如何使用 i-8437-80 或 i-8837-80, 每 0.05 秒記錄 i-8017H 的 1~4 個電壓値於 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: <u>http://www.icpdas.com/products/PAC/i-8000/isagraf-link1.htm</u> 範例程式 demo_71.pia 放於 I-8000 CD-ROM:\napdos\isagraf\8000\demo\ 內 或 <u>ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/8000/demo/</u> 或 <u>www.icpdas.com</u> – 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 UDLoader 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 UDloader 把 UDloader 運行起來. 運行起來後應該會看到步驟 (5) 的畫面.

ISaGRAF - DEMO_71 - Programs	
<u>File Make Project Tools</u> Debug O	rtions <u>H</u> elp
Image: Segin: Image: Segin: Image: Segin: Segin: Image: Segin: Segin	uy 🛄 🕅 🤽 🗒 🖏
ICP DAS	Auto-scan I/O iVIEW Background Downloader
Begin: ST1 (Structured Text)	ICPDAS UDloader

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.

SRAM MODUL	ink Satur		
File ID Fil	mik setup		Upload
1: NotUs 2: NotUs 3: NotUs 4: NotUs	Target Slave Number : Communication port :	1 ETHERNET	Upload 1 Upload 2 Upload 3 Upload 4
5: NotUs 6: NotUs 7: NotUs	Time out (seconds) : Retries :	2	Upload <u>5</u> Upload <u>6</u> Upload 7
8 : Not Us Destination F C:Documen	RS232 Baudrate : Parity : ByteSize : StonBits	19200 - NONE - 8 BITS -	Upload <u>8</u> Upload <u>A</u> ll Link Senur <u>B</u> rowse
Download File Name:	ETHERNET Internet address : Port number :	10.0.0.103 502	<u>O</u> K <u>C</u> ancel <u>H</u> elp

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.

ule II	D File Name	Beg	in Head	Tail	End	Upload
1:(trend1#001.js	$\sum 1$	Ŀ	67200	200000	Upload 1
2:	Not Used	-1	-1	-1	-1	Upload 2
3 :	Not Used	-1	-1	-1	-1	Upload 2
4:	Not Used	-1	-1	-1	-1	Upload 4
5:	Not Used	-1	-1	-1	-1	Upload 3
6:	Not Used	-1	-1	-1	-1	Upload f
7:	Not Used	-1	-1	-1	-1	Upload]
8:	Not Used	-1	-1	-1	-1	Upload §
Des C:\I	tination Folder — Documents and Sett	ings\Admin	istrator/桌面\UDlo	ader		Browse
ownl File N	load				Set Load File	<u>OK</u>

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"

M	іісто	soft Excel - Bo	ok1						_ [
:2	檔3	案(F) 編輯(E)	檢視(♡)	插入①	格式(0)	工具(<u>T</u>)	資料(D)	- 視窗(₩)	說明(H)	
:		開新檔案(N)				Ctrl+N	1		- 1	8×
1	2	開啓舊檔(0)				Ctrl+O			🔕 - <u>A</u>	• •
		關閉檔案(C)								
		儲存檔案③				Ctrl+S	E	F	G	
1		另存新檔(A)								
2	<u>.</u>	另存成網頁(<u>G</u>).								_
3	1	檔案搜尋(H)								
4		櫂限(M)				•				
5		網頁預覽(B)								-
		版面設定(U)								
×		列印範圍(T)				•				
 	۵,	預覽列印(Ⅵ)								
就維	A	ÆIIEΠ/ Ρ\				CtrlLP				//.

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.

