How to save the value of ISaGRAF variables to the Micro_SD memory in the WP-5xx7, WP-8xx7 and VP-25W7 PAC ?

User can download this paper and its example programs (wpdmo56.pia , wpdmo56a.pia , wpdmo56b.pia , wpdmo56c.pia , wpdmo56d.pia , wpdmo56e.pia) from the following web.
http://www.icpdas.com/faq/isagraf.htm > FAQ-155 .
wpdmo56 : Save 17 REAL variables in "\Micro_SD\data56.txt" (max. 255 REALs)
wpdmo56a : Save 17 REAL and 2 BOOL variables in "\Micro_SD\data56.txt" (max. 255)
wpdmo56b : Save 25 integer variables in "\Micro_SD\data56.txt" (max. 255 integers)
wpdmo56c : Save 25 integer and 2 BOOL variables in "\Micro_SD\data56.txt" (max. 255)
wpdmo56d : Save 17 REAL, 2 BOOL and 10 integer variables in "\Micro_SD\data56F.txt" and "\Micro_SD\data56.txt" (max. 255 REALs and "integer+BOOL" <= 255)
wpdmo56e : Save max. 1024 REAL and max. amount of "integer+BOOL" is 1024 .

This paper shows the way to save the ISaGRAF variables to the Micro_SD memory in the WP-5xx7, WP-8xx7, and VP-25W7 PAC. The PAC will restore the last value of variables when power up. And at any time when the value is modified, it will save the last value automatically. This paper is very useful for the WP-5xx7 PAC (like the WP-5147) because its default hardware has no battery backup memory (so it can not use the "new retain variables" if the XW-608 is not purchased and installed inside it).

Important notice:
1. **Please store your application programs and data files in the \Micro_SD , don't store them in the \System_disk**. That is because the \System_Disk is using Nor Flash memory. Its size is small and major purpose is for storing OS, some basic utilities and DLL . The Nor Flash memory is not good for frequently updating files. If update files frequently in the \System_Disk (for examaple, update a file every 1 to 5 seconds, then it will be about ten thousand more updates in one day), the data or files in the \System_disk may crush or lost for some days or months later.
2. To read / write file in the \System_Disk or \Micro_SD memory take lots of CPU time, please do not read / write it frequently. If user read / write file in every PLC scan cycle, the PLC scan time will become very large and the PAC will perform badly. If user need fast retain ,refer to http://www.icpdas.com/faq/isagraf.htm > FAQ-074 for the "New retain variables" .

