

如何使用 Wincon-8xx7, 每 0.01 秒記錄 i-8017H 的 1 ~ 4 個電壓值於 User 配置的一塊 RAM 記憶區內, 連續記錄 1~10 分鐘, 之後再儲存到 RAM Disk File 內, 最後可在 PC 上開 M.S. Excel 來看 1 ~ 4 條趨勢圖?

本範例程式為 Whmi_13.pia 只能在 Wincon-8xx7/8xx6 的 驅動 3.36 版或更高的版本使用, 放於

新的驅動: <http://www.icpdas.com/products/PAC/i-8000/isagraf-link.htm>

範例程式 whim_13.pia 放於 W-8xx7 CD-ROM:\napdos\isagraf\wincon\demo\ 內 或
ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/ 或
www.icpdas.com – FAQ - Software – ISaGRAF – FAQ057

本範例 PC 的人機介面程式 可選擇用 VB6 的 “Demo_5” 放於

W-8xx7 CD-ROM:\napdos\isagraf\wincon\vb6_demo_pc\ 內 或
ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/vb6_demo_pc/ 或
www.icpdas.com – FAQ - Software – ISaGRAF – FAQ057

或使用 Web HMI 來當人機介面程式, 原始碼放於 (關於 Web HMI 請參考 Wincon ISaGRAF Getting Started 第 3, 4, 5 章 : W-8xx7 CD-ROM:\napdos\isagraf\wincon\english_manu\ “getting_started_w8337.pdf”)

W-8xx7 CD-ROM:\napdos\isagraf\wincon\WebHMI_Demo\ 內的 whim_13 或
ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/webhmi_demo/ 或
www.icpdas.com – FAQ - Software – ISaGRAF – FAQ057

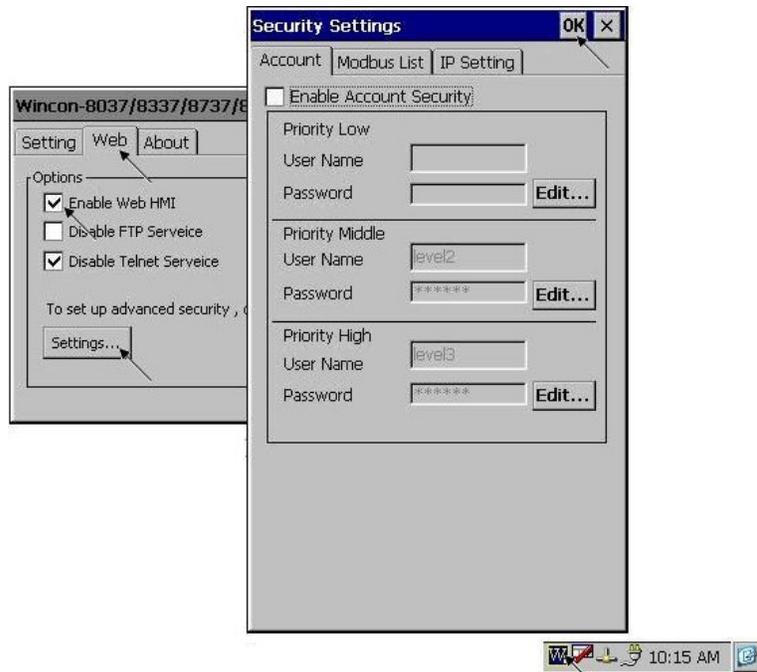
若您 PC 上裝的 ISaGRAF 找不到 Msg_F, Msg_N, ARY_F_R, AFY_F_W 等函式, 請訪問 <http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> 來下載 “ICP DAS utilities For ISaGRAF”, 解壓縮後, 執行其內的 setup.exe 來安裝新的 函式進去 ISaGRAF 內

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)

如何測試本範例:

以下步驟為使用 Web HMI 當人機介面程式的方式, 若您是使用 VB6 的 “Demo_5” 來當人機介面, 只需把 VB6 程式 (Demo_5.exe) 在 PC 上 Run 起來, 之後只做 步驟 1, 步驟 3 與 步驟 6 .

1. 首先請把 i-8024 插在 W-8xx7 的 Slot 2, i-8017H 插在 Slot 3, 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 Wincon, 然後在 ISaGRAF Driver 上啓用 Web HMI 功能, 為方便操作, 請先不要選取 “Enable Account Security”



2. 接下來請把 Web HMI Demo 13 的所有 file 用 ftp (例如在 IE 上輸入 “ftp://10.0.0.103”), 丟到 Wincon 的 \CompactFlash\Temp\HTTP\WebHMI\ 內

Web HMI 的原始碼放於

W-8xx7 CD-ROM:\napdos\isagraf\wincon\WebHMI_Demo\ 內的 whim_13 或
ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/webhmi_demo/

