Classification	Win-GRAF English FAQ-015						
Author	Eva Li	Version	1.0.0	Date	Feb. 2016	Page	1 / 15

How to Set up Retain Variables and Data Storage?

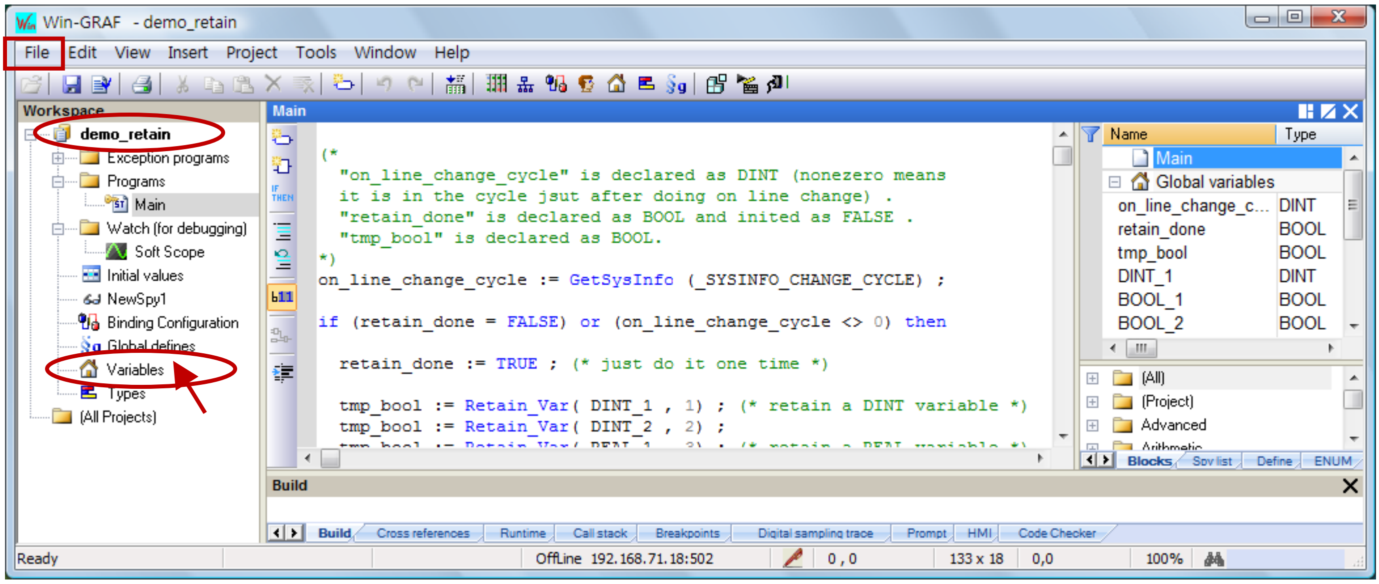
Download FAQ-015 Demo

1.1. Retain Variable

This paper lists the way to use "RETAIN_VAR", "RETAIN_ARY", "RETAIN_FLAG_GET", "RETAIN_FLAG_SET" and "RETAIN_FLAG_CLR” Functions. The Win-GRAF PACs are built-in the Retain memory for users to store the retain variable data that will not lose due to the PAC shutdown and can retain the last value at the next time reboot.

Note: Function "Retain_Var()" or Retain_Ary() can only be used in the first PAC Cycle or in the Cycle that performs the On-line Change. If use them in other Cycle, it will return "FALSE". If the Retain Variable has not assigned any initial value and the PAC calls the Function, the return value is not meaningful; users need to assign appropriate initial values to all Retain Variables at least once.

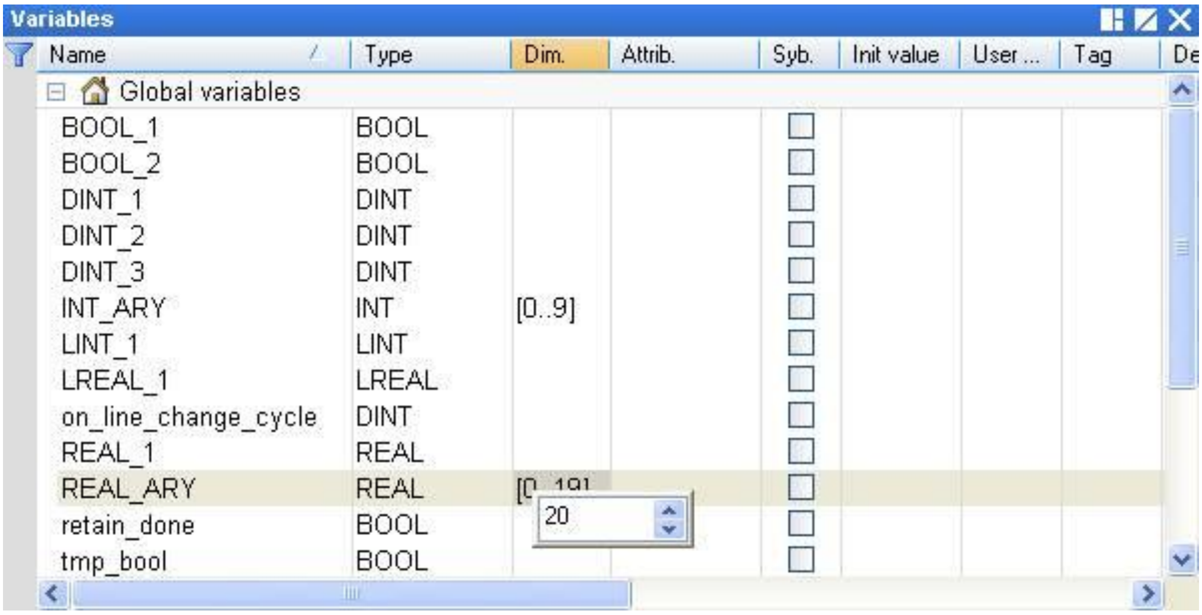
This paper provides two demo projects (i.e., demo_retain.zip and demo_wp5_retain.zip). The user can run the Win-GRAF and then click File > Add Existing Project > From Zip in the menu bar to restore the "demo_retain" project (or refer the FAQ-003).



```
(*
  "on_line_change_cycle" is declared as DINT (nonezero means
  it is in the cycle jsut after doing on line change) .
  "retain_done" is declared as BOOL and inited as FALSE .
  "tmp_bool" is declared as BOOL.
*)
on_line_change_cycle := GetSysInfo (_SYSINFO_CHANGE_CYCLE) ;

if (retain_done = FALSE) or (on_line_change_cycle <> 0) then

  retain_done := TRUE ; (* just do it one time *)

  tmp_bool := Retain_Var( DINT_1 , 1) ; (* retain a DINT variable *)
  tmp_bool := Retain_Var( DINT_2 , 2) ;
  tmp_bool := Retain_Var( REAL_1 , 3) ; (* retain a REAL variable *)
```