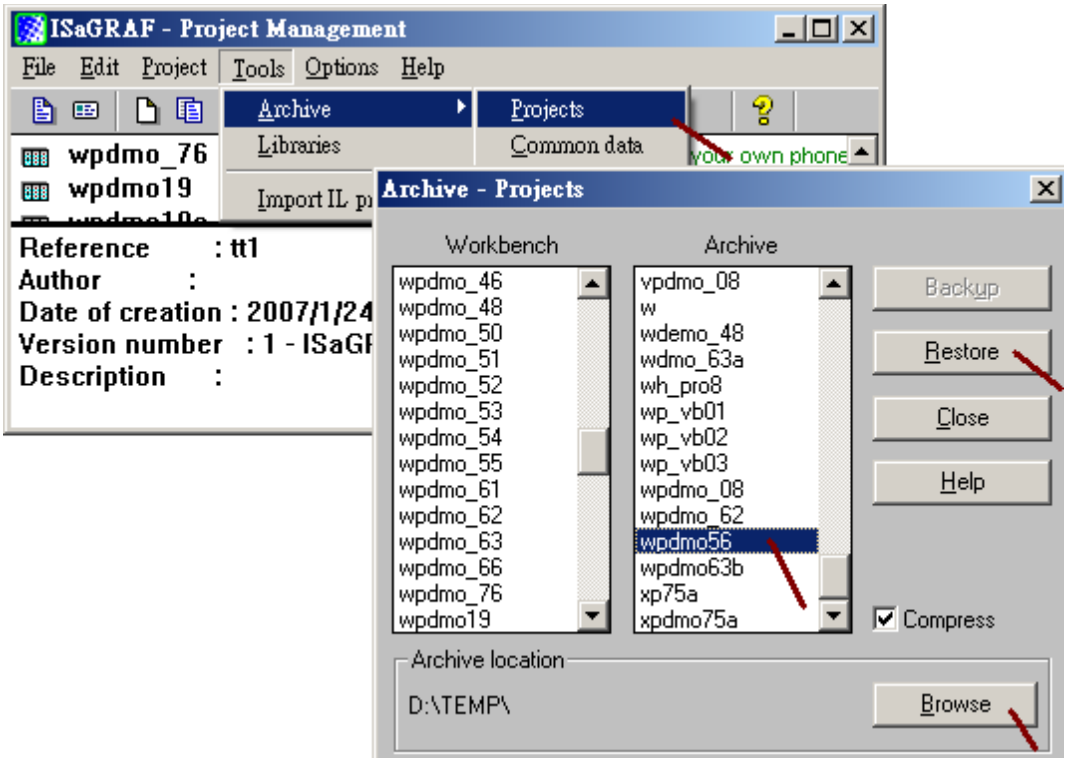
The example programs - wpdmo56.pia , wpdmo56a.pia , wpdmo56b.pia , wpdmo56c.pia , wpdmo56d.pia and wpdmo56e.pia all use the "array variables". You need to setup your PC / ISaGRAF before you can use them. First close all of the ISaGRAF windows. Then open the "isa.ini" file in the directory where the ISaGRAF is installed (it is "C:\ISAWIN\EXE\" normally). Then add two rows on the top of the "isa.ini" file as the following and save this file.
For more information about the "array variables", please refer to
http://www.icpdas.com/faq/isagraf.htm > FAQ-039

```
[DEBUG]
arrays=1
```

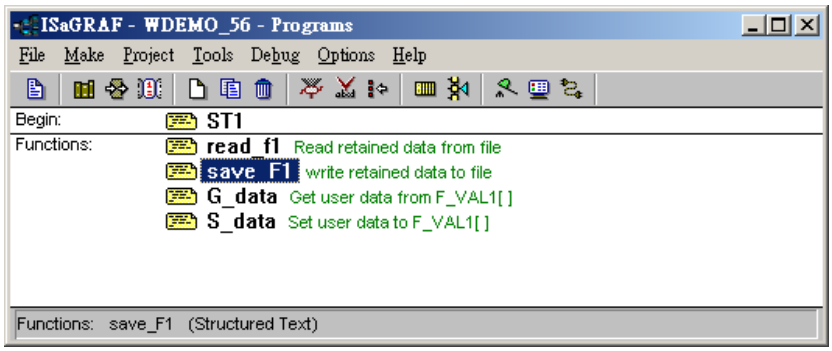
If functions of Msg_F , Msg_N , ARY_F_R, AFY_F_W are not found in your PC / ISaGRAF, download the "ICP DAS utilities for ISaGRAF" at http://www.icpdas.com/products/PAC/i-8000/isagraf.htm . Then run "setup.exe" to restore them to your ISaGRAF workbench (The installation takes about 10 minutes).

This paper only describes about the “wpdmo56” program. The “wpdmo56” reads 1 to 255 REAL values from “\Micro_SD\data56.txt” to related ISaGRAF variables when the PAC is powered up . If this “data56.txt” doesn’t exist, all these 1 to 255 variable value will be inited as 0.0 . At run time, if any value of these variables is modified, all the 1 to 255 variables will be written automatically in the “data56.txt”. If the file doesn’t exist, this program will create it.