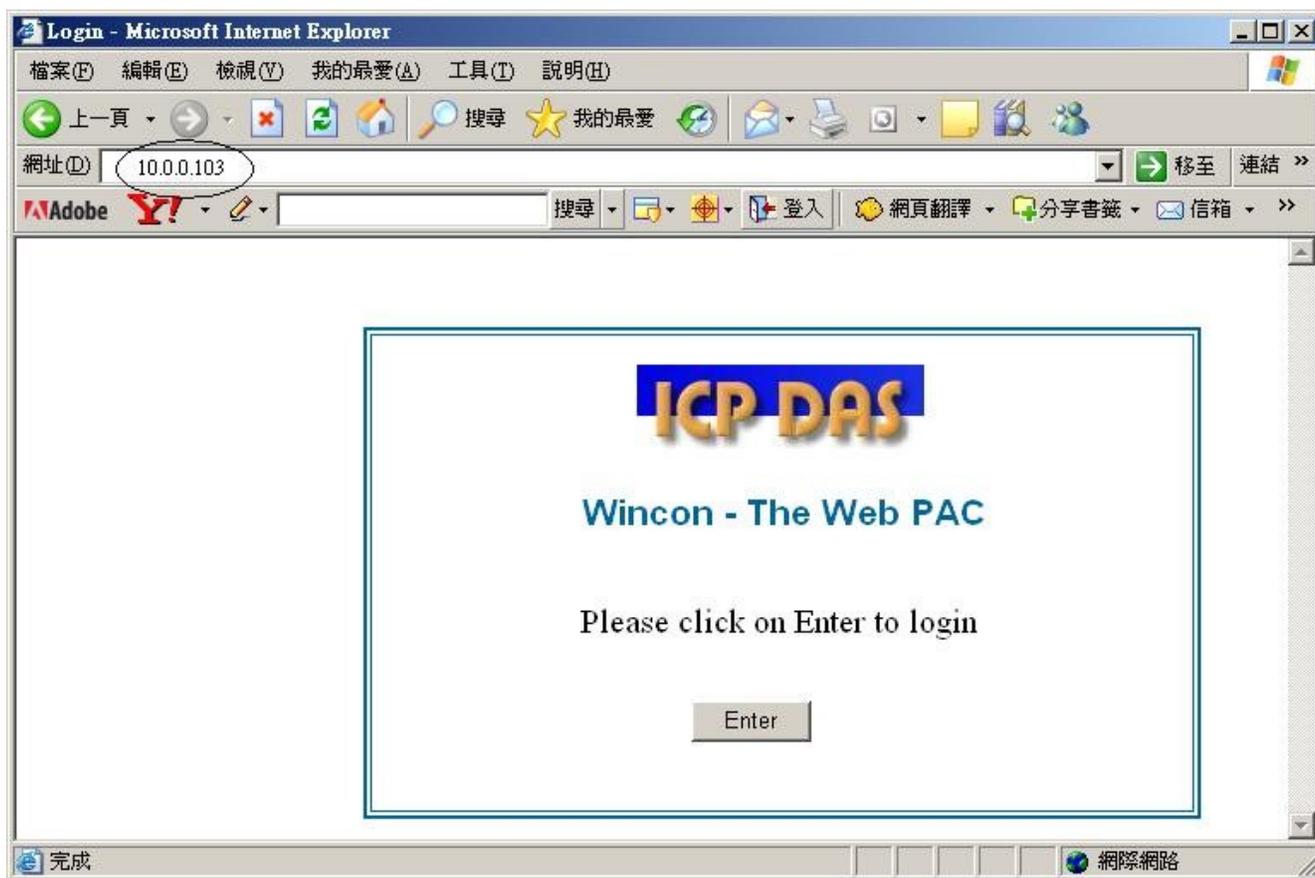
共有 7 個 File 與 2 個目錄要 copy 到 Wincon 的 \CompactFlash\Temp\HTTP\WebHMI\ 內:
Main.htm , menu.htm , index.htm , login.htm , main.dll , login.dll , whim_filter.dll
img 目錄 與 msg 目錄

3. 之後請把 ISaGRAF 的 whim_13 程式下載到 W-8xx7 內. (若是使用 Web HMI, 請做完步驟 2, 再做步驟 3)

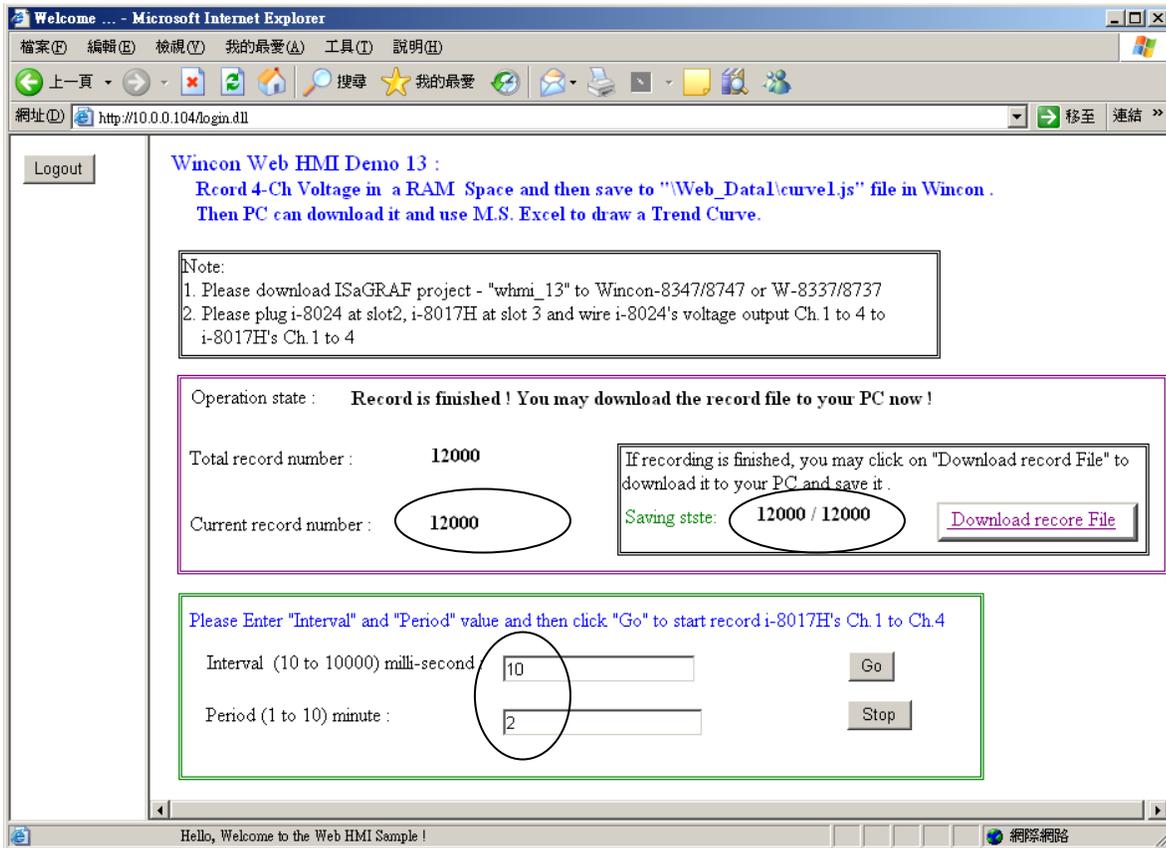
本範例程式為 Whmi_13.pia 只能在 Wincon-8xx7/8xx6 內使用, 放於

