

常用的 COM Port 函式技巧 來讀寫 RS-232 / 422 / 485

ICP DAS ISaGRAF 控制器支持的 Serial COM Port (RS-232/422/485) 通訊規約有以下幾種:

Modbus RTU Slave	參考 ISaGRAF 手冊第 4 章 與 各別產品的規格 或 快速上手手冊
I-7000 與 I-87xxx 遠程 I/O	參考 ISaGRAF 手冊第 6 章
Modbus RTU Master (M-7000)	參考 ISaGRAF 手冊第 8 章 與 第 21 章
Modbus ASCII Master	參考 ISaGRAF 手冊第 8 章
Modem Link	參考 ISaGRAF 手冊第 13 章
MMICON	參考 ISaGRAF 手冊第 16 章
SMS : 收發手機簡訊	參考 ISaGRAF 手冊第 17 章

若有應用需要使用其它通訊規約, 使用者可以使用以下的 Serial COM Port 函式來自行操作.
(函式的詳細說明請參考 ISaGRAF 手冊內附錄 A.4)

COMOPEN	開啓 Serial COM Port (不含 Flow control 參數)
COMOPEN2	開啓 Serial COM Port (含 Flow control 參數, 不適用 I-8xx7)
COMREADY	偵測有無通訊資料進來
COMARY_R	讀取目前已經收進來的通訊資料,並存入 Byte 陣列內
COMARY_W	寫出 Byte 陣列內的多個 Byte 到某個 COM Port
COMREAD	讀取一個 Byte (需先使用 COMREADY 偵測, 確定有資料才可以收)
COMCLEAR	清空已經收到的資料 Buffer
COMARY_NW	寫出 1 個長整數資料到 COM Port, 格式為 Binary, 4 個 Byte
COMARY_WW	寫出 1 個 Word 資料到 COM Port, 格式為 Binary, 2 個 Byte
COMSTR_W	寫出 1 個字串到 COM Port
COMWRITE	寫出 1 個 Byte 到 COM Port
COMCLOSE	關閉 Serial COM Port

注意:

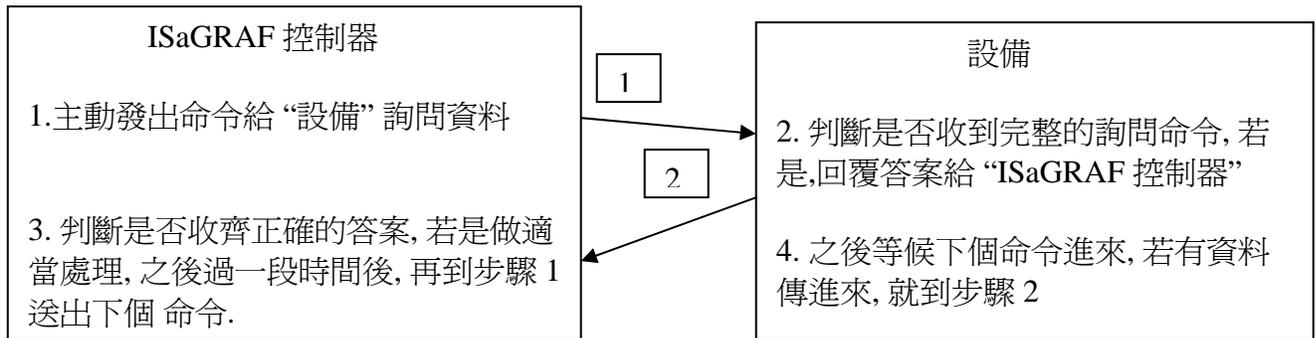
1. I-8xx7 出廠時,它的 COM1 與 COM2(I-8417/8817 才有 COM2) 為 Modbus RTU Slave Port.其中 COM1 可以關閉 Modbus RTU Slave 功能,改成可以使用上表內的函式來自行操作(參考附錄 C.1). 若要使用 I-8xx7 的 COM5 到 COM20,需使用擴充的 I-8112/8114/8142/8144 板卡 (參考第 1.8 節)
2. W-8xx7 / 8xx6 的 COM2 / COM3 可以啟動成為 Modbus RTU Slave Port, 若要自行使用, 請關閉該 Modbus RTU Slave 功能. (請參閱它的包裝盒內附的 “Getting Started:Wincon ISaGRAF PAC” 內的 Appendix A.2). 若要使用 Wincon 上的 COM5 到 COM14, 需使用擴充的 I-8112/8114/8142/8144 板卡 (請參考 “Getting Started:Wincon ISaGRAF PAC” 內的 Appendix E)
3. I-7188EG / 7186EG 的 COM1, 出廠時 內定為 Modbus RTU Slave 功能, 可以關閉該功能,改使用上表的函式. (請參閱它的包裝盒內附的 “快速上手手冊” 內的第 3.6 節). I-7188XG 的 COM1 則永遠支持 Modbus RTU Slave, 無法關閉. 若要使用 I-7188EG/XG, 7186EG 的 COM3 到 COM8, 需選購一片 X5xx 子板, 拔開外殼插在裏面的擴充槽, 之後才能使用

