Chapter 4. Linking Controllers To An HMI Program

Note: For communicating to W-8347/8747 via Modbus TCP/IP protocol, there are two Ethernet ports built in the W-8x47/8x46 controller, please connect your PC/HMI to W-8347/8747 's "LAN1" port. And please using "NS-205" or "NS-208" Ethernet switch.

This chapter details how to make data from the I-8xx7, I-7188EG/XG, uPAC-7186EG, iPAC-8447/8847 & W-8xx7 controller system available to Human Machine Interface (HMI) programs. This is a powerful feature that allows customers to create their own custom HMI programs and link them to the controller system.

After you realize the material described in section 4.1, if you would like to use the I-8xx7, I-7188EG/XG controller as a **Modbus or Modbus TCP/IP I/O**, you may refer to section 4.3. Additionally there are "touch screen" monitors provided by ICP DAS that support the "Modbus" protocol, and these touch screen monitors can also access data from an controller . Section 4.4 illustrates how to link a "Touch 510" monitor to an ISaGRAF controller system.

Note:
1. I-7188EG / 7186EG , uPAC-7186EG, I-8437 / 8837, I-8437-80 / I-8837-80, iPAC-8447/8847 and W-8xx7 controllers all support Modbus TCP/IP Slave protocol at its Ethernet port.
2. I-8417 / 8817 's COM1:RS-232 and COM2:RS-485 default supports Modbus RTU Slave.
3. I-8437 / 8837 's COM1 default supports Modbus RTU Slave protocol. To enable its COM3 to support Modbus RTU Slave , please refer to Chapter 3.5 & 3.10 of I-8xx7 "Getting Started Manual" delivered with the hardware.
4. I-7188EG/XG & uPAC-7186EG 's COM1 default supports Modbus RTU Slave protocol. To enable its COM3 to support Modbus RTU Slave , please refer to Chapter 3.7 & 3.6 of I-7188EG/XG "Getting Started Manual" delivered with the hardware.
5. W-8xx7 defaut no support Modbus RTU Slave port. To enable its COM2 or COM3 or COM5 to COM8 to support Modbus RTU Slave, please refer to the Appendix A.2 , Appendix G & F of W-8xx7 "Getting Started Manual" delivered with the hardware.

