How to read/write file data on WinPAC and XPAC?

Download FAQ-060 Demo.

Note:

1. **WinCon-8xx7/8xx6 has been phased out, visit to the ISaGRAF website for new products.**
   https://www.icpdas.com/en/product/guide+Software+Development__Tools+ISaGRAF#442

2. If the data type in the related file position is not REAL type (32-bit float format), using "F_READ_F" function to read this data may generate a local PAC fault No = 117. Also refer to Chapter 10.6 of the ISaGRAF user manual.

3. Only WinCon-8xx7/8xx6 support File operating functions. Not for I-8xx7 and I-7188EG/XG.

4. If the file path is inside the '\CompactFlash\' folder, for example, '\CompactFlash\data1.txt', the file will be stored in the Compact Flash card in WinCon and this file will exist even the WinCon's power is switched Off. However, it is very slow to Read / Write file in the Compact Flash card.

5. If the file path is not in the '\CompactFlash\' folder, for example, '\Temp\data2.txt', it will be stored in the RAM memory of the WinCon. The file saved in WinCon's RAM will be lost when power is switched OFF. The advantage of Reading/Writing file in the RAM memory is it is much faster than operating in the Compact Flash card.

W-8xx7 supports below ISaGRAF standard functions.

F_ROPEN	Open file in Binary format for read operation (file should exist already) .
F_WOPEN	Open file in Binary format for read and write operation (file should exist already).
F_CLOSE	Close a file.
F_EOF	Test if reach the End-Of-File position.
FA_READ	Read one binary long integer (4-bytes, signed) from file.
FA_WRITE	Write one binary long integer (4-bytes, signed) to file.
FM_READ	Read one message (string) from file.
FM_WRITE	Write one message (string) with <CR> <LF> characters at the end of message to file.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

W-8xx7 supports below ICP DAS c-functions.

F_APPEND	Append one file to the end of the other file.
F_COPY	Copy one file to another file.
F_CREAT	Create a new file.
F_DELETE	Delete a file.
F_DIR	Create a new directory (folder).
F_END	move current file position to the End-Of-File position.
F_SEEK	Move current file position to a specified position.
F_READ_B	Read one binary byte (0 - 255) (1 byte, unsigned) from file.
F_WRIT_B	Write one binary byte (0 - 255) (1 byte, unsigned) to file.
F_READ_W	Read one binary word (-32768 to +32767) (2 byte, signed) from file .
F_WRIT_W	Write one binary word (-32768 to +32767) (2 byte, signed) to file.
F_READ_F	Read one binary REAL (4-bytes, Float) from file. Like 123.45, -2.15E-03, …
F_WRIT_F	Write one binary REAL (4-bytes, Float) to file .
F_WRIT_S	Write one message **without** <CR> <LF> characters at the end of string to file.

Download ISaGRAF demo programs at:
http://www.icpdas.com/en/download/show.php?num=1005&nation=US&kind1=&model=&kw=isagraf
or http://www.icpdas.com/en/faq/index.php?kind=280#751 FAQ-060

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

1. Wdemo_51: Read 10 REAL values from a file. Total 10 rows, each contains one REAL value

The “Wdemo_51.pia” can be found at http://www.icpdas.com/en/faq/index.php?kind=280#751 FAQ-060. If functions of Msg_F, Msg_N, ARY_F_R, AFY_F_W are not found on your PC/ISaGRAF, please download “ICP DAS utilities For ISaGRAF” at http://www.icpdas.com/en/download/show.php?num=368&nation=US&kind1=&model=&kw=isagraf Then run “setup.exe” to restore them to your ISaGRAF workbench.

The “Wdemo_51” program will read 10 REAL values from “\CompactFlash\data51.txt” when the W-8xx7 is just power up or user set the “RE_LOAD” value to become “TRUE” at any time.

(To read/write file in the CompactFlash card take lots of CPU time, please do not read/write it frequently. And please always close the file after the operation. If user read/write file in every PLC scan cycle, the PLC scan time will become a very big time. It will be a bad performance!)

