How to record i-8017H ’s Ch.1 to Ch.4 voltage input in S-256 / 512 in i-8437-80 or i-8837-80 ? The sampling time is one record every 0.05 second. The record period is 1 to 10 minutes. Then PC can download this record and display it as a trend curve diagram by M.S. Excel.

This demo is the “Demo_71” can run in i-8437-80 or in i-8837-80 (80MHz) . The controller driver should be version of 3.19 or later version.

> New drive: http://www.icpdas.com/products/PAC/i-8000/isagraf-link1.htm
> “demo_71.pia” resides at I-8000 CD-ROM:\napdos\isagraf\8000\demo\ or
> ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/8000/demo/ or
> www.icpdas.com – FAQ – Software – ISaGRAF – FAQ058

i-8437-80 and i-8837-80 controller ‘s CPU is running at 80MHz . The speed is about 2 to 4 time faster than -8437 and i-8837 (40MHz CPU). So it can record minimum to 25 milli-second sampling data. While using i-8437 and i-8837 (40MHz) , it can record only 100 ms or above sampling data. This minimum sampling time depends on your ISaGRAF program ‘s PLC scan time. If the PLC scan time is large, like 200ms, then you can do sampling only larger than 200 ms.

How to test this demo ?

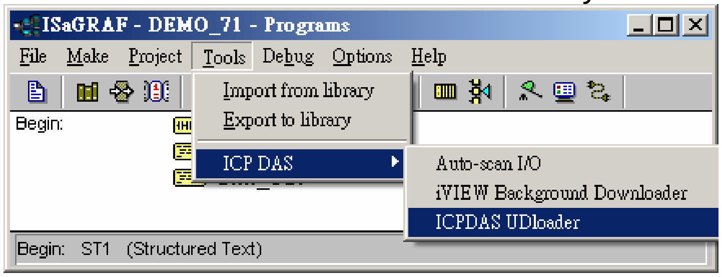
You may run VB 6.0 - “Demo_6” in your PC to on-line control this i-8837-80 via ethernet . Or just push the pushbutton on the front panel of the i-8437-80 / 8837-80 to start / stop it.

1. Please plug one i-8024 in i-8437-80 ’s Slot 2, one i-8017H in Slot 3. (Note: the left-most I/O slot No. of the i-8xx7 is 0 , not 1) .Then connect Ch1. to Ch.4 voltage output of i-8024 to Ch1. to Ch.4 of i-8017H. Then power up this i-8437-80.

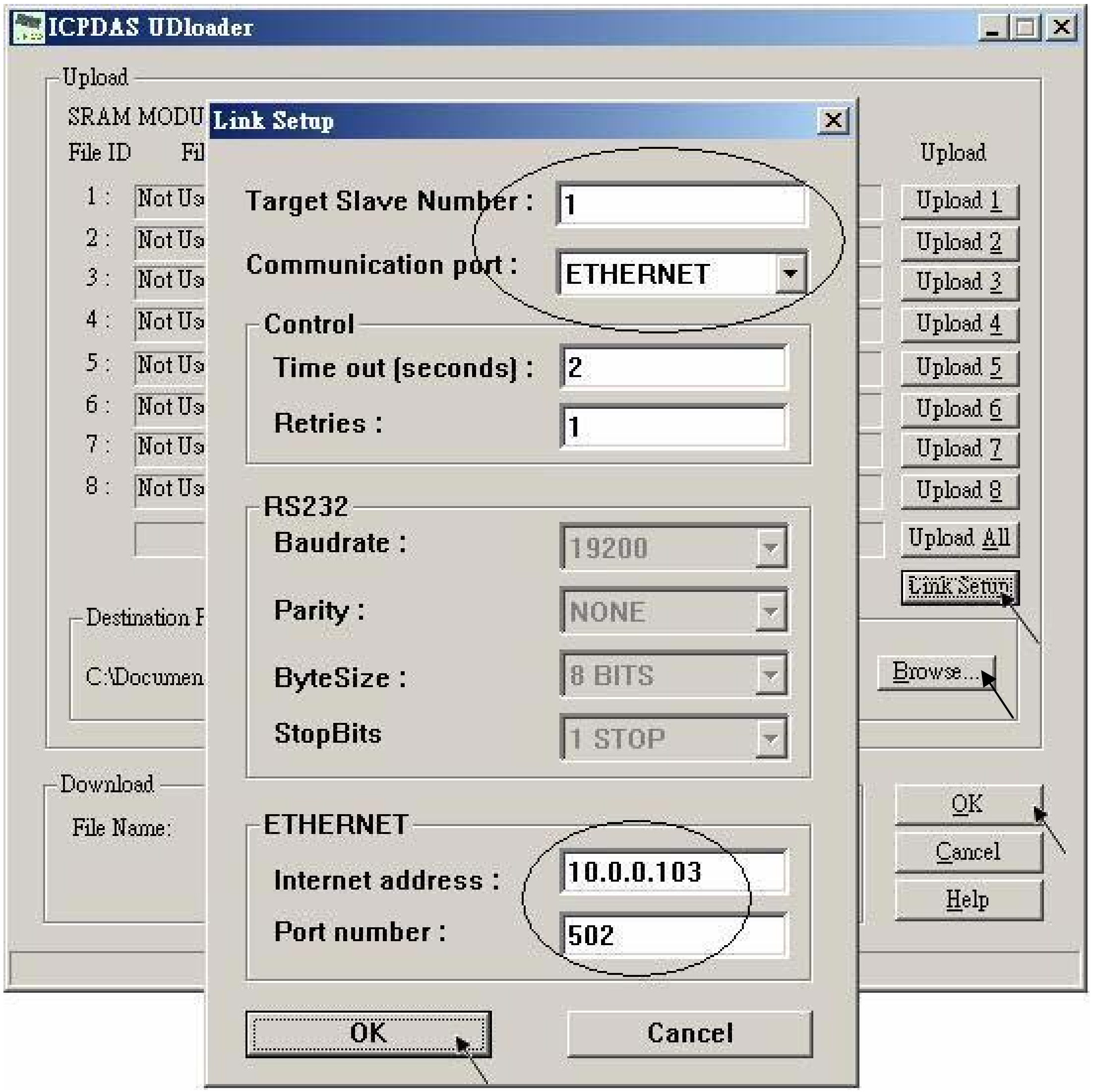
2. Download the ISaGRAF project - “Demo_71” to the i-8437-80 .

3. At run time, you may press pushbutton 1 to start recording. Then it will record data during 2 minutes. You can see the displayed number on the front panel decreasing to 0 and blinking. If recording is finished, the 3 Leds on the front panel will blink and the displayed number will be 0. To stop at anytime, just press pushbutton 4 once.

4. Whe recording is finished, please run UDLoader in your PC to upload the record file in the S-256 / 512 to PC. If your PC is currently running ISaGRAF workbench, please run “Tools” – “ICP DAS” – “ICPDAS UDloader” . Then you will see the windows listed in step (5) .