http://www.icpdas.com/products/PAC/i-o_expansion/x_list.htm

以下幾節針對幾個常用的通訊方式分別說明

1. 主動對設備 一問一答 方式

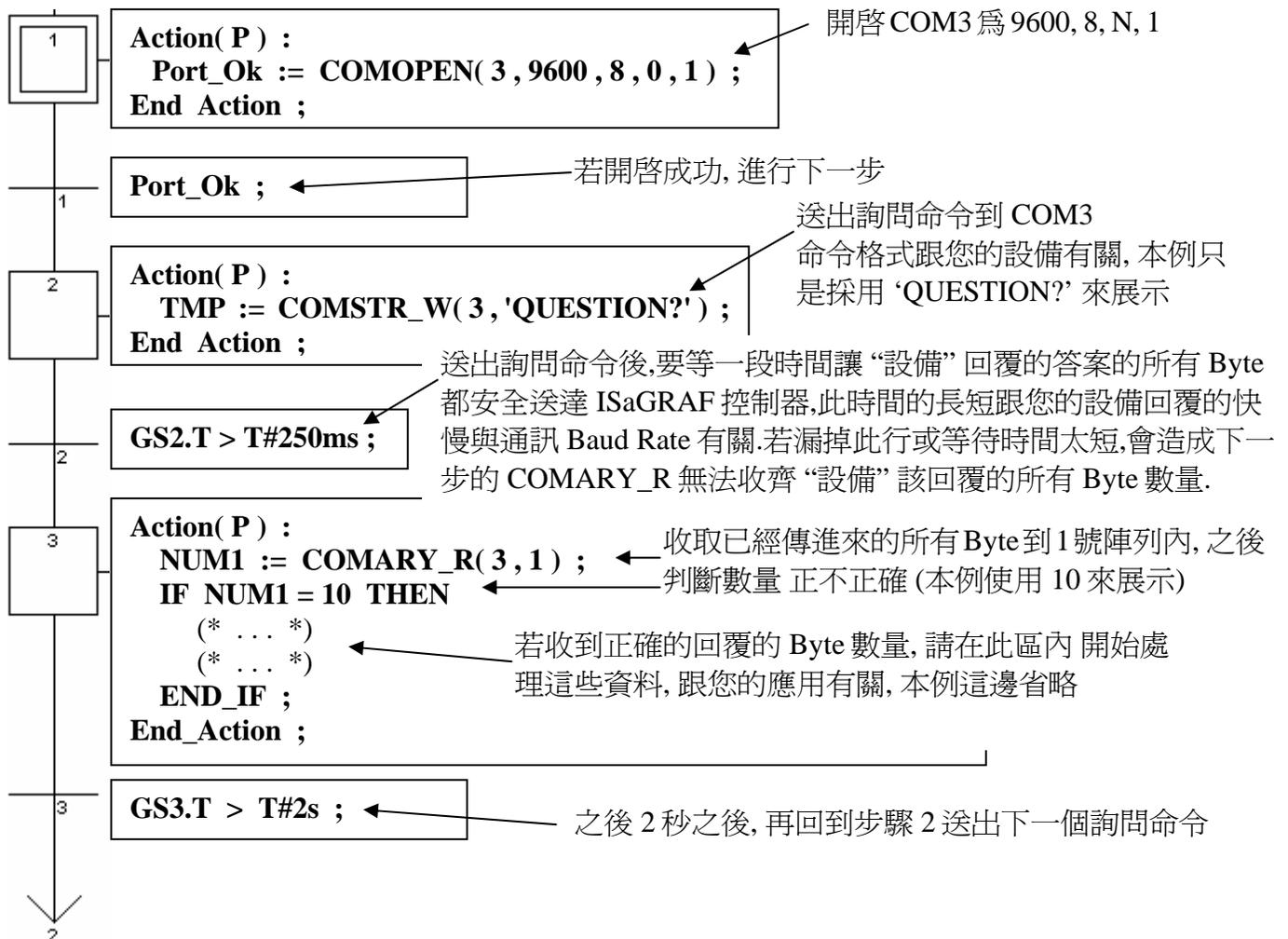
若有應用是要使用 RS-232 / 422 / 485 來取得其它設備的資料, 且它的規約是一問一答方式, 如下