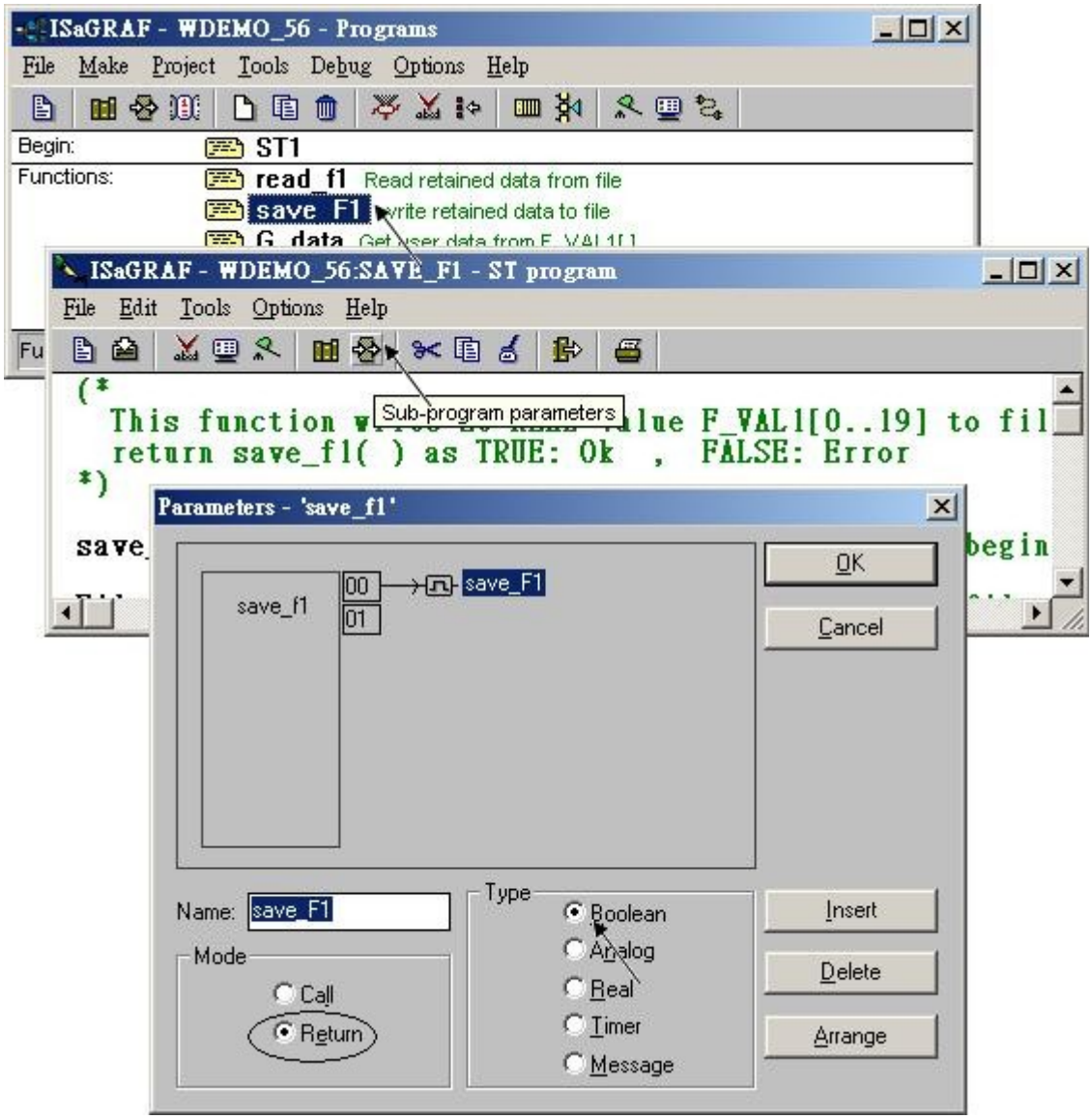
Restore the “wpdmo56” example project :



Project Architecture:
There are five ST programs in this “wpdmo56” project. Four of them are ISaGRAF user-defined functions – “reaf_f1” , “save_f1” , “G_data” and “S_data” .