💌 M	icrosoft Excel	l - curve1.js					_	
2	檔案(E) 編輯	₿(E) 檢視(V) 插入①	格式(0) 工具	具(I) 資料(I	D) 視窗(₩)	說明(<u>H</u>)	
: : (15)a	**:新細明體		• 12 •	вли		= =		
:	× 1	_	£	0.005		= 🖼 🖂 '	<u> </u>	· 두
	AI	•	jx					
	A	В	С	D	E	F	G	
1	0.005	0.997	0.011	1.995				
2	0.009	0.997	0.02	1.995				
3	0.015	0.995	0.027	1.996				
4	0.02	0.997	0.042	1.994				
5	0.027	1.001	0.053	1.996				
6	0.025	0.997	0.063	1.996				
7	0.034	0.999	0.079	1.996				
8	0.042	0.998	0.085	1.994				
9	0.058	0.995	0.084	1.995				
10	0.053	0.997	0.105	1.996				
11	0.056	0.995	0.114	1.995				-
14 4	► ► \curve1				•	1		١Ē
就緒								

之後請按下"圖表精靈"

🔀 Mic:	Microsoft Excel - curve1.js										
:劉 楷	客(E) 編輯	Œ) 檢視(V)	插入① 相	各式(0) 工具(T) 資料(D)	視窗(₩)	説明(H)	輸入需	要解答的問題	•	-8×
i 🗋 🛛	🗧 🔙 🛛 🔓	3 19 - 🚺		新細明體	•	12 - E	IU	= = =	\$ 1	🖂 🕶 🌺 🕶	<u>A</u> - 関
	A1	-	fx 0	.005							
	А	В		D	Е	F	G	Н	Ι	J	
2390	-0.054	0.995	-0.106	1.998							
2391	-0.047	0.997	-0.094	1.992							
2392	-0.044	0.998	-0.086	1.996							
2393	-0.045	0.998	-0.073	1.994							
2394	-0.033	0.997	-0.063	1.996							
2395	-0.028	0.997	-0.052	1.996							
2396	-0.023	0.997	-0.042	1.996							
2397	-0.019	0.998	-0.03	1.995							
2398	-0.012	0.998	-0.02	1.995							
2399	-0.01	0.997	-0.012	1.982							
2400	0.001	0.997	0.002	1.996							
2401	L										
	M <u>curvel</u>						 ▲				
就緒							加總=-3.6	93			/

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:

- ISaGRAF - DEMO_71 - Programs										
File	<u>File Make Project Tools Debug Options H</u> elp									
		8 🗓	D 🗈	1	X 🖡		₿	~	🛄 🐉	
Begir	n:	H	🖻 LD1							
	🗯 ST1									
		E	🖹 Sim	_out						
Begir	n: ST1	(Structu	red Text))						

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	Туре	Attribute	Description
INIT	Boolean	Internl	初值設為 True
Go1	Boolean	Internl	若設為 True 則開始動作, addr 設為 21 (16 進位値為 15)
Stop1	Boolean	Internal	若設為 True 則停止動作, addr 設為 22 (16 進位値為 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 用來啓動, 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	i-8017H 上要記錄幾個 Channel 的電壓值
		常數	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	Integer	常數	S-256 / S-512 內要配置的 Byte 開頭 與 結尾
File_end1			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[07]	Integer	Input	為變數陣列, Dim 設為 8, 需連接到 i-8017H 的 Ch1. 到 8
Volt1[07]	REAL	Internal	爲變數陣列, Dim 設為 8, i-8017H的 Ch1. 到 8 轉換後的電
			壓値
i8024[03]	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,初値為"無動作",addr 設為 41 (16 進
			位值為 29)
Str1	Message	Intenal	長度 255, 程式內部使用

IO 連結:

ISaGRAF - DEMO_71 - I/O connection	
<u>File Edit T</u> ools <u>Options H</u> elp	
🖴 📼 🗟 🎾 💼 🗘 🕂 🕞 🖀	
0 ► Im ref = 121	
1 ava_num_S256 (* Available num	ber of S2:
2 ⊫ i_8024 ~ →	
3 ⊨ i_8017h ~ +	
4	
5	
8 ■ s256_512	
🗖 📼 status 🛛 🗸 🗸	
9	
10 📼 push4key л +	
11 📼 show3led л+	
12	

LD 程式 – LD1



```
(* 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 已上載過該 File *)
if TMP_V <> -1 then
 TMP := S_FL_RST(1); (* 此時再將 State 設為 -1, 讓 PC 可重覆上載該 File *)
end_if ;
```

ISaGRAF 進階使用手冊, Feb.2007, V3.0

```
(* 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 ...' ;
    (* 啓動 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 \ge 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 的記時 *)
   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 ;
```