
如何使用 Aver_N 與 Aver_F 來計算連續平均值 (或稱 移動平均值) ?

By chun@icpdas.com

本文件與範例 wpdmo81.pia 可在 www.icpdas.com > FAQ > Software > ISaGRAF > 120 取得.

有一些應用會每隔一段時間或每次調用它時對一個 整數 或 實數取樣累計一次, 然後求出 連續平均值 (或稱 移動平均值). 比如 每 0.5 秒對某個 實數值累計一次, 每次取樣後算出連續 5 個取樣的其平均值 (前 4 次 加上 本次取樣的平均值)

要達到此功能, 可以編寫一個 “Aver_F” (屬於 ISaGRAF c-function) 來求出一個實數的 連續平均值 (或 “Aver_N” 來求出一個整數的 連續平均值).

其演算法為, 例如連續有 n 次的取樣值為 F1, F2, ..., F10, ..., Fn 然後求 5 個資料的連續平均值, 則其連續平均值依序會是

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

每次計算時, 最早記錄的 1 個資料會依序被刪除掉.

另外有一些應用會每隔一段時間對 1 個 整數 或 實數取樣累計 1 次, 之後累計到一定數量後再求出其平均值 . 比如每秒對某個 Integer input 值累計 1 次, 然後累計 10 次後, 求出其平均值 . 要達到此功能, 可以編寫一個 “Gt_Ave_R” 方塊來求出 1 個實數的平均值 (或 “Gt_Ave_N” 求出 1 個整數的平均值). 請參考 www.icpdas.com > FAQ > Software > ISaGRAF > 099

Aver_N 與 Aver_F 屬於 ISaGRAF c-function, 以下的 ISaGRAF PAC 版本起才有支持.

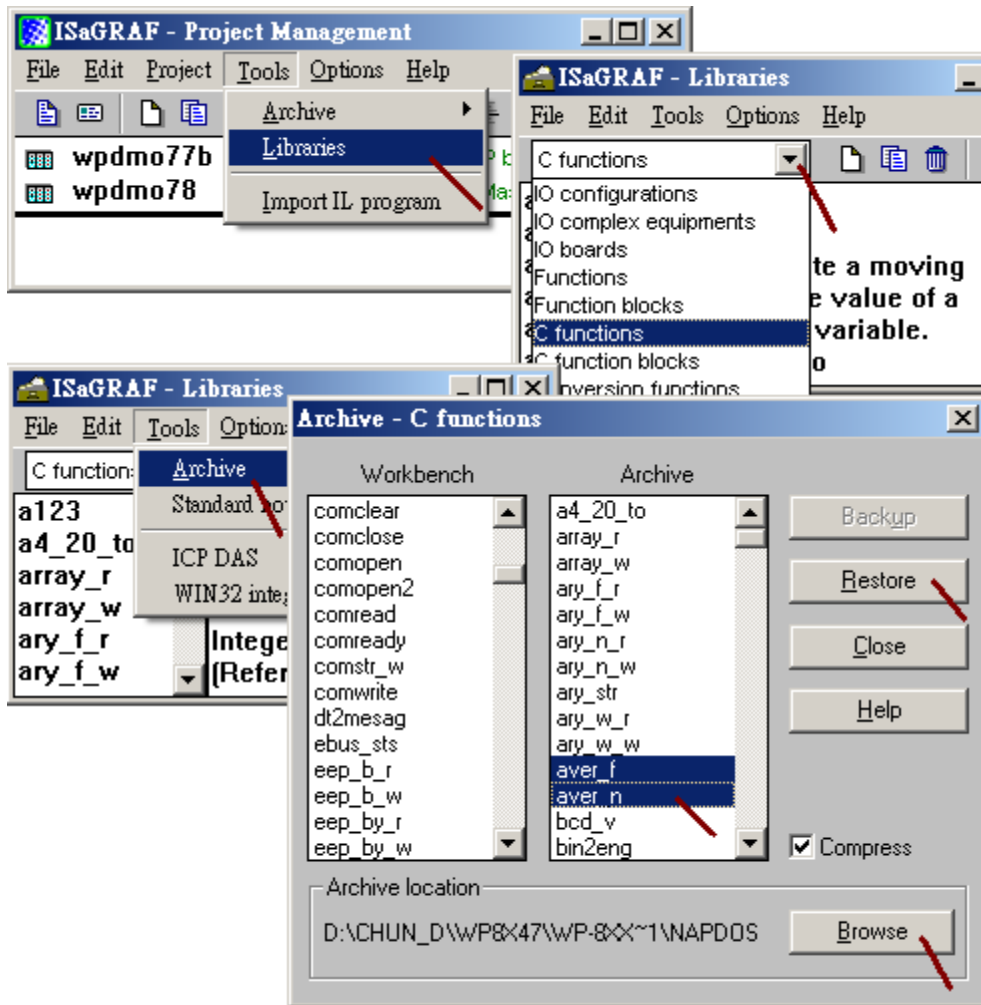
WP-8xx7: driver 1.20 版 , VP-25W7/23W7: driver 1.11 版 , XP-8xx7-CE6: 從 release 之日起 (新的 ISaGRAF driver 請參訪 <http://www.icpdas.com/products/PAC/i-8000/isagraf-link.htm>)