Important note:
1. User may modify the constant value of “SIZE1” in the ISaGRAF “dictionary” window to a value between 1 to 255 according his own application.
2. Please also modify the “Dim” value of the “F_VAL1[ ]” and “Old_F_VAL1[ ]” variable array in the ISaGRAF “dictionary” window to the same value as the “SIZE1” . And also modify the “G_data” and “S_data” program.
3. There is one advantage of retaining value in the Micro_SD memory. The data file can be edited in a PC in advance. Then using the “ftp” utility to download it to the PAC. The file path name of this example is “\Micro_SD\data56.txt” . Then set the value of boolean variable “RE_LOAD” to TRUE, all related variables will update to the new value. The following ST programs are all declared as ISaGRAF functions. They are “read_f1” , “save_f1” , “G_data” and “S_data” . They all return a Boolean value. Please refer to below figure to declare function’s return-value type (more description is in the Chapter 15 of the “ISaGRAF User's manual”)



The “read_f1” and “save_f1” program use “local variables” as below .

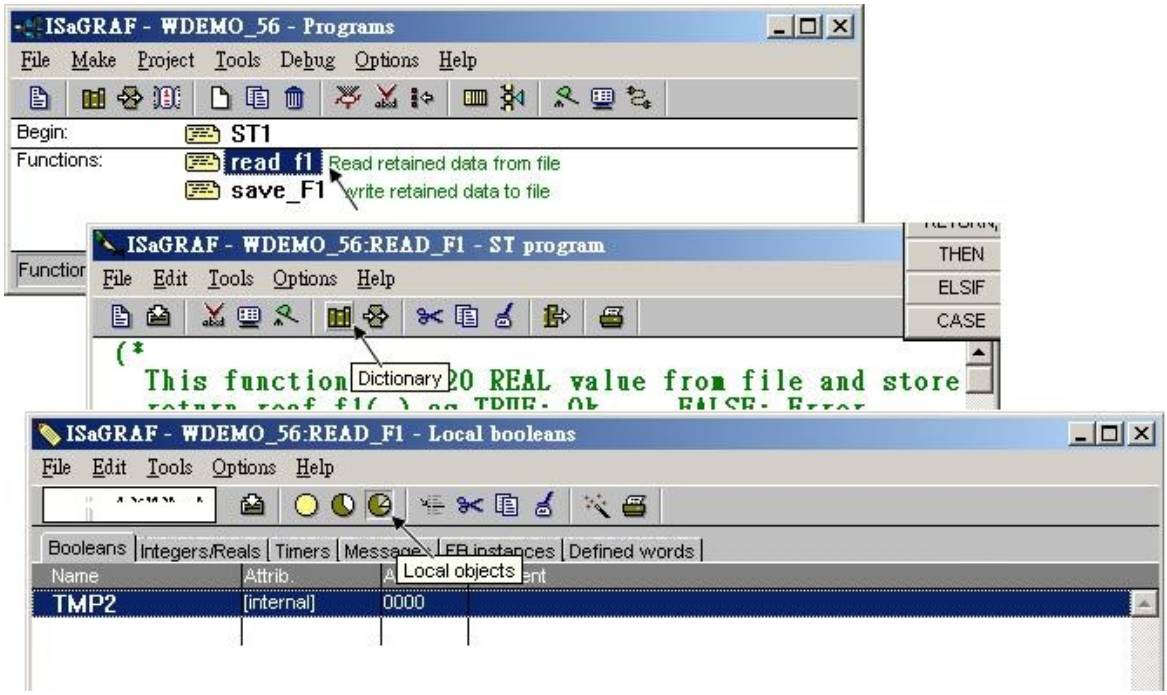
read_f1 :

Name	***Type***	***Attribute***	***Description***
TMP2	Bool	Internal	Internal use
ii2	Integer	Internal	Index of “for” loops
jj2	Integer	Internal	Index of “for” loops
num2	Integer	Internal	Internal use

save_f1 :

Name	***Type***	***Attribute***	***Description***
TMP2	Bool	Internal	Internal use
ii2	Integer	Internal	Index of “for” loops
jj2	Integer	Internal	Index of “for” loops
num2	Integer	Internal	Internal use

To declare “local variable”, please double click “read_f1” to get into this program. Then get into the “Dictionary” window. Then click on “Local objects” to declare them.



