Chapter 3 Setting Up A Web HMI Demo

The WinPAC-8xx7 (or WP-8xx7) is the abbreviation of the WinPAC-8147 / 8447 / 8847 / 8137 / 8437 / 8837.
The WinPAC-8xx6 (or WP-8xx6) is the abbreviation of the WinPAC-8146 / 8446 / 8846 / 8136 / 8436 / 8836.

Important Notice:

1. **WP-8xx7 / 8xx6 supports only High profile I-8K and I-87K I/O cards in its slot 0 to 7**.
   Refer to WP-8xx7 CD: \napdos\isagraf\wp-8xx7\english_manu\ wp-8xx7_datasheet.pdf
2. Please always set a **fixed IP** address to the WinPAC-8xx7. (No DHCP) Recommend to use the NS-205 / NS-208 Industrial Ethernet Switch for WinPAC-8xx6/8xx7.
3. The leftmost I/O slot number of the WinPAC is 0.

3.1 Web Demo List

The Web page location:
WinPAC-8xx7 CD-ROM: \napdos\isagraf\wp-8xx7\wp_webhmi_demo\

The respective ISaGRAF project location:
WinPAC-8xx7 CD-ROM: \napdos\isagraf\wp-8xx7\demo\

Demo list:

Name	Description	IO board
sample	A Web HMI sample	No I/O board
example1	A simple example listed in Chapter 4	slot 0: I-87055W
wphmi_01	Display controller's date & time	No I/O board
wphmi_02	DI & DO demo	slot 0: I-87055W
wphmi_03	Read / Write Long, float & Timer value	No I/O board
wphmi_04	Read / Write controller's String	No I/O board
wphmi_05	Multi-Pages demo Page menu is on the Left	slot 0: I-87055W
wphmi_05a	Multi-Pages demo Page menu is on the Top	slot 0: I-87055W
wphmi_06	AIO demo, scaling is in ISaGRAF	slot 2: I-87024W slot 3: I-8017HW
wphmi_07	AIO demo, scaling is in PC	slot 2: I-87024W slot 3: I-8017HW
wphmi_08	download controller's file to PC	slot 0: I-87055W

wphmi_09	pop up an alarm window on PC	slot 0: I-87055W
wphmi_11	Trend curve.	slot 2: I-87024W slot 3: I-8017hW
wphmi_12	Record 1 to 8 Ch. i8017HW 's volt every 50ms and draw trend curve by M.S.Excel	slot 3: I-8017hW slot 2: I-8024W
wphmi_13	Record 1 to 4-Ch. i8017HW's voltage every 10ms and draw trend curve by M.S.Excel	slot 3: I-8017hW slot 2: I-8024W

3.2 Steps To Set Up A Web HMI Demo

3.2.1 Step 1 - Setup The Hardware

A. Please have one WP-8147/8447/8847 and then plug one I-87055W board in its slot 0.

If you don't have the I-87055W (8 IN & 8 OUT board), please follow the same steps as below however your Web HMI demo may be replaced to "wphmi_01" not "wphmi_05"

B. Prepare one VGA monitor, one USB mouse and one Ethernet cable and then connect them to the WinPAC-8xx7. (Keyboard is using the software keyboard on the bottom-right of the VGA screen)

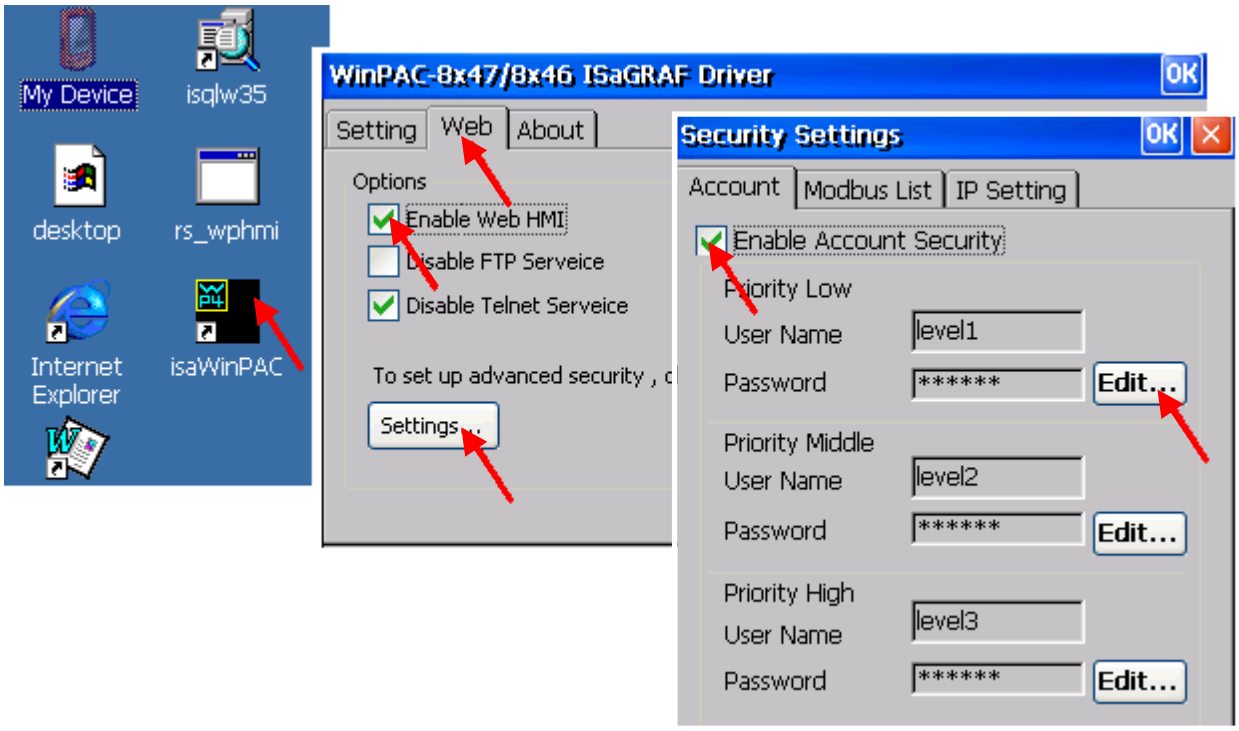
C. Power the WinPAC-8xx7 up.

3.2.2 Step 2 - Setting The Web Options

A. Please refer to the Appendix A.3 to set a **fixed IP** address to the WinPAC. (No DHCP)

B. Check on "Enable Web HMI" and then click on "Setting", Please check the "Enable Account Security" and then click on "Edit" to set (username , password). **Then remember to click on "OK"**

Note: **If "Enable Account Security" is not checked, any user can easily get access to your WinPAC through the Internet Explorer**.



3.2.3 Step 3 - Download ISaGRAF Project

Please download ISaGRAF project “wphmi_05” to the WinPAC-8XX7. This project is in the WP-8xx7 CD-ROM:\napdos\isagraf\wp-8xx7\demo\ “wphmi_05.pia”

wphmi_05 demo need one I-87055W.If you don’t have the I-87055W (8 IN & 8 OUT board), you may download “wphmi_01” (CD-ROM:\napdos\isagraf\wp-8xx7\demo\ “wphmi_01.pia”)

If you know how to restore “wphmi_05.pia” to your ISaGRAF Workbench and download it to the controller, please go ahead to the section 3.2.4.

However if you don’t know it, please refer to the below steps. Please make sure the ISaGRAF Workbench is already installed to your PC. (Refer to the section 2.1 & 2.2)

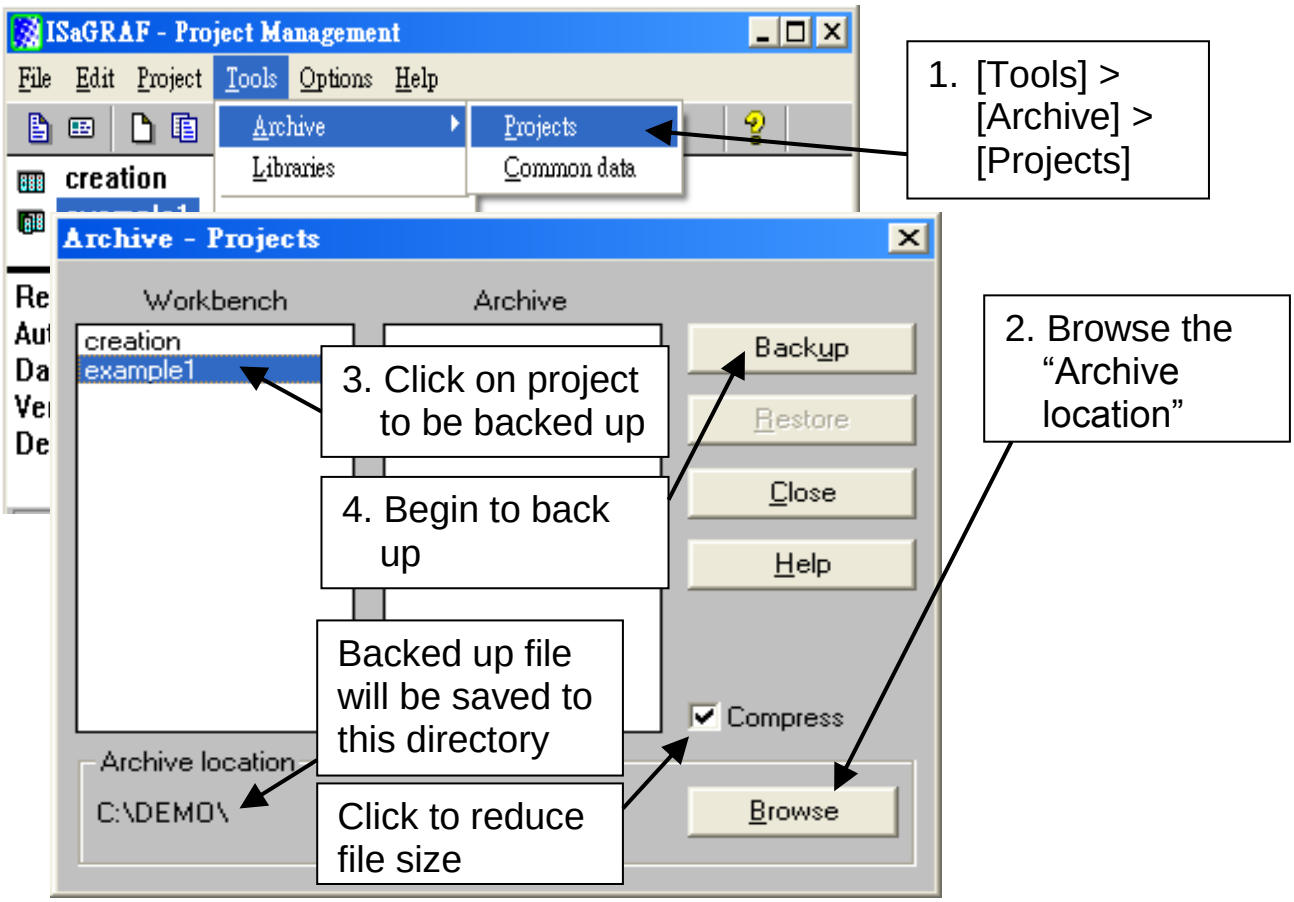
Steps To Backing Up & Restoring An ISaGRAF Project:

For archiving purposes you can "Back Up" and "Restore" an ISaGRAF project. For example, you may want someone to test your program or email to service@icpdas.com for ICP DAS’s ISaGRAF technical service.

Backing Up An ISaGRAF Project

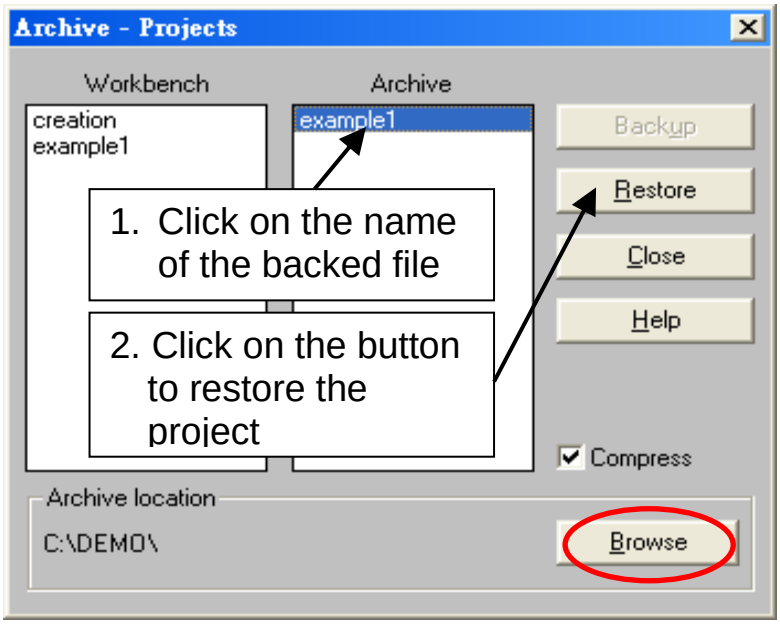
Open the "ISaGRAF Project Management" , select "Tools" from the menu bar, click on "Archive", and then click on "Projects". An "Archive Projects" window will open which allows you to designate where you want to save the ISaGRAF project to. Click on the name of the ISaGRAF project you want to backup, and then click on the "Backup" button. You can compress the size of the file you have backed up by clicking on the "Compress" checkbox BEFORE you click on the "Backup" button.

Then you will now find the backed up ISaGRAF project file in the "Archive" location you have designated. In the example above, the name of the backed up file is "simpleId.pia".



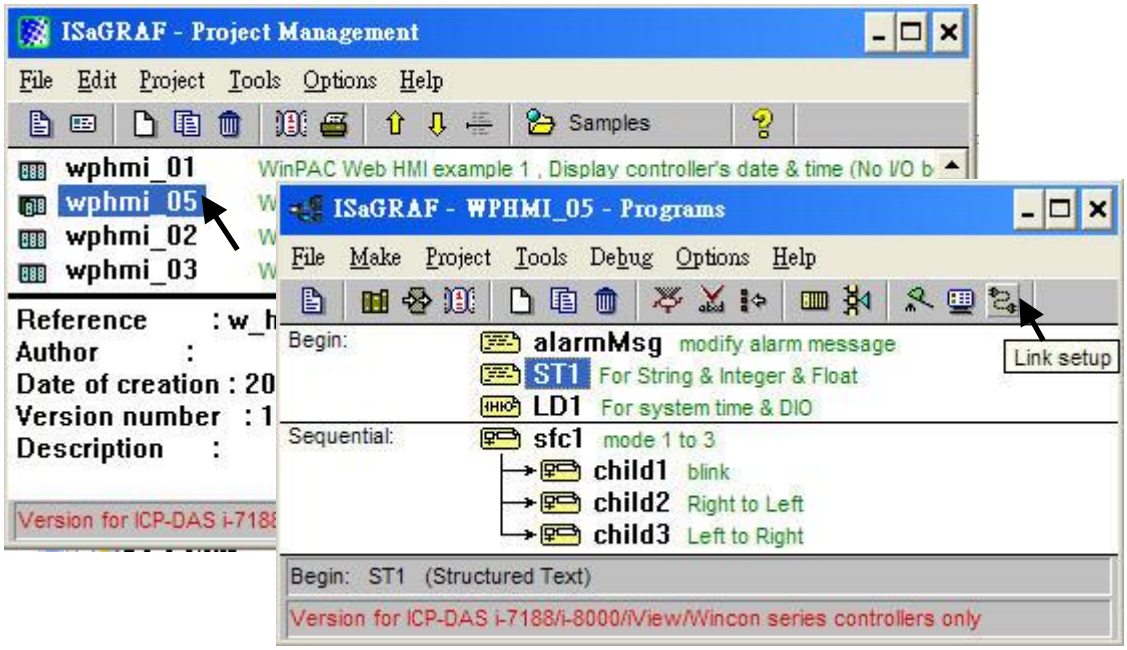
Restoring An ISaGRAF Project

To restore an ISaGRAF project from a backed up file(*.pia), use the same method as above to access the "Archive Projects" window, click on the name of the project you want to restore from the "Workbench" window, then click on the name of the backed up file from the "Archive" window, then click on the "Restore" button. The ISaGRAF project will now be restored to the sub-directory you designated.

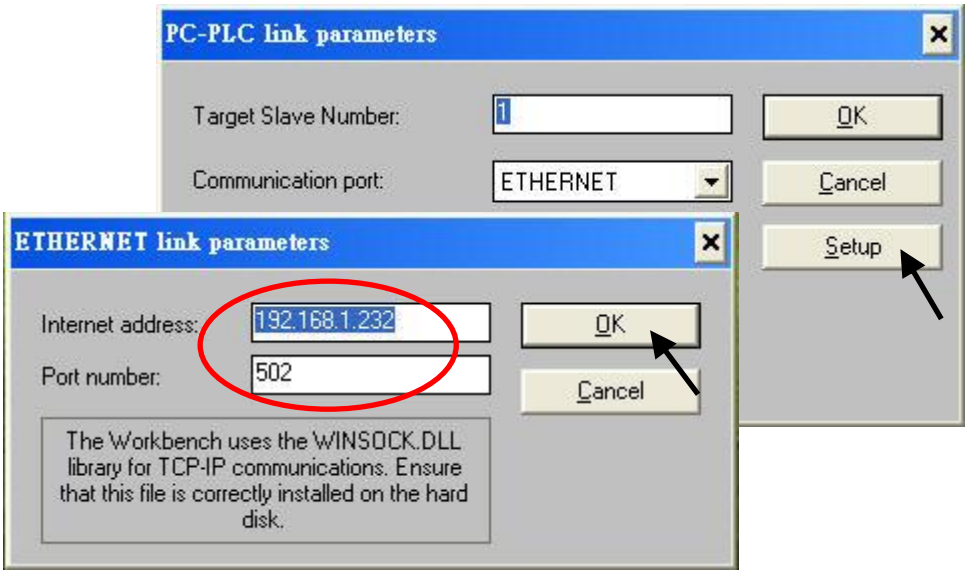


3.2.3.1 Steps To Download an ISaGRAF Project To The Controller:

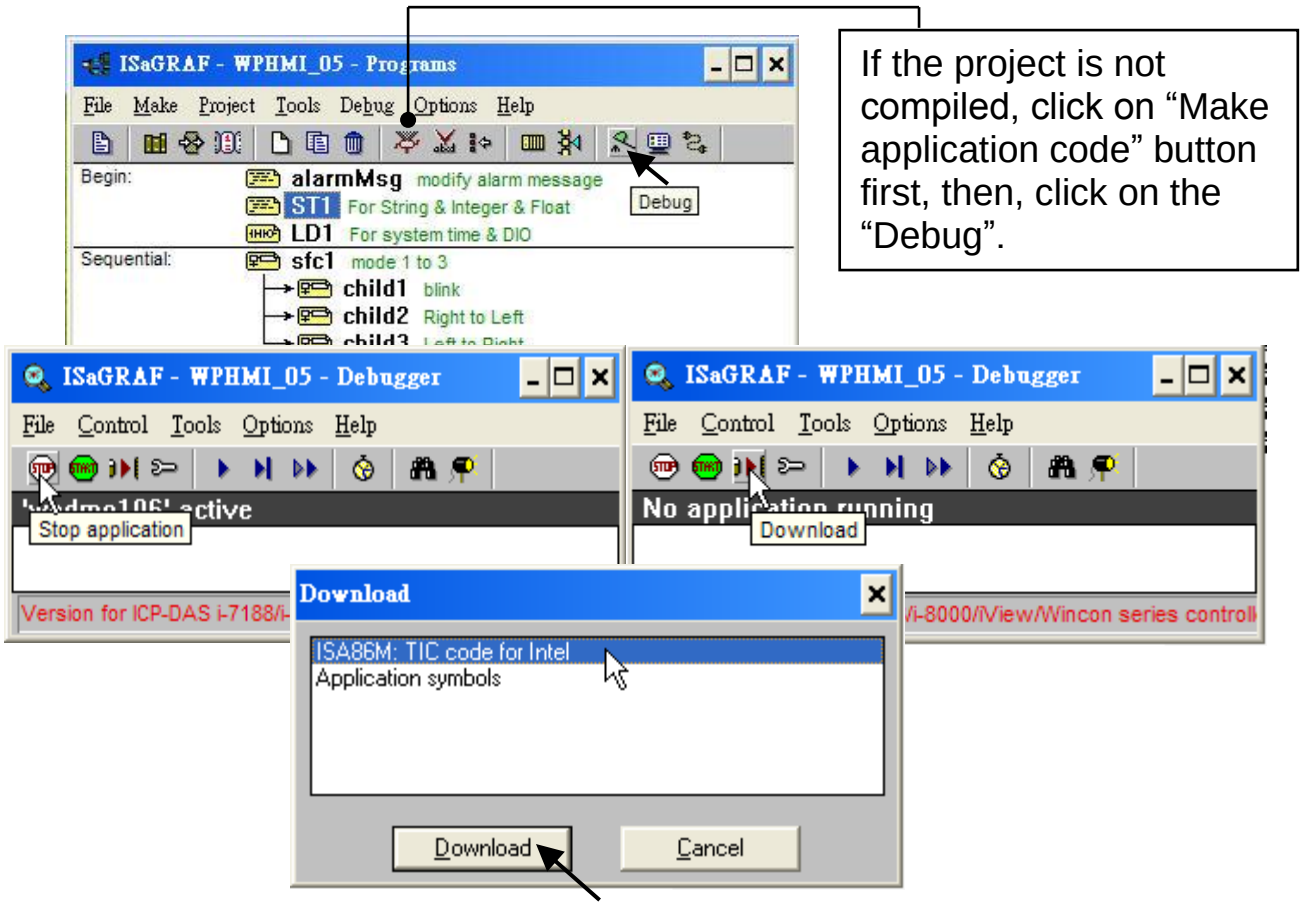
Double click on the “wphmi_05” to get into the project. Then click on “Link setup” .



Click on “Setup” first and then entering the IP address of your controller. The port number should be 502.



To download “wphmi_05” project to the WinPAC-8xx7, Click on “Debug” . If communication is established, click on “stop” first to stop the old project running in the WP-8xx7. Then click on “Download” to download it to the controller.

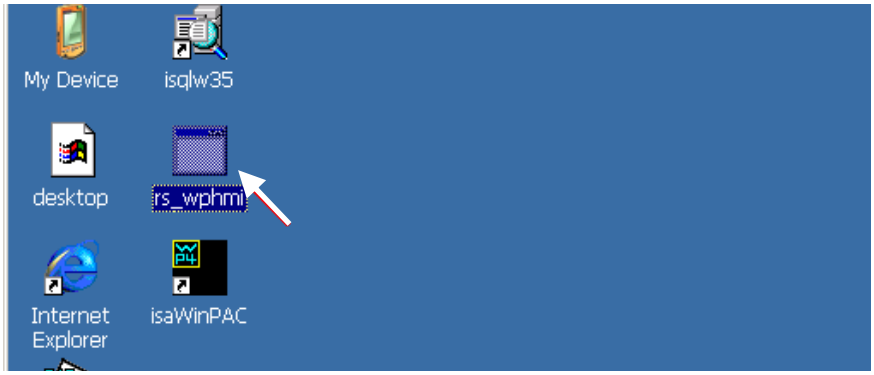


3.2.4 Step 4 - Download Web Pages To The Wincon

A. Please copy all files in the CD-ROM:
WinPAC-8xx7 CD: \napdos\isagraf\wp-8xx7\wp_webhmi_demo\wphmi_05*.*
to the WinPAC-8Xx7 's \Miscro_SD\Temp\HTTP\WebHMI\

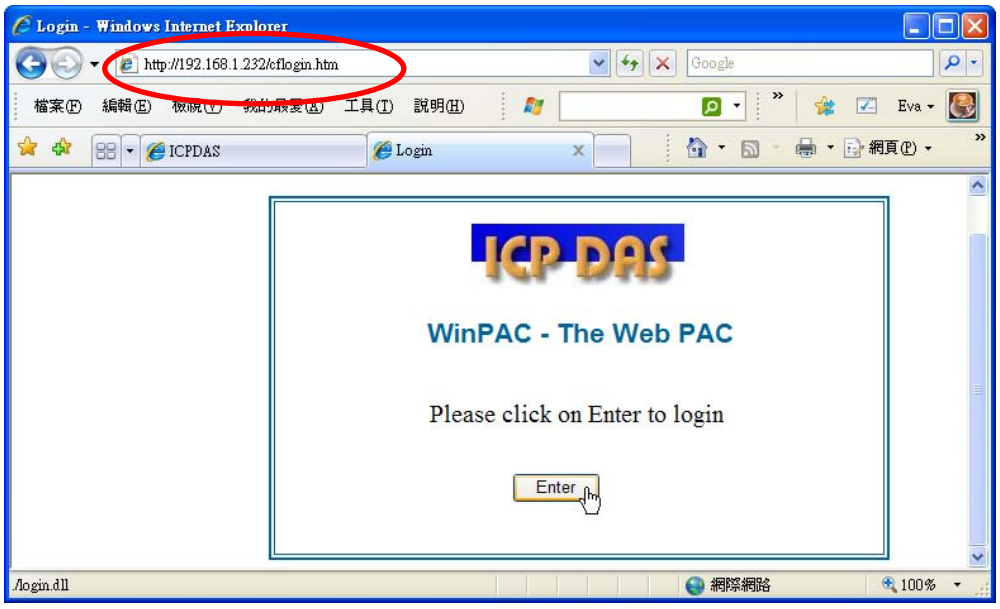
wphmi_05 demo need one I-87055W in its slot 0. If you don't have the I-87055W (8 IN & 8 OUT board), you may download "wphmi_01"

B. Since the Web Pages are modified or new copied, please run "rs_wphmi.exe" to reset the Web server. **The "rs_wphmi.exe" must be run every time when user has modified any file in the WP-8xx7 's \Micro_SD\Temp\HTTP\WebHMI**



3.2.5 Step 5 - Show Time

Please run Internet Explorer (Rev. 6.0 or higher), key in the IP address of your WinPAC-8xx7. For example: 192.168.1.232 or http://192.168.1.232



Chapter 4 Programming A Web HMI Example

This chapter shows you how to build a simple ISaGRAF project and its Web HMI pages.

The WinPAC-8xx7 (or WP-8xx7) is the abbreviation of the WinPAC-8147 / 8447 / 8847 / 8137 / 8437 / 8837.
The WinPAC-8xx6 (or WP-8xx6) is the abbreviation of the WinPAC-8146 / 8446 / 8846 / 8136 / 8436 / 8836.

Important Notice:

1. **WP-8xx7 / 8xx6 supports only High profile I-8K and I-87K I/O cards in its slot 0 to 7**.
   Refer to WP-8xx7 CD: \napdos\isagraf\wp-8xx7\english_manu\ wp-8xx7_datasheet.pdf
2. Please always set a **fixed IP** address to the WinPAC-8xx7. (No DHCP)
3. Recommend to use NS-205 or NS-208 Industrial Ethernet Switch for WinPAC.

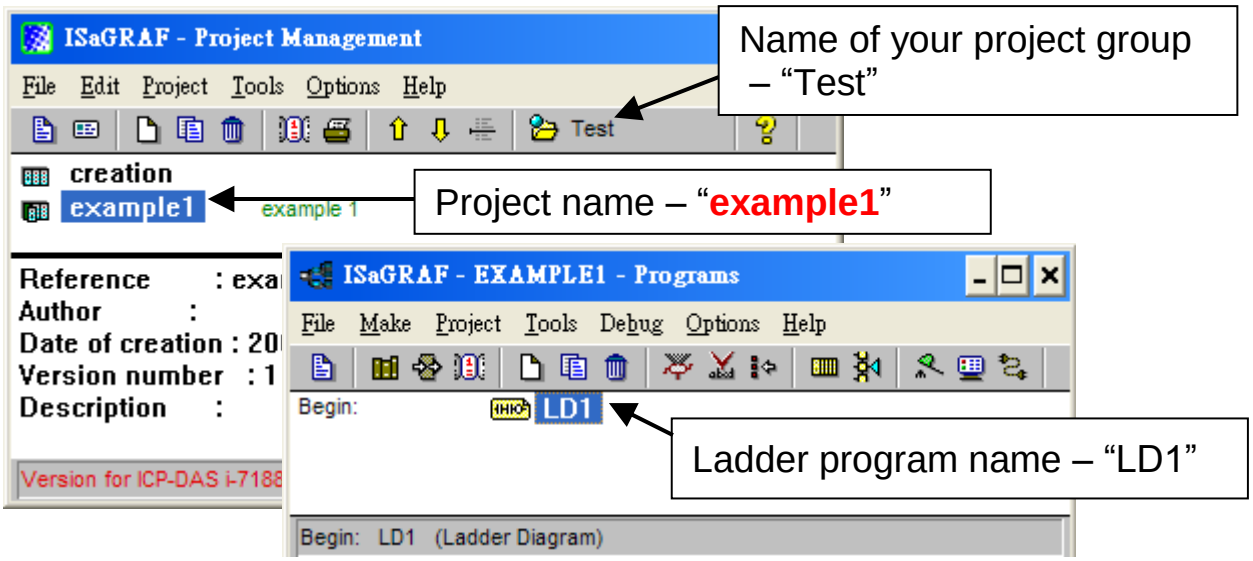
Please refer to CD-ROM: \napdos\isagraf\wp-8xx7\english_manu\ "user_manual_i_8xx7.pdf" - Section 2.1 for detailed ISaGRAF programming basics.

