

如何使用 i-8437-80 或 i-8837-80, 每 0.05 秒記錄 i-8017H 的 1 ~ 4 個電壓值於 S-256 / S-512 內的一個檔案, 連續記錄 1~10 分鐘, 之後可在 PC 上開 M.S. Excel 來看 1 ~ 4 條趨勢圖?

本範例程式為 Demo_71, 適合在 i-8437-80 或 i-8837-80 內執行, (80MHz) 驅動程式請使用 3.19 或以上版本, 放於

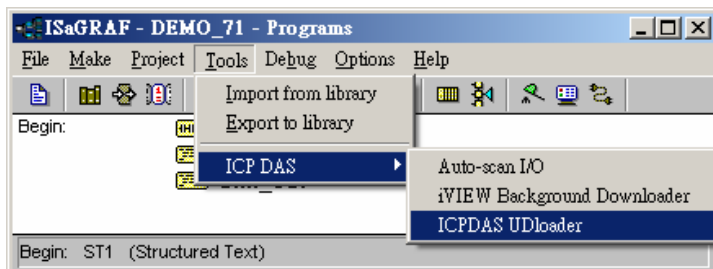
新的驅動: <http://www.icpdas.com/products/PAC/i-8000/isagraf-link1.htm>
範例程式 demo_71.pia 放於 I-8000 CD-ROM:\napdos\isagraf\8000\demo\ 內 或
<ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/8000/demo/> 或
www.icpdas.com – FAQ – Software – ISaGRAF – FAQ058

i-8437-80 與 i-8837-80 的 CPU 為 80MHz, 比 i-8437 與 i-8837 的 40MHz CPU 快約 2~4 倍. 適合用來記錄 最短取樣時間可小到約 25 ms 的資料記錄. 若是採用 i-8437 與 i-8837 (40MHz), 則只適合 100 ms 以上的 取樣時間. 但此 最短取樣時間 也跟程式的複雜度有關, 若您的 PLC 程式 執行起來的 PLC Scan Time 已經是 200ms, 那只能做 大於 200 ms 的取樣記錄.

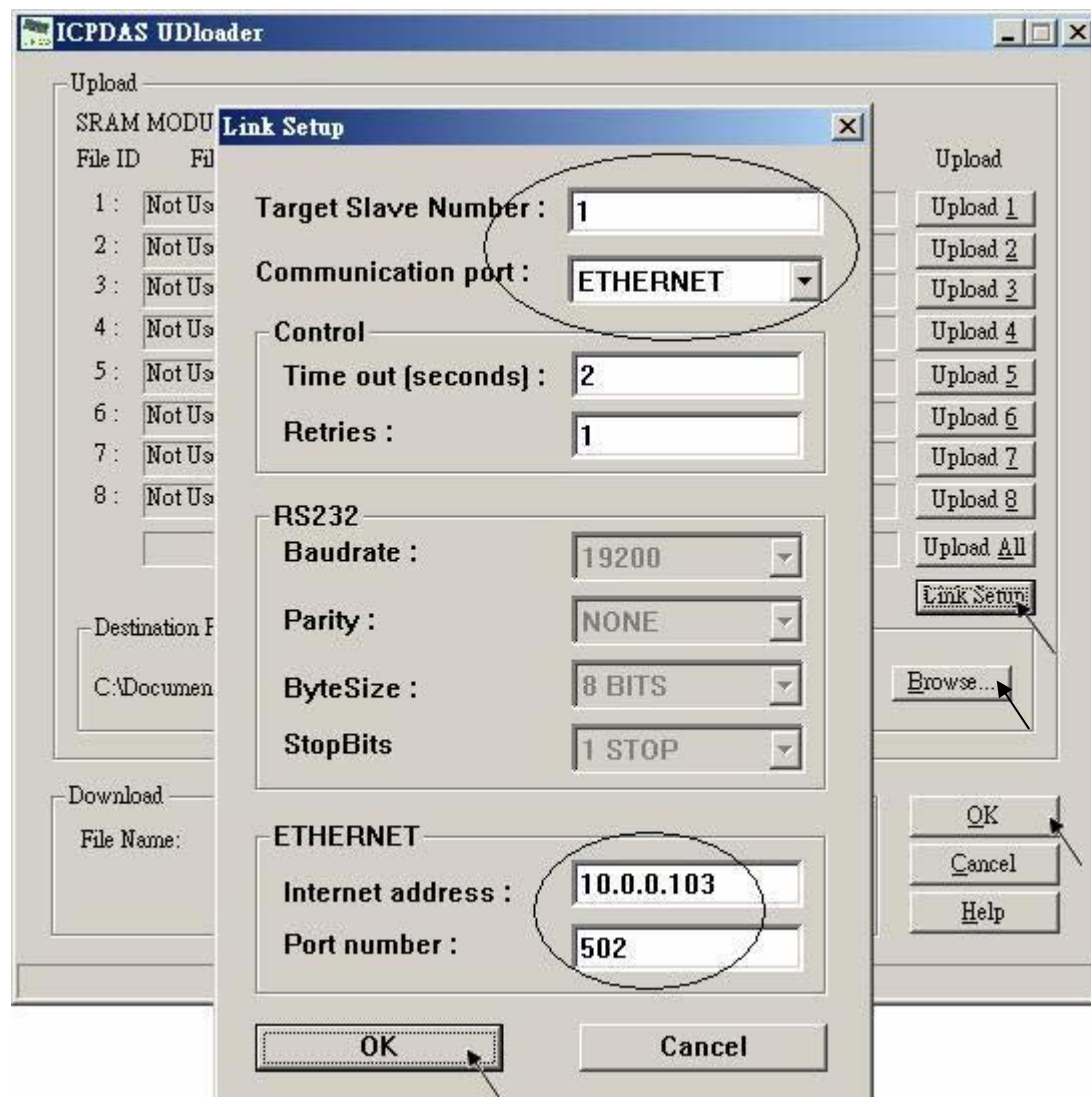
如何測試本範例:

本範例可選擇用 PC 上 VB 6.0 的 “Demo_6” 來連線操作, 或直接在 i-8437-80 / 8837-80 的面板上用按鈕來操作.

