Working eLogger HMI with ISaGRAF SoftLogic in the WP-8xx7, VP-2xW7 and XP-8xx7-CE6 PAC

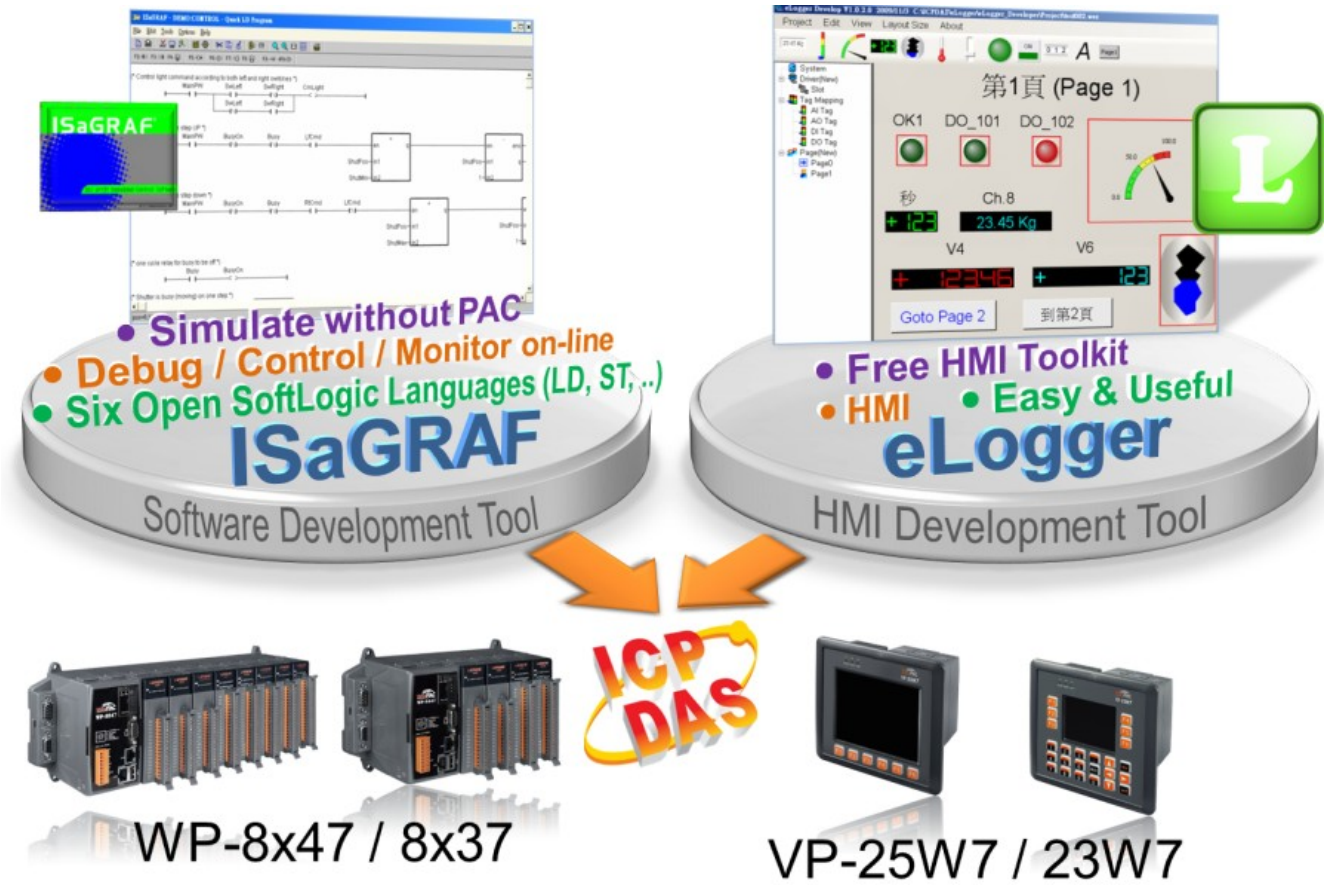
Version of this paper is 1.01 released on Nov.25, 2009. by chun@icpdas.com

The "eLogger" is a HMI development tools developed by ICP DAS. It features with easy use and friendly interface. The eLogger HMI can run with ISaGRAF SoftLogic in the WP-8xx7, VP-2xW7 and XP-8xx7-CE6 (will be available).

The WP-8147/8447/8847, WP-8137/8437/8837 and VP-25W7/23W7 support eLogger in the following driver version.
WP-8xx7: driver version 1.16 or later ver. VP-25W7/23W7: driver version 1.07 or later ver.

Download the latest PAC driver at http://www.icpdas.com/products/PAC/i-8000/isagraf-link.htm

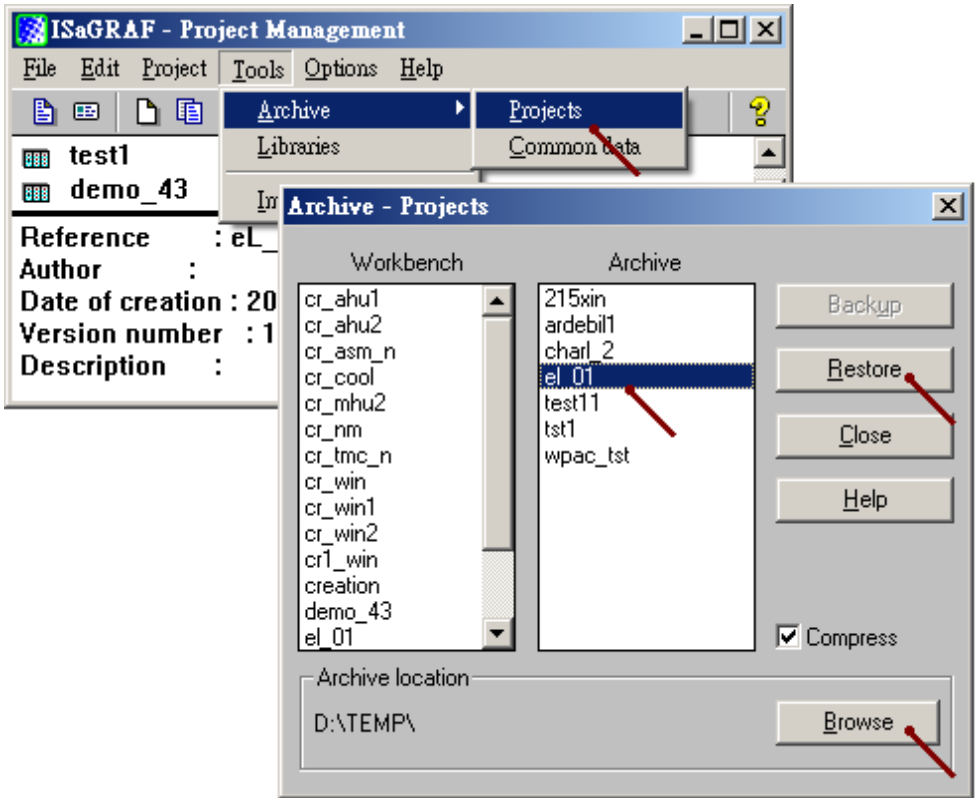
This paper may be modified in the future to include more functions provided by the newer eLogger version. User may visit the below web site to get the latest document and example programs.
Www.icpdas.com > FAQ > Software > ISaGRAF > FAQ-115 .



Reference: http://www.icpdas.com/products/PAC/i-8000/getting_started_manual.htm
ISaGRAF User's Manual
WinPAC ISaGRAF PAC Getting Started
VP-2xW7 Getting Started

1.1: Building A ISaGRAF SoftLogic Project

This sample project name is “eL_01.pia” which can be downloaded at www.icpdas.com > FAQ > Software > ISaGRAF > FAQ-115 . If user is familiar with the ISaGRAF programming, please restore this “eL_01.pia” to the PC / ISaGRAF first. Then download it to the WP-8xx7 PAC. Then go to the section 1.2 to learn more about building eLogger HMI pictures.



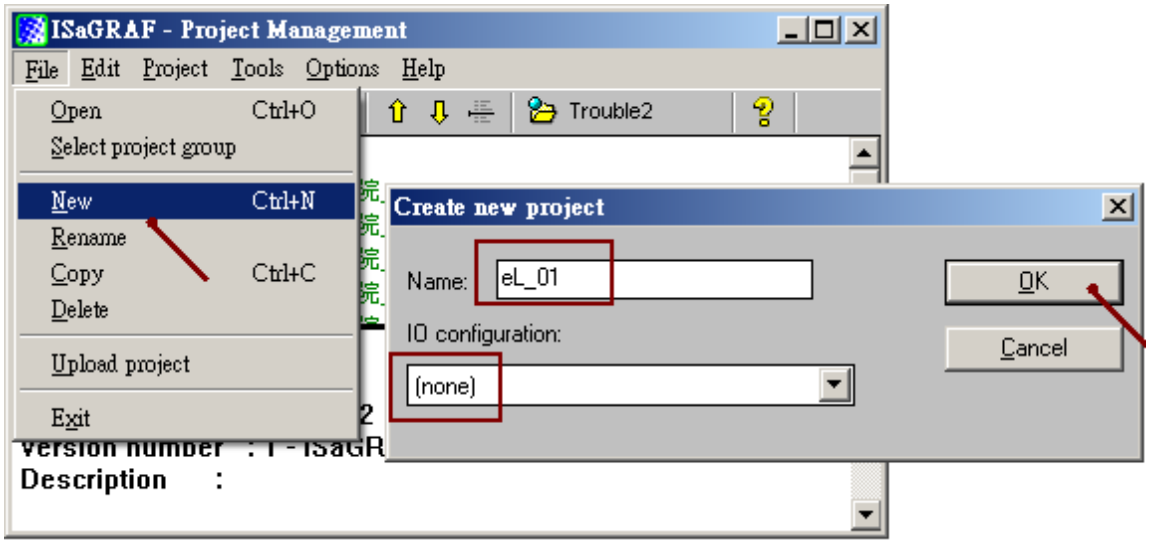
If user is not familiar with ISaGRAF, then please follow steps listed in the section 1.1 .
(New beginner can refer to the Chapter 2 of the “ISaGRAF User's Manual” . The PDF file name are “user_manual_i_8xx7.pdf” and “user_manual_i_8xx7_appendix.pdf” burned in the below CD-ROM.
WP-8xx7 CD-ROM: \Napdos\isagraf\wp-8xx7\english_manu\
VP-2xW7 CD-ROM: \Napdos\isagraf\vp-25w7-23w7\english_manu\
or download at http://www.icpdas.com/products/PAC/i-8000/getting_started_manual.htm)

If your PC hasn't installed the ISaGRAF software, then install it first. (You need to purchase at least one pcs. of ISaGRAF-256 or ISaGRAF-32 software. Please refer to the “Ordering Information” at the bottom of http://www.icpdas.com/products/PAC/i-8000/isagraf.htm)

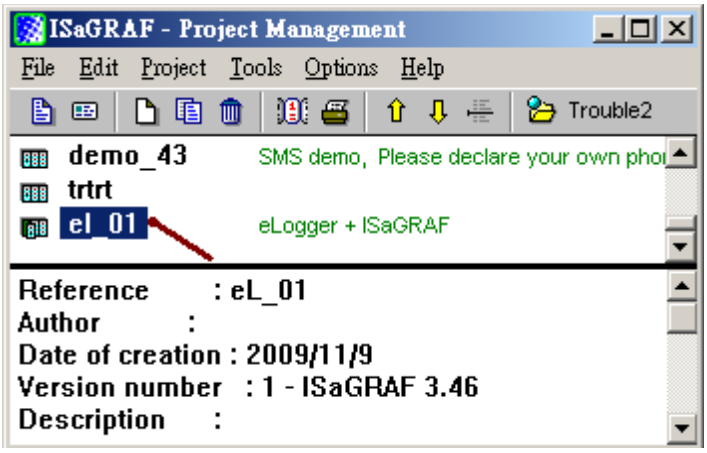
Then follow steps listed in the section 1.2 of the “ISaGRAF User's Manual” to install the “ICP DAS utilities For ISaGRAF”.

1.1.1: Creating A New Project

Run ISaGRAF project software, then create a new project. Name it as “eL_01” .



Then double click the project name to get into it.

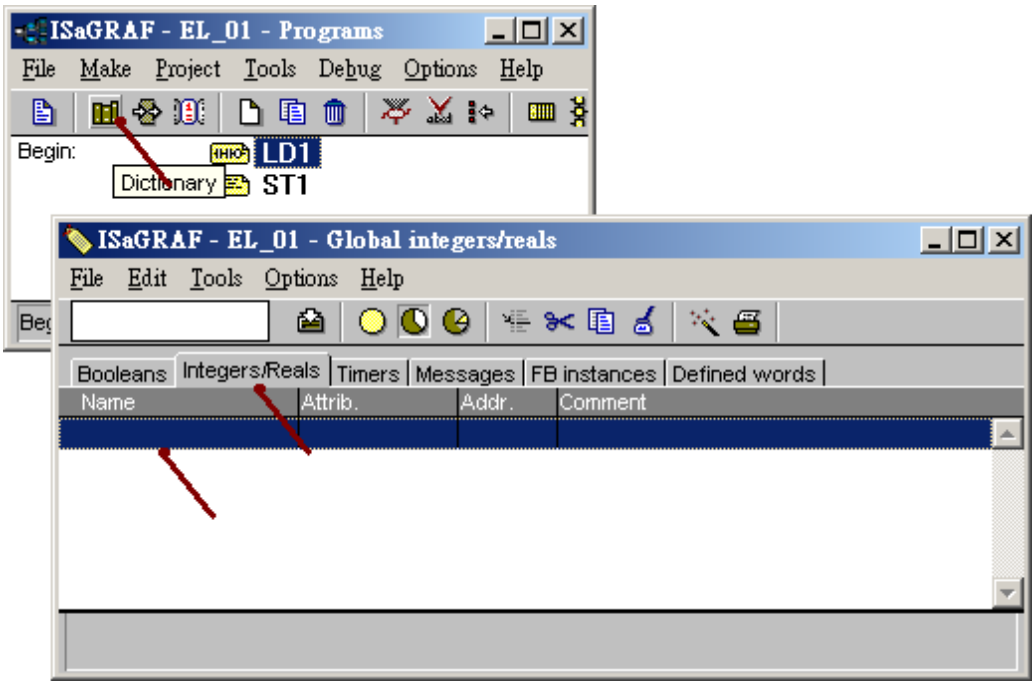


1.1.2: Declaring The ISaGRAF Project Variables

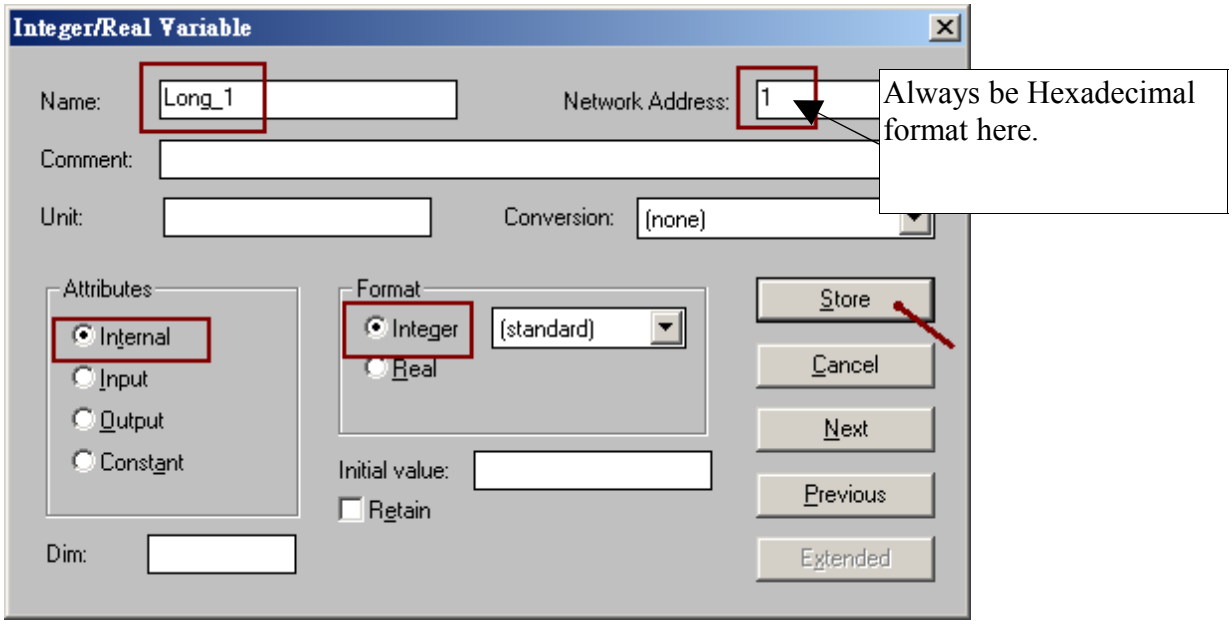
This example will use the following ISaGRAF variables

Name	Type	Attribution	NetWork Addr.	Description
Long_1	Integer	Internal	1	To access to eLogger 's 32-bit Long Tag
Word_3	Integer	Internal	3	To access to eLogger 's 16-bit Integer Tag
Word_4	Integer	Internal	4	To access to eLogger 's 16-bit Integer Tag
Float_5	Real	Internal	5	To access to eLogger 's 32-bit Float Tag
OUT_101	Boolean	**Output**	101	Link to Ch.1 D/O of slot 1: I-87055W
OUT_102	Boolean	**Output**	102	Link to Ch.2 D/O of slot 1: I-87055W
M1	Boolean	Internal	0	
DIR	Boolean	Internal	0	Declared with an initial value “True”

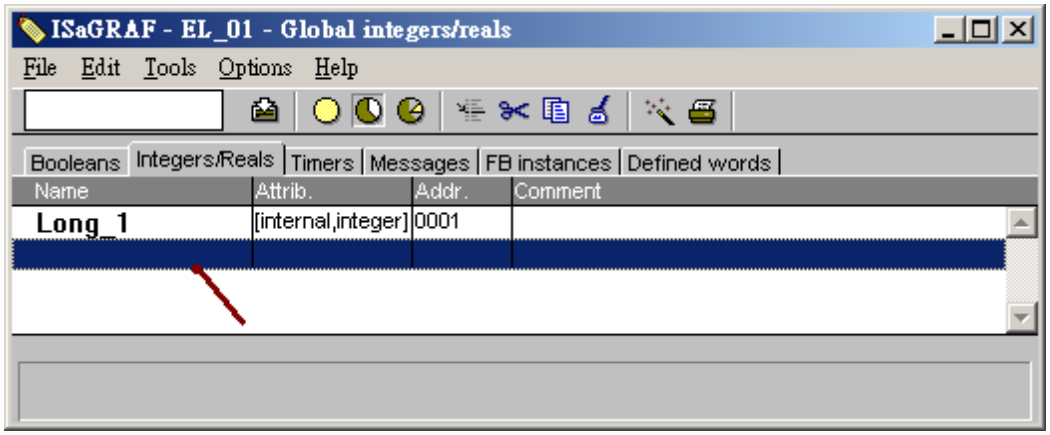
Get into the “Dictionary” , then click “Integers/Reals” , then double click the blue area at below.



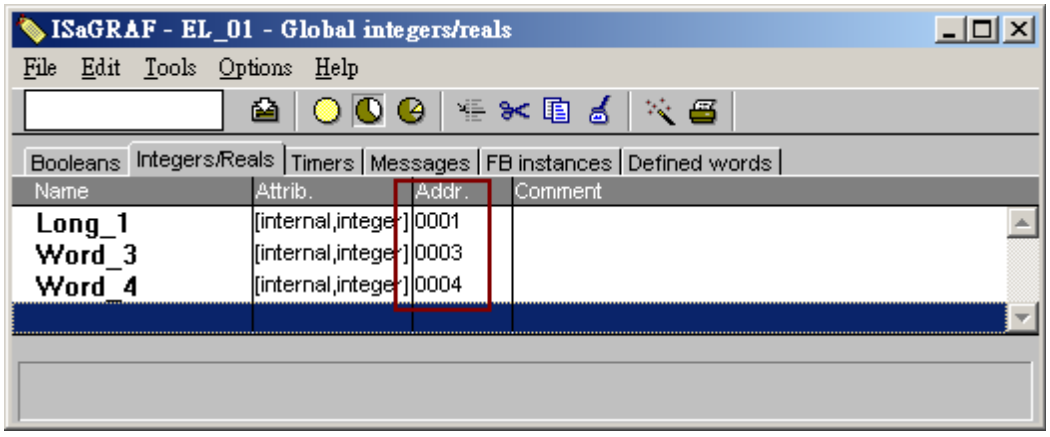
Name it as “Long_1”, set Network Address as 1



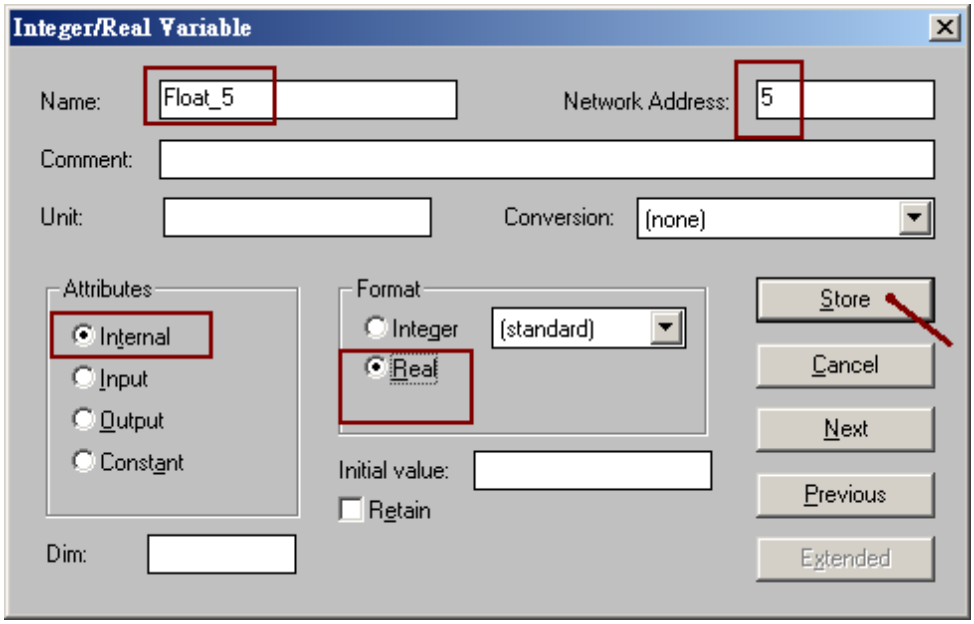
Then click “Store” . Then declare “Word_3” and “Word_4” by the same way. Set their NetWork Address as 3 and 4 respectively.



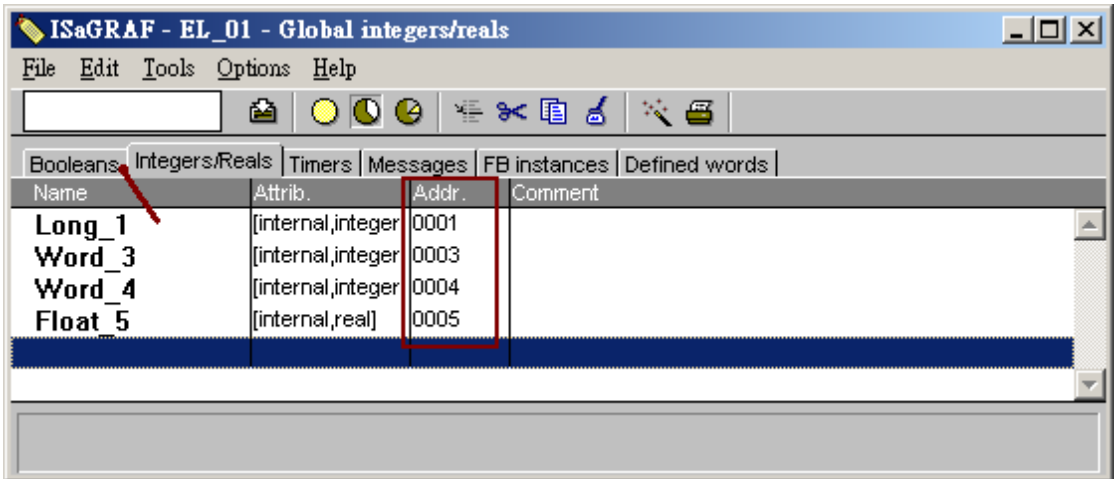
Then we got the following picture.



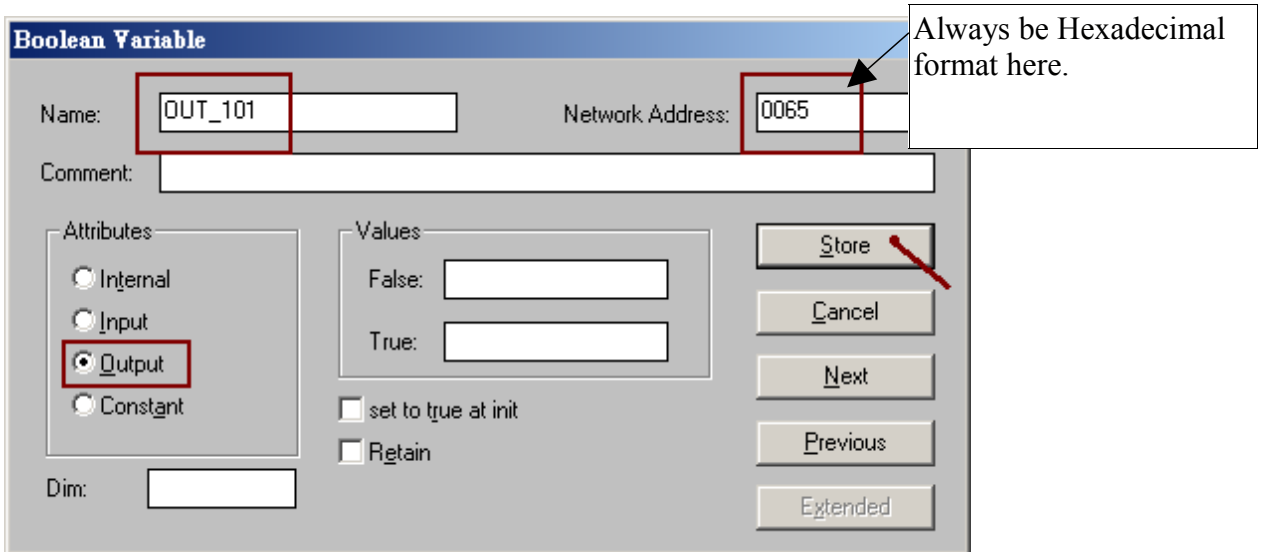
By the similar way to declare the “Float_5” . Set its format as “Real” and the Network Addr as 5.



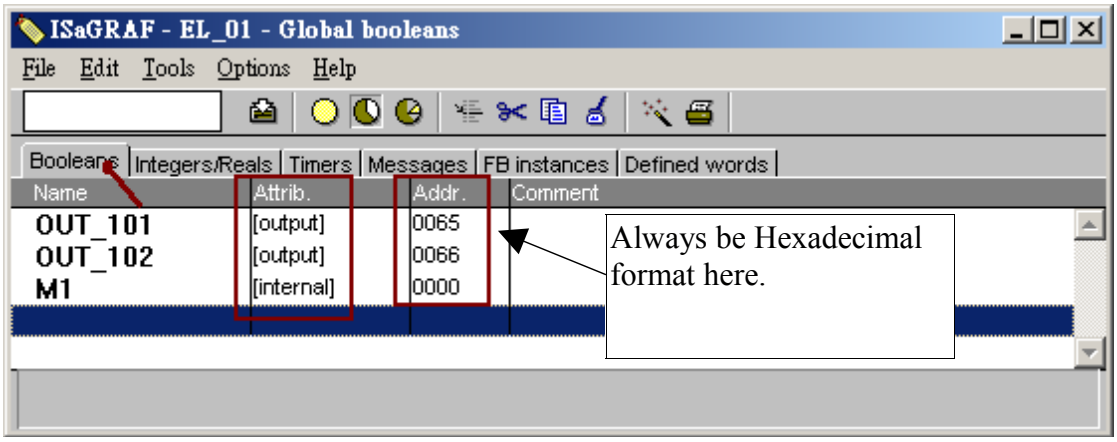
Then we got the following picture. To declare Boolean variables, please click “Booleans” first.



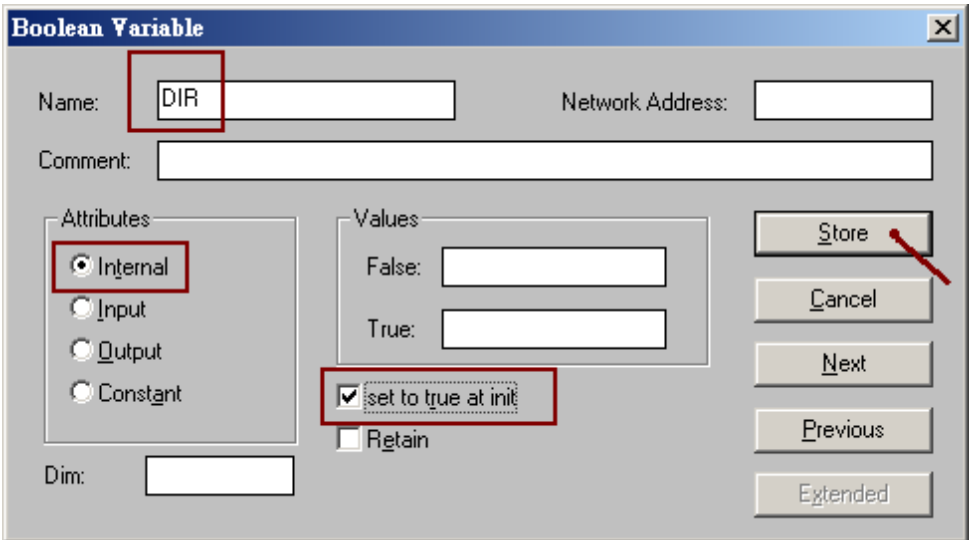
By the similar way as former to declare “OUT_101” and “OUT_102” and “M1” . Set the attribution of “OUT_101” and “OUT_102” as Output. Set their Network address as 101 and 102 respectively. While set M1's attribution as Internal .



Then we got the following picture.

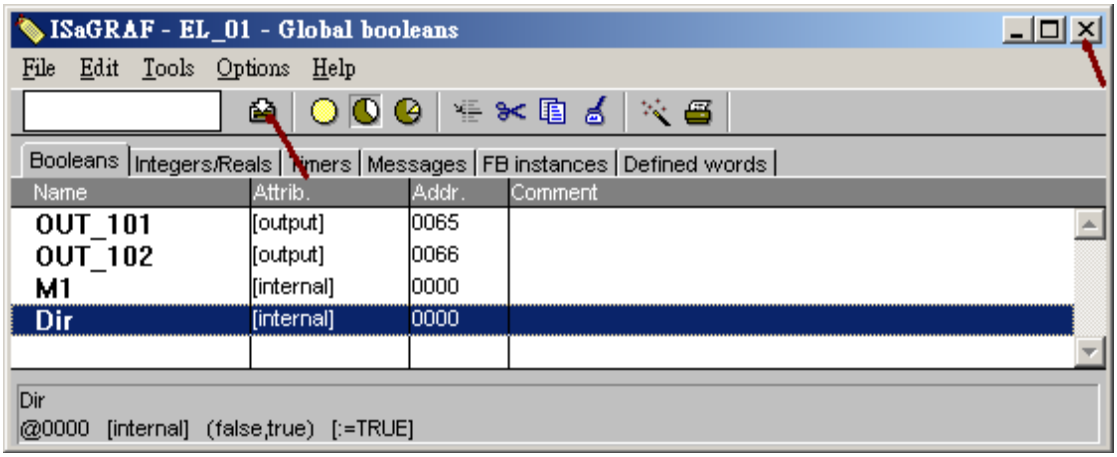


Declare one more boolean variable “DIR” as below. Check the “set to true at init” .

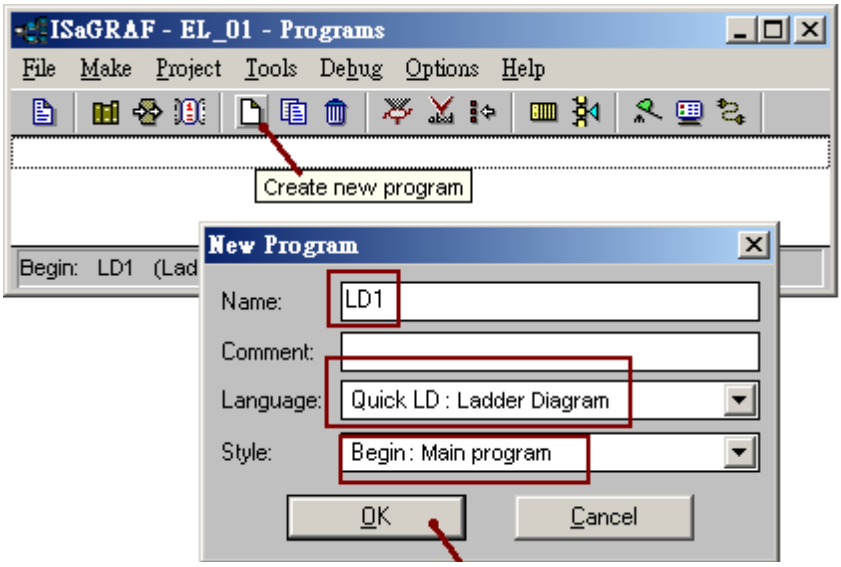


1.1.3: Creating the Sample Ladder Diagram Program - LD1

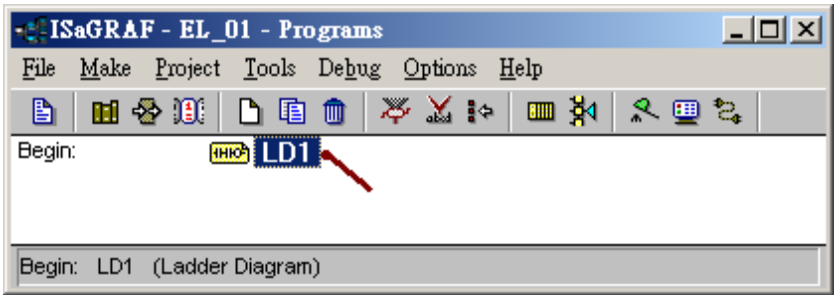
After all variable declaration is finished, click “save” to save it and then exit the following window.