If user would like to program WinPAC-8xx7 by using both ISaGRAF & (EVC++ or VS.net), it is also possible. Please refer to Chapter 6 or Chapter 7.

4.1 Writing A Simple ISaGRAF Program

We are going to use ISaGRAF Workbench to write a simple ISaGRAF example program, then download it to the WinPAC-8xx7 controller (with one **I-87055W** I/O board in its slot 0) to make it work. If you haven't installed "ISaGRAF" & "ICP DAS Utilities for ISaGRAF", please go back to read chapter 2.

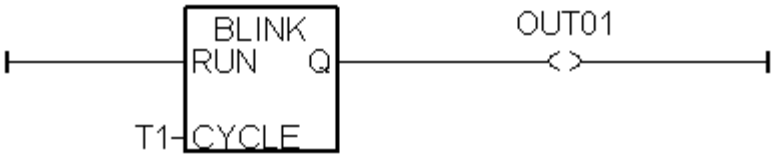
This example contains one Ladder program. (This demo program resides at the
WinPAC-8xx7 CD-ROM: \napdos\isagraf\wp-8xx7\demo\ "example1.pia")



Variables declaration:

Name	Type	Attribute	Description
OUT01	Boolean	**Output**	Output 1 in the I-87055W, Modbus network addr = 1
OUT02	Boolean	**Output**	Output 2 in the I-87055W, Modbus network addr = 2
K1	Boolean	**Input**	Input 1 in the I-87055W, Modbus network addr = 11
K2	Boolean	**Input**	Input 2 in the I-87055W, Modbus network addr = 12
T1	Timer	**Internal**	Time Period of blinking, initial value set as T#8s Modbus network addr = 21

Ladder Logic Program Outline:

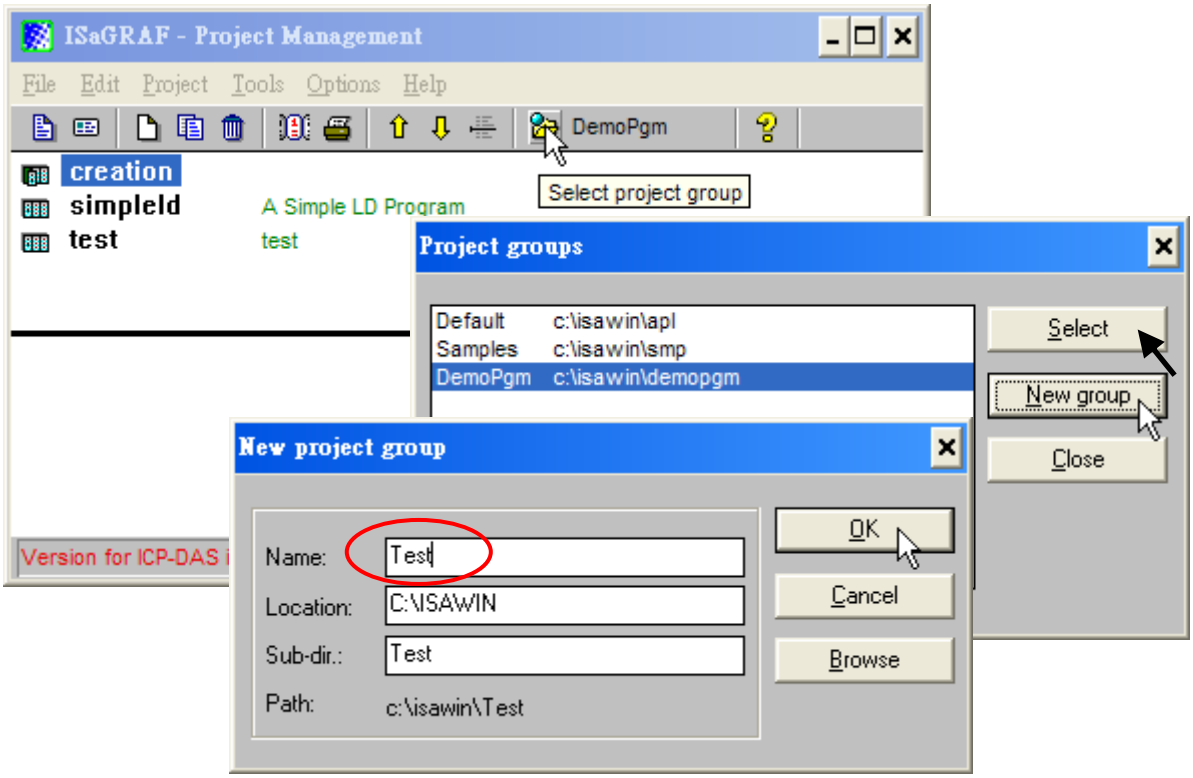


4.1.1 Open ISaGRAF-Project Management

Click on the Windows "Start" button, then click on "Programs" > "ISaGRAF 3.4", (or ISaGRAF 3.5) and then click on "Projects" as shown below.

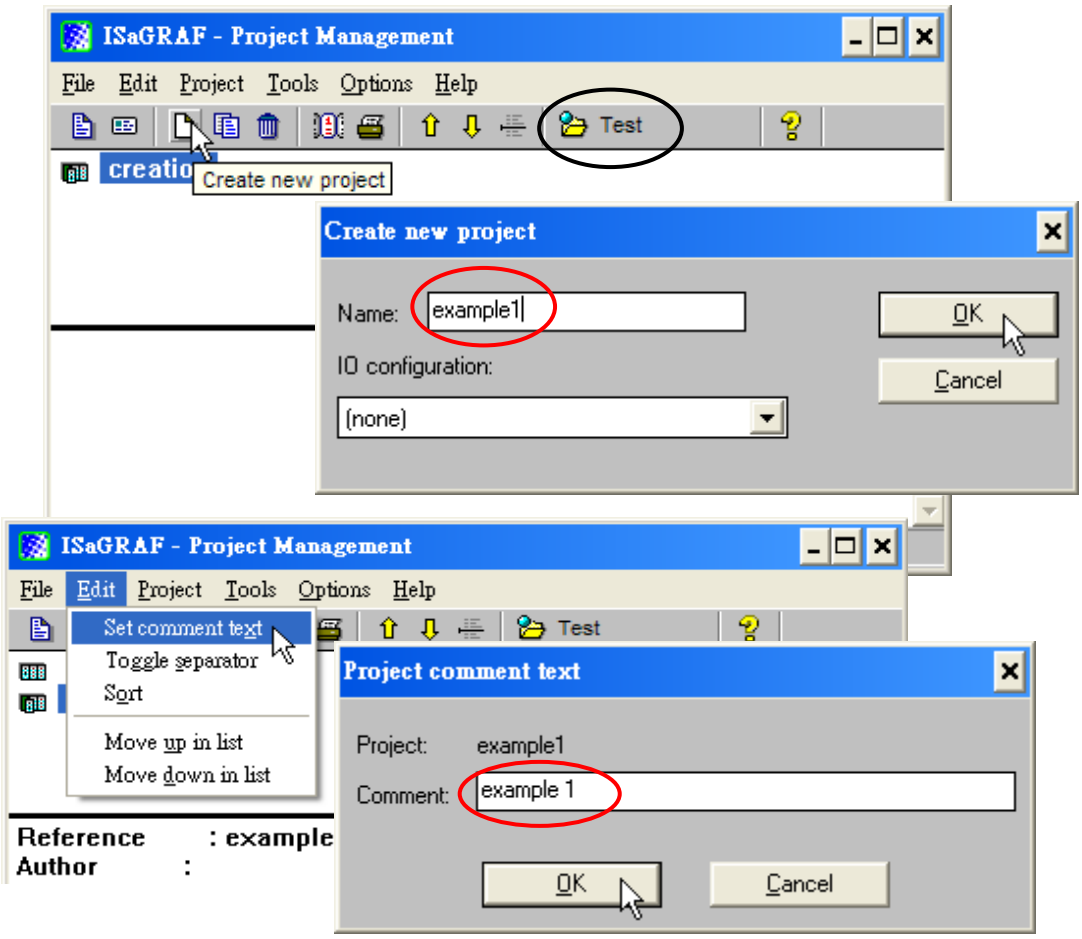
4.1.2 Creating An ISaGRAF User's Group

Click on the "Select Project Group", and then click on "New Group", then type in the name for the new user's group you wish to create, and last click on "OK".

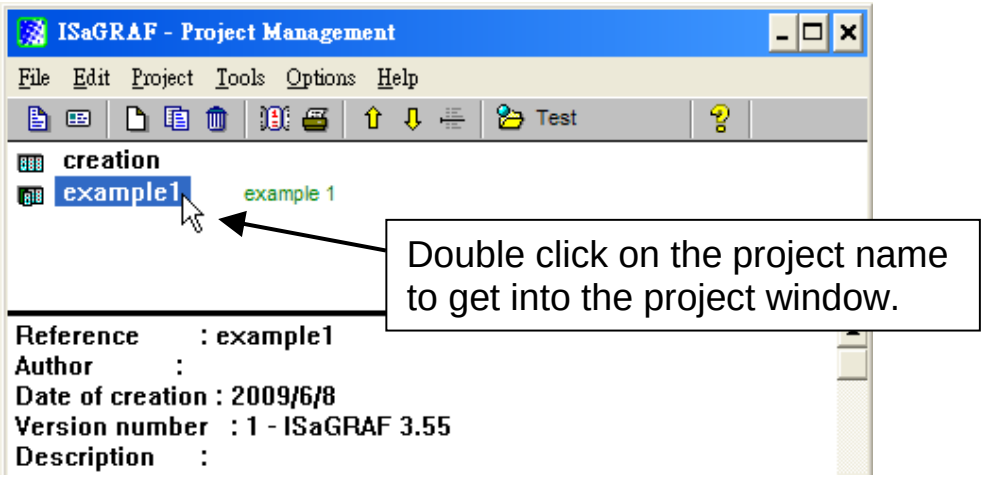


4.1.3 Creating A New ISaGRAF Project

To start a new ISaGRAF project, click on the "Create New Project" icon and then enter in the name for the new project. You can then enter additional information for your project by clicking on the "Edit" and then "Set Comment Text" menu as illustrated below.



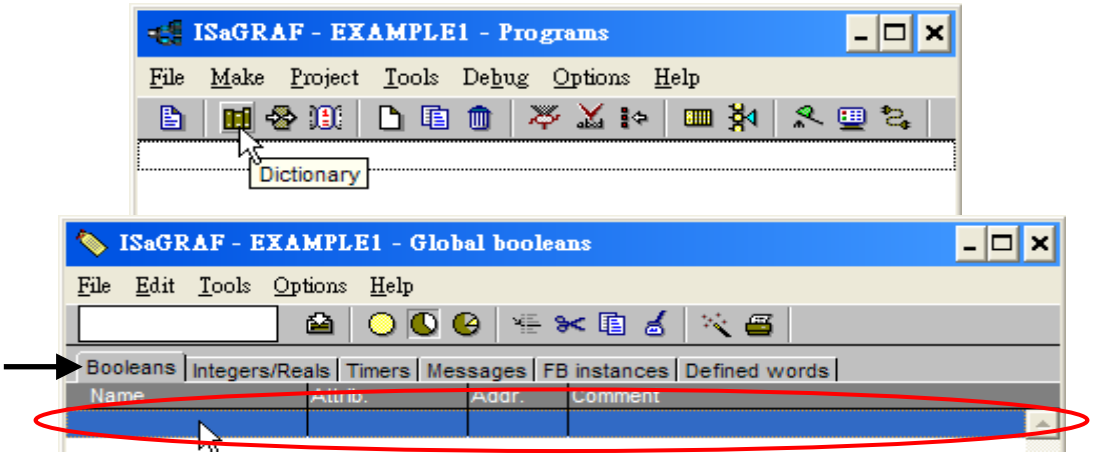
You will now see the name of the new project in the "Project Management" window. Double click on the name of the new project to open the new project.



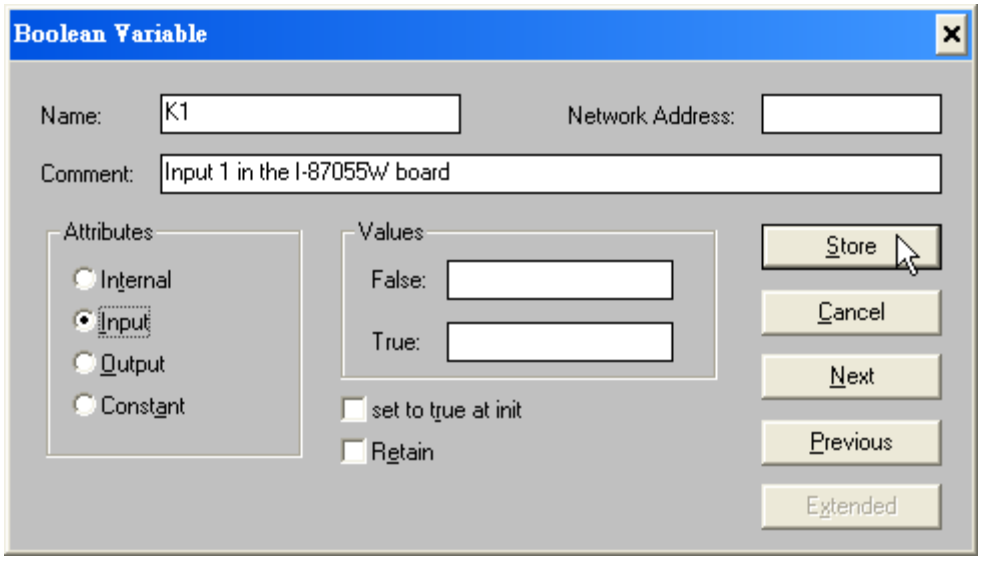
4.1.4 Declaring The ISaGRAF Project Variables

Declare the Boolean Variables

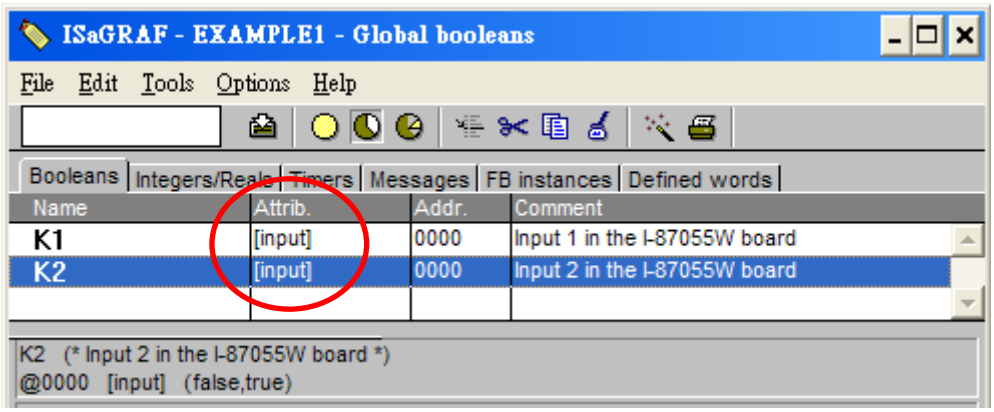
Before you can start creating an ISaGRAF program, you must first declare the variables that will be used in the ISaGRAF program. To begin this process, first click on the "Dictionary" icon and then click on the "Boolean" tab to declare the **Boolean variables** that will be used in our example program.



To declare the program variables for the ISaGRAF project, double click on the colored area below the "Boolean" tab, and a "Boolean Variable" window will open. Enter in the name of the variable to be used in the project. For the purpose of this example program the variable "Boolean Variable Name" is "K1", and "Input 1 in the I-87055W board" is added to the "Comment Section". The next item that must be declared is what type of "Attribute" the variable will possess. In this example program, K1's attribute will be an "Input". Then press the "Store" button to save it.
The new Boolean variable has now been declared.

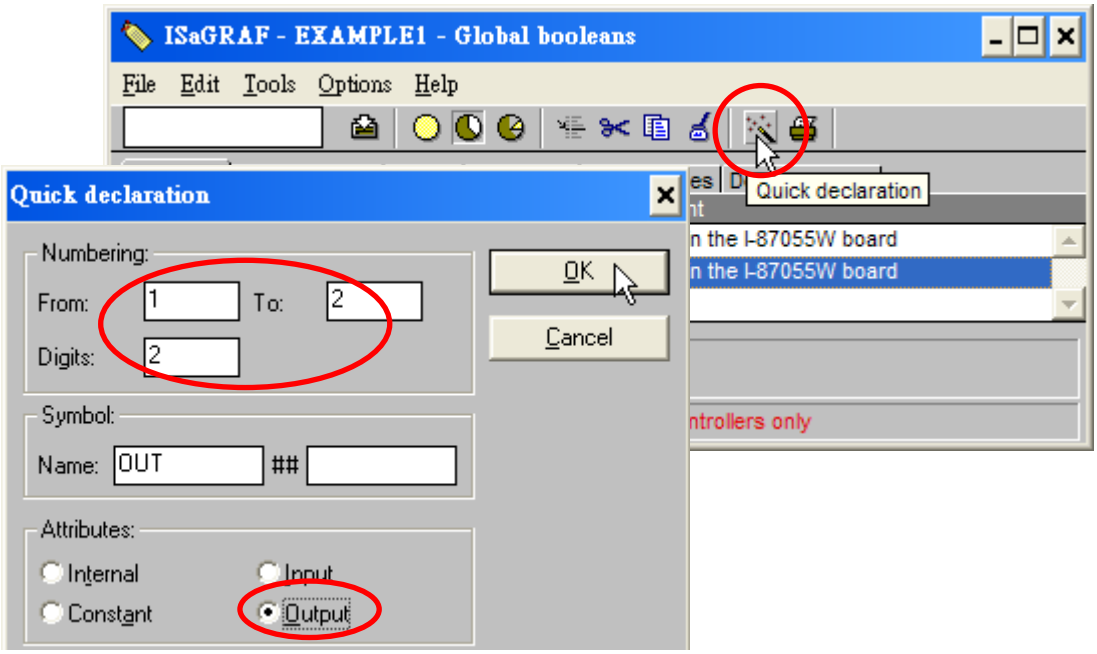


NOTE:
You MUST make sure that the variable you have declared has the desired **Attribute** assigned. If you decide that you want to change a project variable's attribute, just double click on the variable name and you can reassign the attribute for the variable. Please follow the above same step to declare one another Boolean variable – "K2". Then you will have as below.

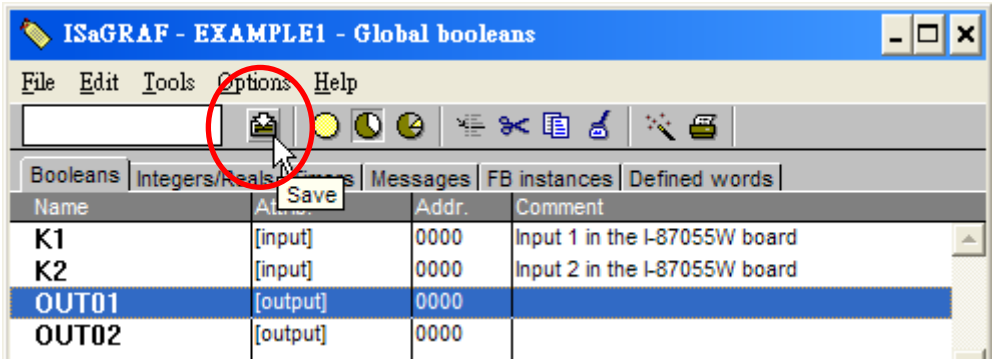


Quick way to declare

There are two outputs used in this example program named "OUT01 and OUT02". ISaGRAF provides a **quick and easy way to declare** like variables that are sequentially ordered. To begin this process, click on the "Quick Declaration" icon, and enter in the output number that you will start with the "Numbering" in "from" and "To" fields (this example uses from 1 to 2). Enter the "Symbol" name for the output variables being declared, and lastly, set the attribute to "Output

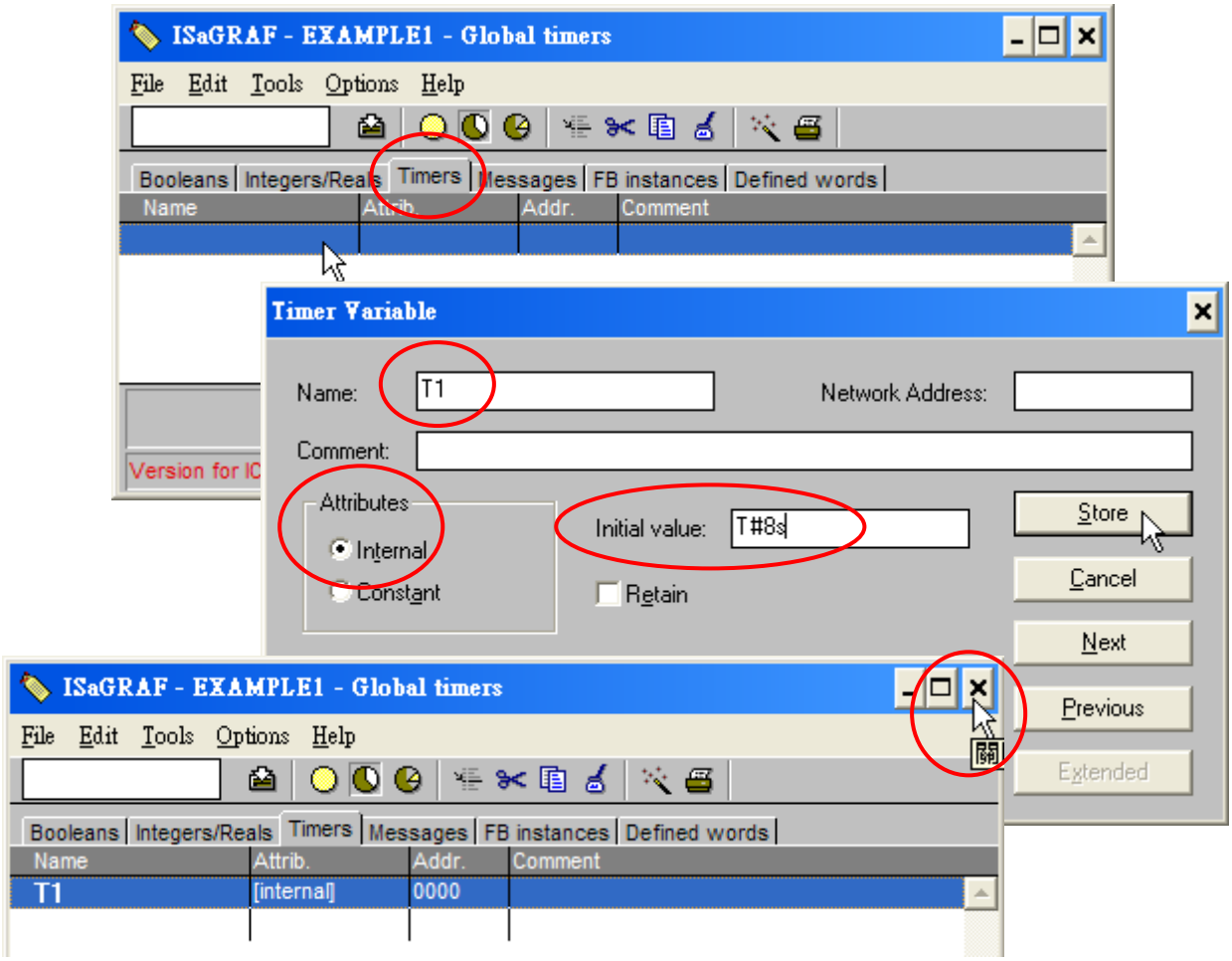


When you click on the "OK" button, all two outputs will be immediately added to the "Global Boolean" window. Click on Save to store them.



Declare the Timer Variables

To declare the timer (T1) variable used in this example program, click on the "Timers" tab in the setup screen. Double click on the colored area and enter the Name as "T1", set the "Attributes" to "Internal", the "Initial Value" to "T#8s", then click on the "Store" button. Then please click on “X” to close the “dictionary” window.



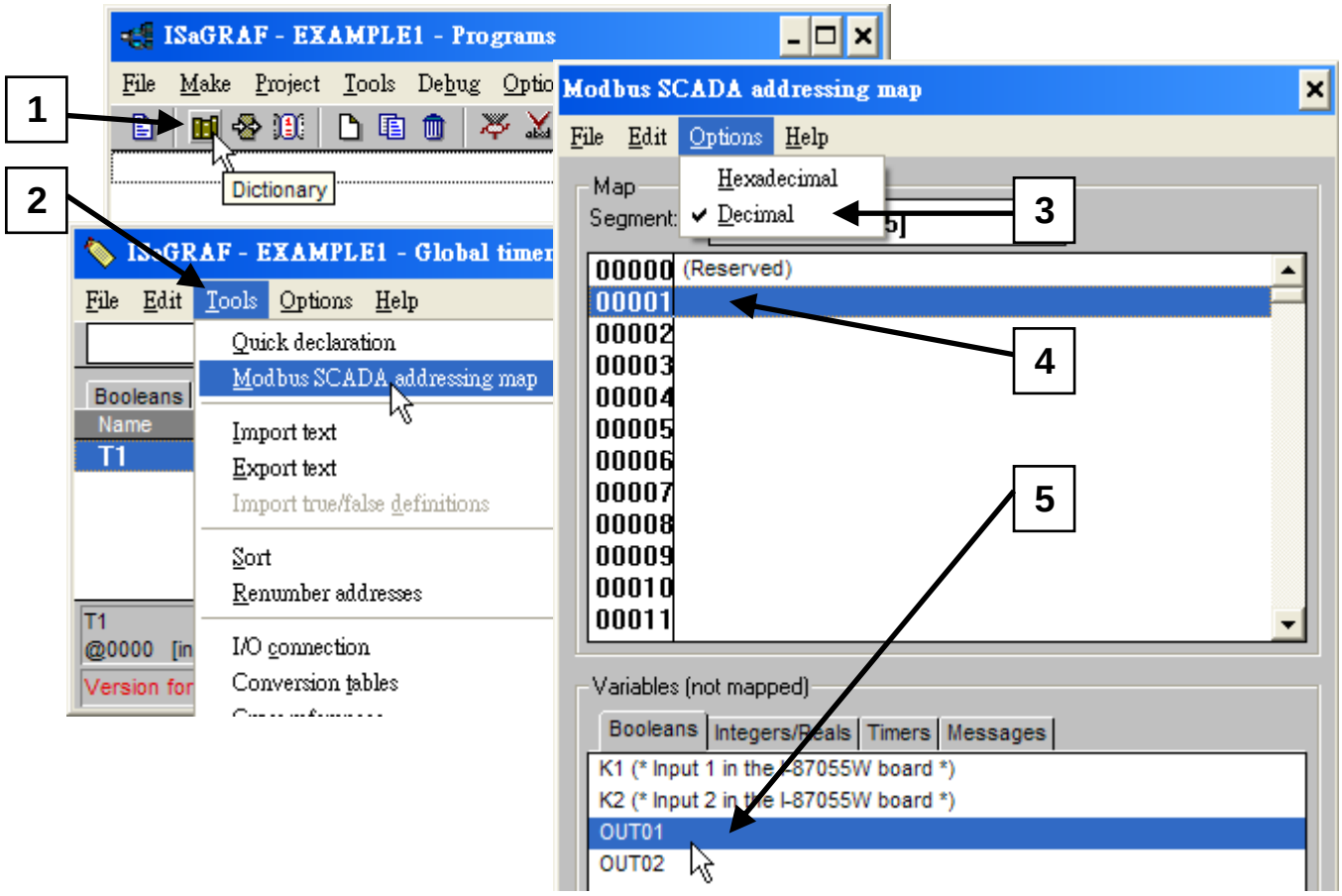
4.1.5 Assign Modbus Network Address No to Variables

The Web HMI will exchange the variable value with the ISaGRAF project if they have assigned the proper “Modbus network address”. The Web HMI only recognize Modbus No. from 1 to 1024. However other SCADA software may R/W the Modbus No. from 1 to 8191 in the WinPAC-8xx7.