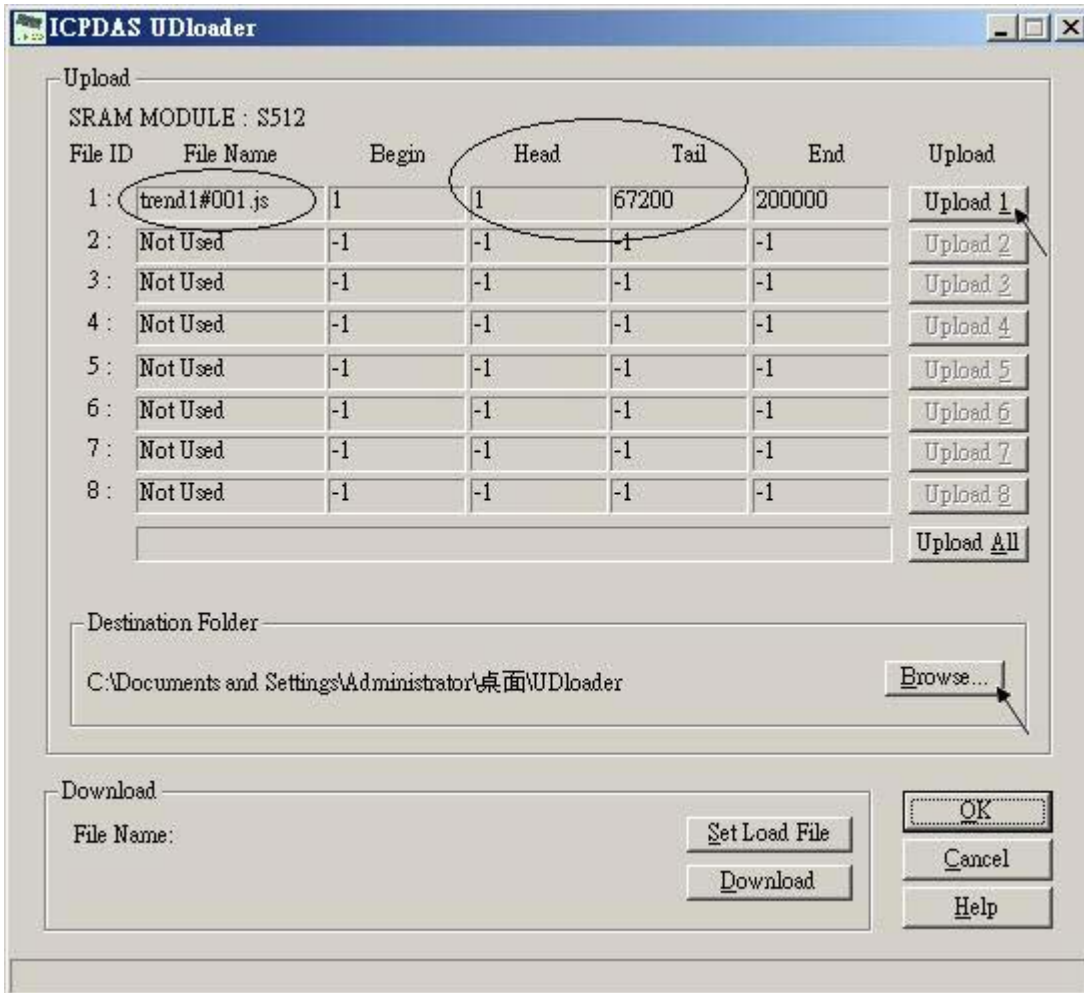
1. 首先請把 i-8024 插在 i-8437-80 的 Slot 2, i-8017H 插在 Slot 3 (注意 i-8xx7 的第 1 個 I/O Slot 為 Slot 0, 跟 W-8xx7 不同, W-8xx7 的第 1 個 I/O 為 Slot 1), 把 i-8024 的第 1 到 4 個電壓輸出點 連接到 i-8017H 的第 1 到 4 個電壓輸入點. 之後上電給 i-8437-80 .
2. 接下來請把 ISaGRAF “Demo_71” 程式下載到 i-8437-80 內 .
3. Demo_71 執行之後, 可按下 i-8437-80 面板上的第一按鈕, 開始記錄. 之後會開始記錄 2 分鐘, 記錄過程中可看到 數字會遞減到 0 並閃爍, 記錄完畢時, 面板上的 3 個 Led 燈會一直閃爍, 且數字會是 0. (若想停止記錄, 可按第 4 個按鈕)
4. 記錄完後, 可在 PC 上執行 UDownloader 把 S-256/ S-512 內的 File 上載到 PC 上來. 若你的 PC 是設計 ISaGRAF 程式的那台 PC, 請在 ISaGRAF 的 Tools – ICP DAS – ICPDAS UDownloader 把 UDownloader 運行起來. 運行起來後應該會看到步驟 (5) 的畫面.



若你的 PC 沒有 ISaGRAF, 請將 i-8000 CD:\napdos\isagraf\some_utility\ 內的 udloader.exe 覆製到 PC 的桌面上. 然後把它執行起來. 請先點選“Link Setup”設好正確的參數(若按下“Browse”可選擇要把 upload 的檔案存放於 PC 的那個路徑). 之後按下“OK” & “OK”, 儲存此設定, 並先離開 UDloader.exe , 之後再執行它.

