Using many Modbus function blocks Mbus_AR and Mbus_AW in a “for” loop in the ISaGRAF PAC

This paper show the way to call Mbus_AW and Mbus_AR function block many times in a “for .. end_for ;” loop to write data to or read data from some Modbus devices.

User can download this paper and example projects from the following web site. https://www.icpdas.com/en/faq/index.php?kind=280#751 > FAQ-161 . The ISaGRAF driver version listed below support the Mbus_AR and Mbus_AW block. If your PAC 's driver is older than it, please visit http://www.icpdas.com/en/download/show.php?num=368&nation=US&kind1=&model=&kw=isagraf to download the newest driver.

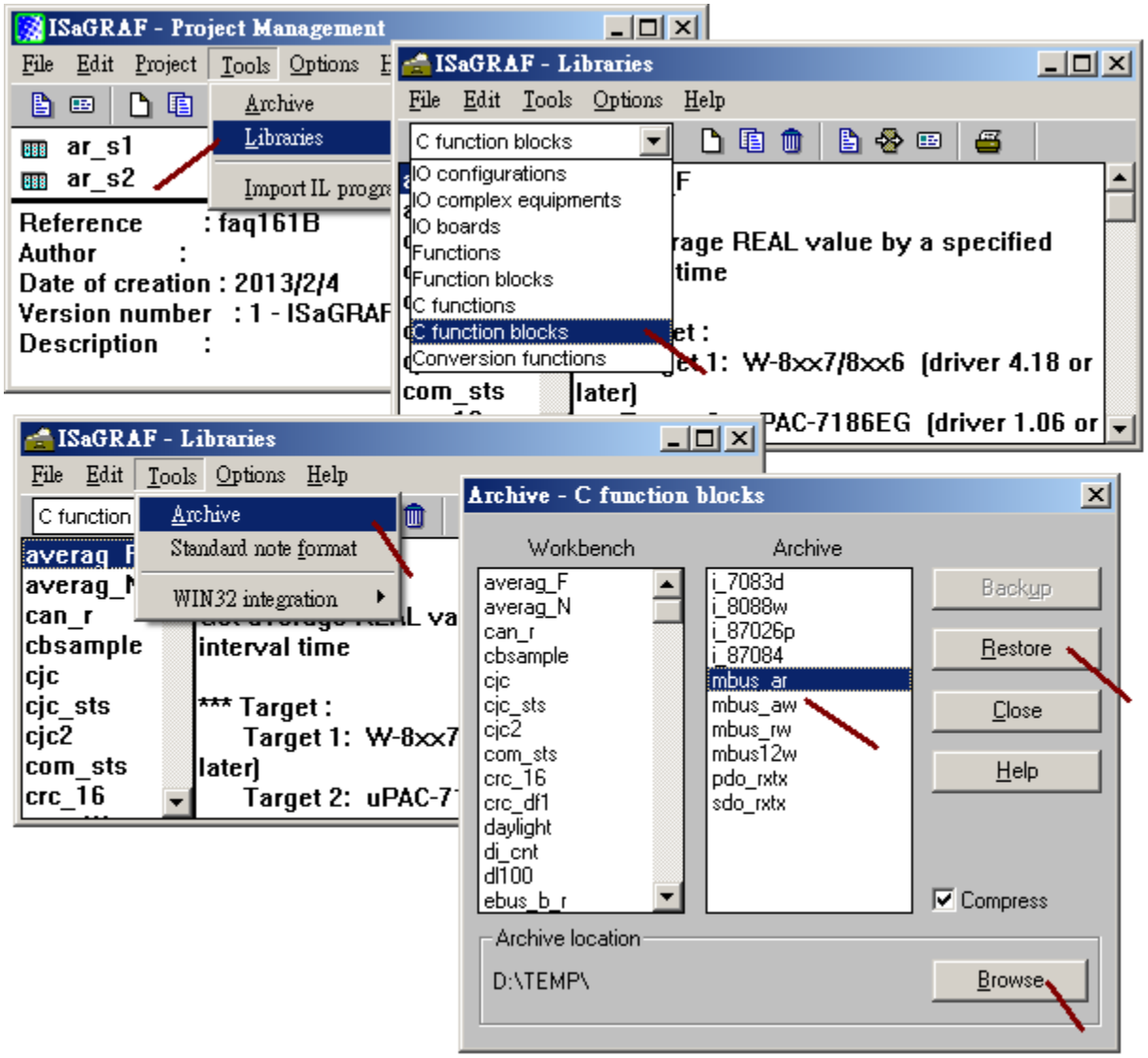
WP-8xx7: version 1.57 or later VP-25W7/23W7: version 1.49 or later
XP-8xx7-CE6: version 1.37 or later WP-5147 : version 1.04 or later
XP-8xx7-ATOM-CE6: version 1.02 or later