To test this sample program, please edit a text file “data51.txt” in your PC by, for example – “Notepad”. Please enter 10 rows, each contains one Real value. Then download this “data51.txt” to WinCon’s “\CompactFlash\” folder by “ftp” utility. The content of the “data51.txt” looks like below.

```
2.345
999.03
-1.01
456.789
2
456.77
5.9E-12
32.3
45.1
33.3
```

Variables:

Name	Type	Attribute	Description
RE_LOAD	Bool	Internal	Set as True to read File once, init as TRUE
TMP	Bool	Internal	Internal use
File_name1	Message	Internal	Len is 64, init as \CompactFlash\data51.txt
Msg1	Message	Internal	Len is 128, File processing state
str1	Message	Internal	Len is 255, internal use
F_VAL[0..9]	REAL	Internal	Variable array, Dim is 10. The 10 REAL value.
TMP_F	REAL	Internal	Internal use
File1	Integer	Internal	File ID
ii	Integer	Internal	Index of “for” loops

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

ST program:

```
if   RE_LOAD   then          (* Read file once if "RE_LOAD" is TRUE   *)

  RE_LOAD   :=   FALSE   ;

  File1 := f_wopen( File_name1 )   ;     (* Open file in Read & Write mode *)

  if   File1 = 0   then   (* 0: open file fail *)
    Msg1   :=   'Can not Open file '   +   File_name1 ;
    return ;              (* Cannot open file, just exit this ST program *)
  end_if   ;

  for   ii   :=   0   to   9   do      (* Total 10 rows *)

    if   f_eof(File1)   =   TRUE   then          (* test if reaches the End-Of-File *)
      Msg1   :=   'Data number is not enough in '   +   File_name1 ;
      Exit   ;       (* Exit this "for" loops *)
    end_if   ;

    str1   :=   fm_read(File1)   ;          (* read one string in the File *)
    TMP_F   :=   str_real(str1) ;          (* convert string to a REAL value *)
    if   TMP_F   =   1.23E-20   then     (* if returns 1.23E-20, it means format error *)
      Msg1   :=   'The '   +   Msg(ii+1)   +   'th Data format is not correct !'   ;
      exit   ;        (* Exit this "for" loops *)
    end_if ;

    F_VAL[ii]   :=   TMP_F   ;   (* Read & Convert Ok. Store value to F_VAL[0..9] *)

  end_for ;

  TMP   :=   f_close(File1)   ;     (* always close File after its operation *)
  If   ii=10   then              (* All data is succefully read and converted, 10 rows *)
    Msg1 := 'Read ' + File_name1 + ' Ok ' ;
  end_if ;

end_if   ;
```


LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2. Wdemo_54: Read 20 REAL values from a file. Total 4 rows, each contains 5 REAL values

The “Wdemo_54.pia” can be found at http://www.icpdas.com/en/faq/index.php?kind=280#751 FAQ-060. If functions of Msg_F, Msg_N, ARY_F_R, AFY_F_W are not found on your PC/ISaGRAF, please download “ICP DAS utilities for ISaGRAF” at http://www.icpdas.com/en/download/show.php?num=368&nation=US&kind1=&model=&kw=isagraf Then run “setup.exe” to restore them to your ISaGRAF workbench.

The “Wdemo_54” program will read 20 REAL values from “\CompactFlash\data54.txt” when the W-8xx7 is just power up or user set the “RE_LOAD” value to become “TRUE” at any time.

(To read/write file in the CompactFlash card take lots of CPU time, please do not read/write it frequently. And please always close the file after the operation. If user read/write file in every PLC scan cycle, the PLC scan time will become a very big time. It will be a bad performance!)

To test this sample program, please edit a text file “data54.txt” in your PC by, for example – “Notepad”. Please enter 4 rows, each contains 5 Real values. Then download this “data54.txt” to WinCon’s “\CompactFlash\” folder by “ftp” utility. The content of the “data54.txt” looks like below.

```
23,     65.9,     0.012,    5.87,     88.2
0.34,   8.0005,   -2.0E8,   4.08,     5.32E-6
2,      -7,       6666.8,   456.07,   1.01
5,      6,        7,        8,        9
```

Variables:

Name	Type	Attribute	Description
RE_LOAD	Bool	Internal	Set as True to read File once, init as TRUE
TMP	Bool	Internal	Internal use
File_name1	Message	Internal	Len is 64, init as \CompactFlash\data54.txt
Msg1	Message	Internal	Len is 128, File processing state
str1	Message	Internal	Len is 255, internal use
F_VAL[0..19]	REAL	Internal	Variable array, Dim is 20. The 20 REAL value
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 another “for” loops