Name	Type
Main	
Global variables	
on_line_change_c...	DINT
retain_done	BOOL
tmp_bool	BOOL
DINT_1	DINT
BOOL_1	BOOL
BOOL_2	BOOL

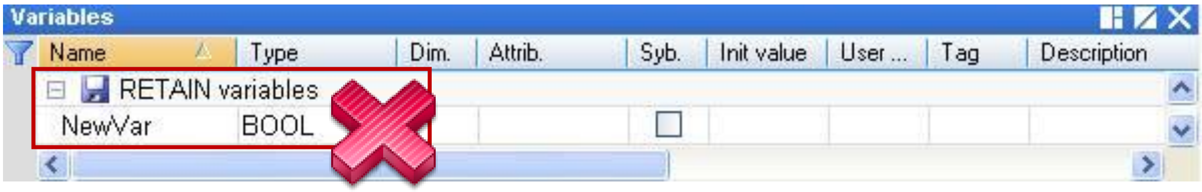
You can click "Variables" to check/set variables in the Variables window (as the figure below), if you want to know the way of the variable declaration, refer the Win-GRAF Getting Started Manual (Section 2.2.2 and Section 2.3.1).



Variables

Name	Type	Dim.	Attrib.	Syb.	Init value	User ...	Tag	De
Global variables								
BOOL_1	BOOL			□				
BOOL_2	BOOL			□				
DINT_1	DINT			□				
DINT_2	DINT			□				
DINT_3	DINT			□				
INT_ARY	INT	[0..9]		□				
LINT_1	LINT			□				
LREAL_1	LREAL			□				
on_line_change_cycle	DINT			□				
REAL_1	REAL			□				
REAL_ARY	REAL	[0..19] 20		□				
retain_done	BOOL			□				
tmp_bool	BOOL			□				

Note: **ICP DAS Win-GRAF PAC does not support the "RETAIN variables" Functions in the "Variables" window,** so using "RETAIN_VAR", "RETAIN_ARY", "RETAIN_FLAG_GET", "RETAIN_FLAG_SET" and "RETAIN_FLAG_CLR" functions to achieve the Retain Variables.



Variables

Name	Type	Dim.	Attrib.	Syb.	Init value	User ...	Tag	Description
RETAIN variables								
NewVar	BOOL			□				

1.2. Description of the "demo_retain" Project

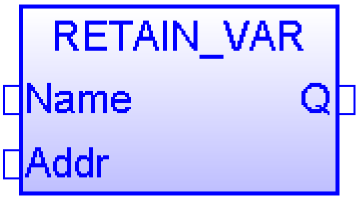
ST Program:

This demo uses Function Retain_Var() and Retain_Ary().

```
 (*   "on_line_change_cycle" is declared as DINT
      (nonezero means it is in the cycle just after doing on line change).
      "retain_done" is declared as BOOL and inited as FALSE.
      "tmp_bool" is declared as BOOL.
  *)

  on_line_change_cycle    :=    GetSysInfo (_SYSINFO_CHANGE_CYCLE) ;
   if (retain_done = FALSE) or (on_line_change_cycle <> 0) then
          retain_done := TRUE ;   (* just do it one time *)
          tmp_bool := Retain_Var ( DINT_1 , 1) ; (* retain a DINT variable *)
          tmp_bool := Retain_Var ( DINT_2 , 2) ;
          tmp_bool := Retain_Var ( REAL_1 , 3) ; (* retain a REAL variable *)
          tmp_bool := Retain_Var ( BOOL_1 , 4) ; (* retain a BOOL variable *)
          tmp_bool := Retain_Var ( BOOL_2 , 5) ;

      (* retain 10 elements of an INT array variable at retain addr starting at 6. *)
          tmp_bool := Retain_Ary ( INT_ARY ,    6 ,    10) ;
      (* retain 20 elements of a REAL array variable at retain addr starting at 16. *)
          tmp_bool := Retain_Ary ( REAL_ARY , 16 , 20) ;
          tmp_bool := Retain_Var ( DINT_3 , 36) ;
          (* 64-bit variable can use only addr from 10,001 to 12,000 *)
          tmp_bool := Retain_Var ( LINT_1    , 10001) ;    (* retain a LINT variable (64-bit) *)
          tmp_bool := Retain_Var ( LREAL_1 , 10002) ;    (* retain a LREAL variable (64-bit) *)
      end_if ;
```

1.3. Introduction of the Retain_xxx Functions

1.3.1. RETAIN_VAR (Retain a Variable)



Tips:
Press “F1” key to see more details.

Name:
A variable name (DO NOT use Array variable or String).
Variable type can be BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT, DWORD, REAL, TIME, LINT or LREAL.

Addr:
Data Type: DINT. The address number for retaining the variable, can be 1 to 12,000.

Q:
Data Type: BOOL. TRUE: Ok; FALSE: Error.

