How to reduce ISaGRAF code size ? How to directly Read / Write ISaGRAF variables by using Network address ?

I-7188EG driver ver. 2.03 or later, I-7188XG driver ver. 2.02 or later, I-8xx7 driver ver. 3.05 or later and W-8x37 driver ver. 3.15 or later all support directly Read / Write ISaGRAF variables by using Network address. There are four functions can be used for this purpose.

R_MB_REL	Read “Real” variable by using Modbus Network address
W_MB_REL	Write “Real” variable by using Modbus Network address
R_MB_ADR	Read “Boolean” or “Integer” variable by using Modbus Network address
W_MB_ADR	Write “Boolean” or “Integer” variable by using Modbus Network address

The R_MB_xxx and W_MB_xxx functions can directly access to the ISaGRAF variables without using the variable name. It is very similar to using pointer in the C program. Since ISaGRAF workbench of ver. 3.3x , 3.4x and 3.5x doesn’t support variable array, these functions are very useful for reducing the ISaGRAF code size. For example,

Doing scaling form (-32768 , +32767) to (-10.0 , +10.0) volt.
AI_8017_01 to AI_8017_24 are declared as Input integer with Network addr 1, 2, … 24.
Volt_8017_01 to Volt_8017_24 are declared as Internal Real with Network addr 31, 33, 35, … 77

Using variable name: (24 ST statements)

```
Volt_8017_01 := Real(AI_8017_01) * 0.0003051757 ;
Volt_8017_02 := Real(AI_8017_02) * 0.0003051757 ;
Volt_8017_03 := Real(AI_8017_03) * 0.0003051757 ;
…
Volt_8017_24 := Real(AI_8017_24) * 0.0003051757 ;
```

Only a “For … End_For” command by using R_MB_xxx and W_MB_xxx functions.

```
For  ii  :=  1  to  24  Do              (* ii and jj are declared as internal integer *)
    Tmp_int := R_MB_ADR(1 , ii) ;           (* Tmp_int is internal integer *)
    jj := 29 + 2*ii ;                   (* jj = Network addr 31, 33, 35, … 77 *)
    Tmp_real := Real(Tmp_int) * 0.0003051757 ;      (* Tmp_real is internal Real *)
    Tmp_boo := W_MB_REL(jj , Tmp_real) ;      (* Tmp_boo is internal boolean *)
End_For ;
```