W-8xx7 CD-ROM:\napdos\isagraf\wincon\demo\ 內 或
ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/

4. PC 上開啓 Internet Explorer (至少需爲 IE 5.0 版, 或更高的版本), 上方輸入 W-8xx7 的 IP, 連上 Wincon 後, 請按 “Enter” 進入



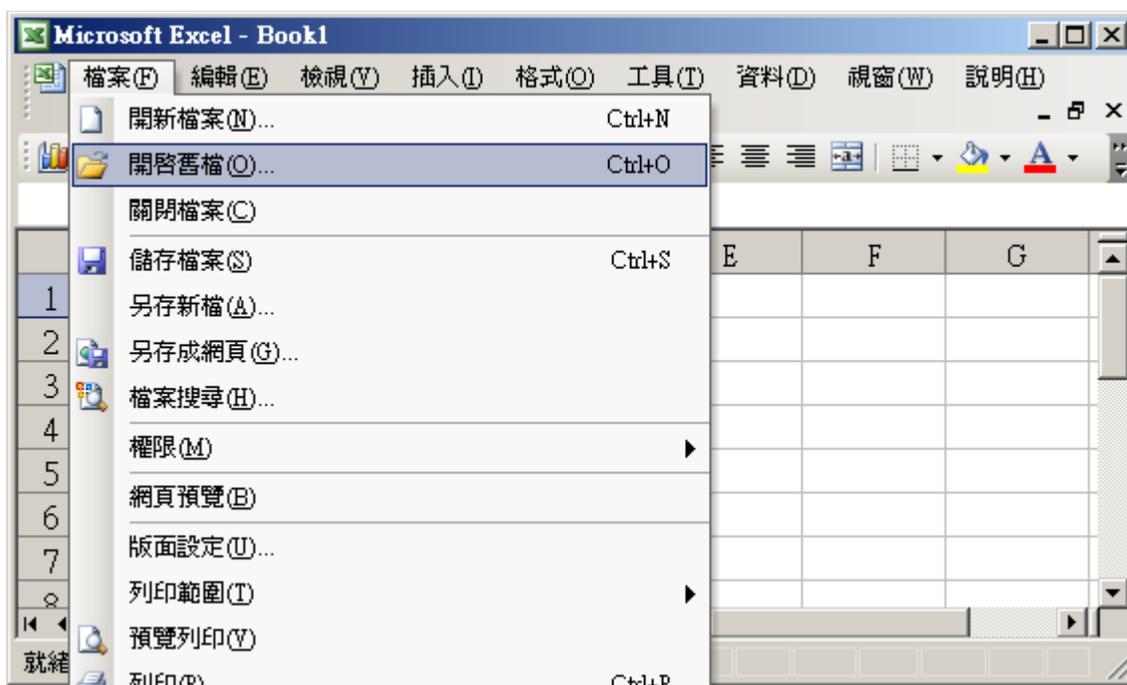
5. 之後請輸入 “Interval” 的值, 單位為 0.001 秒, 如給 10 指的是 每 0.01 秒, 記錄一次. 而 “Period” 值指的是 要記錄多久, 單位為分鐘. 輸入好後, 按下 “Go” 可開始動作. 之後 i-8024 的第 1 到 4 點會輸出不同的電壓, 若有在 步驟 1 內將 i-8024 的第 1 到 4 個電壓輸出點 連接到 i-8017H 的第 1 到 4 個電壓輸入點, 則 i-8017H 的輸入會有不同的曲線變化會被記錄下來.



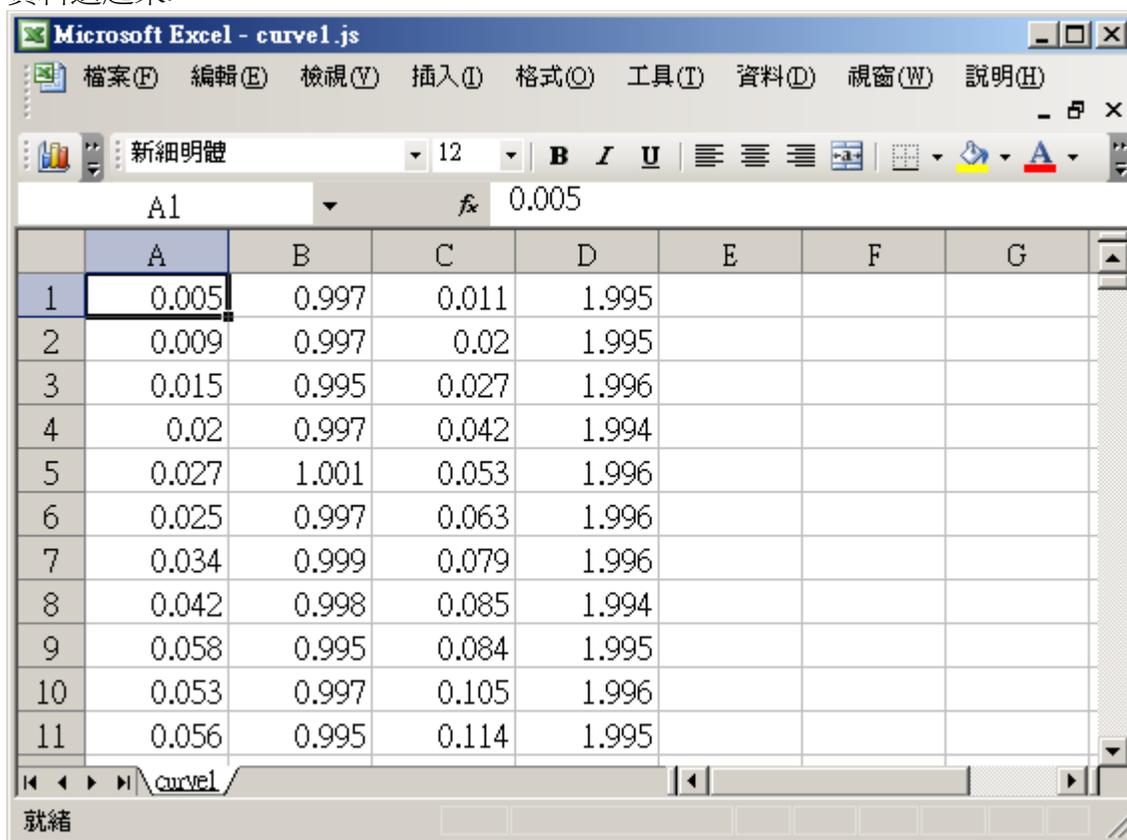
在記錄過程中, 會看到 “Current record number” 的值會一直增加, 當它等於 “Total record number” 之值時, 就表示記錄完畢. 此時 程式會開始把記錄資料存入 RAM File 內, 從 “Saving state” 可以看到進度, 若都完成, 請按下 “Download record File”, 點選 “儲存” 來下載到 PC 內.