這 2 個 c-function 可以在 ISaGRAF 的 ST 程式內的 “for ... end_for ;” 迴圈內重覆使用, 也可在階梯圖內單獨 1 個 1 個來使用. (參考 FAQ-120).

Averag_N 與 Averag_F 屬於 ISaGRAF c-function-block, 不能在 ”for ... end_for ;” 迴圈內重覆使用, 只能單獨 1 個 1 個來使用, 可以用在 WP-8xx7, VP-25W7/23W7, XP-8xx7-CE6, Wincon-8xx7, iP-8xx7, uPAC-7186EG . (參考 FAQ-99)

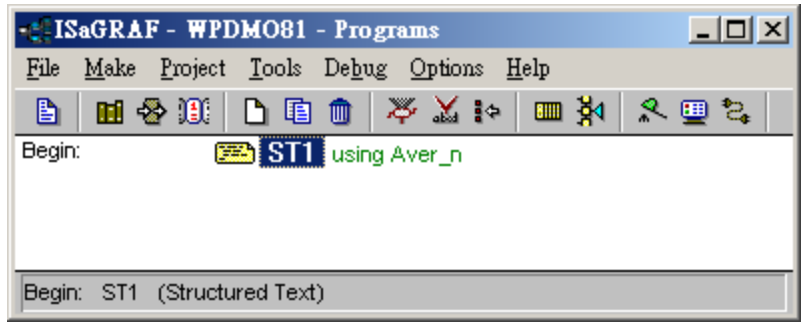
Gt_Ave_N 與 Gt_Ave_R 屬於 ISaGRAF function-block, 不能在 ”for ... end_for ;” 迴圈內重覆使用, 只能單獨 1 個 1 個來使用 . 可以用在 WP-8xx7, VP-25W7/23W7, XP-8xx7-CE6, Wincon-8xx7, iP-8xx7, uPAC-7186EG , i-8xx7-80, i-8xx7, i-7188EG/XG. (參考 FAQ-99)

要使用 Aver_N 與 Aver_F 前需先安裝到你的 PC / ISaGRAF 內, 另外也請確認 ISaGRAF PAC driver 版本是否有符合 (上一頁有說明)