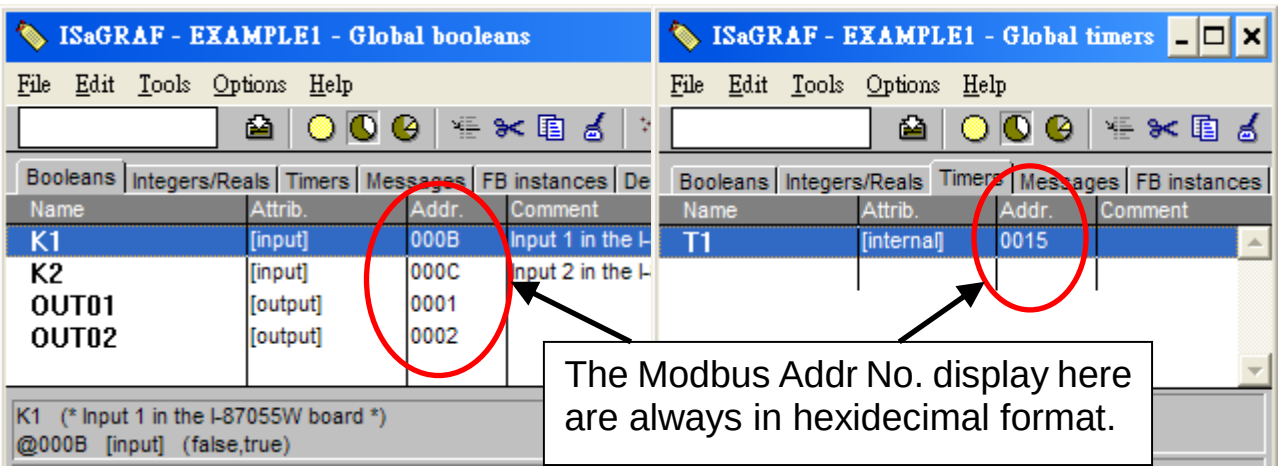
Variables without assigning Modbus No. will not be available by Web HMI and other SCADA software or HMI devices.

Please refer to WinPAC-8xx7 CD-ROM:
\napdos\isagraf\wp-8xx7\english_manu\ "user_manual_i_8xx7.pdf” for section 4.1 & 4.2 for detailed information about assigning Modbus network address.

1. Click on “dictionary” icon
2. Click [Tools] > [Modbus SCADA addressing map]
3. Select [Options] > [Decimal] , or it will use Hexadecimal format as default.
4. click on “00001” on the top window
5. double click on “OUT01” to attach it to the Modbus No. 1.



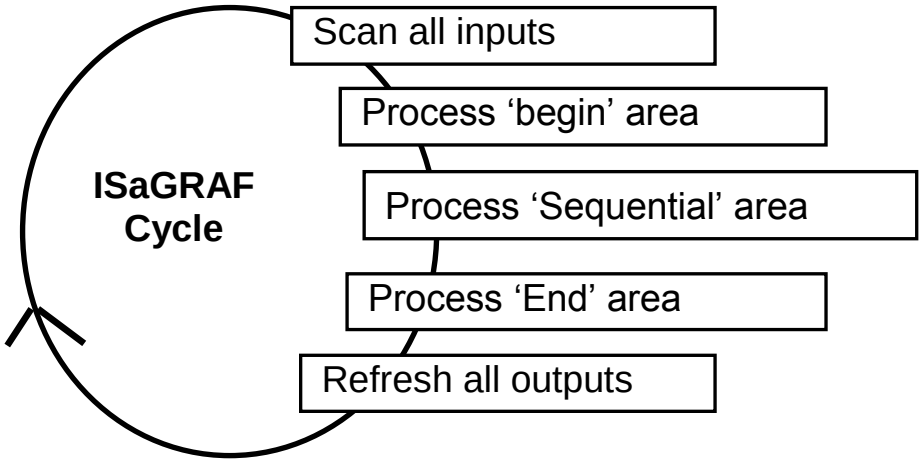
Please follow the same way to assign OUT01 to No.2, K1 to No.11, K2 to No.12 and then Timer variable T1 to No.21. Then we have below window.



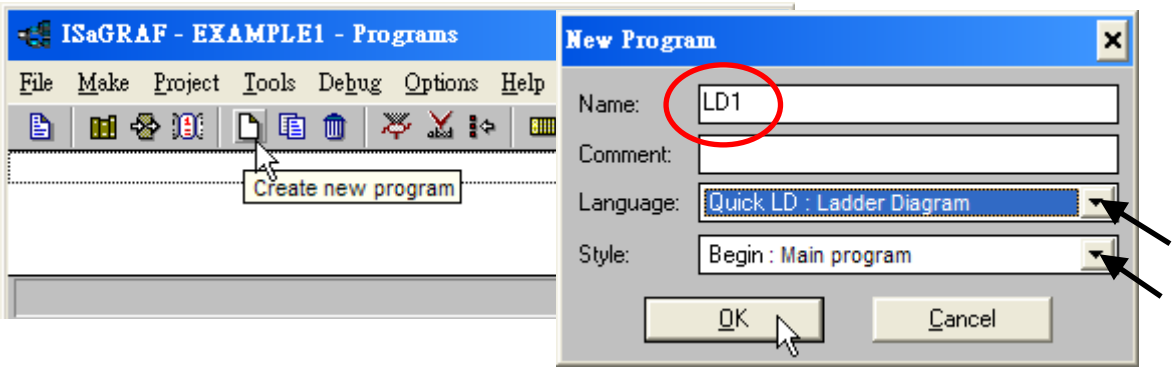
Very Important:
If assign Modbus No. to Long integer or Float or Timer variables, they should occupy two Modbus No. Please refer to WinPAC-8xx7 CD-ROM: \napdos\isagraf\wp-8xx7\english_manu\ "user_manual_i_8xx7.pdf" - Section 4.2 for detailed information.

4.1.6 Create The LD - "LD1" Program

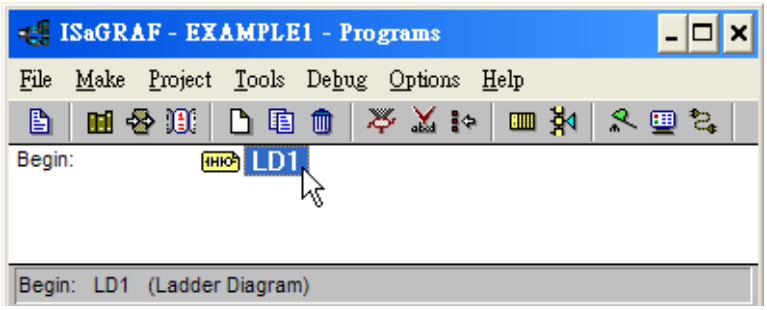
ISaGRAF will run every program one time in each PLC scan cycle. Programs in the "begin" area will run first, then the "Sequential" area, and last the "End" area. An ISaGRAF cycle runs in the way as the below scheme.



Click on the "Create New Program" icon and the "New Program" window will appear. Enter the "Name" as "LD1", next, click on the "Language" scroll button and select "Quick LD: Ladder Diagram", and make sure the "Style" is set to "Begin: Main Program". You can add any desired text to the "Comment" section for the LD program, but it isn't required.

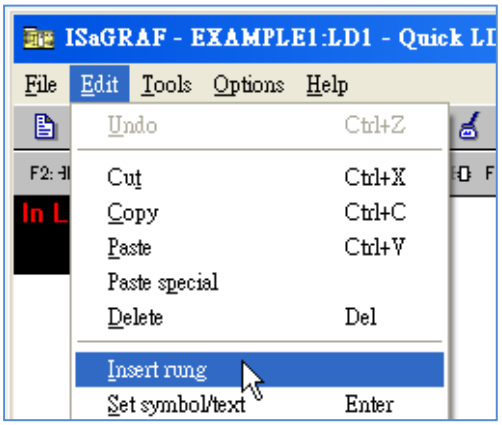


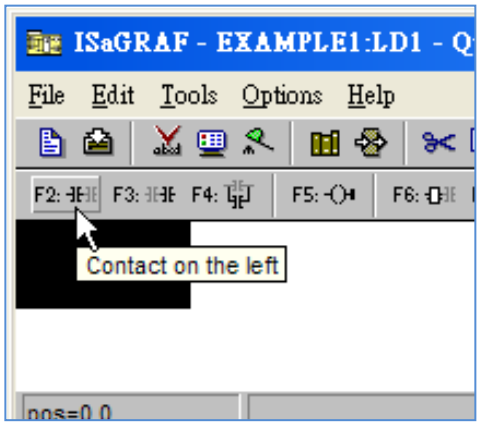
Now we have one program inside this project. Please double click on the "LD1" to get into it.

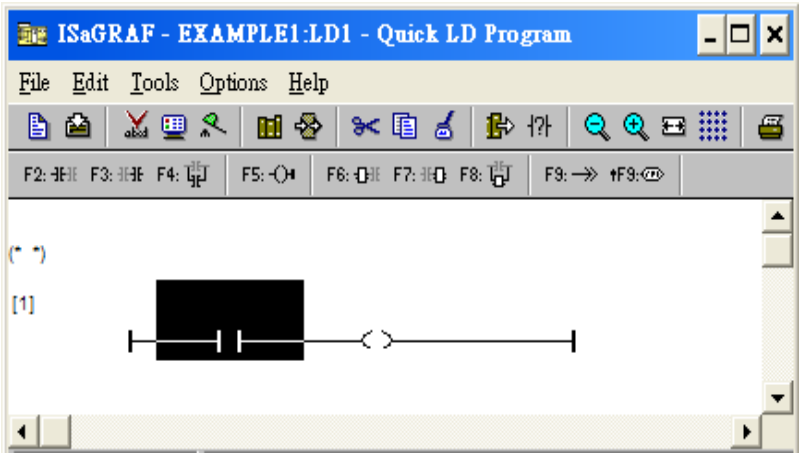


4.1.7 Edit The "LD1" Program

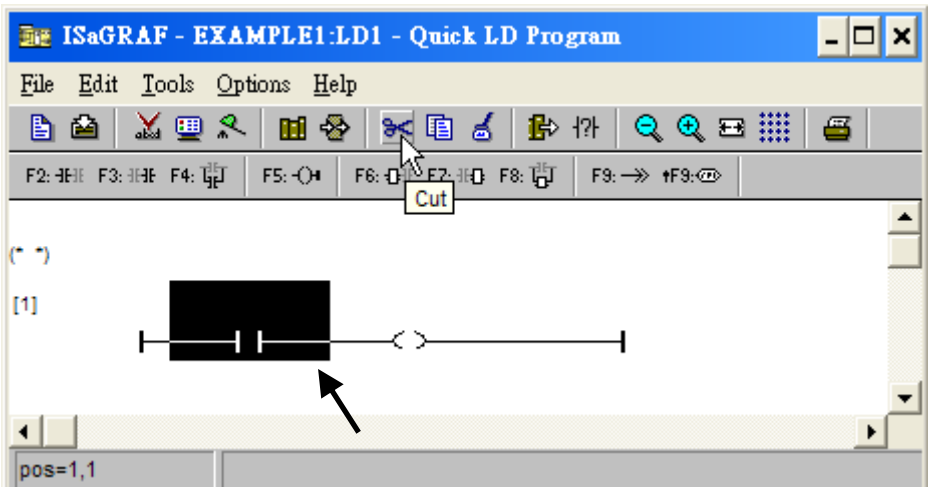
When you double click on the "LD1" name the "Quick LD Program" window will appear. To start programming our LD program, click on "Edit" from the main menu bar, then click on "Insert Rung" . "Insert Rung" means to insert a basic LD rung just above the current position. **Or, you may just simply click on the "F2 (Contact On The Left)"** icon, and the following will appear within the Quick LD Program window.



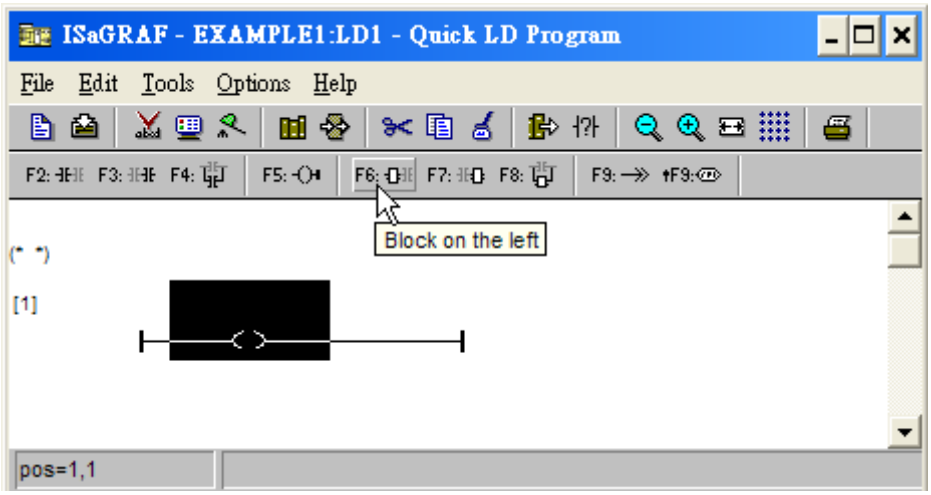




We are going to write the first line of the LD1 program. Move the cursor to the first “contact” and then click on “cut” to delete it.

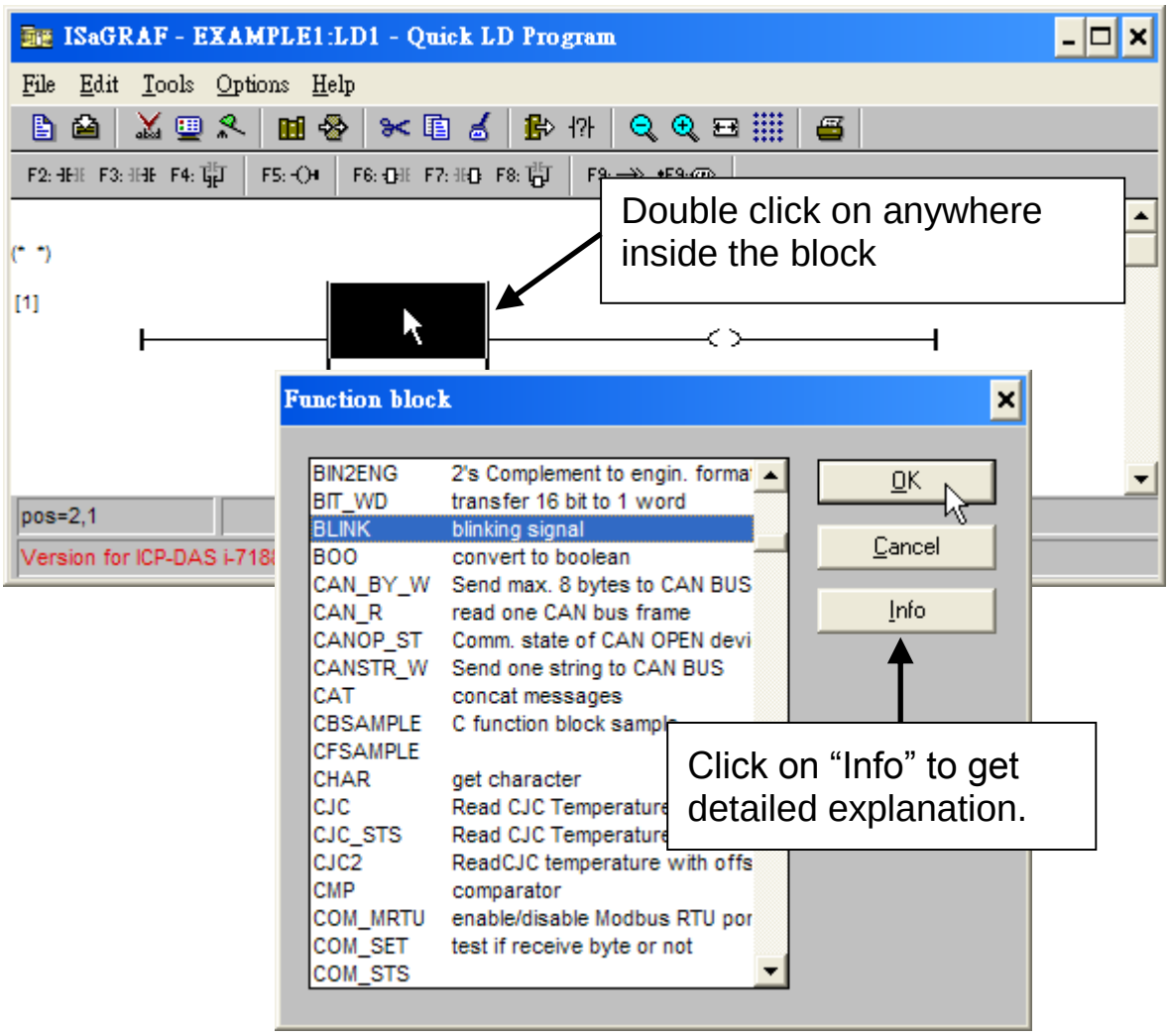


Click on the "F6 (Block on the left)" icon and you will create a block on the left of the “coil”.

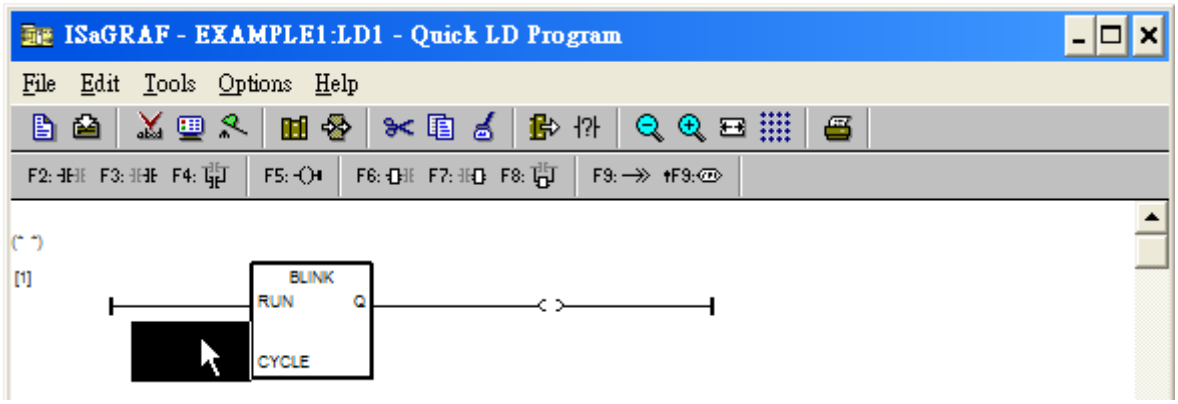


Now we are going to assign the associated variable & constant to each item. Double click anywhere inside of the block and the "Function Block" assignment window appears. Select the "BLINK" type function block. To learn how the

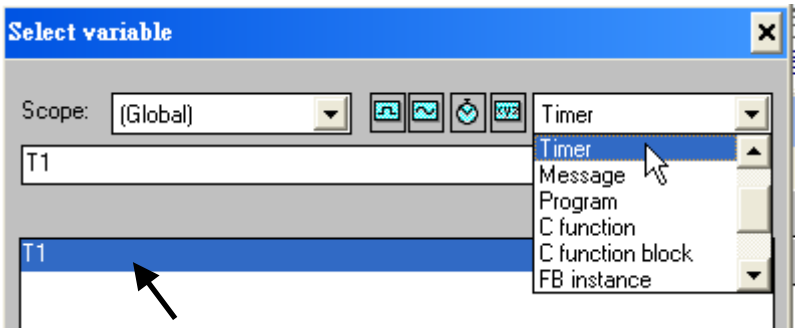
"BLINK" function operates you can click on the "Info" button for a detailed explanation of its functionality



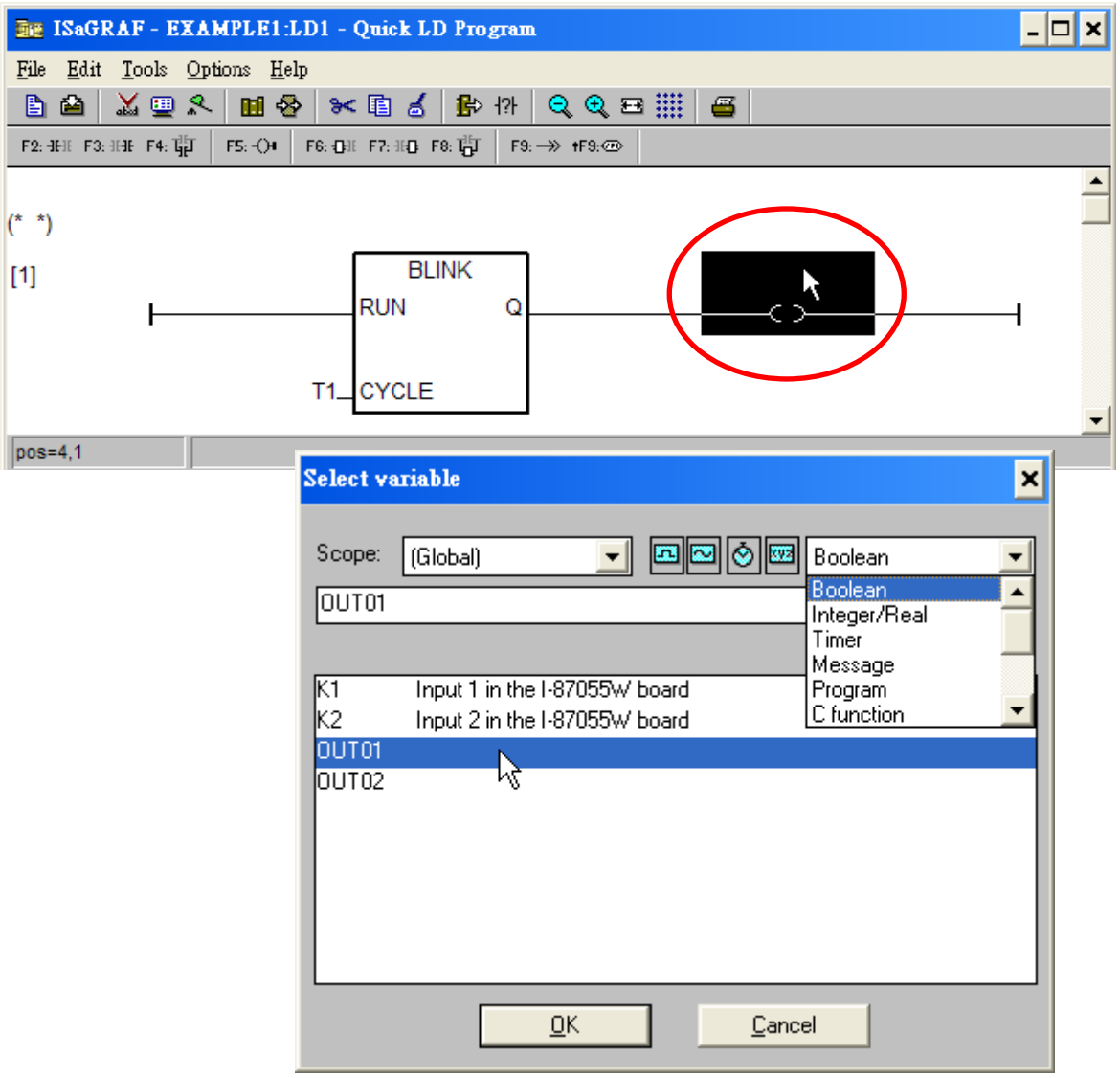
Now move your cursor to the left of the parameter “CYCLE” of the “BLINK” block.



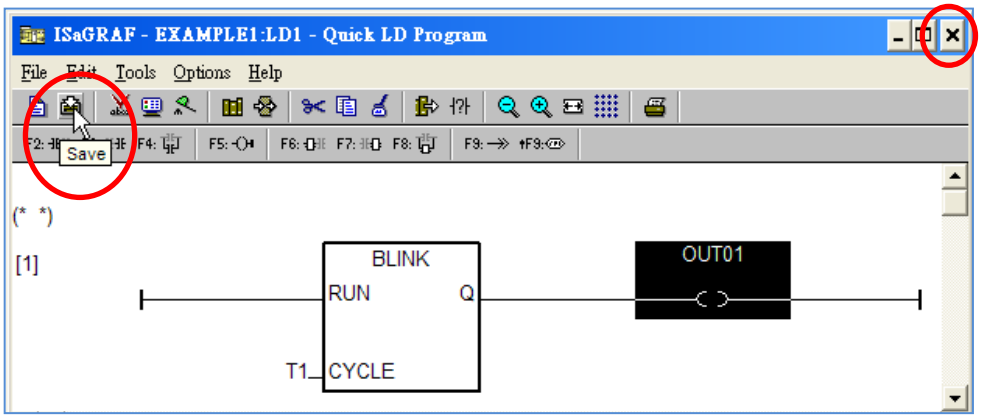
Double click on it, select “Timer” and then double click on variable name - “T1”.



Move your cursor to the “coil”. Double click on it, select “Boolean” and then double click on variable name – “OUT01”.

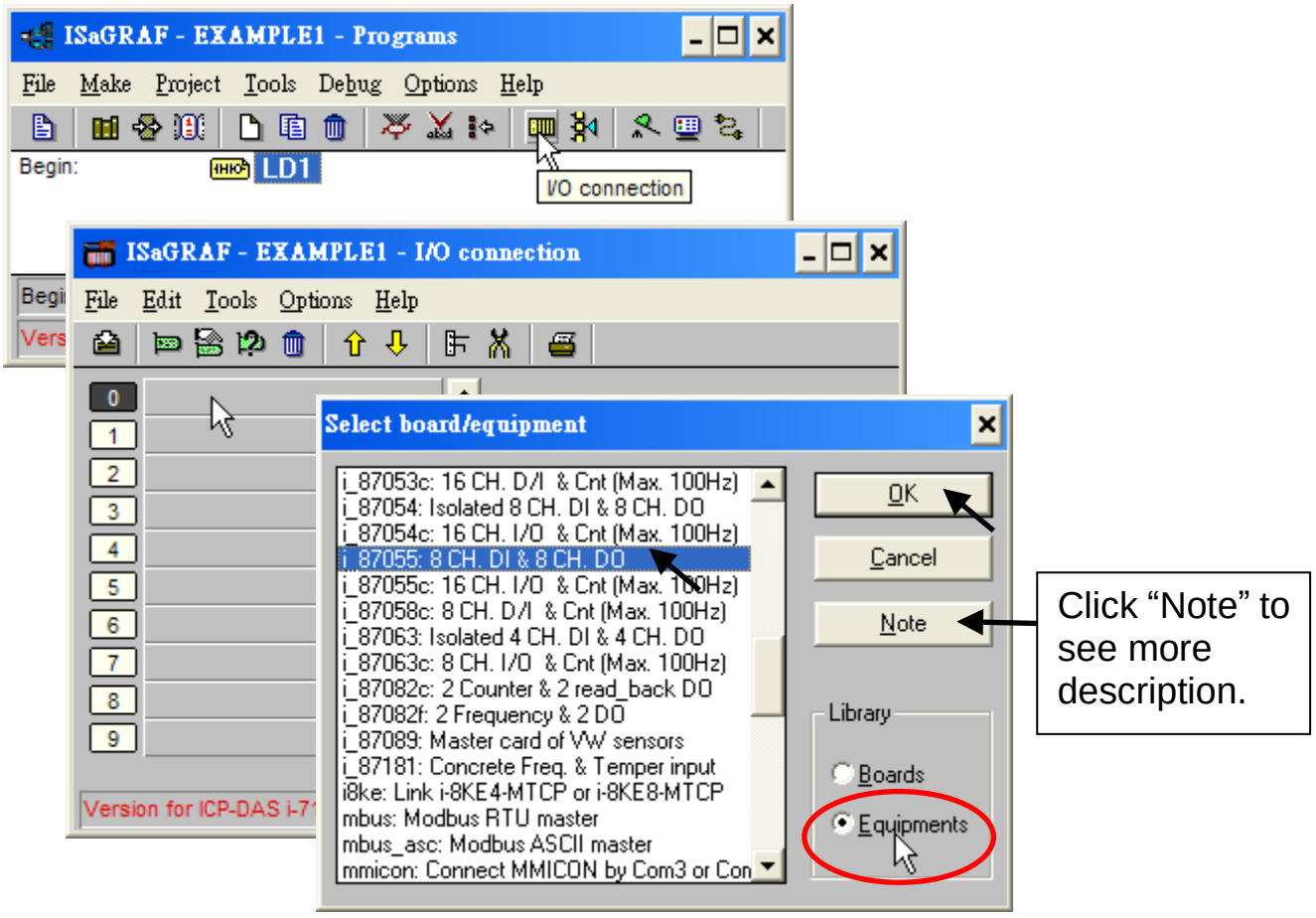


Now we have finished our Ladder code, click on “Save” and then click on “X” to exit.