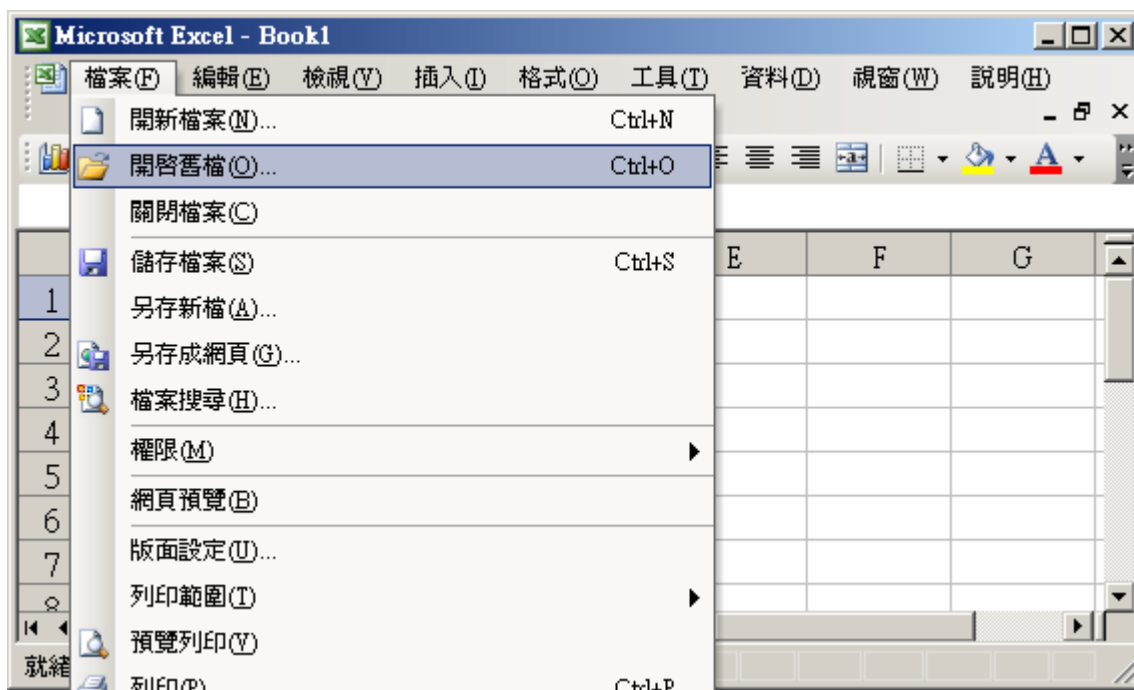


5. 接下來若有正確連上, 會看到有個 File Name 出現. 從 Head 與 Tail 的數字可看出該 S-256 / S-512 內的記錄檔的大小. 以下的例子為 67,200 bytes. (注意 S-256 / S-512 有大小限制, 請參閱第 10.3 節). 若想改變此 File 要上載後存放的位置, 可以點選“Browse ...”來變更, 之後請按下“Upload 1”來上載它.

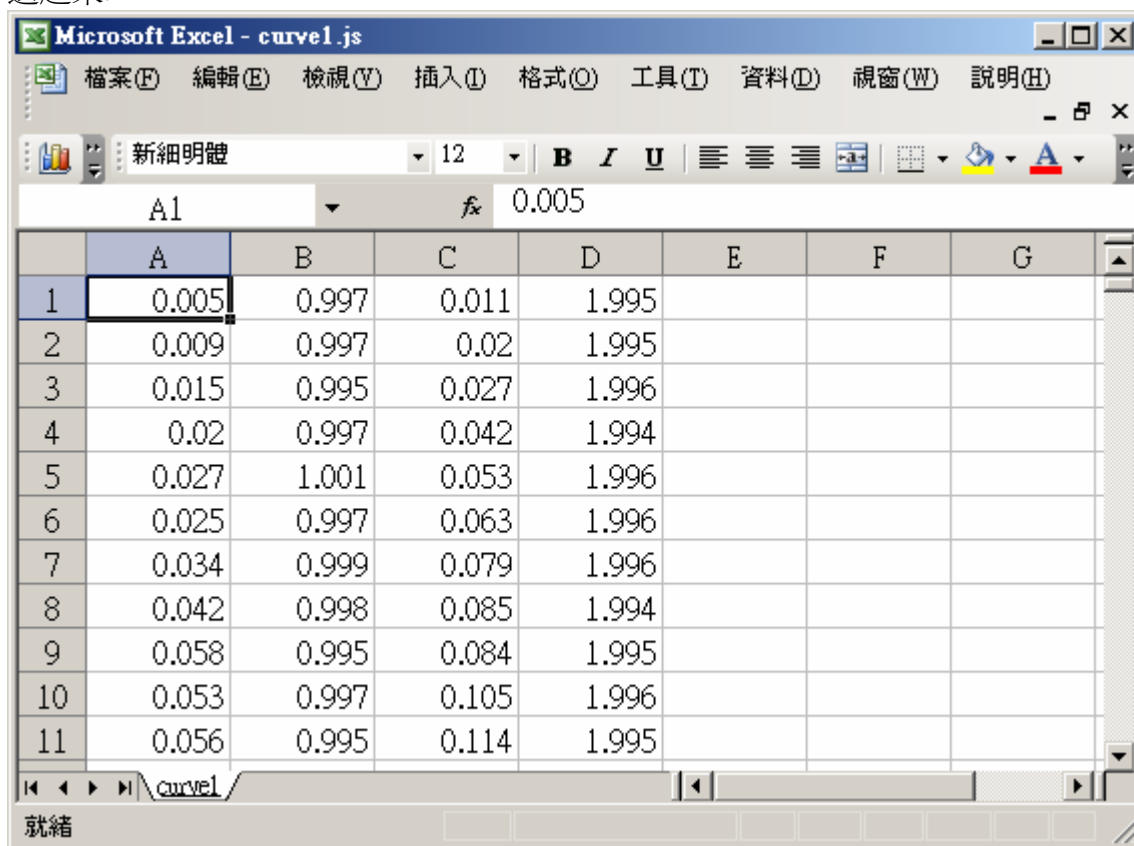


你可以到 PC 同路徑下檢查是否有上載成功.

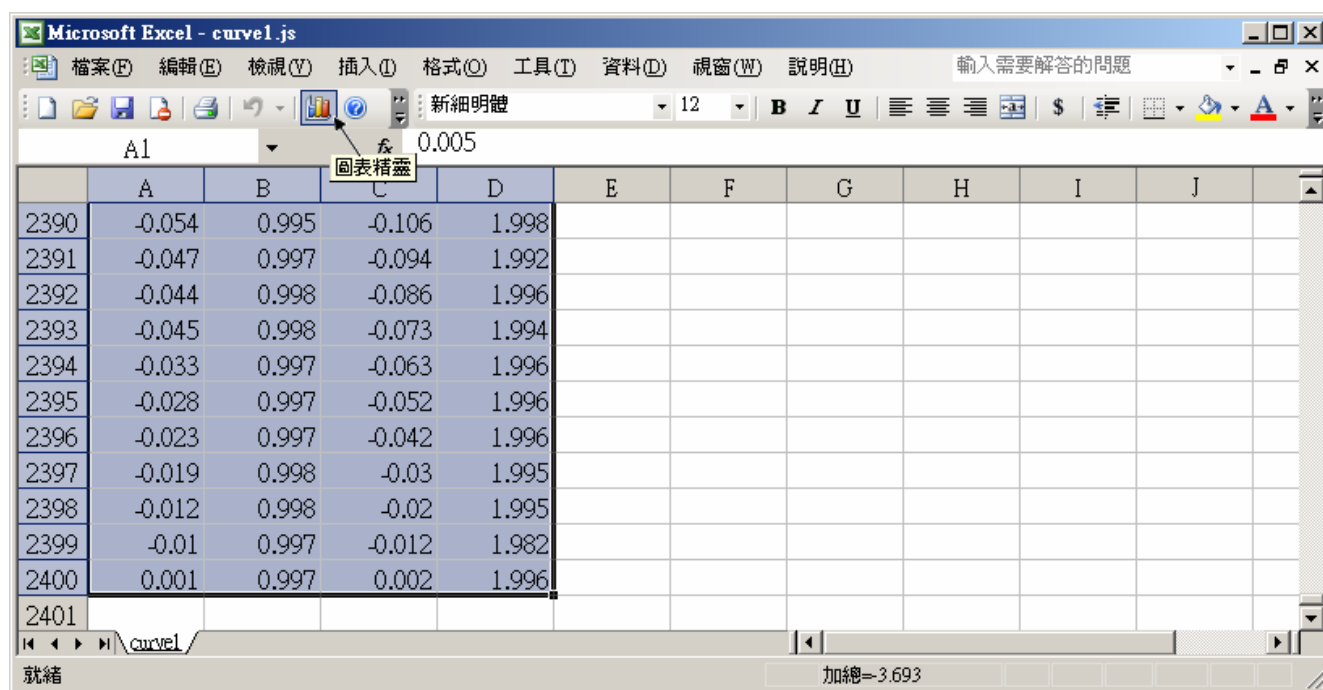
6. 記錄檔 “trend1.js” 下載到 PC 後, 請在 PC 上執行 M.S. Excel , 開啓 “trend1.js”