If your PC is not running ISaGRAF workbench, please copy i-8000-CD:\napdos\isagraf\some_utility\ “udloader.exe” to for example PC windows ‘s desktop. Then please run it. Set proper “Link Setup” (If click on “Browse” , you may modify the file upload location path). Then click on “Ok” and “Ok” to save this setting, then run it again.



Target Slave Number : 1

Communication port : ETHERNET

Control

Time out (seconds) : 2

Retries : 1

RS232

Baudrate : 19200

Parity : NONE

ByteSize : 8 BITS

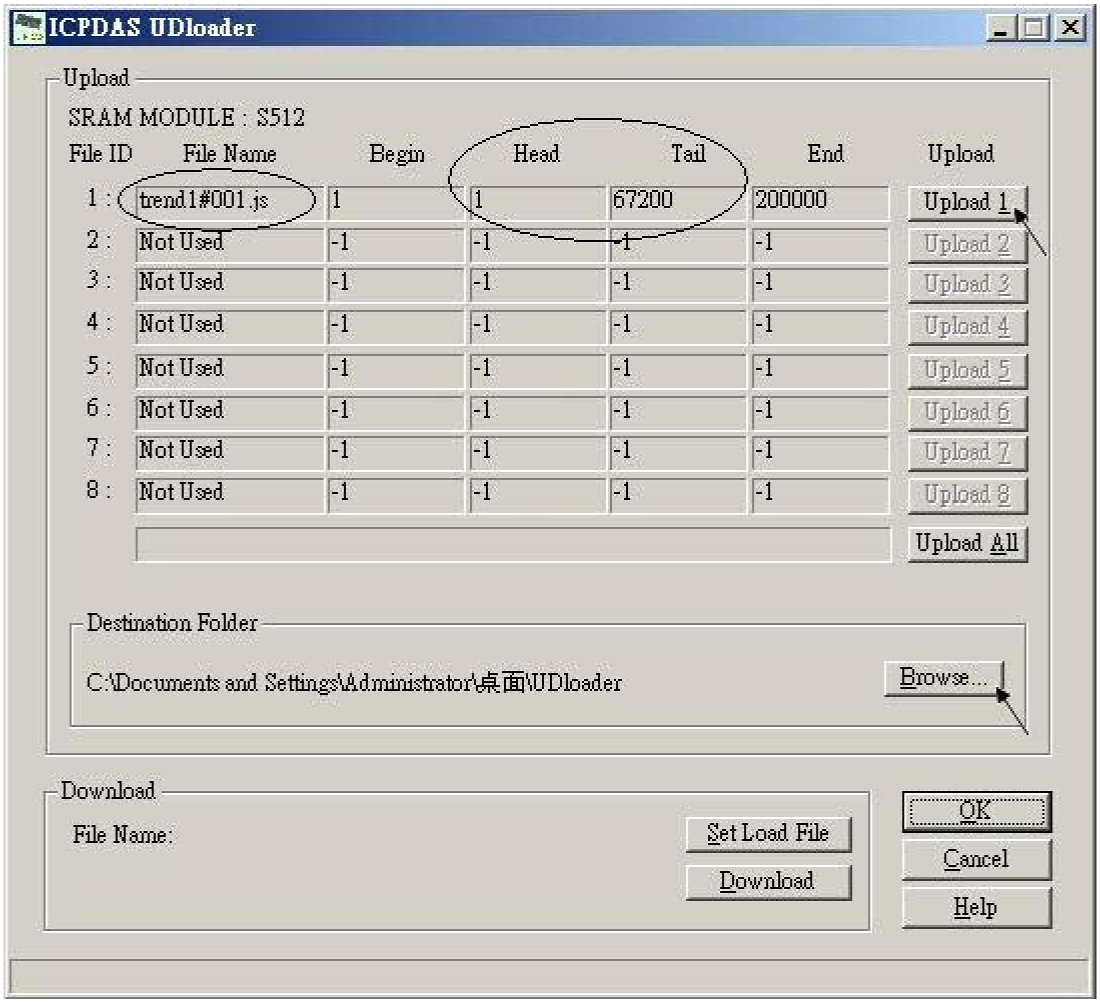
StopBits 1 STOP

ETHERNET

Internet address : 10.0.0.103

Port number : 502

5. If the controller is well connected, you will see a File Name displayed on the below window. Value of “Head” and “Tail” is the current size of the record file in the S-256 / S-512. The below example shows 67,200 bytes. (Note: S-256 / S-512 has size limitation, please refer to Chapter 10.3). you may modify the file upload location path by click on “Browse …” . Then please click on “Upload 1” to upload this record file to your PC.


ICPDAS UDloader

Upload

SRAM MODULE : S512

File ID	File Name	Begin	Head	Tail	End	Upload
1 :	trend1#001.js	1	1	67200	200000	Upload 1
2 :	Not Used	-1	-1	-1	-1	Upload 2
3 :	Not Used	-1	-1	-1	-1	Upload 3
4 :	Not Used	-1	-1	-1	-1	Upload 4
5 :	Not Used	-1	-1	-1	-1	Upload 5
6 :	Not Used	-1	-1	-1	-1	Upload 6
7 :	Not Used	-1	-1	-1	-1	Upload 7
8 :	Not Used	-1	-1	-1	-1	Upload 8

Destination Folder

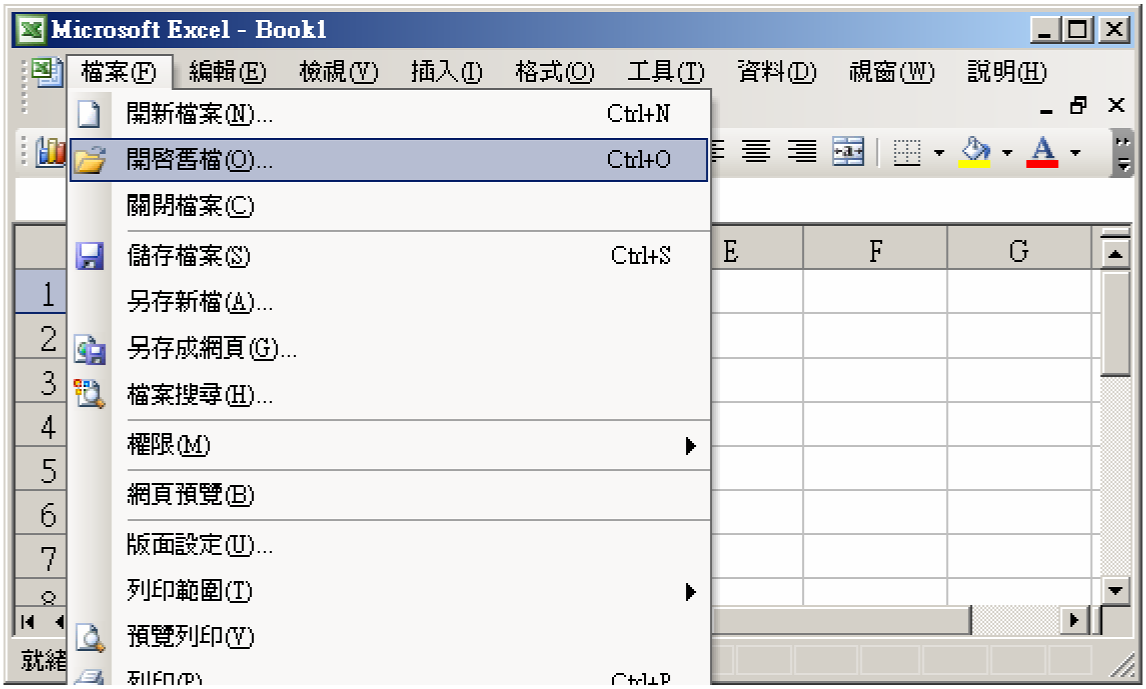
C:\Documents and Settings\Administrator\桌面\UDloader