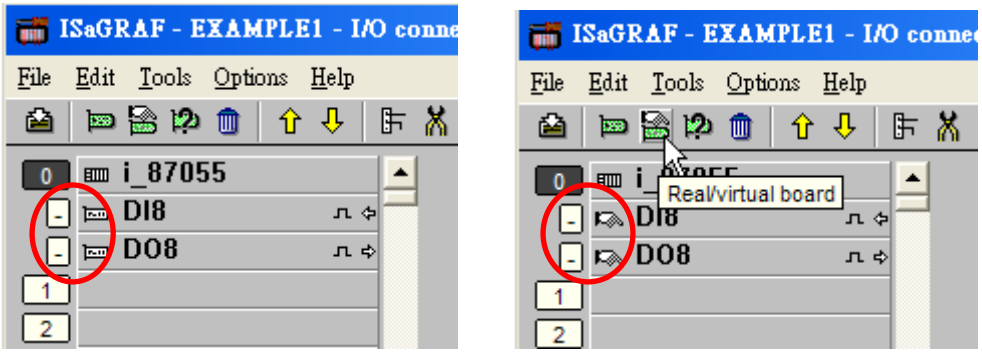


4.1.8 Connecting The I/O

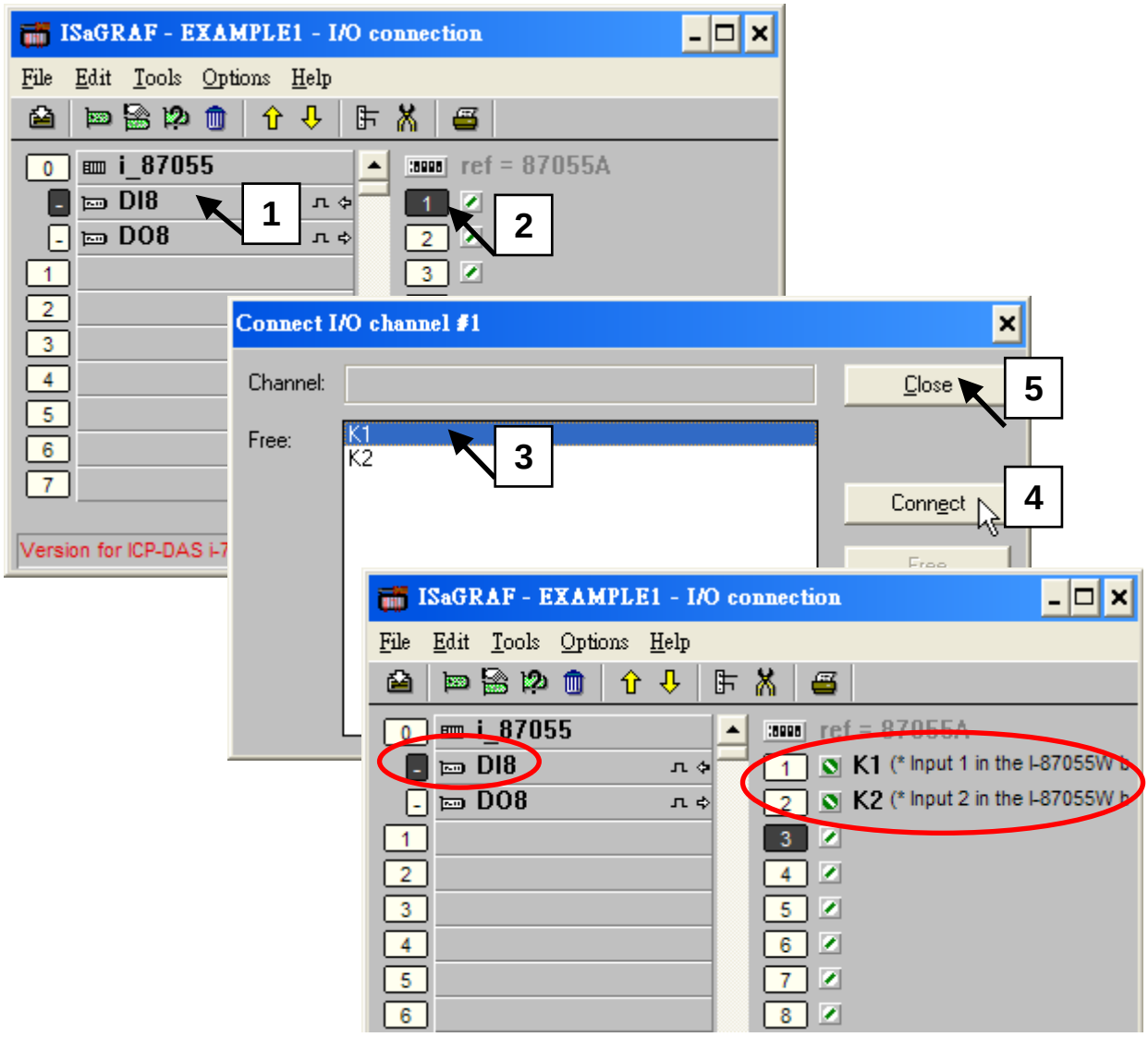
We have defined variables name of “OUT01” , “OUT02” as “output” attribution, while “K1” & “K2” as “input” attribution in step 4.1.4. These “input” & “output” variables should be map to physical I/O in the controller before they can work. To do that, click on “I/O connection” to get into the I/O connection window. Double click on the No. 1 slot (Please make sure your I-87055W I/O board is plug in slot 0 of the WP-8xx7) & then check on the “Equipments” & double click on the “I_87055: 8 CH. DI & 8 CH. DO ”.



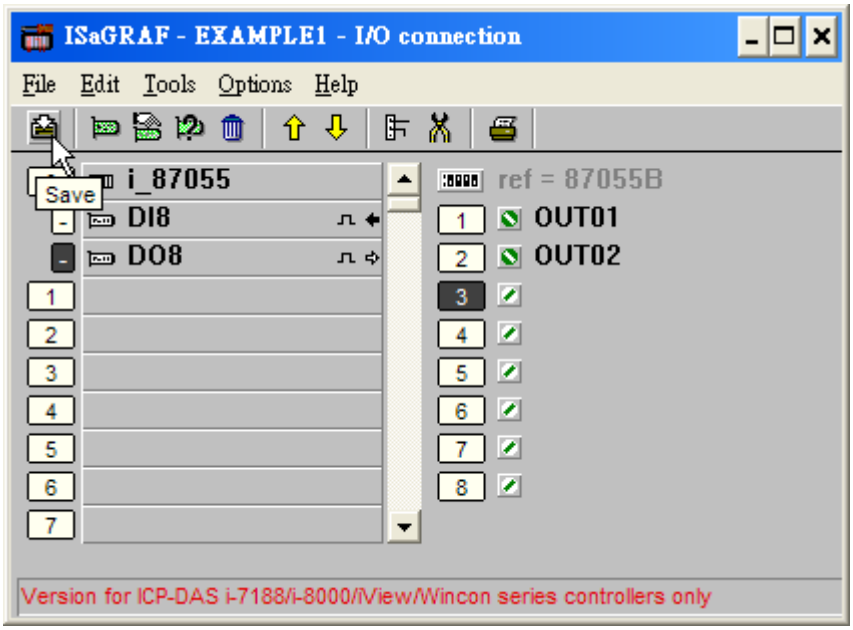
Then we have. (If you don't have the I-87055W, you may click the “Real / Virtual board” to make it become virtual board.)



To map input variables “K1” & “K2” to the input channel No. 1 & 2 of the “I-87055”, double click on the channel 1 and then click on “Connect” .Then click on “Connect” again to connect channel 2.



By the same way, please connect “OUT01” , “OUTPUT02” to output channel 1 to 2. Then we have below window. Click on “Save” and then exit.



IMPORTANT NOTICE:

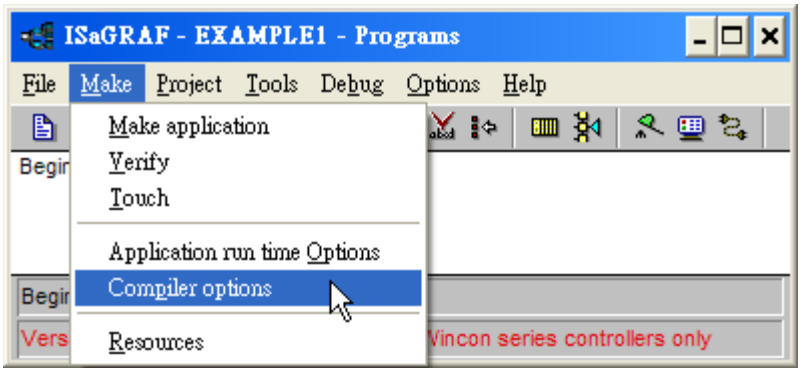
1. I/O Slots 0 through 7 are reserved for REAL I/O boards that will be used in the WP-8xx7. You can use slot No. 8 and above for additional functionality.

2. All of the variables with “Input” and “Output” attribute MUST be connected through the I/O connection as described above for any program to be successfully compiled. Only the Input and Output attributed variables will appear in the "I/O Connections" window. In this example we have only 2 boolean output variables - OUT01, OUT02 and 2 boolean input variables – K1 & K2.

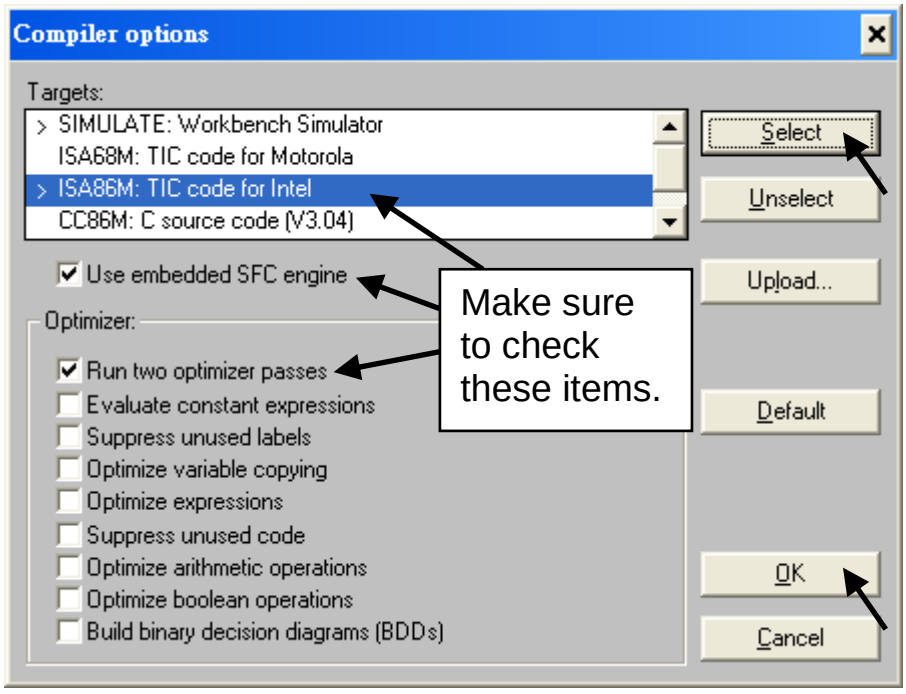
4.2 Compiling & Simulating The Example Project

For ANY AND EVERY ISaGRAF program to work properly with any of the ISaGRAF PACs (ISaGRAF µ PAC, iPAC, WinPAC, ViewPAC...) controller systems, it is the responsibility of the programmer to properly select the correct "Compiler Options". You MUST select the "ISA86M: TIC Code For Intel" option as described below.

To begin the compilation process, first click on the "MAKE" option from the main menu bar, and then click on "Compiler Options" as shown below.



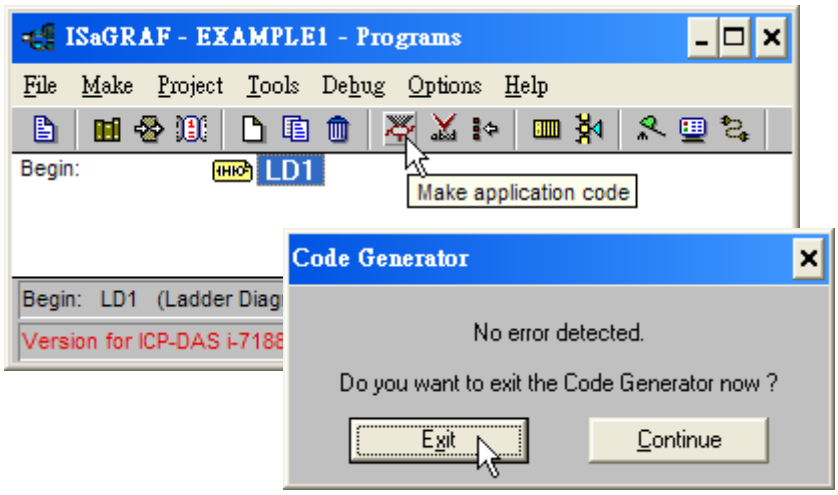
The "Compiler Options" window will now appear. Make sure to select the options as shown below then press the "OK" button to complete the compiler option selections.



Compiling error result in different ISaGRAF Version, please refer to appendix H of this manual.

TIME TO COMPILE THE PROJECT!

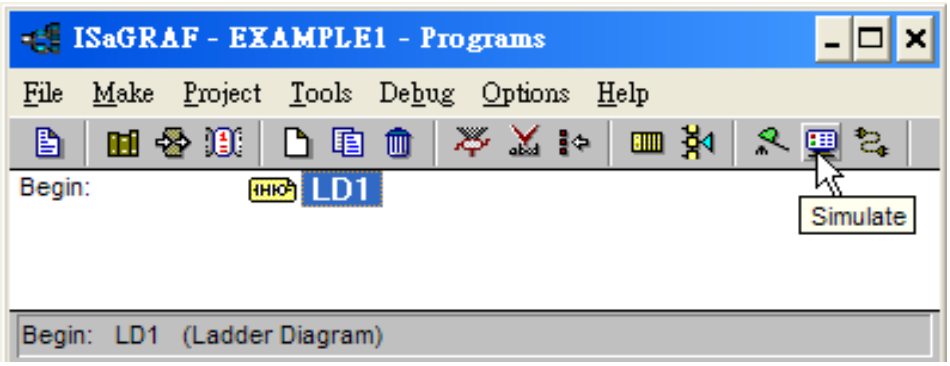
Now that you have selected the proper compiler options, click on the "Make Application Code" icon to compile the example project. If there are no compiler errors detected during the compilation process, CONGRATULATIONS, you have successfully created our example program.



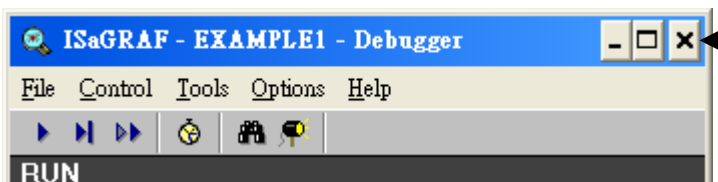
If errors are detected during the compilation process, just click on the "CONTINUE" button to review the error messages. Return to the Project Editor and correct the errors as outlined in the error message window.

TIME TO SIMULATE THE PROJECT!

If the compilation is Ok, you may simulate the project on the PC to see how the program works without the controller. To do that, click on the “Simulate” icon.



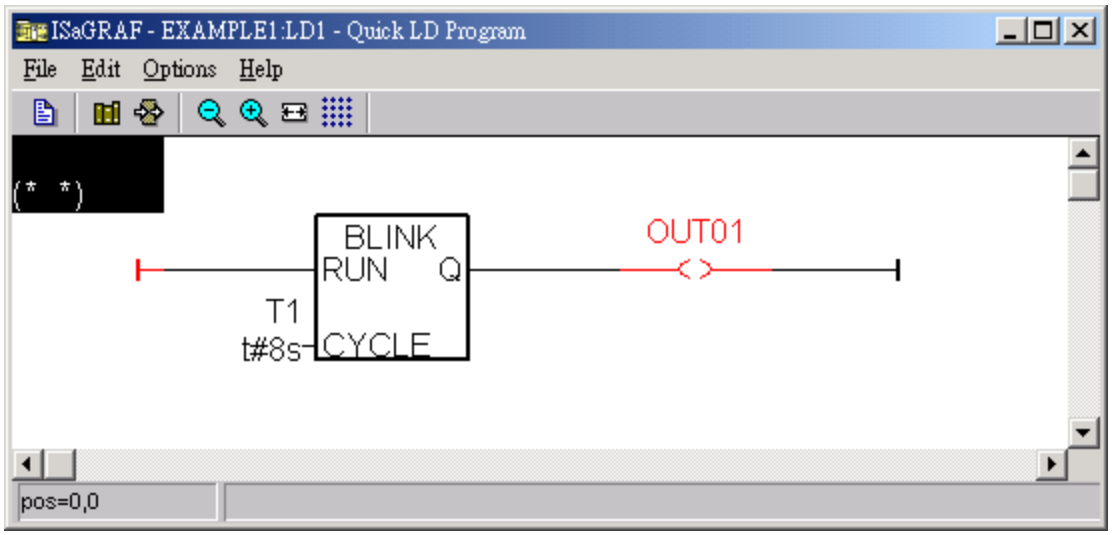
When you click on the "Simulate" icon three windows will appear. The windows are the "ISaGRAF Debugger", the "ISaGRAF Debug Programs", and the "I/O Simulator" windows. If the I/O variable names you have created DO NOT appear in the I/O simulator window, just click on the "Options" and "Variable Names" selection and the variable names you have created will now appear next to each of the I/O’s in the simulator window.
In the "ISaGRAF Debug Program" window, double click on the "LD1" where the cursor below is positioned. This will open up the ISaGRAF Quick LD Program window and you can see the LD program you have created.



Click here to exit the simulator

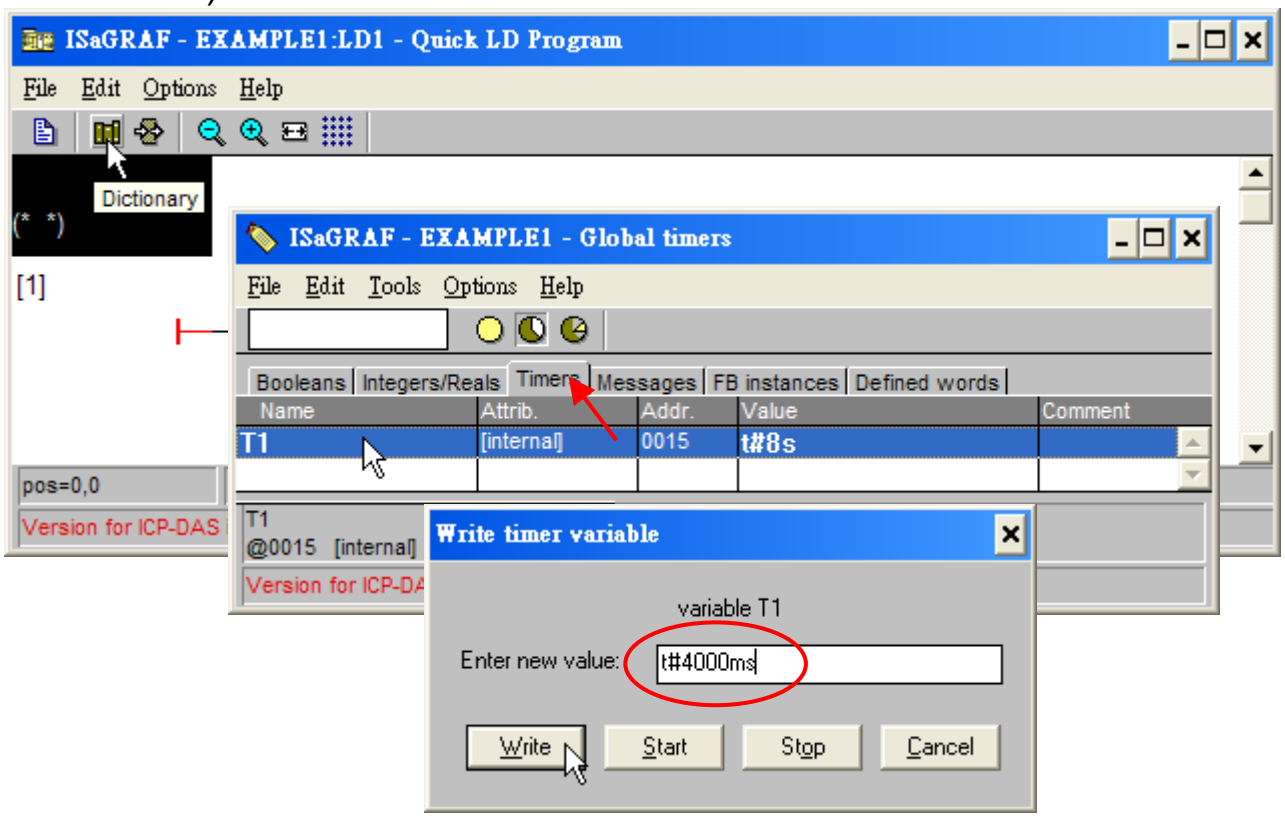
RUNNING THE SIMULATION PROGRAM

When you double click on "LD1" in the "ISaGRAF Debug Programs" window, the follow window should appear.

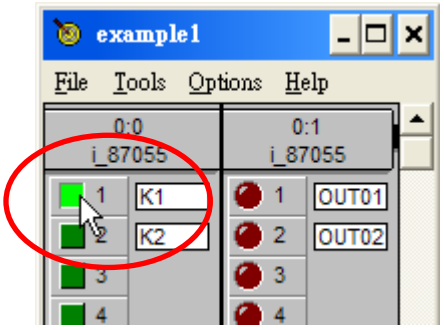


You can see outputs “OUT01” will blink in the period of 8 seconds.

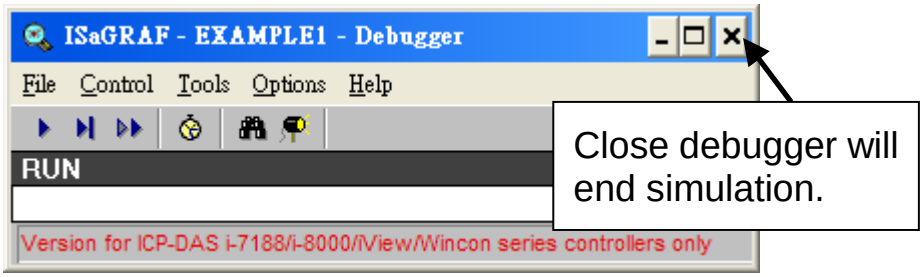
You can adjust the "T1" variable while the program is running. To accomplish this, click on the "Dictionary" icon which will open the "ISaGRAF Global Variables" window as shown in the first two pictures below. Click on “Timer” tab and then double click on “T1” to change the timer value to “T#4000ms” (this means 4000 ms). Then click on “Write”.



Now we are going to simulate the “K1” & “K2” input. Click on “K1” using the left button of the mouse.



To exit simulation, please close the debugger window.

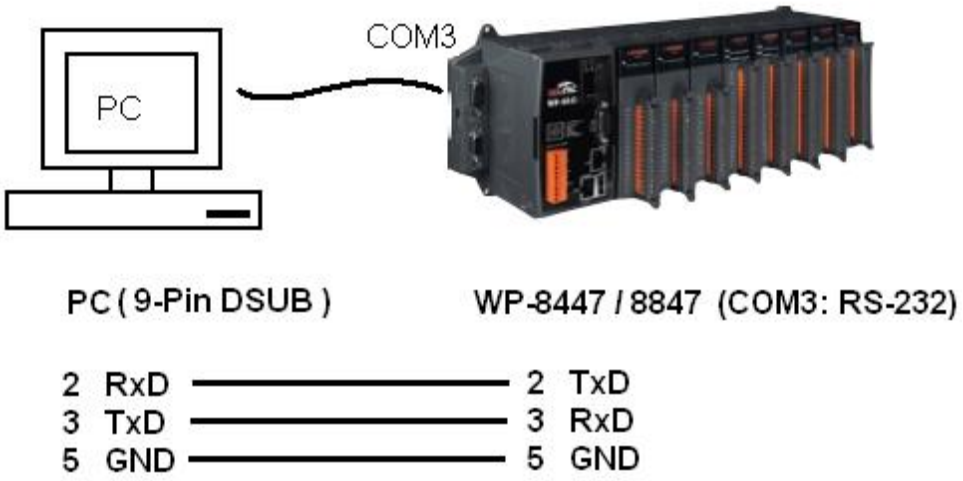


4.3 Download & Debug The Example Project

We have two ways to download the project to the WinPAC-8xx7. One is using Ethernet cable, the other one is using RS-232 cable. Here will show you the RS-232 way. (**Please refer to section 3.2.3.1 if you would like to download the project via Ethernet**)

WIRING THE HARDWARE

To begin this process, please install the hardware as below. The RS-232 cable wiring should be as below figure. (**Please make sure the "Modbus RTU Slave Port" is set as COM3 (refer to Appendix A.2, or it can only be download via Ethernet)**

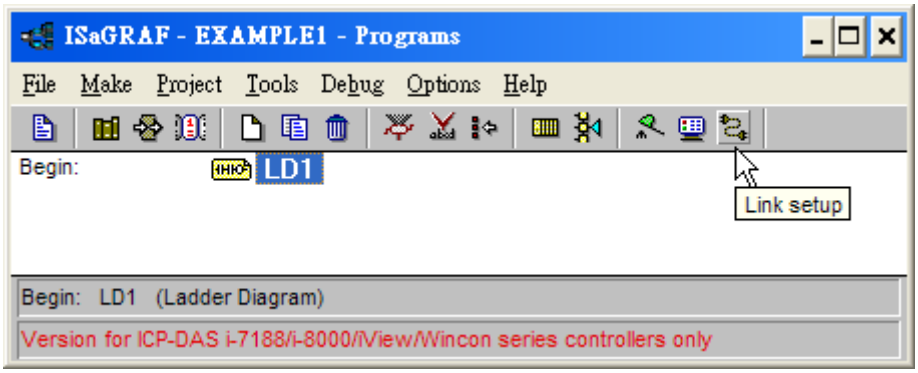


Note: The WP-8147 doesn't have COM3. Only WP-8447 / 8847 have.

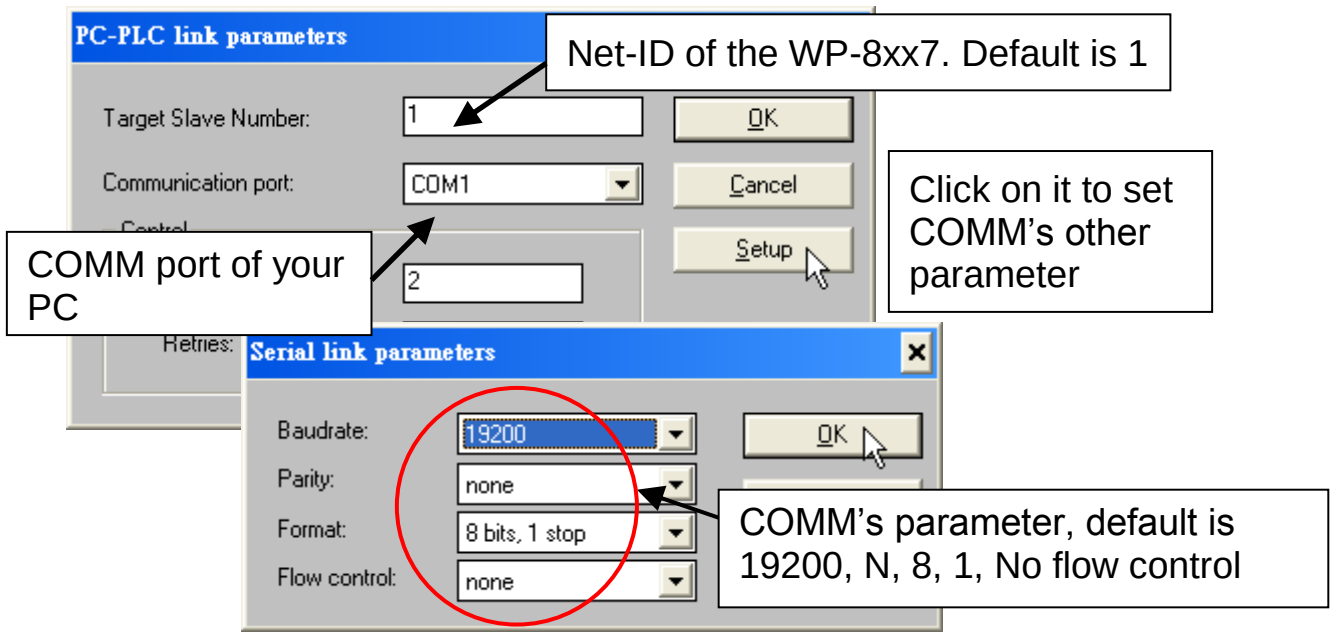
This section lists how to download the ISaGRAF program via RS-232 cable. However user may also use Ethernet cable to download program to the WinPAC-8xx7 (please refer to section 3.2.3.1)

SETUP LINK PARAMETERS

Click on the "Link Setup" icon in the "ISaGRAF Programs" window.