4.1: Declaring Variable Addresses For Network Access

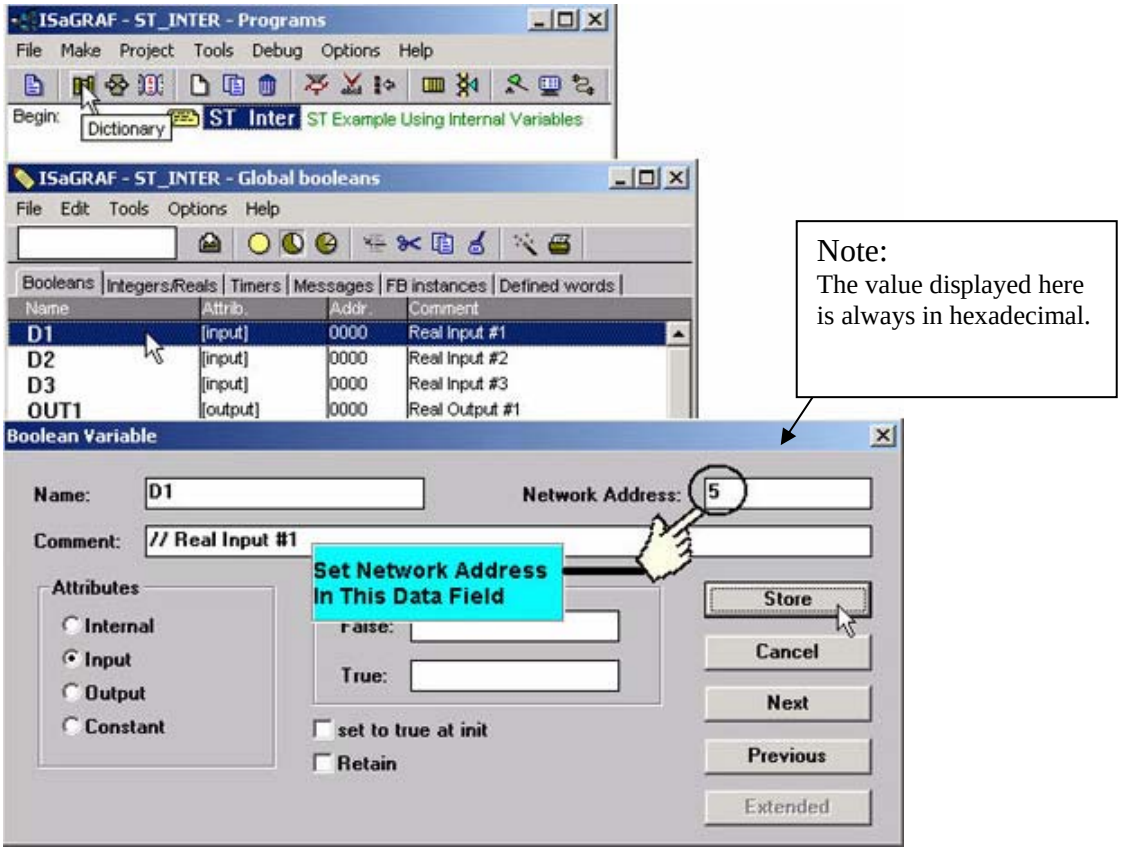
To make data from an I-8xx7, I-7188EG/XG & W-8xx7 controller system available to other software programs or HMI devices, you must first declare the variable with a "Network Address". The variable must be declared with a network address number that is in the "Modbus" format. Other software programs or HMI devices will access the controller information through these network addresses.

There are two methods available to declare a variable for network address access. The first method is described below. Open an "ISaGRAF Programs" windows and click on the "Dictionary" icon, then double click on the variable to assign a network address number.

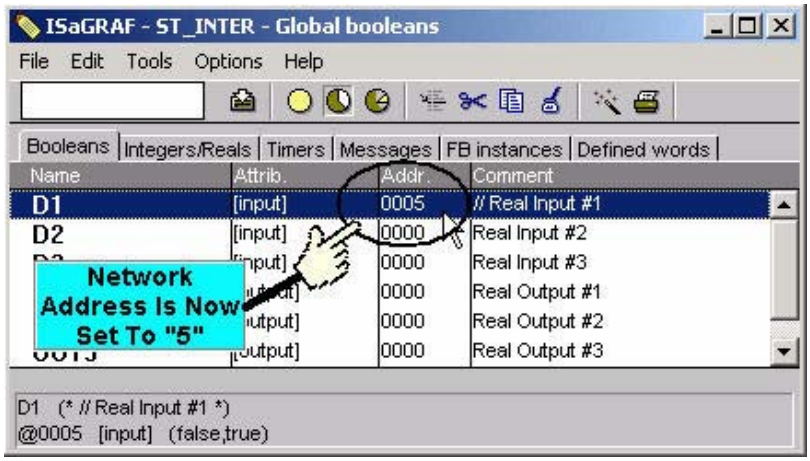
Note:
1. The valid network addresses for an **I-8417/8817/8437/8837 , I-7188EG/XG, uPAC-7186EG and iPAC-8447 / 8847** controller system is from 1 to FFF in hexadecimal (**1 ~ 4095**). Network address **5001 to 8072** is for word and integer arrays, please refer to Section 4.5.

2. The valid network addresses for an **W-8037/8337/8737 & W-8347/8747** controller system is from 1 to 1FFF in hexadecimal (**1 ~ 8191**). Network address **10,001 to 19,216** is for word and integer arrays, please refer to Section 4.5.