Global variables :

Name	***Type***	***Attribute***	***Description***
SIZE1	Integer	Constant	**Amount of retain variables. Can be 1 to 255 . Pls modify the "Dim" value of the "F_VAL1[ ]" and "Old_F_VAL1[ ]" to the same value as "SIZE1"**. Here we use "SIZE1" as 17
num_row1	Integer	Internal	How many rows in the file ? This value is automatically calculated by "SIZE1" . Each row should have 10 REAL values, except the last row.
Last_num1	Integer	Internal	How many data in the last row ? This value is automatically calculated by "SIZE1".
RE_LOAD	Bool	Internal	Set as True to read File once, init as TRUE
TMP	Bool	Internal	Internal use
Data_Ok1	Bool	Internal	TRUE means File Ok
Flag_to_save	Bool	Internal	If program want to save data, it will set this value to TRUE.
File_name1	Message	Internal	Len is 64, init as \Micro_SD\data56.txt
Msg1	Message	Internal	Len is 128, File processing state
str1	Message	Internal	Len is 255, Internal use
F_VAL1[0..16]	REAL	Internal	**Variable array, "Dim" should be init as the same value as "SIZE1"**
Old_F_VAL1 [0..16]	REAL	Internal	Old value of "F_VAL1[ ]" **Variable array, "Dim" should be init as the same value as "SIZE1"** .
NUM1	Integer	Internal	Get return of Msg_F( ), -1 means format error
File1	Integer	Internal	File ID
ii	Integer	Internal	Index of "for" loops
jj	Integer	Internal	Index of "for" loops
Data1 ~ Data5 And Data06 ~ Data17	REAL	Internal	**The User Data variable. Here we have 17 variables in the demo program**. User can declare them to different variable name. If name is modified, the "G_data" and the "S_data" program should be modified also.

ST program - ST1:

```
---------------------------------------------------------------------------------------------------------------
if  RE_LOAD  then      (* if RE_LOAD is TRUE, get retained data from file *)

 RE_LOAD  :=  FALSE ;   (* Set RE_LOAD as FALSE *)

 (* caculate number of rows and data number of the last row *)
 num_row1  :=  SIZE1 / 10 ;
 last_num1  :=  SIZE1 -  10  *  num_row1 ;
 if  last_num1  <>  0  then
   num_row1  :=  num_row1 + 1 ;    (* if last_row has data, num_row1 must plus 1 *)
 else
   last_num1  :=  10 ;
 end_if ;

 (* Get retained value from file when controller is powered up *)
 TMP  :=  read_F1( ) ;

 if  TMP = FALSE  then   (* Read file error or file not exist *)

  for  ii  :=  0  to  SIZE1 - 1  do
     F_VAL1[ ii ]  :=  0.0 ;     (* set all F_VAL1[ ] 's value as 0.0 *)
  end_for ;

  Data_Ok1  :=  FALSE ;    (* set data is not Ok *)
  Msg1  :=  'File : '  +  File_name1  +  ' not exist or data error ! or File is open now' ;

 else   (* Read data Ok *)

  Data_Ok1  :=  TRUE ;    (* set data is Ok *)
  Msg1  :=  'Get Retained data from file Ok ' ;

 end_if ;

 (* Update Old_F_VAL1[ ] *)
 for  ii  :=  0  to  SIZE1 - 1  do
  Old_F_VAL1[ ii ]  :=  F_VAL1[ ii ] ;
 end_for ;

 (* Get user data from F_VAL1[ ] when controller is just powered up *)
 TMP  :=  G_DATA( ) ;

end_if ;

(* At run time, Set user data to F_VAL1[ ] *)
TMP  :=  S_DATA( ) ;
```

```
(* At run time, test any value of F_VAL1[ ] is modified *)
for   ii  :=  0  to  SIZE1 - 1  do

 if  Old_F_VAL1[ ii ]  <>  F_VAL1[ ii ]  then     (* if any value is modified *)
   Flag_to_save  :=  TRUE  ;     (* now save command is given *)
   Old_F_VAL1[ ii ]  :=  F_VAL1[ ii ]  ;    (* Update Old_F_VAL1[ ] if it is modified *)
 end_if  ;

end_for  ;

(* if save command is given, it means value is modified *)
if   Flag_to_save  then

 TMP  :=  save_f1( )  ;    (* save data to file *)

 (* if save file failed, keep this save command *)
 if   TMP  =  FALSE   then
   Msg1  :=  'Can not save data to file. May be file is open now by WinPAC 's screen ! '  ;

 (* Save Ok, cancel this save command *)
 else
  Flag_to_save  :=  FALSE  ;     (* Set as "No save" at the beginning *)

 end_if  ;

end_if  ;
```