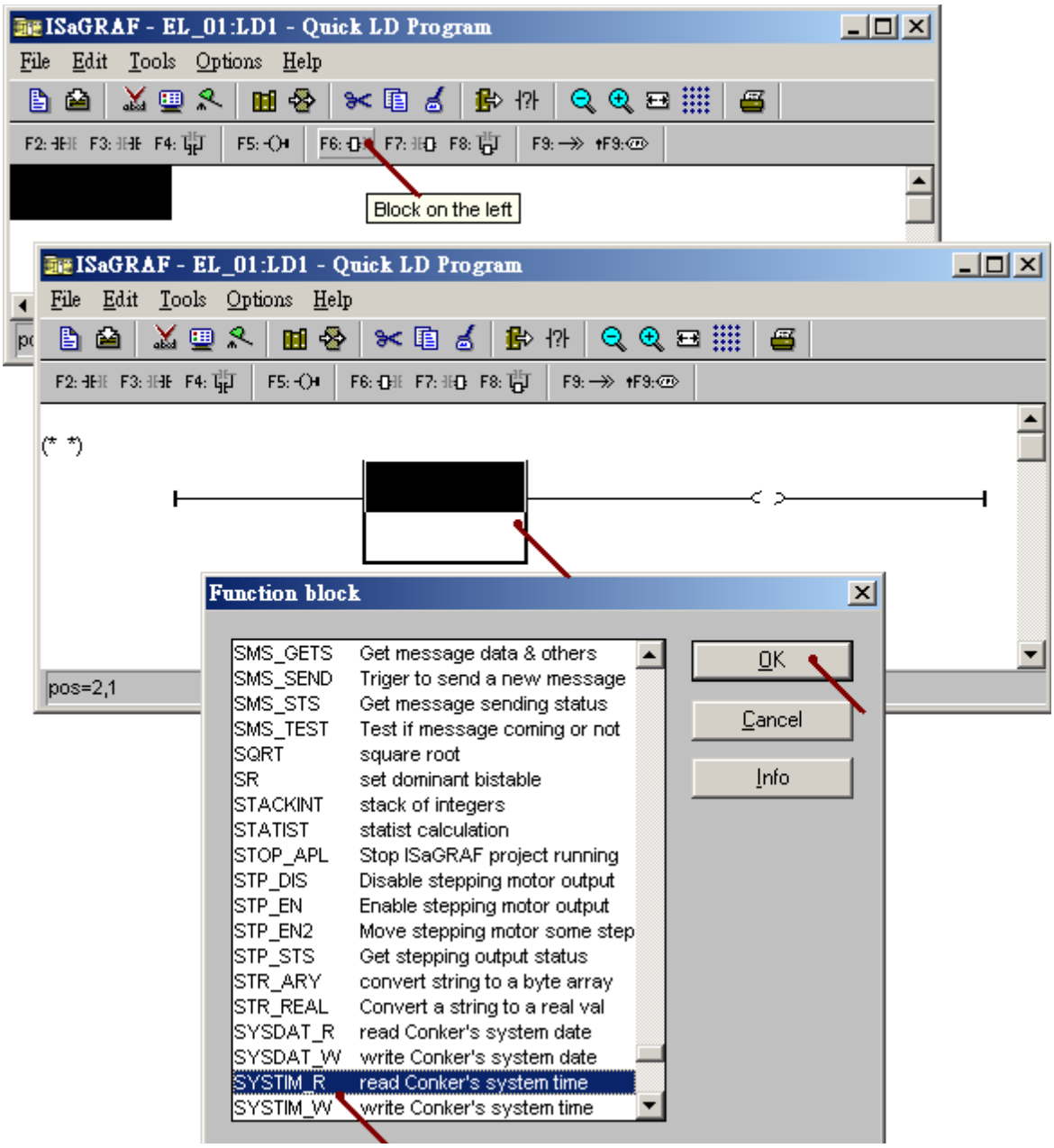
Then create a new program as the following picture. Name it as “LD1” , select its language as “Quick LD : Ladder Diagram” . Set the “Style” as “Begin: Main program” .



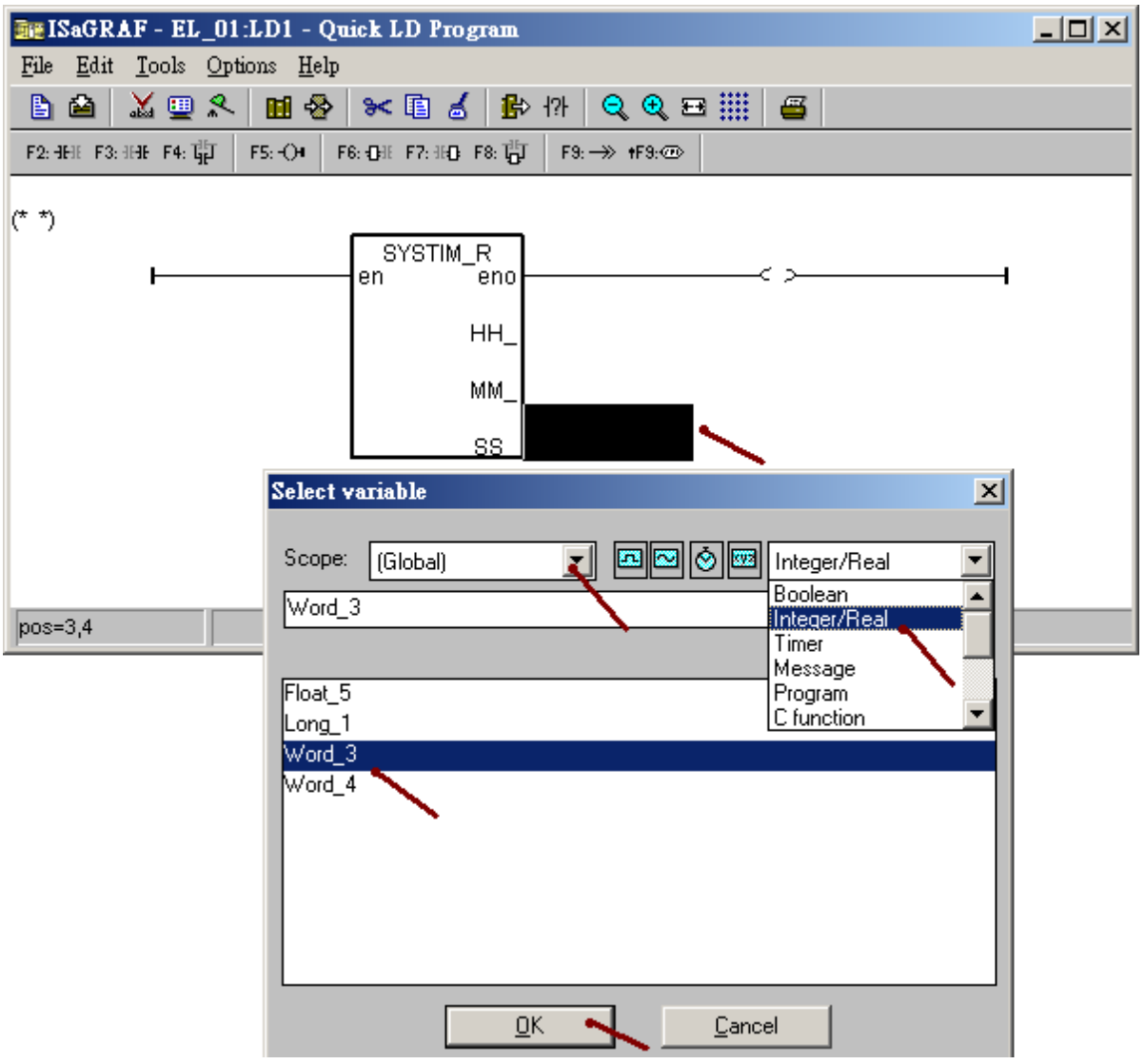
Then we got the following picture. Please double click the program name to get into it.

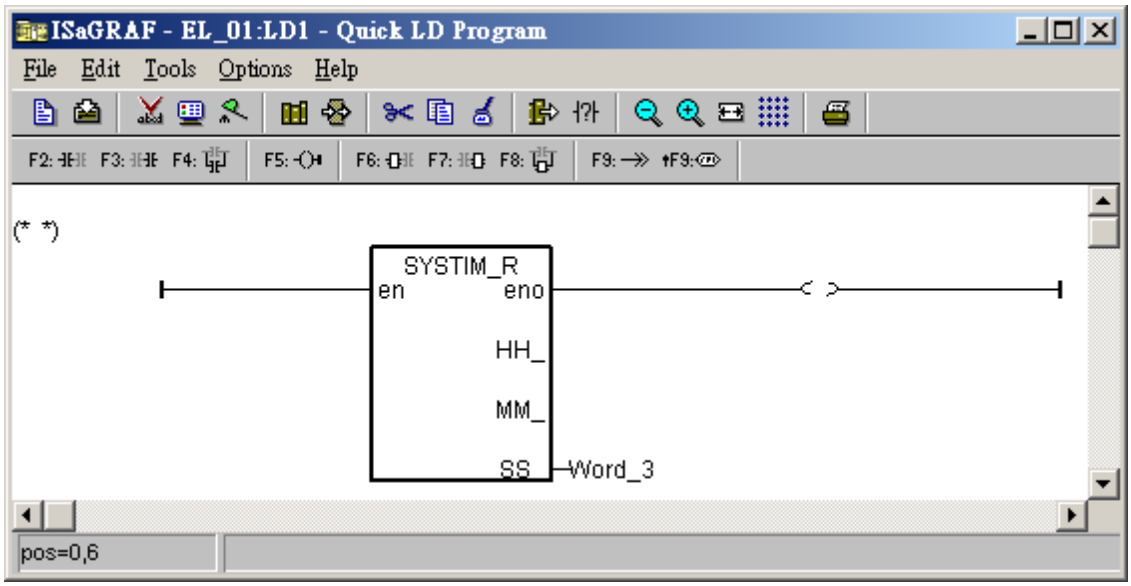


We are going to create a “SYSTIM_R” function block to get the PAC 's Hour/Minute/Second . Click “Block on the left” , then double click inside the block to select the “SYSTIM_R” .

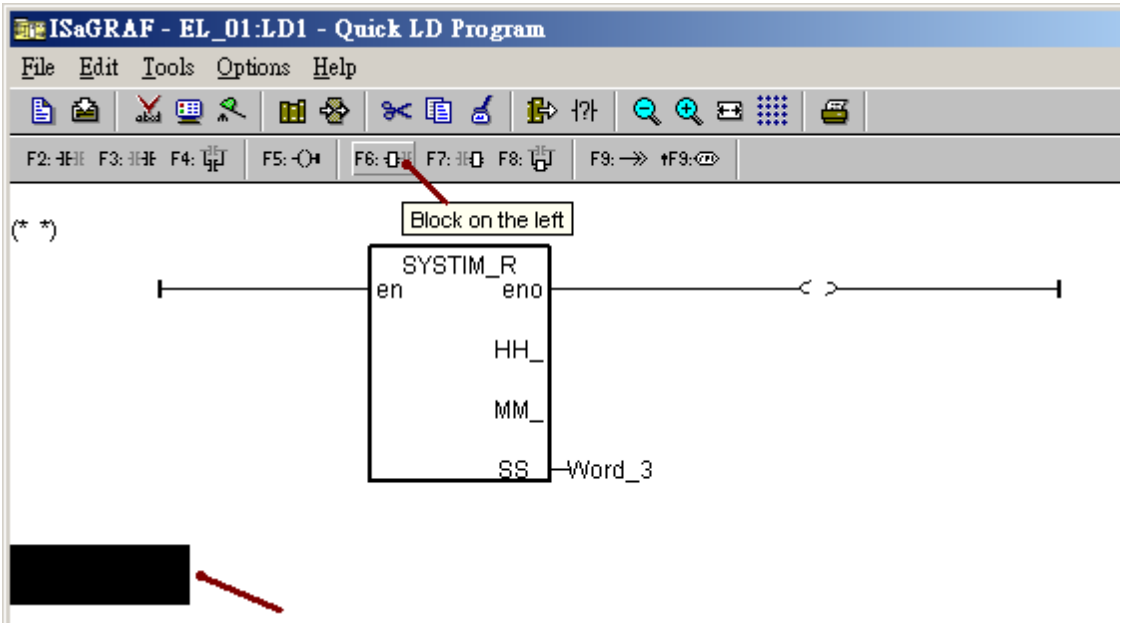


Double click on the right hand side of the “SS_” parameter to connect the “Word_3” on it.

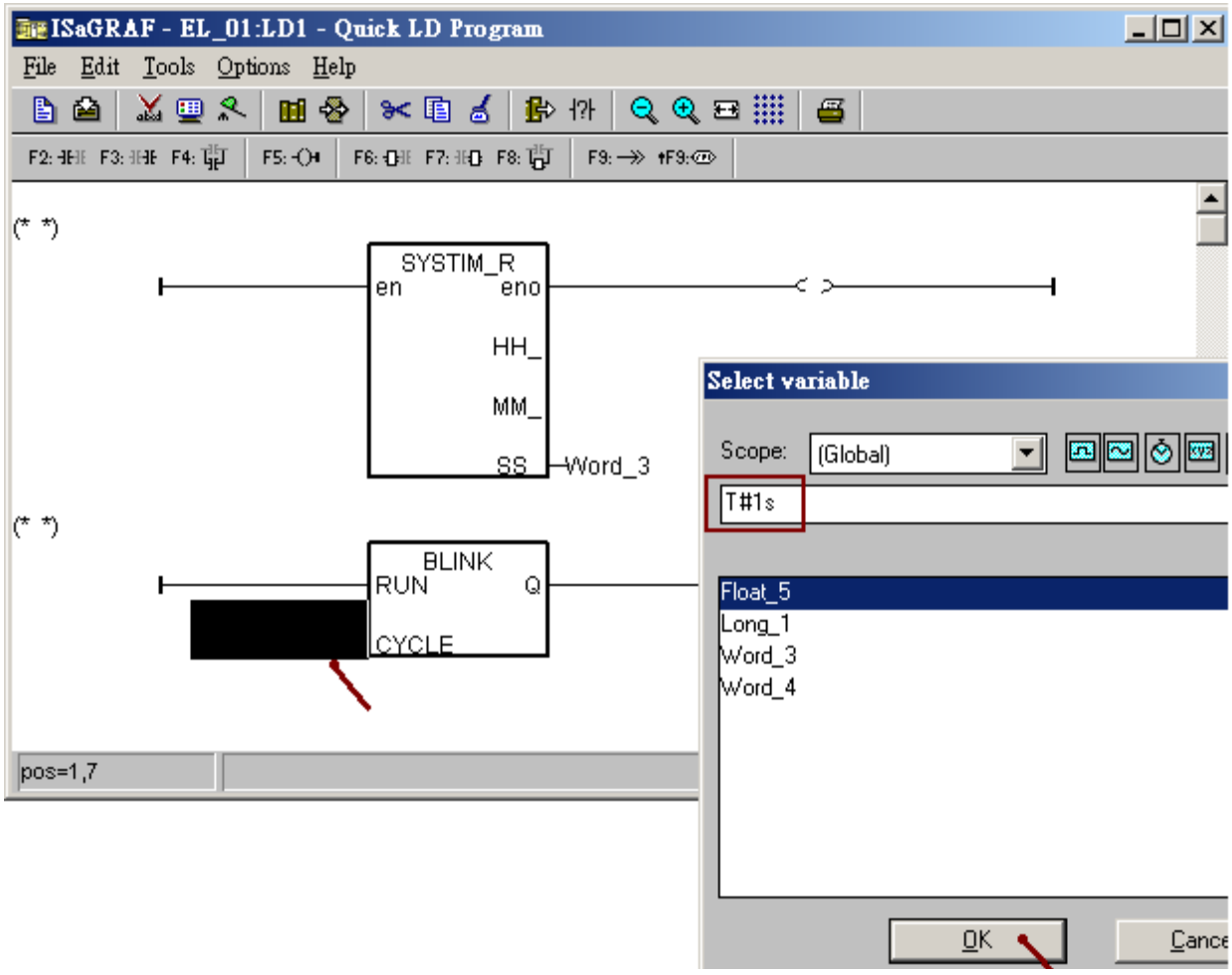




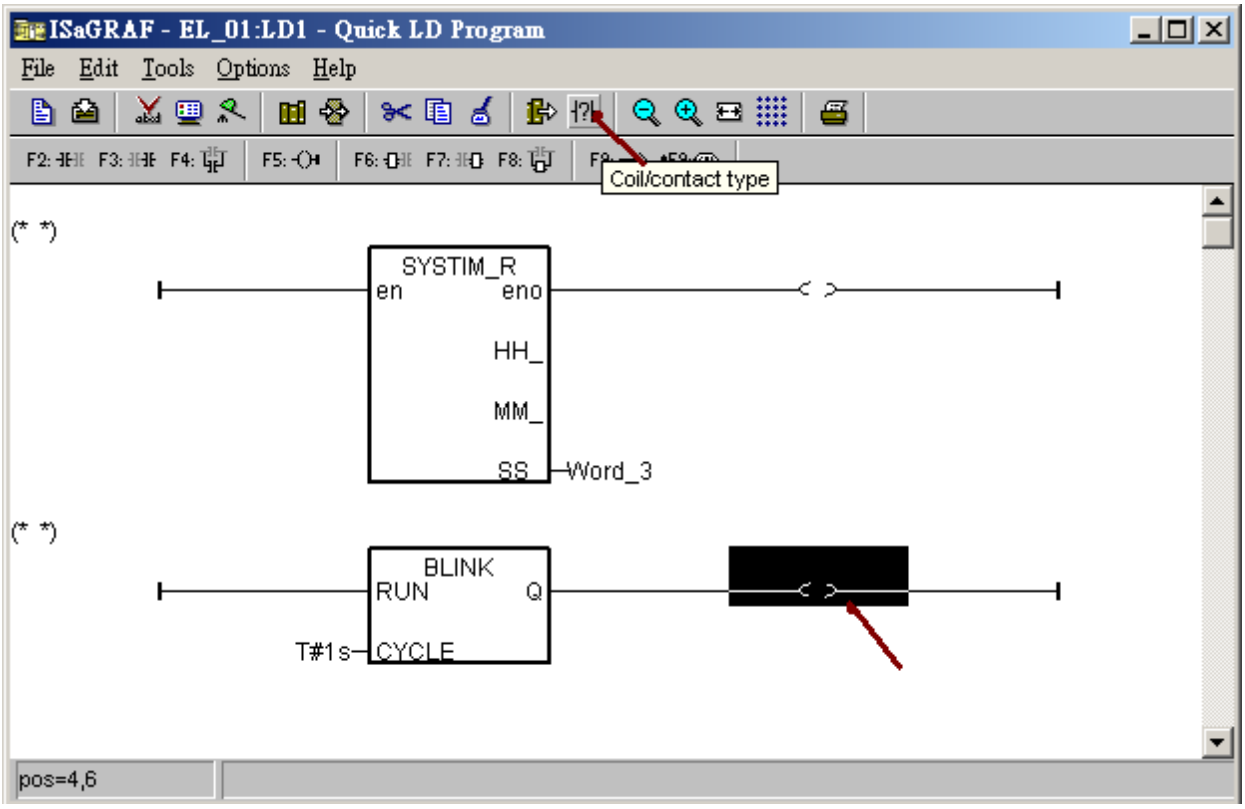
We are going to generate a pulse True every second to the boolean variable “M1” . Move the cursor to the below position under the block first.



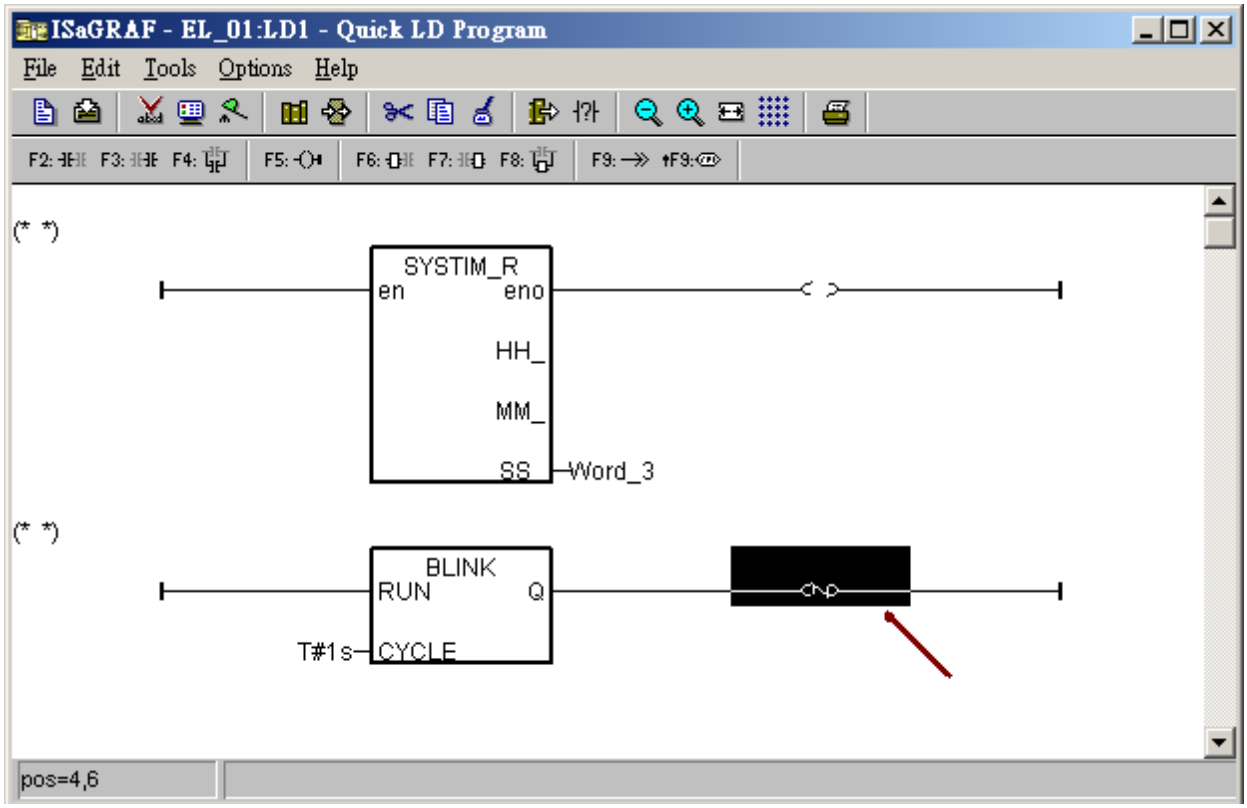
Then double click on the left hand side of the “CYCLE” parameter to key-in “T#1s” .

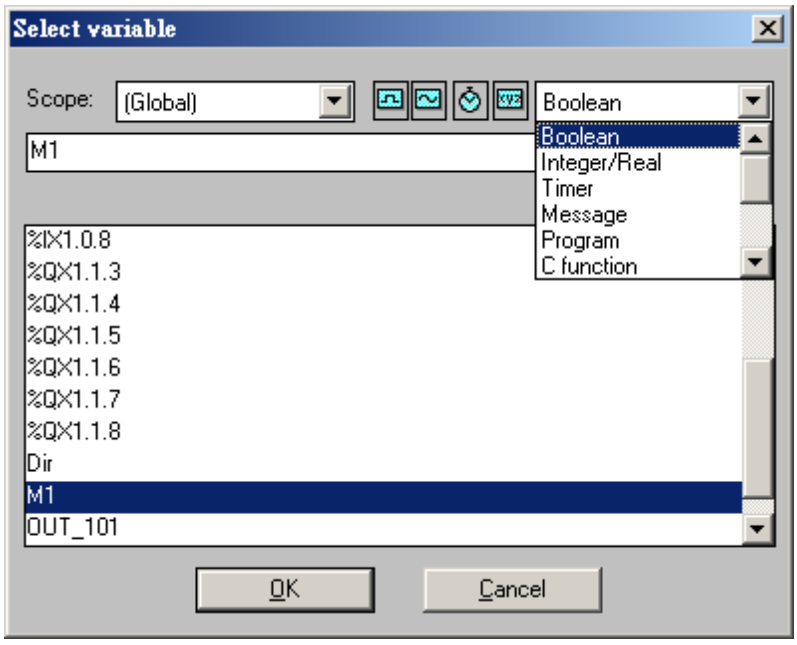


Then move the cursor to the coil at right hand side of the block. Click “Coil / Contact type” several time to select the “N” coil.

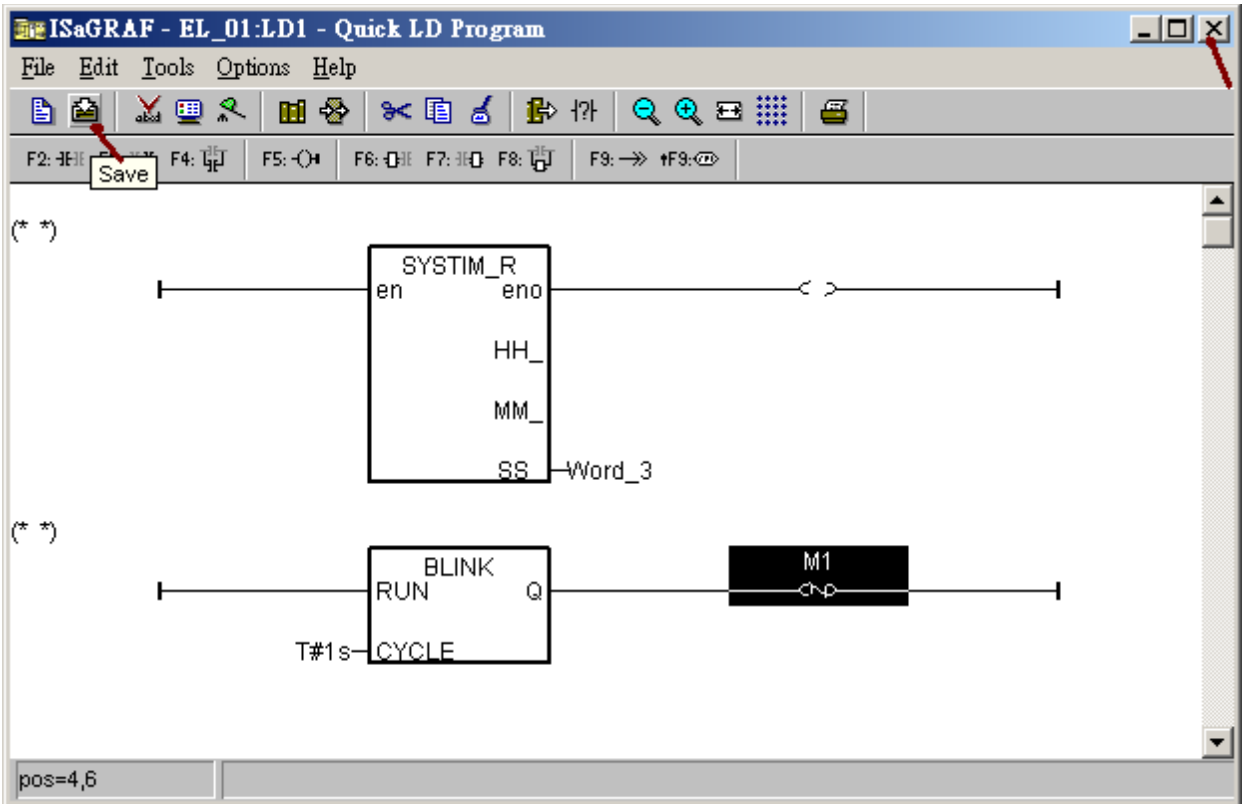


Then double click on the “N” coil to connect the boolean variable “M1” on it.



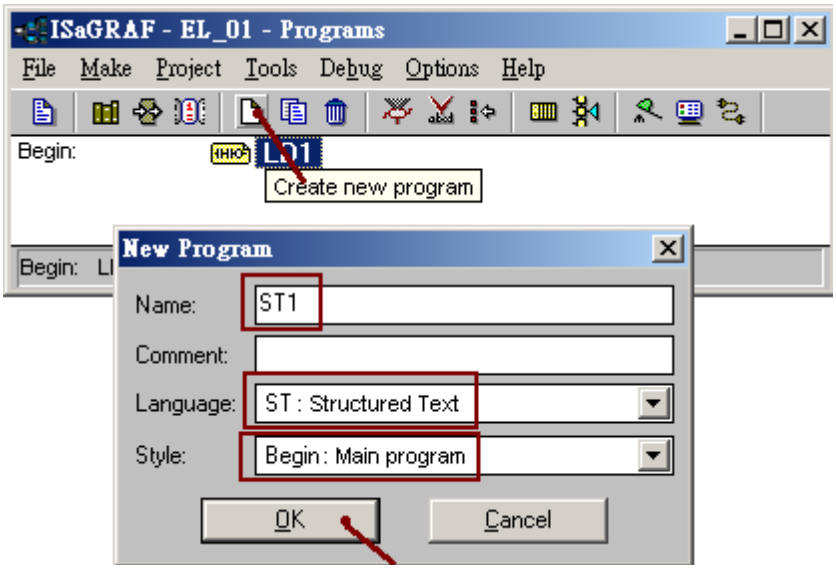


Then we have finished the “LD1” program. Click “save” and exit the window.

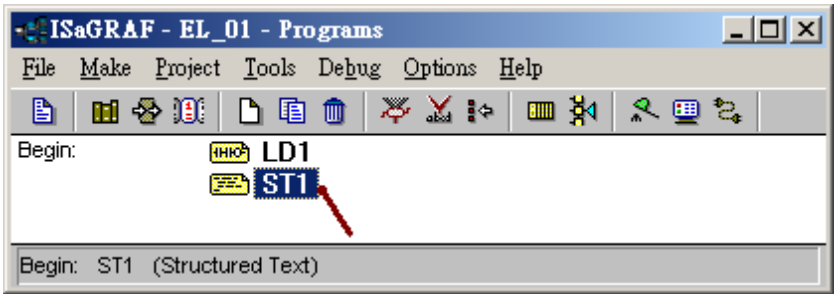


1.1.4: Creating a Structured Text program - ST1

Create a ST program “ST1” as the following picture.



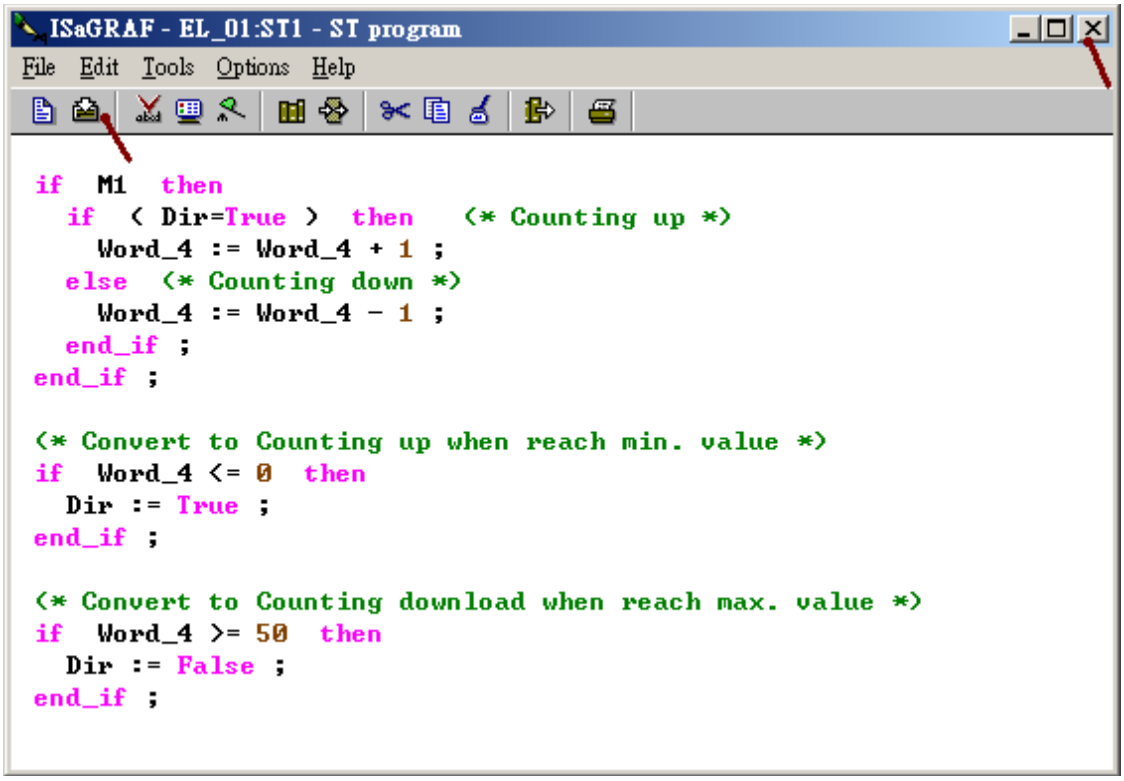
Double click the program name to get into it.



Edit the following code.

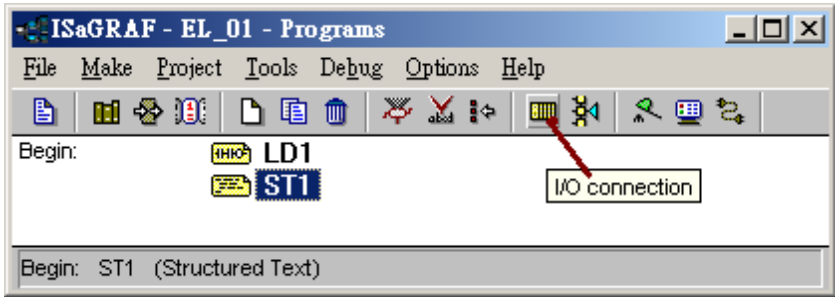
```
if  M1  then
  if  ( Dir=True )  then
    Word_4 := Word_4 + 1 ;    (* Counting up *)
  else
    Word_4 := Word_4 - 1 ;    (* Counting down *)
  end_if ;
end_if ;

if  Word_4 <= 0  then
  Dir := True ;             (* reach Min. value, change to counting up *)
end_if ;

if  Word_4 >= 50  then
  Dir := False ;            (* reach Max. value, change to counting down *)
end_if ;
```

Then we got the following picture. Click “save” and then exit it.

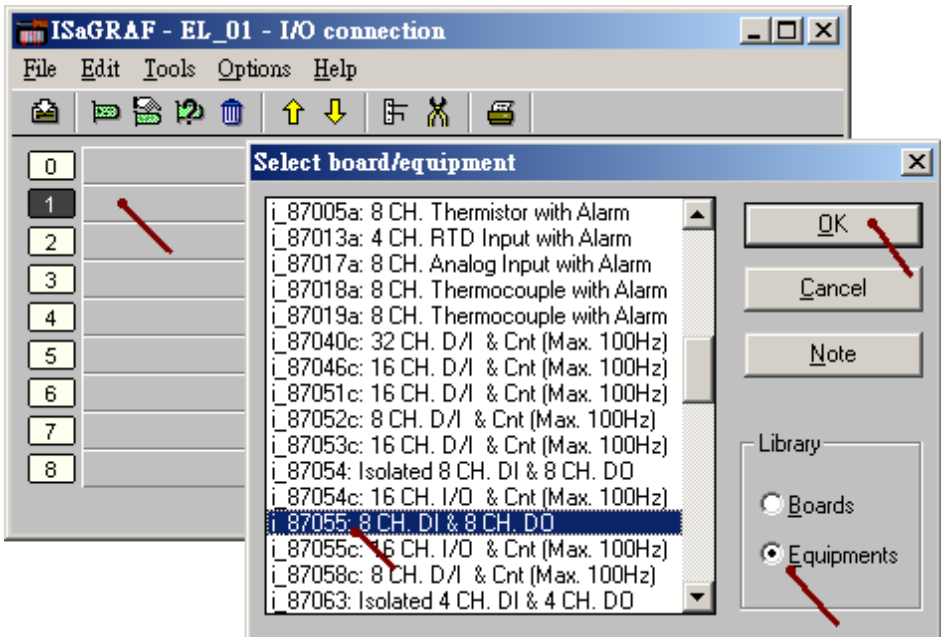


1.1.5: Connecting The I/O

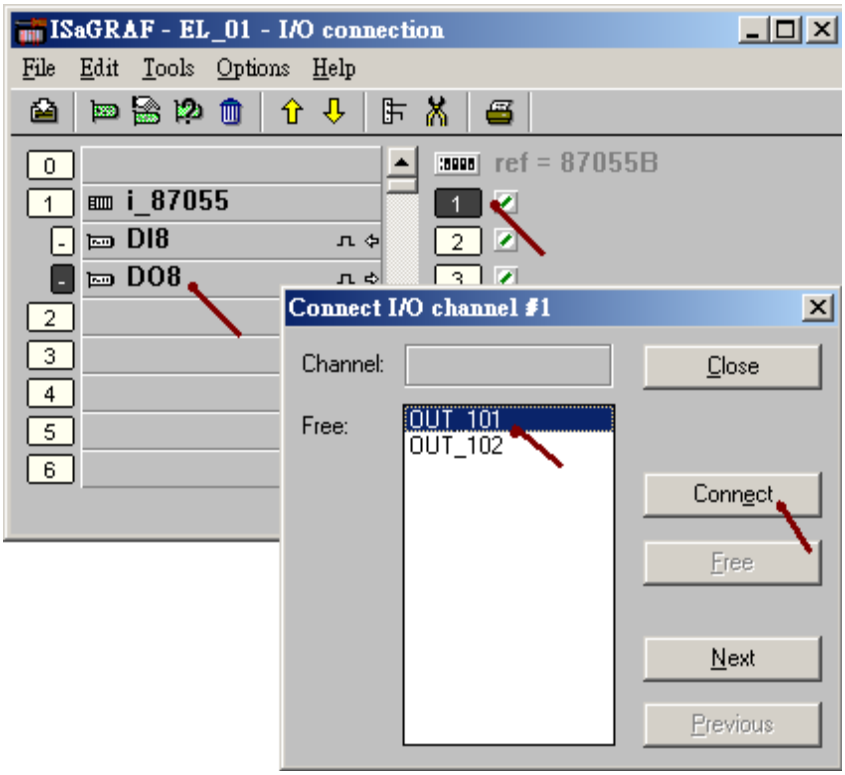
We have declared two boolean variables with attribution “Output”. Their name are “OUT_101” and “OUT_102” . To connect them to the physical I/O board (I-87055W in this example), please click the “ I/O connection” .



