
How to calculate the moving average value of a variable by c-functions “Aver_N” or “Aver_F” ?

By chun@icpdas.com

User may get this paper and the demo program “wpdmo81.pia” at the following web.

www.icpdas.com > FAQ > Software > ISaGRAF > 120 .

There is some applications to calculate the moving average value of a Real or Integer variable. The Real or Integer variable is sampled once and calculated the moving average value once at every fixed interval. To do that, user can use the “Aver_F” c-function to get the moving average value of a Real variable. (or the “Aver_N” c-function to get the moving average value of an integer variable).

Algorithm for the above application:

Consider the sampled data F1, F2, ..., F10, ..., Fn at each interval step and the given sample number to calculate is 5. Then the moving average value at each interval step will be as the following.

$$(5 \times F1) / 5, (4 \times F1 + F2) / 5, (3 \times F1 + F2 + F3) / 5, (2 \times F1 + F2 + F3 + F4) / 5, \\ (F1 + F2 + F3 + F4 + F5) / 5, (F2 + F3 + F4 + F5 + F6) / 5, \dots, \\ (F_{<n-4>} + F_{<n-3>} + F_{<n-2>} + F_{<n-1>} + F_n) / 5$$

The oldest sample value in the list will be erased sequentially by each interval step .

There is also some applications to calculate the average value of a Real or Integer variable. The Real or Integer variable is sampled once every fixed interval and calculated the average value during a given number of samples. To do that, user can use the “Gt_Ave_R” function block to get the average value of a Real variable. (or the “Gt_Ave_N” function block to get the average value of an integer variable).

(Please refer to www.icpdas.com > FAQ > Software > ISaGRAF > 099)

The “Aver_N” and “Aver_F” are ISaGRAF c-functions. The following ISaGRAF PAC drivers version (or later versions) support them.

WP-8xx7: driver Ver. 1.20 , VP-25W7/23W7: driver Ver. 1.11 , XP-8xx7-CE6: since released.

The “Aver_N” and “Aver_F” can apply in the “ for ... end_for ; “ loops of the ISaGRAF ST program . And also they can apply in the Ladder program one by one. (refer to the FAQ-120).

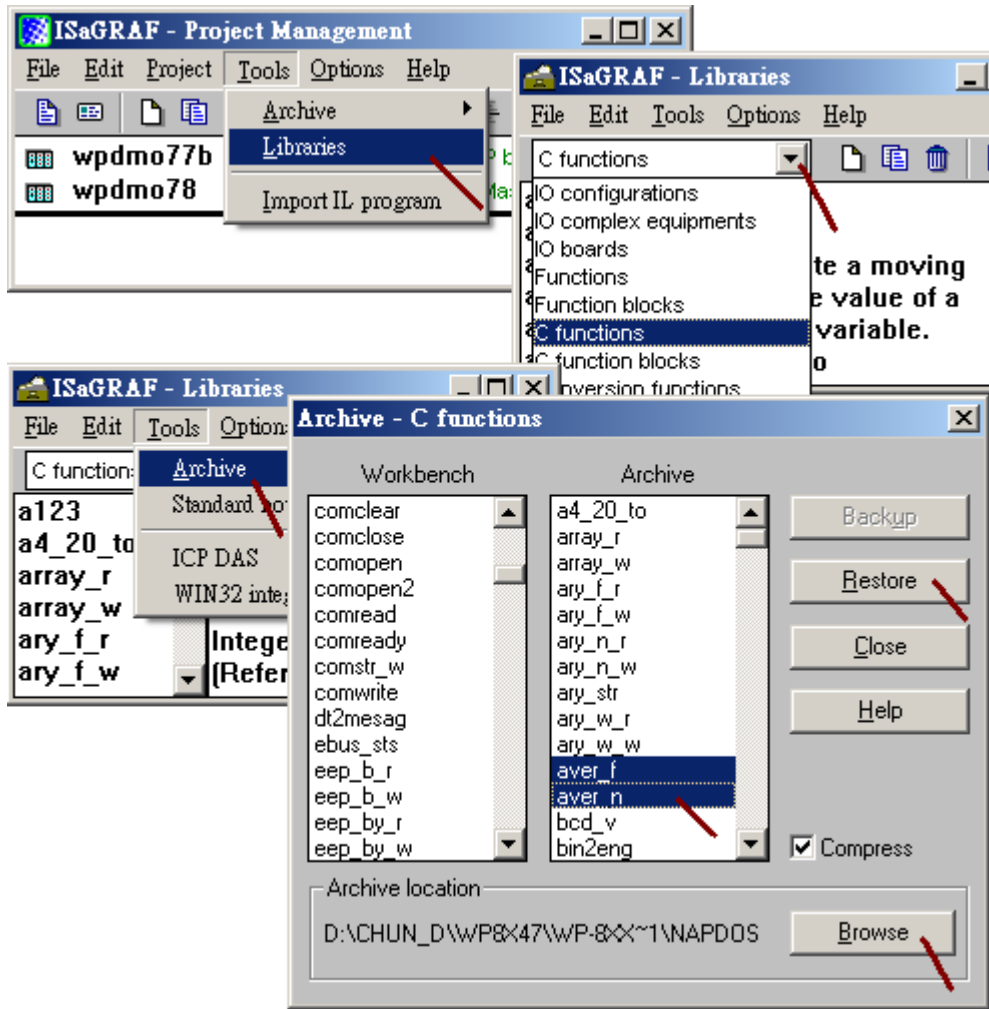
The “Averag_N” and “Averag_F” are ISaGRAF c-function-blocks. They can not apply in the ”for ... end_for ; “ loops. They can only apply in the program one by one. The following PAC support “Averag_N” and “Averag_F” .