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

ST program:

```
if   RE_LOAD   then                    (* Read file once if "RE_LOAD" is TRUE *)

  RE_LOAD   :=   FALSE   ;
  File1 := f_wopen( File_name1 )   ;   (*   Open file in Read & Write mode *)

  if   File1 = 0   then          (* 0: open file fail *)
    Msg1   :=   'Can not Open file '   +   File_name1 ;
    return ;              (* Cannot open file, just exit this ST program *)
  end_if   ;

  for   ii   :=   0   to   3   do        (* total 4 rows *)

    if   f_eof(File1)   =   TRUE   then   (* test if reaches the End-Of-File *)
      Msg1   :=   'There should be at least 4 rows in '   +   File_name1   + ' !!!'   ;
      Exit   ;         (* exit this "for" loops *)
    end_if   ;

    str1   :=   fm_read(File1)   ;          (* read one one row as string from file *)

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

   (* If the amount of the converted REAL values is not 5 , it lacks of data. -1 means format error *)
    if   NUM1   <>   5   then
      Msg1 := 'The ' + Msg(ii+1) + 'th row data format is not correct or data number is not 5 !' ;
       Exit   ;          (* exit this "for" loops *)
    end_if   ;

    for   jj   :=   0   to   4   do

     (* Get 5 REAL values from No.1 Float array's addr=1 to 5 , and store them to F_VAL[0..19] *)
      F_VAL[ 5 * ii + jj]   :=   ARY_F_R( 1 ,   jj + 1 )   ;
    end_for ;
  end_for ;

  TMP   :=   f_close(File1)   ;   (* always close File after its operation *)

  If   ii = 4   then          (* All data is succefully read and converted, 4 rows *)
    Msg1   :=   'Read '   +   File_name1   +   ' Ok '   ;
  end_if   ;
end_if   ;
```


LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

3. Wdemo_55: Read 20 Integer values from a file. Total 2 rows, each contains 10 Integer values

The “Wdemo_55.pia” can be found at http://www.icpdas.com/en/faq/index.php?kind=280#751 FAQ-060. If functions of Msg_F, Msg_N, ARY_F_R, AFY_F_W are not found in your PC/ISaGRAF, please download “ICP DAS utilities for ISaGRAF” at http://www.icpdas.com/en/download/show.php?num=368&nation=US&kind1=&model=&kw=isagraf Then run “setup.exe” to restore them to your ISaGRAF workbench.

The “Wdemo_55” program will read 20 Integer values from “\CompactFlash\data55.txt” when the W-8xx7 is just power up or user set the “RE_LOAD” value to become “TRUE” at any time .

(To read/write file in the CompactFlash card take lots of CPU time, please do not read/write it frequently. And please always close the file after the operation. If user read/write file in every PLC scan cycle, the PLC scan time will become a very big time. It will be a bad performance!)

To test this sample program, please edit a text file “data55.txt” in your PC by, for example – “Notepad”. Please enter 2 rows, each contains 10 Integer values. Then download this “data55.txt” to WinCon’s “\CompactFlash\” folder by “ftp” utility. The content of the “data55.txt” looks like below.

```
-1 , 1 , 2 , 3 , 4 , 5 , -6 , 7 , 8 , 9
100001 , 200002 , +300003 , 404 , -505 , 606 , 7007 , 8008 , 9009 , 10
```

Variables:

Name	Type	Attribute	Description
RE_LOAD	Bool	Internal	Set as True to read File once, init as TRUE
TMP	Bool	Internal	Internal use
File_name1	Message	Internal	Len is 64, init as \CompactFlash\data55.txt
Msg1	Message	Internal	Len is 128, File processing state
str1	Message	Internal	Len is 255, internal use
N_VAL[0..19]	Integer	Internal	Variable array, Dim is 20. The 20 Integer values
NUM1	Integer	Internal	Get return of Msg_N( ), -1 means format error
File1	Integer	Internal	File ID
ii	Integer	Internal	Index of “for” loops
jj	Integer	Internal	Index of another “for” loops