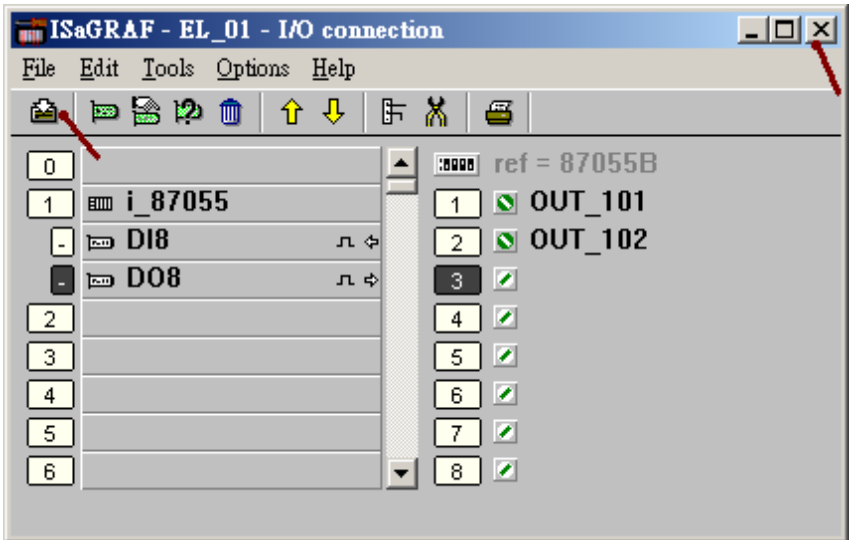
Double click the slot number “1” to insert the “i_87055” . (If find no “Equipments” > “i_87055”, it means the “ICP DAS Utilities For ISaGRAF” has not installed. Please refer to section 1.2 of the “ISaGRAF User's Manual”)



Then double click the channel number on the right hand side to connect “OUT_101” and “OUT_102” . (If find no “OUT_101” and “OUT_102”, it means their attribution were not declared as “Output” . Please refer to section 1.1.2 of this paper to declare their attribution as “Output”)

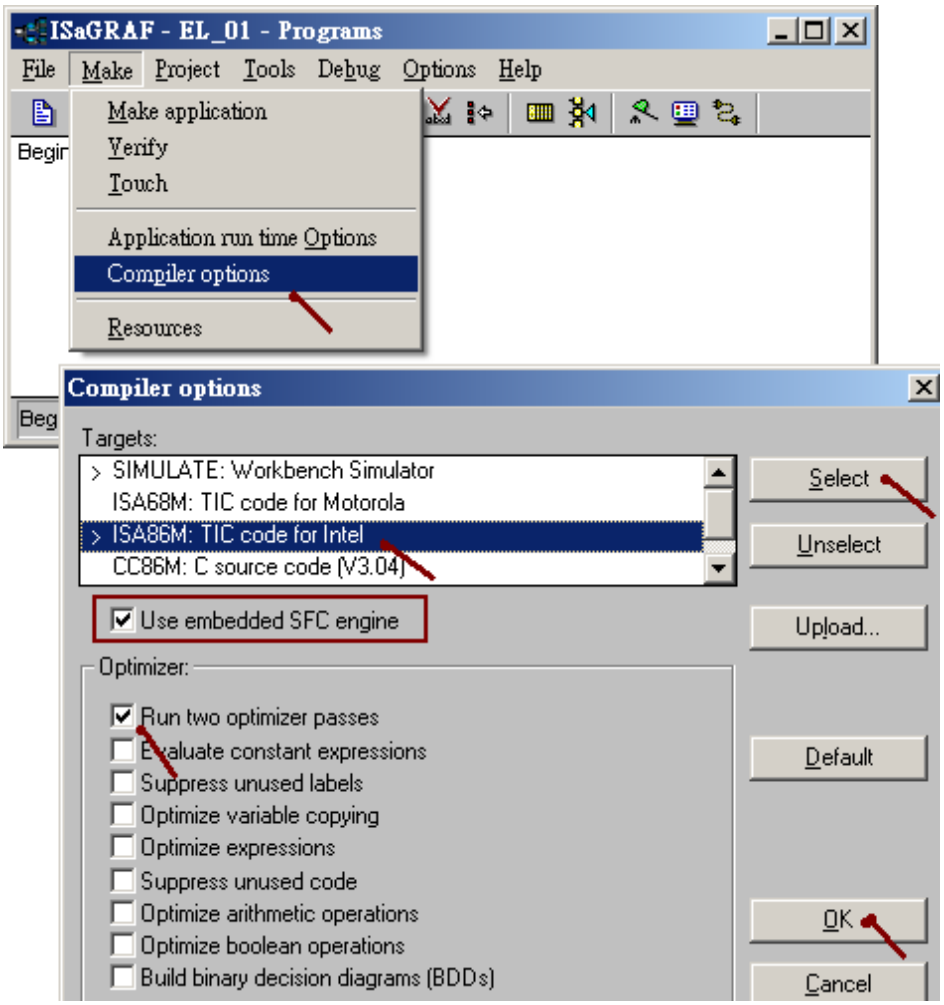


Then click “Save” and exit it.

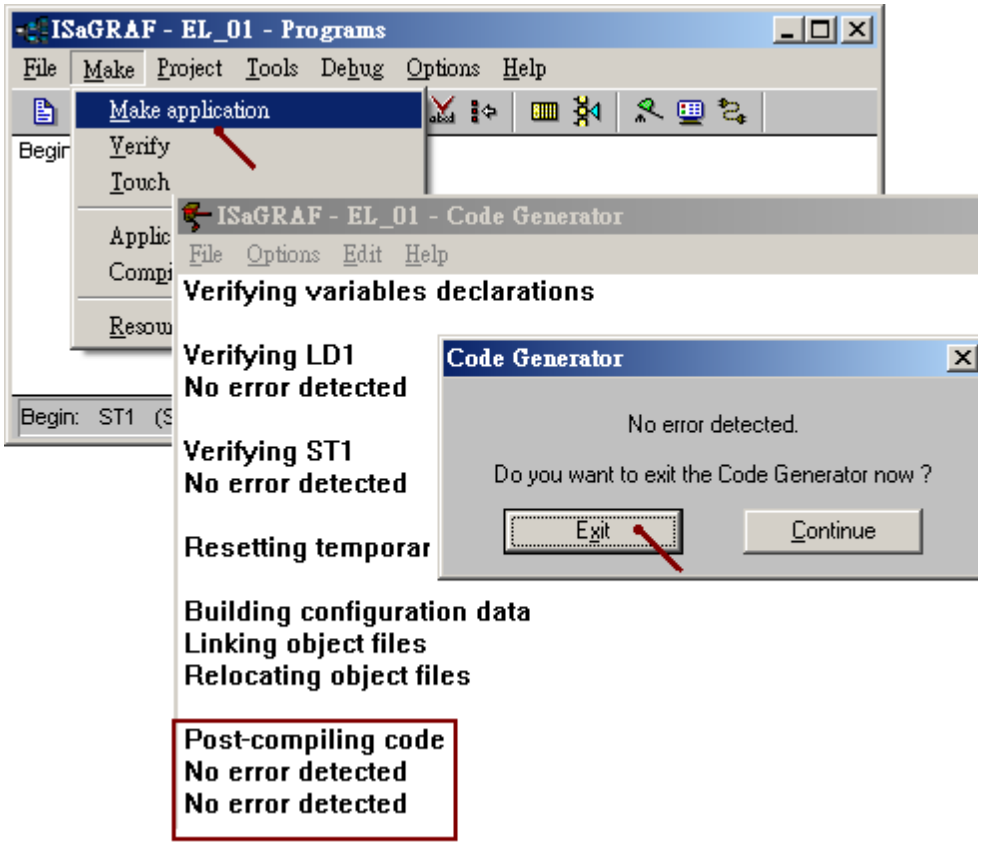


1.1.6: Compiling The ISaGRAF Project

Select the following Compiler options. Must check the “ISA86M: TIC code for Intel” and the “Use embedded SFC engine” and the first Optimizer “Run two optimizer passes” .



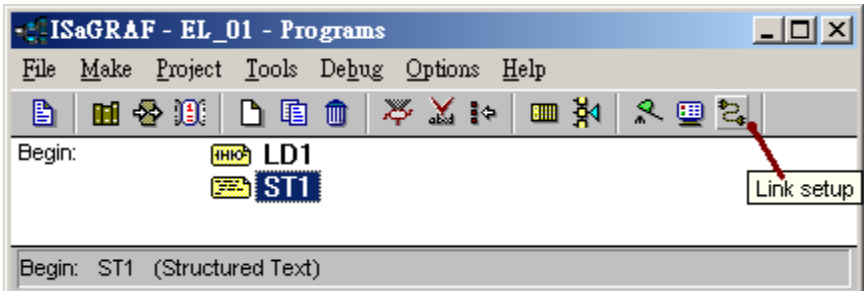
Then run “Make application” to compile the project.



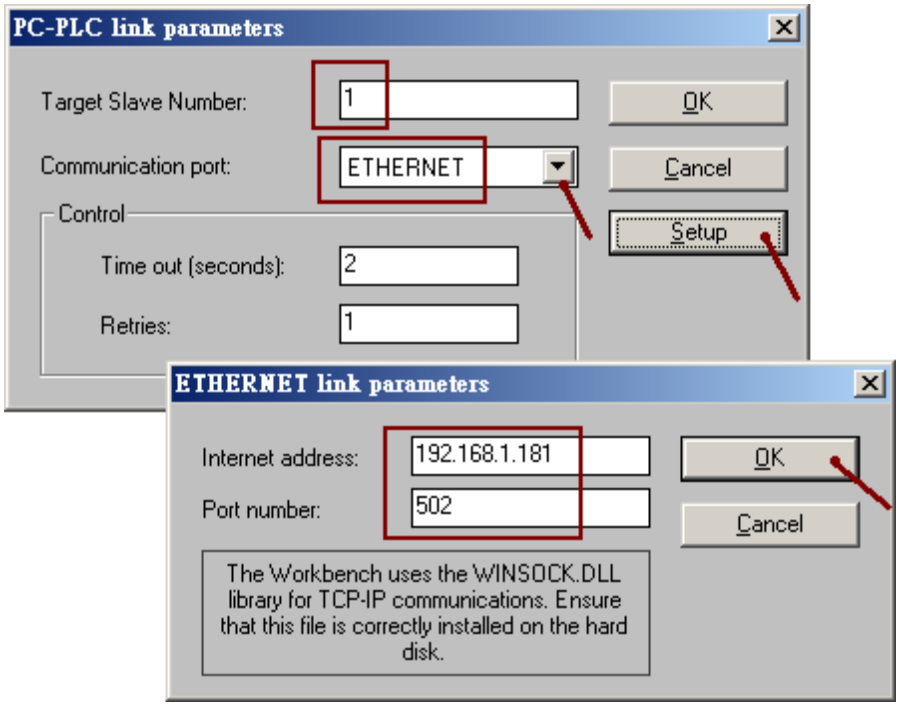
If no error detected, congratulations !

1.1.7: Download the ISaGRAF Project to the PAC

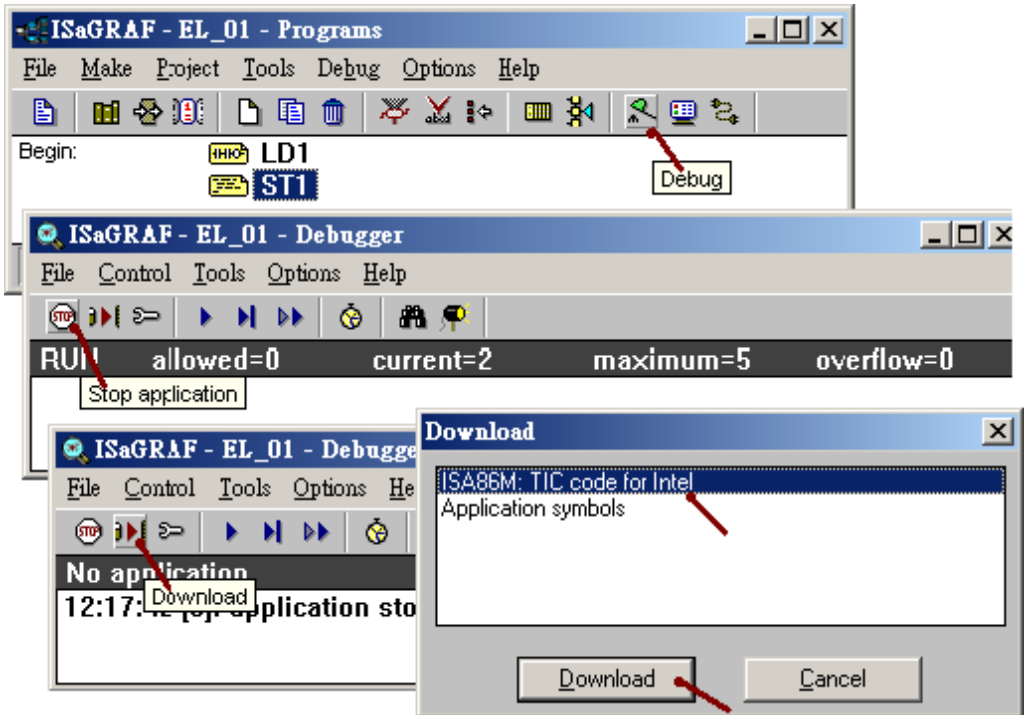
Click “Link Setup”.



To download project via Ethernet cable, please select “ETHERNET” and enter the IP address of the PAC and must set Port number as 502.



Then click “Debug” to connect to the PAC. If it is connected, click “Stop Application” first and then click “Download” to download the “TIC code for Intel” to the controller.



1.2: Build eLogger HMI Project

If there are problems when running the eLogger RunTime, refer to section 1.3 of this paper to fix it.

The “eLogger” includes “Developer” and “RunTime” . The eLogger Developer is the development toolkits installed in PC for building HMI pictures and items. The eLogger RunTime is the driver installed in the PAC (WP-8xx7 or VP-2xW7 or XP-8xx7-CE6). User need to upload his HMI pictures created by the eLogger Developer from PC to the PAC. Then the eLogger RunTime will execute it and display HMI pictures on the VGA Monitor of the controller.

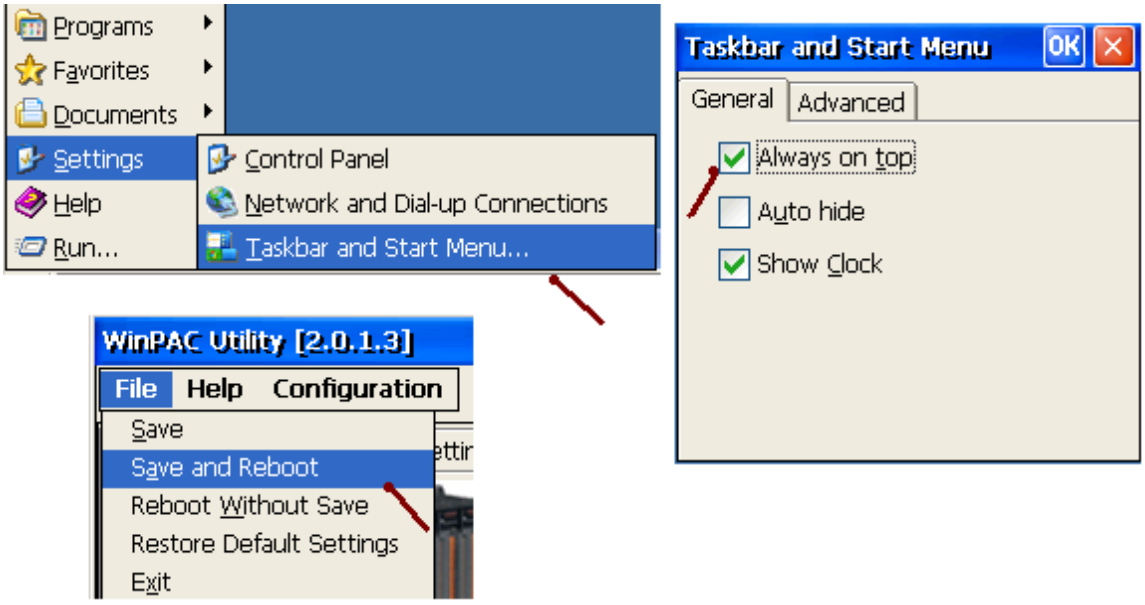
1.2.1: Install the eLogger Developer in PC and the eLogger RunTime in Controller

Please visit ftp://ftp.icpdas.com/pub/cd/winpac-8xx7/napdos/elogger/elogger_developer/ to download the latest eLogger Developer .Then install it in PC.

Then visit ftp://ftp.icpdas.com/pub/cd/winpac-8xx7/napdos/elogger/elogger_runtime/ to download the eLogger RunTime.

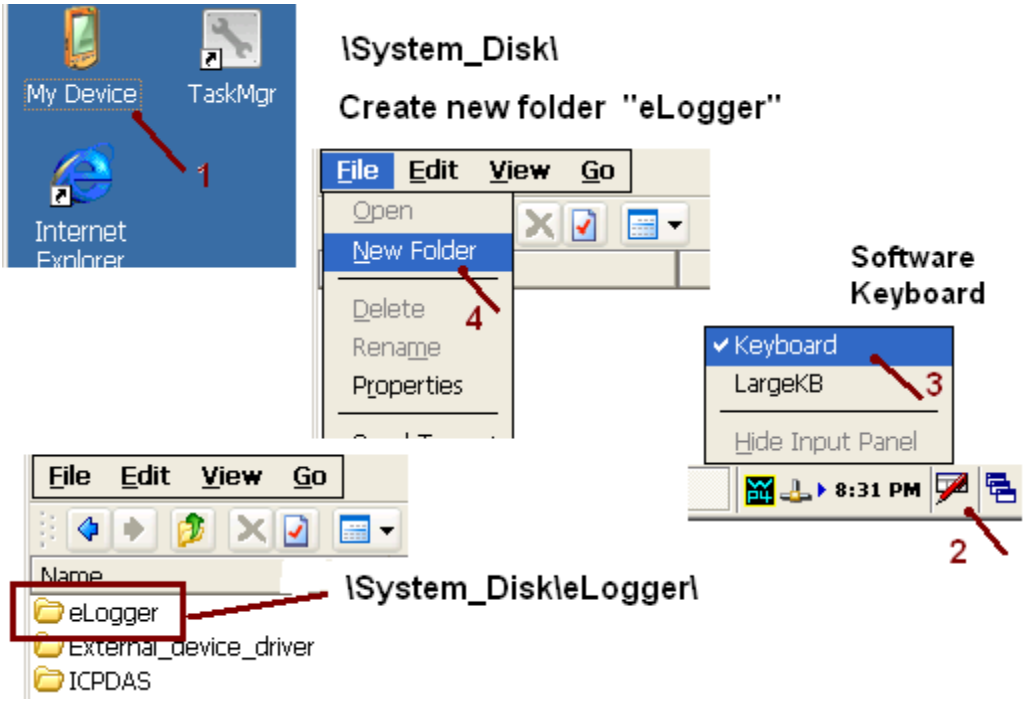
It is better to show the “Taskbar” in the PAC 's VGA monitor at design time. Please enable it as below.

Connect one VGA Monitor to the VGA port of the controller. And connect one USB mouse to the USB port of the controller. Then set the “Taskbar” as “Always on top”. **DO NOT check the “Auto hide”** . Then run WinPAC utility (or ViewPAC Utility) to save the setting.

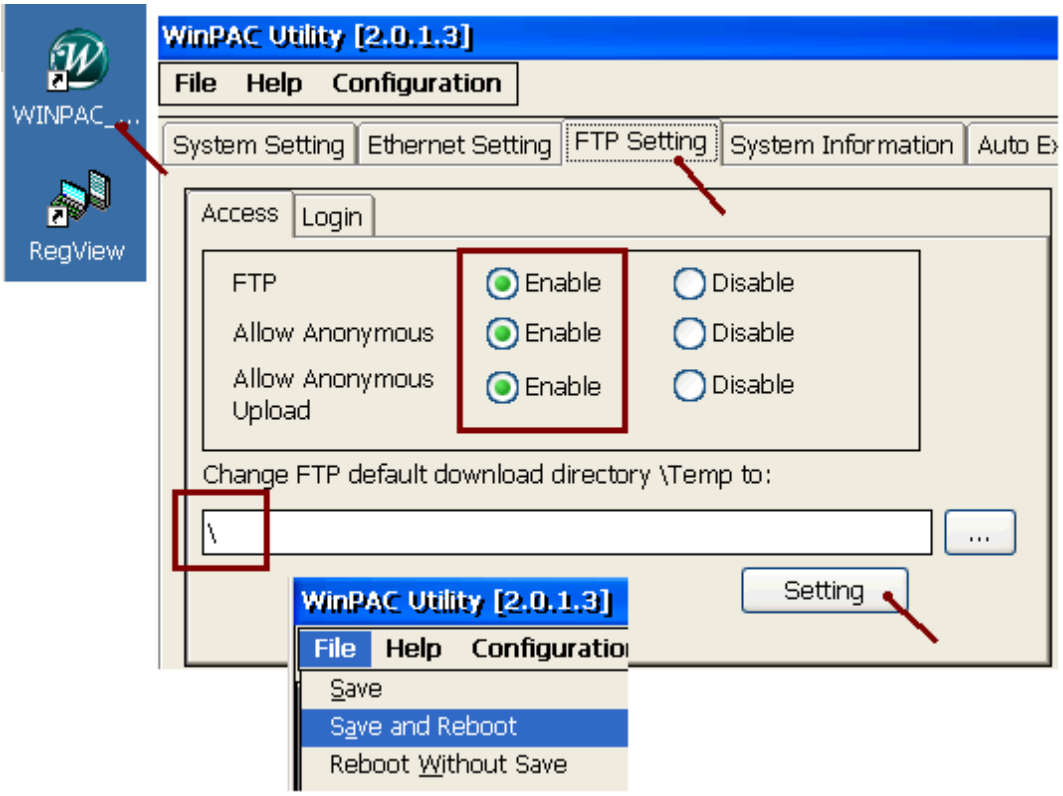


How to install eLogger RunTime:

Please click “My Device” on the controller's VGA. Then get into the “ \System_Disk\” path to create a new folder “eLogger” .

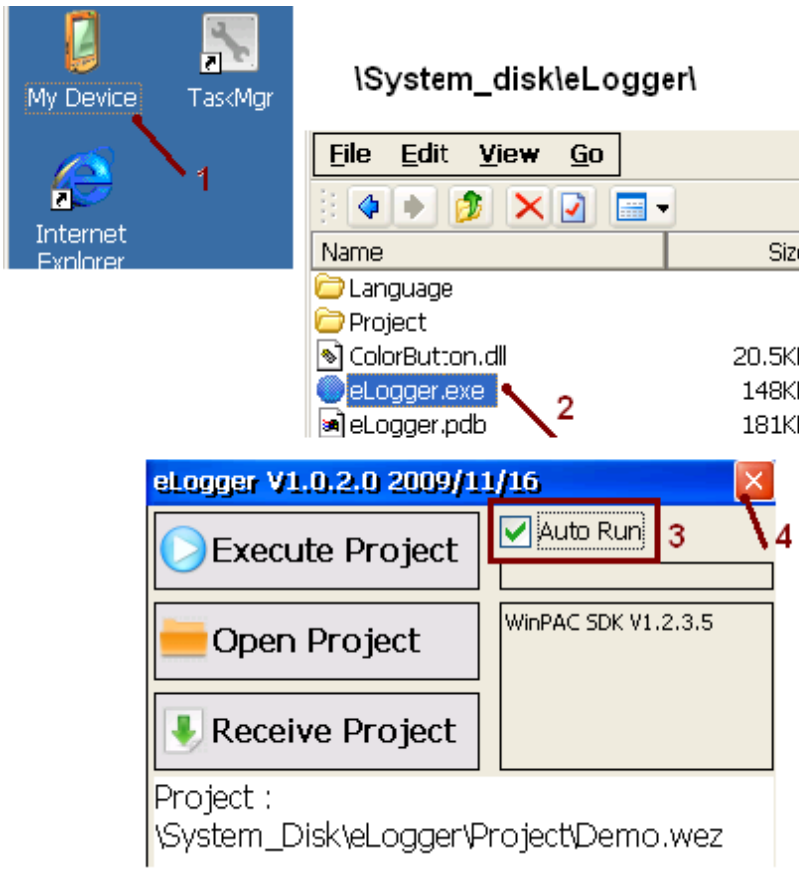


To copy all files of the eLogger RunTime to the “\System_Disk\eLogger\” path of the controller, it is better to use the ftp tools in PC . First make sure FTP is enabled in the controller and its path is set as “\” . Then click “Setting”, then run File > Save and Reboot to save the FTP setting.

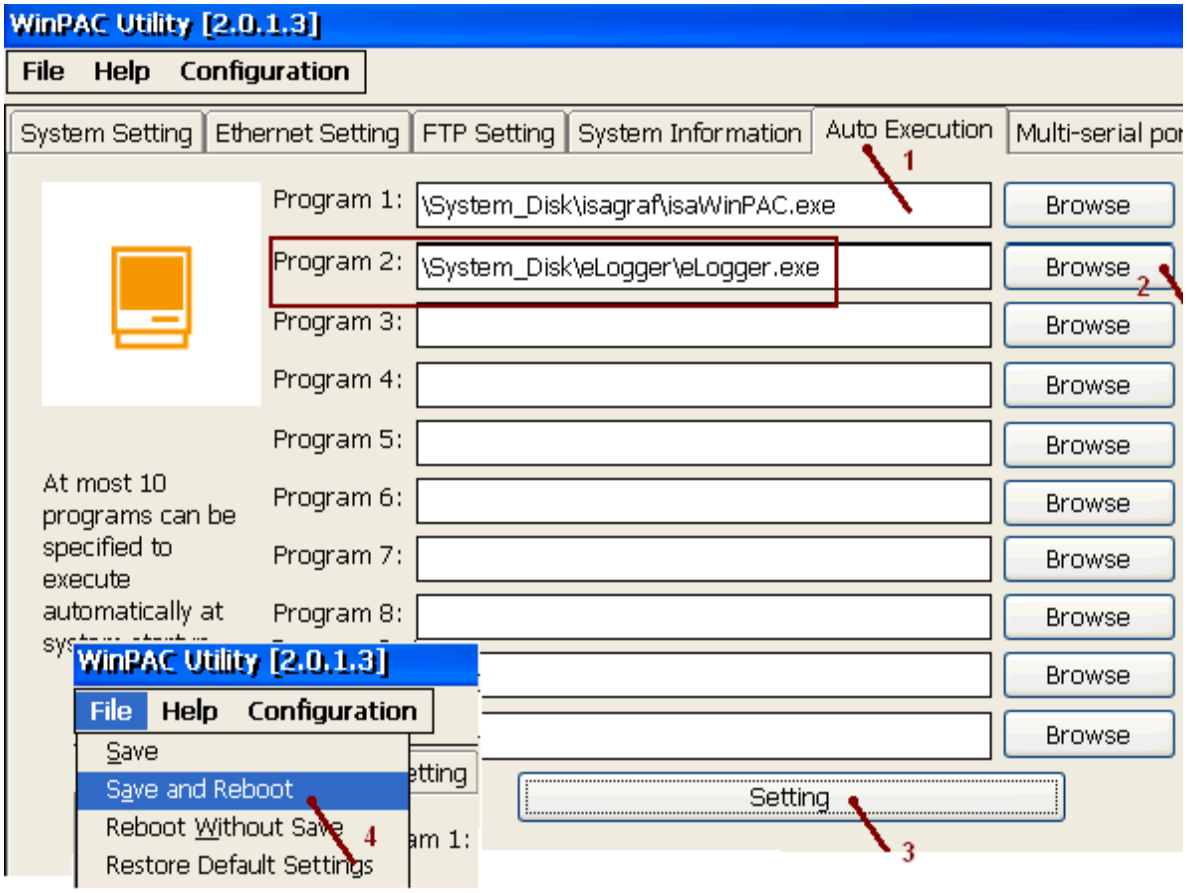


Entering the IP address of the connected controller on the Internet Explorer running in the PC, for example, **ftp://192.168.1.181**
Then get into the "\System_Disk\eLogger\" path, copy all files of the eLogger RunTime from PC to the controller's "\System_Disk\eLogger\" path.

Then click "My Device" on the VGA of the controller, get into the "\System_Disk\eLogger\" path, then run "eLogger.exe" . Check "Auto Run" and exit it .



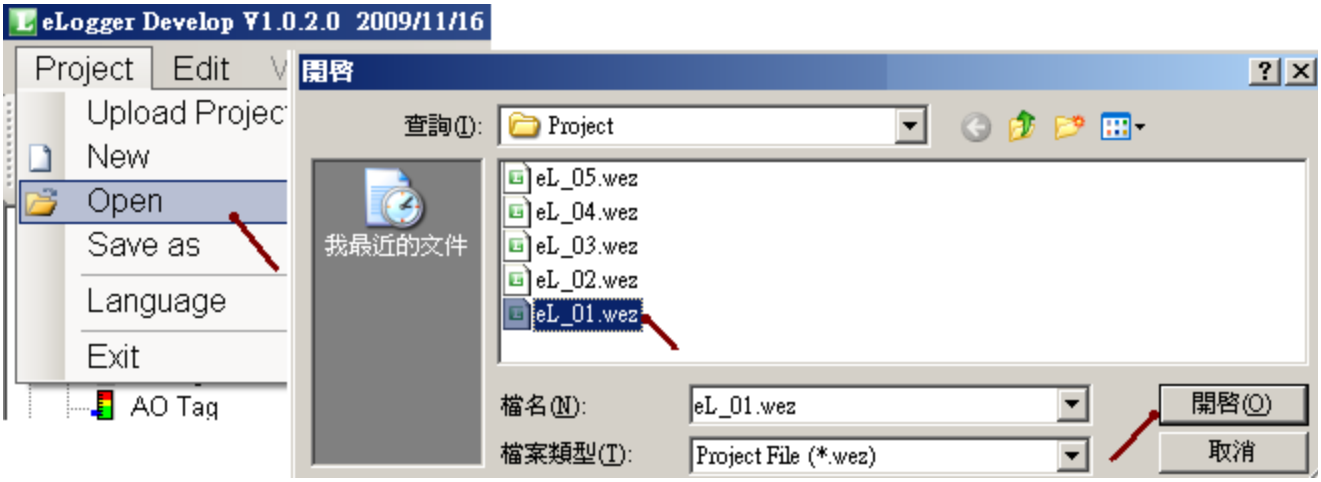
Run the WinPAC Utility (or ViewPAC Utility) to set the second “Auto Execution” as “\System_Disk\eLogger\eLogger.exe” (Note: the first execution of the WP-8xx7 and VP-2xW7 should be their ISaGRAF driver.). Then run File > “Save and Reboot” to save the settings.



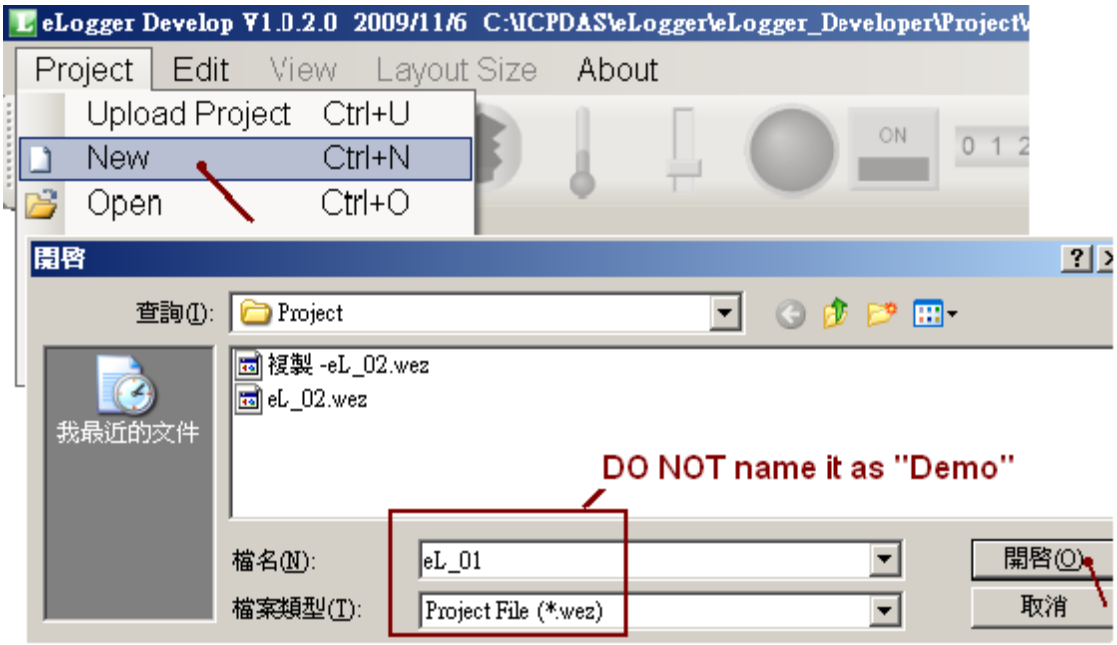
If problem happens while installing the eLogger RunTime, please refer to section 1.3 of this paper to fix it.

1.2.2: Building An eLogger Project

User may open an existing example project for reference. Copy “eL_01.wez” ~ “eL_06.wez” (downloaeded from Web 's FAQ-115) to the PC 's C:\ICPDAS\eLogger\eLogger_Developer\Project\ directory first. Then run eLogger Developer > Project > Open to open them.

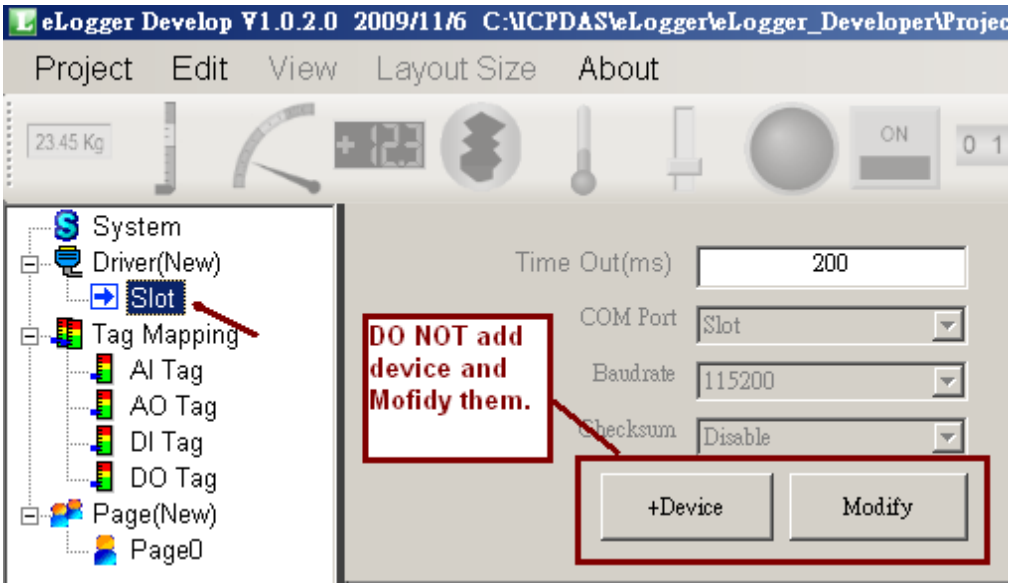


To create a new project - “eL_01” , please run Project > New . (DO NOT name it as “Demo”)



DO NOT check the “Local Database” if your controller will run eLogger HMI with ISaGRAF Softlogic. Recommend to set the “Sampling Time” as 1 second.