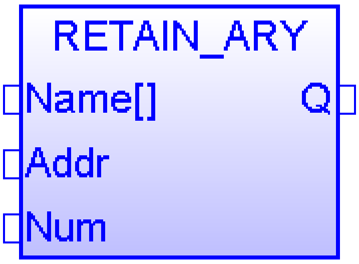
Note:

1. One Addr can accept only one variable (or one element of the array).
   DO NOT assign the same Addr to two variables (or more), or the Retain Value will be wrong.

2. 64-bit data type (LINT or LREAL) can use only the Addr No. from 10,001 to 12,000.

3. Other data type (BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT, DWORD, REAL or TIME) can use the Addr No. from 1 to 12,000.

1.3.2. RETAIN_ARY (Retain an Array Variable)



Tips:
Press “F1” key to see more details.

Name[]:
An ARRAY variable name (DO NOT use String).
Variable type can be BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT, DWORD, REAL, TIME, LINT or LREAL.

Addr:
Data Type: DINT. The starting address number for retaining the array variable; can be 1 to 12,000.

Num:
Data Type: DINT. The amount of elements in the Array variable to be retained.
For example:
If there are 100 elements in an Array variable, set "Num" to "1 to 100" is correct, but if set it more than 100 that is not correct.
If there are 5 elements in an Array variable, set "Num" to "1 to 5" is correct, but if set it more than 5 that is not correct.

Q:
Data Type: BOOL. TRUE: Ok; FALSE: Error.

Note:

1. One Addr can accept only one variable (or one element of array).
   DO NOT assign the same Addr to two variables (or more), or the Retain Value will be wrong.

2. 64-bit data type (LINT or LREAL) can use only the Addr No. from 10,001 to 12,000.

3. Other data type (BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT, DWORD, REAL or TIME) can use the Addr No. from 1 to 12,000.

1.3.3. RETAIN_FLAG_SET/GET/CLR (Set/Get/Clear the Retain Flag)

How to Use:

The "Retain Flag" is a flag (TRUE/FALSE) stored by users in the retain memory. Users can set this retain flag to indicate "All retain data has been assigned a proper value before". When a PAC starts without setting a proper value to retain variable before, the data of the retain variable read from the retain memory is not correct (it is normally a strange value). So users have to assign proper value to all retain variable at least once to let the application work well. Then after user can call the "Retain_Flag_Set()" to set the retain flag. It means "All retain data has been assigned a proper value".

To get the state of the Retain Flag, please call "Retain_Flag_Get()".

To clear the state of the Retain Flag, please call "Retain_Flag_Clr()".

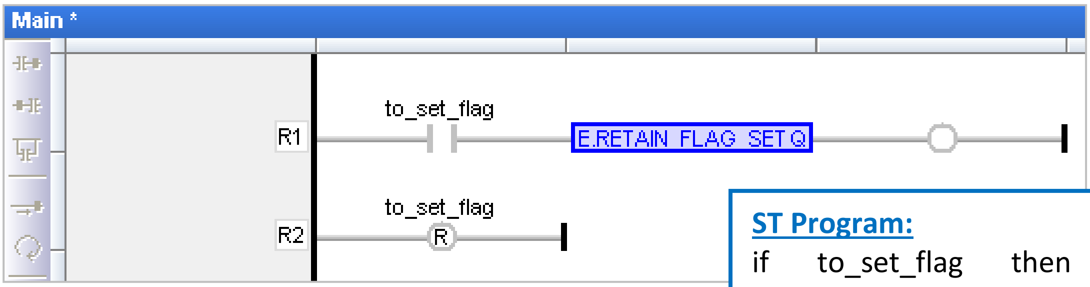
ST Program:

```
 (*  "on_line_change_cycle" is declared as DINT
     (nonezero means it is in the cycle just after doing on line change) .
      "retain_done" is declared as BOOL and inited as FALSE .
      "tmp_bool", "retain_flag" and "to_set_flag"    are declared as BOOL.
   *)

    on_line_change_cycle    :=    GetSysInfo (_SYSINFO_CHANGE_CYCLE) ;

    if (retain_done = FALSE) or (on_line_change_cycle <> 0) then
        retain_done := TRUE ;    (* just do it one time *)
        tmp_bool := Retain_Var( DINT_1 , 1) ;    (* retain a DINT variable *)
        tmp_bool := Retain_Var( DINT_2 , 2) ;
        tmp_bool := Retain_Var( REAL_1 , 3) ; (* retain a REAL variable *)
        tmp_bool := Retain_Var( BOOL_1 , 4) ; (* retain a BOOL variable *)

        (* ... After doing all the Retain Functions ... *)
        retain_flag := Retain_Flag_Get();
        if    (retain_flag = FALSE)    then
      (*If Retain variable does not set up any proper value, you can do some proper operation here. *)
            (* ... *)
        end_if ;
    end_if ;

     (* When all Retain variables are assigned proper values, remember to set the "to_set_flag" to
        "TRUE" for calling "Retain_Flag_Set() once, so that, when next time you use the
        "Retain_Flag_Get()", it can return "TRUE".
     *)
      if    (to_set_flag = TRUE)    then
          to_set_flag := FALSE ;
          tmp_bool := Retain_Flag_Set() ;
      end_if ;
```

LD Program:

(Press "F1" key to see the detailed setting descriptions.**)**

RETAIN_FLAG_SET: Set the retain flag.

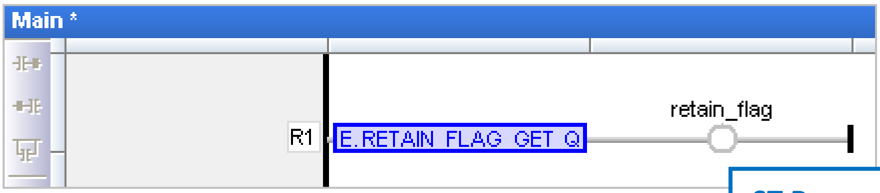


ST Program:

```
if      to_set_flag     then
        to_set_flag   :=   FALSE   ;
        TMP_BOOL    :=
Retain_Flag_Set() ;
end_if ;
```

Q: Data Type: BOOL. Always return TRUE.