ISaGRAF PAC supports Modbus master function to connect other Modbus slave devices to read or write data. For example, the Mbus24R is to read max. 24 words (16-bit signed) or max. 12 REAL (32-bit floating point) or max. 12 long integer (32-bit long). User may refer to the Chapter 8 of the “ISaGRAF User's manual” for more information or visit the following web site . https://www.icpdas.com/en/faq/index.php?kind=280#751 > FAQ-75, 46, 47, 96, 101, 113, 129, 144 ,159 .

When using many Modbus function blocks in a PAC (for instance, using 50 or may be 100 to 2000 Modbus blocks), the ISaGRAF project become huge. This make the program complicated and difficult to maintain. The major reason is the input parameters of those Modbus blocks like Mbus24R, Mobus12W, Mobus_N_R, ... must be a constant. So they can not apply in a ST program 's “for .. end_for ;” loop. Then the program becomes huge. While if using the Mbus_AR and Mbus_AW blocks, all of their input parameters can be a variable value. Then if apply them in a “for .. end_for ;” loop, the program is much smaller and easy to maintain. The other advantage is the Mbus_AR and Mbus_AW blocks consume less CPU time .

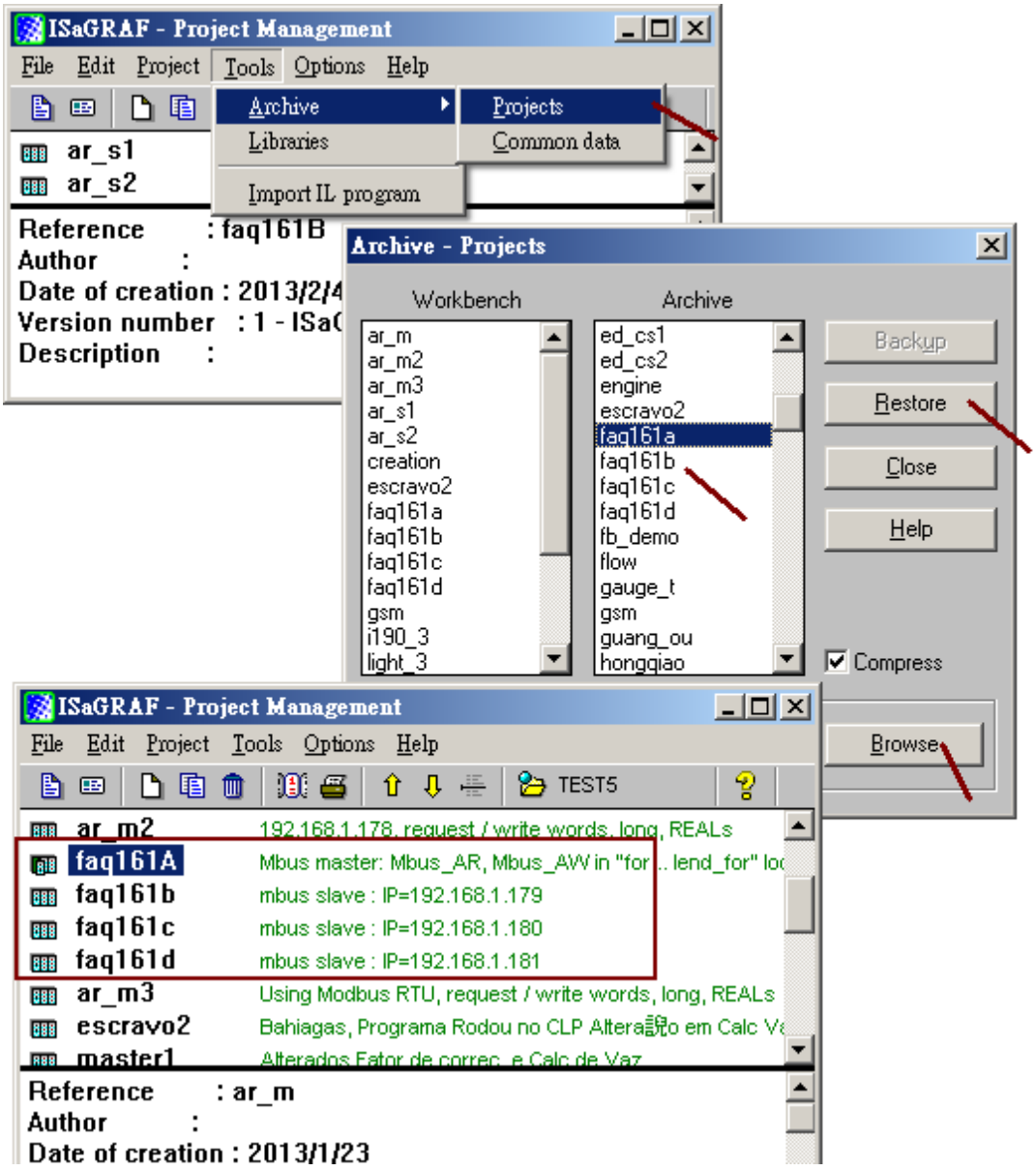
LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