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

ST program:

```
if   RE_LOAD   then        (*Read file once if “RE_LOAD” is TRUE *)

  RE_LOAD   :=   FALSE   ;

  File1 := f_wopen( File_name1 )   ;   (* Open file in Read & Write mode *)

  if   File1 = 0   then        (* 0: open file fail *)
    Msg1   :=   'Can not Open file '   +   File_name1 ;
    return ;       (* Cannot open file, just exit this ST program *)
  end_if   ;

  for   ii    :=    0    to    1    do          (* total 2 rows *)

      if    f_eof(File1)   =   TRUE   then    (* test if reaches the End-Of-File *)
          Msg1    :=    'There should be at least 2 rows in '   +   File_name1   + ' !!!'   ;
          Exit   ;     (* exit this “for” loops *)
      end_if   ;

      str1   :=   fm_read(File1)   ;       (* read one one row as string from file *)

 (* Convert string to become serval Integer values and store them into No. 2 Integer array *)
      NUM1   :=   Msg_N(str1 , 2)   ;

 (* If the amount of the converted Integer values is not 10 , it lacks of data. -1 means format error* )
     if   NUM1   <>   10    then
         Msg1 := 'The ' + Msg(ii+1) + 'th row data format is not correct or data number is not 10 !' ;
           Exit   ;       (* exit this “for” loops *)
      end_if   ;

      for   jj    :=    0   to    9    do

  (* Get 10 Integer values from No.2 Integer array’s addr=1 to 10 , and store them to N_VAL[0..19] *)
          N_VAL[ 10 * ii + jj]   :=   ARY_N_R( 2 ,   jj + 1 )   ;
      end_for ;
  end_for ;

  TMP    :=    f_close(File1)   ;   (* always close File after its operation *)
  If    ii = 2   then                  (* All data is succefully read and converted, 2 rows *)
     Msg1   :=   'Read '   +   File_name1   +   ' Ok '   ;
  end_if   ;
end_if   ;
```


LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

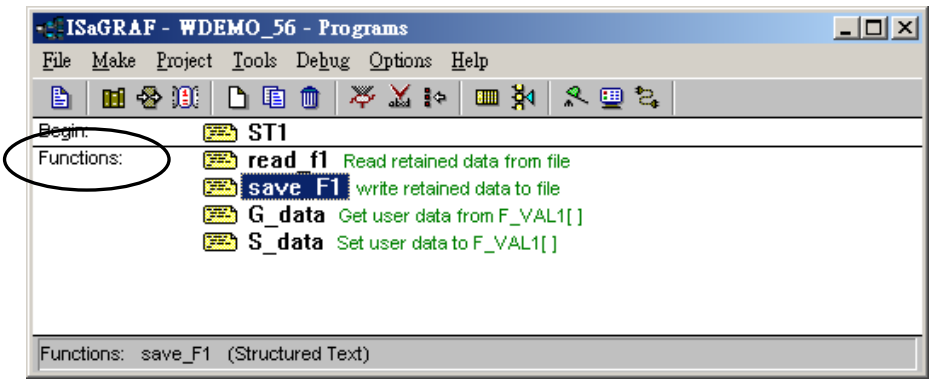
4. Wdemo_56: Retain values of 1 to 255 Real variable in Compact Flash card

The “Wdemo_56.pia” can be found at http://www.icpdas.com/en/faq/index.php?kind=280#751 FAQ-060. If functions of Msg_F, Msg_N, ARY_F_R, AFY_F_W are not found in your PC/ISaGRAF, please download “ICP DAS utilities for ISaGRAF” at http://www.icpdas.com/en/download/show.php?num=368&nation=US&kind1=&model=&kw=isagraf Then run “setup.exe” to restore them to your ISaGRAF workbench.

The “Wdemo_56” program will read 1 to 255 REAL values from “\CompactFlash\data56.txt” to the related variable when the W-8xx7 is just power up. If this “data56.txt” doesn’t exist, all 1 to 255 values will be init as 0.0. At run time, if any value of these variable is modified, all 1 to 255 values will be written once to the “data56.txt” to make sure these variable’s value are well retained in file. If the file doesn’t exist, this program will create it.

To read/write file in the CompactFlash card take lots of CPU time, please do not read/write it frequently. And please always close the file after the operation. If user read/write file in every PLC scan cycle, the PLC scan time will become a very big time. It will be a bad performance! If user need fast retain function, please refer to Chapter 10.1 for retaining data in the S256/S512.