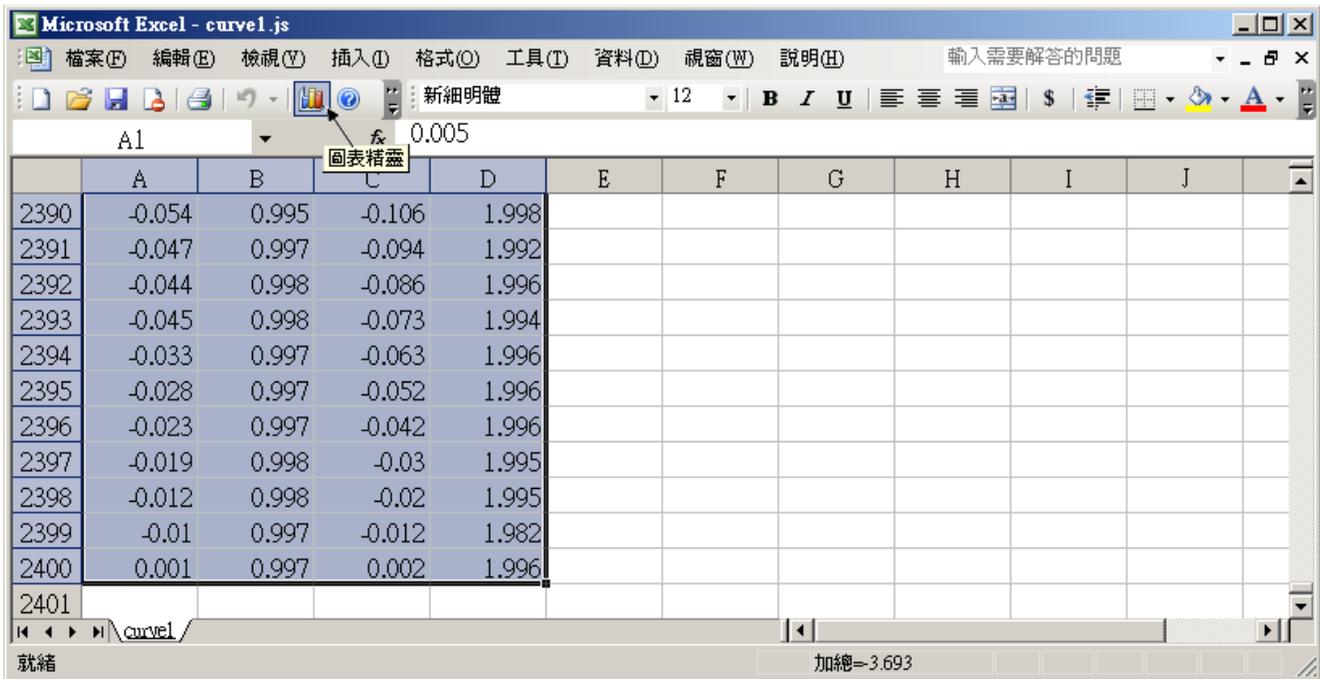
6. 記錄檔 “curver1.js” 下載到 PC 後, 請在 PC 上執行 M.S. Excel , 開啓 “curver1.js”



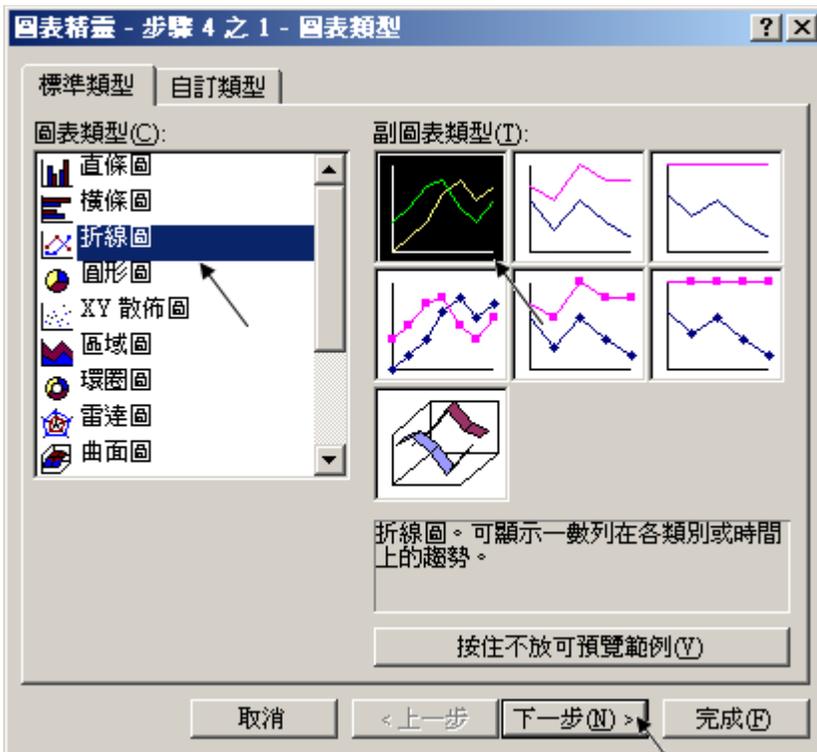
之後請先點選 最左上方的值, 再按住 “Shift” 鍵, 然後同時 按下 “Ctrl” – “End”, 這樣會把所有資料選起來。