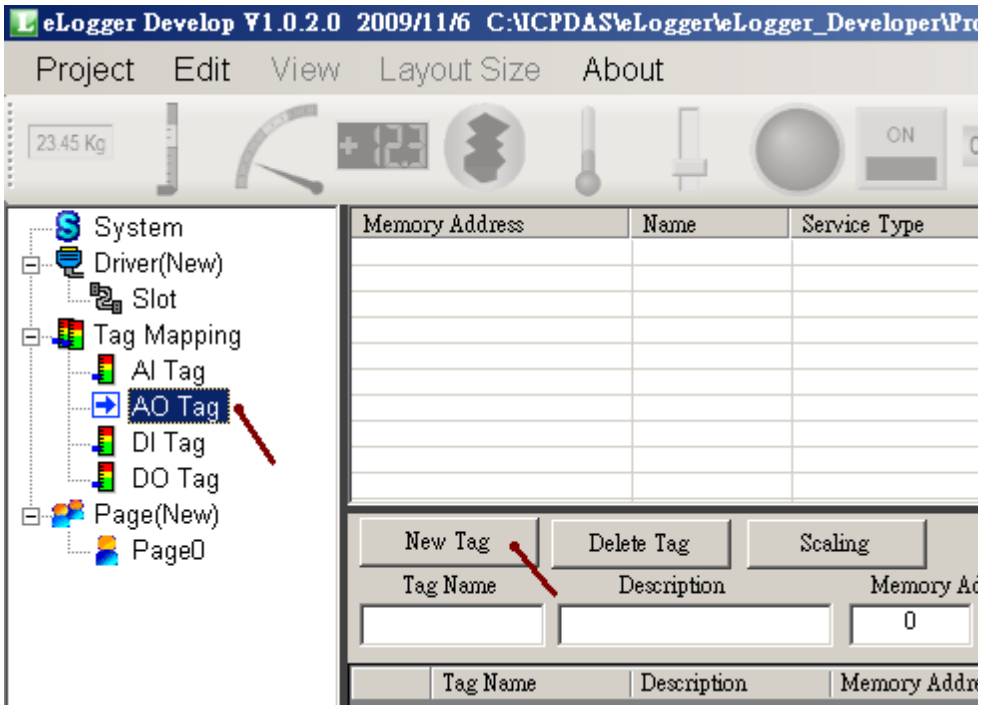
And DO NOT add any Device and Modify them in the eLogger 's Driver menu, leave them empty. (eLogger HMI working with ISaGRAF SoftLogic cannot support any eLogger Device, or it will fail.)



1.2.3: Declare eLogger Tags

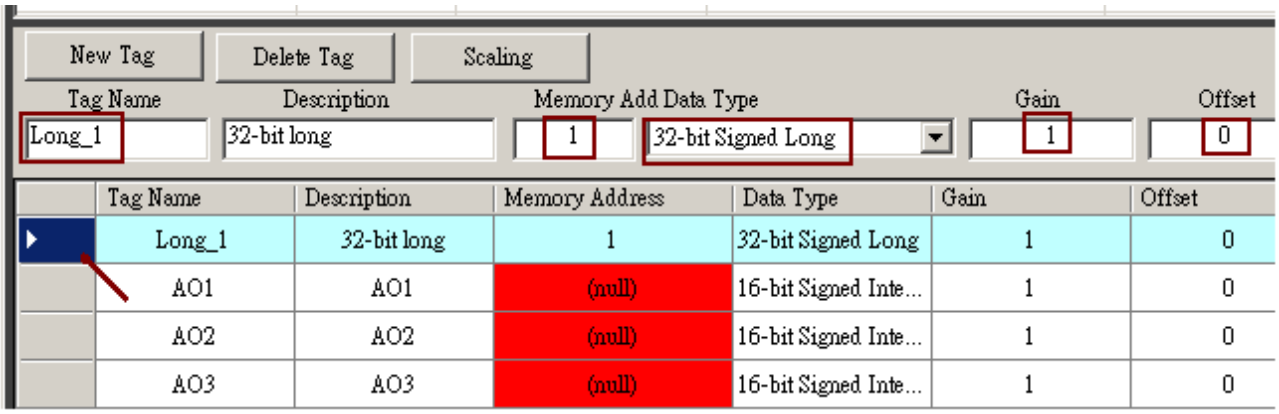
To make eLogger HMI communicate to the ISaGRAF SoftLogic well, only “AO Tag” and “DO Tag” can be used (eLogger 's AI Tag and DI Tag cannot communicate with ISaGRAF 's variables) .

Click “AO Tag” to declare several AO Tags.



Enter 4 in the following pop-up window.

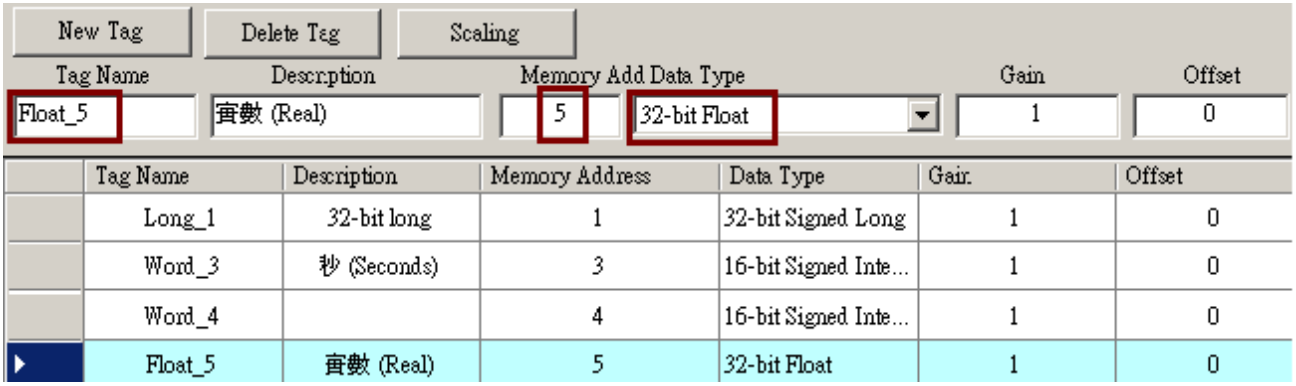
Then click the first tag to key-in the Tag Name as “Long_1” , Memory Add as 1, Data Type as “32-bit Signed Long” , Gain as 1 and Offset as 0 .
(ISaGRAF 32-bit Integer and Real must occupy two network address numbers. Here the address of “Long_1” is 1, so the address number “2” must not assign for other tags. Please refer to the section 4.2 of the ISaGRAF User's Manual)



Then declare the 2nd, 3rd and 4th AO Tag as below table.

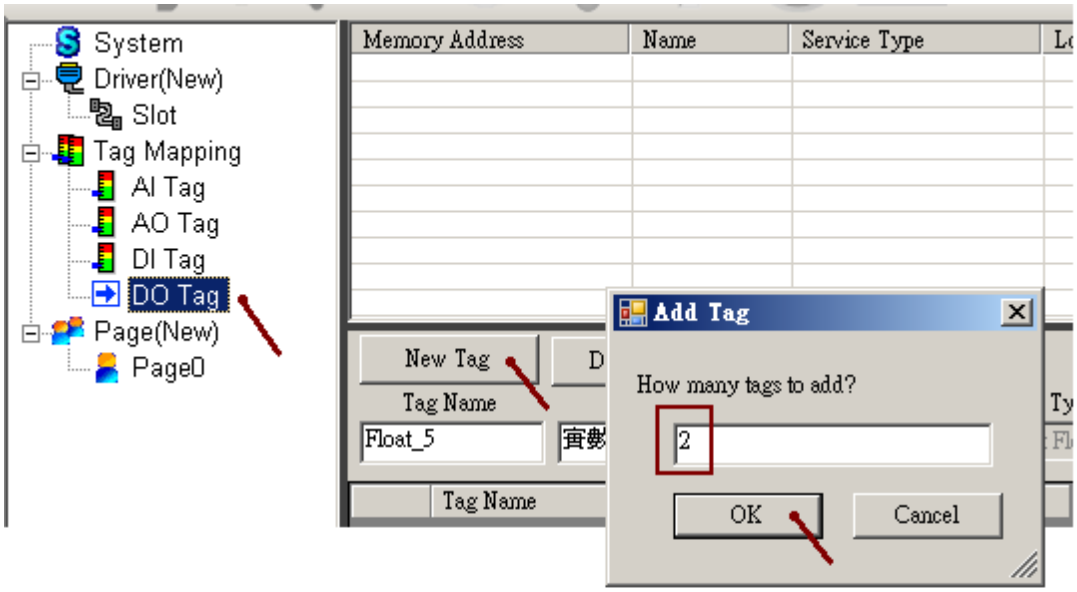
Tag Name	Memory Add	Data Type	Gain	Offset
Word_3	3	16-bit Signed Integer	1	0
Word_4	4	16-bit Signed Integer	1	0
Float_5	5	32-bit Flaot	1	0

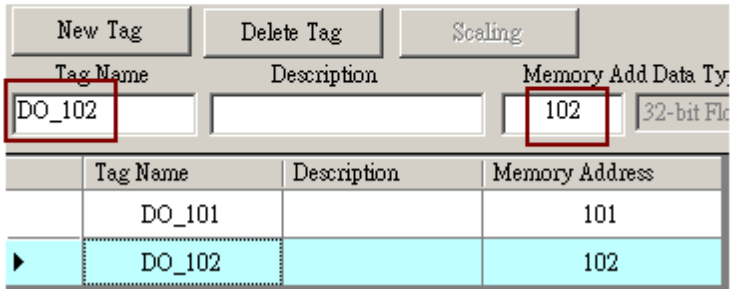
Then we got the following picture.



Click “DO Tag” to declare two DO Tags(DO_101 and DO_102 with address as 101 and 102 resoectively). The eLogger tag name can be a different name from the ISaGRAF variable (Because they are shared data by using the Memory Addr, called “Network Address” in ISaGRAF) .

Tag Name	Memory Add	Data Type	Gain	Offset
DO_101	101	-	-	-
DO_102	102	-	-	-

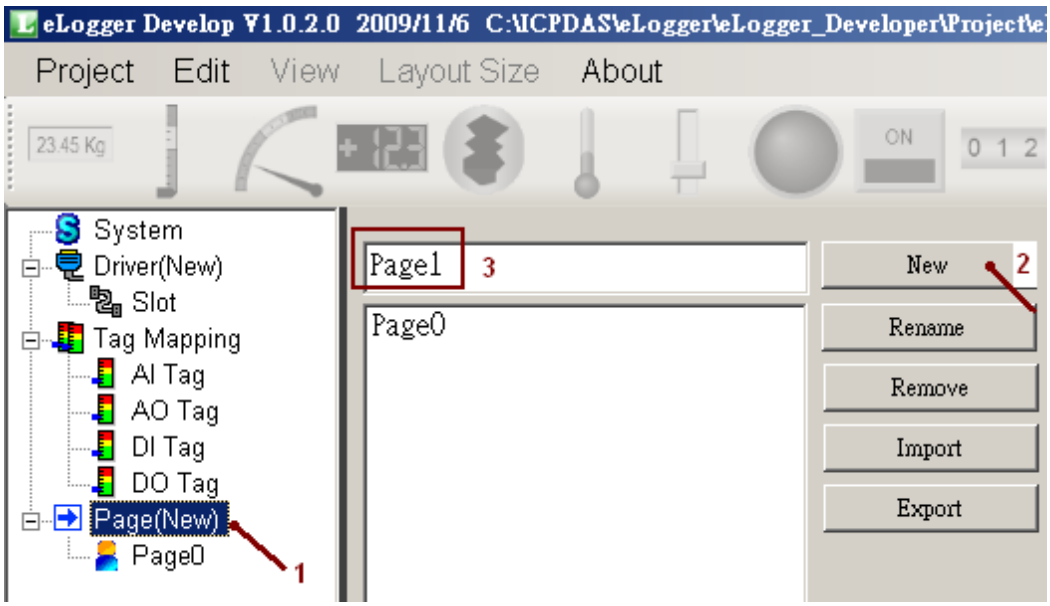




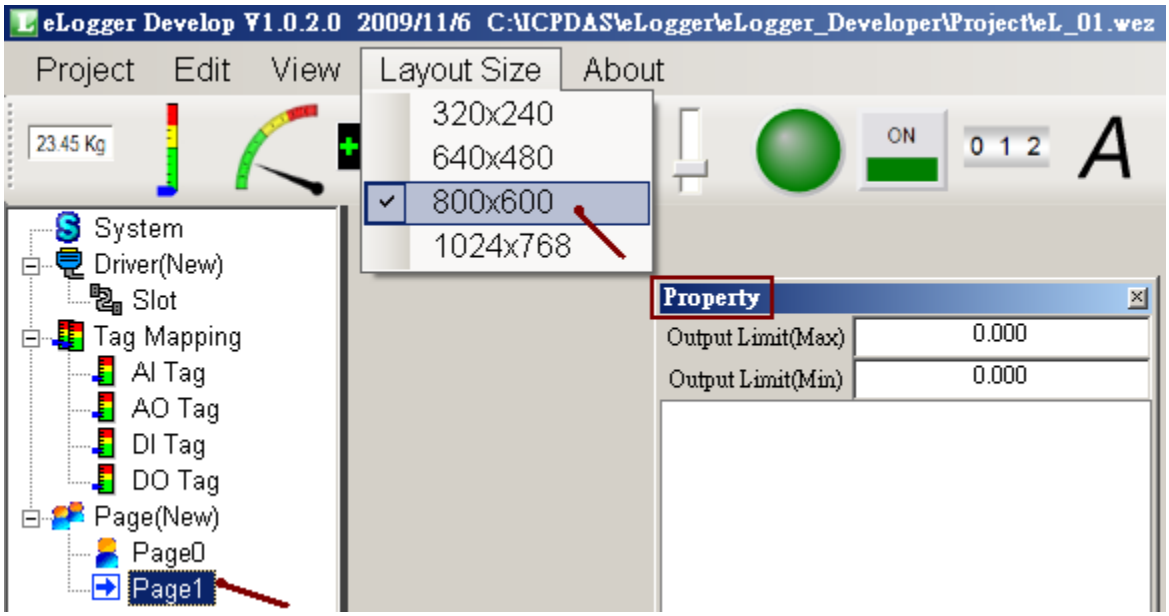
1.2.4: Creating An eLogger Page

The eLogger RunTime will show the “Page0” when started (user can rename it). This sample project uses two pages, the Page0 and Page1.

To declare the second page, click “Page(New)” on the left, then click “New” on the right. You may name it to a different name (or name it as Page1).

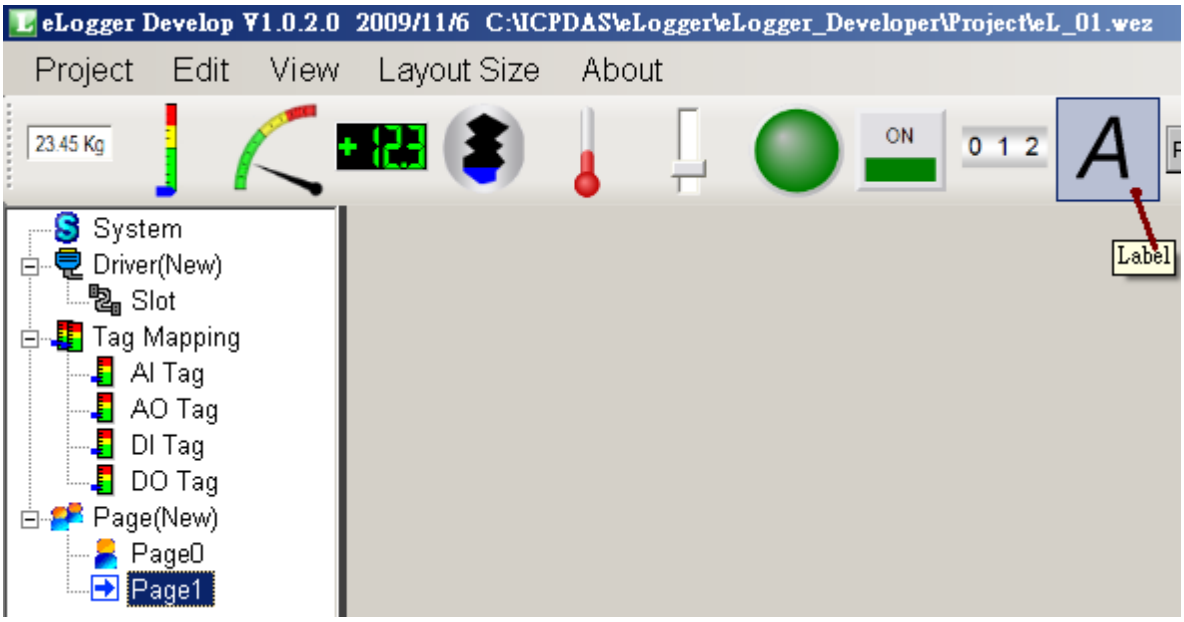


Click the "Page1" on the left to edit the second page. First, Select the VGA layout size to fit the PAC (The max. resolution is WP-8x47: 800 x 600, WP-8x37: 1024 x 768, VP-25W7: 640 x 480, VP-23W7: 320 x 240; The VP-23W7: doesn't build-in a Touch screen, however the VP-25W7 is built-in with a Touch screen). Then the "Property" window will pop-up, or please click View > Component Properties to open it.

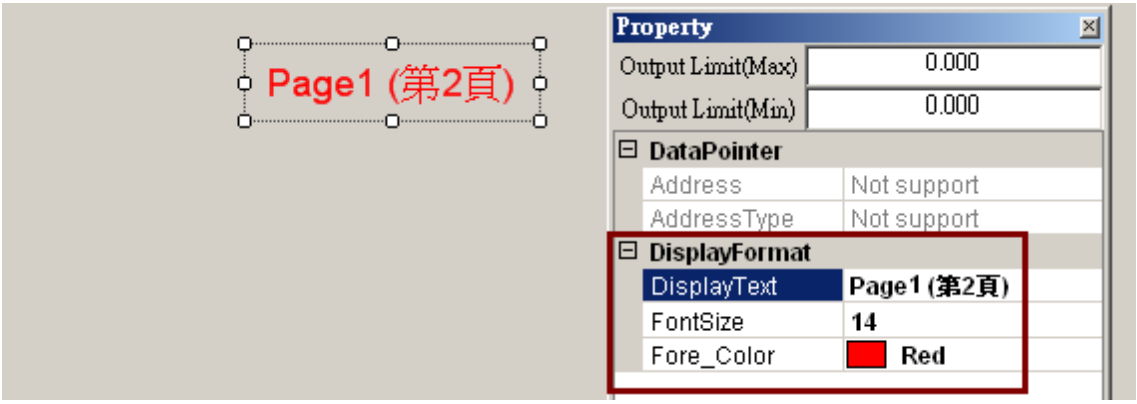


1.2.4.1: Add Component - Label

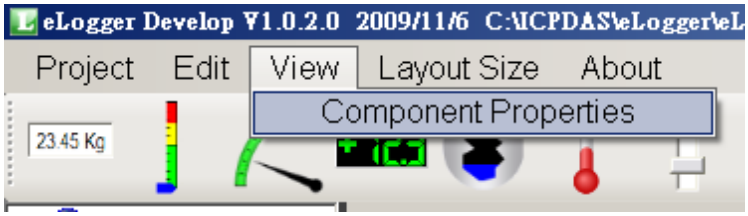
Add a component “Label” in the second page (named Page1) for this example.



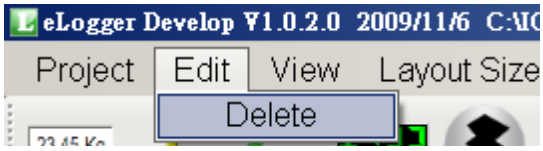
Drag the component to the proper position. Then set up the display text, font size and color in the Property window.



The Property window also can be opened by click View > Component Properties.



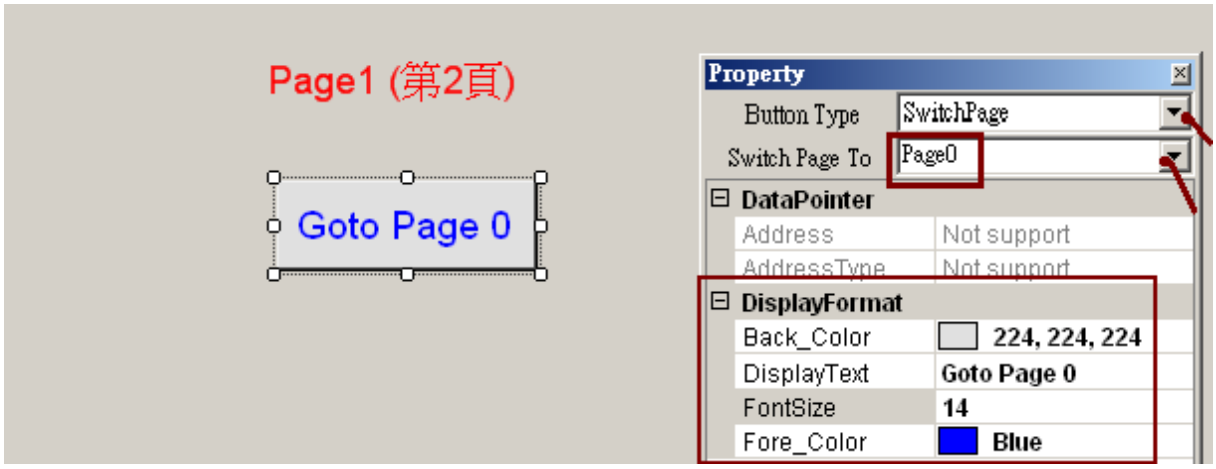
To delete an existed component, please select it then click Edit > Delete .



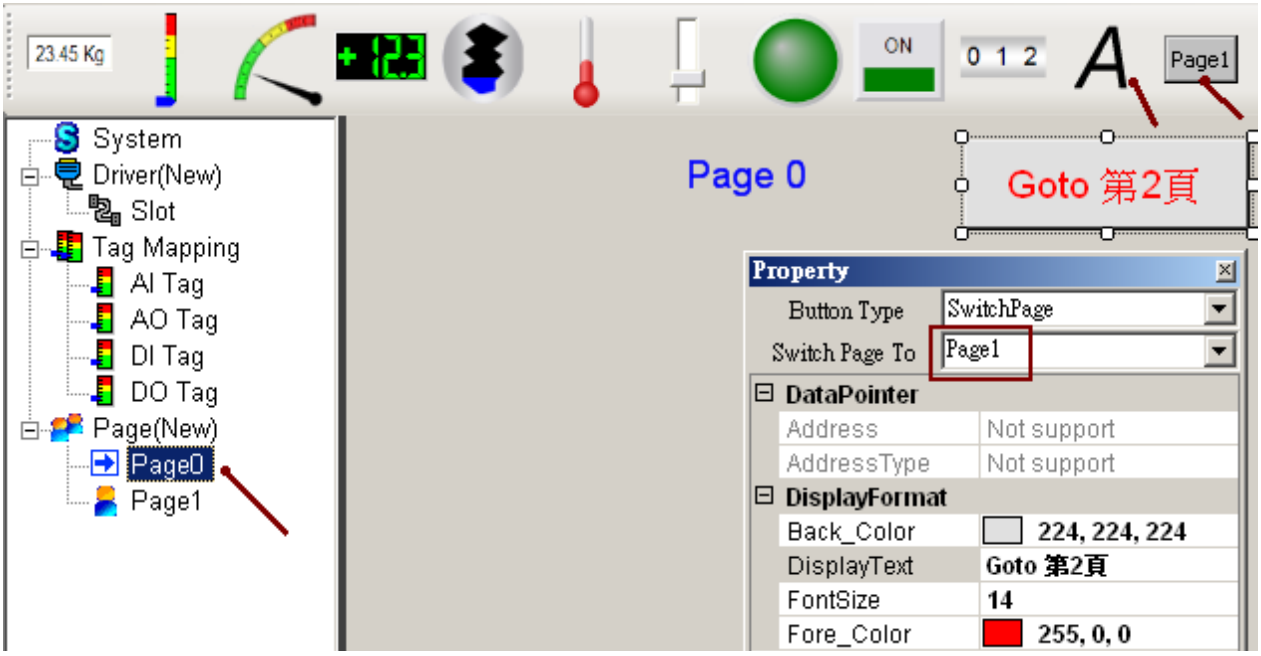
1.2.4.2: Add Component - SwitchPage

To add a “SwitchPage”, please click “Button” in the first page “Page1”.

Set Button Type as “SwitchPage”, Switch Page To as “Page0” . Then, setup the “DisplayText”, “FontSize” and “Fore_Color” you want.



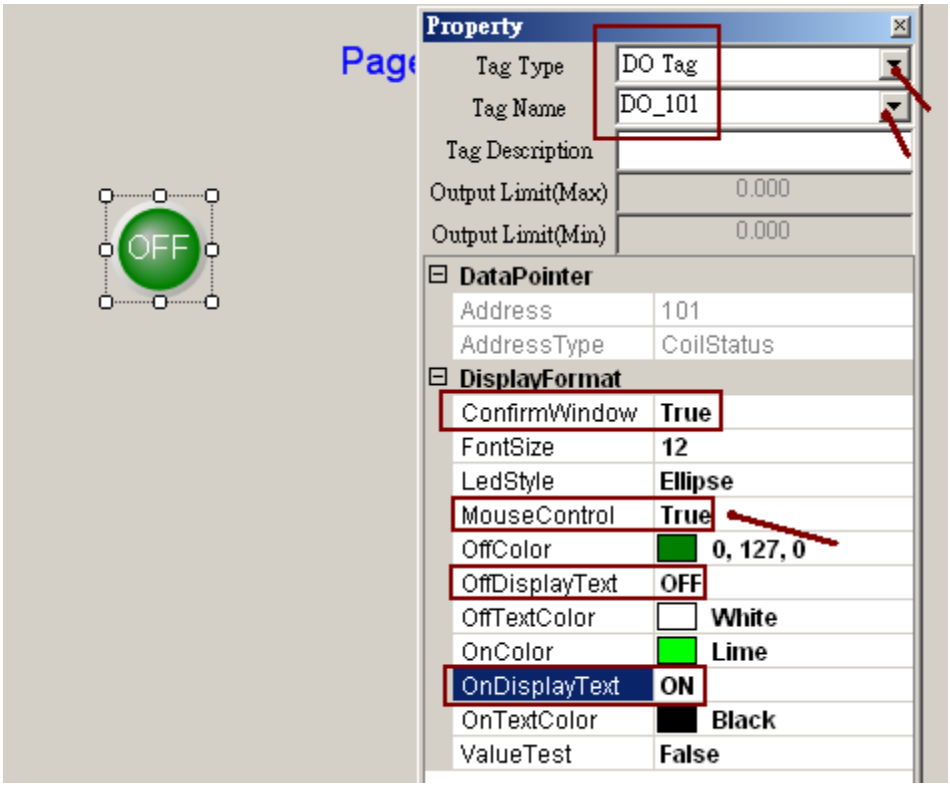
Then, click “Page0” on the left, add a “Label” and a “SwitchPage” by the similar way.



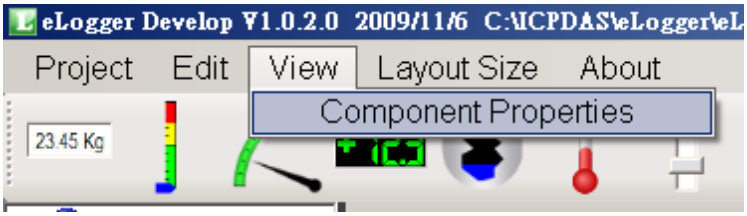
1.2.4.3: Add Component - LED

In the page0, click “LED” to add a LED component.

This “LED” is to control the ISaGRAF variable OUT_101 (the relative DO Tag in eLogger is DO_101). Please set Tag Type as “DO Tag” , Tag Name as “DO_101” , ConfirmWindow as “True” (will output after confirmation), MouseControl as “True” (True: allow Tag outputs its value; False: read only, not allowed to output). Then setup the Text and color you want.

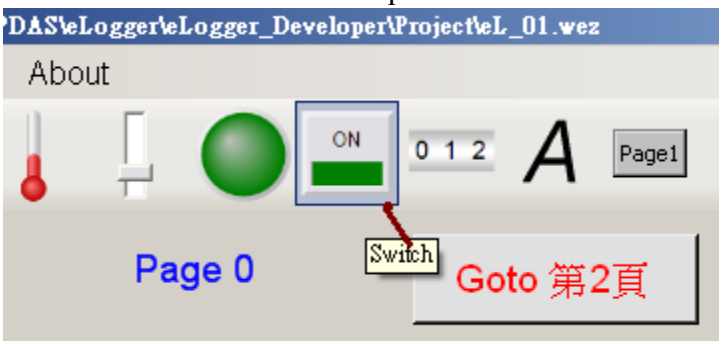


Please click View > Component Properties to open this Property window if it is closed.

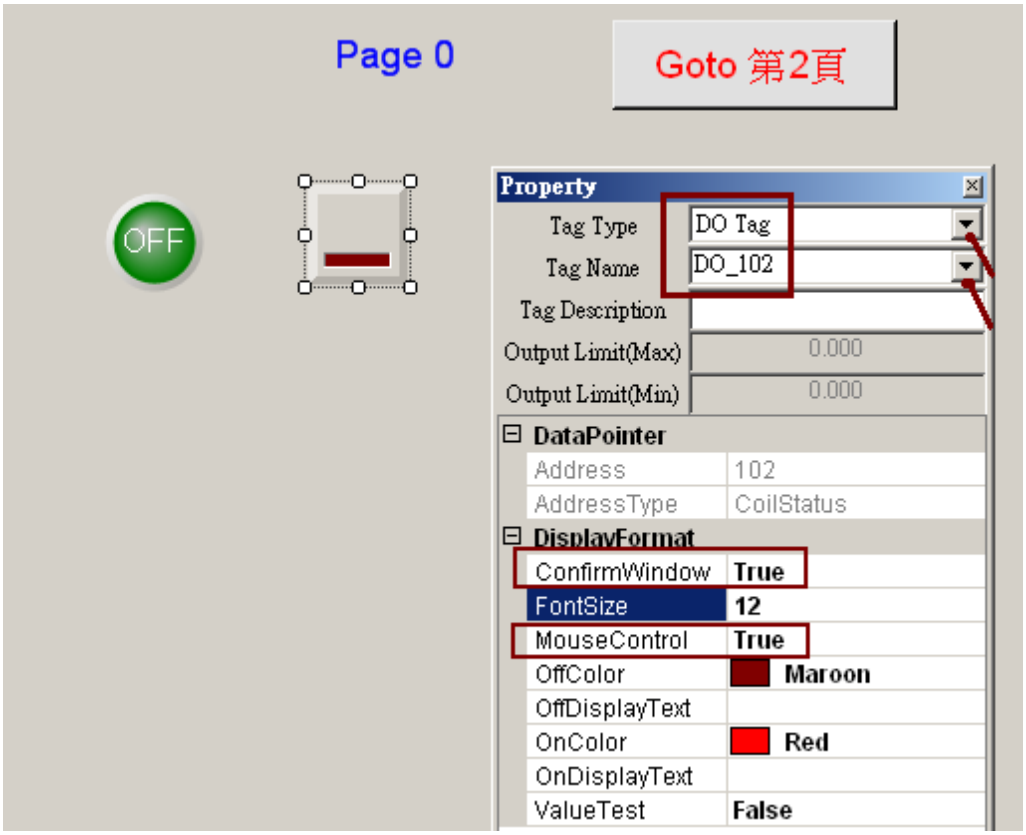


1.2.4.4: Add Component - Switch

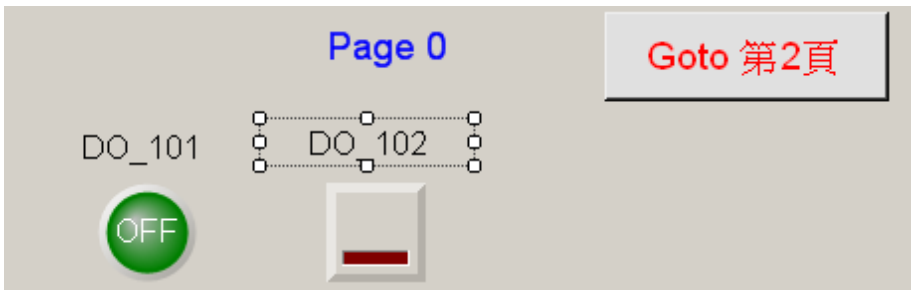
In the Page0, click “Switch” to add a Switch component.



Set Tag Type as “DO Tag” , Tag Name as “DO_102” , ConfirmWindow as “True” (will output after confirmation), MouseControl as “True” (True: allow Tag outputs its value; False: read only, not allowed to output). Then setup the Text and color you want.

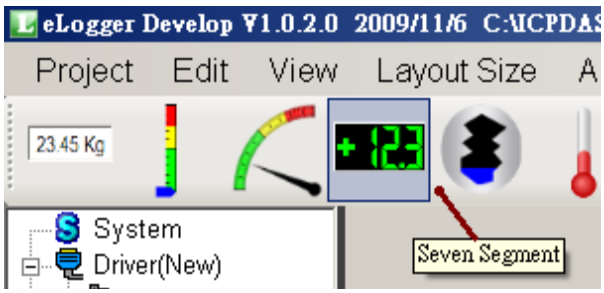


Then add two Labels as the descriptions for the LED and Switch (refer to section:1.2.4.1).

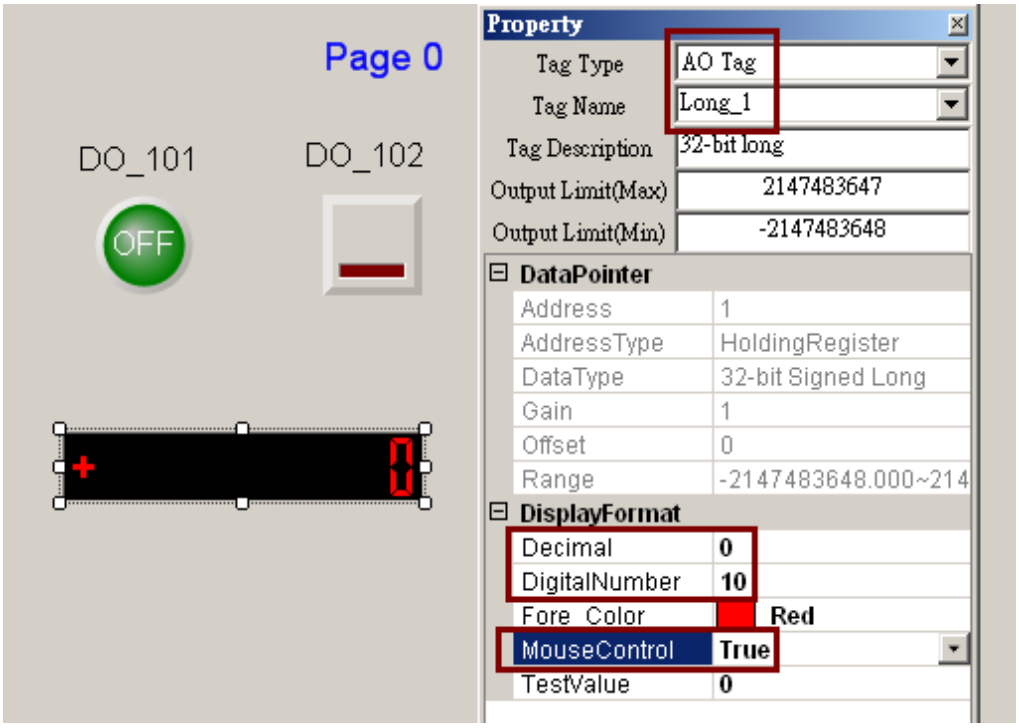


1.2.4.5: Add Component - Seven Segment

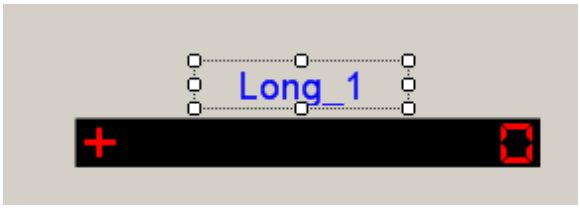
Click “Seven Segment” to add a Seven Segment numerical component.