Download

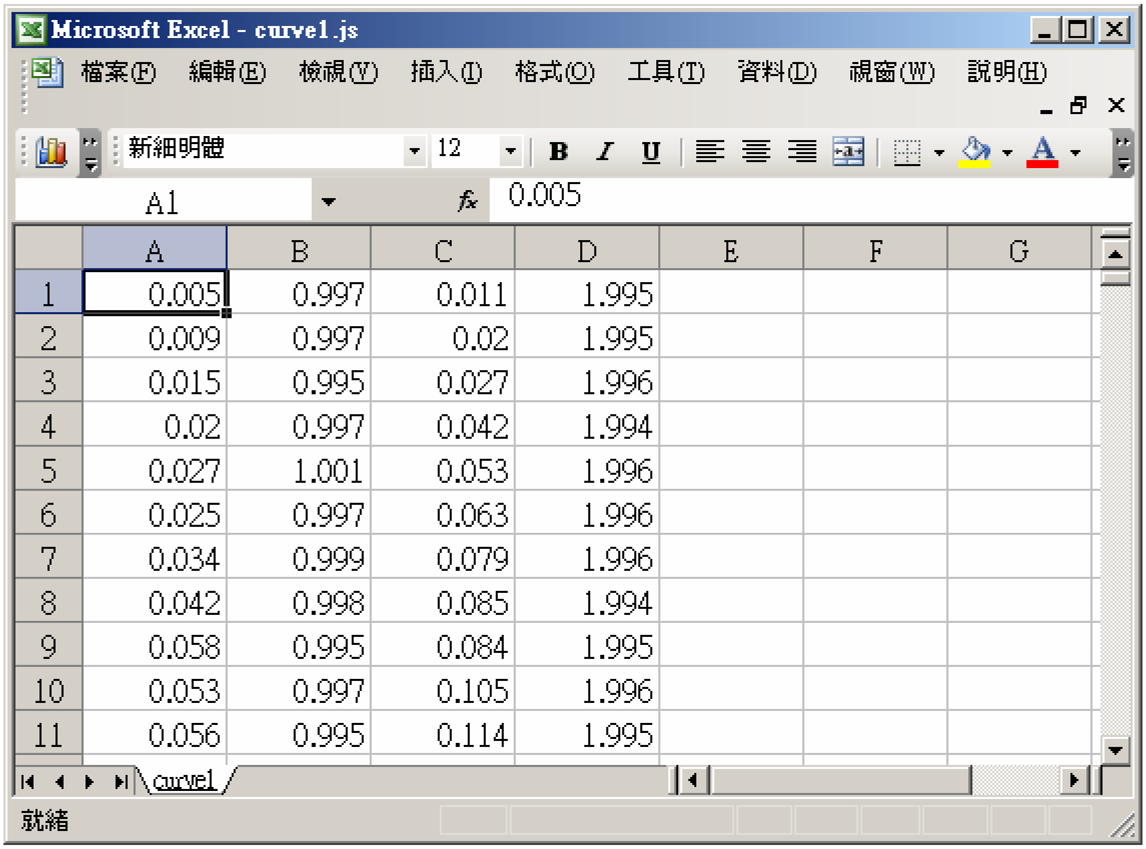
File Name:

Then you may check if the record file is upload to your PC at the same path.

6. Then please open this record file - “trend1.js” on M.S. Excel.

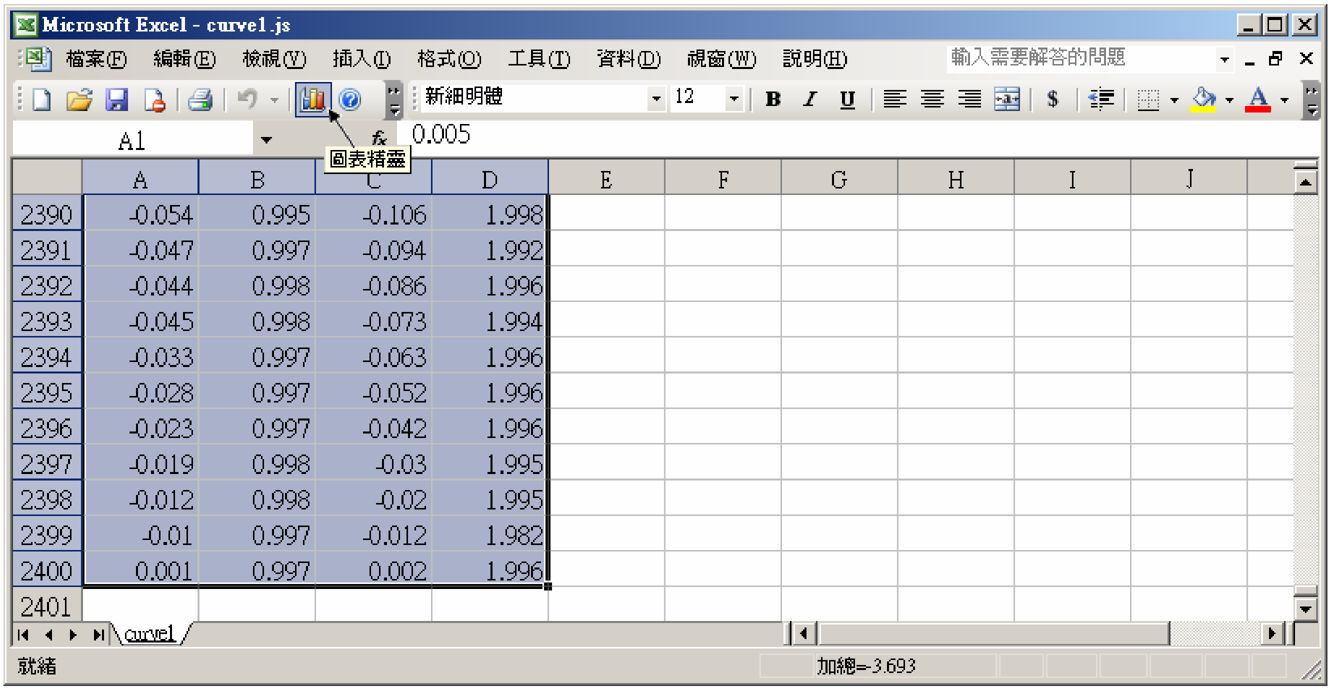


Please click on the first data at the left-top position. Then press and hold in “Shift” , and at the same time press “Ctrl” – “End” . You will see all data been selected.