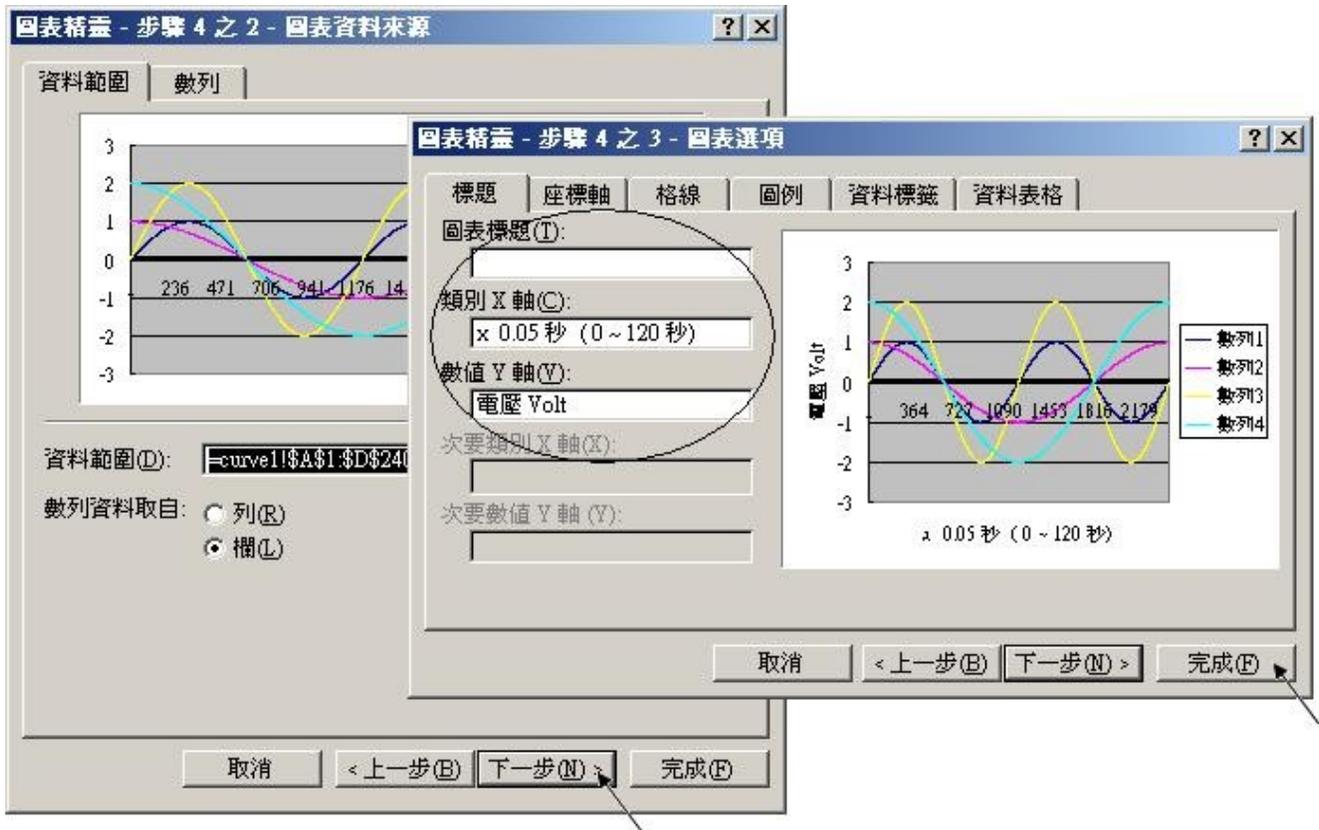


之後請按下 “圖表精靈”

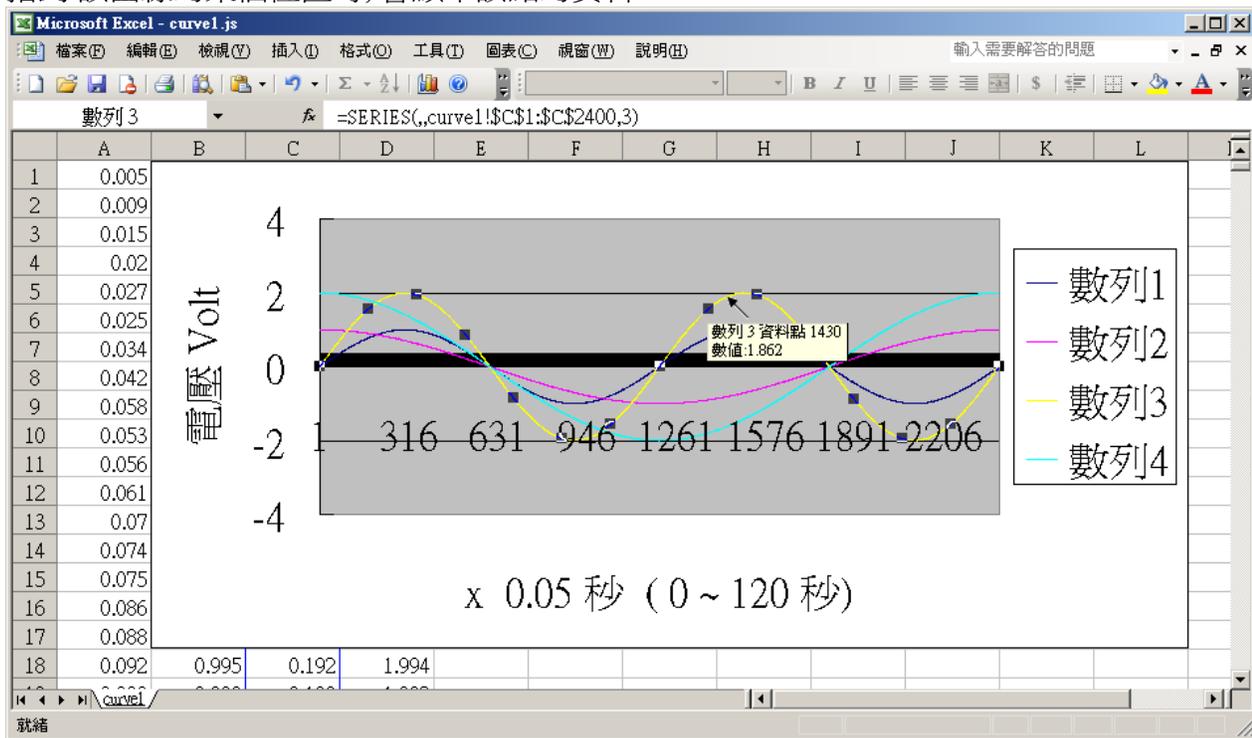


選取 “折線圖”, 右方請選取 左上角 那一個圖形, 再按 “下一步”