RETAIN_FLAG_GET: Get the state of the retain flag.

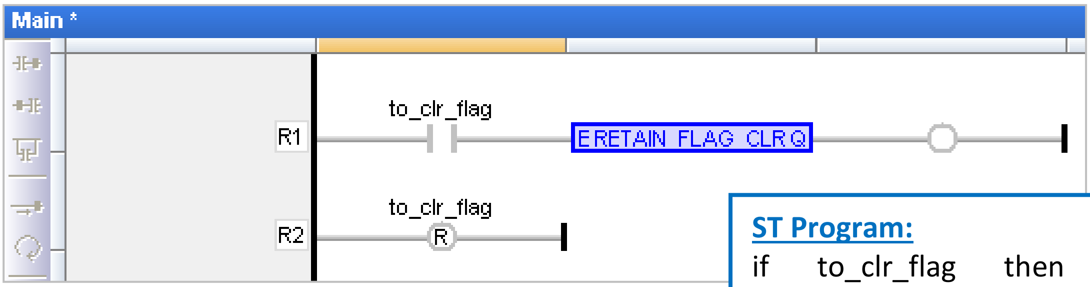


ST Program:

```
retain_flag    :=    Retain_Flag_Get() ;
```

Q: Data Type: BOOL.
"TRUE": flag is set;
"FALSE": flag is not set.

RETAIN_FLAG_CLR: Clear the retain flag.



ST Program:

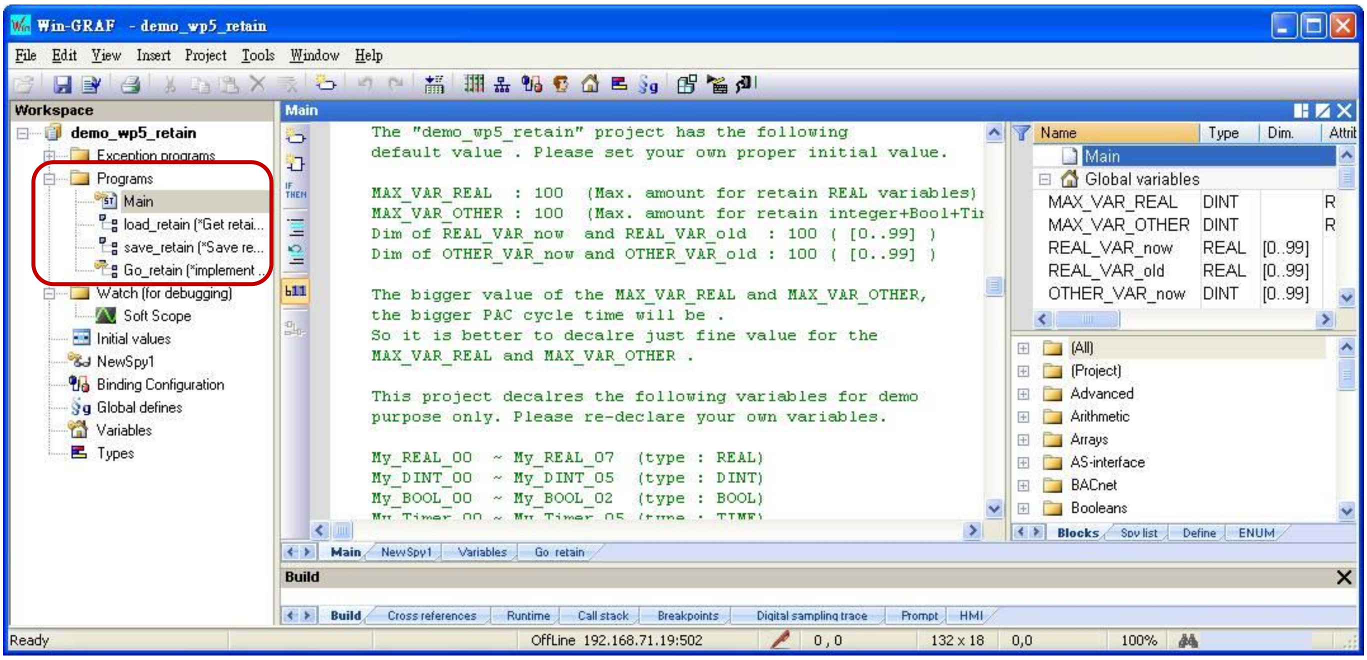
```
if      to_clr_flag     then
        to_clr_flag   :=   FALSE   ;
        TMP_BOOL    :=
Retain_Flag_Clr() ;
end_if ;
```

Q: Data Type: BOOL. Always return TRUE.

1.4. Retain Variable (Using files)

You can refer the following demo project (demo_wp5_retain.zip) to use retain variables with files in the \System_disk\Win-GRAF\ .

You can run the Win-GRAF and then click Execute File> Add Existing Project > From Zip in the menu bar to restore the "demo_wp5_retain" project (or refer the FAQ-003). This project includes an ST program (Main) and 3 ST sub-programs (load_retain, save_retain and Go_retain).



```
The "demo_wp5_retain" project has the following
default value . Please set your own proper initial value.

MAX_VAR_REAL  : 100  (Max. amount for retain REAL variables)
MAX_VAR_OTHER : 100  (Max. amount for retain integer+Bool+Tir
Dim of REAL_VAR_now  and REAL_VAR_old  : 100 ( [0..99] )
Dim of OTHER_VAR_now and OTHER_VAR_old : 100 ( [0..99] )

The bigger value of the MAX_VAR_REAL and MAX_VAR_OTHER,
the bigger PAC cycle time will be .
So it is better to decalre just fine value for the
MAX_VAR_REAL and MAX_VAR_OTHER .

This project decalres the following variables for demo
purpose only. Please re-declare your own variables.

My_REAL_00  ~ My_REAL_07  (type : REAL)
My_DINT_00  ~ My_DINT_05  (type : DINT)
My_BOOL_00  ~ My_BOOL_02  (type : BOOL)
My_Timer_00 ~ My_Timer_05 (type : TIME)
```

Name	Type	Dim.	Attrit
Main			
Global variables			
MAX_VAR_REAL	DINT		R
MAX_VAR_OTHER	DINT		R
REAL_VAR_now	REAL	[0..99]	
REAL_VAR_old	REAL	[0..99]	
OTHER_VAR_now	DINT	[0..99]	

Limitation :

This project is not good at handling Retain variables which value changes frequently. For example, value changed about every second or every minute. That is because these retain values of this project are saved within files in the \System_Disk. The file operations in it consume more CPU time, which will slow down the PAC performance if retain value changed frequently.