WP-8xx7, VP-25W7/23W7, XP-8xx7-CE6, W-8xx7, iP-8xx7, uPAC-7186EG . (refer to the FAQ-99)

The “Gt_Ave_N” and “Gt_Ave_R” are ISaGRAF function-blocks. They can not apply in the ”for ... end_for ; “ loops. They can only apply in the program one by one. The following PAC support “Gt_Ave_N” and “Gt_Ave_R” .

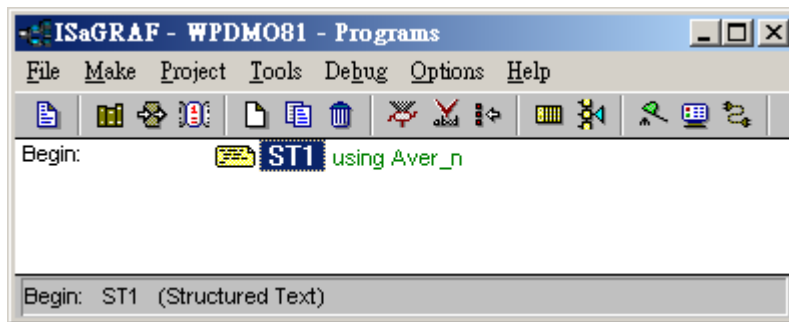
WP-8xx7, VP-25W7/23W7, XP-8xx7-CE6, W-8xx7, iP-8xx7, uPAC-7186EG , i-8xx7-80, i-8xx7, i-7188EG/XG. (refer to the FAQ-99)

To use the “Aver_N” and “Aver_F”, must restore them to your PC / ISaGRAF first. And make sure if the ISaGRAF PAC driver has support them (refer to the first page)