Microsoft Excel - curve1.js
檔案(F) 編輯(E) 檢視(V) 插入(I) 格式(O) 工具(T) 資料(D) 視窗(W) 說明(H)
新細明體 12
B I U
A1
fx 0.005
curve1
就緒

	A	B	C	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			

Then click on 📊

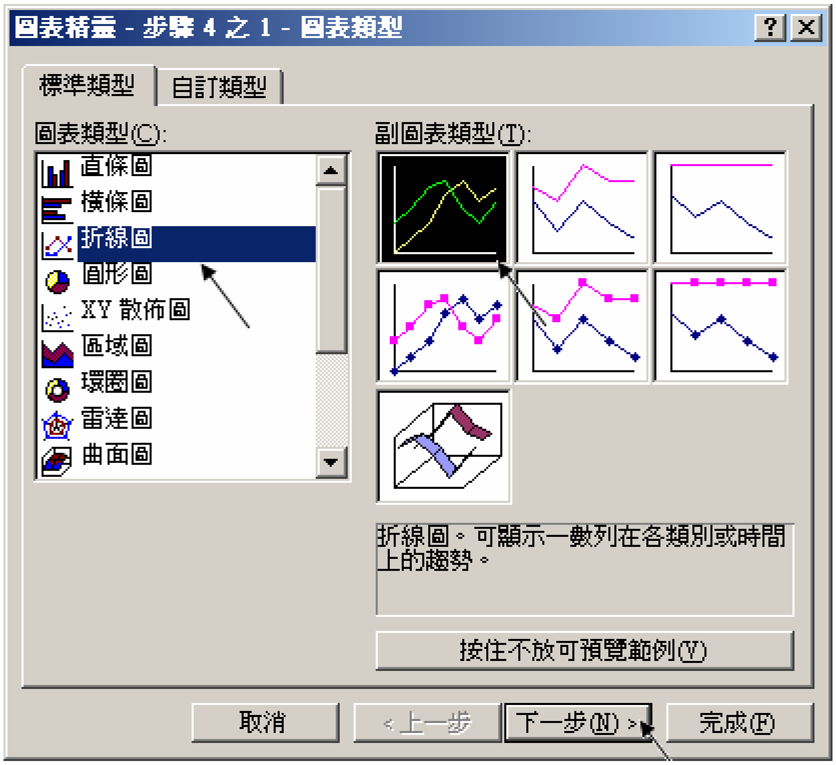

Microsoft Excel - curve1.js
檔案(F) 編輯(E) 檢視(V) 插入(I) 格式(O) 工具(T) 資料(D) 視窗(W) 說明(H)
輸入需要解答的問題
新細明體
12
圖表精靈
curve1

A1 0.005

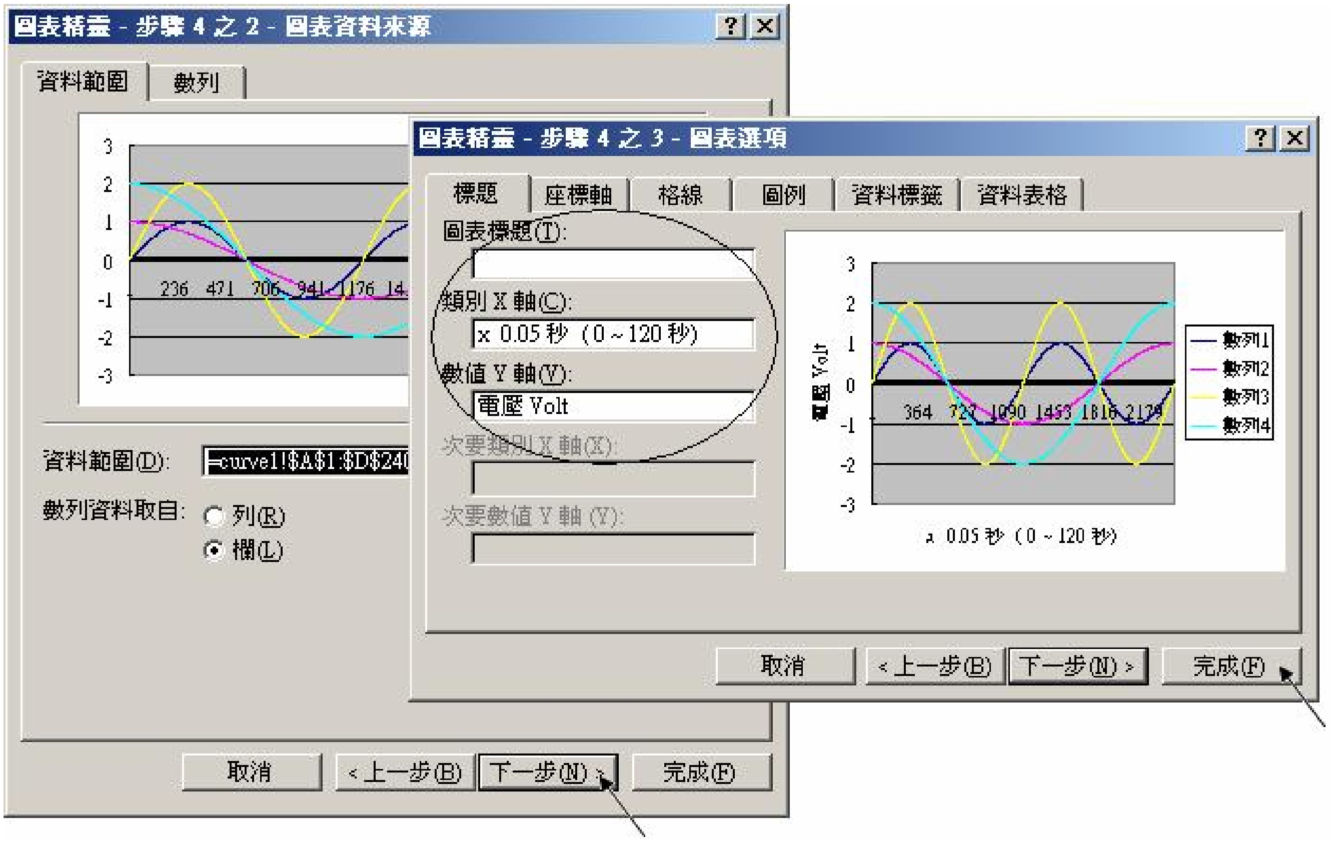
	A	B	C	D	E	F	G	H	I	J
2390	-0.054	0.995	-0.106	1.998						
2391	-0.047	0.997	-0.094	1.992						
2392	-0.044	0.998	-0.086	1.996						
2393	-0.045	0.998	-0.073	1.994						
2394	-0.033	0.997	-0.063	1.996						
2395	-0.028	0.997	-0.052	1.996						
2396	-0.023	0.997	-0.042	1.996						
2397	-0.019	0.998	-0.03	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										

就緒 加總=-3.693

Please select the correct diagram on the left-hand side. And check the left-top type on the right-hand side. Then go Next .