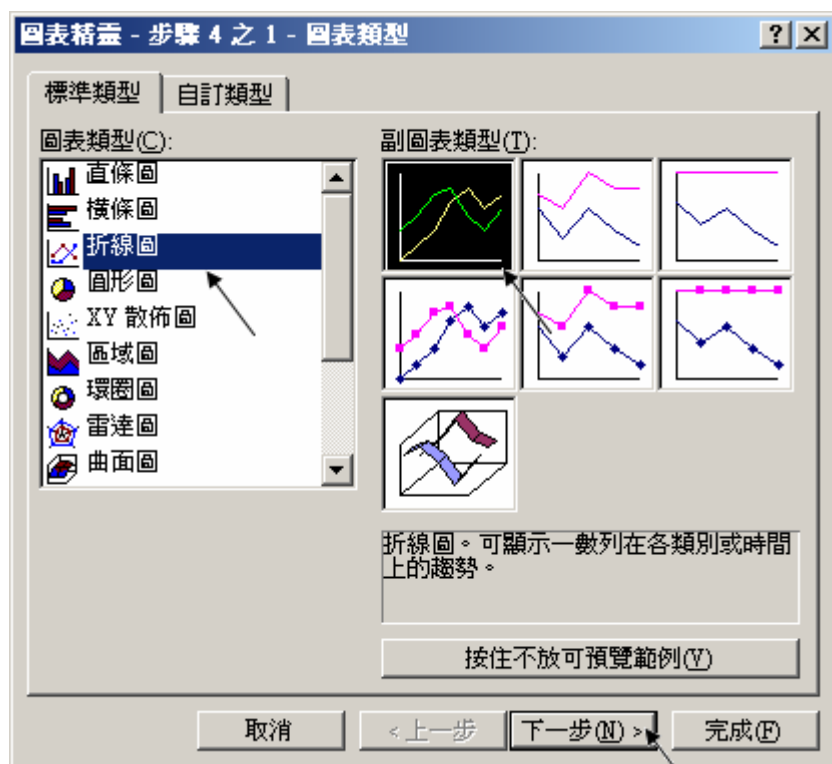
之後請先點選 最左上方的值, 再按住 “Shift” 鍵, 然後同時按下 “Ctrl” – “End”, 這樣會把所有資料選起來.

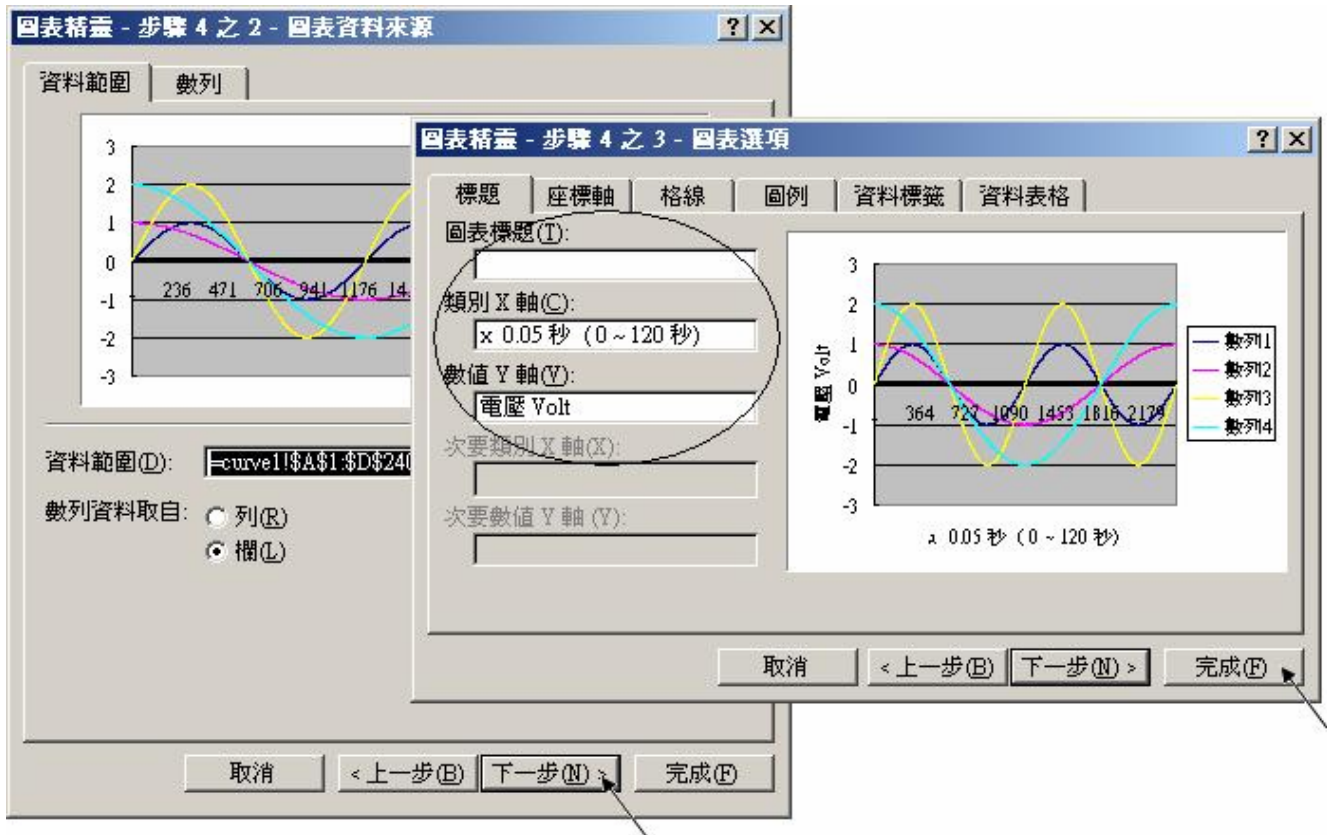


之後請按下“圖表精靈”