Project Architecture:
There are 5 ST programs in this “wdemo_56” project. Four of them are ISaGRAF user-defined functions – “reaf_f1”, “save_f1”, “G_data” and “S_data”.



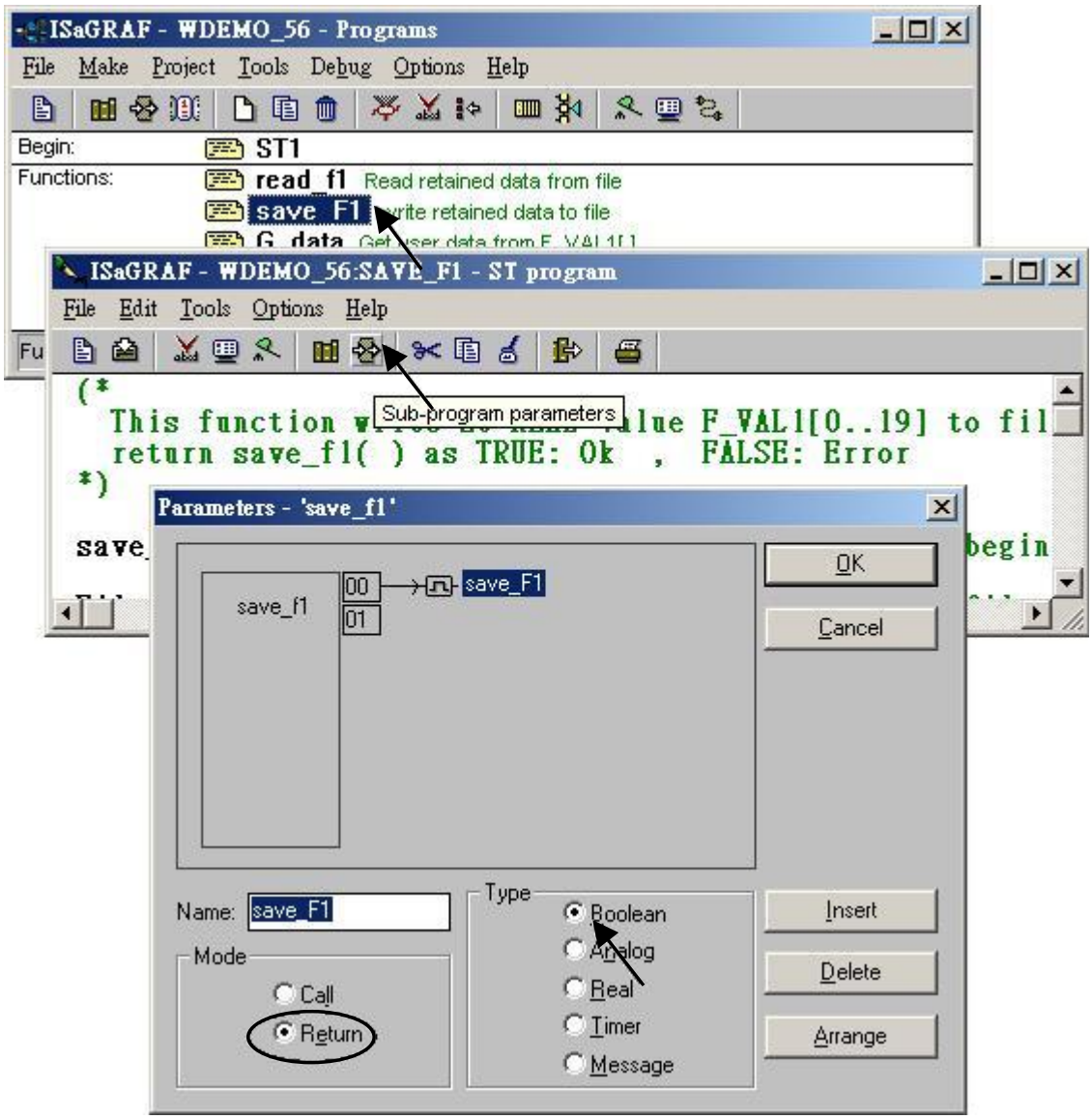
Important note:

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. And then remember to compile it.
2. Please also modify the “Dim” value of the “F_VAL1[ ]” and “Old_F_VAL1[ ]” variable array in the ISaGRAF “dictionary” window to become the same value as the “SIZE1”. And also please modify the “G_data” and “S_data” program.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

3. There is one advantage of retaining vale in the CompactFlash card. The data file can be edited in advance in PC. Then using “ftp” utility to download it to WinCon. The file path name of this example is “\CompactFlash\data56.txt”. Then set “RE_LOAD” value to TRUE once, all related variable will update to the new value.