折線圖。可顯示一數列在各類別或時間上的趨勢。



圖表精靈 - 步驟 4 之 2 - 圖表資料來源

資料範圍 | 數列

資料範圍(D): =curve1!$A$1:$D$240

數列資料取自: ○ 列(R) ● 欄(L)

取消 | < 上一步(B) | 下一步(N) > | 完成(F)

圖表精靈 - 步驟 4 之 3 - 圖表選項

標題 | 座標軸 | 格線 | 圖例 | 資料標籤 | 資料表格

圖表標題(T):

類別 X 軸(C): x 0.05 秒 （0～120 秒）

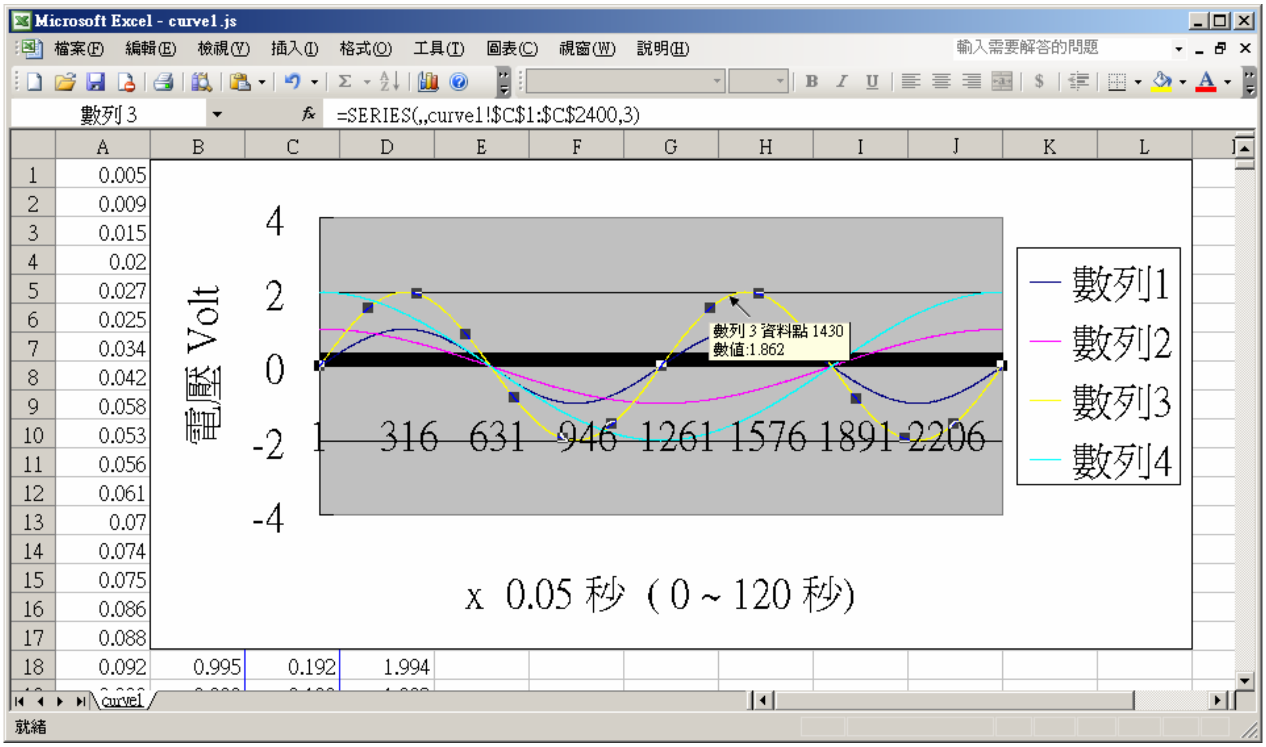
數值 Y 軸(V): 電壓 Volt

次要類別 X 軸(X):

次要數值 Y 軸(Y):

取消 | < 上一步(B) | 下一步(N) > | 完成(F)

By the procedure, you will get the trend curve as the below window. You can modify its size, or check at any trend line. If you move your mouse to point at some position at the trend line, the related data is shown.