When you click on the "Link Setup" icon, the following window will appear. Please set the proper value.



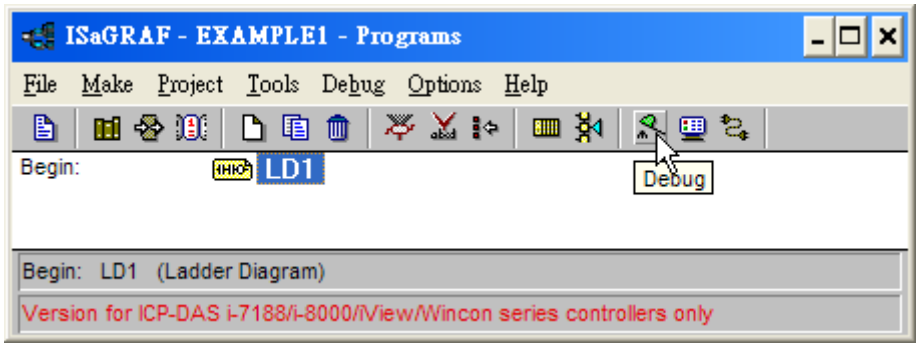
The RS-232 communication parameters for the target WP-8xx7 controller MUST be set to the same serial communication parameters for the development PC. For WP-8xx7 controllers (serial port communications), the default parameters for COM3 (RS-232) port are:

Baudrate:	19200
Parity:	none
Format:	8 bits, 1 stop
Flow control:	none

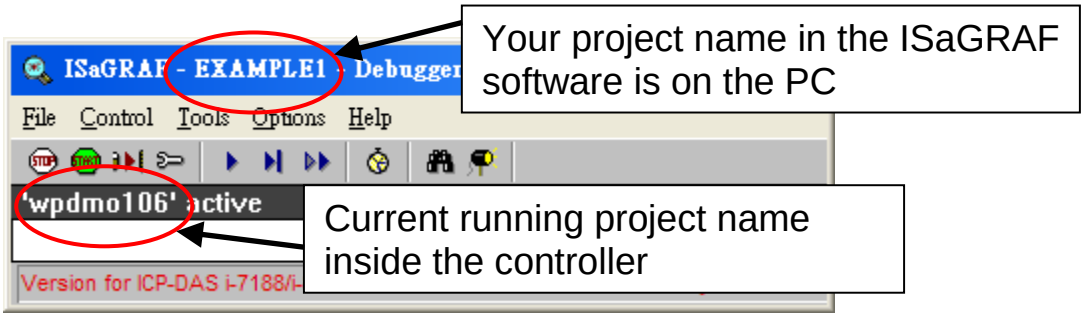
(Please refer to Appendix A.2 to setup COM3 as Modbus RTU slave port)

DOWNLOADING THE EXAMPLE PROJECT

Before you can download the project to the controller, you must first verify that your PC and the controller system are communicating with each other. To verify proper communication, click on the "Debug" icon in the "ISaGRAF Programs" window as shown below.



If the development PC and the WinPAC-8xx7 controller system are communicating properly with each other, the following window displayed below will appear (or if a program is already loaded in the controller system, the name of the project will be displayed with the word "active" following it.

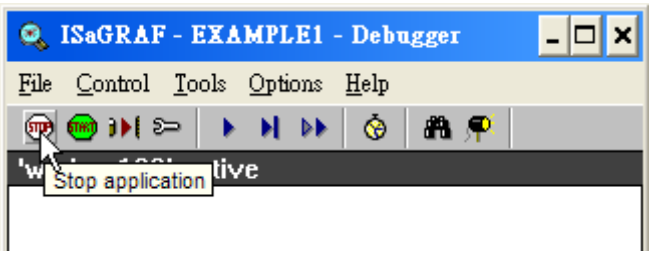


If the message in the "ISaGRAF Debugger" says "Disconnected", it means that the development PC and the controller system have not established communications with each other.

The most common causes for this problem is either the serial port cable not being properly configured, or the development PC's serial port communications DO NOT match that of the WP-8xx7 controller system.

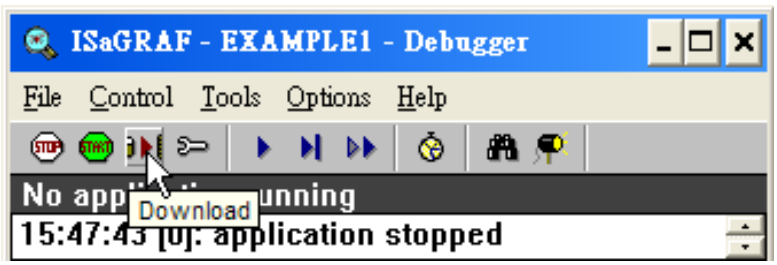
You may have to either change the serial port communication settings for the development PC (which may require changing a BIOS setting) or change the "Serial Link Parameters" in the ISaGRAF program.

If there is a project already loaded in the controller system you will need to stop that project before you can download the example project. Click on the "STOP" icon as illustrated above to halt any applications that may be running.

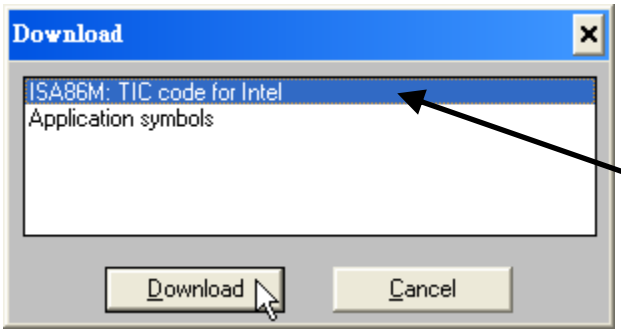


STARTING THE DOWNLOADING PROCESS

Click on the "Download" icon from the "ISaGRAF Debugger" window.

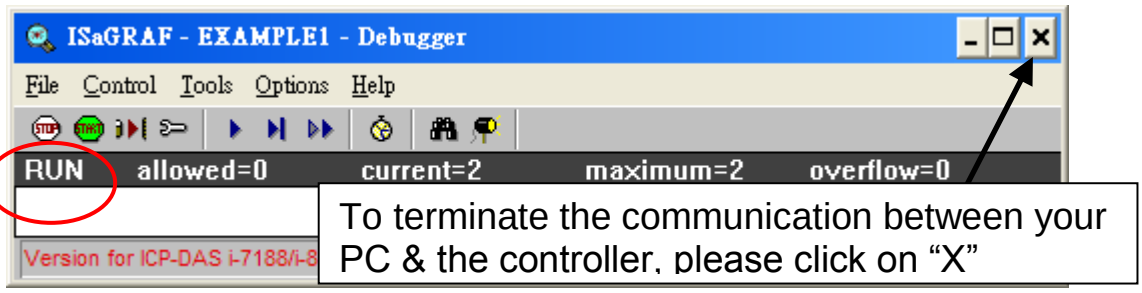


Then click on "ISA86M: TIC Code For Intel" from the "Download" window as shown below.



If “ISA86M: TIC code fort Intel” is not found here, that means the compiler option - “ISA86M: TIC code for Intel” is not checked. Please refer to section 4.2 to check it & re-compile the project again.

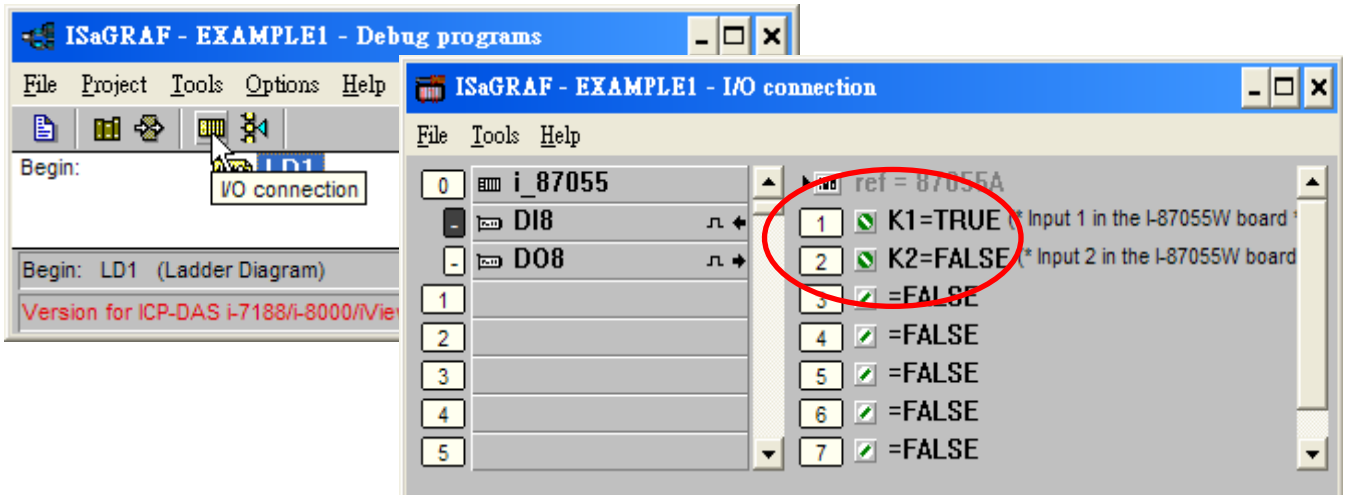
The example project will now start downloading to the WP-8xx7 controller system. A progress bar will appear in the "ISaGRAF Debugger" window showing the project downloading progress.



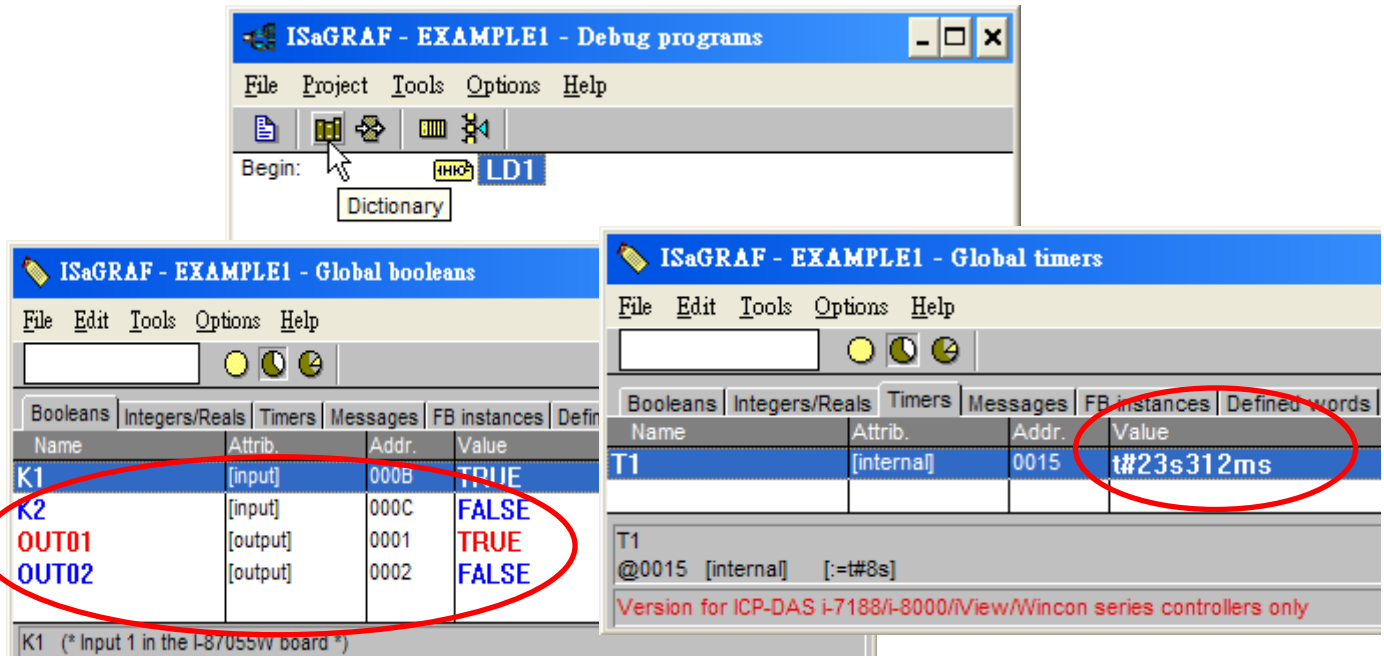
When the example project has successfully completed the downloading process to the W-8xx7 controller system the following two windows will appear.

RUNNING THE EXAMPLE LD PROGRAM

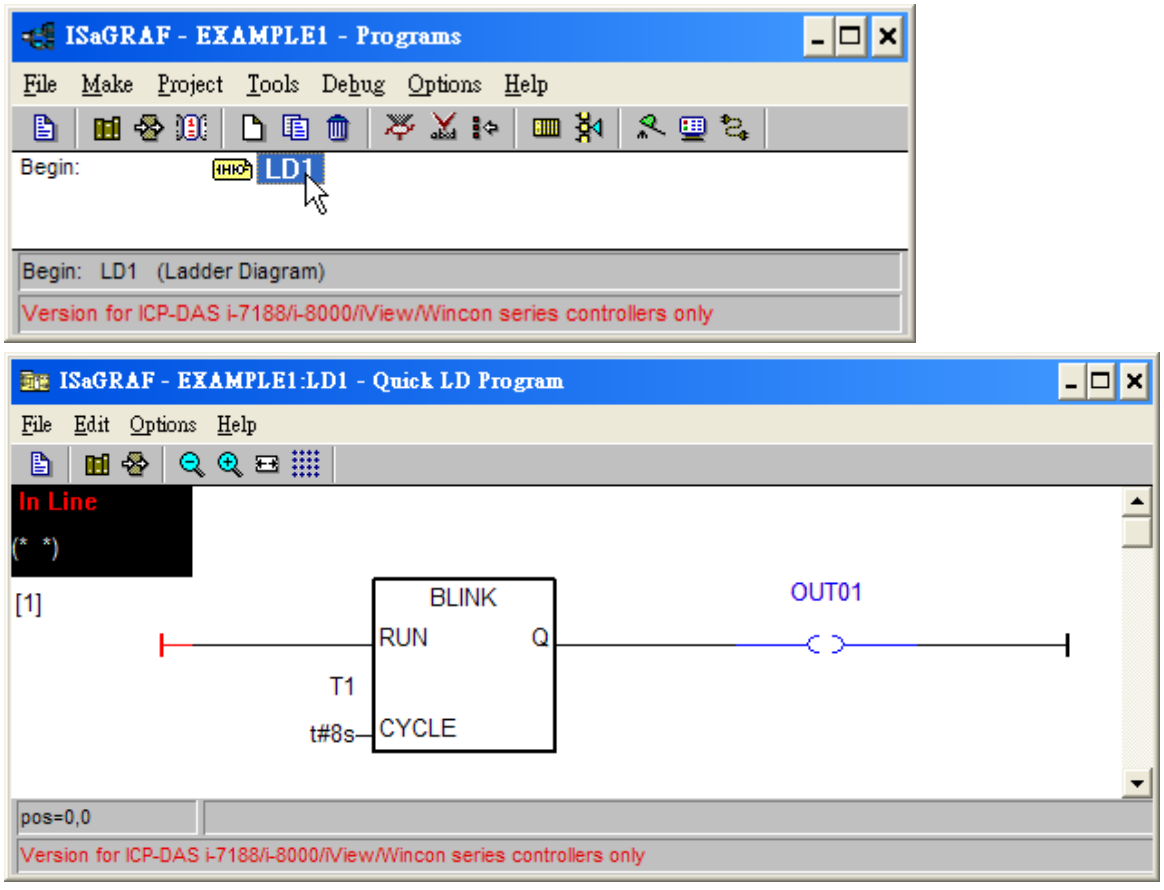
You can observe the real time I/O status from several ISaGRAF windows while you are running the example project. One of the windows is the "I/O Connections" window, which shows each of the inputs and outputs as assigned. Click on the "I/O Connections" icon in the ISaGRAF Debugger window to open the "I/O Connections" screen. You may switch ON/OFF the D/I on the front panel of the I-87055W I/O board to see what happens about “K1” & “K2”



You may also click on “Dictionary” to see the real time variable state.



Another VERY helpful window you can open is the "Quick LD Program" window. From this window you can observe the LD program being executed in real time.



4.4 Design The Web Page

After finishing the ISaGRAF project & download it to the WinPAC-8xx7, we are going to design the Web Page for this ISaGRAF project.

If you haven't practiced "Setting Up A Web HMI Demo" listed in the Chapter 3, it's better to do it once to get familiar with it.

We will use "**Microsoft Office FrontPage 2003**" **(or advanced version)** to build web pages in this manual. User may choose your prefer web page editor to do the same thing.

You may refer to the finished web pages of this example in the WP-8xx7 CD-ROM at design time. However it is better to do it one time by yourself to get more understanding.

WinPAC-8xx7 CD: \napdos\isagraf\wp-8xx7\wp_webhmi_demo\example1\

4.4.1 Step 1 – Copy The Sample Web HMI pages

This is a sample Web HMI pages in the WinPAC-8xx7 CD-ROM:
\napdos\isagraf\wp-8xx7\wp_webhmi_demo\sample\

Please copy this "sample" folder to your drive and rename it, for example, "**example1**".

The basic Web HMI files includes 2 folders and 3 DLL files and 4 htm files as below.

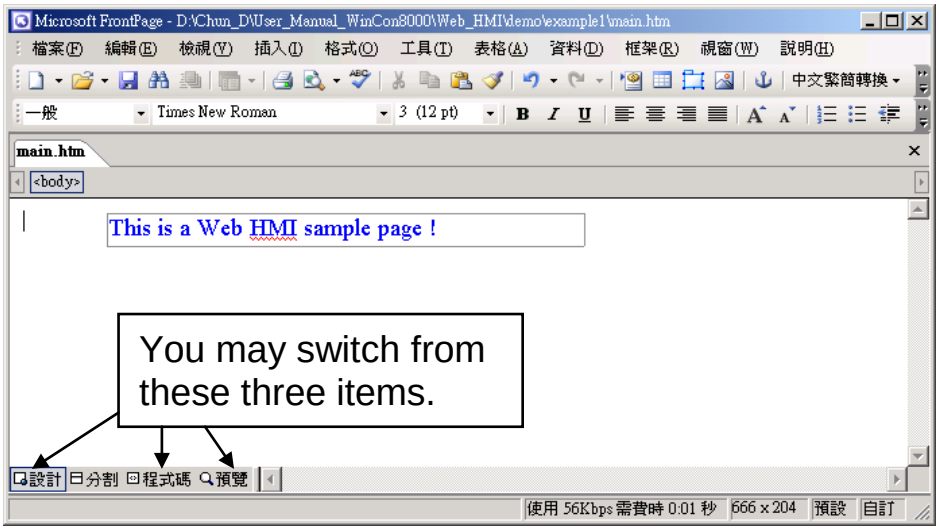
./img/	(default image files - *.jpg , *.bmp , *.gif)
./msg/	(default message files – wincon.js & xxerror.htm)
whmi_filter.dll	(three DLL files)
login.dll	
main.dll	
index.htm	(first default page)
login.htm	(the Web HMI welcome page)
menu.htm	(the page-menu page, normally on the left on the Internet Explorer)
main.htm	(first page when successfully login)

User may put his own image files into the folder named as "user_img". And put user-defined java script file or css file into the folder named as "user_msg". Other folder name is not acceptable by the Wincon Web HMI.

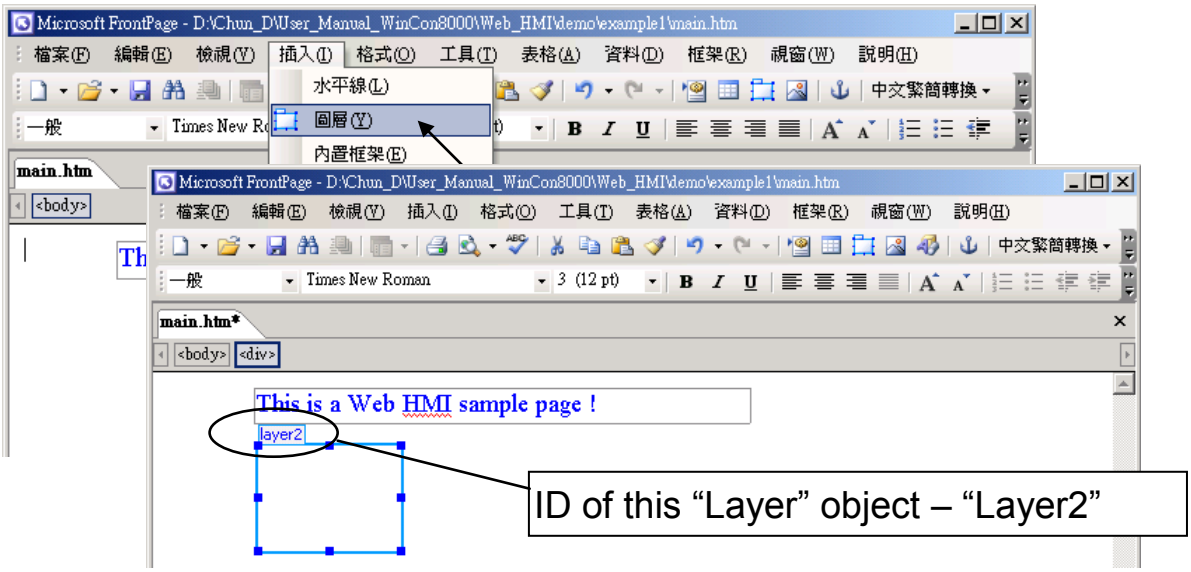
The “index.htm” file is the default entry page of the web server. User should not modify it. The “index.htm” re-directs to the “login.htm” file in 1 to 2 second when someone visits the WinPAC-8xx7 via the Internet Explorer.
User may modify the “login.htm” , “menu.htm” & “main.htm” to fit his own need. We will only modify the “main.htm” in this example.

4.4.2 Step 2 – Building The Main.htm

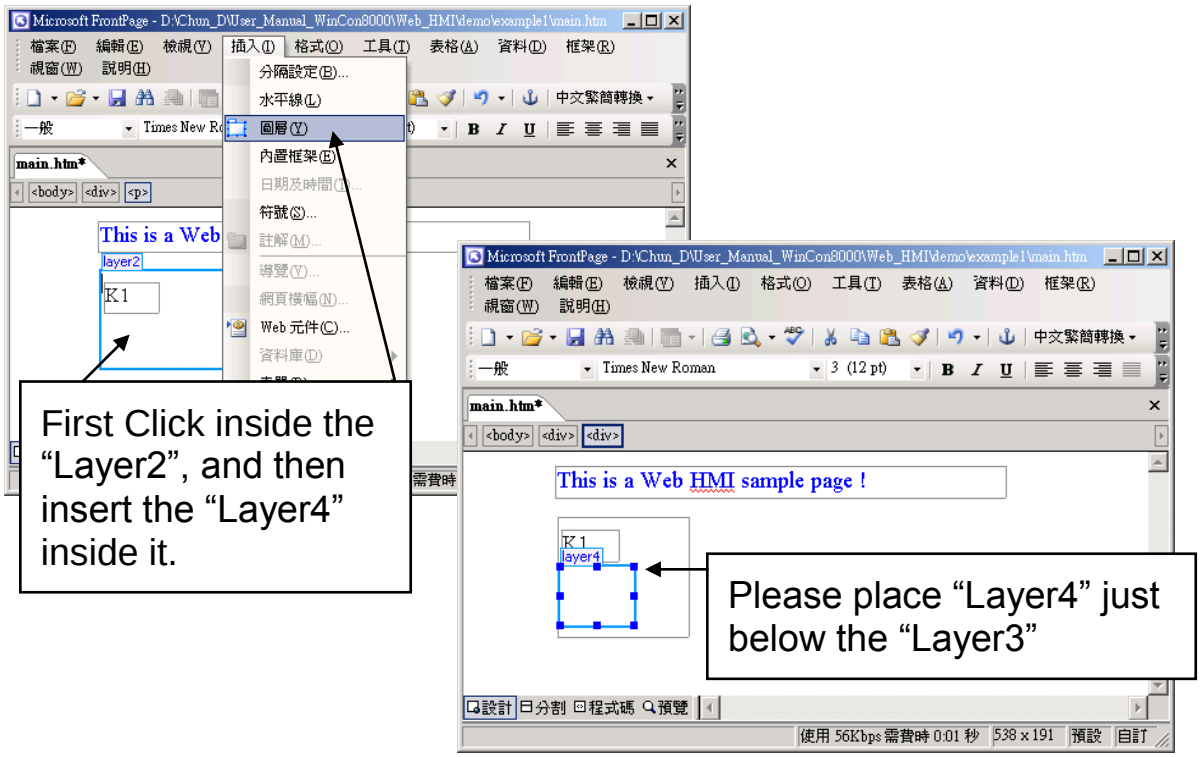
Please run the Microsoft Office FrontPage 2003 (or advanced version) and open the “main.htm”.



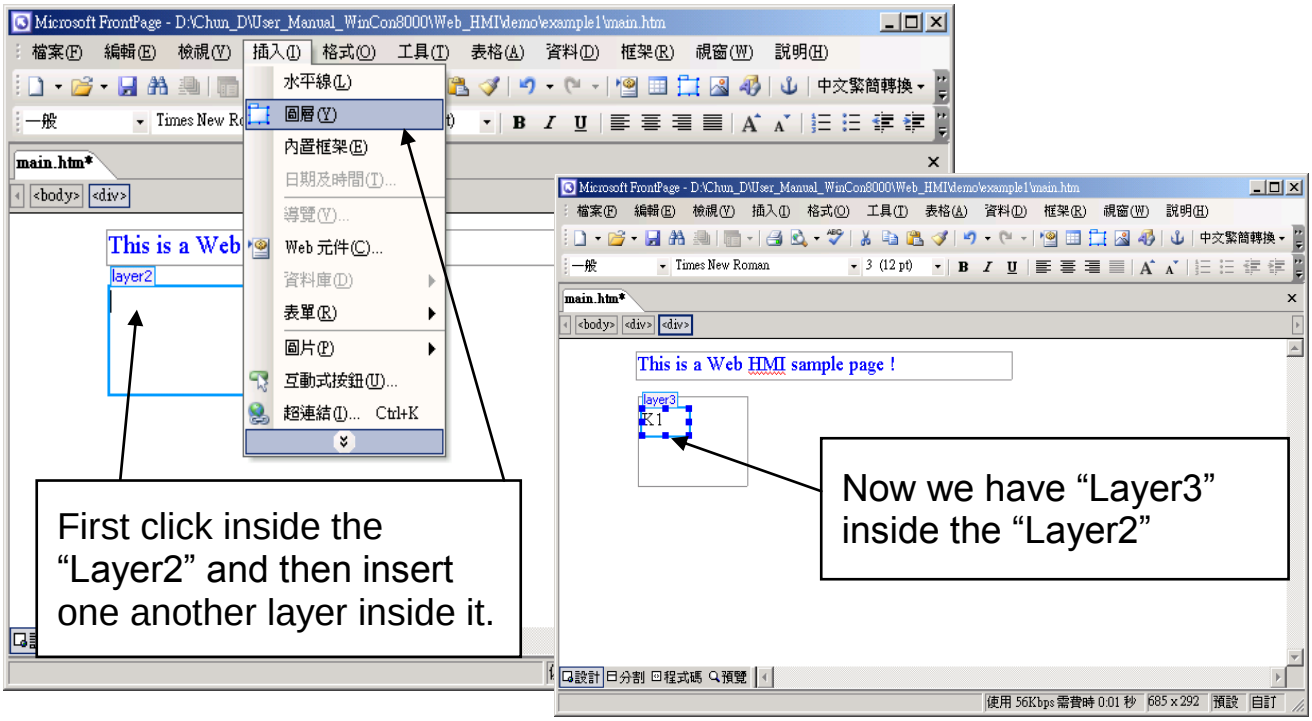
Please switch the window to design the page.
Please insert a layout object – “Layer” as below.