First retore the “mbus_aw.fia” and “mbus_ar.fia” c-function blocks to your PC / ISaGRAF. These two block's files are included in the “faq161_demo.zip” which you download from the FAQ-161.



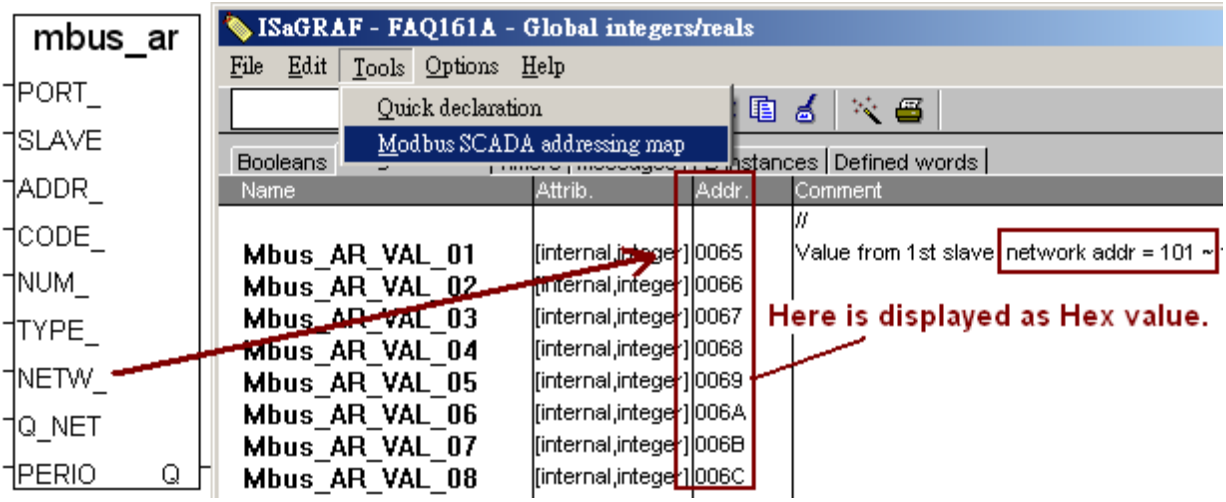
Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

This “faq161_demo.zip” also includes four example projects - faq161a, faq161b, faq161c and faq161d. Please restore them to the PC / ISaGRAF as the below figure. The major example project is the “faq161a” which enables an ISaGRAF WinCE PAC (WP-8xx7, WP-5147, VP-25w7 / 23W7, XP-8xx7-CE6, CP-8xx7-ATOM-CE6) as a Modbus TCP master to connect three Modbus TCP slave devices (IP address are 192.168.1.179 , 192.168.1.180 and 192.168.1.181 respectively, NET-ID are all 1) .