The "demo_wp5_retain" project has the following default values. Please set your own proper initial values.

MAX_VAR_REAL: 100 (Max. amount for retaining REAL variables)
MAX_VAR_OTHER: 100 (Max. amount for retaining integer+Bool+Timer variables)

The dimension of "**REAL**_VAR_now" and "**REAL**_VAR_old" (i.e., [0..99]) must the same with the value of "MAX_VAR_**REAL**" (i.e., 100).

The dimension of "**OTHER**_VAR_now" and "**OTHER**_VAR_old" (i.e., [0..99]) must the same with the value of "MAX_VAR_**OTHER**" (i.e., 100).

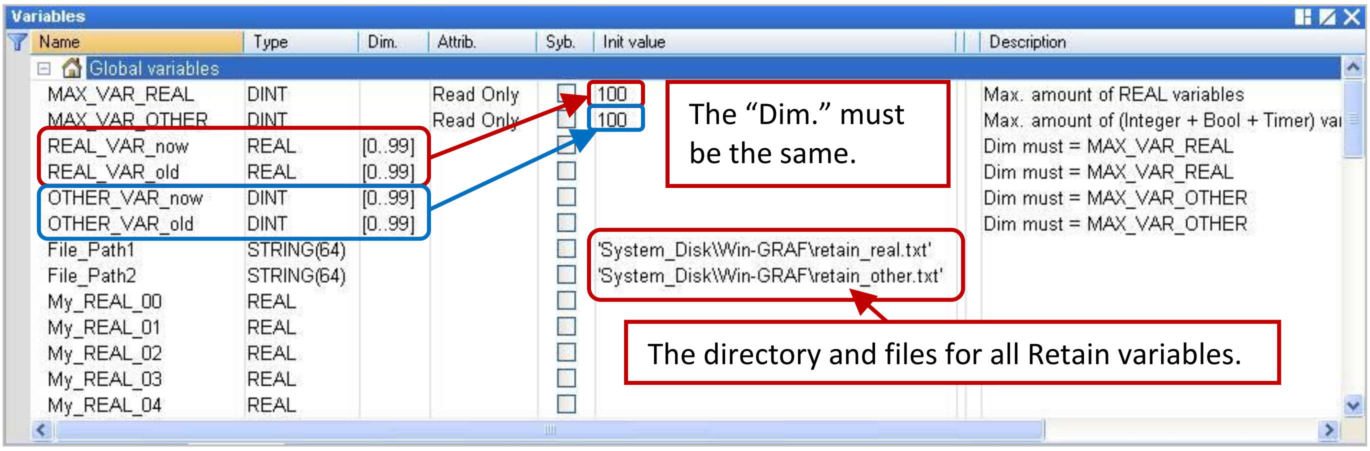
Note:

The bigger the value of the **MAX**_VAR_REAL and **MAX**_VAR_OTHER, the larger the PAC cycle time will be. So it is better to declare just fine value for the MAX_VAR_REAL and MAX_VAR_OTHER.

This project declares the following variables for demo purpose only. Please re-declare your own variables.

My_REAL_00	~	My_REAL_07	(Data Type: REAL)
My_DINT_00	~	My_DINT_05	(Data Type: DINT)
My_BOOL_00	~	My_BOOL_02	(Data Type: BOOL)
My_Timer_00	~	My_Timer_05	(Data Type: TIME)

You can see more variables in the "Variables" window.



Variables

Name	Type	Dim.	Attrib.	Syb.	Init value	Description
Global variables						
MAX_VAR_REAL	DINT		Read Only	□	100	Max. amount of REAL variables
MAX_VAR_OTHER	DINT		Read Only	□	100	Max. amount of (Integer + Bool + Timer) vai
REAL_VAR_now	REAL	[0..99]		□		Dim must = MAX_VAR_REAL
REAL_VAR_old	REAL	[0..99]		□		Dim must = MAX_VAR_REAL
OTHER_VAR_now	DINT	[0..99]		□		Dim must = MAX_VAR_OTHER
OTHER_VAR_old	DINT	[0..99]		□		Dim must = MAX_VAR_OTHER
File_Path1	STRING(64)			□	'System_Disk\Win-GRAF\retain_real.txt'	
File_Path2	STRING(64)			□	'System_Disk\Win-GRAF\retain_other.txt'	
My_REAL_00	REAL			□		
My_REAL_01	REAL			□		
My_REAL_02	REAL			□		
My_REAL_03	REAL			□		
My_REAL_04	REAL			□		

The “Dim.” must be the same.

The directory and files for all Retain variables.

"Go_retain" sub-program is used to do the retain operation. Remember to modify this sub-program. There are 4 sections should to be modified in it. Please search "Add your REAL variables for retain here" and "Add your integer, BOOL, Timer variables for retain here", and depend on your re-declared variables to modify your "Go_retain" sub-program.