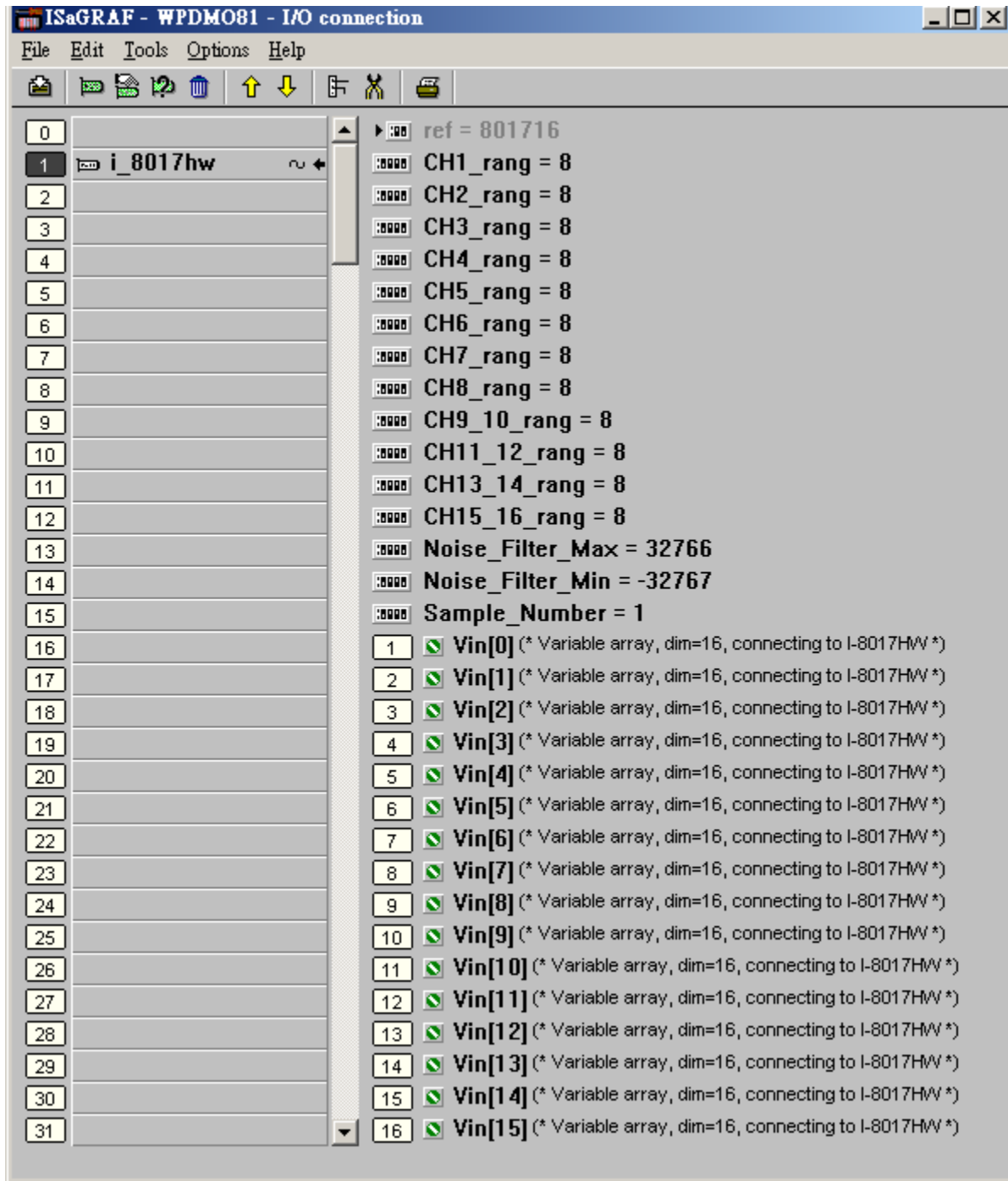


The demo program “wpdmo81.pia” can be obtained from the FAQ-120. It uses the hardware WP-8xx7 plus one i-8017HW in its I/O slot 1(Note: the leftmost I/O slot No. of the WP-8xx7 is 0). Please set the jumper of the I-8017HW to “single-end” and refer to the FAQ-039 for using ISaGRAF Variable array.