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

The Mbus_AR block is to eanble the PAC to read max. 24 words (16-bit signed) or max. 12 long integers (32-bit long) or max. 12 REAL (32-bit floating point) or max. 200 Booleans from a Modbus slave device. It can apply only in the first scan cycle. Applying it in the 2nd , 3rd and later scan cycles has no meanning. One PAC can apply (call) this Mbus_AR block max. 2000 times in the 1st scan cycle. Each calling Mbus_AR means enable one block to read data from the specified slave device.

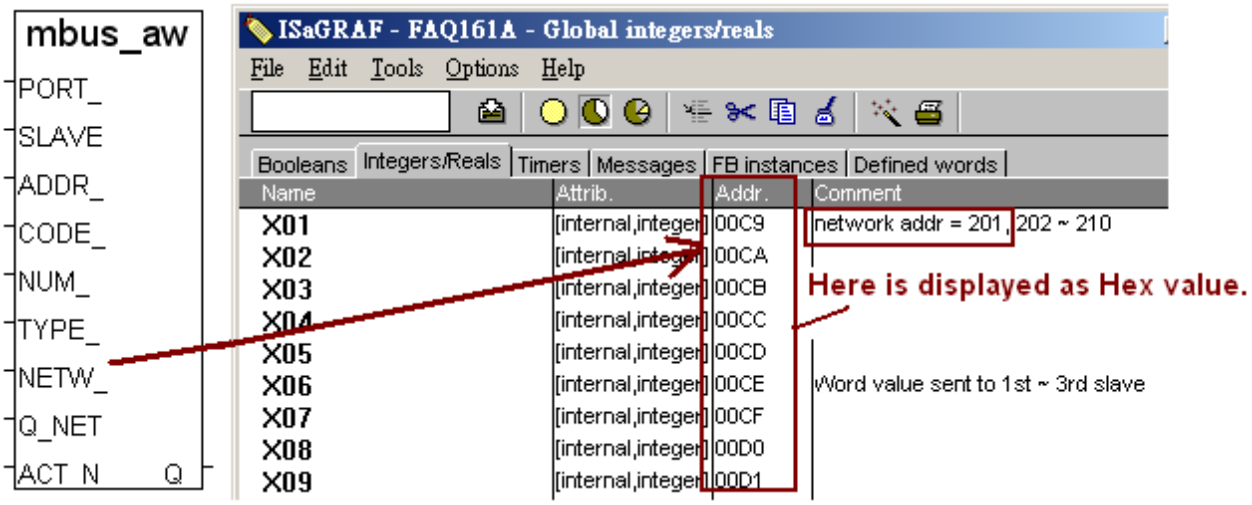


All the following parameters can be a variable value.

PORT	For Mbus_tcp, set it as the value returned from the 1st channel in the IO connection > Mbus_tcp. For Mnus and Mbus_asc, set it as the comm. port number of that RS-232 , RS-485 or RS-422 port.
SLAVE	The NET-ID number of Modbus slave device , 1 ~ 255.
ADDR	The starting Modbus addr. number of slave device (0 or 1 ~ 65535)
CODE	Modbus function number, 1 ~ 4 (1, 2 for Booleans, 3, 4 for Word, Long, REAL).
NUM	data amount to read. Word: 1 ~ 24, Long and REAL: 1 ~ 12. Boolean: 1 ~ 200
TYPE	1: data is Word (16-bit signed integer) , 2: data is Long (32-bit signed integer) 3: data is Real (32-bit floating point), Note: if CODE is 1 or 2, TYPE is ignored.
NETW	1 ~ 7700. The starting network address of those ISaGRAF variables which will receive the data successfully read from the slave device. For CODE 1 and 2, the related ISaGRAF variables should be declared as Boolean. For CODE 3 and 4, if TYPE is 1 or 2, the related ISaGRAF variables should be declared as integers . If TYPE is 3, should be declared as REAL variables . Note: please set the network address of the related ISaGRAF variable to jump one number for TYPE 2 and 3. For example, set them as 101, 103, 105, ...
Q_NETW	0 or 1 ~ 7700, the network address number of an ISaGRAF Boolean variable which will receive the communication state of the Mbus_AR block. Set as 0 means no need to get this communication state.
PERIOD	0 or 1 ~ 600, unit is second. The interval to send this Modbus command . Set 0 means send it continuously.