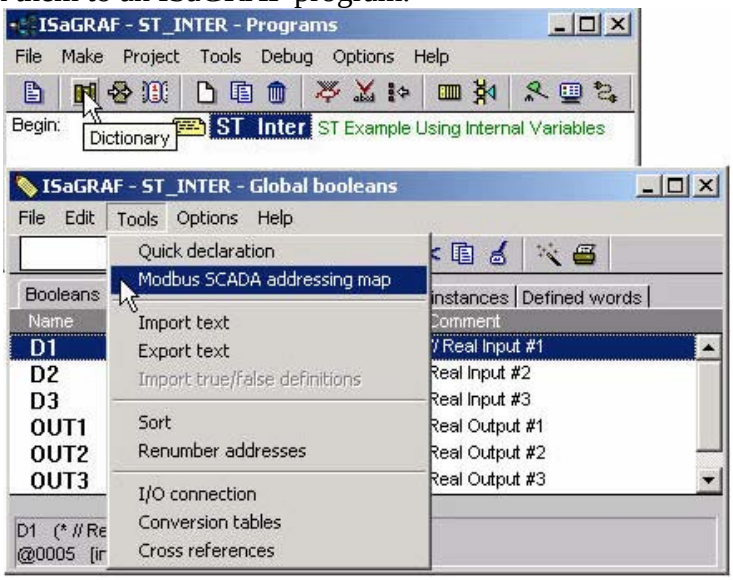
There are two ways to assign a Modbus network address No. to a variable. One is as below figure. (To assign many Modbus Network address No. to the "Variable Array", please refer to Chapter 2.6)



When you click on the "Store" button you will see that "ISaGRAF Global Variables" window will now be updated with the new network address for the variable.

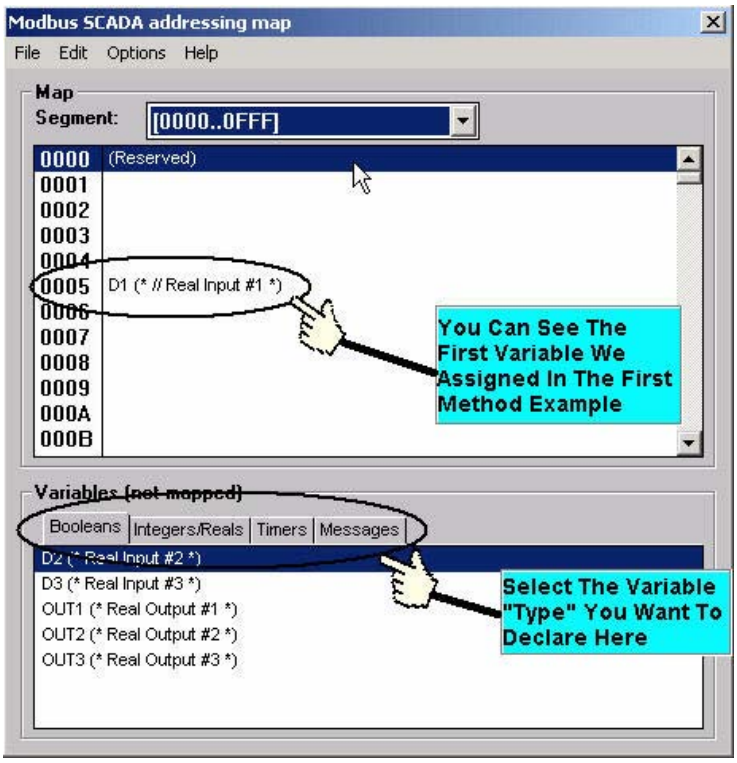


The second method for assigning network addresses to variables requires that you declare the variables BEFORE you assign them. This method allows you to assign numerous network address variables before you link them to an ISaGRAF program.