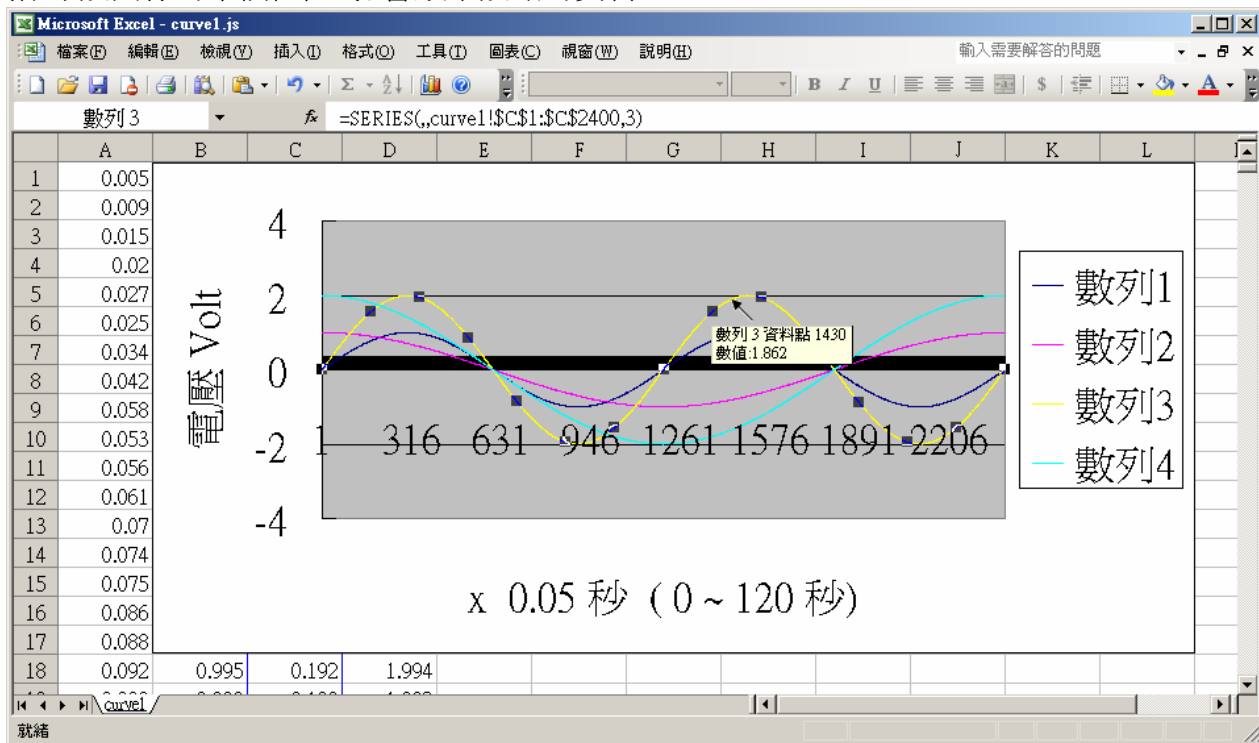


選取“折線圖”，右方請選取左上角那一個圖形，再按“下一步”

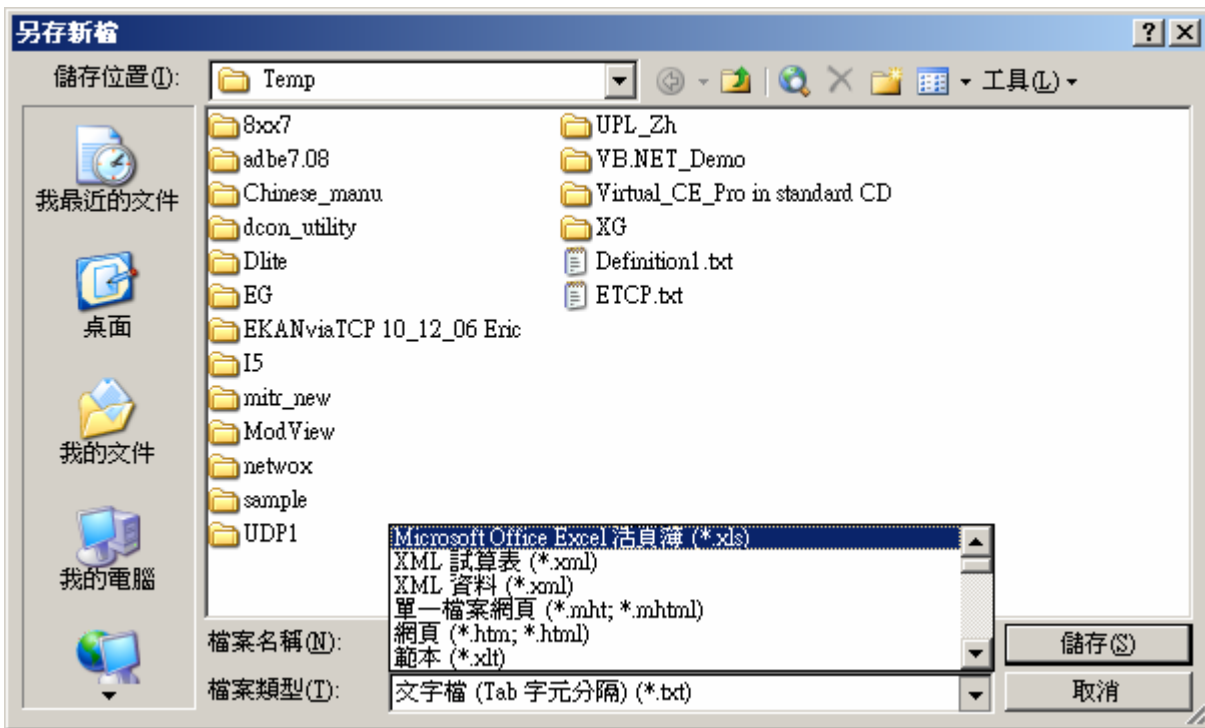
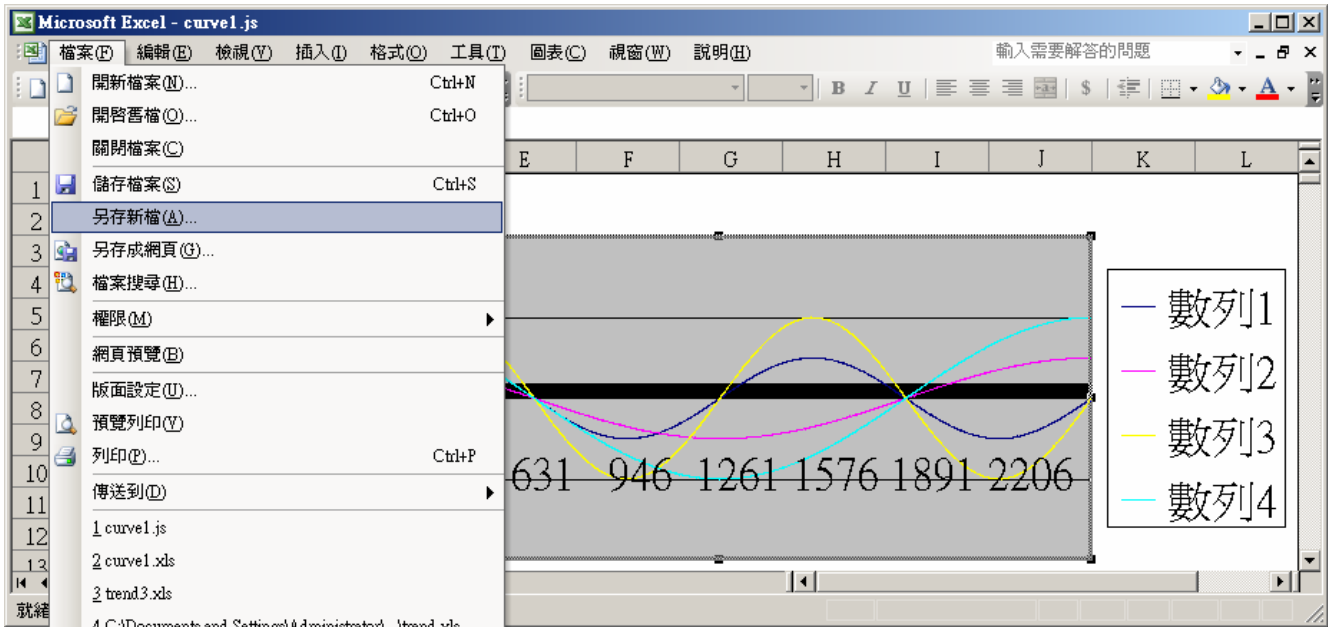