可以使用類似以下的程式來處理.

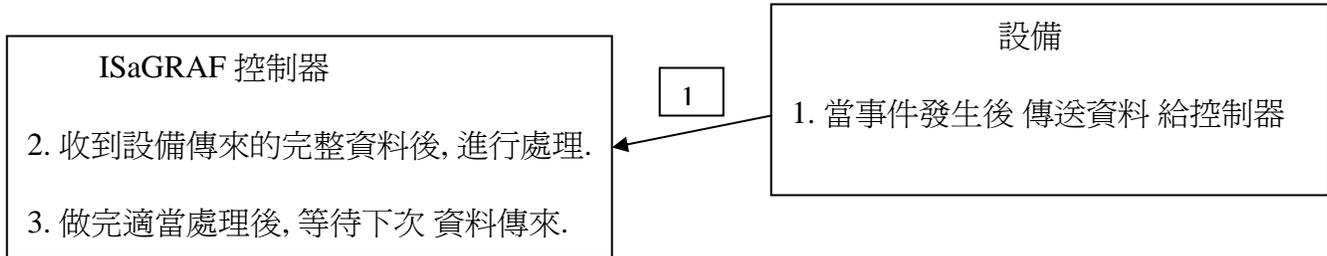
本例先由“ISaGRAF 控制器”由 COM3 送出一個字串‘QUESTION?’ 給“設備”, 之後就準備收取“設備”回覆的答案, 之後 就再過 2 秒, 再送出 同個命令‘QUESTION?’, ... 如此重覆進行.

使用 SFC 語法: (本例 Port_OK 與 TMP 宣告為 Boolean Internal , NUM1 宣告為 Integer Internal)



2. 被動等待設備傳資料過來方式

此種方式在一般的商場或便利商店很常見, 比如像條碼閱讀機這種設備, 當讀到商品的條碼後, 它會傳送條碼資料到ISaGRAF 控制器的COM Port (RS-232 / 422 / 485), 而且不需要 ISaGRAF 控制器回覆任何訊息給它。(請參訪 www.icpdas.com – FAQ – Software – ISaGRAF – 066 來取得 demo program 與更多的說明)



使用 ST 語法:

```

IF INIT THEN
  INIT := FALSE ;
  Port_Ok := COMOPEN( 3 , 9600 , 8 , 0 , 1 ) ;
  T1 := T#0s ;
  STEP := 0 ;
END_IF ;
  
```

本例請宣告:

INIT 為 Boolean Internal, 設初值為 TRUE
 Port_Ok 與 TMP 為 Boolean Internal
 T1 為 Timer Internal (計時器)
 STEP 與 NUM1 為 Integer Internal

開啓 COM3 為 9600, 8, N,1

```

IF Port_Ok = False THEN
  Return ;
END_IF ;
  
```

若開啓失敗, 以下的 Code 也不用做了, 直接跳出本 ST 程式

CASE STEP OF

```

0 : IF COMREADY(3) THEN
  STEP := 1 ;
  T1 := T#0s ;
  Tstart(T1) ;
END_IF ;
  