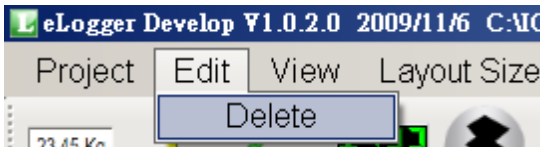
Set Tag type as “AO_Tag” , Tag Name as “Long_1” and MouseControl as “True” (True: allow Tag outputs its value; False: read only, not allowed to output). In this example, the data type of long_1 is 32-bit long, no decimal places needed, so set “Decimal” as 0 , DigitalNumber as 10 (can be 1 ~ 24). If user’s application has output range limitation for local operators, please do the extra setting in the “Output Limit (Max)” and “Output Limit (Min)” .



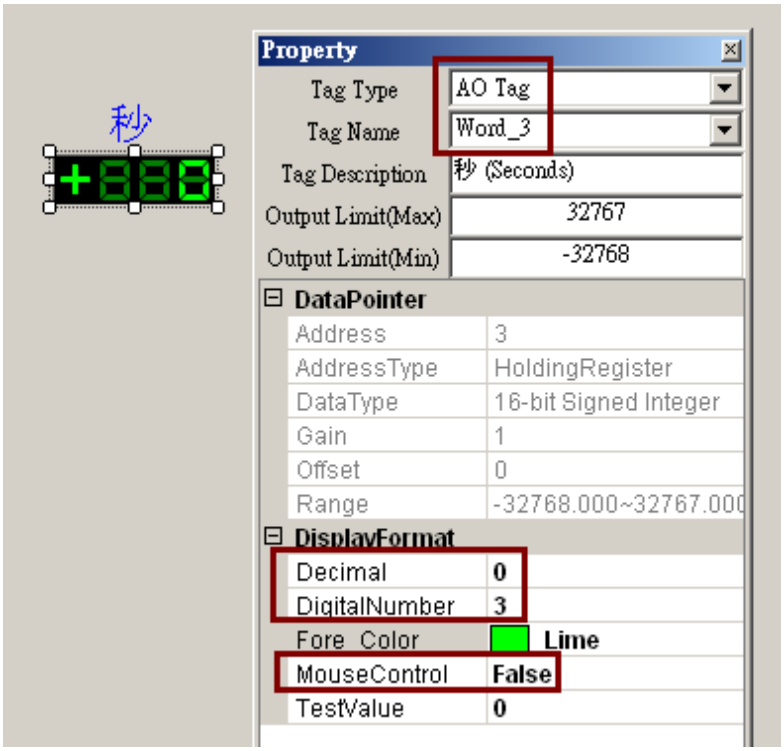
Then add a Label as its description.



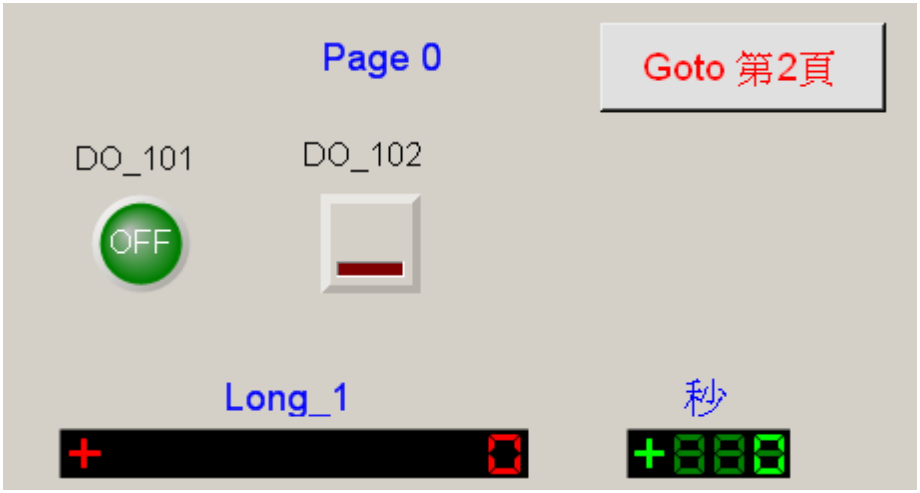
To delete an un-need component, please select it, then click Edit > Delete .



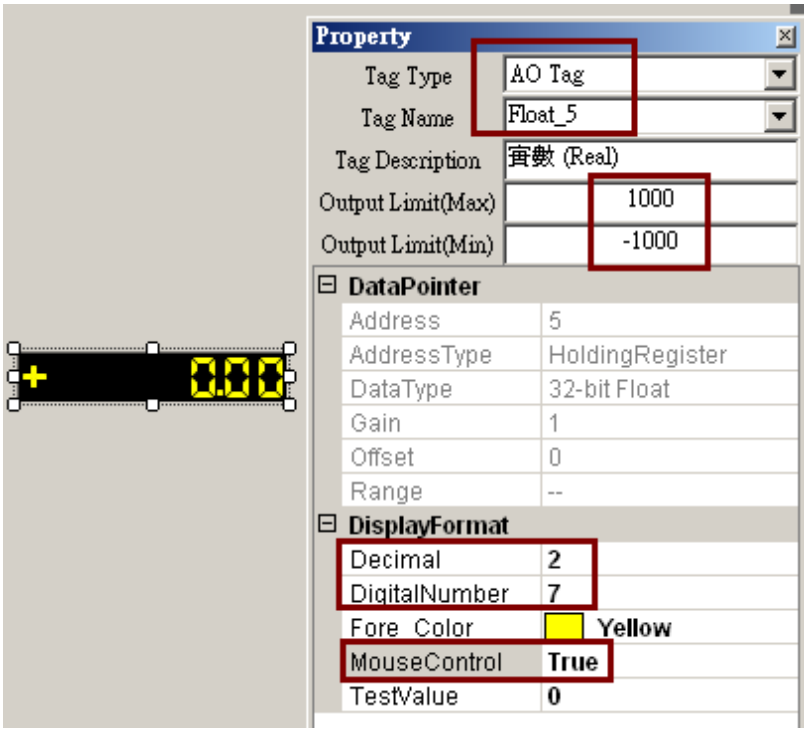
By the similar way, add one more “Seven Segment” component with Tag Type as “AO Tag” , Tag Name as “Word_3” (In this example, this value is to read the “second” value from the WP-8xx7 PAC’s time value), “MouseControl” as “False” (True: allow Tag outputs its value; False: read only, not allowed to output) , Decimal as 0 and DigitalNumber as 3.



Then we got the following picture – Page0.



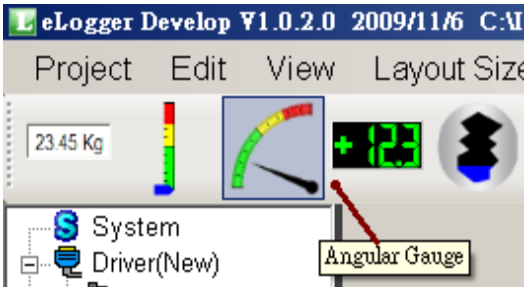
Then by the similar way, add a “Seven Segment” component with Tag Type as “AO Tag” , Tag Name as “Float_5” , MouseControl as “True” (True: allow Tag outputs its value; False: read only, not allowed to output), DigitalNumber as 7 and “Decimal” as 2 (display 2 decimal places). The range that operator can input is -1000 ~ +1000, so please set its “Output Limit(Max)” as 1000 and “Output Limit(Min)” as -1000 .



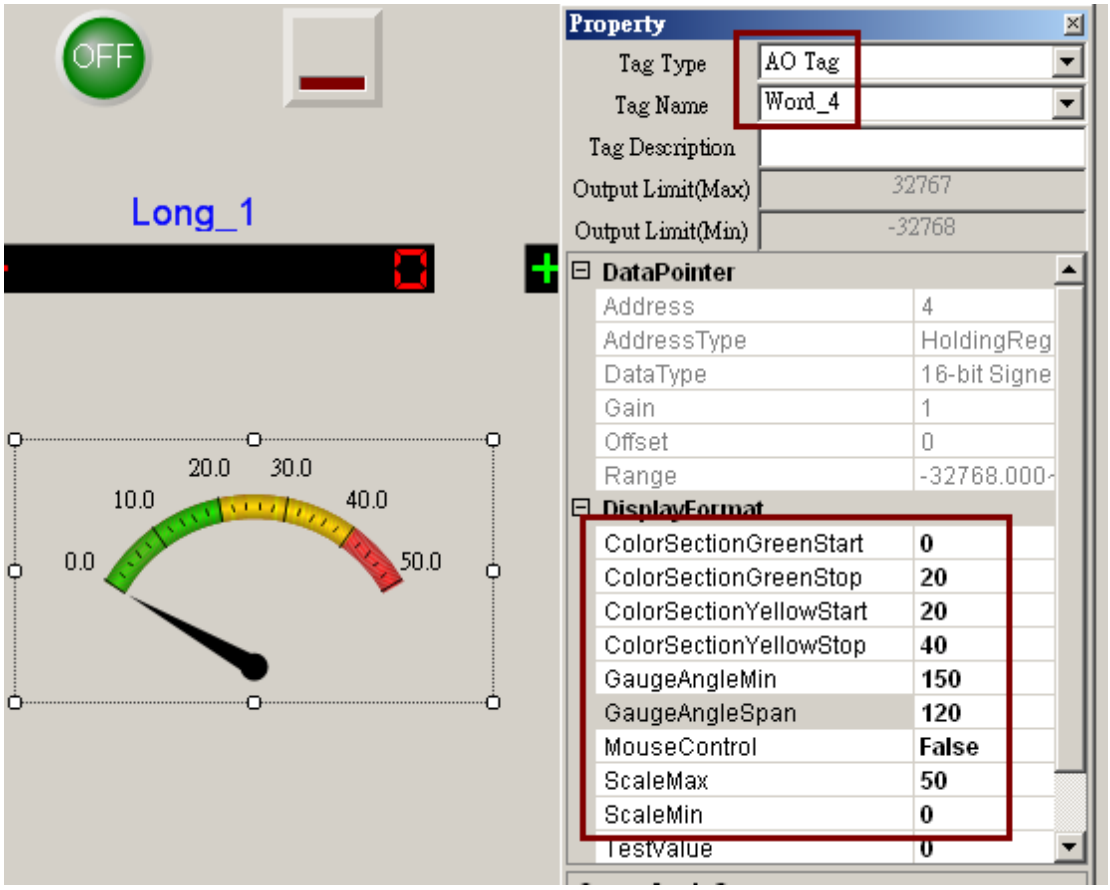
Then add a Label as its description. Then we got the following picture.

1.2.4.6: Add Component - Angular Gauge

To add an “Angular Gauge”, please click Angular Gauge as below.

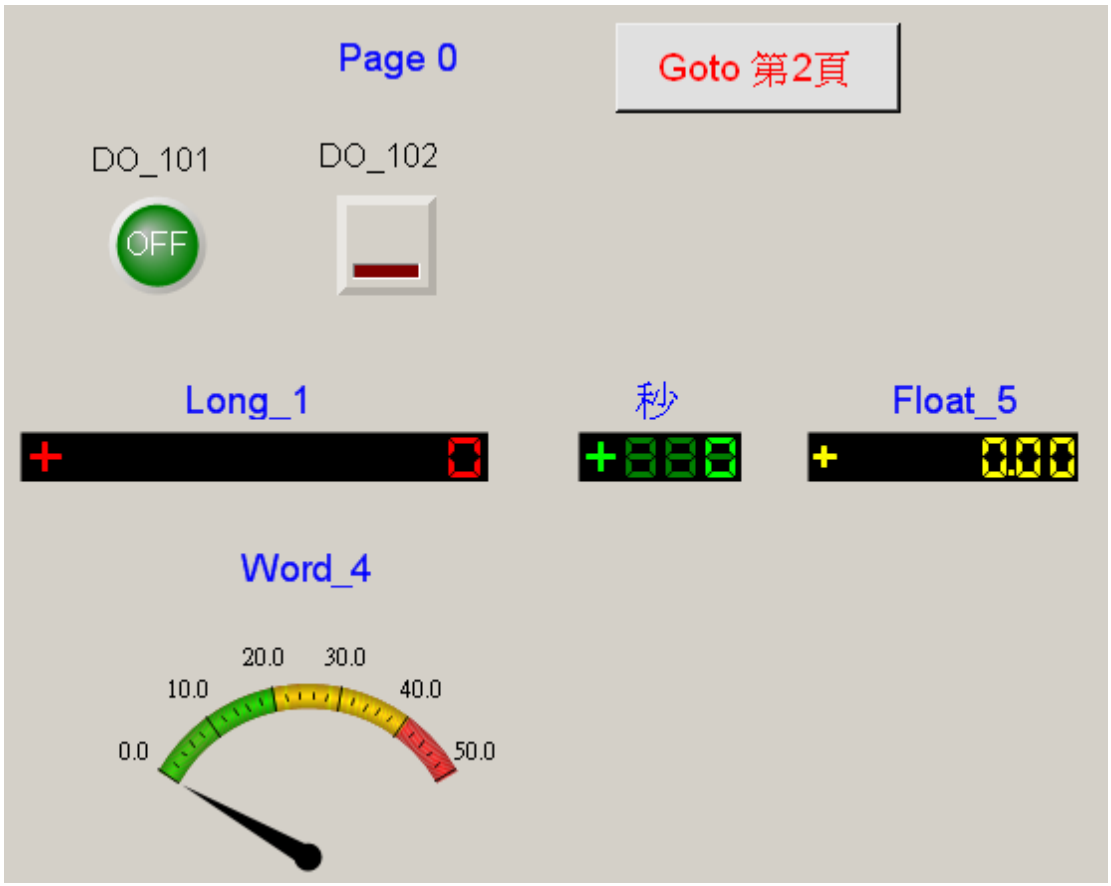


First, set Tag Type as “AO Tag” , Tag Name as “Word_4”, MouseControl as False, “ScaleMin” as 0, “ScaleMax” as 50 (In this example, ISaGRAF output range is 0 ~ 50). Next set “GaugeAngleMin” as 150 degrees (It means the minimum value is in the location of 150 degrees, anti-clockwise direction), “GaugeAngleSpan” as 120 degrees (It means the entire header open up can be 120 degrees.), “ColorSectionGreenStart” as 0 , “ColorSectionGreenStop” as 20 , “ColorYellowSectionStart” as 20 and “ColorYellowSectionStop” as 40.



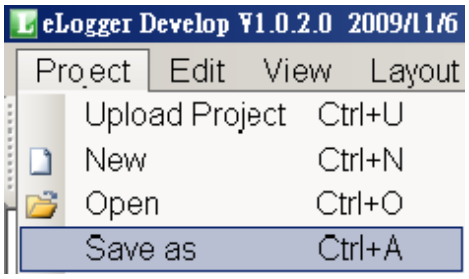
Add a Label as its description. At last, we got the following picture.

Page 0:



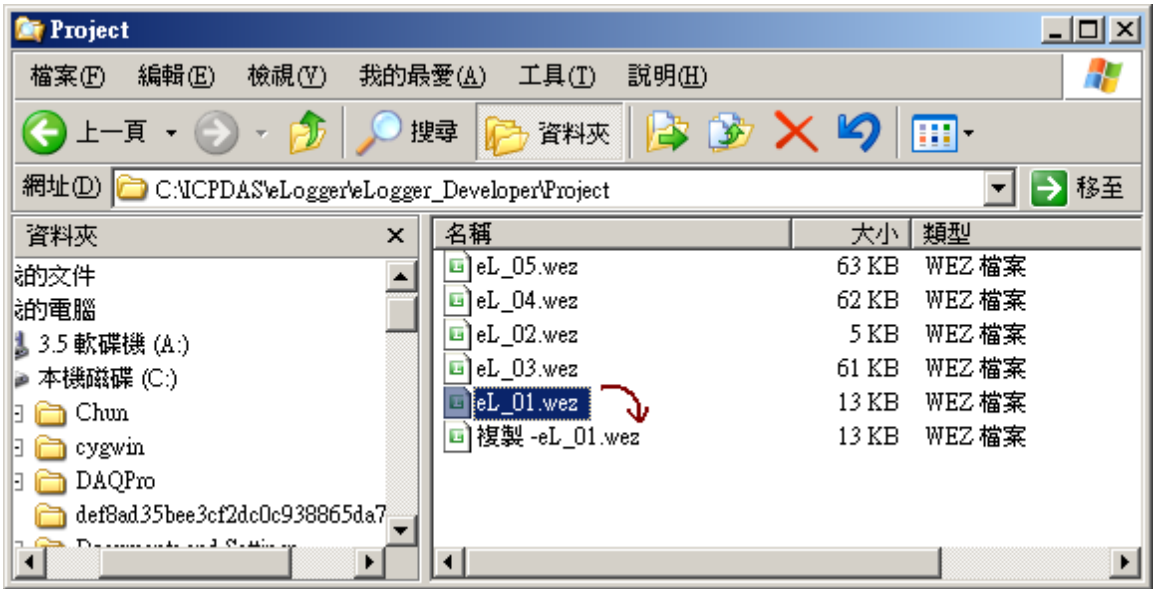
Page 1:

Save Project.



1.2.4.7: Recommend To Backup The Finished Project File For Safety

For safety reason, recommend to backup the project file.The finished eLogger Project file resides in the C:\ICPDAS\eLogger\eLogger_Developer\Project\ directory.

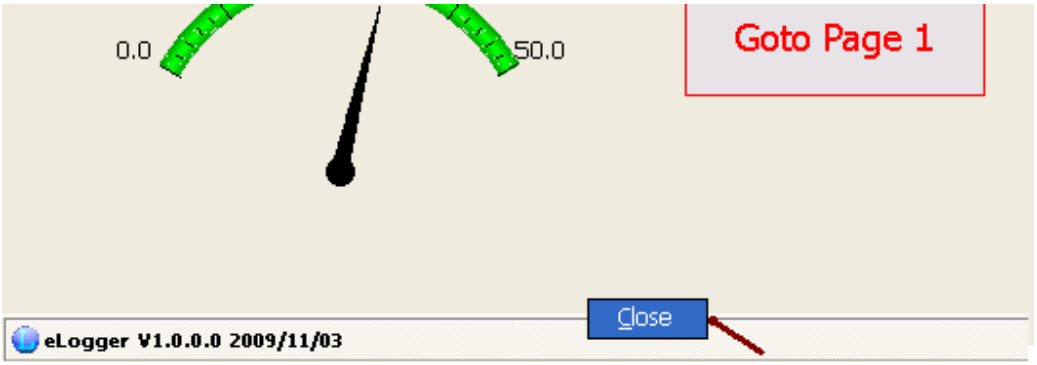


1.2.5: Download the eLogger project to PAC

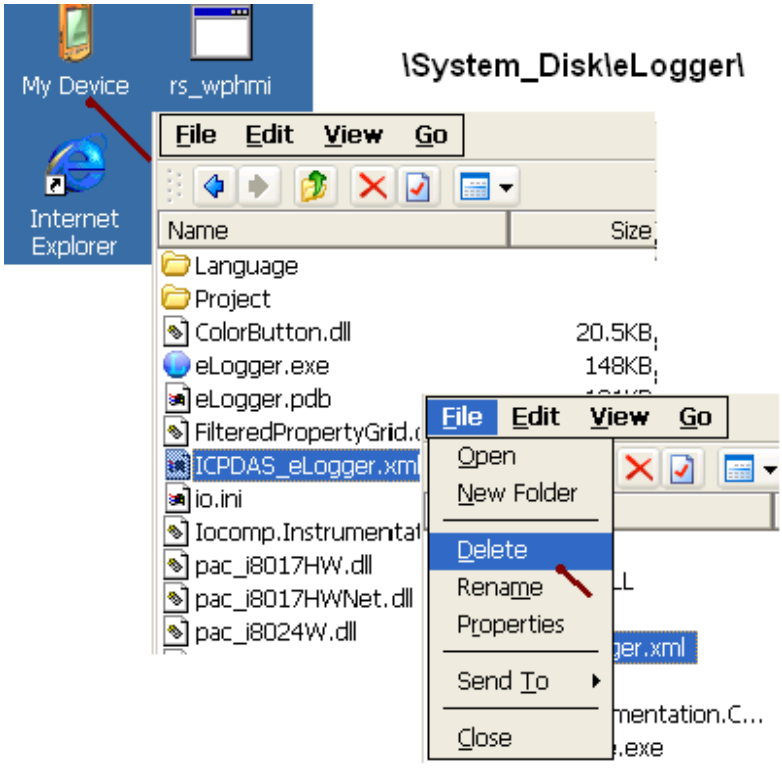
This example uses a WP-8447 + slot 1: I-87055W(leftmost I/O slot number is 0) as the hardware and uses the ISaGRAF SoftLogic + eLogger HMI as the software. First, please download the ISaGRAF project into WP-8447. (Please refer to the section 1.1.7 or 1.1 of this paper.)

The PAC must pre-install the eLogger RunTime and setup the relative configuration. (Please refer to the section 1.2.1 of this paper.) Then turn on the power, ISaGRAF and eLogger RunTime will start up automatically.

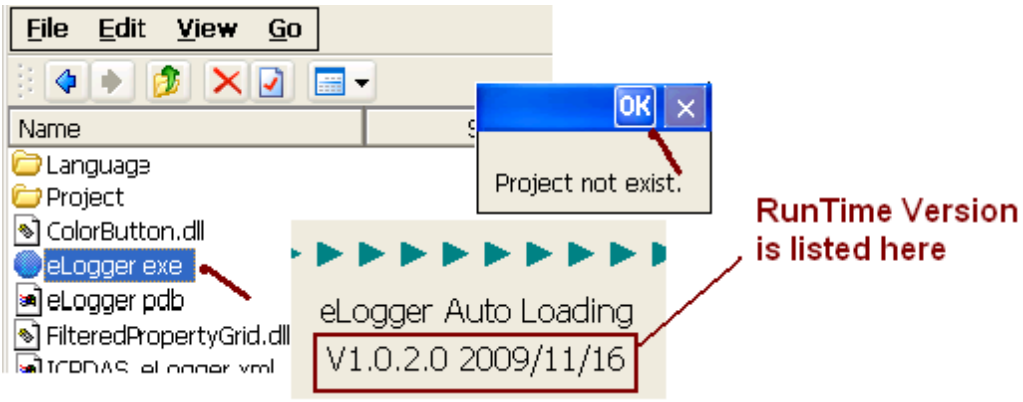
Before downloading the eLogger project, must close the eLogger project currently running in the PAC. Please use mouse right key click on the "eLogger" Taskbar in the screen bottom to close it. (If using Touch Monitor, such as VP-25W7, there is no Mouse right key, please turn off the WP-8xx7/VP-2xW7 PAC, switch its Rotary Switch to 1, then reboot the PAC. The eLogger will not do the Auto-Execution and ready for the next step. After download the eLogger project to the PAC, remember switch the Rotary Switch to 0, then reboot PAC).



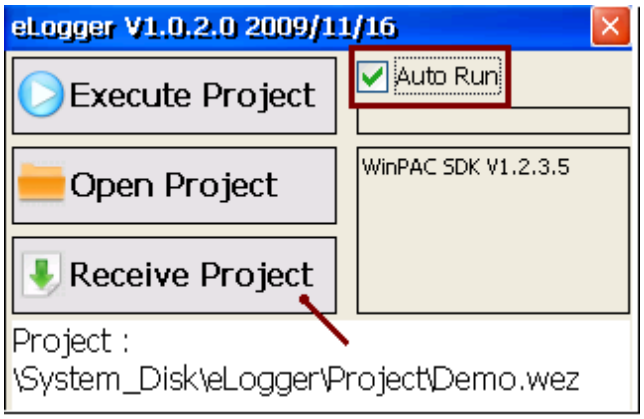
Double click "My Device", then get into " \System_Disk\eLogger\" path to delete the file "ICPDAS_eLogger.xml" .



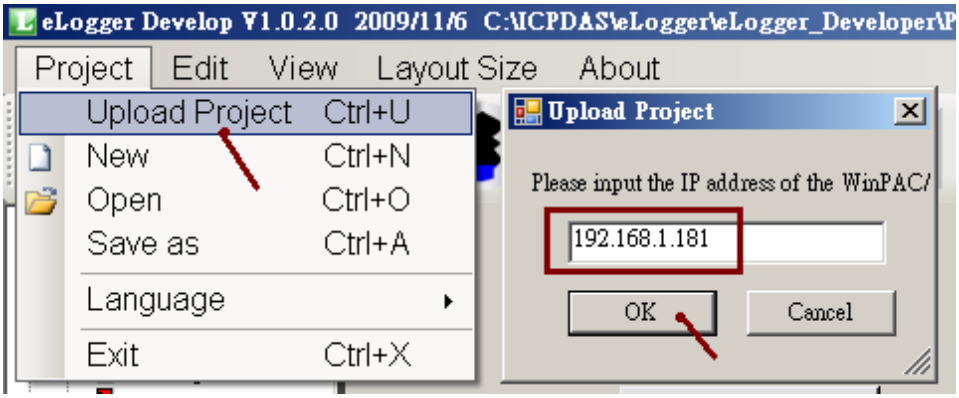
After that, run the eLogger.exe again (mouse double click). The eLogger will show “Project not exist” (Because the file ICPDAS_eLogger.xml has been deleted.)



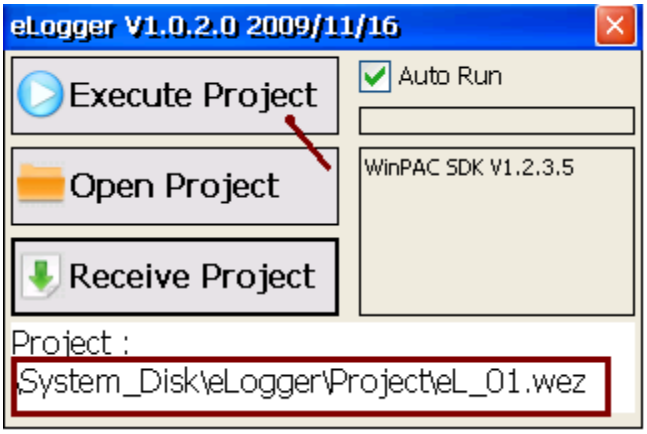
Click “OK” in the top. Don’t change the “Auto Run”, that had checked in the section 1.2.1.
Please click “Receive Project” to wait the PC / eLogger Developer sending over the eLogger project.



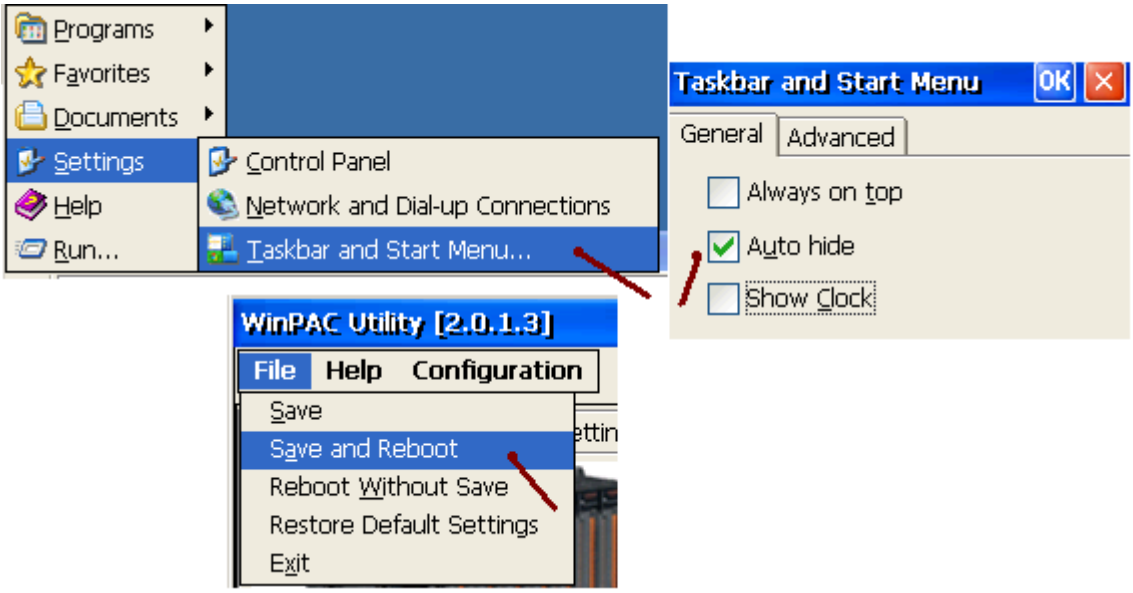
On the PC side, to send the eLogger project to PAC, please click Project > “Upload Project” of eLogger Developer, then input the PAC 's IP address and click “OK”.



If the operation is successful, the eLogger Run Time will show the eLogger Project name (eL_01.wez in this example). Please click “Execute Project” to run the new eLogger project, or reset the PAC that also can run the new eLogger project (If the eLogger does not auto-run after resetting PAC, it means the eLogger does not set to Auto-Execution and Auto-Run. Please refer to the last 2 steps of the section 1.2.1.).



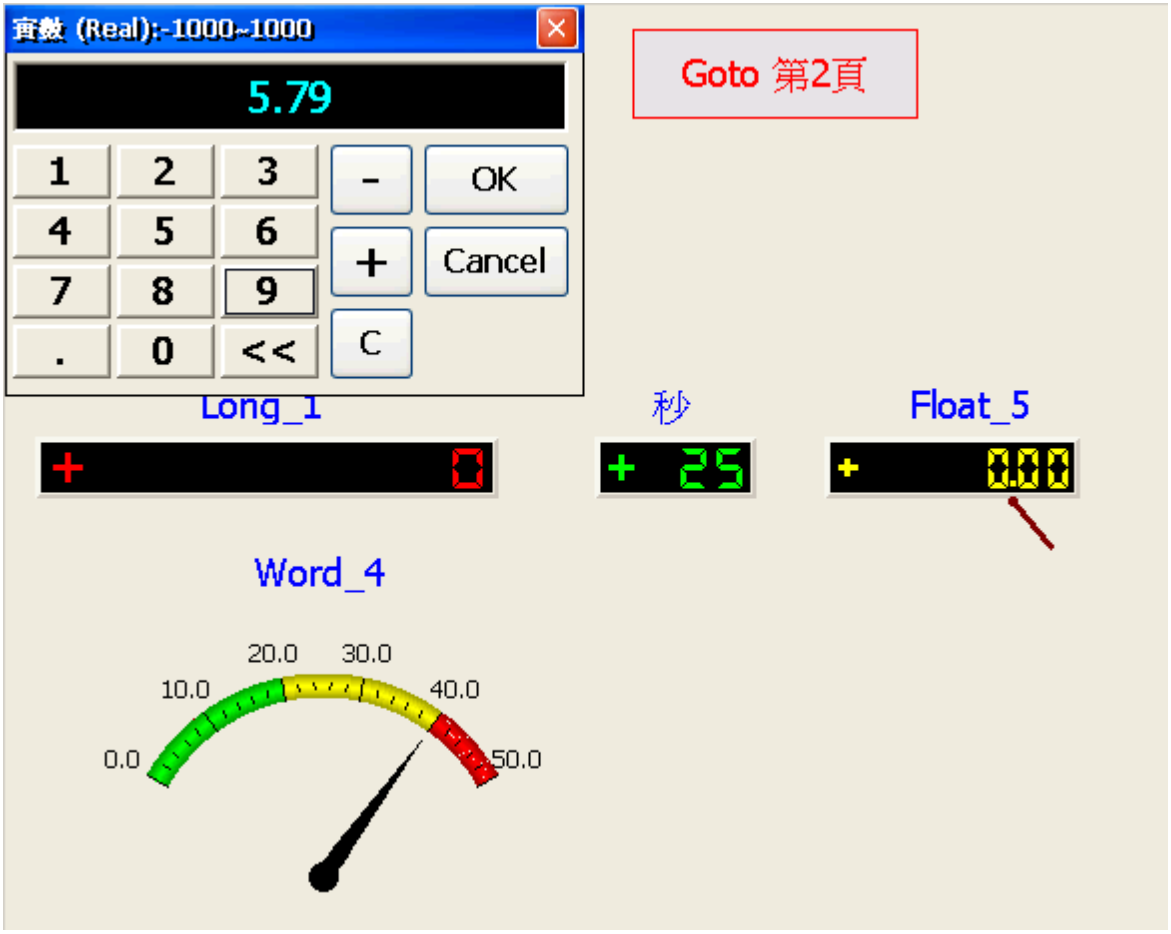
After test and confirm the Project, when the PAC is ready for long time running, user can hide the Taskbar, so that the eLogger project can be displayed on full screen. Then run WinPAC Utility (or ViewPAC Utility), click File > “Save and Reboot” to save the setting to the Registry.



It is suggested to display the Taskbar when it is in the design and test time.
When using the full screen display for the eLogger HMI (hide the Taskbar), please refer to the section 1.3 if user want to modify the eLogger project again.

1.2.6: Testing The eLogger HMI

When WP-8xx7 / VP-2xW7 (or the XP-8xx7-CE6 available in the future) has received and Auto-Run the eLogger project, the “eL_01” (in this example) will show on the VGA of PAC as below. You can click “Float_5” and input a value to test it, and use ISaGRAF to connect the PAC to see if the value correct or not.

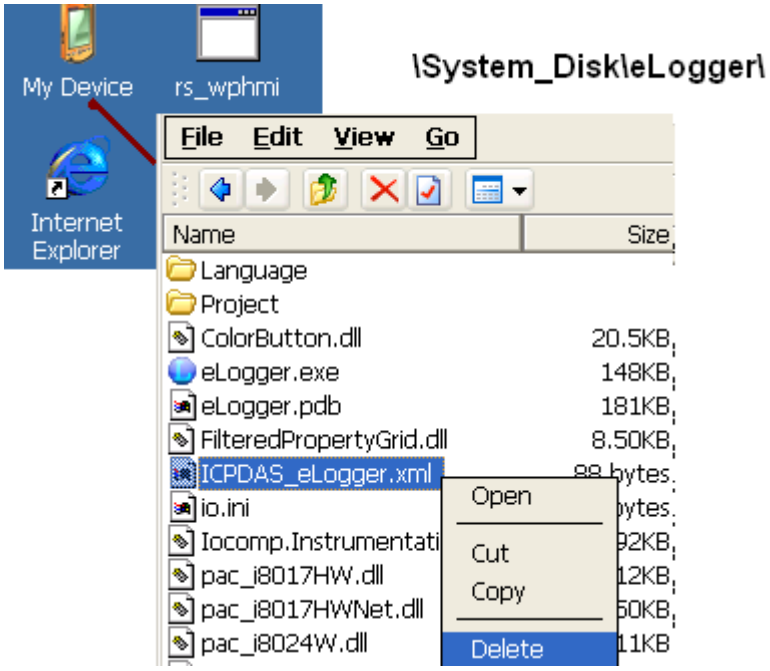


1.3: eLogger Run Time Trouble Shooting

If there are troubles when installing the eLogger Run Time, or the eLogger Run Time cannot be turned off when using the eLogger full screen, please power off the PAC, switch the Rotary Switch to 1 , then reboot the PAC.