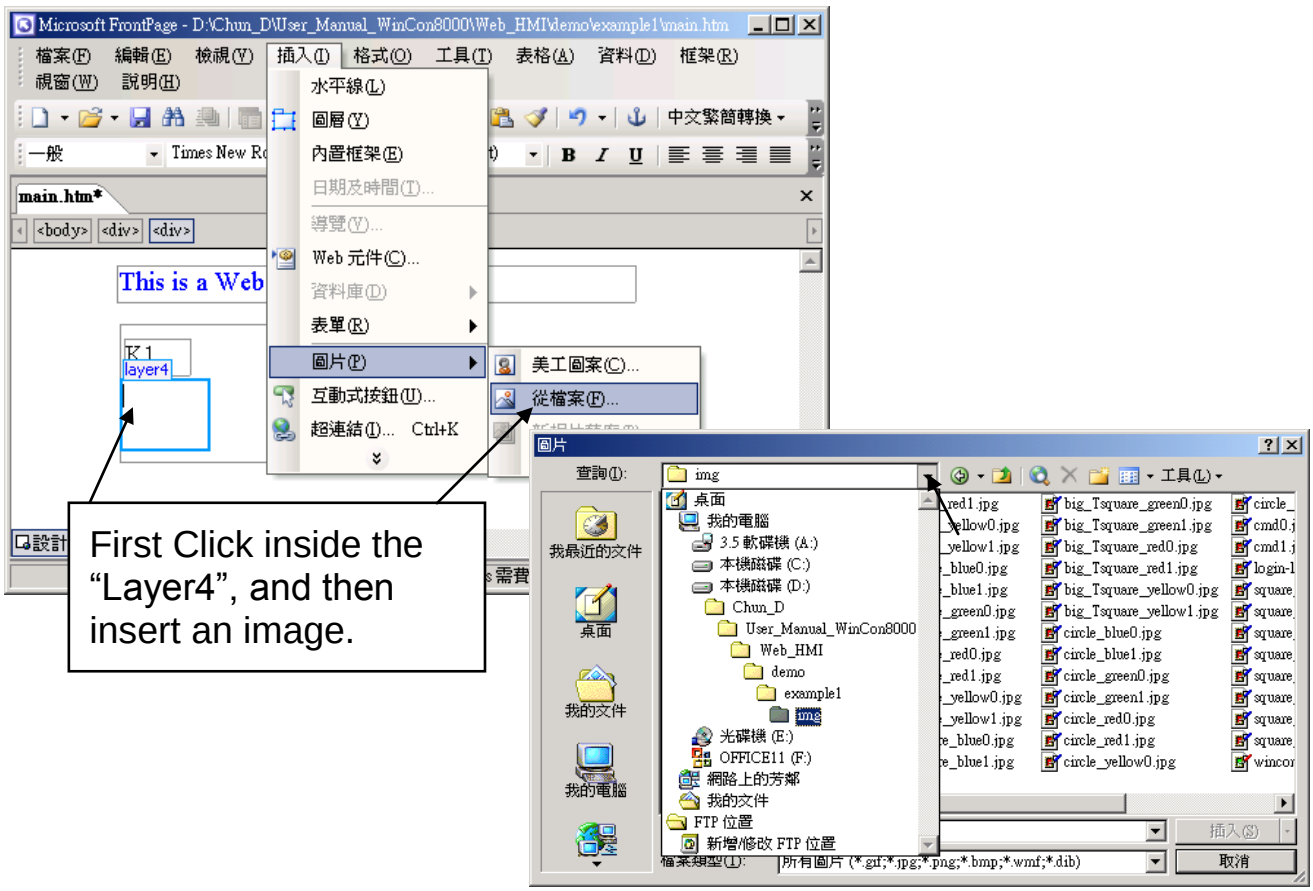
Click inside this “Layer” and then insert one another layer inside it as below. Please enter “K1” into the new created “Layer”.



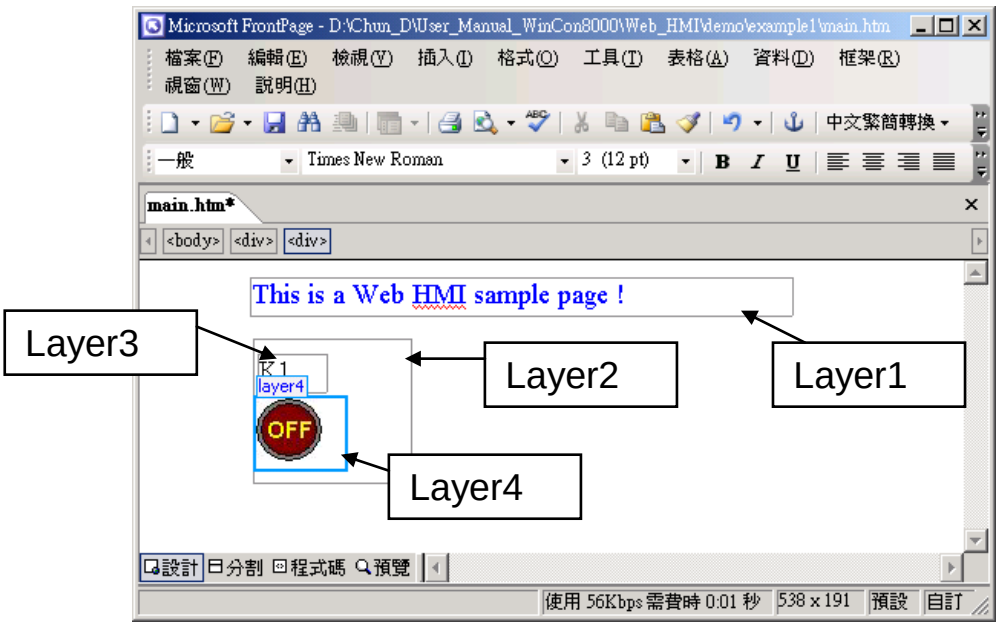
Follow the same former steps to insert one another “Layer” to be in just below the “Layer3” as below.



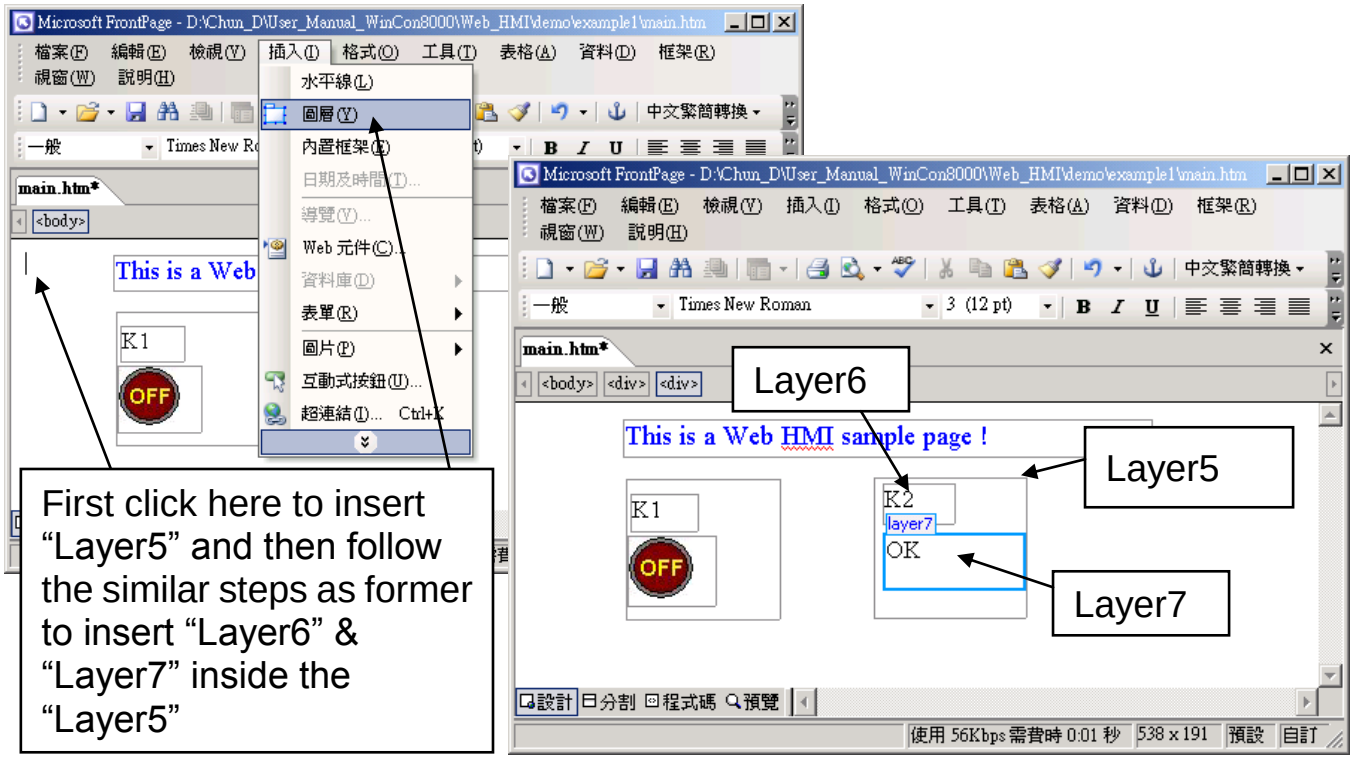
Inside the “Layer4”, we are going to insert one image file to it as below. The image file name is “./img/big_Tcircle_red0.jpg”. Please browse to the correct folder in your hard driver. Here we use “example1/img/” in this example.



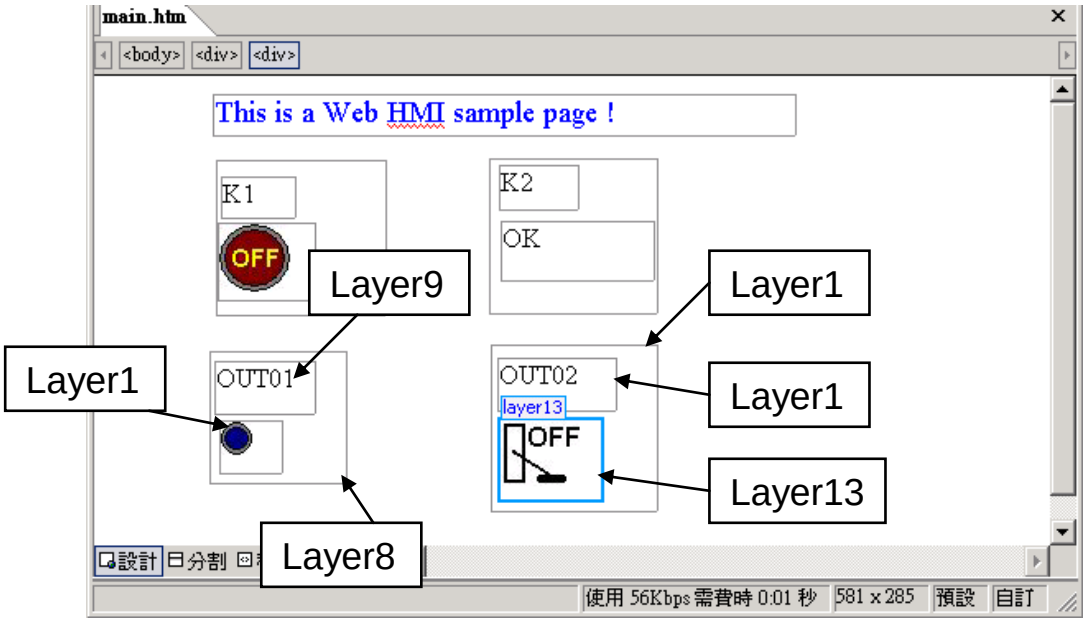
You will see a window as below.



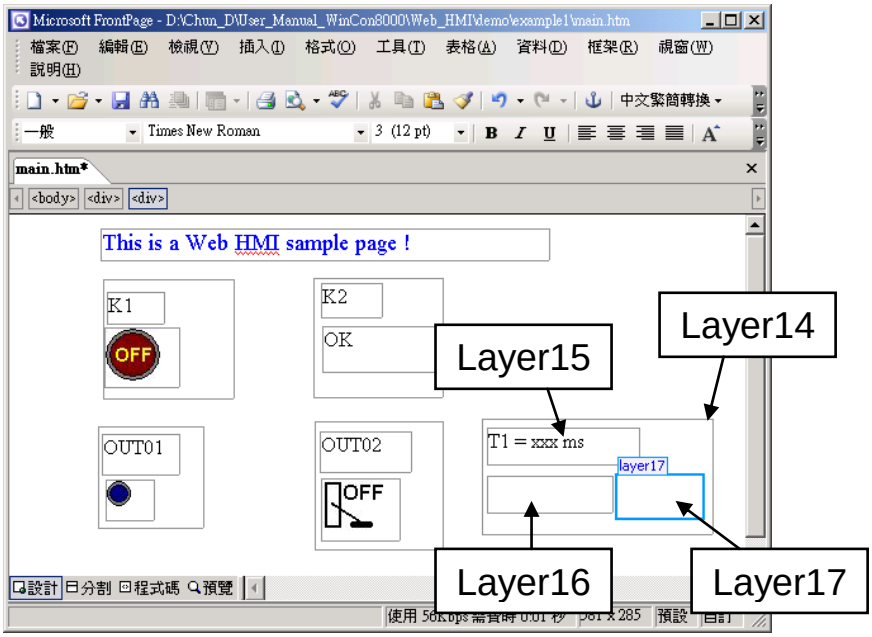
Please follow the similar steps to insert one another “Layer5” and one “Layer6” with a “K2” symbol inside it, and also a “Layer7” with a “OK” symbol inside it as below. We will use “K1” to display the state of the first input of the I-87055W board, and “K2” for its second input.



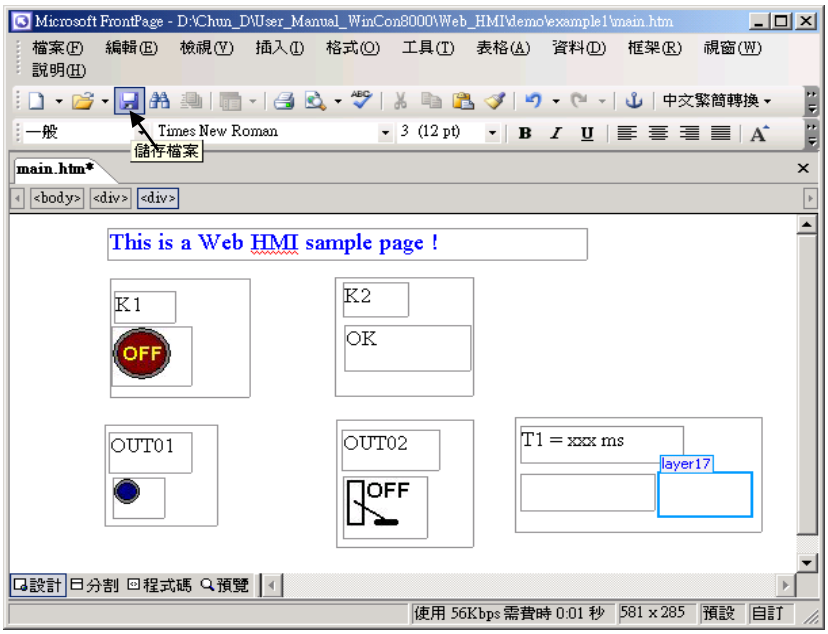
Please follow the similar steps to insert “OUT01” & “OUT02” as below. The OUT01 uses “./img/circle_blue0.jpg” as its image source, while OUT02 using “./img/cmd0.jpg”. We will use OUT01 to display the state of the first output of the I-87055W board, while “OUT02” is for controlling and displaying the second output of the I-87055W.



Now please insert one another “Layer14”. Inside the “Layer14” please insert one “Layer15” with a “T1 = xxx ms” symbol. And two empty Layers – “Layer16” & “Layer17” just below the “Layer15”. We will use T1 to display the Timer value “T1” in the ISaGRAF project.

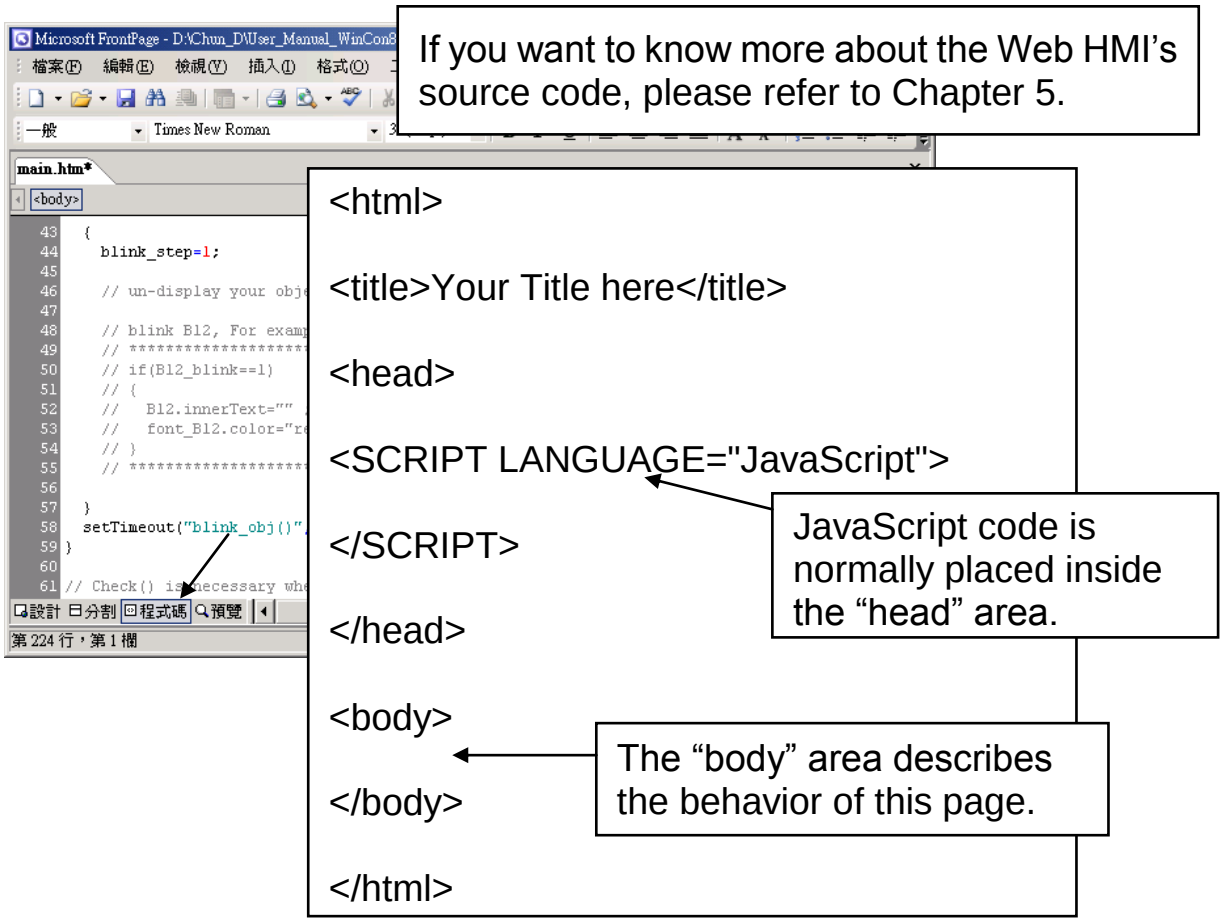


Click on “Save” to save this page.



4.4.3 Step 3 – Adding Control Code To The Main.htm

Please switch the window to the source code. A valid HTML document will contain the basic objects as below.



Please go to the <body> area and then modify the code as below.

Caption Area: Layer1
A Layer is starting with "<div " & ending with " " tag

```
<!-- Caption -->
<font color="blue" size="4">
<div style="position: absolute; width: 353px; height: 24px; z-index: 1; left: 73px;
top: 12px" id="layer1">
This is a Web HMI sample page !</div>
</font>
```

K1 Area: Layer2 to Layer4

```
<div style="position: absolute; width: 102px; height: 93px; z-index: 2; left: 75px;
top: 52px" id="layer2">
<div style="position: absolute; width: 44px; height: 24px; z-index: 1; left: 3px;
top: 10px" id="layer3">
K1</div>
```

<img **name=”B11”** border="0" src="img/big_Tcircle_red0.jpg" width="43" height="41">
<p>

Please insert name=”B11” just after the “<img “

K2 Area: Layer5 to Layer7

K2

****
** OK **
****

<p>

Please modify “OK ” to become

 OK

OUT01 Area: Layer8 to Layer10

OUT01

<img **name=”B1”** border="0" src="img/circle_blue0.jpg" width="19" height="20">
<p>

Please insert name=”B1” just after the “<img “

OUT02 Area: Layer11 to Layer13

OUT02

<img **style=”cursor:hand” name=”B2” onclick="ON_OFF(form_B2, form_B2.B2, boolean_val[2])"** border="0" src="img/cmd0.jpg" width="50" height="40">

<form name="form_B2" method="post" action="./main.dll">
<input name="BEGIN" type="hidden">
<input name="B2" type="hidden" value="0">
<input name="END" type="hidden">
</form>

<p>

Please insert
Style=”cursor:hand” name=”B2” onclick="ON_OFF(form_B2, form_B2.B2, boolean_val[2])"
just after the “<img “ tag

Please insert
<form name="form_B2" method="post" action="./main.dll">
<input name="BEGIN" type="hidden">
<input name="B2" type="hidden" value="0">
<input name="END" type="hidden">
</form>

T1 Area: Layer14 to Layer17

T1 = xxx ms

Please modify “T1 = xxx ms ” to become
T1 = xxx ms

```
<form name="form_L21" method="post" action="./main.dll">
  <input name="BEGIN" type="hidden">
  <input name="L21" type="text" size="8" value="xxx">
  <input name="END" type="hidden">
</form>
```

Please insert below code inside “Layer16”

```
<form name="form_L21" method="post" action="./main.dll">
   <input name="BEGIN" type="hidden">
   <input name="L21" type="text" size="8" value="xxx">
   <input name="END" type="hidden">
 </form>
```

```
 </div>
<div style="position: absolute; width: 67px; height: 33px; z-index: 3; left: 106px;
top: 44px" id="layer17">
  <input type="button" value="Enter" onclick="Check_L21( )">
 </div>
<p> </div>
```

Inside the “Layser17”, please insert

```
<input type="button" value="Enter" onclick="Check_L21( )">
```

We have finished the code in the <body> </body> area.

Now please go to the “head” area.
In the “head” area, please modify the sample code to be as below.

```
// variable to record object's blink state, 0:not blink, 1: blink, For example:
// ********************************
var B12_blink=0;  // init as 0:not blink
// ********************************
// function to blink object
var blink_step=0;
function blink_obj()
{
  if(blink_step==1)
  {
    blink_step=0;

    // display your object here
    // blink B12, For example:
    // ********************************
    if(B12_blink==1)
    {
      B12.innerText="Error !" ;
      font_B12.color="red";
    }
    // ********************************
  }
  else
  {
    blink_step=1;

    // un-display your object here
    // blink B12, For example:
    // ********************************
    if(B12_blink==1)
    {
      B12.innerText="" ;
      font_B12.color="red";
    }
    // ********************************
  }
  setTimeout("blink_obj()", blink_period);
}
```

The “Error !” symbol will blink when the K2 = True in this example. Please un-mask the code inside these 3 areas.

We need a function “Check_L21 to check the entered T1 value and post it to the Wincon. Please un-mask the sample code to be as below.

// form sample, to check value of L21 & then post val to controller
// For example:
// **

```
function Check_L21()
{
  var val=form_L21.L21.value;
  if(val>12000 || val<4000)
  {
    alert("T1's value should be in the range of 4000 to 12000");
    return;
  }
  Check(form_L21);  // post value to the controller
}
```

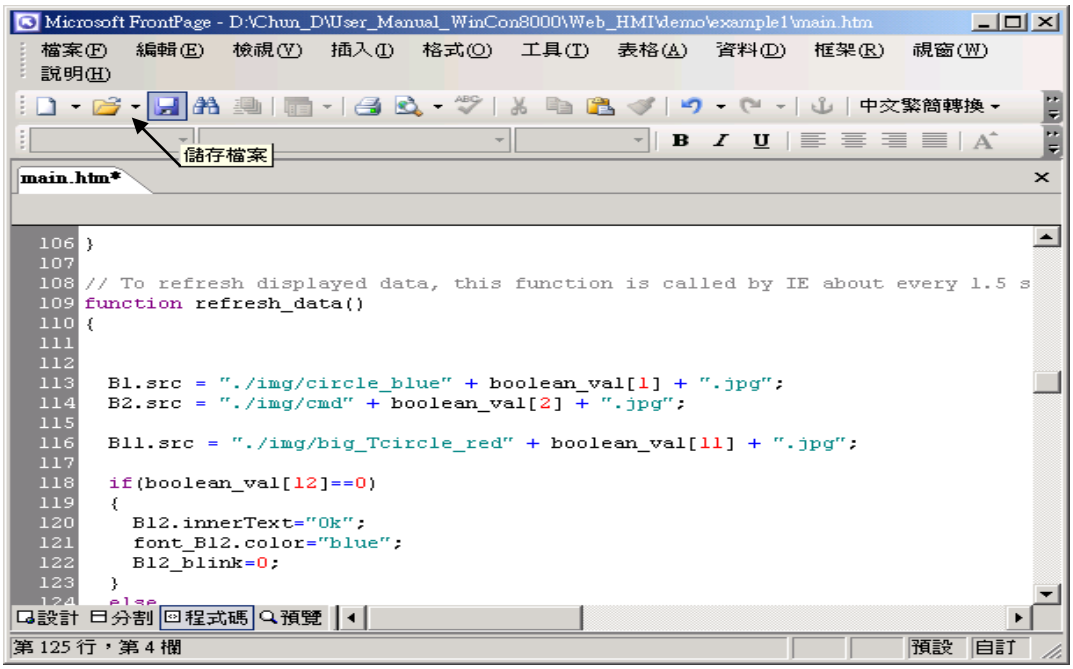
// **

And also inside the “refresh_data() “ function, please insert below code.

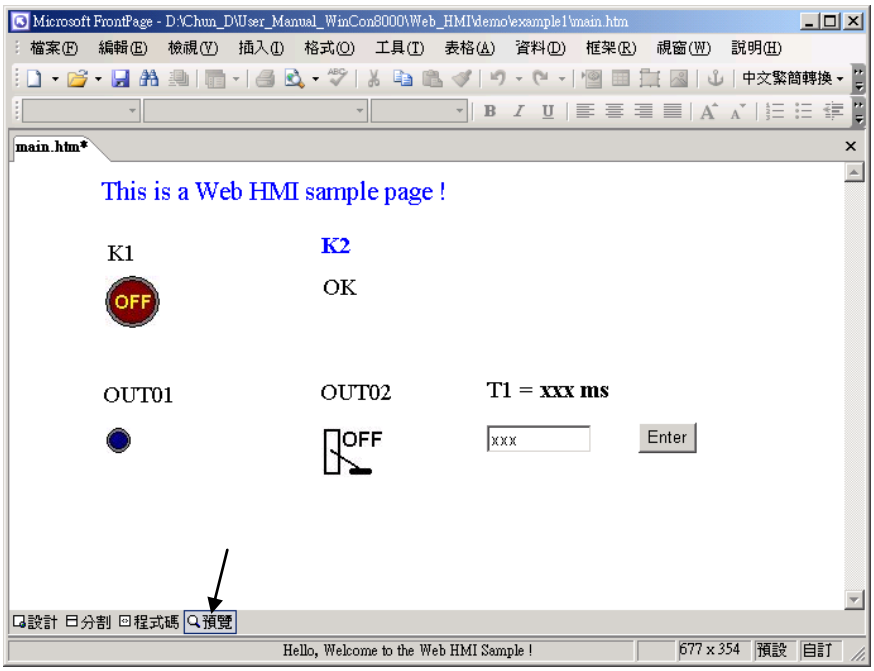
// To refresh displayed data, this function is called by IE about every 1.5 sec later

```
function refresh_data()
{
  B1.src = "./img/circle_blue" + boolean_val[1] + ".jpg";
  B2.src = "./img/cmd" + boolean_val[2] + ".jpg";

  B11.src = "./img/big_Tcircle_red" + boolean_val[11] + ".jpg";

  if(boolean_val[12]==0)
  {
    B12.innerText="Ok";
    font_B12.color="blue";
    B12_blink=0;
  }
  else
  {
    B12_blink=1;
  }
  T1.innerText=timer_val[21] + " ms";
}
```

Now we have finished all the code. Please save it.