```

STEP 為 0 表示等待中, 要去檢查是否有資料傳到 COM3
 若 COMREADY 回傳 TRUE, 表示至少有 1 個 byte 傳進來,
 則將 STEP 設成 1, 並開始啓用 T1 計時器來計時.

```

1 : IF T1 > T#250ms THEN
  Tstop(T1) ;
  T1 := T#0s ;
  STEP := 0 ;
  NUM1 := COMARY_R(3, 1) ;
  
```

STEP 為 1 表示資料正在傳進來, 要等 T1 計時到一段時間
 過後, 再用 COMARY_R 去收所有 Byte 到 1 號 Byte 陣列內
 (這樣才能確保“設備”傳過來的所有 Byte 數量 都已經傳
 送過來了). 此等待時間跟“設備”的規格與 Baud Rate 有
 關, 本例使用 0.25 秒來展示. (若設的太短有可能會造成有
 些 Byte 還沒傳送過來而漏掉它們). 要記得將 STEP 設回 0
 來等待下次資料再傳進來.

```

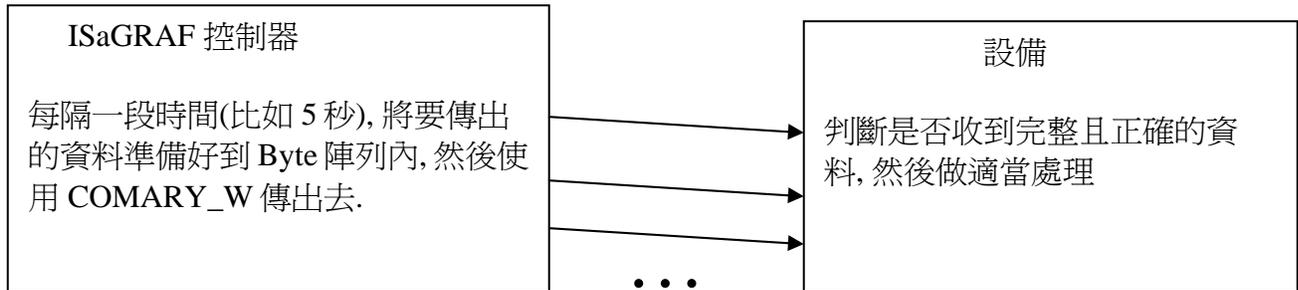
  IF NUM1=10 THEN
    (* ... *)
  END_IF ;
END_IF ;
  
```

收到資料後, 判斷資料對不對, 再做適當處理.
 本例只判斷 Byte 數量是否為 10 個. 實際的情況需依造您的應用
 來修改. (本展示例 省略“適當處理”的 Code)

END_CASE ;