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

The Mbus_AW block is to eanble the PAC to write max. 12 words (16-bit signed) or max. 6 long integers (32-bit long) or max. 6 REAL (32-bit floating point) or max. 16 Boolean to a Modbus slave device. It can apply only in the first scan cycle. Applying it in the 2nd , 3rd and later scan cycles has no meanning. One PAC can apply (call) this Mbus_AW block max. 2000 times in the 1st scan cycle. Each calling Mbus_AW means enable one block to write data to slave device.



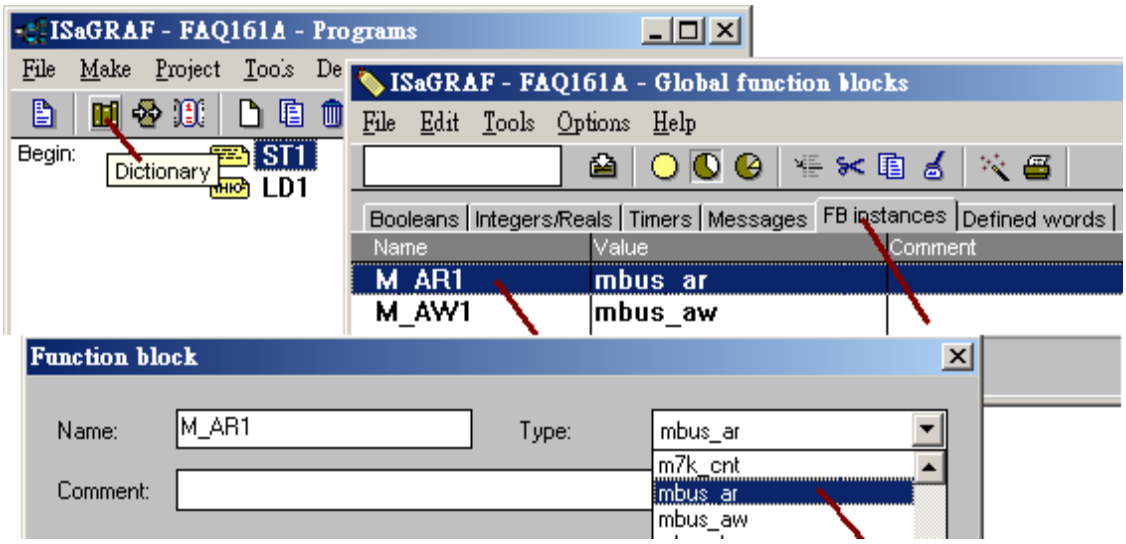
All the following parameters can be a variable value.

PORT	For Mbus_tcp, set it as the value returned from the 1st channel in the IO connection > Mbus_tcp. For Mnus and Mbus_asc, set it as the comm. port number of that RS-232 , RS-485 or RS-422 port.
SLAVE	The NET-ID number of Modbus slave device , 1 ~ 255.
ADDR	The starting Modbus addr. number of slave device (0 or 1 ~ 65535)
CODE	Modbus function call number, 6 or 16 (or 5 or 15 for boolean) Note: function call 6 can write only 1 word. While many data for function 16. function call 5 can write only 1 boolean. While 1 to 16 booleans for function 15.
NUM	The data amount to write. Word : 1 ~ 12, Long and REAL : 1 ~ 6 If CODE is 5, NUM is ignored. If CODE is 15, NUM can be 1 to 16 .
TYPE	1: data is Word (16-bit signed integer) , 2: data is Long (32-bit signed integer) 3: data is Real (32-bit floating point). TYPE is ignored if CODE is 5 or 15 .
NETW	1 ~ 7700. The starting network address of those ISaGRAF variables which will hold the data being written to the slave device. If CODE is 5 or 15, the related ISaGRAF variables should be declared as Boolean. However, if CODE is 6 or 16 and TYPE is 1 or 2, the related ISaGRAF variables should be declared as integers with proper network address. If CODE is 6 or 16 and TYPE is 3, should be declared as REAL variables . Note: please set the network address of the related ISaGRAF variable to jump one number for TYPE 2 and 3. For example, set them as 101, 103, 105, ...
Q_NETW	0 or 1 ~ 7700, the network address number of an ISaGRAF Boolean variable which will receive the communication state of the Mbus_AW block. Set as 0 means no need to get this communication state.