You may click on “Preview” to simulate its run time behavior.



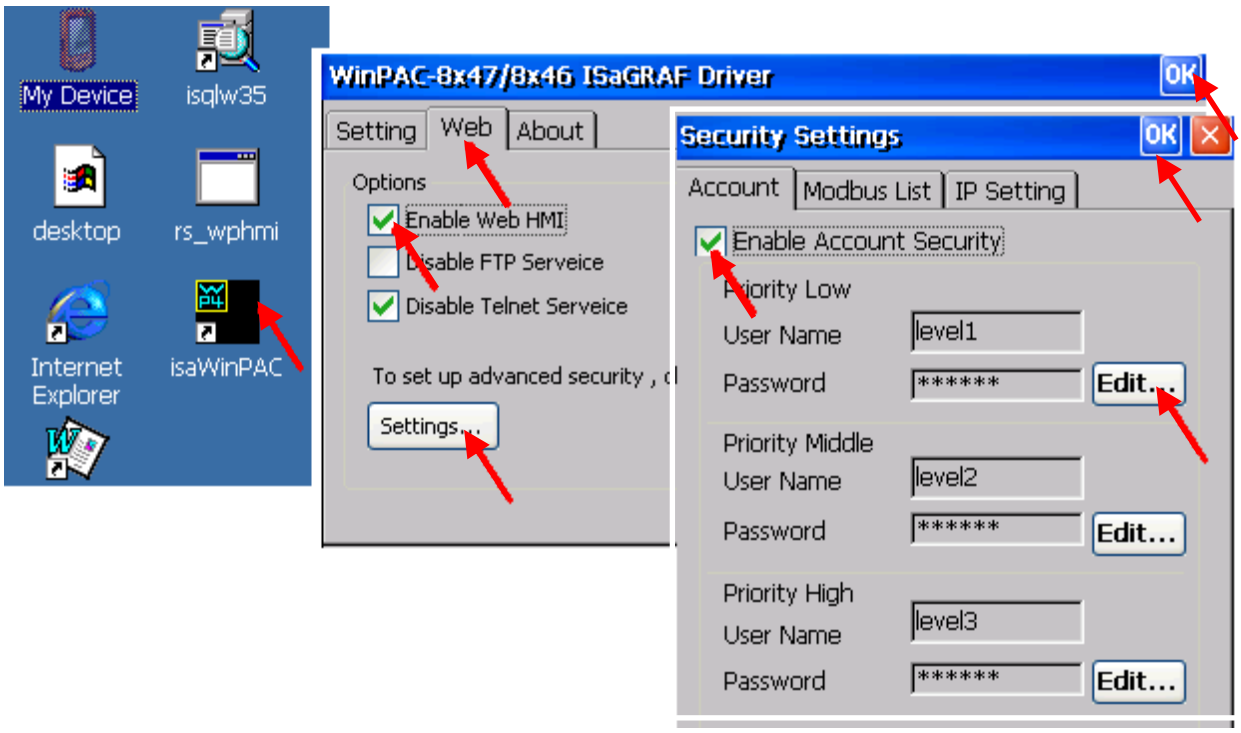
4.4.4 Step 4 – Download Web HMI Pages To The Controller

The steps are similar as listed in Section 3.2. If you haven't practiced "Setting Up A Web HMI Demo" listed in the Chapter 3, it's better to do it once to get familiar with it.

First set the web options

Check on "Enable Web HMI" and then click on "Setting", Please check on "Enable Account Security" and then click on "Edit" to set (username , password). **Then remember to click on "OK"**

Note: If "Enable Account Security" is not checked, any user can easily get access to your WinPAC-8xx7 through the Internet Explorer.



And then, please copy all files in this example1 to the controller

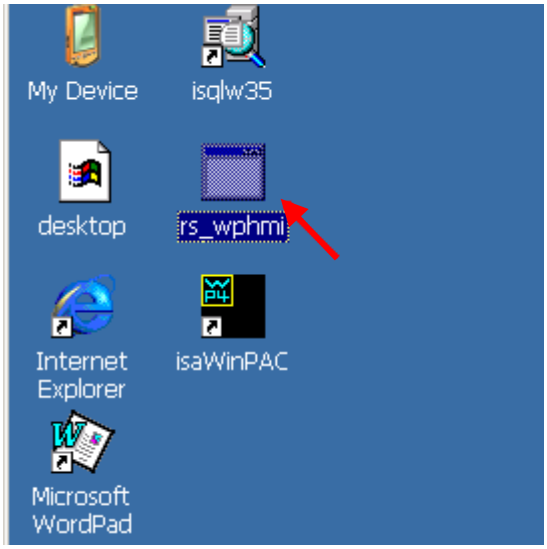
<your hard drive>:\example1\ *.*

to the WinPAC-8xx7's

Micro_SD\Temp\HTTP\WebHMI\

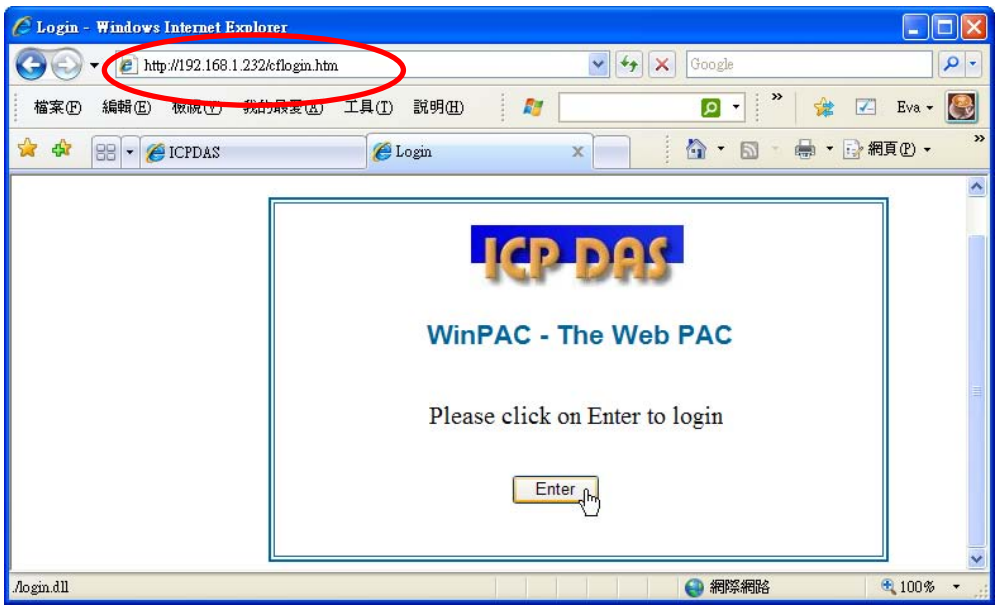
Since the Web Pages are modified or new copied, please run "rs_wphmi.exe" to reset the Web server. **The "rs_wphmi.exe" must be run every time when**

user has modified any file in the WP-8xx7's \Micro_SD\Temp\HTTP\WebHMI

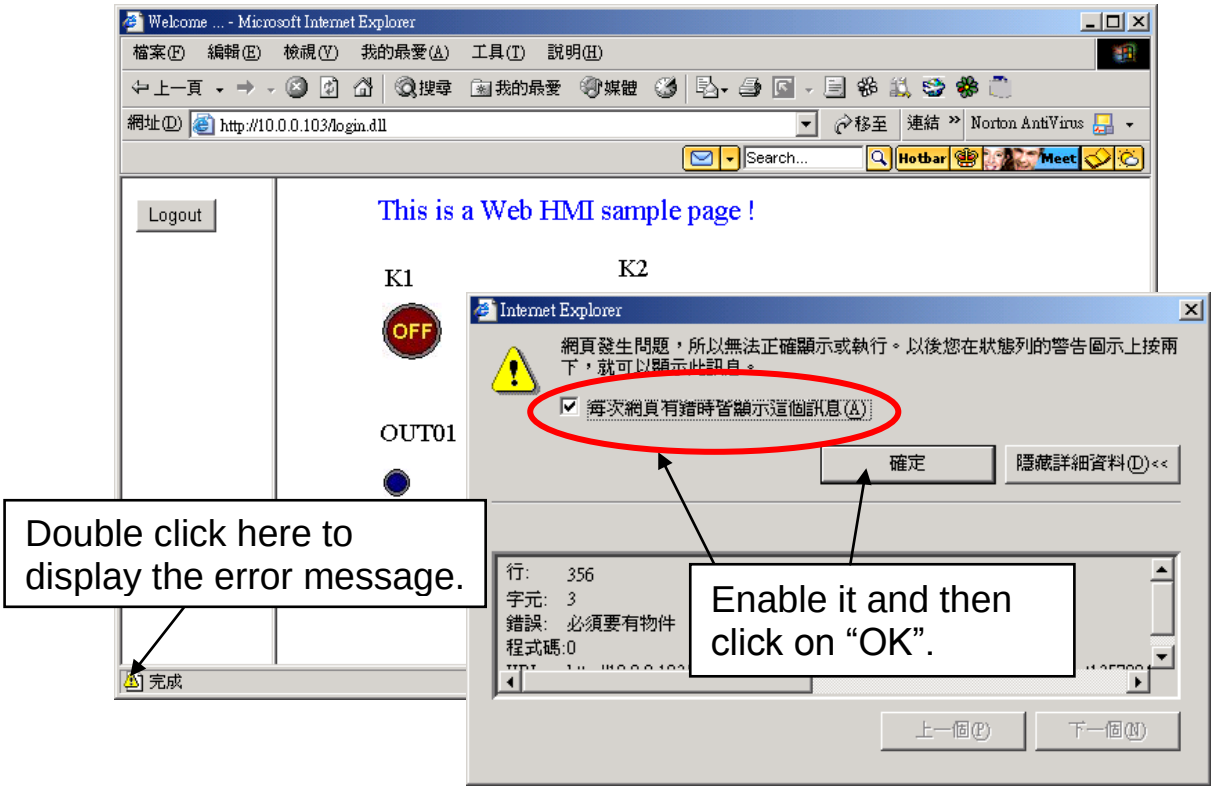


Show Time:

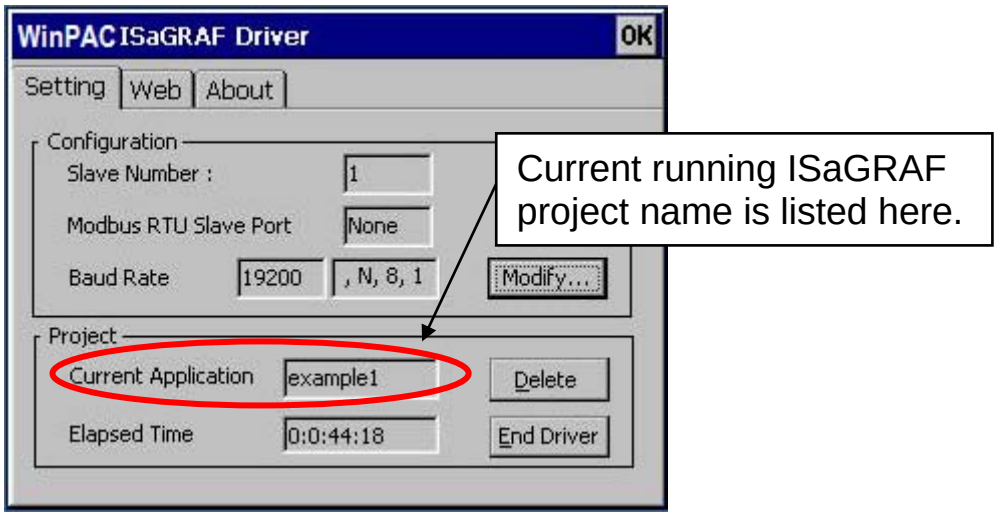
Please run Internet Explorer (Rev. 6.0 or higher), key in the IP address of your WP-8xx7.
For example: 61.218.42.10 or http://61.218.42.10



If there is something wrong with the web page. You may enable the below item to display the debug message every time it has error.



And also check if your ISaGRAF project already downloaded to the controller (Section 4.3 or section 3.2.3). And do you assign the correct Modbus Network address to the respective ISaGRAF variables? (Section 4.1.5).



Chapter 5 Web HMI Basics

The WinPAC-8xx7 (or WP-8xx7) is the abbreviation of the WinPAC-8147 / 8447 / 8847.
The WinPAC-8xx6 (or WP-8xx76) is the abbreviation of the WinPAC-8146 / 8446 / 8846.

Important Notice:

1. **WP-8xx7 / 8xx6 supports only High profile I-8K and I-87K I/O cards in its slot 0 to 7**. (**Refer to wp-8xx7_datasheet.pdf in the WP-8xx7 CD: \napdos\isagraf\wp-8xx7\english_manu**
2. Please always set a fixed IP address to the WinPAC-8xx7. (No DHCP)

Note:

1. This chapter describes the programming basics for the Web HMI. We will not focus on the HTML basics. If you want to know more about the HTML programming, the best way is to "buy a HTML related book" from the bookstore. There are a lot of books doing this job.
2. The Web HMI only supports the basic HTML tags. It doesn't support ASP, PHP or JSP or other Page Server language.
3. Please do not use <frameset> </frameset> , <frame> </frame> in the Web HMI.
4. Note: The object name, object ID, code, variable name and function name is case sensitive. For example, refresh_data( ) and Refresh_data( ) is different.
5. There are more than ten Web HMI examples in the WinPAC-8xx7's CD-ROM. Please refer to section 3.1.

5.1 Basic Files For The Web HMI

The basic Web HMI files include 2 folders and 3 DLL files and 4 htm files as below.

./img/	(default image files - *.jpg , *.bmp , *.gif)
./msg/	(default message files – wincon.js & xxerror.htm)
whmi_filter.dll	(three DLL files)
login.dll	
main.dll	
index.htm	(first default page)
login.htm	(the Web HMI welcome page)
menu.htm	(the page-menu page, normally on the left on the Internet Explorer)
main.htm	(first page when successfully login)

User may put his own image files into the folder named as “user_img”. And put user-defined javascript file or css file into the folder named as “user_msg”. Other folder name is not acceptable by the Wincon Web HMI.

The “index.htm” file is the default entry page of the web server. User must not modify it. The “index.htm” re-directs to the “login.htm” file in 1 to 2 seconds when someone visits the WinPAC-8xx7 via the Internet Explorer.

User may modify the “login.htm” , “menu.htm” and “main.htm” to fit the requirement.

5.2 Login.htm

Login.htm is the first welcome page when a user visiting in. It can be modified. Below is the basic code for the login.htm

```
<html>
<head>

<title>Login</title>

<meta http-equiv=pragma content=no-cache>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" >

<script language="JavaScript">
var random_val=123;
function get_random_val()
{
  var rightNow = new Date();
  random_val += 323456789*rightNow.getMinutes() +
                107654321*(rightNow.getTime()%1000);
  setTimeout("get_random_val()", 197); // repeat call
}

//check if username and possword are empty
function validate(fm)
{
  setKey(fm);
  return true;
}

//Embed key whille submitting
function setKey(fm)
{
  var rightNow = new Date();
  cookieVal = random_val+rightNow.getTime();
  fm.key_.value = cookieVal;
}
```

This line is only for the “Login.htm” , please do not apply to other pages, for example, the “menu.htm” & “main.htm” & other .htm pages.

Please apply your charset here.
For example,
English: UTF-8
Traditional Chinese: big5
Simplified Chinese: gb2312
or other language

```
</script>

</head>

<body onload="get_random_val()">

<div style="position: absolute; width: 332px; height: 34px; z-index: 5; left: 147px;
top: 27px" id="layer1">
Welcome !</div>

<div style="position:absolute; width:122px; height:38px; z-index:4; left: 171px;
top: 95px;"  id="layer2">

  <form name="form1" action="./login.dll" method="post">
    <input type="hidden" name="key_">
    <input type="submit" name="Submit" value="  Enter  " style="cursor:hand"
onClick="return validate(this.form)">
  </form>

</div>

</body>

<!-- To ensure no-cache work -->
<head>
<meta http-equiv=pragma content=no-cache>
</head>

</html>
```

get_random_val( ) should be always called at the beginning of the Login.htm . It is the entry point of the Login.htm

Your caption is here.

"form1" is necessary

You may modify " **Enter** " to your own word. For example "請進" .This may require user to modify the related "charset" at the beginning of this page.

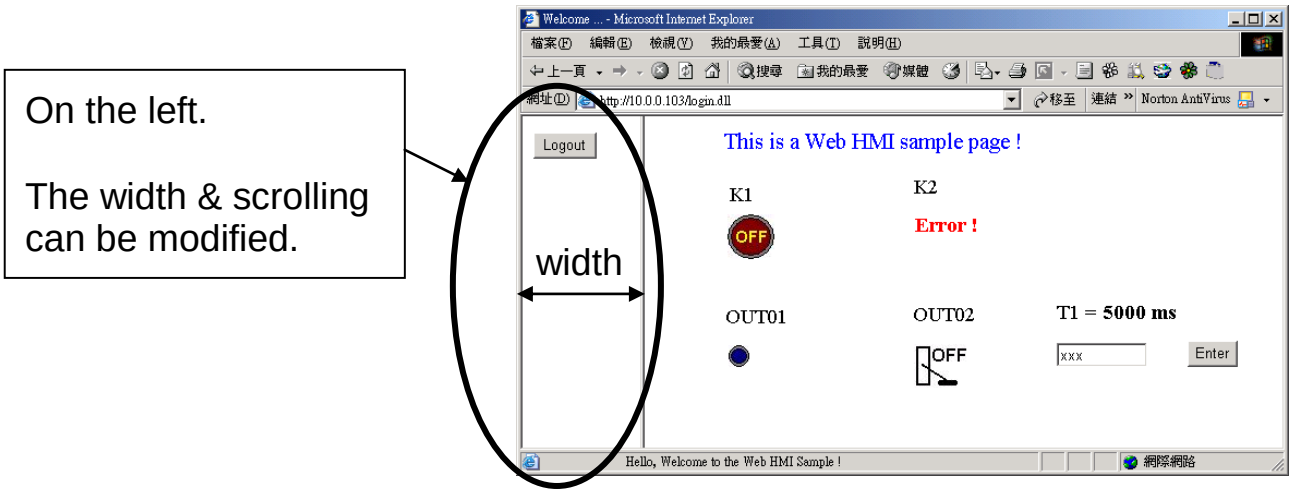
This code is only for the "Login.htm" , please do not apply to other pages, for example, the "menu.htm" & "main.htm" & other .htm pages.

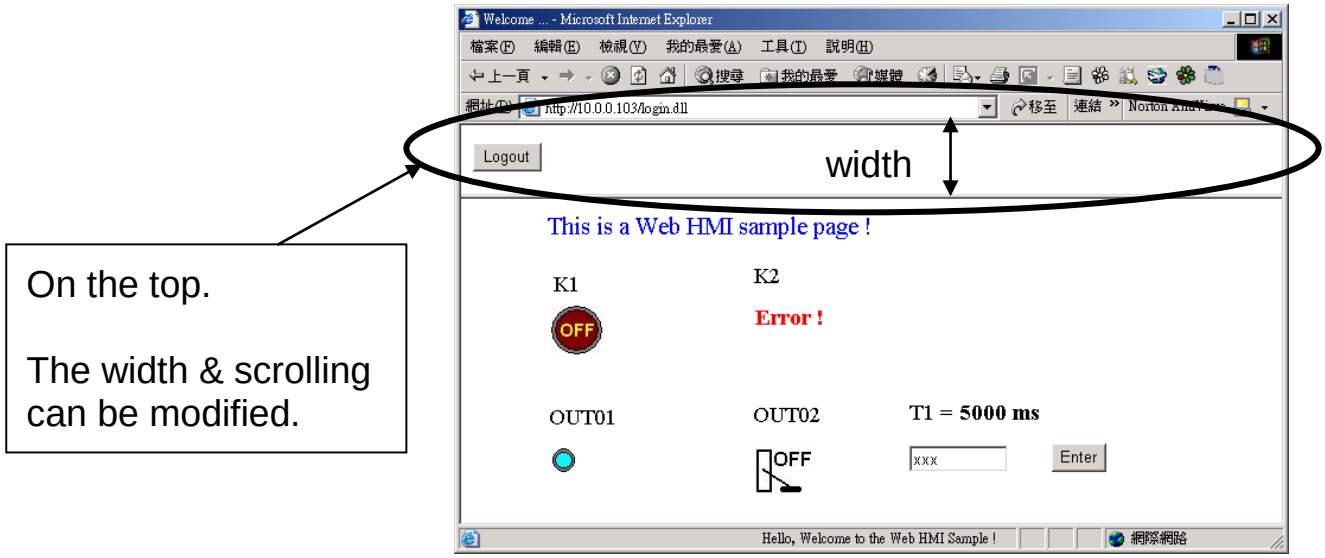
That's all the login.htm need. You can insert more images or text to it. Only remember to keep its basic code.

5.3 Menu.htm

Note:
If you want to know more about the multi-page application, there are two demos in the WinPAC-8xx7 CD-ROM:
\napdos\isagraf\wp-8xx7\wp_webhmi_demo\wphmi_05 & wphmi_05a . The "wphmi_05" place its page-menu on the left, while "wphmi_05a" on the top.

The "Menu.htm" defines the Page-menu of the Web HMI especially for the multi-page application. The page-menu can place only on the left or on the top.





Below is the basic code for the menu.htm

```
<!-- top_or_left=1 , scrolling=0 , width=60 , resize=1 -->
<html>
<head>
<title>Title1</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" >
<SCRIPT LANGUAGE="JavaScript" src="./msg/wincon.js"></SCRIPT>
<SCRIPT LANGUAGE="JavaScript">
function start1()
{
  A_11();
}
function refresh_data()
{
  if(run_at_pc==1) return;
}
</SCRIPT>
</head>
<body onload="start1()">
<!-- Logout button -->
<form name="form_logout" method="post" action="./login.dll">
  <input style="cursor:hand" name="CMD" type="submit" value="Logout"
onClick="return logout(this.form)">
</form>
</body>
</html>
```

The first row is not a comment, it defines the Page-Menu behavior
- top_or_left: 1:Top , 0:Left
- scrolling: 1:Yes , 0:No
- width: width of the Menu Frame, 0 – 999 (unit is pixel)
- resize: 1:Yes , 0:No

This row is necessary for menu.htm , main.htm & other multi-pages

Please apply your charset here.
For example,
- English: UTF-8
- Simplified Chinese: gb2312
- Traditional Chinese: big5
- or other language

start1( ) is the entry point of the menu.htm

form_logout is for the logout button.

Note:
If you want to know more about the multi-page application, there are two demos in the WinPAC-8xx7 CD-ROM: \napdos\isagraf\wp-8xx7\wp_webhmi_demo\wphmi_05 & wphmi_05a . The “wphmi_05” place its page-menu on the left, while “wphmi_05a” on the top.

5.4 Main.htm

5.4.1 A Simple Main.htm Example

Before going further in the main.htm, first take a look at a simple main.htm example. This example only display a “Hello !” message when successfully login, nothing else.

```
<html>
<head>
<title>Title1</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" >

<SCRIPT LANGUAGE="JavaScript" src="./msg/wincon.js"></SCRIPT>


<SCRIPT LANGUAGE="JavaScript">
show_scroll_word(200,"Hello, Welcome to the Web HMI Sample !");

function refresh_data()
{
}


</SCRIPT>
</head>

<body onLoad="init()">

<font color="blue" size="4">
<div style="position: absolute; width: 353px; height: 24px; z-index: 1; left: 73px;
top: 12px" id="layer1">   Hello !</div>
</font>

</body>
</html>
```

Please apply your charset here. For example, English: UTF-8 Simplified Chinese: gb2312, Traditional Chinese: big5 , or other language

This line is necessary for menu.htm , main.htm & other multi-pages

Calling show_scroll_world( ) will display a moving word at the bottom of the Internet Explorer. Here 200 means 200 ms. You may make it slower, for example, using 500.

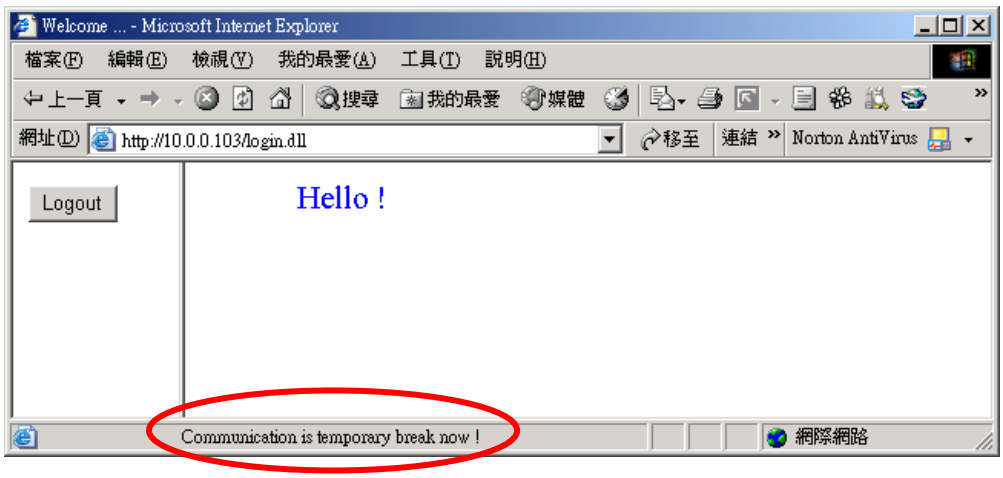
refresh_data( ) is called when the Internet Explorer has received the requested data from the controller. It is called in the period about 1.25 to 5 seconds depends on the communication quality.

init( ) is the entry pint of the main.htm & other multi-pages.

A layout object is starting with “<div” & ending at “ ” tags.
Here only show a message “Hello !”

You may replace the main.htm in the WinPAC-8xx7 CD-ROM:
\napdos\isagraf\wp-8xx7\wp_webhmi_demo\sample
to the above main.htm & download it to the controller (refer to section 4.4.4).
You will see the below window when you login successfully.

User may try to plug out the Ethernet cable of the WinPAC or of your PC. You will see it show “Communication is temporary break now !” in about 10 seconds. When you plug the cable back, the communication will be recovered in about 10 to 45 seconds.



If the communication broken time exceeds 120 seconds, it will show the below message. You have to close the Internet Explorer & open it again to re-login.

5.4.2 More About The refresh_data( ) Function And Dynamic Data

Note: The code, variable name and function name is case sensitive. For example, refresh_data( ) is correct, however Refresh_data( ) is not correct.

The refresh_data( ) function must always apply in the main.htm and other multi-pages. It is called when the Internet Explorer has received the requested data from the controller. The calling period is about 1.25 to 5 seconds depends on the communication quality

The refresh_data( ) is often used for refreshing the dynamic data. For example, the boolean value , integer value, timer value or float value of the variables in the ISaGRAF project.

The Internet Explorer can access to the data in the ISaGRAF project only when they are assigned a unique Modbus Network Address No (refer to section 4.1.5). The Web HMI only accepts Network Address No in the range of 1 to 1024. The data without a Network Address No (No. = 0) or not in the range of (1 to 1024) is not accessible by the Internet Explorer.

The main.htm and other multi-pages can use the below variable array to access to the ISaGRAF's data (case sensitive). The identifier appeared in the [ ] is the related Network Address No. For example boolean_val[2] means the boolean value of the ISaGRAF boolean data which is assigned with the Network Address No. = 2.

boolean_val	boolean value in the ISaGRAF
word_val	word value in the ISaGRAF, -32768 to +32767
float_val	real value in the ISaGRAF, for ex, 1.234 , -0.456E-02
timer_val	timer value in the ISaGRAF, unit is ms, max = 86399999 (< 1 day)
string_val	message value in the ISaGRAF, max string length is 255

To access to long integer value (32-bit integer) please use get_long_val( ) function. For example, get_long_val(11) , get_long_val(13) , get_long_val(15).

get_long_val( )	long integer value in the ISaGRAF, -2147483648 to +2147483647

Note:
The long integer, timer and float variable's Network Address No. must occupy 2 No. in the ISaGRAF project (refer to section 4.2 of "User's Manual of ISaGRAF Embedded Controllers" or in the CD-ROM: \napdos\isagraf\wp-8xx7\english_manu\ " User_Manual_I_8xx7.pdf").