ST functions – G_data :

(* If any name of Data1 to Data17 is modified or value of “SIZE1” is modified, User must modify the below code *)

```
Data1    :=  F_VAL1[0] ;     (* get variable value from F_VA1L[0..16] *)
Data2    :=  F_VAL1[1] ;
Data3    :=  F_VAL1[2] ;
Data4    :=  F_VAL1[3] ;
Data5    :=  F_VAL1[4] ;
Data06   :=  F_VAL1[5] ;
Data07   :=  F_VAL1[6] ;
Data08   :=  F_VAL1[7] ;
Data09   :=  F_VAL1[8] ;
Data10   :=  F_VAL1[9] ;
Data11   :=  F_VAL1[10] ;
Data12   :=  F_VAL1[11] ;
Data13   :=  F_VAL1[12] ;
Data14   :=  F_VAL1[13] ;
Data15   :=  F_VAL1[14] ;
Data16   :=  F_VAL1[15] ;
Data17   :=  F_VAL1[16] ;
G_data  :=  TRUE  ;       (* function  returns TRUE *)
```

ST functions – S_data :

(*If any name of Data1 to Data17 is modified or value of “SIZE1” is modified, User must modify the below code *)

```
F_VAL1[0]  :=  Data1 ;    (* store variable value to F_VAL1[0..16] *)
F_VAL1[1]  :=  Data2 ;
F_VAL1[2]  :=  Data3 ;
F_VAL1[3]  :=  Data4 ;
F_VAL1[4]  :=  Data5 ;
F_VAL1[5]  :=  Data06 ;
F_VAL1[6]  :=  Data07 ;
F_VAL1[7]  :=  Data08 ;
F_VAL1[8]  :=  Data09 ;
F_VAL1[9]  :=  Data10 ;
F_VAL1[10]  :=  Data11 ;
F_VAL1[11]  :=  Data12 ;
F_VAL1[12]  :=  Data13 ;
F_VAL1[13]  :=  Data14 ;
F_VAL1[14]  :=  Data15 ;
F_VAL1[15]  :=  Data16 ;
F_VAL1[16]  :=  Data17 ;
S_data  :=  TRUE ;          (* function returns TRUE *)
```

ST functions - read_f1 :

```
---------------------------------------------------------------------------------------------------------------------
(*    This function read "SIZE1" number of REAL value from file and store them to F_VAL1[ ]
  return reaf_f1( ) as TRUE: Ok  ,  FALSE: Error  *)
read_f1  :=  FALSE  ;                    (* set as FALSE: Error at the beginning *)
File1  :=  f_wopen( File_name1 )  ;   (* Try to open file in Read & Write mode *)

if   File1 = 0  then      (* File doesn't exists *)
   return  ;    (* exit this function *)
end_if  ;

 (* max  "num_row1"  rows to read these "SIZE1" number of REAL values, Each row in the file
contains 10 REAL values *)
for   ii2   :=   0   to   num_row1 - 1   do

  if   f_eof( File1 )  =  TRUE   then   (* test if End_Of_File reached *)
    exit  ;       (* Reach End Of File,  exit “for” loop *)
  end_if  ;

  str1  :=  fm_read( File1 )  ;   (* Read one row as String (message) *)

  (* Convert this string to some REAL values and store them into No.1 Float array *)
  NUM1  :=  Msg_F( str1 , 1 ) ;

  (* if data number of last row is not correct *)
  if  ( ( ii2 = num_row1 – 1 )   and   ( NUM1  <>  last_num1 )  )  or

     (* non-last row must have 10 REAL values *)
     ( ( ii2  <>  num_row1 – 1 )   and   ( NUM1  <>  10 )  )     then

    (* error, it means the format is not correct REAL values or data number is not enough *)
    exit ;       (* exit for loop *)

  end_if;

  (* conversion Ok, store these REAL values to F_VAL1[ ] *)
  if   ii2  =  num_row1 - 1   then      (* last row *)
      num2  :=  last_num1  ;        (* last row has only "last_num1" number of data *)
  else
      num2 := 10 ;    (* non-last row has 10 data *)
  end_if ;
  (* Get these converted REAL values from No.1 Float array 's addr. 1 to 10 (or 1 to last_num1
for last row) *)
  for   jj2  :=  0   to   num2 - 1   do
      F_VAL1[ 10*ii2  +  jj2 ]   :=   ARY_F_R( 1 ,  jj2  +  1 )  ;
  end_for  ;

end_for  ;
```