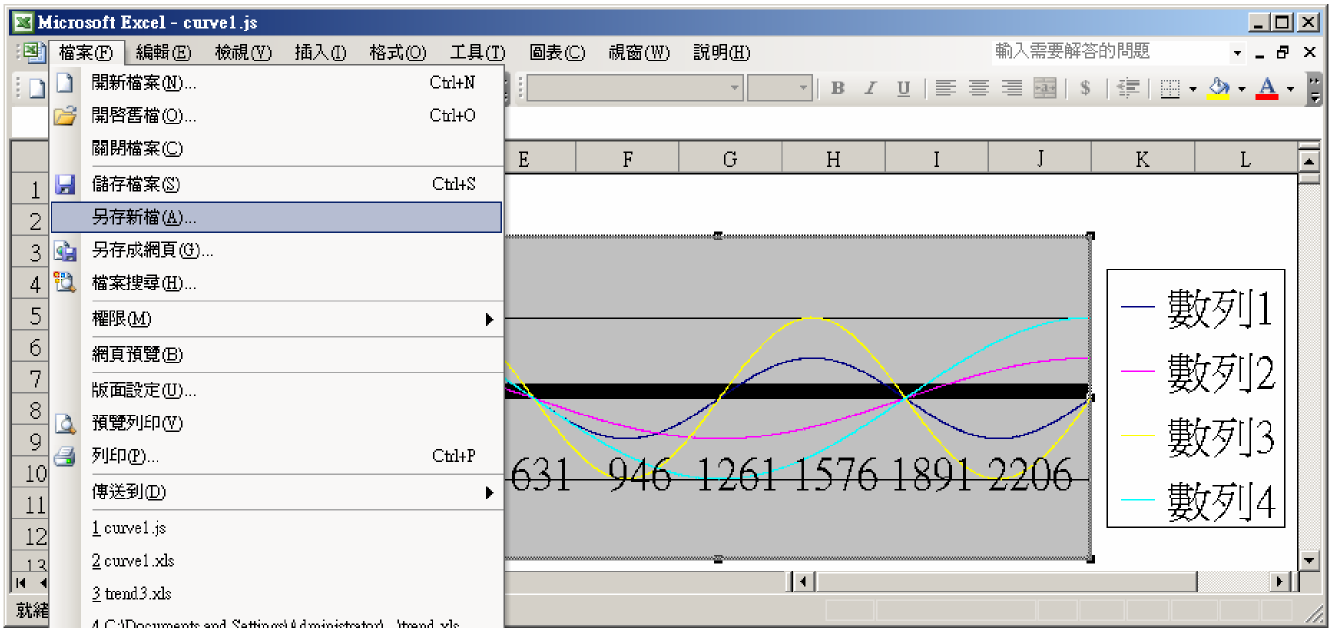
Microsoft Excel - curve1.js

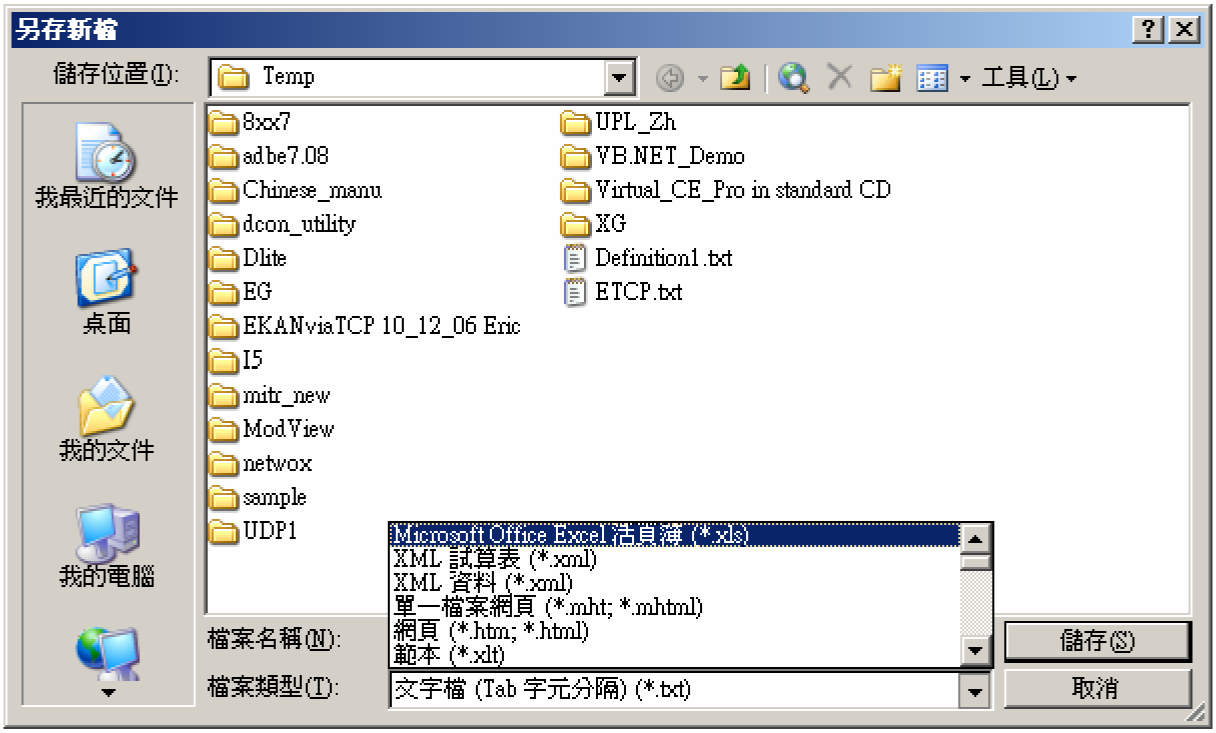
數列 3 | =SERIES(,,curve1!$C$1:$C$2400,3)

	A	B	C	D
1	0.005			
2	0.009			
3	0.015			
4	0.02			
5	0.027			
6	0.025			
7	0.034			
8	0.042			
9	0.058			
10	0.053			
11	0.056			
12	0.061			
13	0.07			
14	0.074			
15	0.075			
16	0.086			
17	0.088			
18	0.092	0.995	0.192	1.994

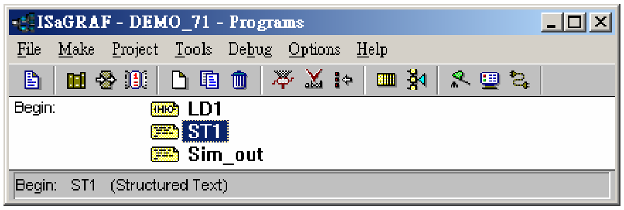
數列 3 資料點 1430
數值:1.862

Please save this trend curve diagram as a “Microsoft Office Excel (*.xls)” format. Then at any later time, you can open it to display the trend curve directly.





ISaGRAF project architecture:


ISaGRAF - DEMO_71 - Programs
File Make Project Tools Debug Options Help

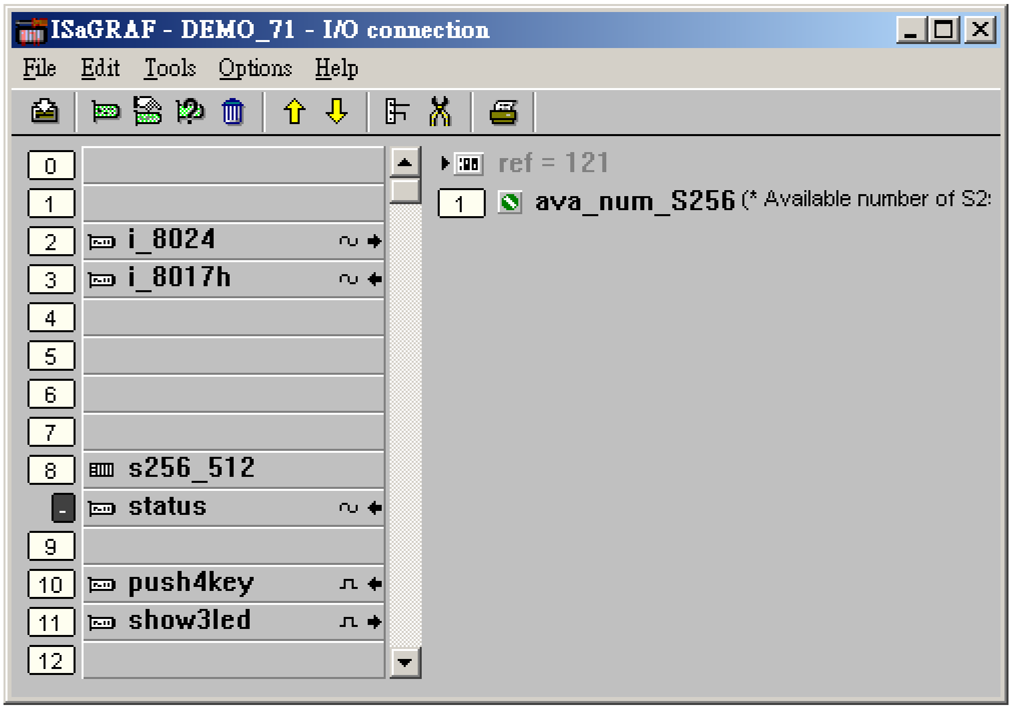
Begin: LD1
ST1
Sim_out

Begin: ST1 (Structured Text)