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

Act_NETW	0 or 1 ~ 7700, the network address number of an ISaGRAF Boolean variable which controls the Mbus_AW block to write or stop writing. Set as 0 means write it continuously.

To use Mbus_AR and Mbus_AW , first declare FB instance “M_AR1” and “M_AW1” as the following figure.



The below ST program is from the example project “faq161a” . It runs only in the first scan cycle to enable three M_AR1 to read 24 words in each slave device (3 devices). The slave NET-ID are 1. The data address in slave is starting from 0 . Using Modbus function 4 to read 24 words. The data successfully read will be stored in those ISaGRAF integer variables with network address (101 ~ 124) , (125 ~ 148) and (149 ~ 172) respectively . The communication state will be stored in the ISaGRAF Boolean variable with network address 1 , 2 and 3 respectively. The last parameter value 2 means the interval to read data is 2 second.

Note: please set the network address of the related ISaGRAF variable to jump one number for TYPE 2 and 3. For example, set them as 211 , 213 , 215 , 217 , ...

```
if   INIT   then     (* This example declare“INIT” with an initial value TRUE *)
   INIT := FALSE ;
   TCP_NO[0] := TCP_ID01 ;    (* Ch.1 value in the first mbus_tcp in the IO connection *)
   TCP_NO[1] := TCP_ID02 ;    (* Ch.1 value in the 2nd mbus_tcp in the IO connection *)
   TCP_NO[2] := TCP_ID03 ;    (* Ch.1 value in the 3rd mbus_tcp in the IO connection *)
   for   ii   :=   0   to   2   do
   (* Mbus_AR(PORT, SLAVE, ADDR, CODE, NUM, TYPE, NETW, Q_NETW, PERIOD);*)
   (* TYPE = 1: data is wrod , 2: data is long, 3: data is REAL *)
      M_AR1( TCP_NO[ii] , 1 , 0    , 4 , 24 , 1 , 24*ii+101 , ii+1 , 2 ) ;
   end_for ;
end_if ;
```


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

The below ST program is from the example project “faq161a” . It runs only in the first scan cycle to enable three M_AW1 to read 5 REAL value in each slave device (3 devices). The slave NET-ID are 1. The data address in slave is starting from 110 (they are 110, 112, 114, 116 and 118) . Using Modbus function 16 to write 5 REAL. The data being written are stored in those ISaGRAF REAL variables with network address 211 , 213 , 215 , 217 and 219 respectively . No need to get the communication state of the Mbus_AW block (because the 2nd last parameter is 0) . The network address number of those ISaGRAF Boolean variables which control the Mbus_AW block to write or stop writing are 14 , 15 and 16.

Note: please set the network address of the related ISaGRAF variable to jump one number for TYPE 2 and 3. For example, set them as 211 , 213 , 215 , 217 , ...

```
if  INIT  then     (* This example declare“INIT” with an initial value TRUE *)
   INIT := FALSE ;
   TCP_NO[0] := TCP_ID01 ;   (* Ch.1 value in the first mbus_tcp in the IO connection *)
   TCP_NO[1] := TCP_ID02 ;   (* Ch.1 value in the 2nd mbus_tcp in the IO connection *)
   TCP_NO[2] := TCP_ID03 ;   (* Ch.1 value in the 3rd mbus_tcp in the IO connection *)
   for  ii  :=  0  to  2  do
    (* Mbus_AW( PORT, SLAVE, ADDR, CODE, NUM, TYPE, NETW, Q_NETW, ACT_NETW ); *)
    (* TYPE = 1: data is wrod , 2: data is long, 3: data is REAL *)
     M_AW1( TCP_NO[ii] , 1 , 110   , 16 , 5 , 3 , 211 , 0 , ii+14 ) ;
   end_for ;
end_if ;
```

Action of the Boolean variable which controls the Mbus_AW to write or stop writing :
If the Boolean value is TRUE, it make the Mbus_AW to write data.
If the Boolean value is FALSE, it make the Mbus_AW to stop writting data.
If the Boolean value is a pulse TRUE, it make the Mbus_AW to write data one time.

If set the “ACT_NETW” parameter as 0 (Network addr = 0 means no related ISaGRAF variable), this Mbus_AW block will write data continuously .

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