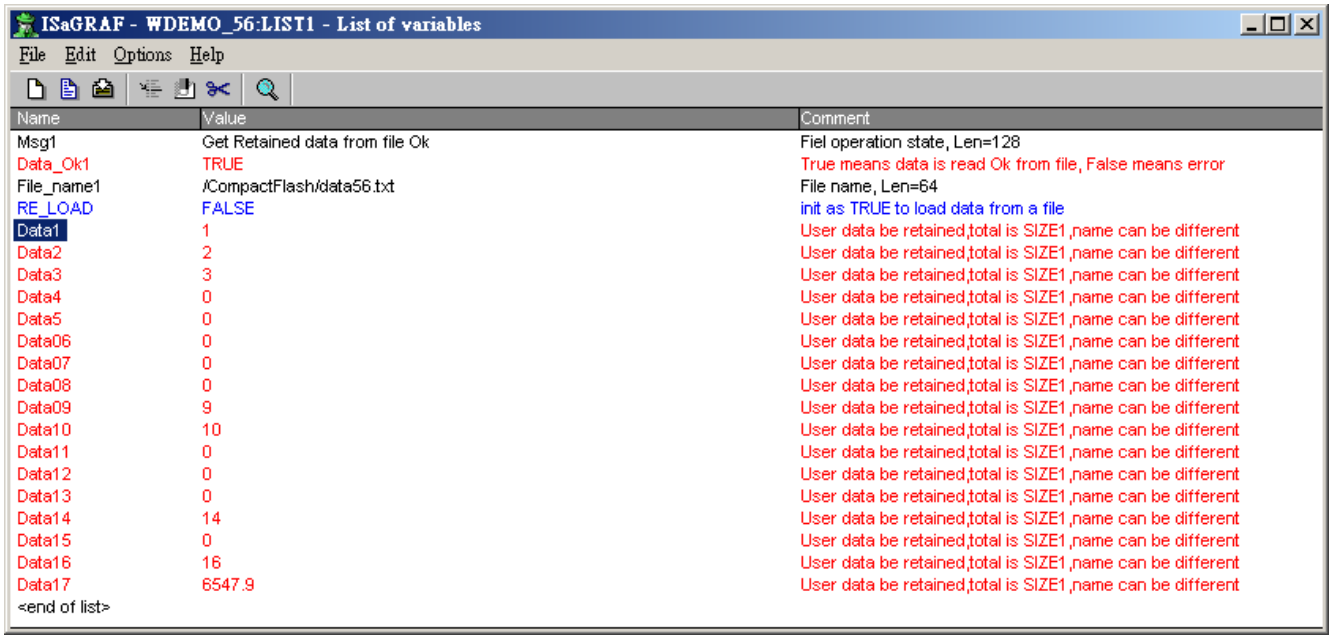
```
(* Any file been open should be closed by f_close( ) *)
TMP2  :=  f_close( File1 )  ;

(* All rows are read Ok *)
if   ii2  =  num_row1    then
   read_F1   :=   TRUE  ;      (* return value as TRUE:Ok *)
end_if  ;
```