That means if you assign a Network Address No.= 11 to a Real type variable(or Timer or integer will have 32-bit value – larger than 32767 or smaller than -32768), the next No. 12 should not assigned to any other variable in the ISaGRAF project. However you may assign No.=13 to one another variable.

5.4.2.1 Displaying Dynamic Boolean Data

Demo example: whmi_02 and whmi_05 (section 3.1)

Let's look back to the refresh_data function. If user want to display the dynamic boolean value, the below code can be used.

```
…
function refresh_data()
{
  B1.src = "./img/circle_blue" + boolean_val[1] + ".jpg" ;
}
…

<body onLoad="init()">
…

<div style="position: absolute; width: 214px; height: 53px; z-index: 2; left: 102px;
top: 79px">
<img name="B1" src="img/circle_blue0.jpg"></div>

…
</body>
```

The action of the image object "B1" is defined here.

if boolean_val[1]=1, it display image "B1" as "img/circle_blue1.jpg"
if boolean_val[1]=0, it display image "B1" as "img/circle_blue0.jpg"

The layout (or location) of the image object "B1" is defined here by the "<div" and " " tags.

The declaration of image "B1" is defined here by the "img" tag & name="B1" src= … ← "src=" defines the initial value of B1

5.4.2.2 Displaying Dynamic Float & Word & Timer Data

Demo example: wphmi_01 , wphmi_03 and wphmi_05 (section 3.1)

If user want to display the dynamic float value, the below code can be used.

```
…
function refresh_data()
{
  F21.innerText = float_val[21] ;
}
…
```

The action of the Text object “F21” is defined here.

If want to display Word data, please use “word_val[ ]“
If want to display Timer data, please use “timer_val[ ]“.
For ex, F21.innerText = timer val[21] + “ ms”;

```
<body onLoad="init()">
…
```

The layout (or location) of the Text object “F21” is defined here by the “<div” “ ” tags.

```
<div style="position: absolute; width: 214px; height: 53px; z-index: 2; left: 102px;
top: 79px">
<b id="F21"> xxxx </b> </div>
…
</body>
```

The declaration of Text object “F21” is defined here by the “<b” tag & id=”F21” & “” tag initial value of this F21 is “xxxx”

5.4.2.3 Displaying Dynamic Long Integer Data

Demo example: wphmi_03 and wphmi_05 (section 3.1)

If user want to display the dynamic long integer value (32-bit format), the below code can be used.

```
function refresh_data()
{
  L11.innerText = get_long_val(11) ;
}
…
<body onLoad="init()">
…
```

The action of the Text object “L11” is defined here.

The layout (or location) of the Text object “L11” is defined here by the “<div” and “ ” tags.

```
<div style="position: absolute; width: 214px; height: 53px; z-index: 2; left: 102px;
top: 79px">
<b id="L11"> xxx </b> </div>
…
</body>
```

The declaration of Text object “L11” is defined here by the “<b” tag and id=”L21” and “” tag , the initial value of this L11 is “xxx”.

5.4.2.4 Displaying Dynamic String Data

If user want to display the dynamic string value (max length is 255), the below code can be used.

```
…
function refresh_data()
{
  S31.innerText = string_val[31] ;
}
…
<body onLoad="init()">
…
<div style="position: absolute; width: 214px; height: 53px; z-index: 2; left: 102px;
top: 79px">
<b  id="S31"> empty </b> </div>

…
</body>
```

The action of the Text object “S31” is defined here.

The layout (or location) of the Text object “S31” is defined here by the “<div” and “ ” tags.

The declaration of Text object “S31” is defined here by the “<b” tag and id=”S31” and “” tag, the initial value of this S31 is “empty”.

5.4.2.5 Trigger A Boolean Object To Blink

Demo example: wphmi_02 and wphmi_05 (section 3.1)

Some application may need a message to blink when the boolean value changes. For example, If boolean_val[12] is False, it means “OK”. However if boolean_val[12] is True, it means “Error !” . User may want to make this “Error !” blink to attract viewer’s attention.

The below code can do this job.

```
…
var blink_period=500;

setTimeout("blink_obj()", blink_period);

var B12_blink=0;   // init as 0:not blink

var blink_step=0;

function blink_obj()
{
  if(blink_step==1)
```

The blinking period, unit is ms

Setup a timer to handle the blinking action

1: to blink , 0: no blink

```
  {
    blink_step=0;
    if(B12_blink==1)
    {
      B12.innerText="Error !" ;
      font_B12.color="red";
    }
  }
  else
  {
    blink_step=1;
    if(B12_blink==1)
    {
      B12.innerText="" ;
      font_B12.color="red";
    }
  }
  setTimeout("blink_obj()", blink_period);
}
```

Blink step 1:
To display “Error !” in red color.

Blink step 2:
To display “” (nothing) in red color.

```
…function refresh_data()
{
  if(boolean_val[12]==0)
  {
    B12.innerText="Ok";
    font_B12.color="blue";
    B12_blink=0;
  }
  else
  {
    B12_blink=1;
  }
}
…
<body onLoad="init()">
…
<div style="position: absolute; width: 214px; height: 53px; z-index: 2; left: 102px;
top: 79px">
```

The action of the Text object “B12” is defined here.
If boolean_val[12]=0, no blink.
If boolean_val[12]=1, blink.

The layout (or location) of the Text object “B12” is defined here by the “<div” and “ ” tags.

```
<font  id=”font_B12” color=”blue” size=”3”>

<b  id="B12">OK</b>

</font>

</div>
…
</body>
```

The ““ & “” tags can be used for controlling the font’s color and font’s size.

The declaration of Text object “B12” is defined here by the “<b” tag and id=”B12” and “” tag, the initial value of this B2 is “OK”

5.4.2.6 Displaying Float Value With Fixed Digit Number Behind The “.” Symbol

Demo example: wphmi_06 and wphmi_07 (section 3.1)

The float_str1(para1 , para2) function can convert float value to a string with fixed digit number behind the dot "." symbol

para1 is the float value to be converted, for ex, 1.234567
para2 is the digit number behind the “.” dot symbol, 0 to 6
for ex, float_str1(1.234567, 3) return “1.234” ,
float_str1(1.234567, 2) return “1.23”

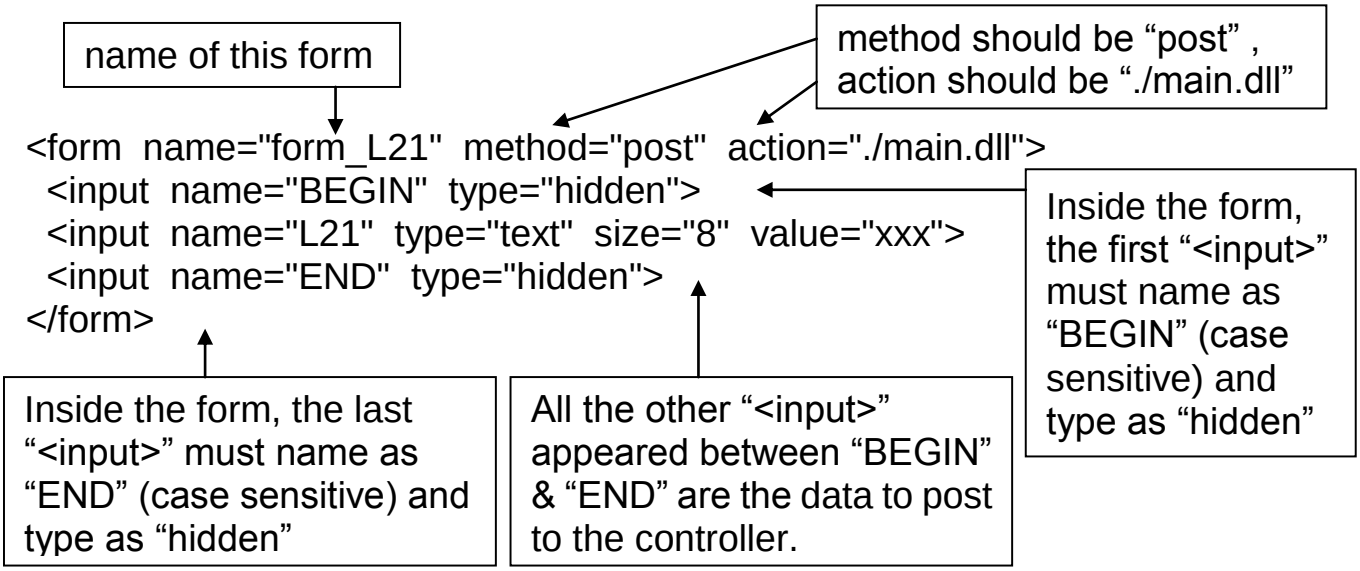
```
…
function refresh_data()
{
  F21.innerText = float_str1( float_val[21] , 3) ;
}
```

Convert float val at Network Address 21 to a string with digit number = 3 behind the “.” dot symbol.

5.4.3 Post Data To The Controller

The former section 5.4.2 listing how to get and display data from the controller. This section focuses on posting data to the controller, in other word to control the WinPAC via the Internet Explorer.

To set a new value to the boolean, word, long integer, float , timer and string variables in the ISaGRAF project, we need “form” object appeared in the main.htm or other multi-pages. A “form” object looks like as below.



The “<input>” name to control the WinPAC’s data must follow below format. The number followed behind the first letter should be in the range from 1 to 1024. This number is point to the variable name in the ISaGRAF project with the same Modbus Network Address No.

B	point to the ISaGRAF boolean data , for ex, B5 , B109
W	point to the ISaGRAF word data (-32768 to +32767), for ex, W9 , W1001
L	point to the ISaGRAF long integer data (-2147483648 to +2147483647), for ex, L21. This “L” Also point to the ISaGRAF timer data
F	point to the ISaGRAF real data, for ex, F13 , F235
S	point to the ISaGRAF message data , for ex, S18

Note:
The long integer, timer and float variable’s Network Address No. must occupy 2 No. in the ISaGRAF project. (refer to section 4.2 of “User’s Manual of ISaGRAF Embedded Controllers” or in the CD-ROM: \napdos\isagraf\wp-8xx7\english_manu\ ” User_Manual_I_8xx7.pdf”)

That means if you assign a Network Address No.= 11 to a Real type variable(or Timer or integer will have 32-bit value – larger than 32767 or smaller than -32768), the next No. 12 should not assigned to any other variable in the ISaGRAF project. However you may assign No.=13 to one another variable.

5.4.3.1 Post Boolean Value to The Controller

A. To post by the image

ON_OFF function is used for posting Boolean value to the controller by refer to the current Boolean value.

```
…
function ON_OFF(form_obj, obj, current_boo_value)
{
  if(current_boo_value==0)
  {
    flag = confirm("turn ON ?");
    if(flag) obj.value=1;
  }
  else
  {
    flag = confirm("turn OFF ?");
    if(flag) obj.value=0;
  }
  if(flag)
  {
   if(GetUserID(form_obj)==true) form_obj.submit();
  }
}
function refresh_data()
{
  B2.src = “img/cmd” + boolean_val[2] + “.jpg” ;
}
…
<body onLoad="init()">
…
<div style="position: absolute; width:100px;height:100px; z-index: 5; left: 242px;
top: 164px" >
<img style="cursor:hand" name="B2"  onclick="ON_OFF(form_B2, form_B2.B2,
boolean_val[2])"   src="img/cmd0.jpg">
```

The first parameter is the name of the “form”. The second parameter is the “<input>” name inside the form.

Demo example: wphmi_02 and wphmi_05

Display the current boolean image. In this example,

The layout (or location) of the image object “B2” is defined here by the “<div” and “ ” tags.

“cursor:hand” will display the mouse arrow as a hand when entering the image area

Name of the image object

Name of the form

The onclick will call ON_OFF( ) when the mouse click on it.
The first parameter is the name of the “form”. Here is “form_B2”.
The second parameter is the “<input>” name inside the form. Here is “form_B2.B2”.
The last is the current Boolean value. Here is boolean_val[2].

```
<form name="form_B2" method="post" action="./main.dll">
   <input name="BEGIN"  type="hidden">
   <input    name="B2"     type="hidden" value="0">
   <input name="END"      type="hidden">
</form>
```

Name of “<input>” inside the form. Here is “B2”. Because it is inside “form_B2”, then we must use the name of “form_B2.B2” to identify it.

```
</div>
…
</body>
```

B. To post by buttons

Demo example: wphmi_02 and wphmi_05

```
function ON_(form_obj, obj)
{
  flag = confirm("turn ON ?");
  if(flag)
  {
    obj.value=1;
    if(GetUserID(form_obj)==true) form_obj.submit();
  }
}
```

ON_ function is used for posting boolean value as “True” to the controller .

```
function OFF_(form_obj, obj)
{
  flag = confirm("turn OFF ?");
  if(flag)
  {
    obj.value=0;
    if(GetUserID(form_obj)==true) form_obj.submit();
  }
}
```

OFF_ function is used for posting boolean value as “Fasle” to the controller .

```
function refresh_data()
{
  B2.src = "img/big_Tcircle_red" + boolean_val[2] + ".jpg" ;
}
…
```

Display the current Boolean image. In this EX,
0: “img/big_Tcircle_red0.jpg” ,
1: “img/ big_Tcircle_red1.jpg”

The layout (or location) of the image object “B2” is defined here by the “<div” and “ ” tags.

```
<body onLoad="init()">
…
<div style="position: absolute; width: 56px; height:40px; z-index: 5; left: 82px;
top: 69px" >
<img name="B2" src="img/big_Tcircle_red0.jpg">
</div>
<div style="position:absolute; left:85px; top:124px; width:42px; height:27px;">
```

```
<input type="button" value="ON" style="cursor:hand" onClick="ON_(form_B2,
form_B2.B2)">
```

A button to call ON_( )
First parameter is the name of the form. Here is “form_B2”
The second is the name of the “<input>” inside the form.
Here is “form_B2.B2”

```
<form name="form_B2" method="post" action="./main.dll">
   <input name="BEGIN"  type="hidden"  value="">
   <input  name="B2"       type="hidden"  value="1">
   <input name="END"     type="hidden"  value="">
</form>
</div>
```

Name of “<input>” inside the form. Here is “B2”. Because it is inside “form_B2”, then must use the name of “form_B2.B2” to identify it.

```
<div style="position:absolute; left:85px; top:166px; width:47px; height:31px">
<input type="button" value="OFF" style="cursor:hand" onClick="OFF_(form_B2,
form_B2.B2)">
</div>
…
</body>
```

A button to call OFF_( )
First parameter is the name of the form. Here is “form_B2”.
The second is the name of the “<input>” inside the form. Here is “form_B2.B2”

5.4.3.2 Post Word & Long & Float & Timer & String Value to The Controller

…

```
function Check(form_obj)
{
  flag = confirm("Are you sure?");
  if(flag)
  {
    if(GetUserID(form_obj)==false) { return false; }
    form_obj.submit();
    return true;
  }
  else
  {
    return false;
  }
}
```

Check( ) is used for posting any “form”.

Demo example:
wphmi_03, wphmi_04, wphmi_05, wphmi_06 and wphmi_07

```
function refresh_data()
{
  L15.innerText=get_long_val(15);
  F17.innerText=float_val[17];
}
…
<body onLoad="init()">
…
<div style="position: absolute; width: 195px; height: 25px; z-index: 2; left: 45px;
top: 52px" >
L15 = <b id="L15">xxxx</b></div>
<div style="position: absolute; width: 196px; height: 29px; z-index: 3; left: 45px;
top: 82px" >
F17 = <b id="F17">xxxx</b></div>

<div style="position:absolute; left:47px; top:131px; width:204px; height:60px">

  <form name="form1" method="post" action="./main.dll">
    <input name="BEGIN"  type="hidden"  value="">
    <input name="L15"    type="text"    value="Enter long val (L15)">
    <input name="F17"    type="text"    value="Enter float val (F17)">
    <input name="END"    type="hidden"  value="">
  </form>
</div>

<div style="position:absolute; width:74px; height:31px; left: 234px; top:
150px;">
  <input type="button" style="cursor:hand" onClick="return Check(form1)"
value="Enter">
</div>
…
</body>
```

Display dynamic value here.
If data is word , please use word_val[ ]
If data is timer, please use timer_val[ ]
If data is string, please use string_val[ ]

The layout (or location) of the text object “L15” & “F17” are defined here by the “<div” “ ” tags.

Text input L15 & F17 inside the “form1”.
If data is timer, please use “L”; And “W” for word; “S” for string.

“cursor:hand” will display the mouse arrow as a hand when entering the button area

When mouse click on this button, it calls Check( ) to post to the controller

5.5 Multi-Pages

The Web HMI in the WinPAC-8xx7 supports multi-pages application. You may refer to Chapter 3 to setup the multi-page demo – “wphmi_05” to see how it work.

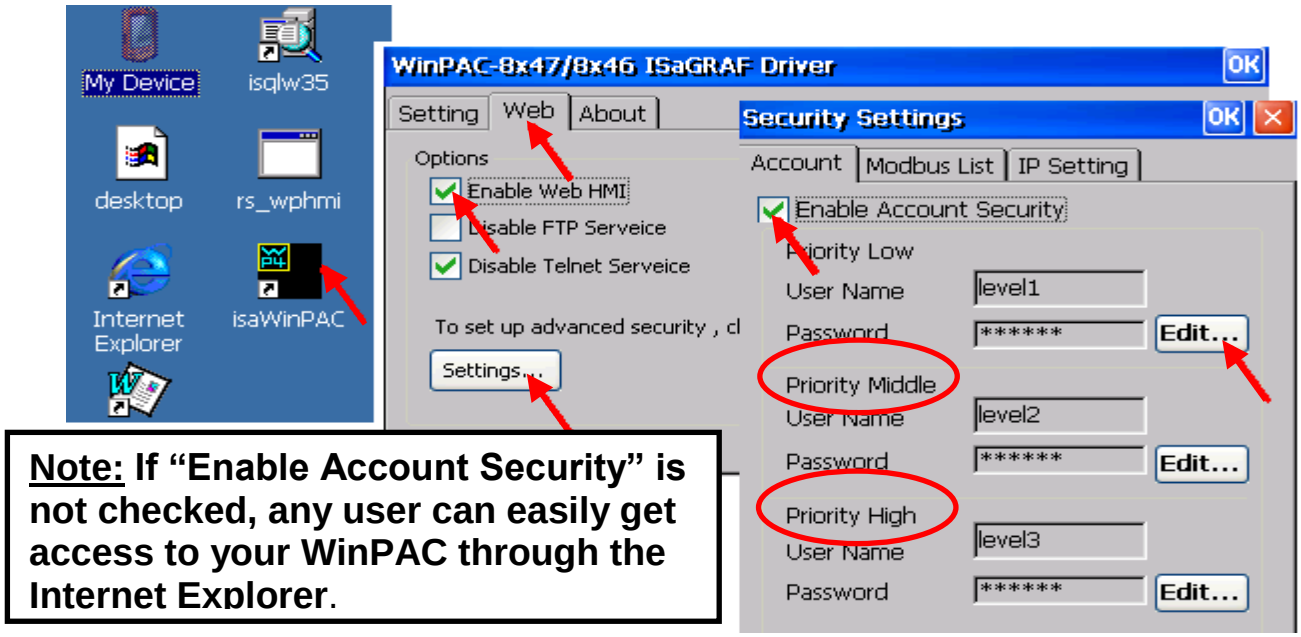
5.5.1 Level 2 And Level 3 Page

The multi-page name can be any valid html file name. For example, “page2.htm”, “kitchen.htm” , “u2-page4.htm” .

If “u2-” appear in front of the page name, the page will become a Level 2 page.
For example, the “u2-Page4.htm” in the “wphmi_05” demo.
If “u3-” appear in front of the page name, the page will become a Level 3 page.
For example, the “u3-time.htm” in the “wphmi_05” demo.

What is a Level2 page? Only users login with the Middle or High priority can get access to it. To access to the Level3 page, users have to login as a High priority user. The page name without “u2-“ and “u3-“ is identified as Level 1 page. That means any user successfully login can access to it. For example: the “main.htm”.

The other rules for multi-pages are almost the same as “main.htm” (section 5.4)



<u>Note:</u> If “Enable Account Security” is not checked, any user can easily get access to your WinPAC through the Internet Explorer.

5.5.2 Switch One Page To One Another Page

Please take a look at the “menu.htm” of the “wphmi_05” demo as below. The “goto_R_page( )” function can be used for switching to other page.

```
<!-- top_or_left=0 , scrolling=0 , width=110 , resize=1 -->

<html>
<head>
<title>Title1</title>
<meta http-equiv="Content-Type" content="text/html; charset=big5" >
<SCRIPT LANGUAGE="JavaScript" src="./msg/wincon.js"></SCRIPT>

<SCRIPT LANGUAGE="JavaScript">
function start1()
{
  A_11();
}
function refresh_data()
{
  if(run_at_pc==1) return;  // if simulate at the PC, just return
  …
}
</SCRIPT>
</head>
<body onload="start1()">

<!-- Logout button -->
<form name="form_logout" method="post" action="./login.dll">
  <input style="cursor:hand" name="CMD" type="submit" value="Logout"
onClick="return logout(this.form)">
</form>
<br/>
<br/>

<!-- Goto main.htm -->
<A style="cursor:hand" onClick="goto_R_page('main.htm')">第1頁</A>
<br/>
<br/>

<!-- Goto kitchen.htm -->
<A style="cursor:hand"
onClick="goto_R_page('kitchen.htm')">Kitchen</A><br/>
<br/>
<br/>
```

“cursor:hand” will display the mouse arrow as a hand when entering the button area

Switch page to “main.htm”

Switch page to “kitchen.htm”

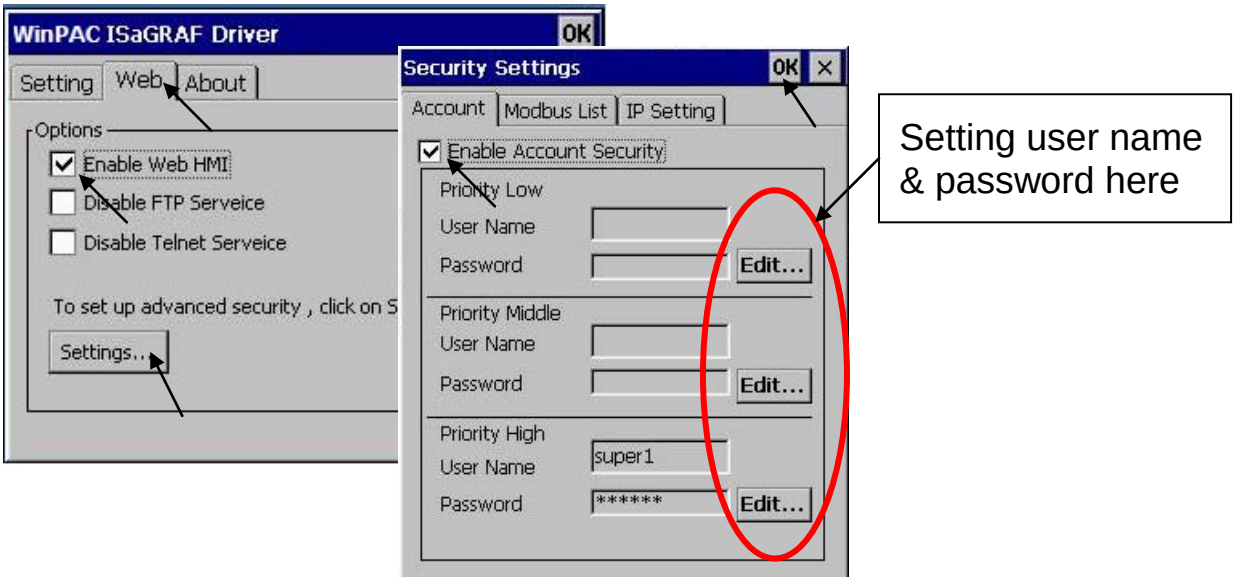
5.6 Web Security

There are some ways user can get access to the WinPAC-8xx7 via Ethernet port.

1. Using Modbus TCP protocol at port No.= 502. (ISaGRAF & other HMI do this)
2. Using ftp (for example, key in “ftp://10.0.0.103” on the Internet Explorer)
3. Using telnet (for example, key in “telnet 10.0.0.103 in the “command” window)
4. Using the Web server (The Web HMI does)

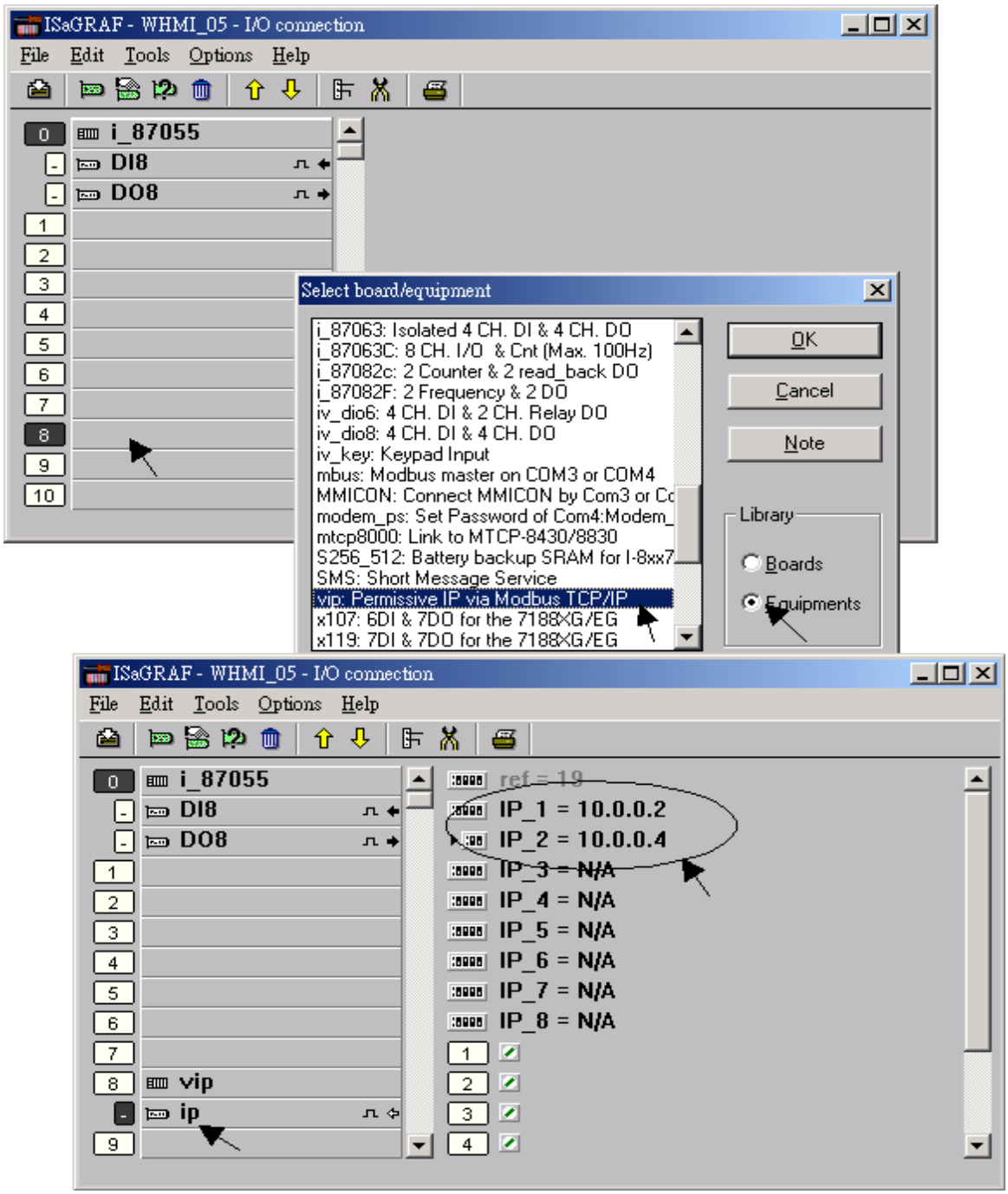
For safety, recommend to disable item 2 and 3 at run time.

And about item 4, please set proper username & password for the Web HMI.



About item 1, user may set up to 8 IP address for ISaGRAF or other HMI to get access to the WP-8xx7 via the Modbus TCP/IP protocol as below.
On the IO connection window of ISaGRAF, please connect “vip” and entering the IP which can get access to the WP-8xx7 via Modbus TCP/IP protocol. If “vip”

is not connected, any remote IP can get access to your WP-8xx7 via Modbus TCP/IP protocol. If “vip” is connected and No IP is entered (all assigned as “N/A”), No HMI and ISaGRAF can get access to it anymore.



Please re-compile your ISaGRAF project and download it to the controller if you have modified the IO connection.