最後得到 趨勢圖如下，您可以調整圖的大小，也可以用滑鼠選取某一條顏色的曲線，當您用滑鼠指到 該曲線的某個位置時，會顯示該點的資料。

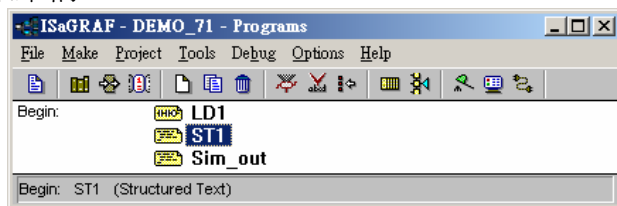


爲了日後可以方便重覆查詢該曲線，請使用“另存新檔”，將曲線儲存成“Microsoft Office Excel (*.xls)”格式。這樣下回就可直接用滑鼠雙擊“curve1.xls”來直接開啓該趨勢圖。



注意：本範例可選擇用 PC 上 VB 6.0 的“Demo_6”來連線操作

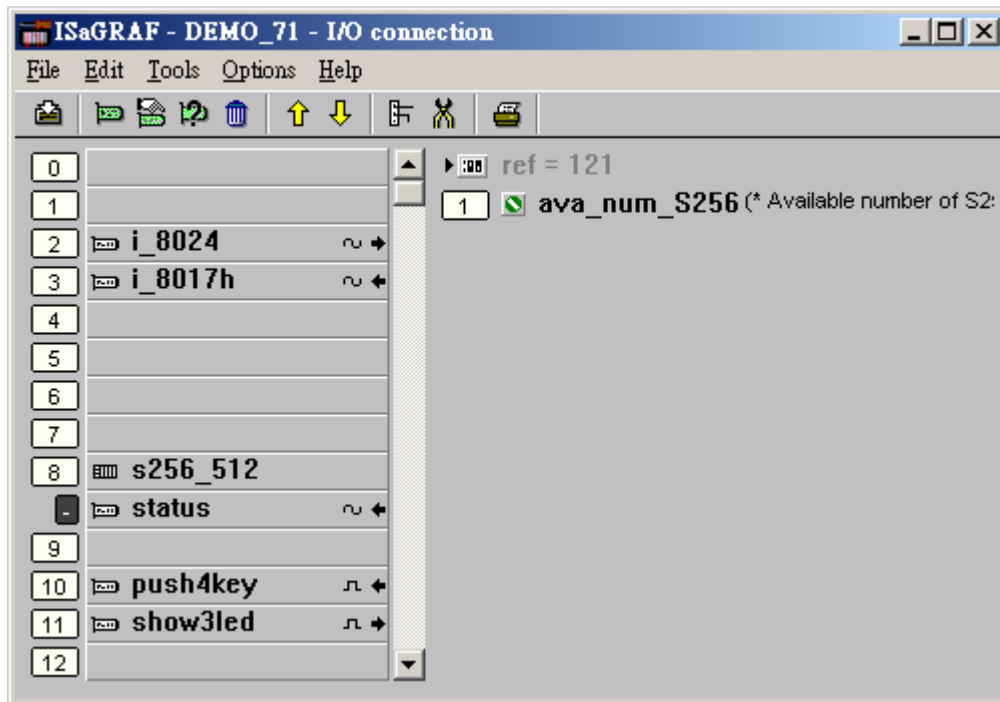
ISaGRAF 專案架構:



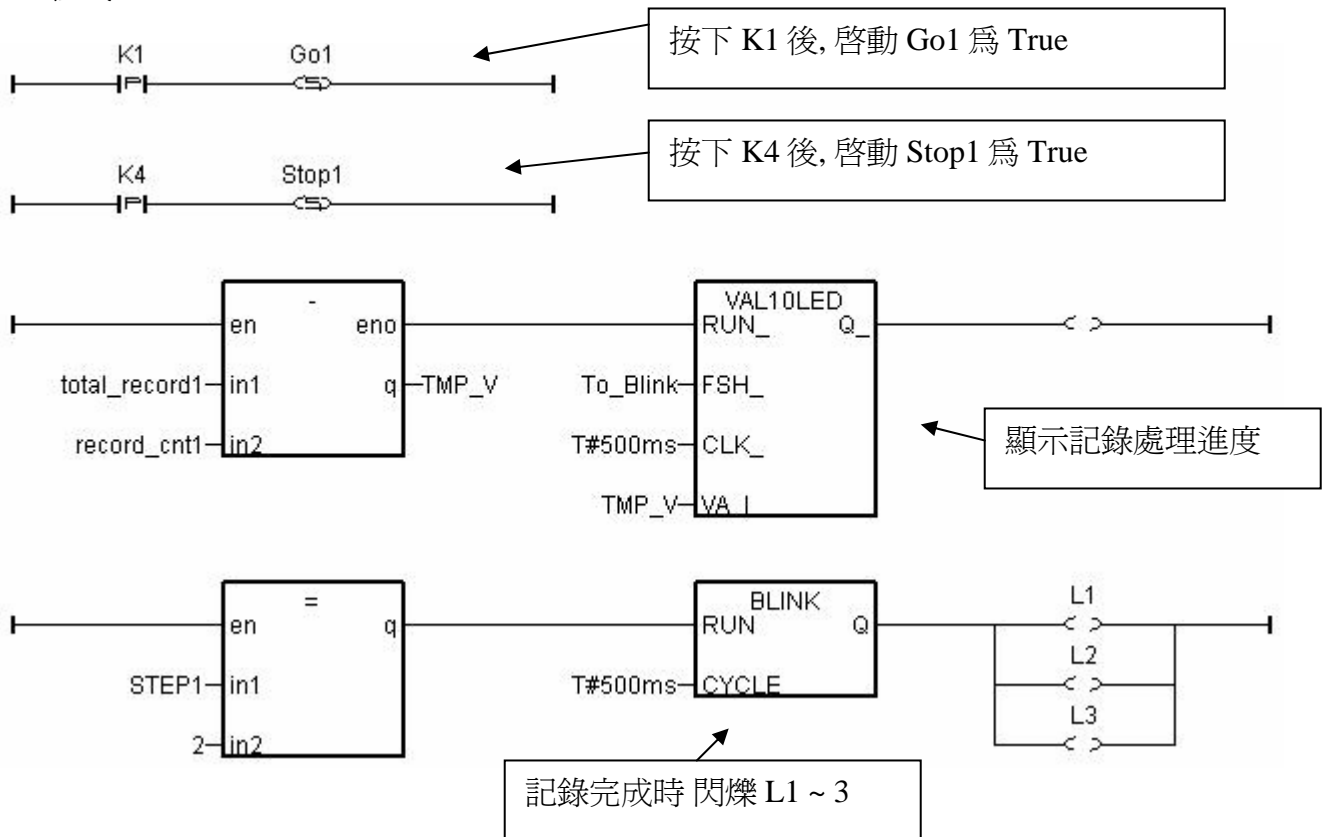
變數定義：本程式有使用 變數陣列，請參考 第 2.6 節關於 變數陣列 的詳細說明 (或 FAQ039)

Name	Type	Attribute	Description
INIT	Boolean	Internal	初值設為 True
Go1	Boolean	Internal	若設為 True 則開始動作, addr 設為 21 (16 進位值為 15)
Stop1	Boolean	Internal	若設為 True 則停止動作, addr 設為 22 (16 進位值為 16)
TMP	Boolean	Internal	暫時使用的變數
L1 , L2 , L3	Boolean	Output	連結到 show3Led 的 Ch.1 ~ Ch.3 上
K1 , K4	Boolean	Input	連結到 push4key 的 Ch.1 與 Ch.4 上 K1 用來啟動, K4 用來停止
To_Blink	Boolean	Internal	用來控制面板數字要不要閃爍
MUM_CH	Integer	Constant 常數	i-8017H 上要記錄幾個 Channel 的電壓值 本範例 記錄 4 個 Channel (Ch.1 到 4)
Ava_num_s256	Integer	Input	連結到 S256_512 的 Ch.1 ,值若為 0 表示找不到 S256/512
Current_pos1	Integer	Internal	目前正在記錄 S256/512 的那個 byte 位址
Len1 , TMP_V	Integer	Internal	程式內部使用
File_begin1 File_end1	Integer	常數	S-256 / S-512 內要配置的 Byte 開頭 與 結尾 給 File 使用
STEP1	Integer	Internal	0:無動作 , 1:記錄中 , 2:記錄完畢
Period1	Integer	Internal	共要記錄多久, 單位為分鐘, addr 設為 3
Interval1	Integer	Internal	多久記錄一筆資料, 單位為 0.001 秒, addr 設為 1
Total_record1	Integer	Internal	本次共有多少筆資料要記錄, 由程式根據 Period1 與 Interval1 之值 自動算出, addr 設為 5
Record_cnt1	Integer	Internal	目前已經完成記錄的資料數量, addr 設為 7
ii	Integer	Internal	給 for 迴圈使用
i8017H[0..7]	Integer	Input	為 變數陣列, Dim 設為 8, 需連接到 i-8017H 的 Ch1. 到 8
Volt1[0..7]	REAL	Internal	為 變數陣列, Dim 設為 8, i-8017H 的 Ch1. 到 8 轉換後的電 壓值
i8024[0..3]	Integer	Output	為 變數陣列, Dim 設為 4, 需連接到 i-8024 Ch1. 到 4
T1	Timer	Internal	作為計時用的 Timer
T1_next	Timer	Internal	下一筆資料將於何時去記錄下來, 程式自動算出
T1_Interval	Timer	Internal	每 2 筆記錄的時間 間隔
Msg1	Message	Internal	操作狀態, 長度 255, 初值為 “無動作”, addr 設為 41 (16 進 位值為 29)
Str1	Message	Internal	長度 255, 程式內部使用

IO 連結:



LD 程式 – LD1



ST 程式 – Sim_out

(* 讓 i-8024 的 Ch1 到 Ch4 輸出不同的 Sin , Cos 電壓曲線 *)

(* 2 * Pi * T1 / 60000 = T1 * 1.047197E-4 *)

(* 2 * Pi * T1 / 120000 = T1 * 5.235985E-5 *)

i8024[0] := ANA(sin(REAL(T1) * 1.047197E-4) * 3276.8) ;

i8024[1] := ANA(cos(REAL(T1) * 5.235985E-5) * 3276.8) ;

i8024[2] := ANA(sin(REAL(T1) * 1.047197E-4) * 6553.6) ;

i8024[3] := ANA(cos(REAL(T1) * 5.235985E-5) * 6553.6) ;