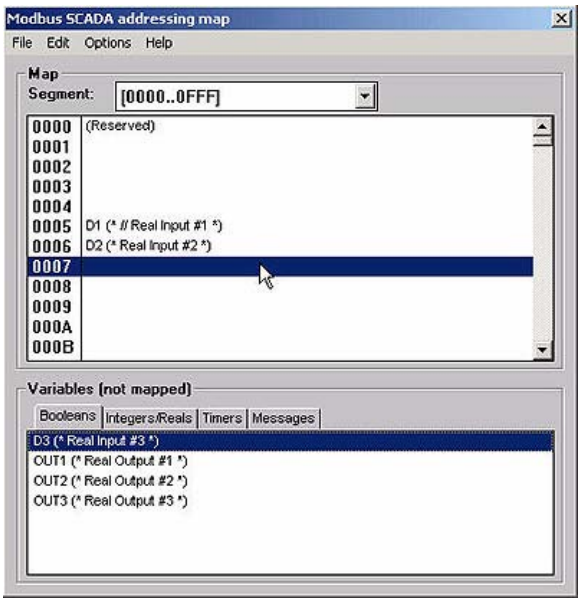


When you click on "Modbus SCADA Addressing Map" (SCADA is an industrial process control acronym that stands for "Supervisory Control And Data Acquisition") the "Modbus SCADA Addressing Map" window will open.

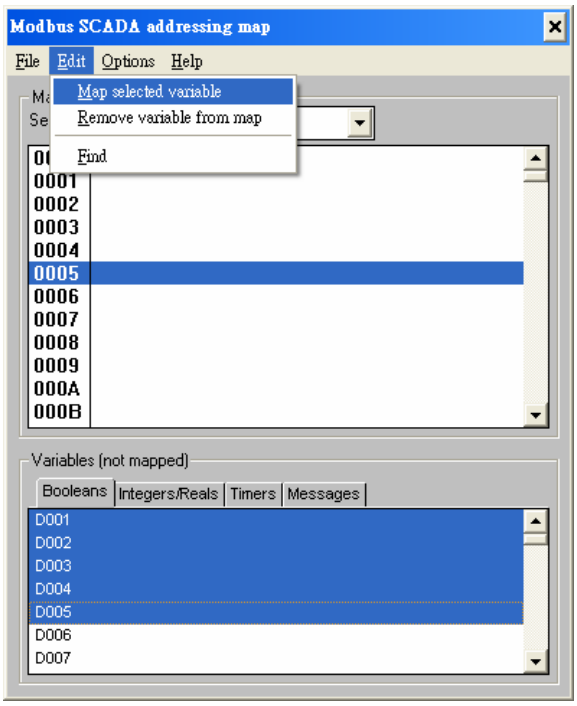
Note that one of the variables (D1) is already assigned from our previous example. You will note that the other variables that are not yet mapped are displayed in the lower portion under the "Variables (Not Mapped)" portion of the "Modbus SCADA Addressing Map" window.



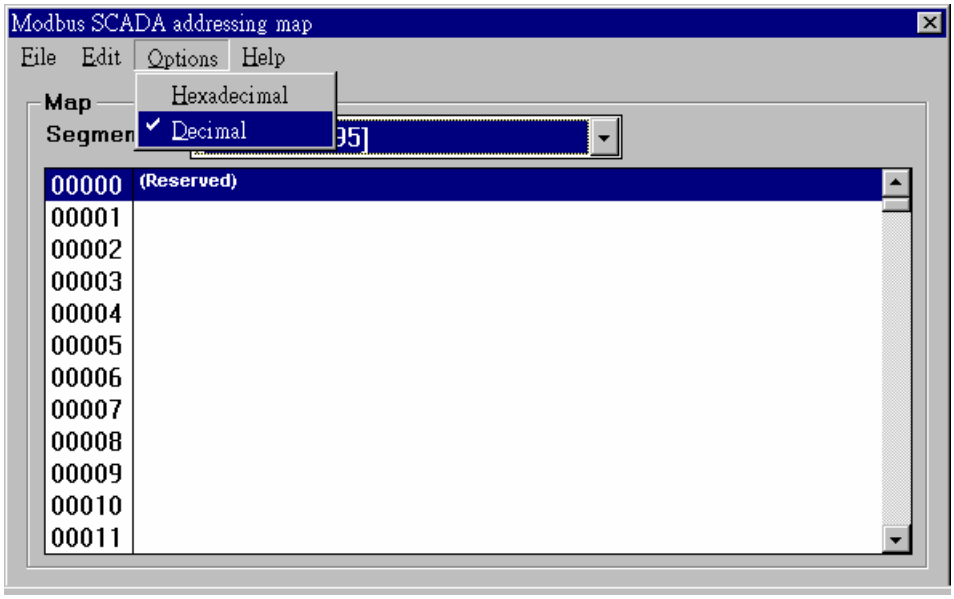
To assign the other variable address click on an unassigned "Map Segment" number, and then double click on the variable you want to assign to the address and the variable will automatically assign itself to the "Map Segment".