4. The following ST programs are all declared as 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 in the Chapter 15)



LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

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

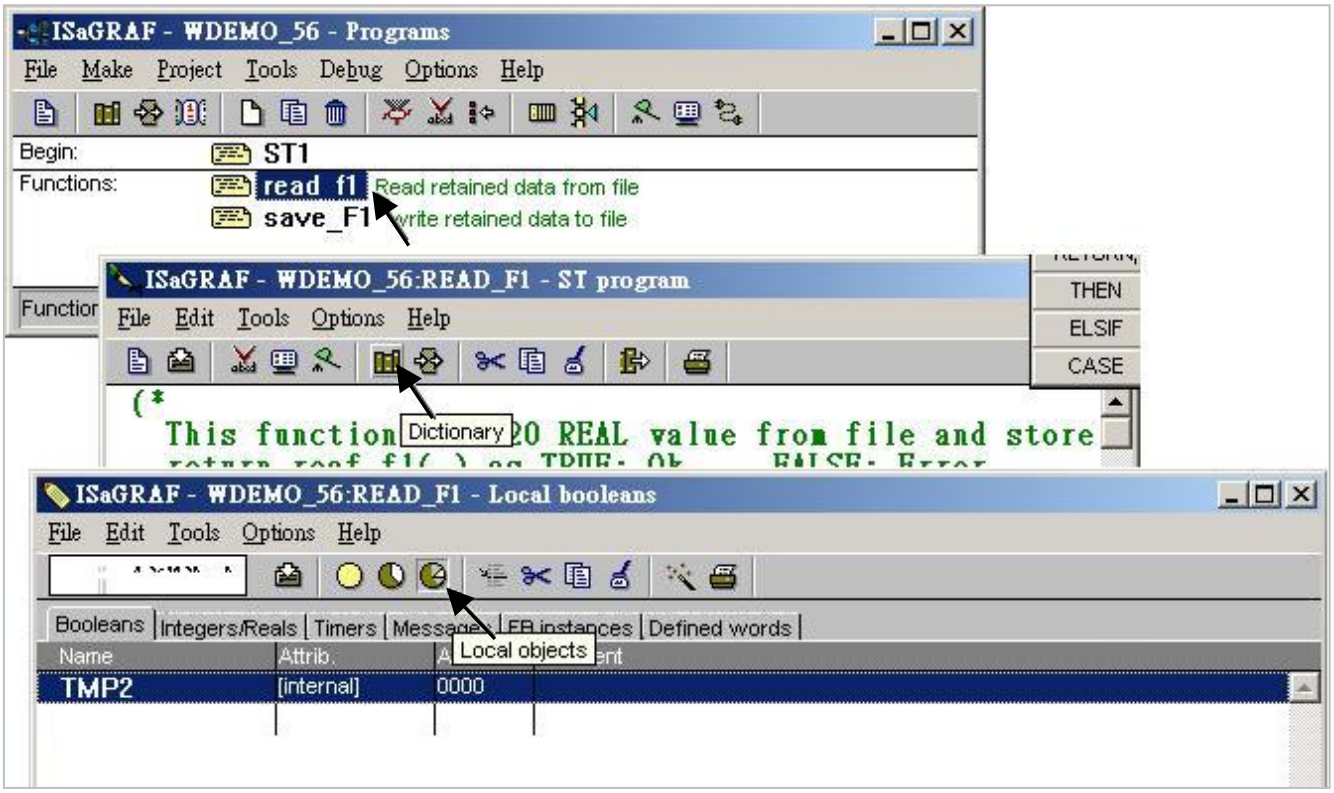
read_f1:

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.



LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Variables (Global variable):

Name	Type	Attribute	Description
SIZE1	Integer	Constant	The number of retained variables. Can be 1 to 255. Please also 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 \CompactFlash\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.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

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 *)

  (* calculate 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 PAC 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   ;
```


LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
 (* Get user data from F_VAL1[ ] when PAC 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 WinCon screen ! '    ;

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

   end_if   ;

end_if   ;
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

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 *)
```


LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

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 *)
```


LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

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;
```


LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
    (* 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    ;
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

ST functions program 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 )   ;     (* Create a new file to write *)

if     File1 = 0 then
   return ;   (* create 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 ;

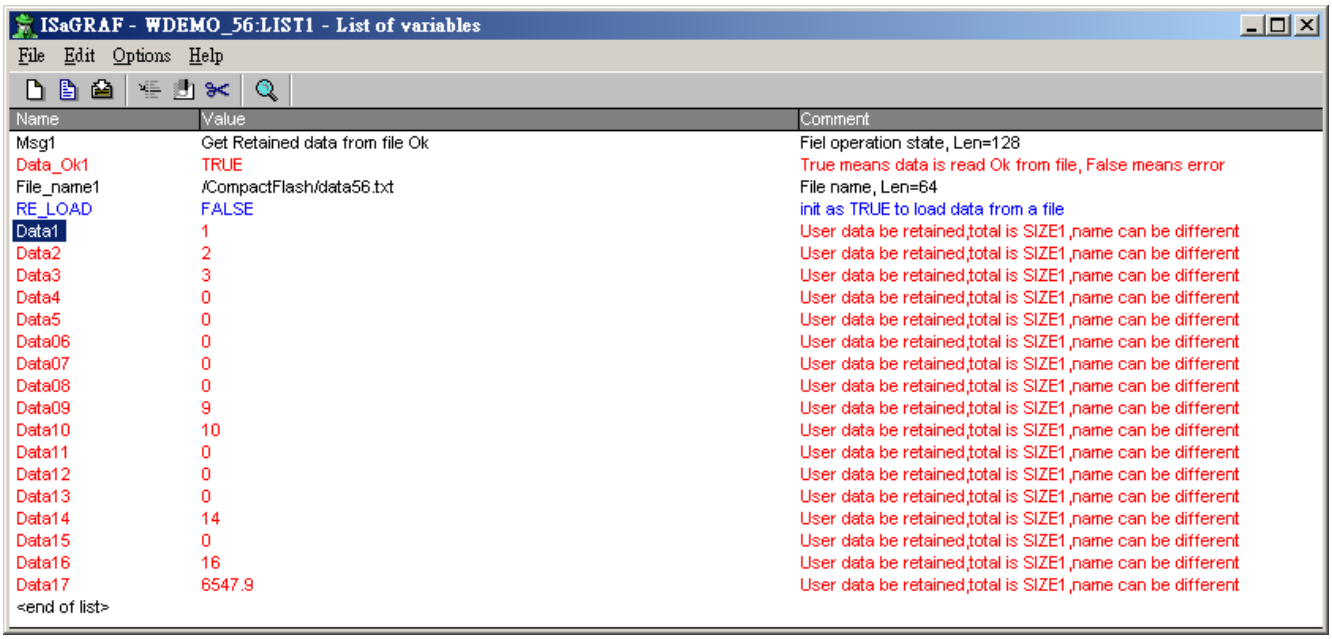
(* Any file been open should be closed by f_close( ) *)
TMP2   :=   f_close( File1 ) ;

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


LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

How to test this "wdemo_56" project ?

1. Please download "wdemo_56" to W-8xx7, 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 once into file of "\CompactFlash\data56.txt".

Then you can open this file on the WinCon'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. Please 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 WinCon. You will see the value keep at the last modified value when WinCon is boot up well.

3. Edit a "data56.txt" file on PC as below by "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
```

4. Then please download this "data56.txt" file to WinCon's \CompactFlash\ path by "ftp" utility. Then set "RE_LOAD" to become TRUE on ISaGRAF "Spy list" window. You will see the related variable value is updated.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

5. Record I-8017H's Ch.1 to Ch.4 voltage input in a user allocated RAM memory in the WinCon-8xx7?

The sampling time is one record every 0.01 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.

Please refer to Section 11.3.10 of the ISaGRAF user manual or visit to FAQ-057

Click the link for more ISaGRAF FAQ:
http://www.icpdas.com/en/faq/index.php?kind=280#751

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000