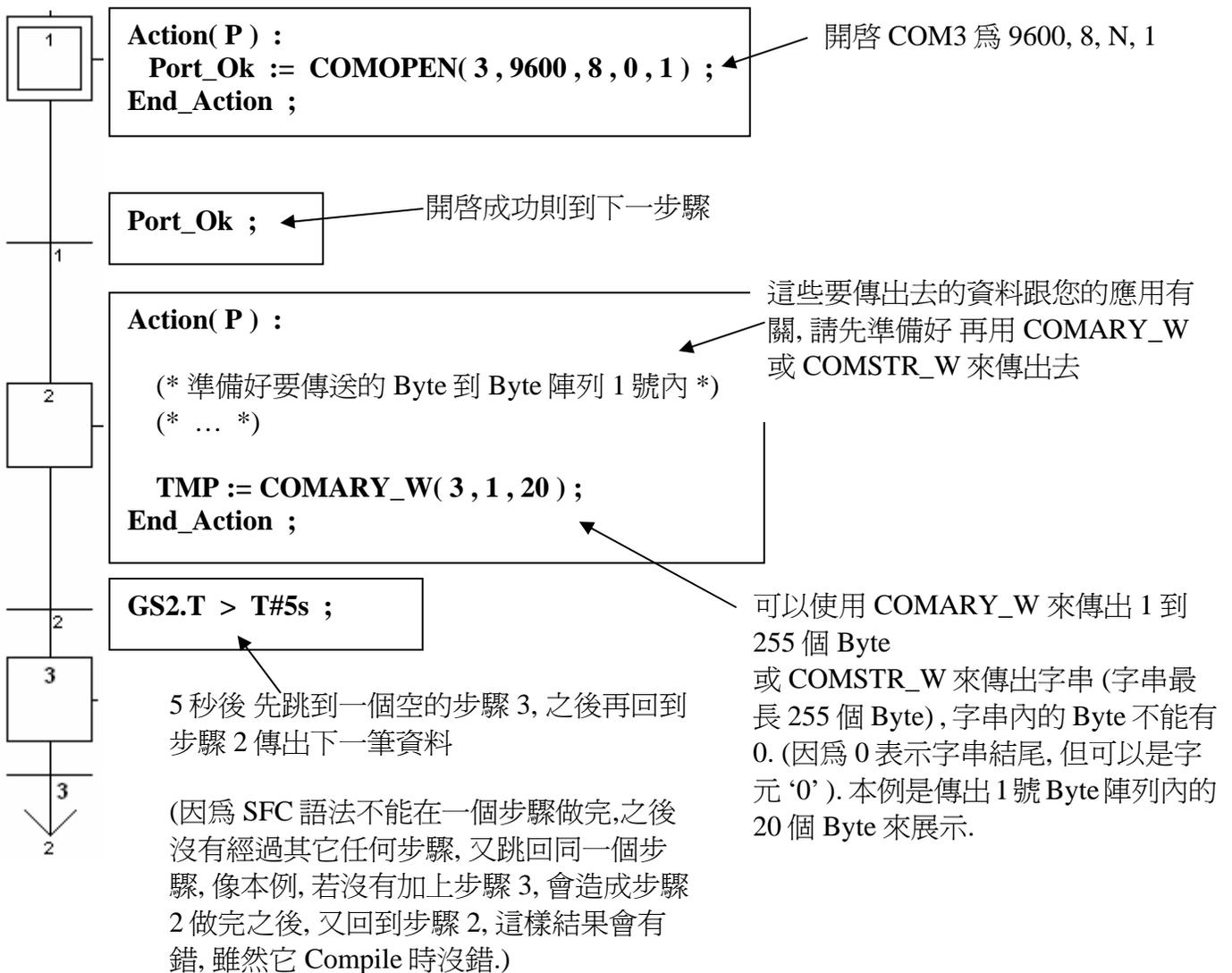
3. 每隔一段時間 主動回報資料 方式

若是要每隔一段時間就將回報資料,使用 RS-232 / 422 / 485 COM Port 傳到其它設備或 PC,如下



則可以使用類似以下的程式來處理.

使用 SFC 語法: (本例請宣告 TMP 為 Boolean Internal , Port_Ok 為 Integer Internal)



4. 有事件發生時 主動回報資料 方式

本範例當 Alarm 1 ~ 8發生時, 從Com2送出Alarm 訊息.

本程式可用於Wincon-8xx7/8xx6或I-8xx7. 若是 Wincon , 請把 “PORT” 變數的初值宣告為2, 若是 I-8xx7, 請宣告為3.

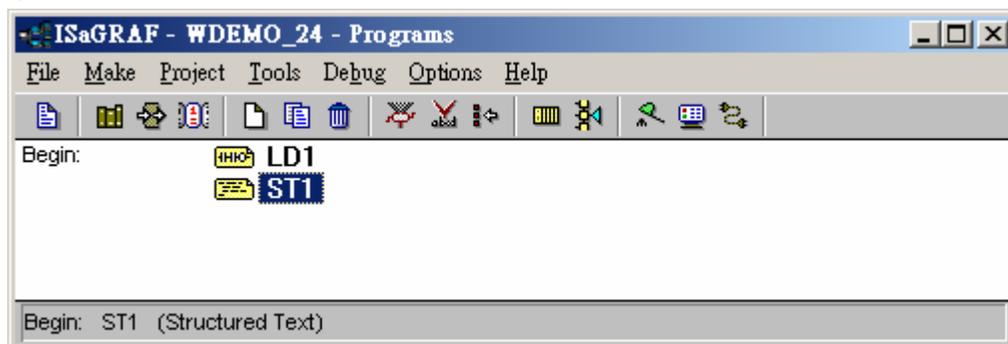
本範例放於 W-8xx7 CD-ROM:\napdos\isagraf\wincon\demo\ “wdemo_24” 或
ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/ “wdemo_24” 或
www.icpdas.com – FAQ – Software – ISaGRSAF – FAQ059