After rebooting, ISaGRAF and eLogger will not auto-Run.

Next, get into " \System_Disk\eLogger\" path to delete the file "ICPDAS_eLogger.xml" (or rename it). Then when the eLogger auto-run again, it will not find the project, so that the user can download the new Project to it. (If want to re-install the eLogger Run Time, first, delete all the files in the folder " \System_Disk\eLogger \" , then follow the steps in section 1.2.1 to re-install it.)



After deleting \System_Disk\eLogger\ICPDAS_eLogger.xml, please turn off the PAC, switch the Rotary Switch to 0, reboot the PAC. The eLogger project will not auto-run this time. (Because it cannot find the eLogger project.)

1.4: Other eLogger Advanced Functions

1.4.1: Setting Gain & Offset for scalling data

In many applications, it needs to display the engineering data values rather than the integer data values. For instance, the I-8017HW module, when setting the "Range" as 8, its value measured in ISaGRAF is (-32768 ~ +32767). That is a signed 16-bit integer to represent (-10 ~ +10) Volt. To display the -10 ~ +10 value on the eLogger HMI, we must set the Gain and Offset.

NOTE: If the Gain is set to 1 and the Offset is set to 0, it means without data scalling.

Formula : Y = [Gain] * X + [Offset]

Convert (X0 ~ X1) to (Y0 ~ Y1), that

Gain = (Y1 – Y0) / (X1 – X0)
Offset = (X1 * Y0 – X0 * Y1) / (X1 – X0)

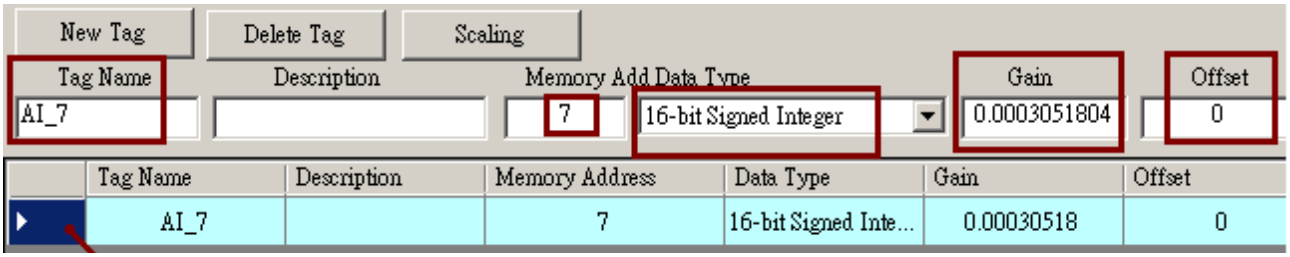
For example: convert (-32768 ~ 32767) to (-10 ~ 10) , that

X0 = -32768 , X1 = 32767 , Y0 = -10 , Y1 = 10

So the result will be

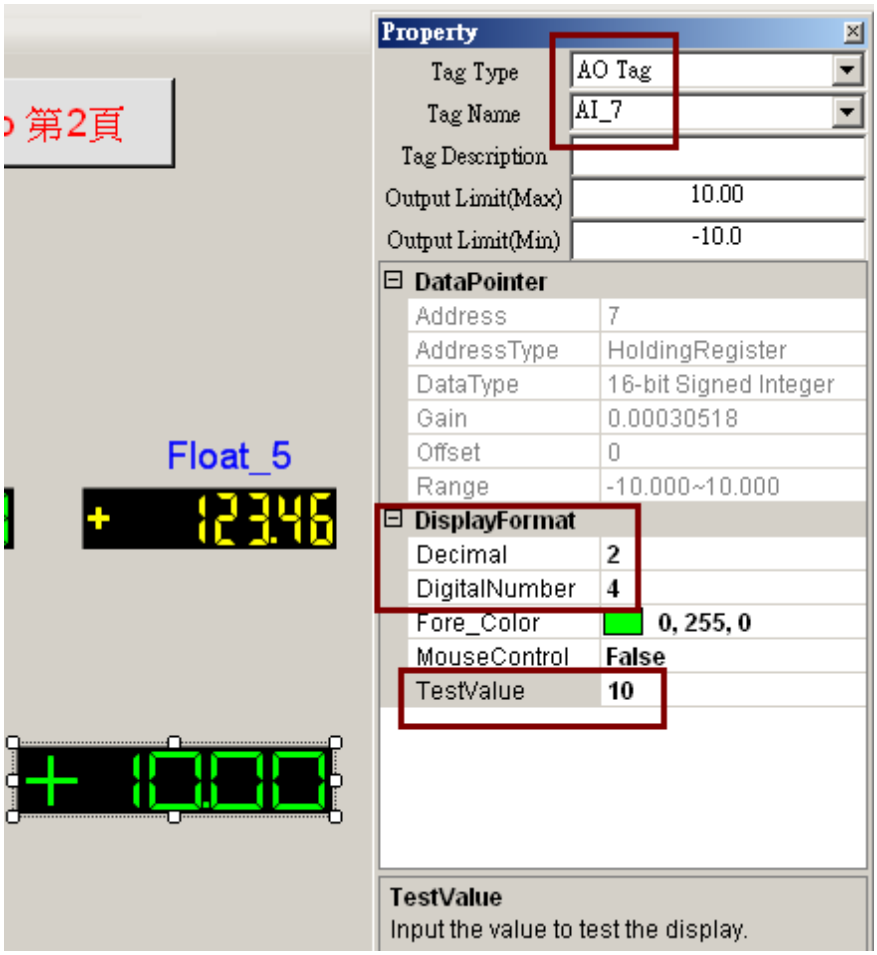
Gain = 0.0003051804 (Gain : even its value is very small, don't ignore it.)
Offset = 0.0001525902 (Offset : if its value is too small, we usually ignore it and set it to 0.)

So if the "AI_7" Tag is to read the value from one Channel of the I-8017HW, the Gain and the Offset can be set as below. (NOTE: Data_Type is a "16-bit Signed Integer".)



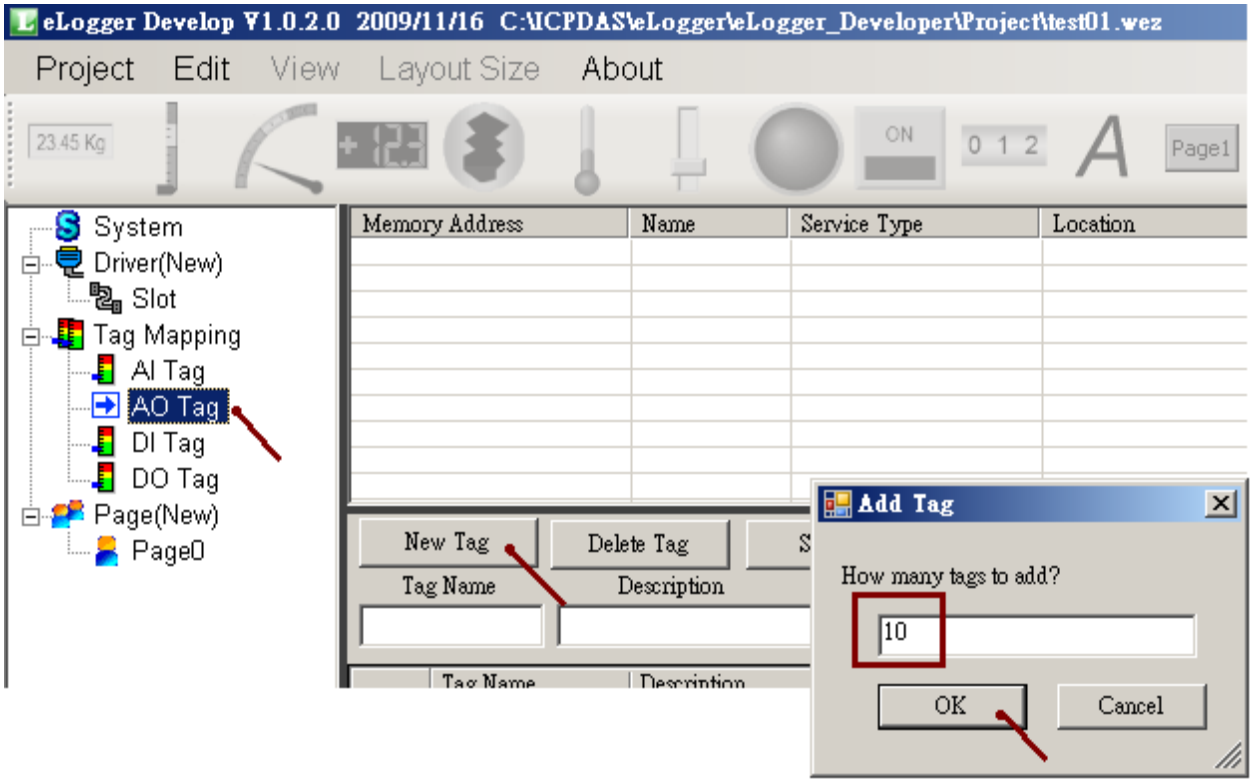
The converted value displayed in the eLogger project is showing in the next page.

In the Property window, please set “TestValue” to the maximum converted value while doing the test, 10 in this example.

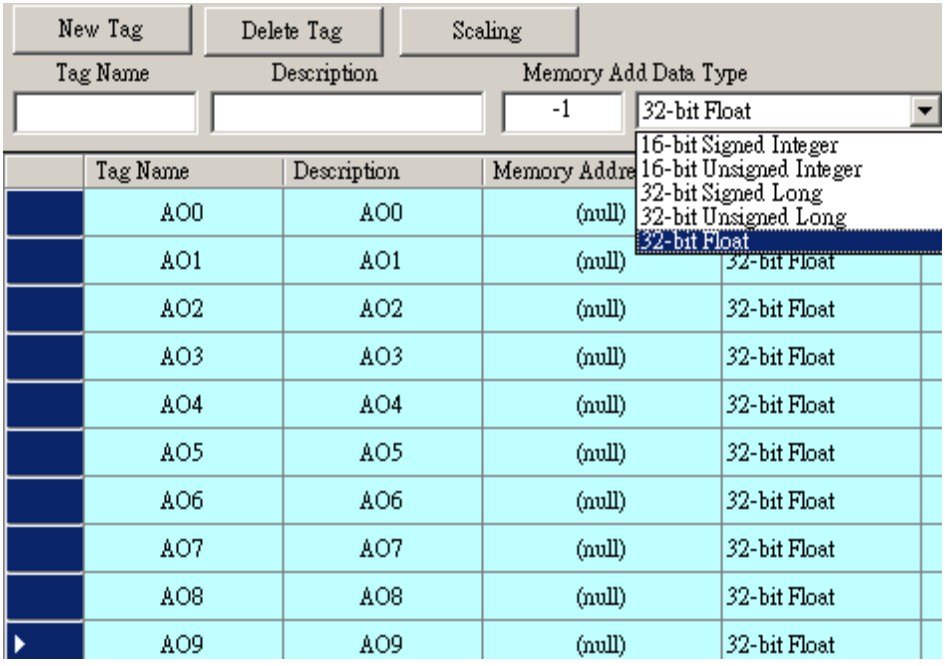


1.4.2: Declare Many eLogger Tags With Similar Names And Same Format

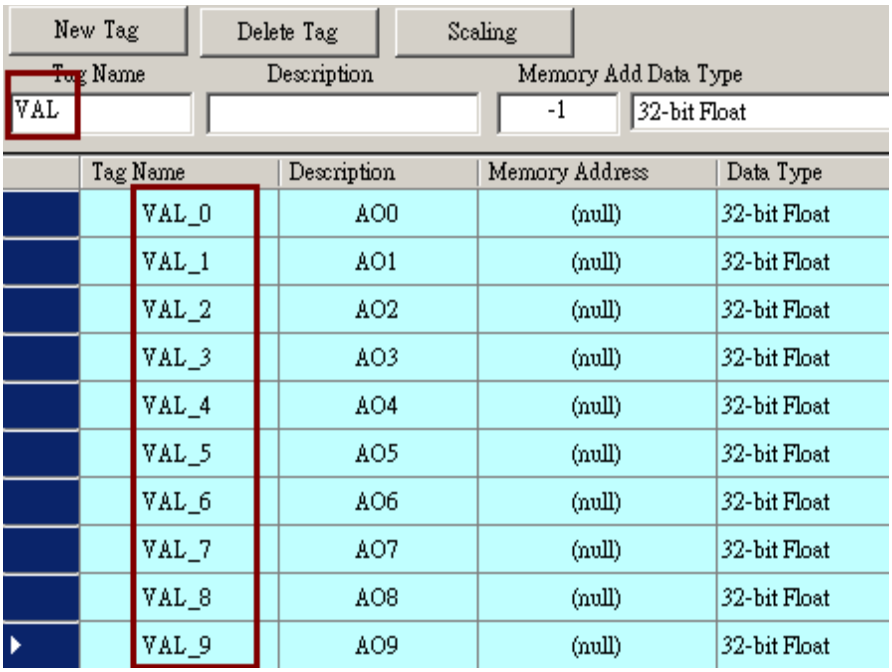
User may declare may eLogger Tags with similar names and same format as the following.



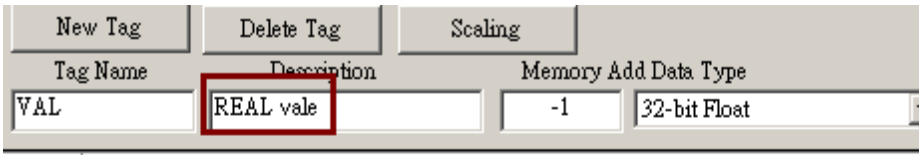
Use “Shift” key and Mouse to select these Tags, then set them to have the same format.



Then setting as similar Names and similar Descriptions.



Tag Name	Description	Memory Address	Data Type
VAL_0	AO0	(null)	32-bit Float
VAL_1	AO1	(null)	32-bit Float
VAL_2	AO2	(null)	32-bit Float
VAL_3	AO3	(null)	32-bit Float
VAL_4	AO4	(null)	32-bit Float
VAL_5	AO5	(null)	32-bit Float
VAL_6	AO6	(null)	32-bit Float
VAL_7	AO7	(null)	32-bit Float
VAL_8	AO8	(null)	32-bit Float
VAL_9	AO9	(null)	32-bit Float



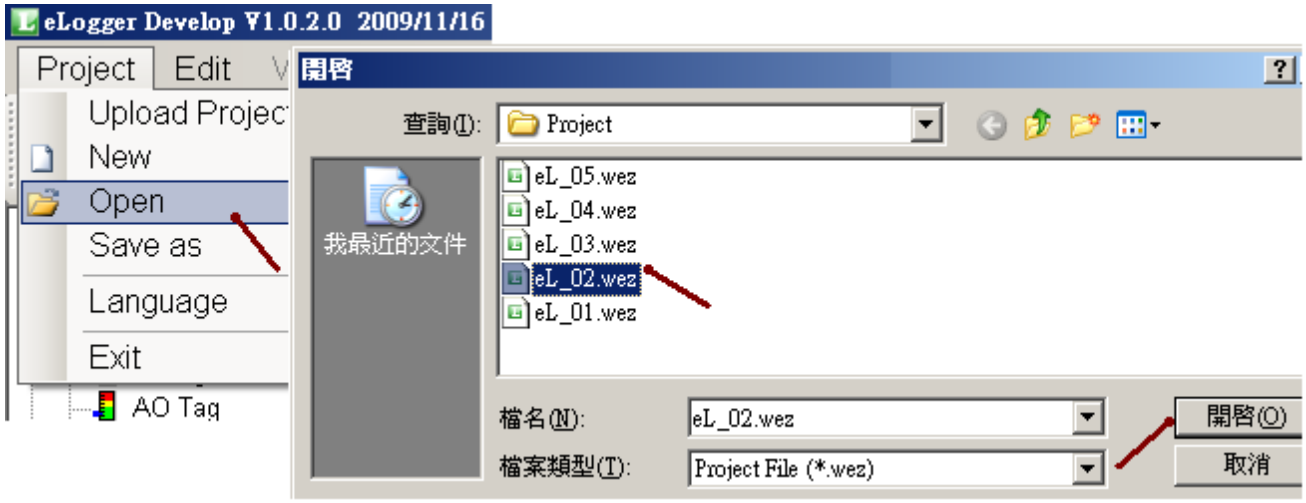
Tag Name	Description	Memory Address	Data Type
VAL_0	REAL vale_0	(null)	32-bit Float
VAL_1	REAL vale_1	(null)	32-bit Float
VAL_2	REAL vale_2	(null)	32-bit Float
VAL_3	REAL vale_3	(null)	32-bit Float
VAL_4	REAL vale_4	(null)	32-bit Float
VAL_5	REAL vale_5	(null)	32-bit Float
VAL_6	REAL vale_6	(null)	32-bit Float
VAL_7	REAL vale_7	(null)	32-bit Float
VAL_8	REAL vale_8	(null)	32-bit Float
VAL_9	REAL vale_9	(null)	32-bit Float

1.5: Some Useful eLogger + ISaGRAF Example Programs

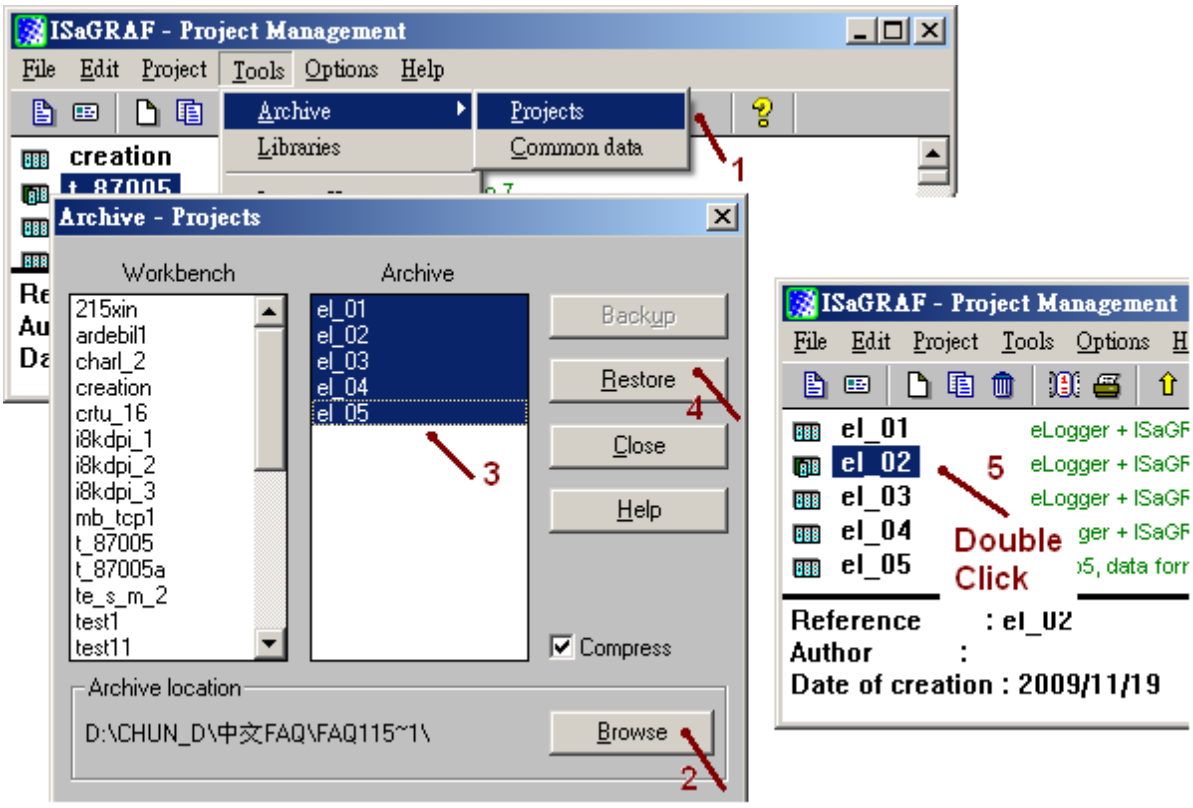
The example programs listed in this section can be downloaded at the following web site.

www.icpdas.com > FAQ > Software > ISaGRAF > FAQ-115

User may open the eLogger example programs for studying. First copy "eL_01.wez" ~ "eL_06.wez" to PC 's C:\ICPDAS\eLogger\eLogger_Developer\Project\ directory, then run Project > Open to open them.



User may open the ISaGRAF programs for studying. First restore "eL_01.pia" ~ "eL_06.pia" to PC/ ISaGRAF, then double click the project name to open it.

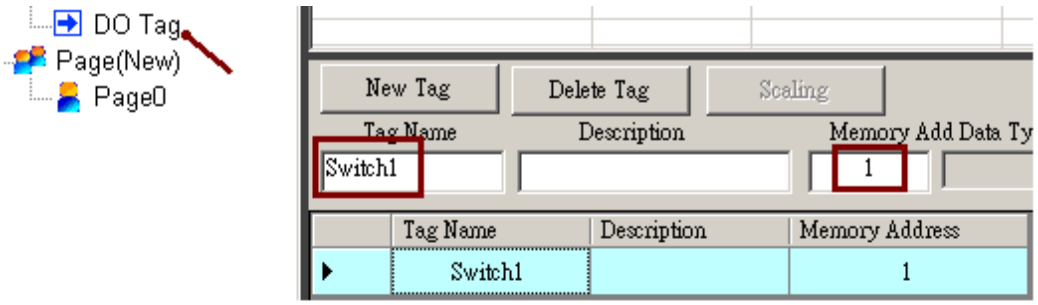


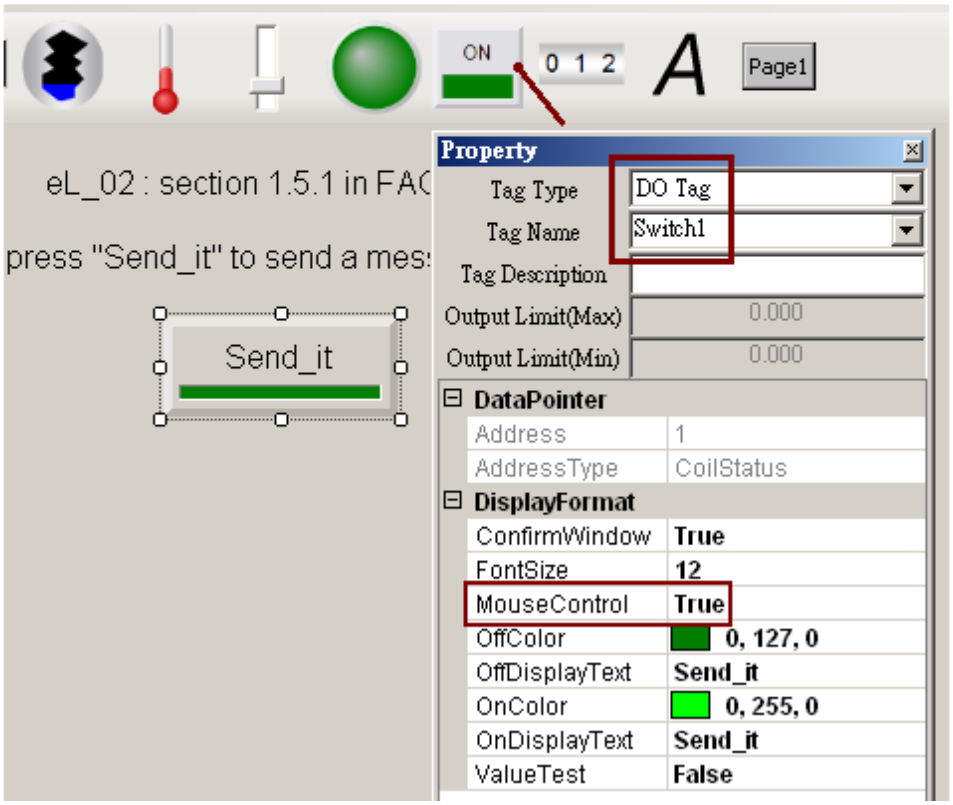
1.5.1: Using A Pulse ON Then OFF Button To Control Something

ISaGRAF project file : eL_02.pia
eLogger project file : eL_02.wez

This example setup an eLogger “Switch” . The ISaGRAF program will send a message which contains the PAC’s date and time from the WP-8xx7 ‘s COM1: RS-232 to PC when setting this “Switch” to ON. User may run one Hyper-Terminal utility in PC to open a RS-232 port (9600,8,N,1) to receive this message. Then the “Switch” state will reset to OFF automatically by the ISaGRAF program.

eLogger Developer:



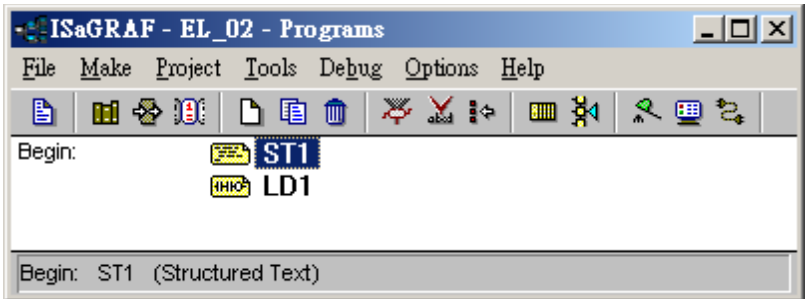


ISaGRAF :

Variables:

Name	Type	Attribution	NetWork Addr.	Description
Switch1	Boolean	Internal	1	To access to eLogger ‘s Switch
TMP	Boolean	Internal	0	
INIT	Boolean	Internal	0	Declare its initial value as TRUE
STR1	Message	Internal	0	Declare its max. Length as 64
Year1	Integer	Internal	0	To get PAC ‘ s year
Month1	Integer	Internal	0	To get PAC ‘ s month
Day1	Integer	Internal	0	To get PAC ‘ s day in month
WeekDay1	Integer	Internal	0	To get PAC ‘ s weekday
Hour1	Integer	Internal	0	To get PAC ‘ s hour
Minute1	Integer	Internal	0	To get PAC ‘ s minute
Second1	Integer	Internal	0	To get PAC ‘ s second

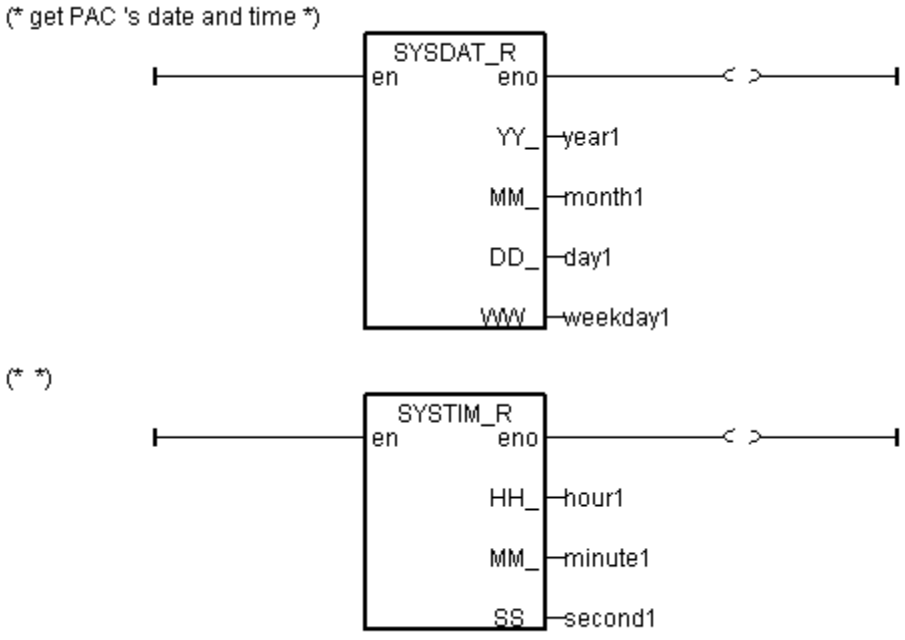
Project :



Program - ST1:

```
(* Open COM1 as 9600,8,N,1 at 1st scan cycle
   must declare "INIT" with an initail value at True *)
if  INIT  then
   INIT := False ;   (* only do it once in the 1st scan cycle *)
   TMP := COMOPEN( 1 , 9600 , 8 , 0 , 1 ) ;
end_if ;

(* If eLogger set "Switch1" as True , send message to COM1 *)
if  switch1  then

  (* Convert data and time to a string , like 'Feb/18/2010,13:25:45' *)
  str1 := time_str( year1 , month1 , day1 , weekday1 , hour1 , minute1 , second1 ) ;
  str1 := str1 + MSG('$0D$0A') ;  (* Add  <CR><LF> at the end of string *)

  (* send one message to COM1 *)
  TMP := COMSTR_W( 1 , str1 ) ;

  switch1 := False ;  (* reset it when finished *)

end_if ;
```

Program - LD1:

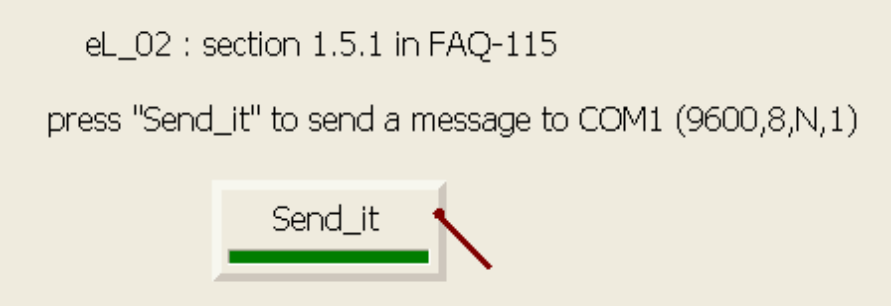


How to test ? :

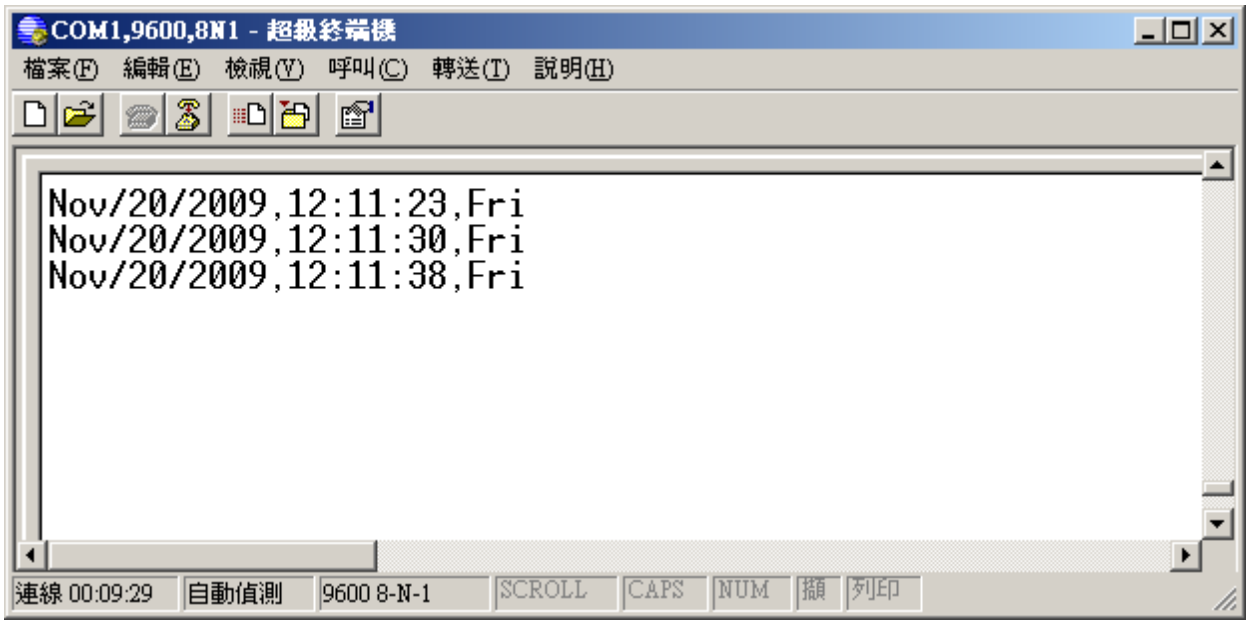
Connect one RS-232 Cable from the WP-8xx7 ‘s COM1 to one RS-232 port of PC (refer to Appendix A.5 of the WP-8xx7 Getting Started Manual for COM1 ‘s pin assignment). Then power up the WP-8xx7 and run the Hyper-Terminal utility in PC to open that RS-232 port as 9600, 8, N, 1.

WP-8xx7 - eLogger HMI :

You may click the “Send_it” button and set as ON to send one message to PC .



PC – Hyper Terminal (9600, 8, N, 1) :



1.5.2: Read Or Save Process Parameters From/To A File. The File Contains Two Rows, Each has Ten REAL Values.

ISaGRAF project file : eL_03.pia
eLogger project file : eL_03.wez

There is one another “eL_04” project similar to this “eL_03”. However its data format is integer.
ISaGRAF project file : eL_04.pia
eLogger project file : eL_04.wez

There is one another “eL_05” project similar to this “eL_03”. Its data format is also REAL. However it can read or save these process parameters from/to several files.
ISaGRAF project file : eL_05.pia
eLogger project file : eL_05.wez

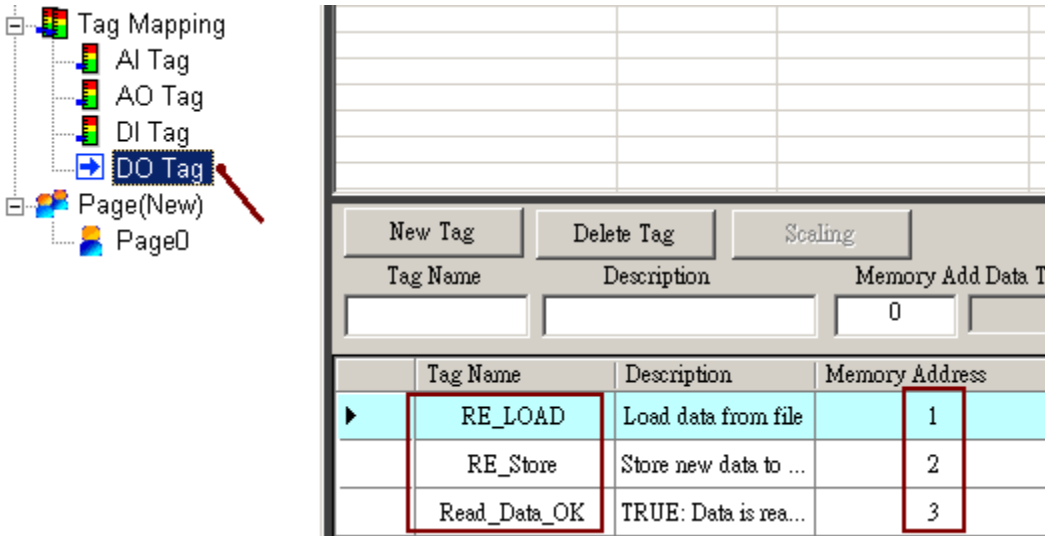
This example provides 20 process parameters for controlling the process. These parameter values will be stored in the “Working_Real.txt” file in the WP-8xx7 ‘s \System_Disk\ directory. The “Working_Real.txt” is a pure text file. It contains 2 rows, each has 10 REAL values. It looks like as the following.