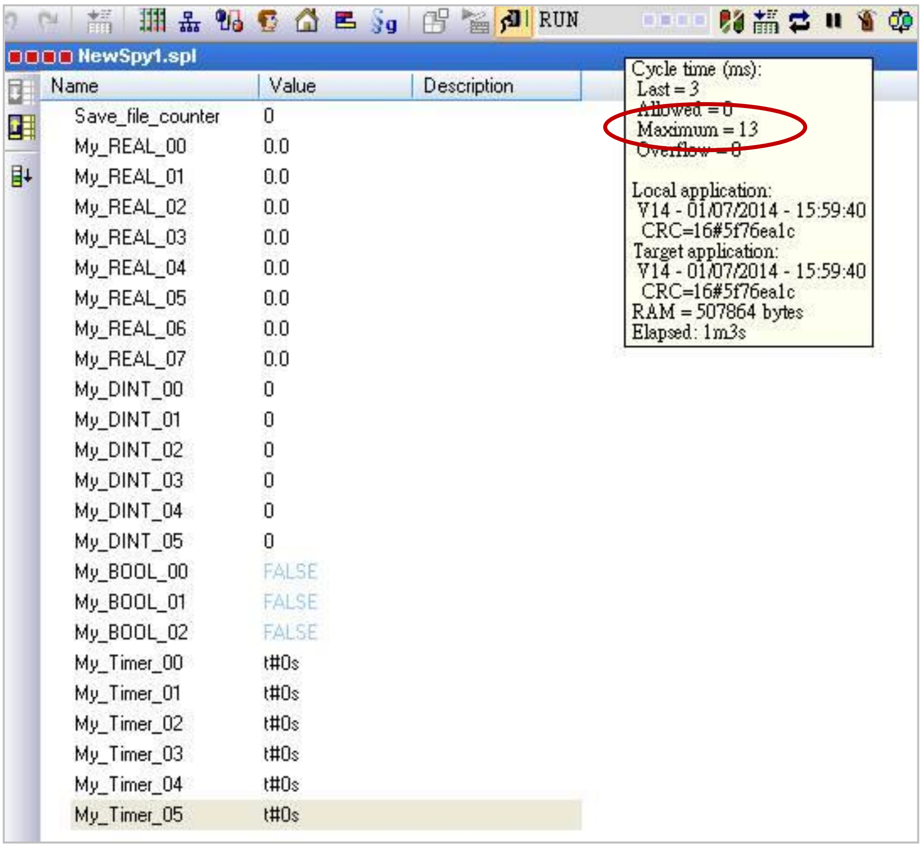
```
(* Add your REAL variables for retain here *)
(* ---------------------------------------------- *)
My_REAL_00 := REAL_VAR_now[0] ;
My_REAL_01 := REAL_VAR_now[1] ;
My_REAL_02 := REAL_VAR_now[2] ;
My_REAL_03 := REAL_VAR_now[3] ;
My_REAL_04 := REAL_VAR_now[4] ;
My_REAL_05 := REAL_VAR_now[5] ;
My_REAL_06 := REAL_VAR_now[6] ;
My_REAL_07 := REAL_VAR_now[7] ;
(* ---------------------------------------------- *)
(* Add your integer, BOOL, Timer variables for retain here *)
(* ................................................ *)
My_DINT_00 := OTHER_VAR_now[0] ;
My_DINT_01 := OTHER_VAR_now[1] ;
My_DINT_02 := OTHER_VAR_now[2] ;
My_DINT_03 := OTHER_VAR_now[3] ;
My_DINT_04 := OTHER_VAR_now[4] ;
My_DINT_05 := OTHER_VAR_now[5] ;

My_BOOL_00 := Any_to_BOOL( OTHER_VAR_now[6] ) ;
My_BOOL_01 := Any_to_BOOL( OTHER_VAR_now[7] ) ;
My_BOOL_02 := Any_to_BOOL( OTHER_VAR_now[8] ) ;

My_Timer_00 := Any_to_TIME( OTHER_VAR_now[9] ) ;
My_Timer_01 := Any_to_TIME( OTHER_VAR_now[10] ) ;
My_Timer_02 := Any_to_TIME( OTHER_VAR_now[11] ) ;
My_Timer_03 := Any_to_TIME( OTHER_VAR_now[12] ) ;
My_Timer_04 := Any_to_TIME( OTHER_VAR_now[13] ) ;
My_Timer_05 := Any_to_TIME( OTHER_VAR_now[14] ) ;
(* ................................................ *)

(* Add your REAL variables for retain here *)
(* ------------------------------------------------ *)
REAL_VAR_now[0]   := My_REAL_00 ;
REAL_VAR_now[1]   := My_REAL_01 ;
REAL_VAR_now[2]   := My_REAL_02 ;
REAL_VAR_now[3]   := My_REAL_03 ;
REAL_VAR_now[4]   := My_REAL_04 ;
REAL_VAR_now[5]   := My_REAL_05 ;
REAL_VAR_now[6]   := My_REAL_06 ;
REAL_VAR_now[7]   := My_REAL_07 ;
(* ------------------------------------------------ *)

(* Add your integer, BOOL, Timer variables for retain here *)
(* ................................................ *)
OTHER_VAR_now[0]   := My_DINT_00 ;
OTHER_VAR_now[1]   := My_DINT_01 ;
OTHER_VAR_now[2]   := My_DINT_02 ;
OTHER_VAR_now[3]   := My_DINT_03 ;
OTHER_VAR_now[4]   := My_DINT_04 ;
OTHER_VAR_now[5]   := My_DINT_05 ;

OTHER_VAR_now[6]   := Any_to_DINT( My_BOOL_00 ) ;
OTHER_VAR_now[7]   := Any_to_DINT( My_BOOL_01 ) ;
OTHER_VAR_now[8]   := Any_to_DINT( My_BOOL_02 ) ;
```

```
OTHER_VAR_now[9]    := Any_to_DINT( My_Timer_00 ) ;
OTHER_VAR_now[10] := Any_to_DINT( My_Timer_01 ) ;
OTHER_VAR_now[11] := Any_to_DINT( My_Timer_02 ) ;
OTHER_VAR_now[12] := Any_to_DINT( My_Timer_03 ) ;
OTHER_VAR_now[13] := Any_to_DINT( My_Timer_04 ) ;
OTHER_VAR_now[14] := Any_to_DINT( My_Timer_05 ) ;
(* ................................................ *)
```

Test Project:

Before testing, make sure you have already set up the PAC IP and compile/download the project into the PAC (refer the Win-GRAF Getting Started Manual - Section 2.3.4 and Section 2.3.5). After connecting with the PAC, all values in the Spy list (refer the Win-GRAF User Manual - Section 11.3) will be "0" (or "FALSE") at the begging.