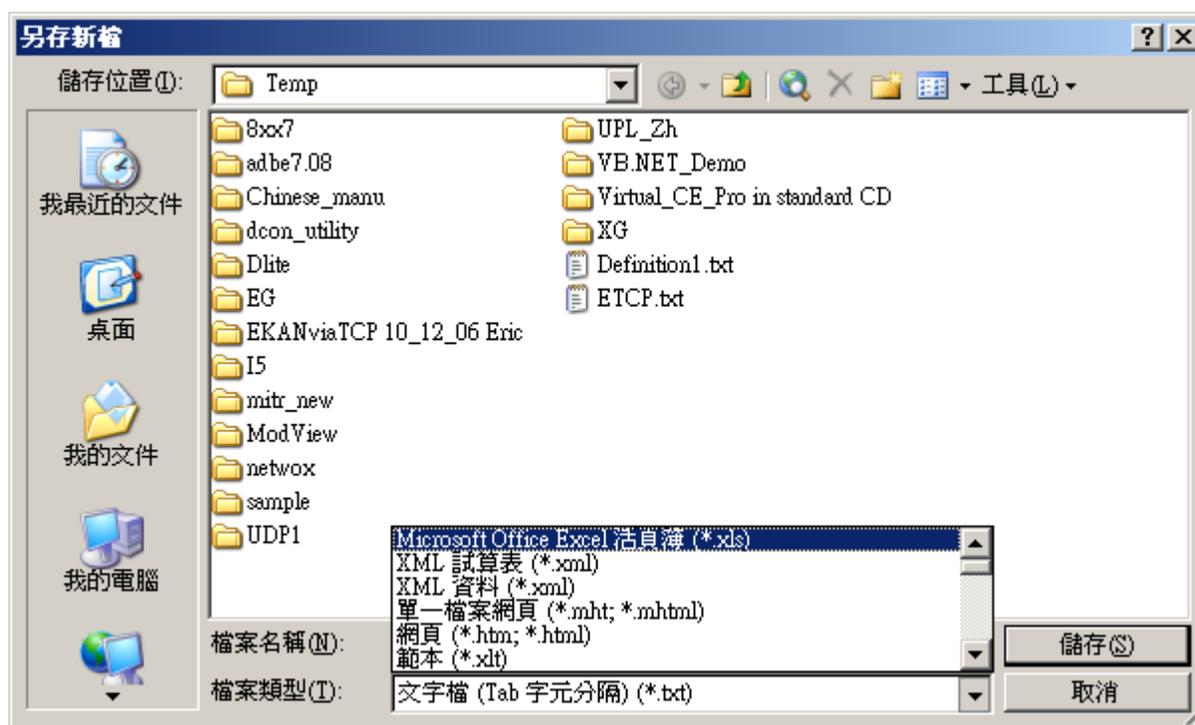
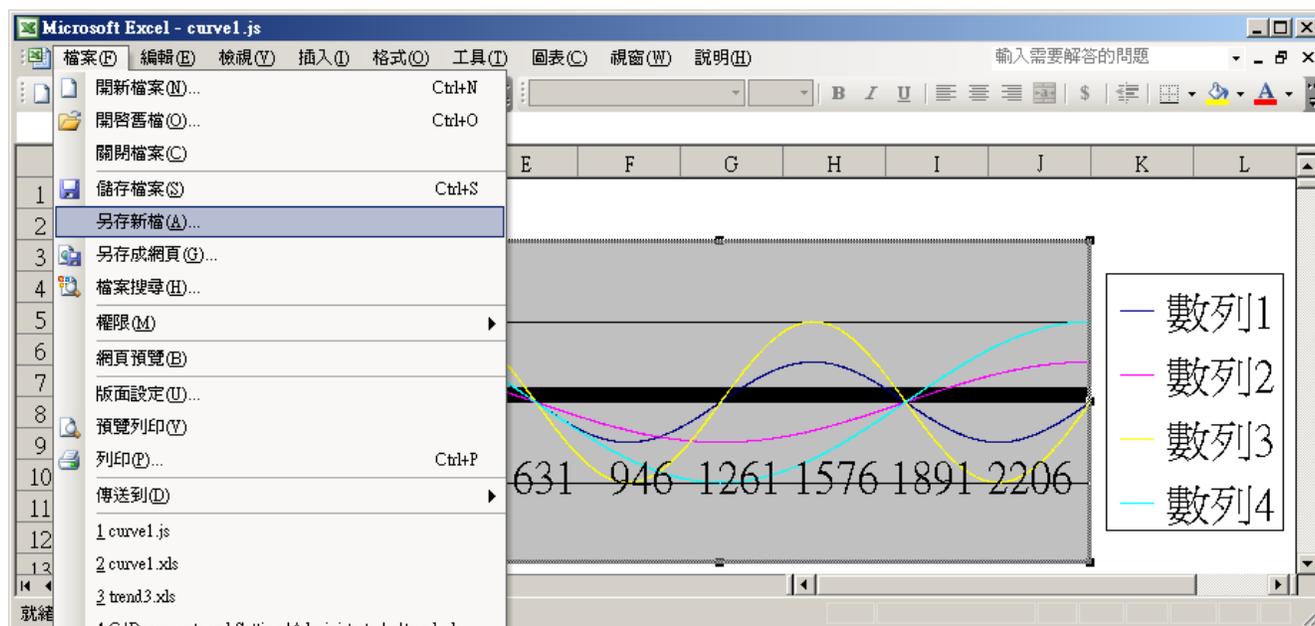




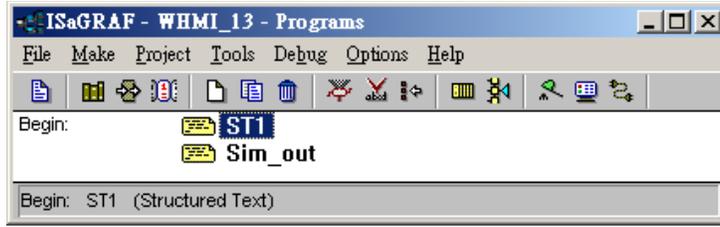
最後得到趨勢圖如下，您可以調整圖的大小，也可以用滑鼠選取某一條顏色的曲線，當您用滑鼠指到該曲線的某個位置時，會顯示該點的資料。



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.