Variable : Please refer to Chapter 2.6 or FAQ-039 for description of “Variable array”

Name	Type	Attribute	Description
Go1	Boolean	**Internl**	Set as True to start, **addr defined as 21** (Hex. is 15)
Stop1	Boolean	**Internal**	Set as True to stop, **addr defined as 22** (Hex. is 16)
TMP	Boolean	**Internal**	Internal use
INIT	Boolean	**Internal**	Init as True
L1 , L2 , L3	Boolean	**Output**	connect to show3Led ’s Ch.1 to Ch.3
K1 , K4	Boolean	**Input**	Connect to push4key ‘s Ch.1 to Ch.4 Push K1 to start recording. K4 to stop
To_Blink	Boolean	**Internal**	To control blinking of the number on the front pannel
MUM_CH	Integer	**Constant**	How many chanels in i-8017H to record ? We use 4 channels in this demo (Ch.1 to 4)
Ava_num_s256	Integer	**Input**	Connect to “S256_512” ‘s Ch.1 . if value is 0, it means can not find S-256 / 512
Current_pos1	Integer	**Internal**	Current operating byte address in S-256 / 512
Len1 , TMP_V	Integer	**Internal**	Internal use
File_begin1 File_end1	Integer	**Constant**	The Begin & End byte address in the S-256 / S-512 allocated for the record file
STEP1	Integer	**Internal**	Recording state. 0:No action , 1:recording , 2:finished
Period1	Integer	**Internal**	How long to record ? unit is minute, **addr as 3**
Interval1	Integer	**Internal**	How long to save a record ? unit is ms, **addr as 1**
Total_record1	Integer	**Internal**	How many records in this recording action ? This value is calculated by the IsaGRAF program automatically. **addr declared as 5**
Record_cnt1	Integer	**Internal**	Current finished record count. **addr declared as 7**
ii	Integer	**Internal**	To use in “for” loops
i8017H[0..7]	Integer	**Input**	Variable array, Dim as 8. link to i-8017H ‘s Ch1 to Ch. 8
Volt1[0..7]	REAL	**Internal**	Variable array, Dim declared as 8. The voltage value converted from “i8017H[0..7]”
i8024[0..3]	Integer	**Output**	Variable array, Dim declared as 4. link to i-8024 ‘s Ch1 to Ch. 4
T1	Timer	**Internal**	For counting time
T1_next	Timer	**Internal**	The time to get and save next record
T1_Interval	Timer	**Internal**	The interval time between two record
Msg1	Message	**Internal**	Operation state message, Len is 255, init as “No Action now”, **addr as 41** (Hex. is 29)
Str1	Message	**Intenal**	Len is 255, internal use

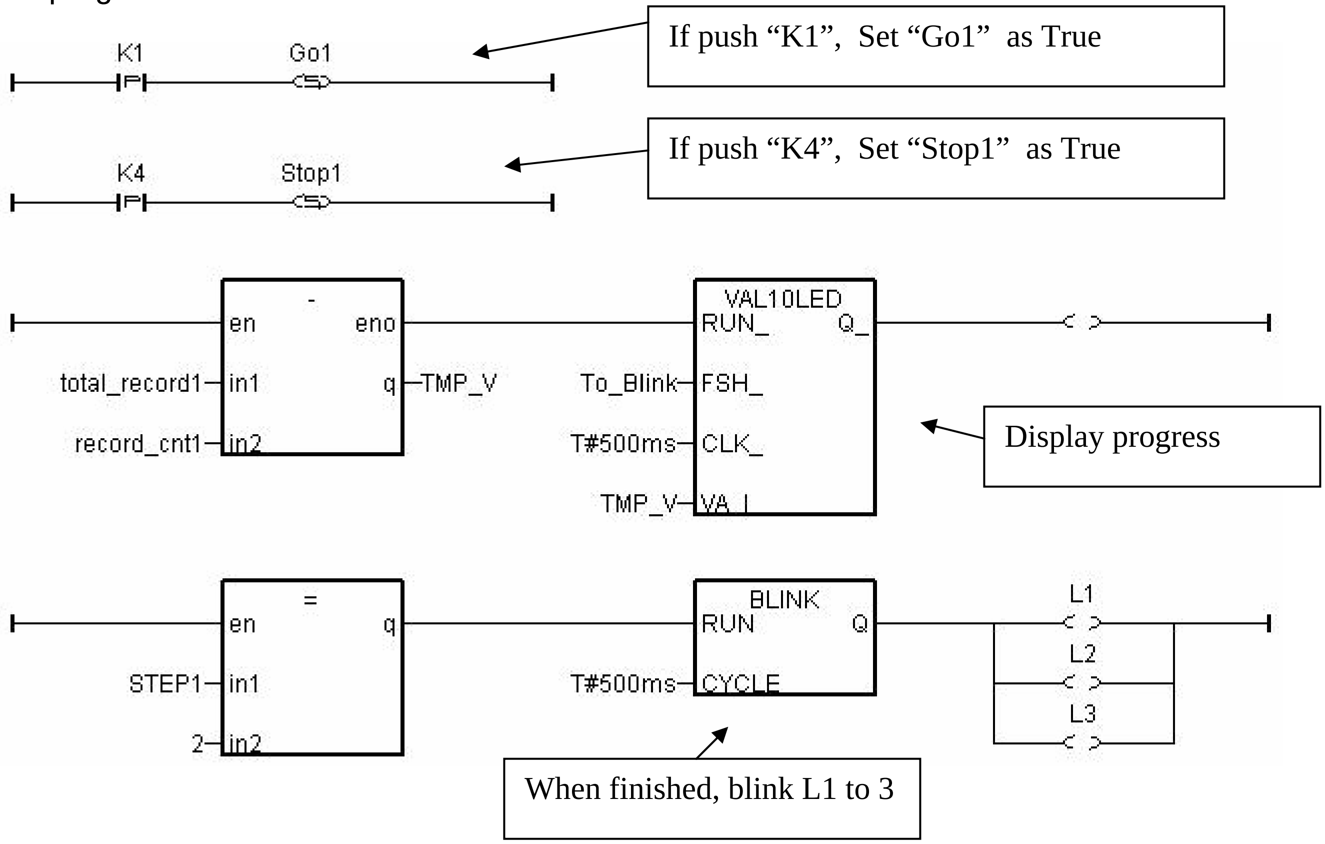
IO connection:



0	
1	
2	i_8024
3	i_8017h
4	
5	
6	
7	
8	s256_512
-	status
9	
10	push4key
11	show3led
12	

ref = 121
1 ava_num_S256 (* Available number of S2

LD program – LD1



If push “K1”, Set “Go1” as True

If push “K4”, Set “Stop1” as True

Display progress

When finished, blink L1 to 3

ST program – Sim_out

```
-----------------------------------------------------------------------------------------------------------------
(* Output i-8024 ‘s Ch1  to Ch4 as different voltage curve *)

(*   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 program – ST1

```
-----------------------------------------------------------------------------------------------------------------

if  INIT  then

  INIT := FALSE  ;      (* set as False to only do it once at 1st PLC scan *)

  if   ava_num_S256 = 0  then   (* S256 / S512 is not installed in i-8xx7, return *)
    Msg1  :=  'S256 / S512 is not installed in i-8xx7 controller !' ;
    Return ;
  end_if  ;

  (* Allocate S256/512 memory of byte No.1 to 200,000 for file ID = 1 , name='trend1.js' *)
  TMP  :=  S_FL_INI( 1 , 'trend1.js' , File_begin1 , File_end1) ;

  TMP  :=  S_FL_AVL( 1 , -1 , -1 ) ;    (* Init file content as No data at the beginning *)

end_if  ;

if   ava_num_S256 = 0  then   (* S256 / S512 is not installed in i-8xx7, return *)
  return ;
end_if ;

(* If stop command is gived *)
if  Stop1  then
  Stop1 := False ;
  STEP1 := 0 ;    (*  0: no action *)
  TStop(T1)  ;     (* stop T1 *)
  T1 := T#0s ;
  Msg1 := 'User stop recording !' ;
  To_Blink := FALSE ;  (* Set as FALSE not to blink the display value *)
end_if ;
```

```
(* Get file status in S256 or S512 *)
(* -1: PC hasn't load the file yet *)
(* others: the end byte No. that PC has load the file *)
TMP_V  :=  S_FL_STS(1) ;
if   TMP_V <> -1  then      (* PC has load the file *)
 TMP  :=  S_FL_RST(1)  ;      (* reset status to -1 ( PC hasn't load the file yet) *)
end_if  ;


(* If start command is gived *)
if  Go1  then

  Go1 := False ;

  (* STEP1 :  0: no action , 1: recording , 2: recond finished *)
  if   STEP1=1   then

    (* It is still recording now *)
    Msg1  :=  'It is still recording now ...'  ;

  else

    (* Check interval valid or not *)
    (* we assume 25 to 10000 ms is valid in this example *)
    (* If your average PLC scan time is larger, for example, near 20 ms,
       Please use Interval larger than 25 ms. Or the record time won't be correct *)
    if   (Interval1 < 25)  or  (Interval1 > 10000)  then
     Msg1  :=  'Wrong Interval value, it should be in 25 to 10000 milli-second !' ;

    (* Check period valid or not *)
    (* we assume 1 to 10 minute is valid in this example *)
    Elsif  ( Period1 < 1)   or  (Period1 > 10)  then
     Msg1  :=  'Wrong Period value, it should be in 1 to 10 minute !' ;

    else

     (* parameter is correct, start recording *)
     total_record1 := (Period1 * 60000) / Interval1 ;  (* calculate total record number *)
     record_cnt1 := 0 ;  (* reset current record count as 0 *)
     STEP1 := 1 ;  (* set step as 1:recording *)
     Msg1  :=  'Recording now ...'  ;

     (* start ticking T1 from 0 second *)
     T1 := T#0s ;
     T1_Interval  :=  TMR(Interval1) ;
     T1_next := T1 + T1_Interval ;
     TStart(T1) ;   (* ticking now *)
     Current_pos1 := 1 ;  (* reset current data position in S256/S512 as 1 *)
```

```
         To_Blink := TRUE ;  (* Set as TRUE to blink the display value *)

       end_if  ;

  end_if  ;

end_if  ;

 (* in reconrding state *)
if    STEP1=1    then

  (* store one record *)
  if   T1 >=  T1_next   then

    (* Re-calculate next T1 *)
    T1_next  :=  T1_next  +  T1_Interval ;

    (* T1 will be overflow after T#23h59m59s999ms, so reset it at T#20h *)
    if   T1 >= T#20h   then
      T1  :=   T#0s ;
      T1_next   :=   T1 + T1_Interval ;
    end_if ;

    str1  :=  '' ;  (* init str1 as empty string *)
    for    ii  :=  0  to  NUM_CH-1  do

       (* convert i8017H analog input value to Volt value *)
       Volt1[ii]  :=  Real(i8017H[ii]) * 0.000305176 ;    (* 10.0 / 32768 = 0.000305176 *)
       str1  :=  str1 + Rea_Str2(Volt1[ii] , 3) +  '$09' ;     (* delimiter is <TAB> character *)

    end_for ;

    str1  :=  str1 + '$0D$0A'  ;  (* add <CR> <LF> at the end of each row *)
    Len1  :=  MLEN(str1) ;  (* get string length *)

    (*  data number larger than file's max. allocated memory *)
    if   (Current_pos1 + Len1 - 1)  >  File_end1   then
       STEP1 := 0;  (* 0: no action *)
       Msg1 := 'File allocated memory is not enough to hold the data ! '  ;
       Tstop(T1) ;
       T1 := T#0s ;
       To_Blink := FALSE ;  (* Set as FALSE not to blink the display value  *)
       Return ;
    end_if  ;

    TMP  :=  S_M_W( Current_pos1 , Len1 , str1 ) ;   (* write all bytes in str1 to S256/S512 *)
    Current_pos1  :=  Current_pos1 + Len1 ;          (* Current position move on *)
    TMP := S_FL_AVL( 1 , 1 , Current_pos1-1 ) ;     (* Re-caculate File 's Head & Tail *)
```

```
   (* Check if record number reach the end *)
    record_cnt1  :=  record_cnt1 + 1 ;  (* current record count plus 1 *)
    if   (record_cnt1 >= total_record1)   then
      STEP1  :=  2  ;       (* 2: recond finished *)
      Msg1 :=  'Record is finished ! You may download the record file to your PC now !' ;
      Tstop(T1)  ;
      T1  :=  T#0s   ;
      To_Blink  :=   FALSE ;  (* Set as FALSE not to blink the display value *)
    end_if ;

  end_if ;

end_if  ;
  ------------------------------------------------------------------------------------------------------------------
```