本程式有使用 變數陣列, 請參考 第2.6節關於 變數陣列 的詳細說明 (或FAQ039)

變數定義 :

Name	Type	Attribute	Description
INIT	Boolean	Internal	宣告初值為 TRUE
TMP	Boolean	Internal	暫時使用的變數
Tick1	Boolean	Internal	每秒用來產生1個 pulse True
IN[0..7]	Boolean	Input	輸入訊號, 宣告為 變數陣列, Dim 欄位設為8
OLD_IN[0..7]	Boolean	Internal	用來記錄 輸入訊號 的前一個Scan的舊值 宣告為 變數陣列, Dim 欄位設為8
ii	Integer	Internal	給ST語言 For 迴圈使用的變數
Port	Integer	Internal	Alarm訊息要送出去的COM PORT 編號, 宣告初值為2
CNT[0..7]	Integer	Internal	用來計算 輸入訊號 觸發後經過的秒數 宣告為 變數陣列, Dim 欄位設為8
Msg1	Message	Internal	要送出的Alarm訊息, 長度宣告為 128

專案架構:



動作說明:

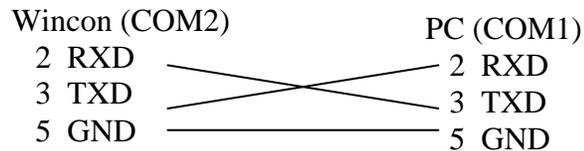
1. 當輸入訊號 IN[0..7] 由False上升到True時, 會開始計時3秒, 若這3秒內輸入訊號 一直維持在 True, 則3秒時間一到, 會發出第一個告警訊息到COM2串口
2. 當第一次告警訊息發出後, 重新再計時30秒, 若這30秒內輸入訊號還是 一直維持在True, 則30秒時間一到, 會發出另一個告警訊息到COM2串口, 之後一樣再重新再計時30秒, ...

3. 只要 輸入訊號 IN[0..7] 下降為 False, 就不再計時.

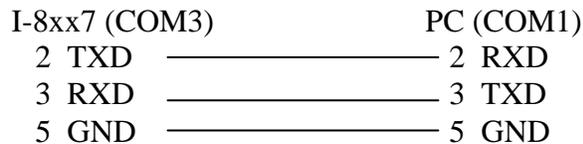
如何測試:

1. 請下載wdemo_24到W-8xx7 + slot 1: I-8077 (若是I-8xx7 請宣告 PORT初值為 3)

2. 連接RS232通訊線如下



若是用I-8xx7則如下

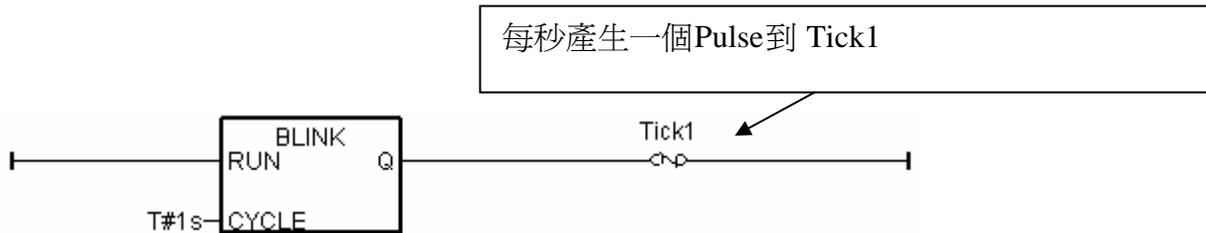


3. 開啓PC的“超級終端機”(Hyper terminal) 於 COM1, baud rate設爲9600, 8 char. size, No parity, 1 stop bit與 No flow control. 之後將I-8077的 Input 1或2或... 從FALSE切換爲TRUE, 並維持在True 等候約3秒. 若正常, Hyper terminal 應該會顯示對應的第1個告警訊息. 之後再等約30秒, 若正常, Hyper terminal 應該會顯示對應的第2個告警訊息. 若都不把 Input 切回 False, 將每過30秒就會送出一個告警訊息