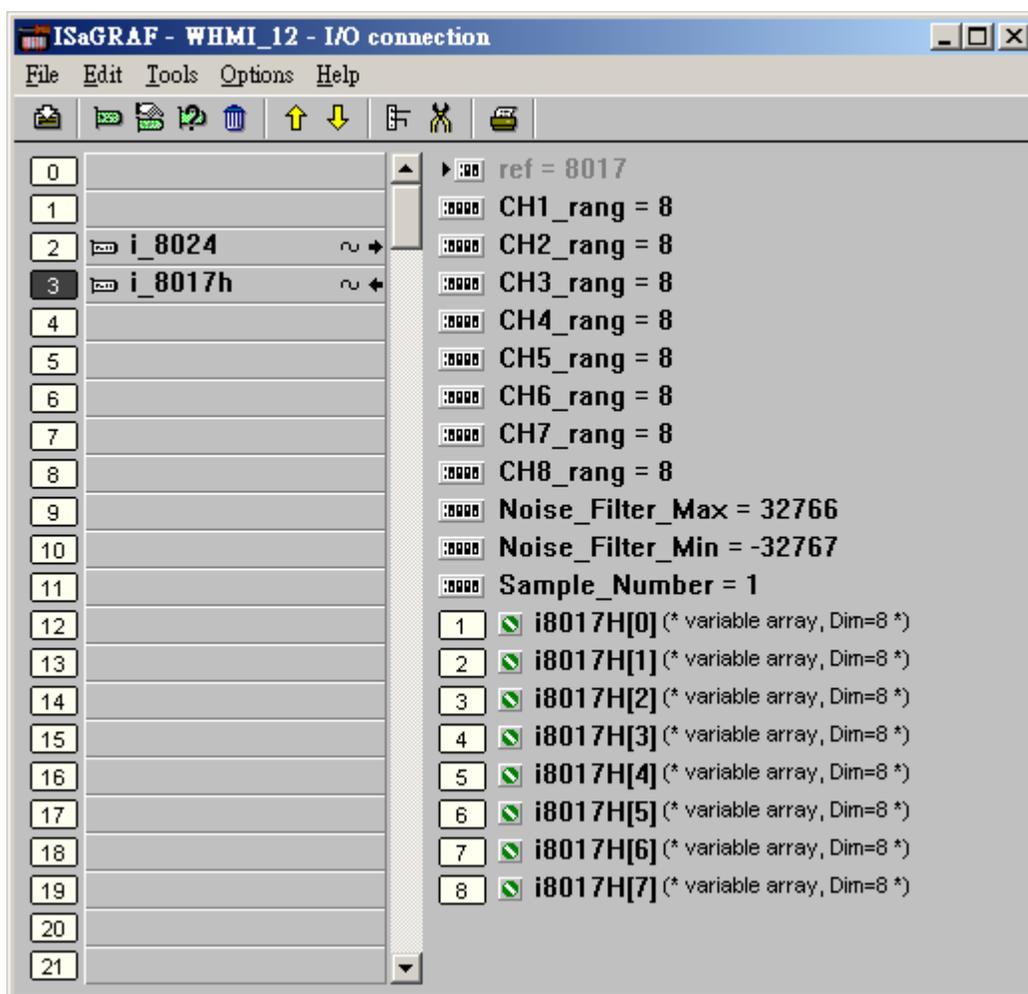
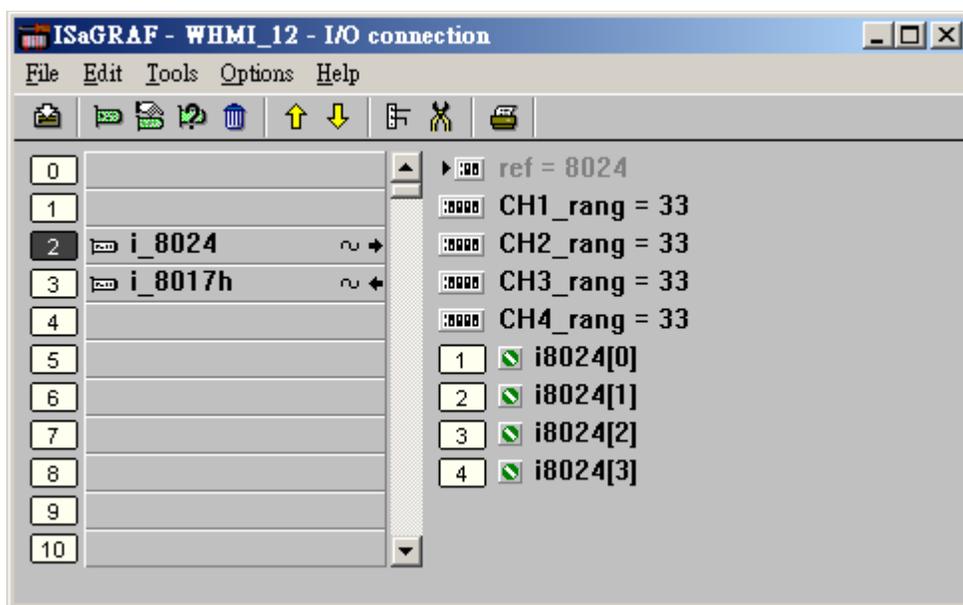
ISaGRAF 專案架構:



變數定義：

Name	Type	Attribute	Description
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
INIT	Boolean	Internal	初值設為 True
Save_file1	Boolean	Internal	程式會自動設為 True 來儲存 RAM Disk File
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)
File1	Integer	Internal	File ID
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
Intervall1	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 Intervall1 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 與 ii2	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
Save_cnt1	Integer	Internal	目前儲存於 RAM disk File 內的 Record 數量, addr 設為 9
TMP_v	Integer	Internal	程式內部使用
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
File_name1	Message	Internal	File name, 長度 64, 初值為 \Web_Data1\curve1.js Web HMI 只支持 RAM Disk File 於 \Web_Data1\ 內, 若是 CompactFlash File 則 Web HMI 只支持於 \CompactFlash\Temp\HTTP\Data\ 內 (參見 11.2 節的 Whmi_08 範例)
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:



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

(* 以下這區只在第 1 個 PLC scan 內 Run 一次 *)

if INIT then

INIT := False ;

(* 配置 1 塊可存放 500,000 個整數 (或實數) 的 RAM 記憶區, W-8xx7 最多只能配置 3,000,000 個整數/實數, 一個整數/實數為 4 個 bytes, 所以最多是 12,000,000 bytes *)

(* ARcreate() 只能在 ISaGRAF 程式內使用一次, 且第 1 個參數只可以是 1 *)

TMP_v := ARcreate(1, 500000) ;

if TMP_v <> 1 then (* 回傳值若不是 1 表示失敗 *)

Msg1 := 'Parameter error or Can not allocate enough memory by ARcreate() function !' ;

end_if ;

(* 設定 W-8xx7 以最高速度執行, 此動作會使得同時在 Wincon 上 Run 的非 ISaGRAF 程式變慢 *)

TMP := PLC_mode(-1) ;

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 !' ;

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 ... Please wait' ; (* 更新狀態顯示為 “仍在記錄中 …” *)

else

    (* 檢查 Interval1 之值 設的 對不對, 需為 10 到 10000 之間, 單位為 0.001 秒 *)
    If ( Interval1 < 10 ) or ( Interval1 > 10000 ) then
        Msg1 := 'Wrong Interval value, it should be in 10 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

        (* 值設的正確 , 接下來 準備開始記錄資料 *)
        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 ... Please wait' ;

        (* 啟動 T1 開始從 0 起計時 *)
        T1 := T#0s ;
        T1_Interval := TMR( Interval1 ) ;
        T1_next := T1 + T1_Interval ;
        TStart( T1 ) ;

    end_if ;
end_if ;
end_if ;

if STEP1 = 1 then      (* 1:記錄中 *)

    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 ;
(* ----- *)

(* 準備寫入一筆資料於 User 配置的 RAM 記憶區內 *)
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 *)

  (* 先使用 Real_int() 將實數 Map 成整數, 再存入 User 配置的 RAM 記憶區內 *)
  TMP_v := ARwrite( 1 , NUM_CH * record_cnt1 + ii , Real_int( Volt1[ii] ) ) ;

  (* 檢查是否寫入成功 *)
  if TMP_v <> 1 then (* 回傳值若不是 1 表示失敗, 有可能是 配置的記憶區不夠大 *)
    Msg1 := 'Can not operate ARwrite( ) !' ;
    STEP1 := 0 ; (* Step 設為 0: 無動作 *)
    TStop( T1 ) ; (* 停止 T1 的計時 *)
    T1 := T#0s ;
  end_if ;
end_for ;

  record_cnt1 := record_cnt1 + 1 ; (* 已記錄的資料數量 加 1 *)
  if ( record_cnt1 >= total_record1 ) then (* 檢查是否全部資料都已記錄完畢 *)

    (* 全部資料都已記錄完畢, 準備開始把資料從 User 配置的 RAM 記憶區, 存入 RAM Disk File
    *)
    STEP1 := 0 ; (* Save 之前, 先暫時將 Step 設為 0: 無動作 *)
    Tstop( T1 ) ; (* 停止 T1 的計時 *)
    T1 := T#0s ;

    (* 建立一個新的 RAM Disk File, File_name1 於 \Web_Data1\ 內 *)
    File1 := F_creat( File_name1 ) ;
    if File1 = 0 then (* 建立 File 失敗 *)
      Msg1 := 'Create File ' + 'File_nam1 Error !!!' ;
    else

      (* 將 save_file1 設為 TRUE 來開始分散於多個 PLC scan 內把資料存成 File *)
      Msg1 := 'Please wait ... Saving data to file : ' + File_name1 + ' ...' ;
      save_file1 := True ;
      save_cnt1 := 0 ; (* 先將目前已存入記錄數量設為 0 *)
    end_if ;

```

```
end_if ;
```

```
end_if ;
```

```
end_if ;
```

(* 以下 每個 PLC scan 只存入 50 筆記錄, 請不要在 1 個 PLC scan 內就把所有記錄存完, 因為 存資料到 File 會花較多的 CPU 時間, 若強迫在 1 個 PLC scan 內存完, 該 PLC scan 的 Scan time 會變的很大, 會影響到其它的 I/O 邏輯操作*)

```
if save_file1 then
```

```
for ii2 := 0 to 50 do (* 每個 PLC scan 只存入 50 筆記錄 *)
```

```
if save_cnt1 < total_record1 then (* 若還未存完, 繼續存 *)
```

```
str1 := '' ; (* 先將 str1 設為空字串 *)
```

```
for ii := 0 to NUM_CH - 1 do (* 將實數資料轉成字串, 分隔字元為 <TAB> *)
```

```
str1 := str1 + Rea_Str2( Int_real( ARread( 1, NUM_CH * save_cnt1 + ii ) ), 3 ) +  
'$09' ;
```

```
end_for ;
```

```
str1 := str1 + '$0D$0A' ; (* 加入 <CR> <LF> 字元於每行尾端 *)
```

```
TMP := F_writ_s( File1 , str1 ) ; (* 寫入字串 *)
```

```
save_cnt1 := save_cnt1 + 1 ; (* 目前已存入記錄數量加 1 *)
```

```
else
```

```
(* 所有資料都已存完 *)
```

```
save_file1 := False ; (* 設為 False 表示不再存了 *)
```

```
TMP := F_close( File1 ) ; (* 關閉該 File *)
```

```
STEP1 := 2 ; (* 設為 2: 記錄完畢 *)
```

```
Msg1 := 'Record is finished ! You may download the record file to your PC now !' ;
```

```
end_if ;
```

```
end_for ;
```

```
end_if ;
```