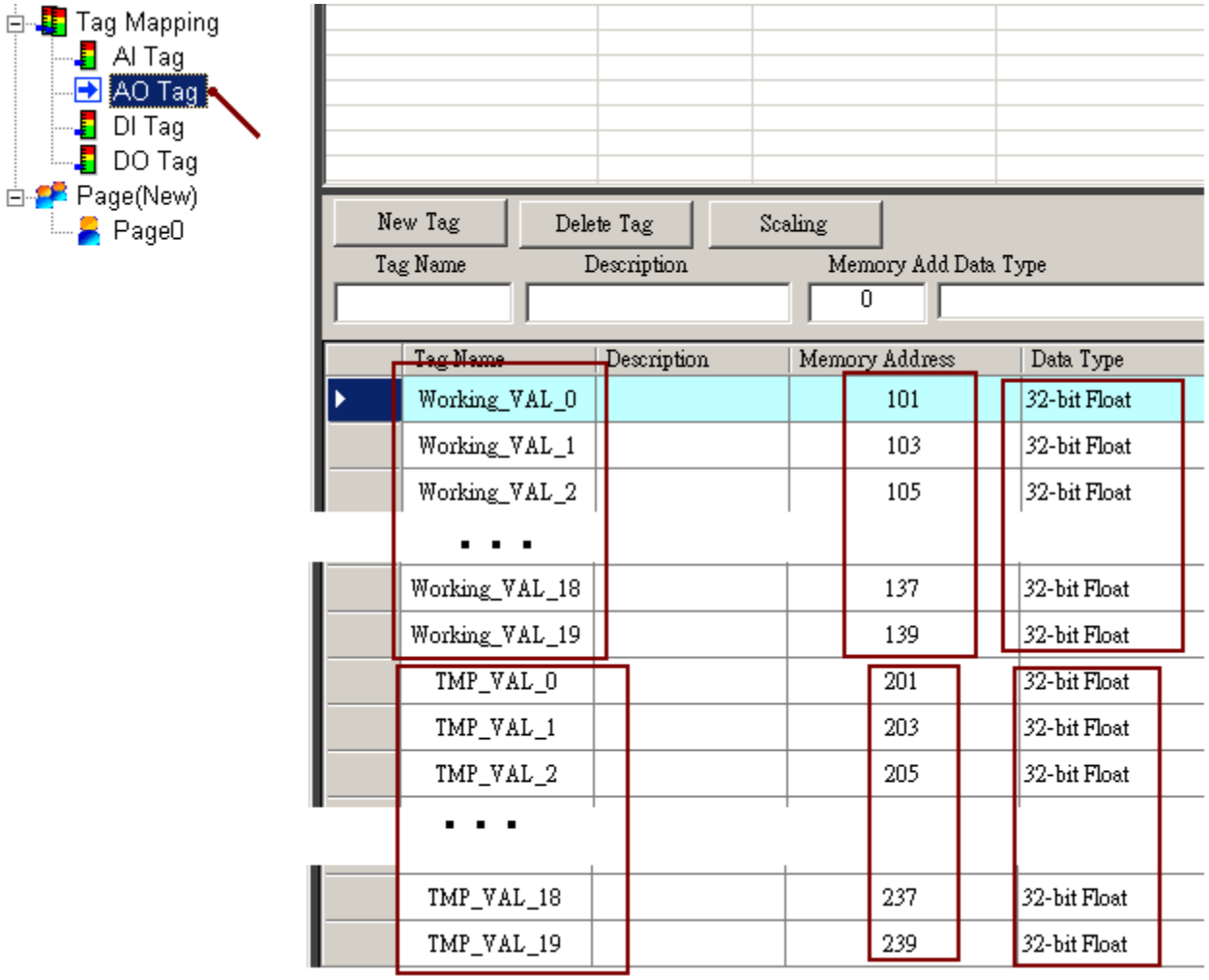
23, 65.9, 0.12, 5.87, 88.2, 0.34, 8.5, -2.08, 4.08, 5.32
2, -7, 6666.8, 456.07, 1.01, 5, 6, 7, 8, 9

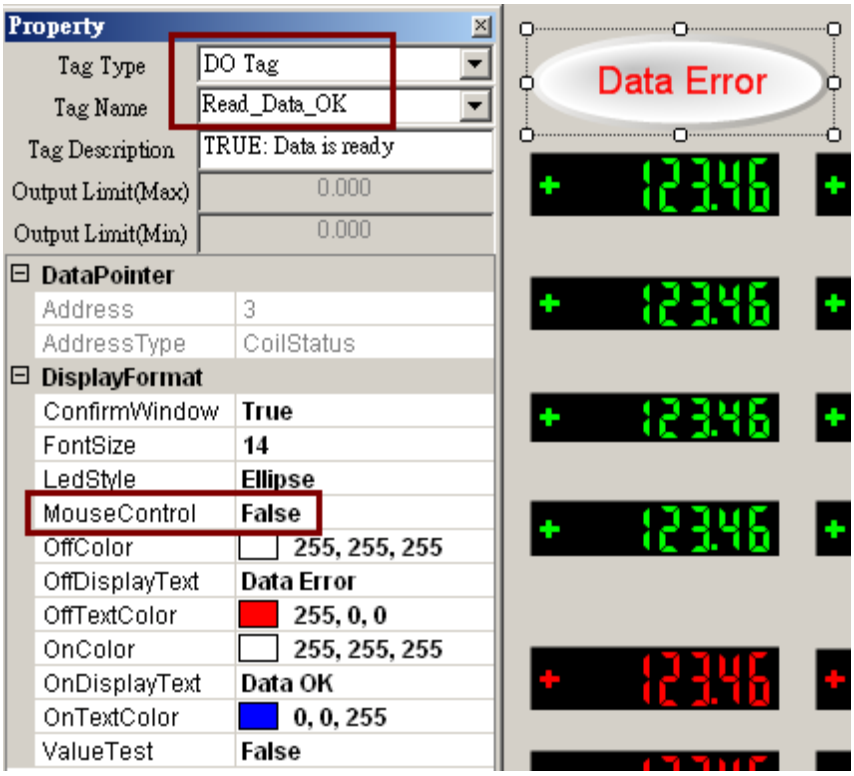
The eLogger HMI picture has 20 “Seven Segment” items on the top to display the current working parameters (Their addresses are 101, 103, 105, ..., 139, the “MouseContrl” properties are set as FALSE for read-only). And there are other 20 “Seven Segment” items to display and for entering new process parameters (Their addresses are 201, 203, 205, ..., 239, the “MouseContrl” properties are set as TRUE for reading and writting).

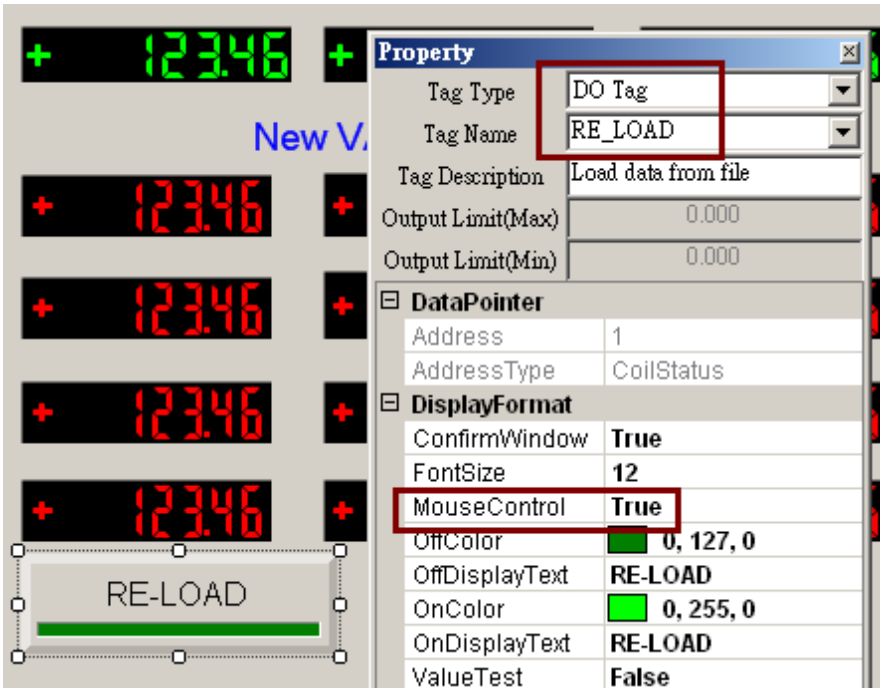
There are 3 “Switch” items to re-load parameters and re-store parameters and display data is OK or not. Their addresses are 1, 2, 3 respectively .

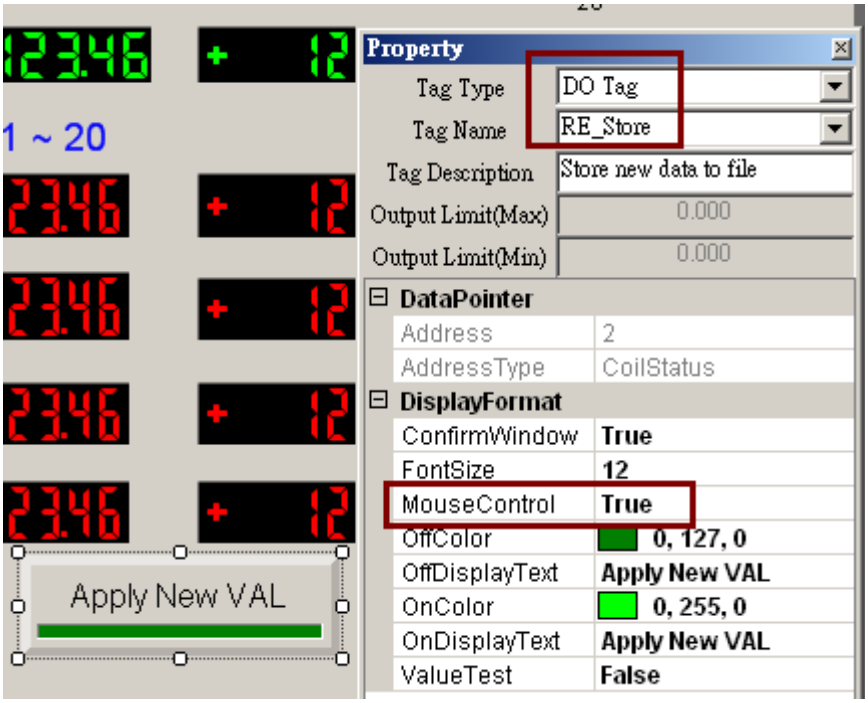
eLogger Developer:

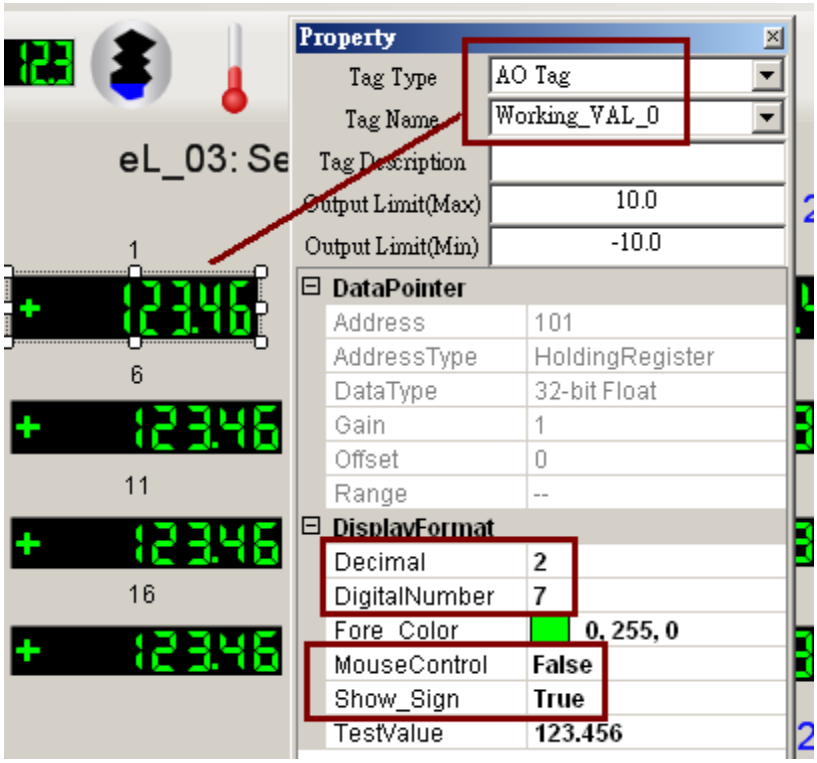


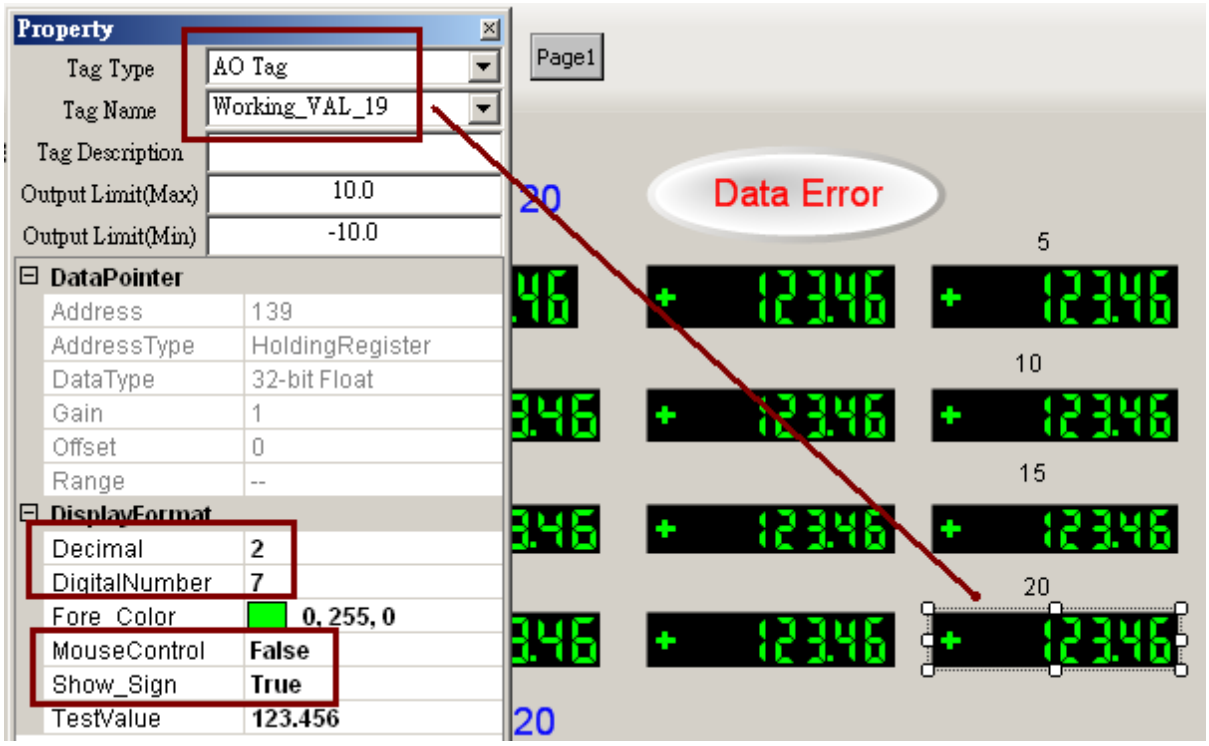

Tag Mapping
AI Tag
AO Tag
DI Tag
DO Tag
Page(New)
Page0
New Tag
Delete Tag
Scaling
Tag Name
Description
Memory Add
Data Type
0
Tag Name
Description
Memory Address
Data Type
Working_VAL_0
101
32-bit Float
Working_VAL_1
103
32-bit Float
Working_VAL_2
105
32-bit Float
Working_VAL_18
137
32-bit Float
Working_VAL_19
139
32-bit Float
TMP_VAL_0
201
32-bit Float
TMP_VAL_1
203
32-bit Float
TMP_VAL_2
205
32-bit Float
TMP_VAL_18
237
32-bit Float
TMP_VAL_19
239
32-bit Float

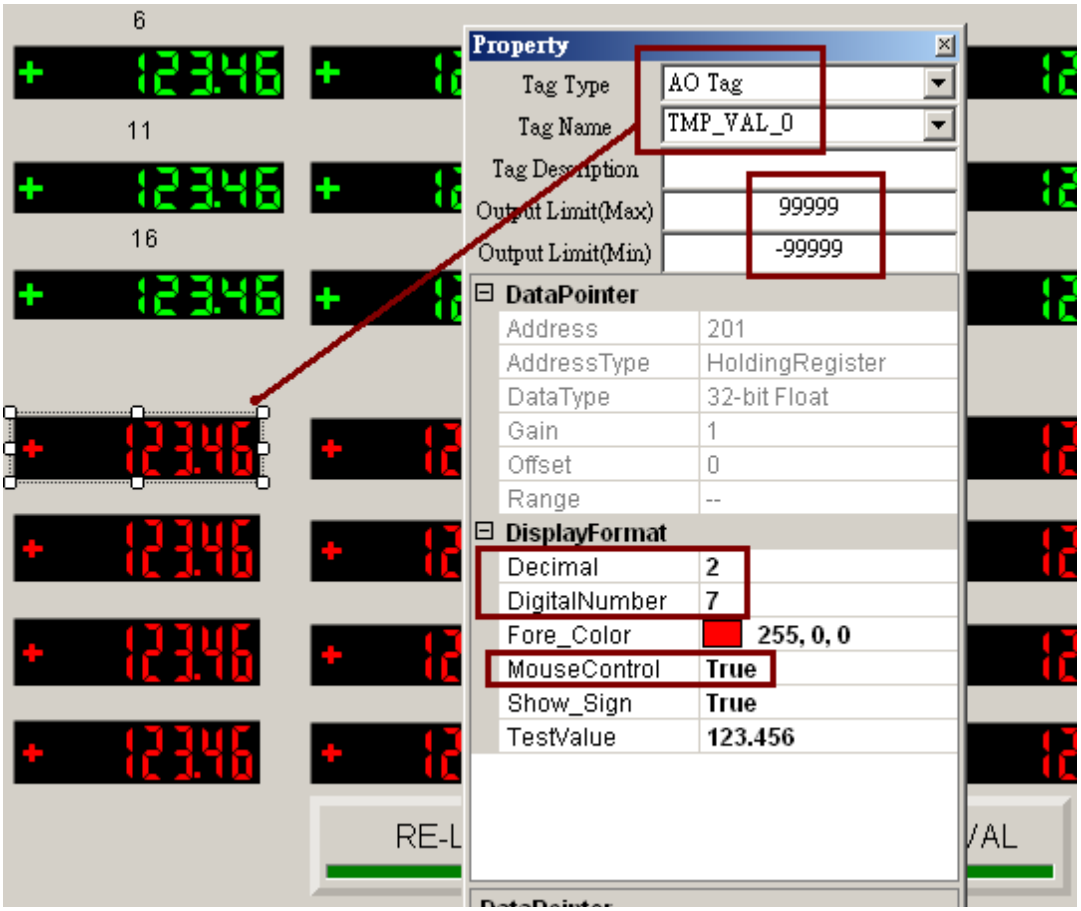

Property
Tag Type
DO Tag
Tag Name
Read_Data_OK
Tag Description
TRUE: Data is ready
Output Limit(Max)
0.000
Output Limit(Min)
0.000
DataPointer
Address
3
AddressType
CoilStatus
DisplayFormat
ConfirmWindow
True
FontSize
14
LedStyle
Ellipse
MouseControl
False
OffColor
255, 255, 255
OffDisplayText
Data Error
OffTextColor
255, 0, 0
OnColor
255, 255, 255
OnDisplayText
Data OK
OnTextColor
0, 0, 255
ValueTest
False
Data Error
123.46

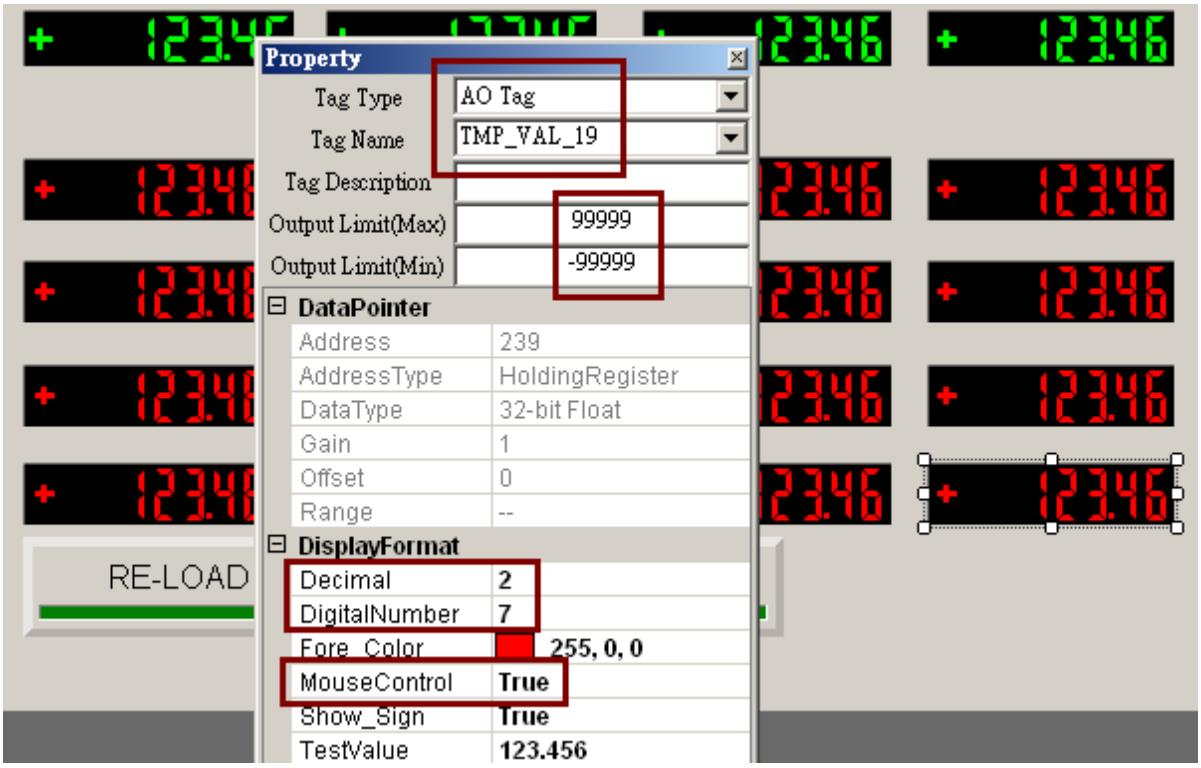

123.46
Property
Tag Type
DO Tag
Tag Name
RE_LOAD
Tag Description
Load data from file
Output Limit(Max)
0.000
Output Limit(Min)
0.000
DataPointer
Address
1
AddressType
CoilStatus
DisplayFormat
ConfirmWindow
True
FontSize
12
MouseControl
True
OffColor
0, 127, 0
OffDisplayText
RE-LOAD
OnColor
0, 255, 0
OnDisplayText
RE-LOAD
ValueTest
False
RE-LOAD


1 ~ 20
Apply New VAL
Property
Tag Type
DO Tag
Tag Name
RE_Store
Tag Description
Store new data to file
Output Limit(Max)
0.000
Output Limit(Min)
0.000
DataPointer
Address
2
AddressType
CoilStatus
DisplayFormat
ConfirmWindow
True
FontSize
12
MouseControl
True
OffColor
0, 127, 0
OffDisplayText
Apply New VAL
OnColor
0, 255, 0
OnDisplayText
Apply New VAL
ValueTest
False


Property
Tag Type
AO Tag
Tag Name
Working_VAL_0
Tag Description
Output Limit(Max)
10.0
Output Limit(Min)
-10.0
DataPointer
Address
101
AddressType
HoldingRegister
DataType
32-bit Float
Gain
1
Offset
0
Range
--
DisplayFormat
Decimal
2
DigitalNumber
7
Fore_Color
0, 255, 0
MouseControl
False
Show_Sign
True
TestValue
123.456


Property
Tag Type
AO Tag
Tag Name
Working_VAL_19
Tag Description
Output Limit(Max)
10.0
Output Limit(Min)
-10.0
DataPointer
Address
139
AddressType
HoldingRegister
DataType
32-bit Float
Gain
1
Offset
0
Range
--
DisplayFormat
Decimal
2
DigitalNumber
7
Fore_Color
0, 255, 0
MouseControl
False
Show_Sign
True
TestValue
123.456
Page1
Data Error


Property
Tag Type
AO Tag
Tag Name
TMP_VAL_0
Tag Description
Output Limit(Max)
99999
Output Limit(Min)
-99999
DataPointer
Address
201
AddressType
HoldingRegister
DataType
32-bit Float
Gain
1
Offset
0
Range
--
DisplayFormat
Decimal
2
DigitalNumber
7
Fore_Color
255, 0, 0
MouseControl
True
Show_Sign
True
TestValue
123.456

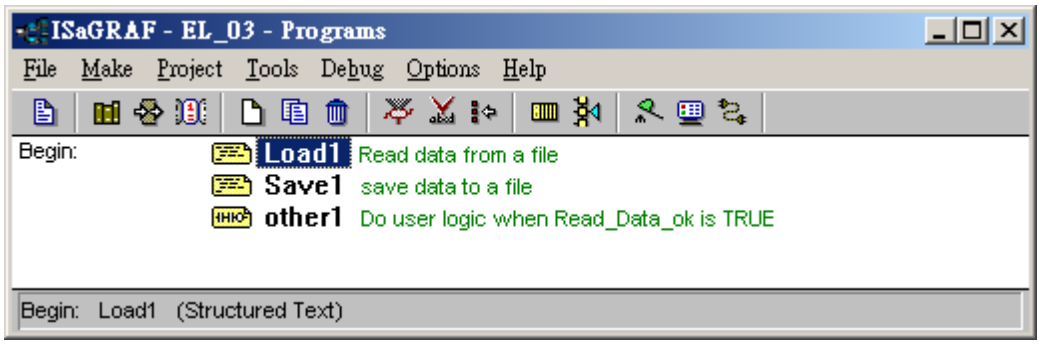

Property
Tag Type
AO Tag
Tag Name
TMP_VAL_19
Tag Description
Output Limit(Max)
99999
Output Limit(Min)
-99999
DataPointer
Address
239
AddressType
HoldingRegister
DataType
32-bit Float
Gain
1
Offset
0
Range
--
DisplayFormat
Decimal
2
DigitalNumber
7
Fore_Color
255, 0, 0
MouseControl
True
Show_Sign
True
TestValue
123.456
RE-LOAD

ISaGRAF :

Variables: (Please refer to the FAQ-039 to get more information about the ISaGRAF variable array.)

Name	Type	Attrib.	NetW. Addr.	Description
TMP	Boolean	Internal	0	
RE_LOAD	Boolean	Internal	1	Set as TRUE to re-load a File
RE_Store	Boolean	Internal	2	Set as TRUE to store a File
INIT	Boolean	Internal	0	Declare its initial value as TRUE
Read_Data_OK	Boolean	Internal	3	True: data is ok, False: Data error.
Msg1	Message	Internal	0	Declare max. len as 128
str1	Message	Internal	0	Declare max. len as 255
TMP_file_name	Message	Internal	0	Declare max. len as 128
File1	Integer	Internal	0	
Working_ReaL_VAL[0..19]	REAL	Internal	101	The 20 process working parameters. variable array, Dim = 20 Declare its 1st Addr. as 101
TMP_Real_VAL[0..19]	REAL	Internal	201	variable array, Dim = 20. Declare its 1st Addr. as 201
NUM1	Integer	Internal	0	
ii	Integer	Internal	0	Index of “for” loops
jj	Integer	Internal	0	Index of “for” loops
Data_Cnt	Integer	Internal	0	
Row_Cnt	Integer	Internal	0	The row amount, set as 2
Data_Cnt_in_Row	Integer	Internal	0	The data amount in each row, Set as 10

Project :



Program – Load1 :

```
(* Operations in the first PLC Scan *)
if  INIT  then

  (* Assign Network addr for Array Variable *)
  (* assign Working_Real_VAL[ ] as 101 , 103, 105, ..., 139 *)
  TMP := S_MB_ADR( 101 , 20 , 1 ) ;
 (* assign TMP_Real_VAL[ ] as 201 , 203, 205, ..., 239 *)
  TMP := S_MB_ADR( 201 , 20 , 1 ) ;

  (* file name which stores the current working parameters *)
  TMP_file_name := '\System_Disk\Working_Real.txt'  ;
  RE_LOAD := True ;   (* trigger to read data from file when PLC is started *)

end_if ;

(* if RE_LOAD is TRUE, Open one file and read it *)
if  RE_LOAD  then

  RE_LOAD  :=  FALSE  ;    (* Reset as FALSE *)
  Read_Data_ok := False ;  (* set data error at the beginning *)
  Data_Cnt := 0 ;  (* set as no data read at the beginning *)

  File1 := f_wopen( TMP_file_name )  ;   (* Try to open file in Read & Write mode *)

  if  File1 = 0 then

    (* Open error, exit this ST program to run next program *)
    Msg1  :=  'Can not Open file ' + TMP_file_name ;
    INIT := False ;  (* reset it before exit this ST program *)
    return ;

  end_if  ;

  (* max 2 rows to read these 20 REAL values, Each row in the file contains 10 REAL values *)
  for  ii  :=  0  to  ( Row_Cnt - 1 )  do

     if  f_eof(File1) = TRUE  then   (* test if End_Of_File reached *)

        (* Reach End Of File *)
 Msg1  :=  'There should be at least ' + MSG(Row_Cnt)+ ' rows in ' +TMP_file_name+ ' !!!' ;
        exit ;   (* exit for loop *)

     end_if ;
```

```
   str1 := fm_read(File1) ;   (* Read one row as String (message) *)

  (* Convert this string to some REAL values and store them into No.1 Float array *)
   NUM1 := Msg_F( str1 , 1 );

   if  NUM1 <> Data_Cnt_in_Row  then  (* there should be 10 REAL values in each row *)

(* error, it means the message is not correct REAL values or data number is not enough *)
      Msg1  :=  'The data format of No.' + Msg( ii + 1 ) + ' row is not correct or data number is
not ' + MSG( Data_Cnt_in_Row ) ;
      exit ;   (* exit for loop *)

   end_if;
   Data_Cnt := Data_Cnt + Data_Cnt_in_Row;  (* ok ! Data_Cnt plus Data_Cnt_in_Row *)

   (* conversion Ok, store these 10 REAL values to TMP_Real_VAL[ ] *)
   for jj := 0 to ( Data_Cnt_in_Row - 1 ) do
     (* Get these converted REAL values from No.1 Float array 's addr. 1 to 10 *)
     TMP_ReaL_VAL[ Data_Cnt_in_Row * ii + jj ] := ARY_F_R( 1 , jj + 1 ) ;
   end_for ;

 end_for ;

 (* Any file been open should be closed by f_close( ) *)
 TMP := f_close(File1) ;

 (* All data are read Ok *)
 if  ( Data_Cnt = Row_Cnt * Data_Cnt_in_Row )  then
   Msg1 := 'Read ' + TMP_file_name + ' Ok ' ;
   Read_Data_Ok := True ;  (* set as read OK *)

   (* if this scan is the first scan, store TMP_Real_Val[] to Working_ReaL_Val[] *)
   if  INIT  then
     for  ii := 0  to  ( Data_Cnt - 1 )  do
       Working_ReaL_VAL[ii] := TMP_ReaL_VAL[ii] ;
     end_for ;
   end_if ;

 end_if ;

end_if  ;

INIT := False ;  (* No more first PLC scan *)
```

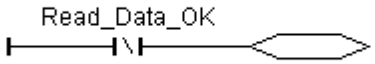
Program – Save1 :

```
(* Store data to a file when RE_Store is True *)
(* This "RE_Store" can be triggered by eLogger HMI *)
if  RE_Store  then

  RE_Store := False ;  (* reset it *)
  Read_Data_ok := False ;  (* set data error at the beginning *)
  Data_Cnt := 0 ;  (* set as no data read at the beginning *)

  TMP := f_delete( TMP_file_name )  ;   (* delete file first and then create and write it *)

 (* creat a new file *)
  File1 := f_creat( TMP_file_name ) ;

 (* Creat failed, exit this ST program *)
  if  File1 = 0  then
    MSG1 := 'Can not Create a new file - ' + TMP_file_name + ' !' ;
    return ;   (* exit this ST program *)
  end_if ;

 (* max 2 rows to save these 20 REAL values, Each row in the file contains 10 REAL values *)
 for  ii  :=  0  to  ( Row_Cnt - 1 )  do

    str1 := '' ;   (* set as an empty string at the beginning *)

    (* There is 10 data in one row *)
    for  jj := 0  to  ( Data_Cnt_in_Row - 2 )  do
      str1 := str1 + Real_Str( TMP_ReaL_VAL[Data_Cnt_in_Row * ii + jj] ) + ', ' ;
    end_for ;
    (* add last data of this row to str1 *)
    str1 := str1 + Real_Str( TMP_ReaL_VAL[Data_Cnt_in_Row * ii + jj] ) + '$0D$0A' ;

    (* write this row to file *)
    TMP := F_writ_S( File1 , str1 ) ;
    if  TMP = False  then
       MSG1 := 'Write data to file - ' + TMP_file_name + ' failed !' ;
       exit ;   (* exit this for loops *)
    end_if ;
    Data_Cnt := Data_Cnt + Data_Cnt_in_Row;  (* ok ! Data_Cnt plus Data_Cnt_in_Row *)

 end_for ;
```

```
 TMP := f_close(File1) ;   (* Any file been open should be closed by f_close() *)

 (* All data are saved Ok *)
 if  ( Data_Cnt = Row_Cnt * Data_Cnt_in_Row )  then
    Msg1 := 'Write ' + TMP_file_name + ' Ok ' ;
    Read_Data_ok := True ;  (* set data Ok *)

    (* Store TMP_Real_Val[] to Working_ReaL_Val[] *)
    for  ii := 0  to  ( Data_Cnt - 1 )  do
      Working_ReaL_VAL[ii] := TMP_ReaL_VAL[ii] ;
    end_for ;

 end_if ;

end_if ;
```

Program - Other1 :

(* if Read_Data_OK is False, that means the process parameter "Working_ReaL_VAL[ ]" is not ready. Then return. *)

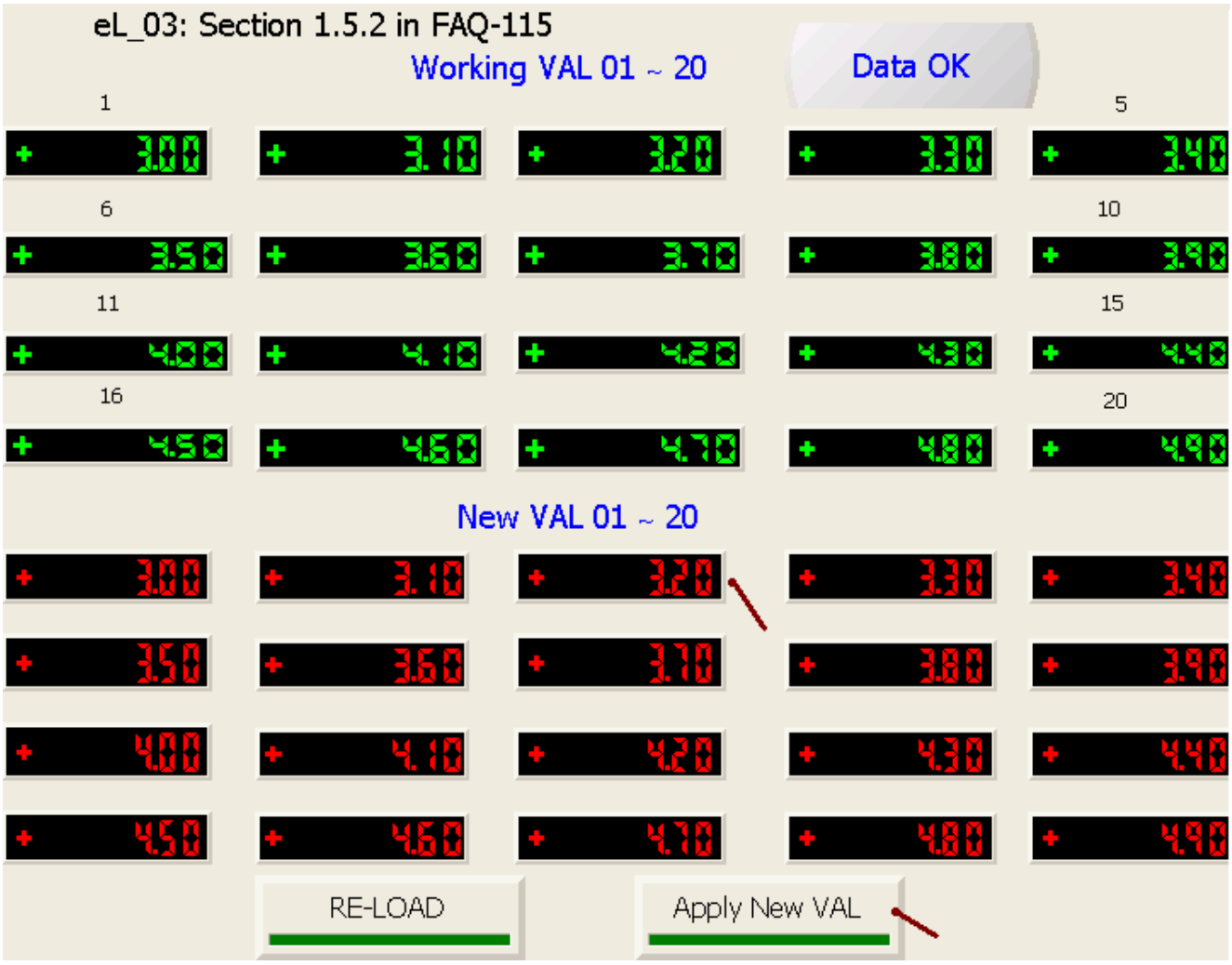


(* If Read_Data_OK is True, then you can use "Working_ReaL_VAL[ ]" in your Logic below. ... *)

How to test ?