Project - “wpdmo81” :

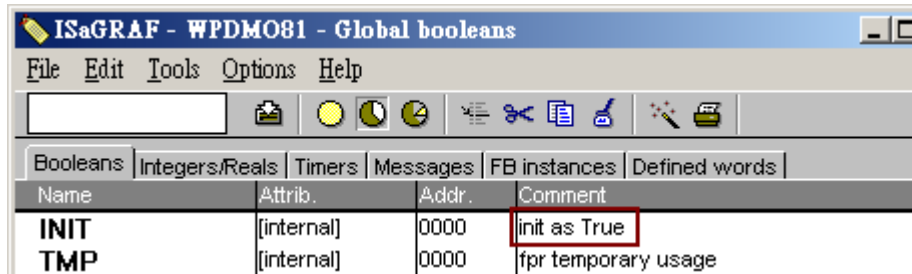


IO connection:



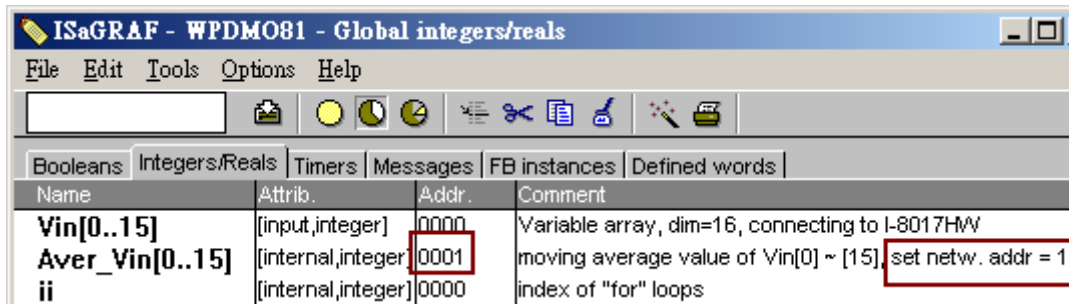
Variables:

Boolean:



Name	Attrib.	Addr.	Comment
INIT	[internal]	0000	init as True
TMP	[internal]	0000	fpr temporary usage

Integers/Reals:



Name	Attrib.	Addr.	Comment
Vin[0..15]	[input, integer]	0000	Variable array, dim=16, connecting to I-8017HW
Aver_Vin[0..15]	[internal, integer]	0001	moving average value of Vin[0] ~ [15], set netw. addr = 1
ii	[internal, integer]	0000	index of "for" loops

ST1 program:

```
(* operations in the 1st PLC scan, must declare "INIT" with an initial value TRUE *)
if INIT then

  INIT := False ; (* no more *)

  (* Must declare "Aver_Vin" 's network address as 1 in the dictionary.
  assign Modbus network address No. 1,2,3, ..., 16 to Variable Array Aver_Vin[0] ~ [15] .
  then PC/SCADA/HMI can access to them by Modbus TCP or RTU protocol *)
  TMP := s_mb_adr( 1 , 16 , 0 ) ;

end_if ;

(* Calculate a moving vaerage value for Vin[0] ~ [15] and store the result to Aver_Vin[0] ~
[15]. Using "Aver_N" ID from 1 to 16 . (Max. 1024 "Aver_N" and "Aver_F" , ID=1 to
1024) . The following code apply sample number as 10 for each "Aver_N".
*)
for ii := 0 to 15 by 1 do

  (* The last parameter (can be T#0ms, or T#10ms to T#1h), setting T#0ms means to
  sample and calculate the moving average once at each calling "Aver_N". *)
  Aver_Vin[ii] := Aver_N( ii+1 , TRUE , Vin[ii] , 10 , T#0ms ) ;

end_for ;
```

User may also program the “Aver_N” and “Aver_F” in the Ladder program as the following (Then you need to program 16 “Aver_N”).

ID can be 1 to 1024.

RUN : Set as True to start sampling and calculating. If set as False, then return Q_ = IN_ (no calculation).

IN_ : the variable value to be calculated.

N_ : number of samples. Can be 1 to 50.

TIME_ : Can be T#0ms or T#10ms to T#1h. The interval to sample and calculate once. Set as T#0ms means to sample and calculate once at each calling it.

