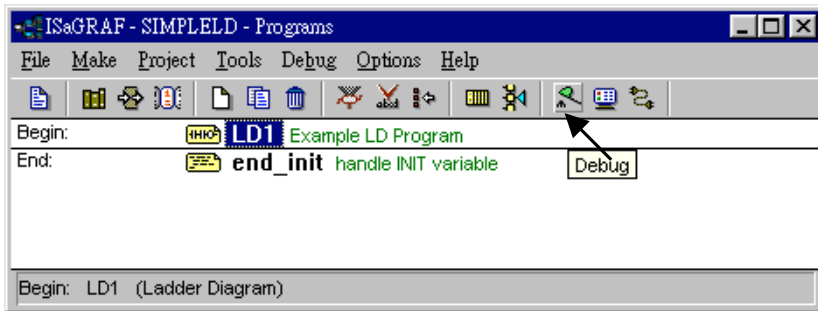
Baudrate:	19200
Parity:	none
Format:	8 bits, 1 stop
Flow control:	none

注意：

您必需確定電腦上的通訊埠和 I-8xx7 是否能夠正常通訊，若不行則必需修改電腦通訊埠的設定讓電腦和 I-8xx7 能夠順利溝通。

下載 LD 範例程式

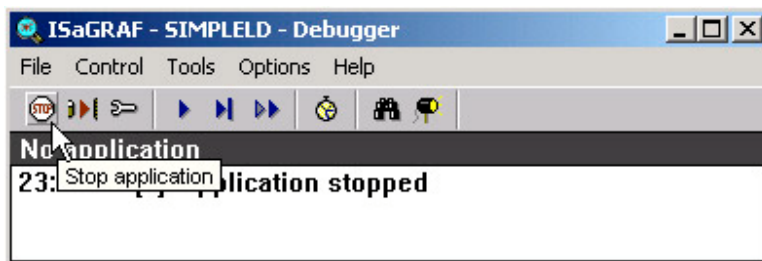
在您下載 LD 範例程式到 I-8xx7 控制器之前,您必需先確認電腦和控制器之間的連線是否正常,若正常無誤,按下 ISaGRAF 程式視窗中的 Debug 鍵,開啓如下之畫面:



若出現程式的名字和“Run”這兩個字,則表示控制器中有一個應用程式在跑。若在 ISaGRAF Debugger 視窗中出現“Disconnect”則表示電腦和控制器並沒有連線成功。

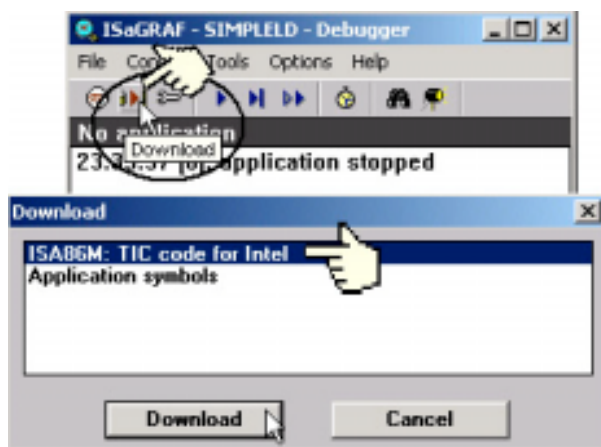
最常導致連結失敗的原因是彼此之間的通訊埠沒有插好,或是彼此間的通訊參數未設正確。這時您必需重插電腦上的通訊線到正確的通訊埠或是重設 ISaGRAF 程式的連結設定。

若原先控制器內已有應用程式在執行,需先停掉該應用程式之後才能下載新的應用程式,方法為在“ISaGRAF Debugger”視窗下選取 STOP 鍵停止應用程式,此時便可重新下載應用程式至控制器。

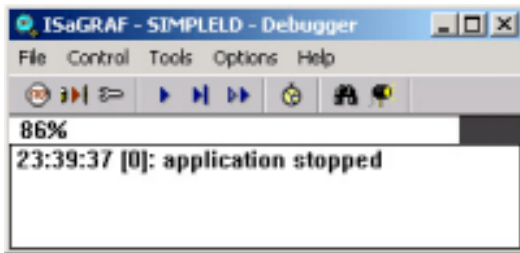


開始下載程式

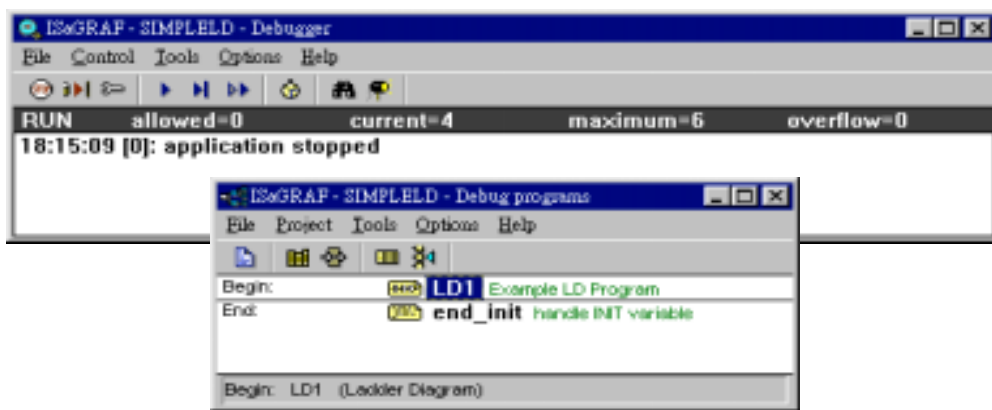
從“ISaGRAF Debugger”視窗內按下“Download”鍵,接著選擇“ISA86M:TIC Code For Intel”



將會開始下載 LD 範例程式至 I-8xx7 控制器內，而在“ISaGRAF Debugger”視窗中會出現程式下載進度的畫面。



當 LD 程式下載完成後，將會出現下面兩個視窗



監視 LD 範例程式

當您在監視 LD 範例程式時，透過 ISaGRAF 視窗畫面您可以觀察即時的 I/O 狀態。在 ISaGRAF Debugger 視窗中按下“I/O connections”鍵開啓“I/O connection”，透過“I/O Connections”視窗會顯示所有的輸出及輸入點的狀態。另外一個有用的視窗即“Quick LD program”，經由這個視窗可以即時的看到 LD 程式的執行狀態。