User may click some “Seven Segment” items in the “NEW VAL” area to enter some different values. Then click the “Apply New VAL” and set as ON to apply the new value to become the working parameters. You will see value be changed in the “Working VAL” area. If connecting PC / ISaGRAF to the WP-8xx7 controller, you will find the related value of the ISaGRAF variable be changed too.

Then reset the WP-8xx7 controller once. You will also find the parameter values in the “Working VAL” area now are the new values after re-start the controller .

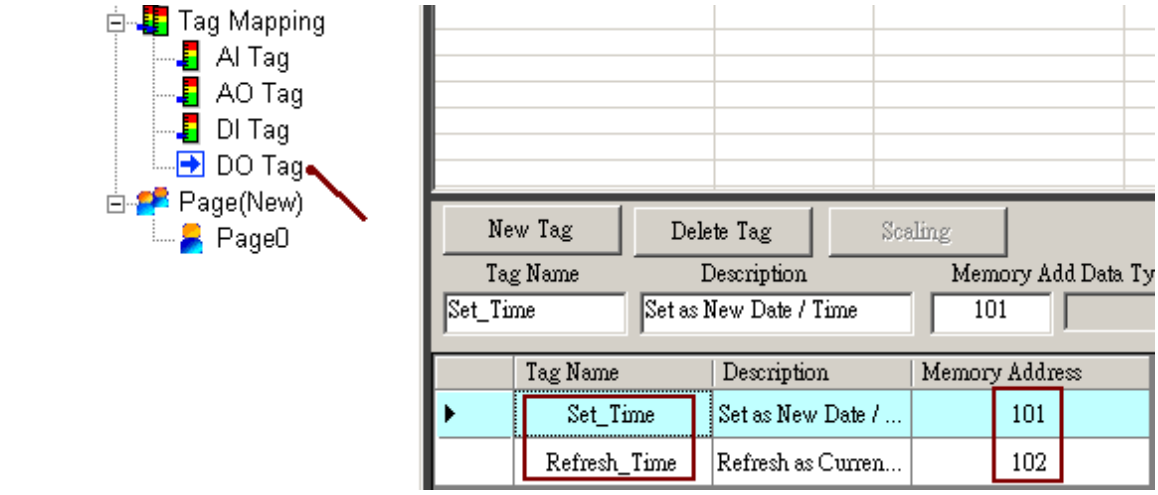


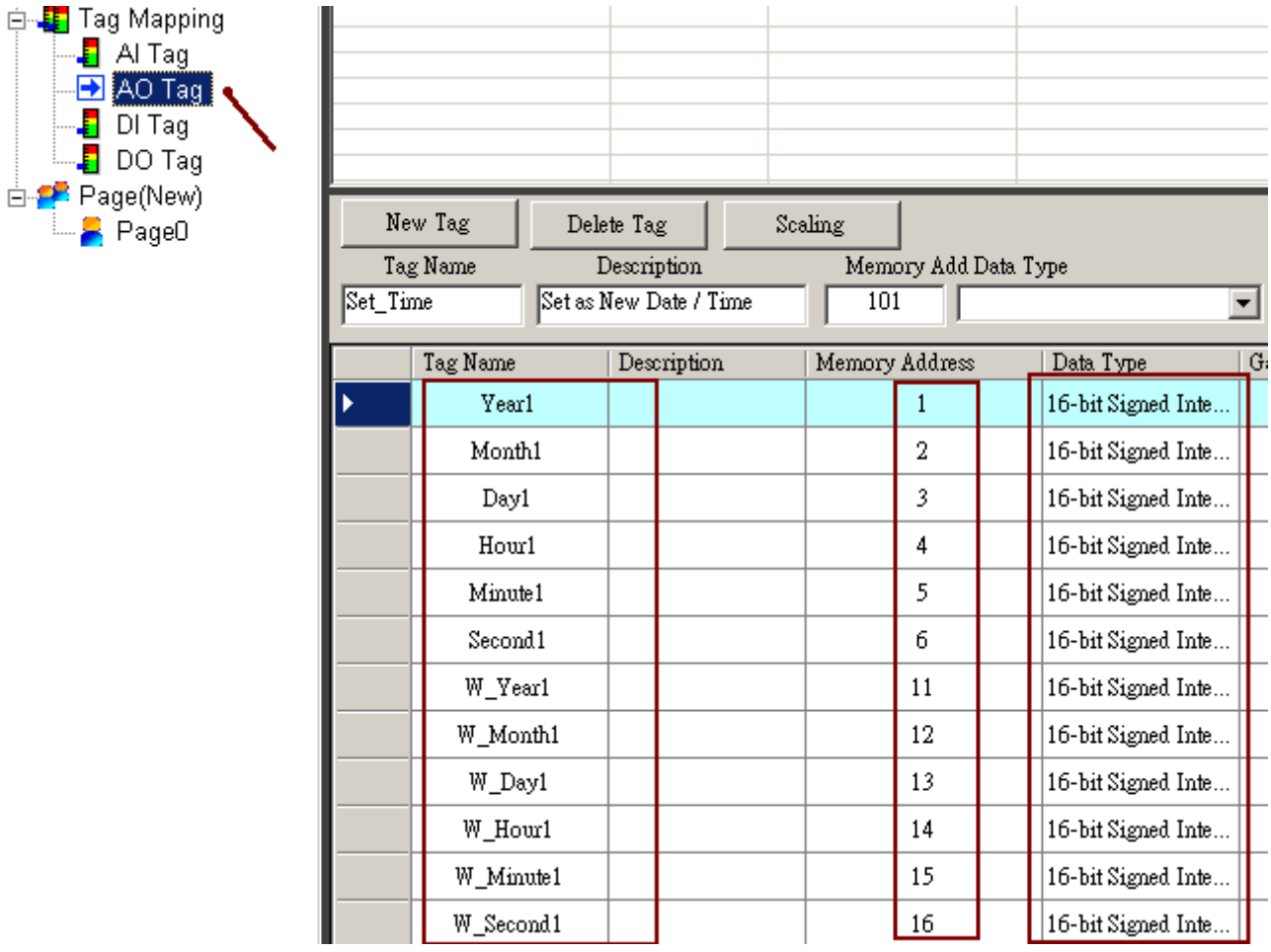
1.5.3: Read And Modify The Date / Time And Doing Some Control

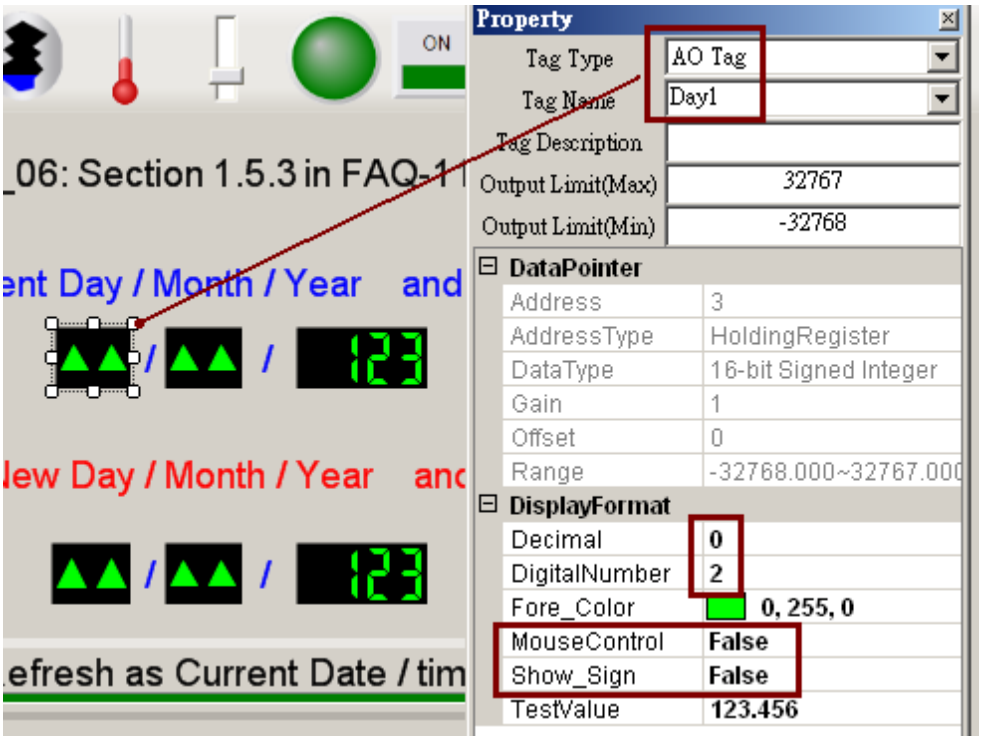
ISaGRAF project file : eL_06.pia
eLogger project file : eL_06.wez

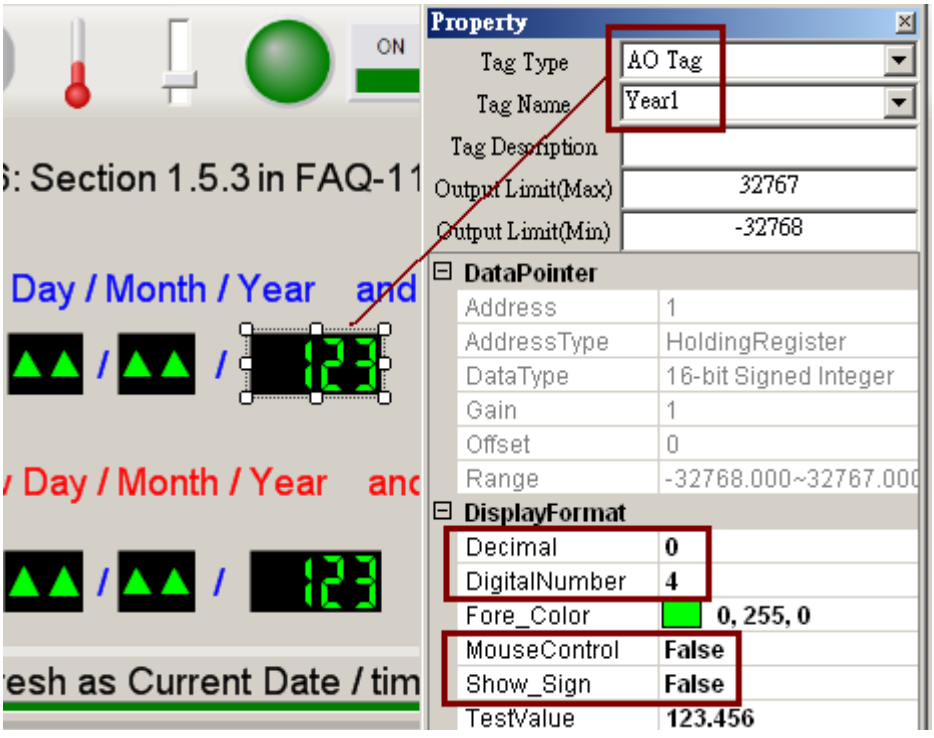
This example can read and then disply the current date and time of the controller, and can modify it to be a new date and time. This example also outputs “OUT1” as ON at 09:00 to 18:00 on Monday to Friday, while outputs “OUT2” as ON at 09:00 to 12:00 on saturday and Sunday.

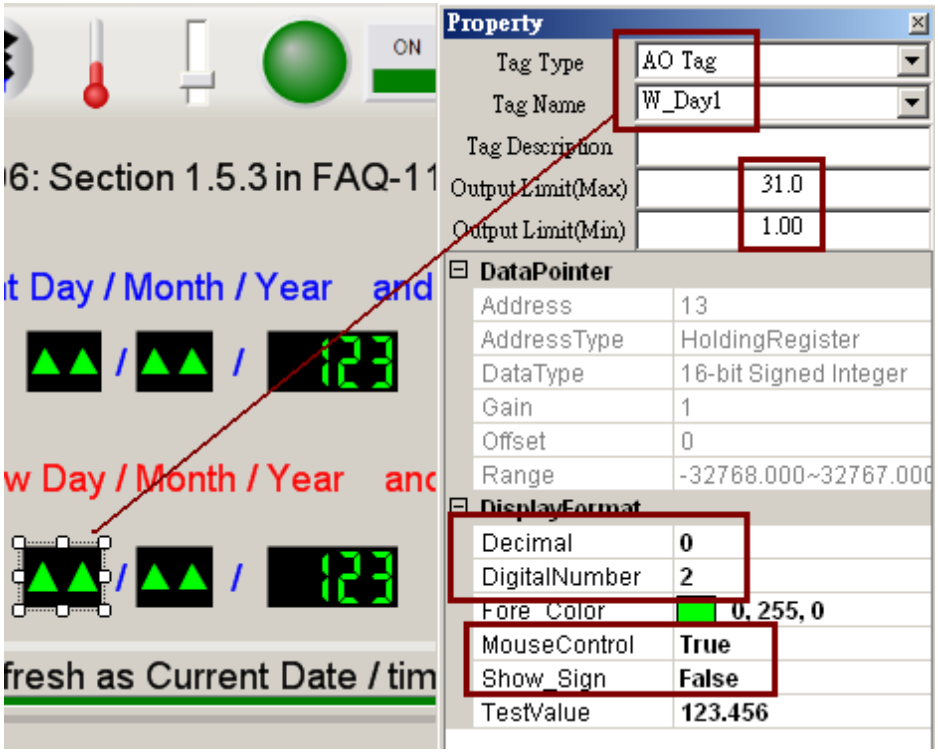
eLogger Developer:

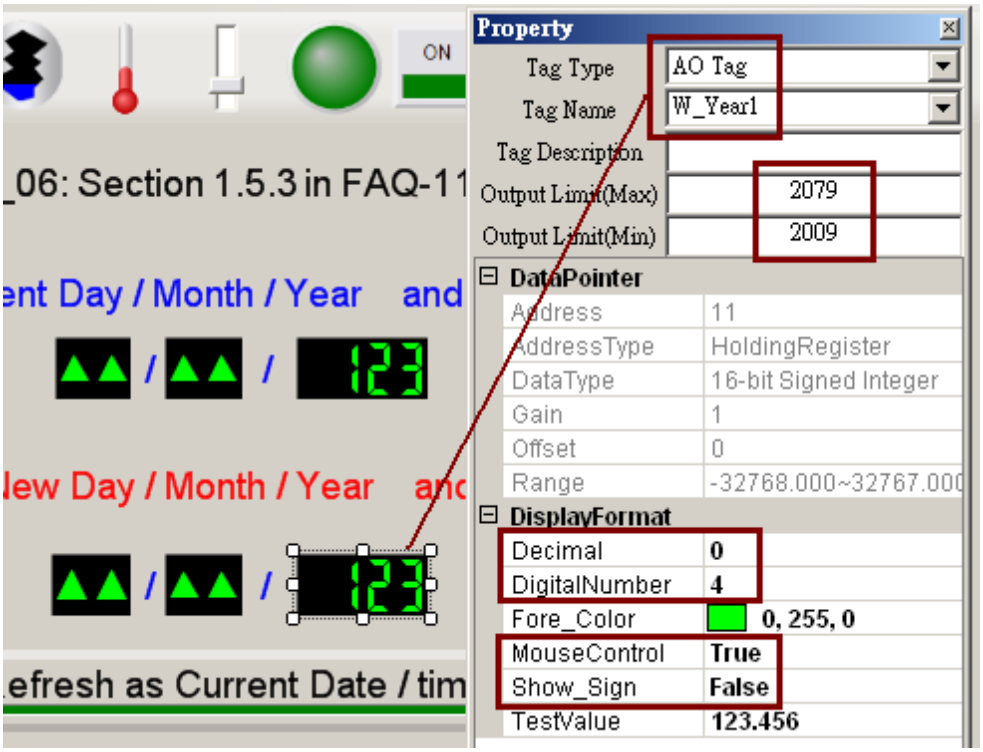


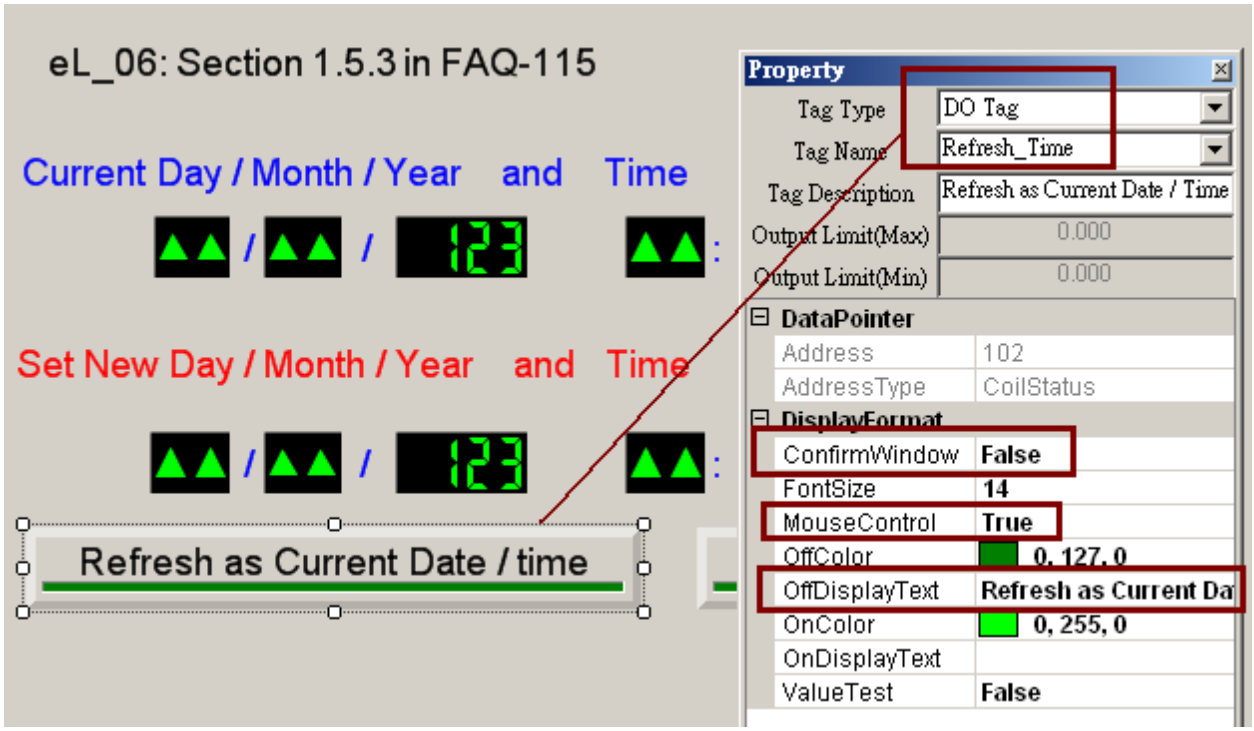


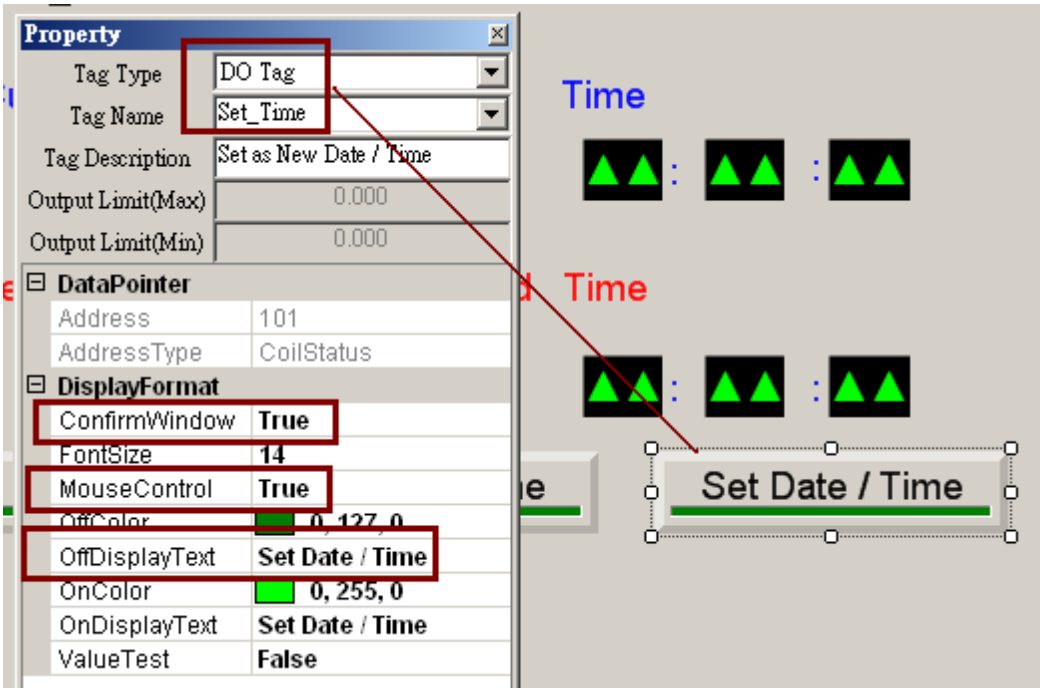

ON
_06: Section 1.5.3 in FAQ-1
ent Day / Month / Year and
/ /
123
lew Day / Month / Year anc
/ /
123
efresh as Current Date / tim
Property
Tag Type
AO Tag
Tag Name
Day1
Tag Description
Output Limit(Max)
32767
Output Limit(Min)
-32768
DataPointer
Address
3
AddressType
HoldingRegister
DataType
16-bit Signed Integer
Gain
1
Offset
0
Range
-32768.000~32767.000
DisplayFormat
Decimal
0
DigitalNumber
2
Fore_Color
0, 255, 0
MouseControl
False
Show_Sign
False
TestValue
123.456


ON
: Section 1.5.3 in FAQ-11
Day / Month / Year and
/ /
123
Day / Month / Year anc
/ /
123
esh as Current Date / tim
Property
Tag Type
AO Tag
Tag Name
Year1
Tag Description
Output Limit(Max)
32767
Output Limit(Min)
-32768
DataPointer
Address
1
AddressType
HoldingRegister
DataType
16-bit Signed Integer
Gain
1
Offset
0
Range
-32768.000~32767.000
DisplayFormat
Decimal
0
DigitalNumber
4
Fore_Color
0, 255, 0
MouseControl
False
Show_Sign
False
TestValue
123.456


ON
6: Section 1.5.3 in FAQ-11
t Day / Month / Year and
w Day / Month / Year anc
fresh as Current Date / tim
Property
Tag Type
AO Tag
Tag Name
W_Day1
Tag Description
Output Limit(Max)
31.0
Output Limit(Min)
1.00
DataPointer
Address
13
AddressType
HoldingRegister
DataType
16-bit Signed Integer
Gain
1
Offset
0
Range
-32768.000~32767.000
DisplayFormat
Decimal
0
DigitalNumber
2
Fore_Color
0, 255, 0
MouseControl
True
Show_Sign
False
TestValue
123.456


ON
_06: Section 1.5.3 in FAQ-11
ent Day / Month / Year and
lew Day / Month / Year anc
efresh as Current Date / tim
Property
Tag Type
AO Tag
Tag Name
W_Year1
Tag Description
Output Limit(Max)
2079
Output Limit(Min)
2009
DataPointer
Address
11
AddressType
HoldingRegister
DataType
16-bit Signed Integer
Gain
1
Offset
0
Range
-32768.000~32767.000
DisplayFormat
Decimal
0
DigitalNumber
4
Fore_Color
0, 255, 0
MouseControl
True
Show_Sign
False
TestValue
123.456


eL_06: Section 1.5.3 in FAQ-115
Current Day / Month / Year and Time
Set New Day / Month / Year and Time
Refresh as Current Date / time
Property
Tag Type
DO Tag
Tag Name
Refresh_Time
Tag Description
Refresh as Current Date / Time
Output Limit(Max)
0.000
Output Limit(Min)
0.000
DataPointer
Address
102
AddressType
CoilStatus
DisplayFormat
ConfirmWindow
False
FontSize
14
MouseControl
True
OffColor
0, 127, 0
OffDisplayText
Refresh as Current Da
OnColor
0, 255, 0
OnDisplayText
ValueTest
False

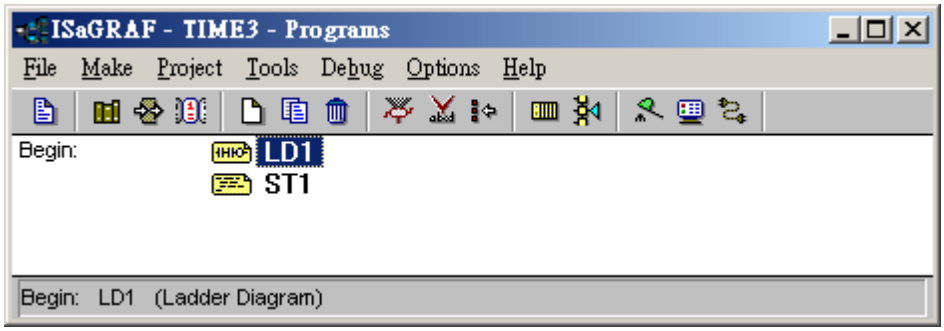

Property
Tag Type
DO Tag
Tag Name
Set_Time
Tag Description
Set as New Date / Time
Output Limit(Max)
0.000
Output Limit(Min)
0.000
DataPointer
Address
101
AddressType
CoilStatus
DisplayFormat
ConfirmWindow
True
FontSize
14
MouseControl
True
OffDisplayText
Set Date / Time
OnColor
0, 255, 0
OnDisplayText
Set Date / Time
ValueTest
False
Time
Time
Set Date / Time

ISaGRAF:

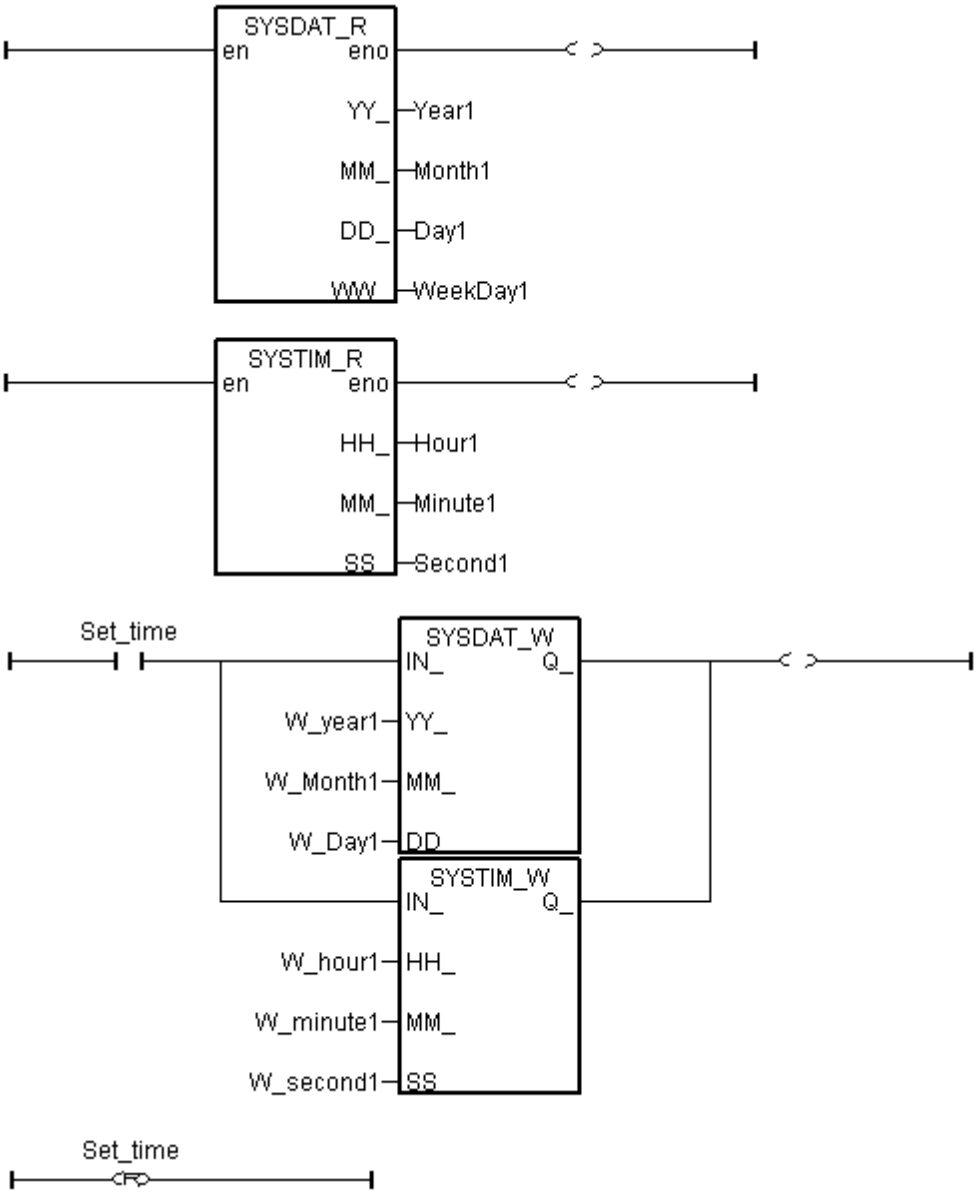
Variables:

Name	Type	Attrib.	NetW. Addr.	Description
Set_time	Boolean	Internal	101	Set True to modify to new time
Refresh_Time	Boolean	Internal	102	Set True to refresh to current time
OUT1	Boolean	Output	0	Link to I-87055W 's CH.1
OUT2	Boolean	Output	0	Link to I-87055W 's CH.2
time_val	Integer	Internal	0	Unit is minute 0 ~ 1439 means 00:00 ~ 23:59
Year1	Integer	Internal	1	Get Year
Month1	Integer	Internal	2	Get Month
Day1	Integer	Internal	3	Get day of Month
Hour1	Integer	Internal	4	Get Hour
Minute1	Integer	Internal	5	Get Minute
Second1	Integer	Internal	6	Get Second
WeekDay1	Integer	Internal	0	Get Weekday
W_Year1	Integer	Internal	11	New Year
W_Month1	Integer	Internal	12	New Month
W_Day1	Integer	Internal	13	New Day
W_Hour1	Integer	Internal	14	New Hour
W_Minute1	Integer	Internal	15	New Minute
W_Second1	Integer	Internal	16	New Second

Project :



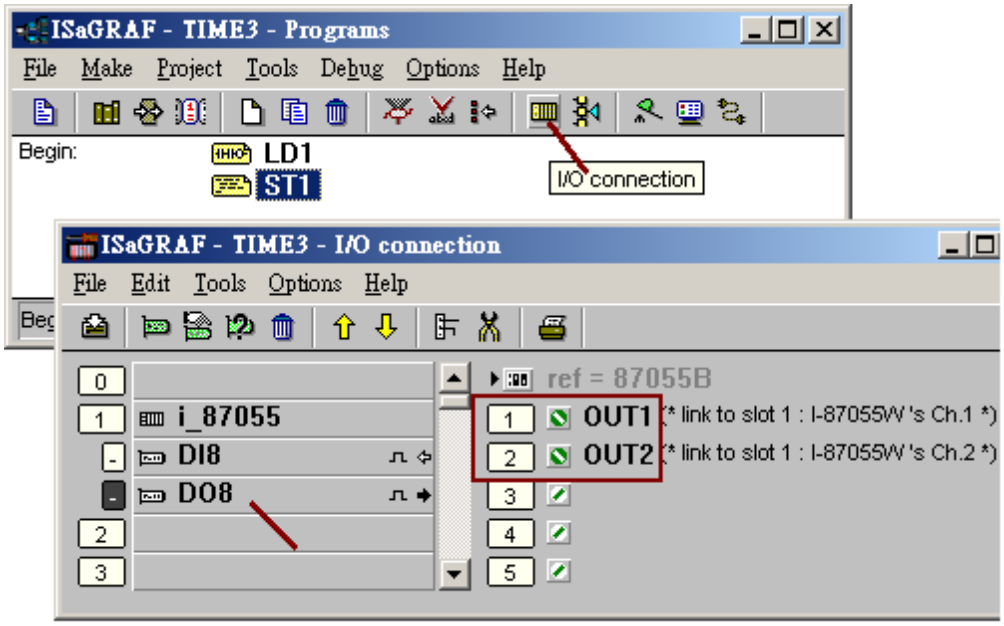
Program - LD1 :


SYSDAT_R
en
eno
YY_
Year1
MM_
Month1
DD_
Day1
WW
WeekDay1
SYSTIM_R
en
eno
HH_
Hour1
MM_
Minute1
SS
Second1
Set_time
SYSDAT_W
IN_
Q_
W_year1
YY_
W_Month1
MM_
W_Day1
DD
SYSTIM_W
IN_
Q_
W_hour1
HH_
W_minute1
MM_
W_second1
SS
Set_time

Program - ST1:

```
(* Unit is second,  0 ~ 86399
Time_val := 3600 * Hour1 + 60* Minute1 + Second1 ;
*)

(* Unit is minute, 0 ~ 1439 *)
Time_val := 60 * Hour1 + Minute1 ;

OUT1 := False ;   (* set as False first *)
OUT2 := False ;

(* set OUT1 as True at 9:00 to 18:00 on Monday to Friday *)
if (WeekDay1>=1) and (WeekDay1<6) and
   (Time_val >= 540) and (Time_val < 1080)  then
  OUT1 := True ;
end_if ;

(* set OUT2 as True at 9:00 to 12:00 on Saturday and Sunday *)
if (WeekDay1=6) and (WeekDay1=7) and
   (Time_val >= 540) and (Time_val < 720)  then
  OUT2 := True ;
end_if ;

(* Refresh to Current date and time *)
if  Refresh_Time  then
  Refresh_Time := False ;  (* reset as False *)
  W_Year1 := Year1 ;
  W_Month1 := Month1 ;
  W_Day1 := Day1 ;
  W_Hour1 := Hour1 ;
  W_Minute1 := Minute1 ;
  W_Second1 := Second1 ;
end_if ;
```

I/O Connection:



How to test ?

Click and set a new date and time and then click “Set Date / Time” button and set it as ON to implement the new date and time .