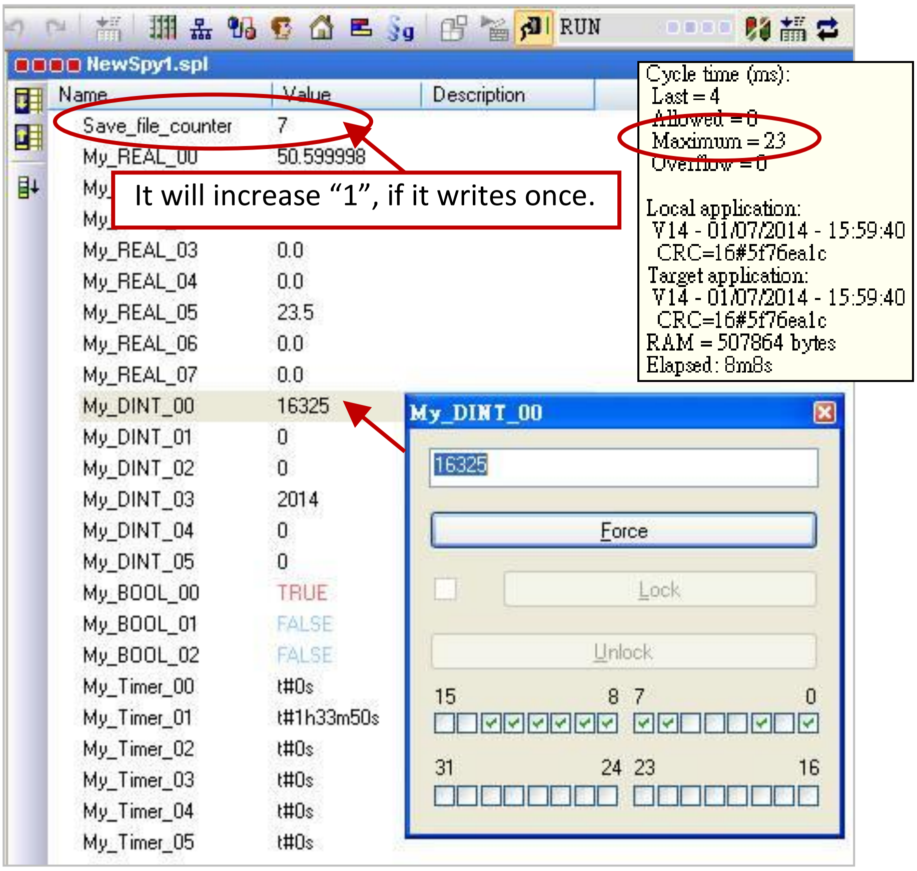
Note: The "Save_file_counter" will show the number of times the file is written, if this value is changing rapidly (e.g., to write several times every second/minute.), it is not suitable for this application (Because to write into files frequently in the "\System_disk" will reduce the PAC effectiveness.).



Name	Value	Description
Save_file_counter	0	
My_REAL_00	0.0	
My_REAL_01	0.0	
My_REAL_02	0.0	
My_REAL_03	0.0	
My_REAL_04	0.0	
My_REAL_05	0.0	
My_REAL_06	0.0	
My_REAL_07	0.0	
My_DINT_00	0	
My_DINT_01	0	
My_DINT_02	0	
My_DINT_03	0	
My_DINT_04	0	
My_DINT_05	0	
My_BOOL_00	FALSE	
My_BOOL_01	FALSE	
My_BOOL_02	FALSE	
My_Timer_00	t#0s	
My_Timer_01	t#0s	
My_Timer_02	t#0s	
My_Timer_03	t#0s	
My_Timer_04	t#0s	
My_Timer_05	t#0s	

```
Cycle time (ms):
 Last = 3
 Allowed = 0
 Maximum = 13
 Overflow = 0

Local application:
 V14 - 01/07/2014 - 15:59:40
  CRC=16#5f76ea1c
Target application:
 V14 - 01/07/2014 - 15:59:40
  CRC=16#5f76ea1c
RAM = 507864 bytes
Elapsed: 1m3s
```

Enter some values at will. When the value has changed, it will create a text file in the PAC (\System_disk\Win-GRAF\retain_real.txt and retain_other.txt) and Write data to the files.



Name	Value	Description
Save_file_counter	7	
My_REAL_00	50.599998	
My_		
My_		
My_REAL_03	0.0	
My_REAL_04	0.0	
My_REAL_05	23.5	
My_REAL_06	0.0	
My_REAL_07	0.0	
My_DINT_00	16325	
My_DINT_01	0	
My_DINT_02	0	
My_DINT_03	2014	
My_DINT_04	0	
My_DINT_05	0	
My_BOOL_00	TRUE	
My_BOOL_01	FALSE	
My_BOOL_02	FALSE	
My_Timer_00	t#0s	
My_Timer_01	t#1h33m50s	
My_Timer_02	t#0s	
My_Timer_03	t#0s	
My_Timer_04	t#0s	
My_Timer_05	t#0s	

It will increase “1”, if it writes once.