ST 程式 – ST1

if INIT then

INIT := FALSE ; (* 本區一開機只 Run 一次 *)

if ava_num_S256 = 0 then (* 一開機 若找不到 S-256/512 就離開 *)

Msg1 := 'S256 / S512 is not installed in i-8xx7 controller !' ;

Return ;

end_if ;

(* 在 S256/512 內配置 200K bytes 給 File 編號 1 使用, 檔名為 “trend1.js” *)

TMP := S_FL_INI(1 , 'trend1.js' , File_begin1 , File_end1) ;

TMP := S_FL_AVL(1 , -1 , -1) ; (* 一開始先設 File 內無資料 *)

end_if ;

if ava_num_S256 = 0 then (* 每個 PLC Scan 若發現 找不到 S-256/512 就離開 *)

return ;

end_if ;

(* 若 Stop1 被設為 TRUE, 則停止 T1 的計時, STEP1 設為 0 *)

if Stop1 then

Stop1 := False ;

STEP1 := 0 ; (* STEP1 設為 0:無動作 *)

TStop(T1) ; (*停止 T1 的計時 *)

T1 := T#0s ;

Msg1 := 'User stop recording !' ;

To_Blink := False ; (*停止 數字閃爍 *)

end_if ;

TMP_V := S_FL_STS(1) ; (* S_FL_STS 若回傳 非 -1 之值, 表示 PC 已上傳過該 File *)

if TMP_V <> -1 then

TMP := S_FL_RST(1) ; (* 此時再將 State 設為 -1, 讓 PC 可重覆上傳該 File *)

end_if ;

(* 若 Go1 被設為 TRUE, 準備開始動作 *)

```
if Go1 then  
  Go1 := False ;
```

(* STEP1: 0:無動作, 1:記錄中, 2: 記錄完畢 *)

```
if STEP1 = 1 then  
  Msg1 := 'It is still recording now ...' ; (* STEP1 為 1, 表示記錄中 )
```

```
else
```

(* 檢查 Interval 值, 需在 25 到 10000 ms 之間, 本例使用 50 ms *)

```
if ( Interval1 < 25 ) or ( Interval1 > 10000 ) then  
  Msg1 := 'Wrong Interval value, it should be in 25 to 10000 milli-second !' ;
```

(* 檢查 Period1 之值 設的 對不對, 需為 1 到 10 之間, 單位為 分鐘 *)

```
elsif ( Period1 < 1 ) or ( Period1 > 10 ) then  
  Msg1 := 'Wrong Period value, it should be in 1 to 10 minute !' ;
```

```
else
```

```
  total_record1 := ( Period1 * 60000 ) / Interval1 ; (* 算出 total_record1 值 *)
```

```
  record_cnt1 := 0 ; (* 將目前已記錄資料數量設為 0 *)
```

```
  STEP1 := 1 ; (* 將 STEP1 設為 1:記錄中 *)
```

```
  Msg1 := 'Recording now ...' ;
```

(* 啟動 T1 開始從 0 起計時 *)

```
  T1 := T#0s ;
```

```
  T1_Interval := TMR( Interval1 ) ;
```

```
  T1_next := T1 + T1_Interval ;
```

```
  TStart(T1) ;
```

```
  Current_pos1 := 1 ; (* 將 S256/512 內的資料位址 RESET 為 1 *)
```

```
  To_Blink := TRUE ; (* 設為 TRUE 讓數字閃爍 *)
```

```
  end_if ;
```

```
end_if ;
```

```
end_if ;
```

(* 以下為 STEP1 為 1: 記錄中需作的 code *)

```
if STEP1 = 1 then
```

```
  if T1 >= T1_next then (* 當 T1 計時到 下 1 筆該記錄的時間點時, 就準備記錄 1 筆資料 *)
```

```
    T1_next := T1_next + T1_Interval ; (* 重新設定 下 1 筆該記錄的時間點 *)
```

(* 因為 Timer 最多只能計時到 23 小時 59 分 59 秒,所以提前在剛過 20 小時之後,就 Reset 為 0 *)
(* 本範例最多只有計時 10 分鐘,所以不會超過,以下這段 的用意,在於 給使用者參考用 *)

```
if T1 >= T#20h then
  T1 := T#0s ;
  T1_next := T1 + T1_Interval ;
end_if ;
```

```
str1 := '' ; (* 一開始將 str1 設為空字串 *)
for ii := 0 to NUM_CH - 1 do
  (* 轉換 i8017H 的類比輸入值為 電壓值 *)
  Volt1[ii] := Real( i8017H[ii] ) * 0.000305176 ; (* 10.0 / 32768 = 0.000305176 *)
  str1 := str1 + Rea_Str2( Volt1[ii] , 3 ) + '$09' ; (* 分隔字元為 <TAB> *)
end_for ;
```

```
str1 := str1 + '$0D$0A' ; (* 加入 <CR> <LF> 字元於每行尾端 *)
Len1 := MLEN( str1 ) ; (* 取得字串 str1 的目前長度 *)
```

(* 若記錄的資料 byte 數量 將 超過 S256/512 配置給 File 的使用量,就離開,不再記錄下去了 *)

```
if ( Current_pos1 + Len1 - 1 ) > File_end1 then
  STEP1 := 0 ; (* 設為 0:無動作 *)
  Msg1 := 'File allocated memory is not enough to hold the data !' ;
  Tstop( T1 ) ; (* 停止 T1 的計時 *)
  T1 := T#0s ;
  To_Blink := FALSE ; (*停止 數字閃爍 *)
  Return ; (* 離開此 ST 程式 *)
end_if ;
```

(* 將字串 str1 內的所有 byte 寫入 S-256/512 的目前位址內 *)

```
TMP := S_M_W( Current_pos1 , Len1 , str1 ) ;
Current_pos1 := Current_pos1 + Len1 ; (*目前位址往前移 *)
TMP := S_FL_AVL( 1 , 1 , Current_pos1 - 1 ) ; (*重設 S256/512 的 File 頭/尾位址 *)
```

```
record_cnt1 := record_cnt1 + 1 ; (* 資料數量加 1 *)
if ( record_cnt1 >= total_record1 ) then (* 檢查是否所有資料都已記錄完畢 *)
  STEP1 := 2 ; (* 若是則設 STEP1 為 2:記錄完畢 *)
  Msg1 := 'Record is finished ! You may download the record file to your PC now !' ;
  Tstop( T1 ) ;
  T1 := T#0s ;
  To_Blink := FALSE ; (*停止 數字閃爍 *)
end_if ;
```

```
end_if ;
end_if ;
```