ST functions 程式 save_f1 :

```
(*  This function write 20 REAL value F_VAL1[0..19] to file
    return save_f1( ) as TRUE: Ok  ,  FALSE: Error  *)
save_f1  :=  FALSE  ;   (* set as FALSE: Error at the beginning *)
File1  :=  f_creat( File_name1 )  ;   (* Creat a new file to write *)

if   File1 = 0 then
  return ;  (*creat failed , exit this function *)
end_if  ;

(* max "num_row1"  rows to save these REAL values, Each row in the file contains 10 REAL
values *)
for  ii2  :=  0  to  num_row1 - 1  do

  str1  :=  ' '  ;   (* set initial value of str1 *)

  if   ii2  =  num_row1 – 1   then        (* last row *)
    num2   :=   last_num1 ;      (* last row has only "last_num1" number of data *)
  else       (* non-last row *)
    num2  := 10 ;    (* non-last row has 10 data *)
  end_if  ;

  for    jj2  :=  0  to  num2 - 2    do
    str1  :=  str1  +  REAL_STR(  F_VAL1[ 10 * ii2  +  jj2 ]  )  +  ' , '   ;
  end_for ;

  (* the last data in each row should end with <CR> <LF> character *)
  str1  :=  str1  +  REAL_STR(  F_VAL1[ 10 * ii2  +  num2 - 1]  )  +  '$0D$0A'  ;
  TMP2  :=  f_writ_s( File1  , str1 )  ;     (* write one row to file *)

end_for ;

(* Any file been open should be closed by f_close( ) *)
TMP2  :=  f_close( File1 ) ;

save_f1  :=  TRUE  ;   (* return value as TRUE:Ok *)
```

How to test this “wpdmo56” project ?

1. Please download “wpdmo56” to the PAC, then the “Spy list” window will pop-up as below.



Name	Value	Comment
Msg1	Get Retained data from file Ok	Fiel operation state, Len=128
Data_Ok1	TRUE	True means data is read Ok from file, False means error
File_name1	/CompactFlash/data56.txt	File name, Len=64
RE_LOAD	FALSE	init as TRUE to load data from a file
Data1	1	User data be retained,total is SIZE1,name can be different
Data2	2	User data be retained,total is SIZE1,name can be different
Data3	3	User data be retained,total is SIZE1,name can be different
Data4	0	User data be retained,total is SIZE1,name can be different
Data5	0	User data be retained,total is SIZE1,name can be different
Data06	0	User data be retained,total is SIZE1,name can be different
Data07	0	User data be retained,total is SIZE1,name can be different
Data08	0	User data be retained,total is SIZE1,name can be different
Data09	9	User data be retained,total is SIZE1,name can be different
Data10	10	User data be retained,total is SIZE1,name can be different
Data11	0	User data be retained,total is SIZE1,name can be different
Data12	0	User data be retained,total is SIZE1,name can be different
Data13	0	User data be retained,total is SIZE1,name can be different
Data14	14	User data be retained,total is SIZE1,name can be different
Data15	0	User data be retained,total is SIZE1,name can be different
Data16	16	User data be retained,total is SIZE1,name can be different
Data17	6547.9	User data be retained,total is SIZE1,name can be different
<end of list>		

You may modify any value of user data - Data1 to Data17. Then the new value will be saved into the file “\Micro_SD\data56.txt” .
Then you can open this file on the PAC’s monitor screen by double click on the file name. You will see the related value is modified. (Please do not always keep this file open. Close close it later, or the new modified data will not be saved . That is because the file is open, write operation is not allowed)

2. Recycle the power of the PAC. You will see the value keep at its last-modified value when the PAC is boot up well.

3. Edit a “data56.txt” file on a PC as below by the “NotePad” utility. (total 17 data)

1.1 , 2.2 , 3.3 , 4.4 , 5.5 , 6.66 , 7.77 , 8.88 , 9.99 , 10.01
0.01 , 0.02 , 0.03 , 0.04 , 0.05 , 0.06 , 0.07

Then download this “data56.txt” file to PAC ’s \Micro_SD\ path by the “ftp” utility. Then set “RE_LOAD” to become TRUE on ISaGRAF “Spy list” window. You will see the related variable value is updated.