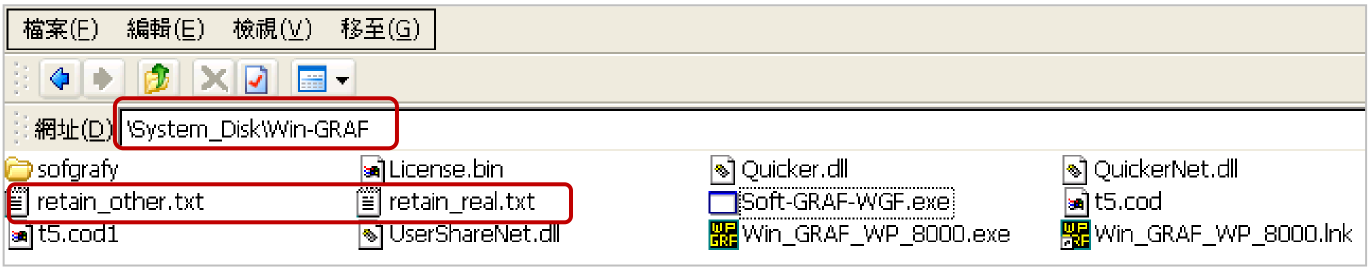
Cycle time (ms):
Last = 4
Allowed = 0
Maximum = 23
Overflow = 0

Local application:
V14 - 01/07/2014 - 15:59:40
CRC=16#5f76ea1c
Target application:
V14 - 01/07/2014 - 15:59:40
CRC=16#5f76ea1c
RAM = 507864 bytes
Elapsed: 8m8s

My_DINT_00

16325

Win-GRAF PAC:



網址(D) \System_Disk\Win-GRAF

sofgrafy
retain_other.txt
t5.cod1
License.bin
retain_real.txt
UserShareNet.dll
Quicker.dll
Soft-GRAF-WGF.exe
Win_GRAF_WP_8000.exe
QuickerNet.dll
t5.cod
Win_GRAF_WP_8000.lnk

1.5. Save Data to EEPROM

The Win-GRAF PAC has a built-in EEPROM memory for users to read and write data, which will not lose data when the PAC shut down. Compared to the read and write of the SRAM, EEPROM has the following disadvantages:

Note: Some PAC have no EEPROM memory (like the WP-5238, WP-5248). They don't support EEP_Read() and EEP_Write().

Advantages:

It provides another way to save the important data, besides the Retain Variable (Section 1.1).

Disadvantages:

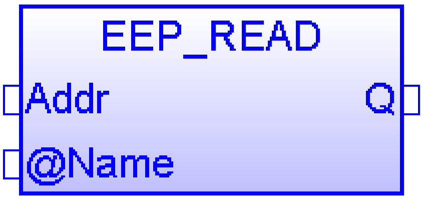
1. The operation to read/write EEPROM will use much more CPU time (about 5 ~ 50 ms), but changing to the way of "Retain variable", CPU time is much less than 1 ms. Therefore, do not use the "EEP_Read" and "EEP_Write" Functions too frequently, or it will increase the PAC Cycle time.
2. EEPROM has a “write” limitation (depending on the PAC), it is not suitable to write the same data many times. So, **DO NOT** call "EEP_Write" Function within each PAC Cycle to do the “write” operation.

ST Program: **(Following will show the safe and** dangerous coding ways.**)**

```
(* Declare "FIRST_CYCLE" as a “BOOL” variable and has an initial value “TRUE”.
   Declare "tmp_bool" as a “BOOL” variable 。
   Declare "New_Val" and "Old_Val" as “DINT” variables. *)
(* Read the EEPROM once in the first Cycle. *)

  if   FIRST_CYCLE   then
       FIRST_CYCLE   :=   FALSE ;   (*means it is not the first Cycle any more *)
       tmp_bool :=   EEP_Read ( 1 , New_Val) ;
  end_if ;

  (* Safe Coding Way: write to the EEPROM only when the value is changed. *)
  if   New_Val <> Old_Val   then
       Old_Val    :=   New_Val ;
       tmp_bool :=   EEP_Write ( 1 , New_Val ) ;
  end_if ;
```

```
(* Dangerous Coding Way: EEPROM may be destroyed very soon. *)
(* Declare "FIRST_CYCLE" as “BOOL” variable and has an initial value “TRUE”.
   Declare "tmp_bool" as “BOOL” variable.
   Declare "New_Val" and "Old_Val" as “DINT” variables. *)
(* Read the EEPROM once in the first Cycle. *)

if   FIRST_CYCLE   then
     FIRST_CYCLE   :=   FALSE ;   (* means it is not the first Cycle any more *)
     tmp_bool   :=   EEP_Read ( 1 , New_Val) ;
end_if ;

(*Dangerous Coding Way: Write the "New_Val" value to the EEPROM one time in every Cycle. *)
tmp_bool   :=   EEP_Write ( 1 , New_Val ) ;
```

1.5.1. EEP_READ (Read a Value from the EEPROM)



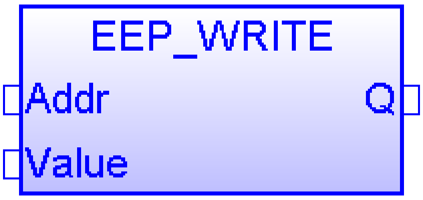
Tip:
Press “F1” key to see more details.

Addr: (Data Type: “DINT”)
Address, can be 1 to 1200. If the variable type of the "@Name" parameter is a 64-bit data (e.g., LINT or LREAL), the "Addr" can be 1001 to 1200 only.

@Name :
A variable name to store the value from the EEPROM.
(DO NOT use string variable. Variable type can be BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT, DWORD, REAL, TIME, LINT, or LREAL.)

Q:
Data Type: BOOL. TRUE: Ok; FALSE: Error.
If the type of the "@Name" parameter is REAL or LREAL, will return "Q" as FALSE if the value is NaN (Not a Number) or other error happens. In the case of NaN, the REAL/LREAL variable will get the value "0.0".

1.5.2. EEP_WRITE (Write a Value to the EEPROM)



Addr: (Data Type: “DINT”)
Address, can be 1 to 1200. If the variable type of the "Value" parameter is a 64-bit data (e.g., LINT or LREAL), the "Addr" can be 1001 to 1200 only.

Value :
The value to write to the EEPROM.
(DO NOT use string variable. The value type can be BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT, DWORD, REAL, TIME, LINT, or LREAL.)

Q:
Data Type: BOOL. TRUE: Ok; FALSE: Error.

Click the link for more Win-GRAF FAQ:
https://www.icpdas.com/en/faq/index.php?kind=273#840