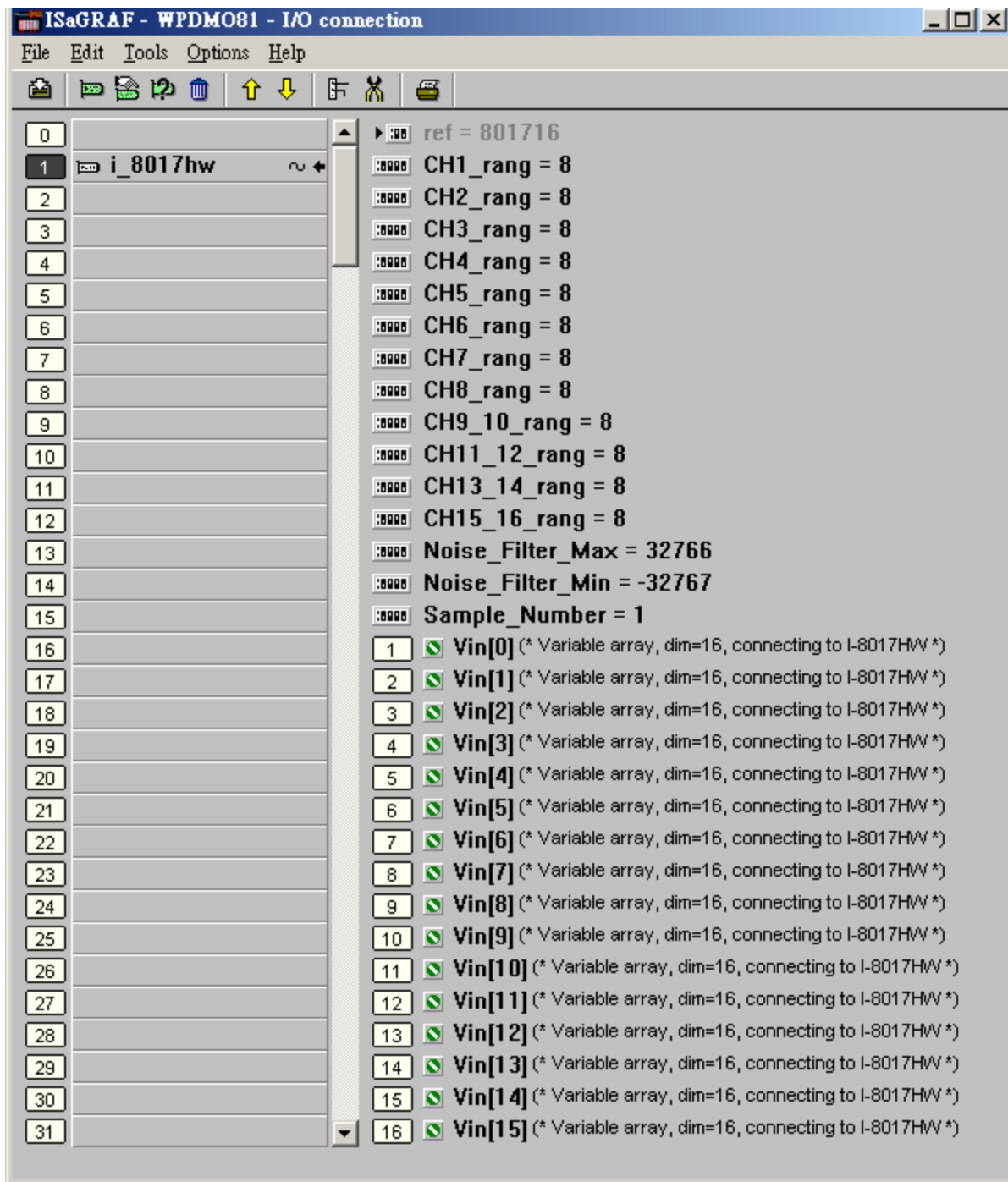


範例程式 wpdmo81.pia 可由 FAQ-120 取得, 它使用 WP-8xx7 + slot 1: i-8017HW (WP-8xx7 的最左方是 slot 0), I-8017HW 上的 Jumper 使用 16 Channel (Single-End 的接線方式). 並搭配使用 ISaGRAF Variable Array (關於 Variable Array, 請參考 FAQ-039 的說明)

Project - "wpdmo81":

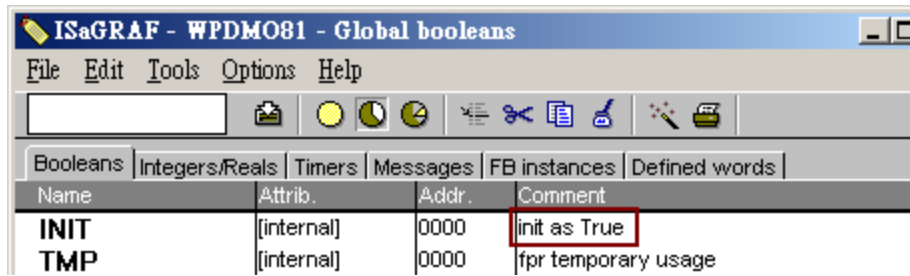


IO 連結:



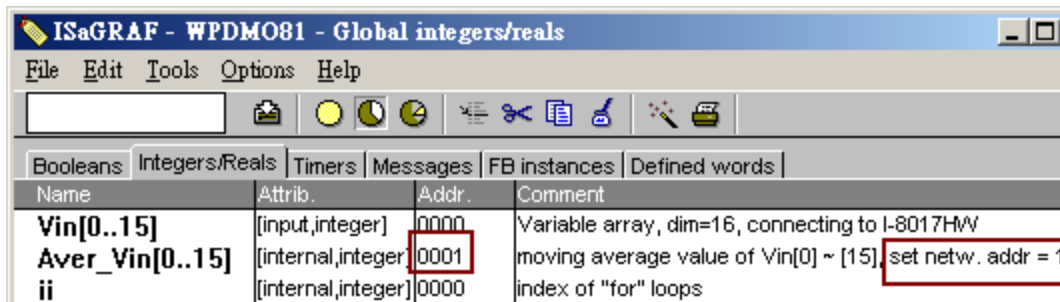
變數:

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 程式:

```
(* 第 1 個 PLC Scan 內的動作, 必需在 Dictionary 內宣告 INIT 初值為 TRUE *)
if INIT then

  INIT := False ; (* 不再是第 1 個 PLC Scan *)

  (* 必需在 Dictionary 內宣告“Aver_Vin”的 network address 為 1.
  下方的 s_mb_adr 用來設定 network address 1, 2, 3, ..., 16 給 Aver_Vin[0] ~ [15].
  之後 PC/SCADA/HMI 就可使用 Modbus TCP 或 RTU protocol 來讀/寫 它們
  *)
  TMP := s_mb_adr( 1 , 16 , 0 ) ;

end_if ;

(* 計算 Vin[0] ~ [15]的連續平均值 並存放在 Aver_Vin[0] ~ [15]. 使用 "Aver_N"
ID = 1 到 16 來計算 .(最多 1024 個 Aver_N, ID 可以是 1 ~ 1024). 下方是使用
Smample 數量 10 來計算. *)
for ii := 0 to 15 by 1 do

  (* 最後一個輸入值 可以是 T#0ms 或 T#10ms ~ T#1h, 若為 T#0ms 表示每次調用
  它就取樣與計算一次. *)
  Aver_Vin[ii] := Aver_N( ii+1 , TRUE , Vin[ii] , 10 , T#0ms ) ;

end_for ;
```

也可以在階梯圖內單獨一個一個 調用“Aver_N” 與“Aver_F”
(那本例就需要使用 16 個“Aver_N”).

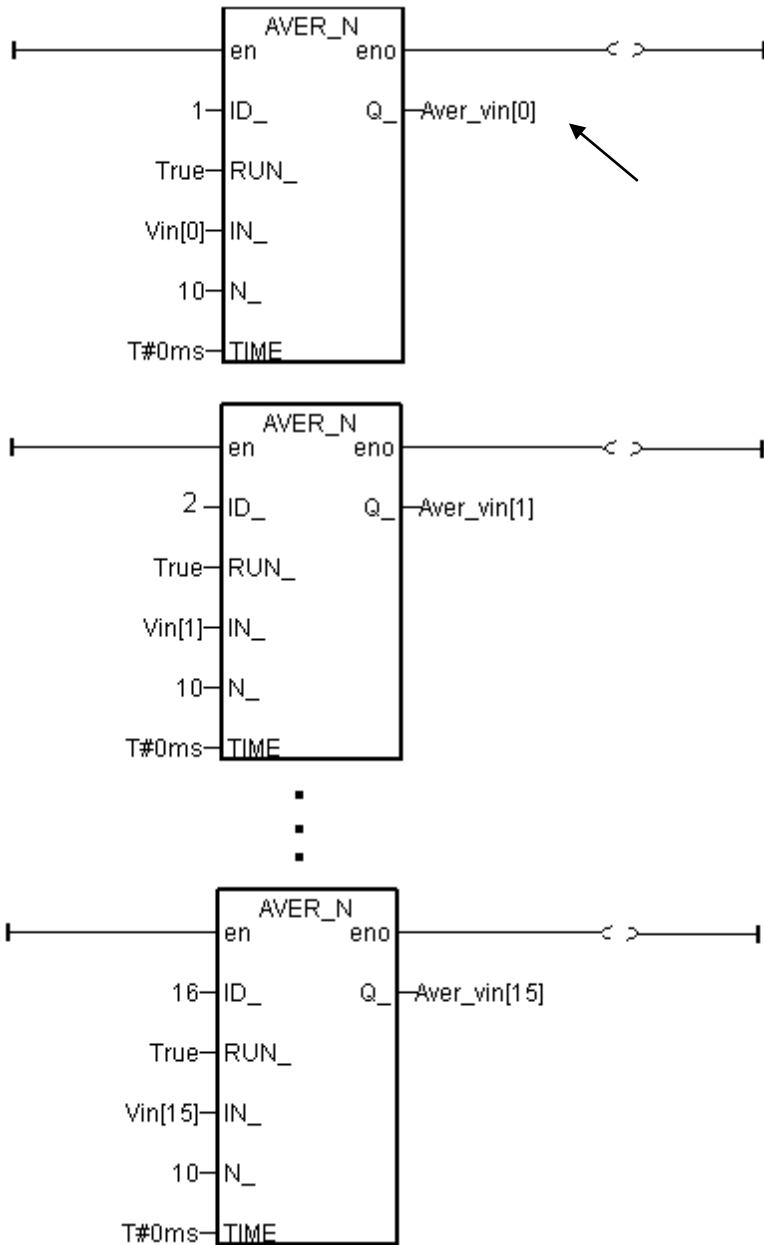
ID 可以是 1 ~ 1024.

RUN : True 則開始進行計算,
若為 False 直接 傳回 Q_ =
IN_

IN_ : 要被取樣 與 計算的值.

N_ : 取樣的數量. 可以是 1 ~
50

TIME_ : 可以是 T#0ms 或
T#10ms ~ T#1h
每隔多久取樣與計算一次.
設為 T#0ms 表示, 每使用
它



算出來的 連續平均值.