程式內容:

LD1 程式:



ST1 程式:

If INIT then

INIT := FALSE ; (* 設INIT為False, 因此這段code只有第一個PLC Scan 會執行到 *)

TMP := COMOPEN(PORT , 9600 , 8 , 0 , 1) ; (* 開啓COM2串口, 9600,8,N,1 *)

(* 設CNT[0..7] 的初值為 -7 *)

for ii := 0 to 7 do

CNT[ii] := -7 ;

end_for ;

end_if ;

(* 測試IN[0..7] 是否由 False上升為True *)

for ii := 0 to 7 do

(*測試IN[0..7] 訊號是否上升為True *)

if (IN[ii] = True) and (OLD_IN[ii] = False) then

(* 訊號有上升, 設對應的CNT[] 值為 -3 *)

(* 如果CNT[] 之值不是 -7, 表示“輸入訊號被 觸發了” *)

(* 之後CNT[]之值將會每過1秒加1, 除非訊號切換回 False *)

CNT[ii] := -3 ;

end_if ;

(* 如果輸入訊號切換回 False, 對應的CNT[] 之值會設為 -7來表示“無訊號被觸發” *)

if IN[ii] = False then (* 若訊號切換回 False *)

(*將對應的CNT[]之值會設為 -7來表示“無訊號被觸發” *)

CNT[ii] := -7 ;

end_if ;

if Tick1 then (* Tick1是在LD1程式內每秒會產生1次Pulse True *)

(* 如果CNT[]之值 大於 -7, 表示訊號有被觸發 *)

```
if CNT[ii] > -7 then
```

```
    CNT[ii] := CNT[ii] + 1 ; (* 值加1 , Tick1 為 True 表示時間又1秒過去了 *)
```

```
    (* ----- *)
```

```
    (* 輸入訊號被觸發後 且 維持在觸發狀態 已經超過3秒, 從COM2送出第1次Alarm訊息 *)
```

```
    if ( CNT[ii] = 0 ) then (*當 CNT[ ] 之值由 -3, -2, -1 到變為 0 時, 送出第1次Alarm訊息 *)
```

```
        CNT[ii] := 0 ; (* 將值回復為0, 準備計時到 30 秒 *)
```

```
        (* 從COM2送出Alarm訊息 *)
```

```
        msg1 := 'Alarm ' + MSG( ii + 1 ) + ' $0A$0D' ;
```

```
        TMP := comstr_w( PORT , msg1 ) ;
```

```
    end_if ;
```

```
    (* ----- *)
```

```
    (* ----- *)
```

```
    (*輸入訊號被觸發後 且 維持在觸發狀態 又經過超過了30秒, 從COM2送出第2訊息 *)
```

```
    if ( CNT[ii] = 30 ) then (*當 CNT[ ] 之值由 1, 2, ... 到變為 30 時, 送出第2次訊息 *)
```

```
        CNT[ii] := 0 ; (*將值回復為0, 準備計時到下一次 30 秒 *)
```

```
        (* 從COM2送出Alarm訊息 *)
```

```
        msg1 := 'Alarm ' + MSG( ii + 1 ) + ', 30 sec past ! $0A$0D' ;
```

```
        TMP := comstr_w( PORT , msg1 ) ;
```

```
    end_if ;
```

```
    (* ----- *)
```

```
end_if ; (* "if CNT[ ] > -7 then" *)
```

```
end_if ; (* "if Tick1 then" *)
```

```
(* 更新OLD_IN[ ]之值 *)
```

```
OLD_IN[ii] := IN[ii] ;
```

```
end_for ;
```