To assign continuous Network address to similar variables, for example, assigning No. 1 to 5 for D001 to D005, please select those variable names and then click on “Edit” – “Map selected variable”.



For human's thinking way, network address represented in hexadecimal format is inconvenient and it increases the chance to make mistake. Therefore, it's better to change it to be represented in decimal format. To do that is as following.



IMPORTANT NOTE REGARDING MODBUS NETWORK ADDRESSING

The Modbus network address definition scheme is sometimes different between HMI devices and other software programs. The difference is typically that the other programs may assign a network address number that is one (1) less than that of the I-8xx7, I-7188EG/X & W-8xx7 controller system.

HMI or devices such as Indusoft, Iconics, Citech, Wizcon, Kepware's OPC server, iFix, Wonderware's "Intouch", National Instruments "Labview", and ICP DAS's Touch 506L, Touch 506T and Touch 510T do have the exact same addressing scheme as the I-8xx7, I-7188EG/X & W-8xx7 controller system.

Known addressing disparities include "LabLink" and "Hitech" HMI software programs and devices. If you are assigning a network address of "B" (hexadecimal) of these products the I-8xx7 network address should be set to "C". A network address of "2" should be associated with a network address of "3" in the ISaGRAF controller system.

Another things mistaked very often is the first digit of the network address of many HMI softwares resprent the data type and Read/Write authority not one part of the network address.
For example, the network address relation between "iFix" and ISaGRAF is as below.

iFix(Decimal)	I-8xx7 (Decimal)
00001 (R/W Boolean)	1
…	…
10010 (Read Boolean)	10
…	…
31000(Read Word)	1000
…	…
42101(R/W Word)	2101

ICP DAS has not been able to test every possible HMI software program or hardware device that has Modbus addressing capability. If you are trying to connect your HMI software program or hardware device with Modbus to an I-8xx7, I-7188EG/X & W-8xx7 controller system, **REMEMBER** that you **may** have to offset the Modus addressing by 1 between these products so they will properly communicate with each other.

Developers who design and write their own software interface programs using Microsoft's Visual Basic or Visual C++ programming language should refer to Chapter 5 of this manual for more information on how to interface the Modbus protocol to these programming languages.

NOTE:

While communicating with the I-8xx7, I-7188EG/XG , uPAC-7186EG, iPAC-8447/8847 and Wincon-8xx7, **One single Modbus frame** cannot request more than **255 bits** except the Wincon-8xx7 (Max. 1968 bits for W-8xx7) , and also cannot request more than **120 words** in one single modbus frame. It should be divided into 2 or more reading frames to achieve it. To write bits to the controllers, **One single Modbus frame** cannot write more than **255 bits**, and also cannot write more than **120 words** in one single modbus frame. It should be divided into 2 or more writing frames to achieve it.

4.2:Read/Write Word, Long Word & Float through Modbus

Modbus protocol provides function 3 and 4 for reading multiple words while function 6 and 16 to write words. Please refer to Chapter 5 for more information about the protocol.

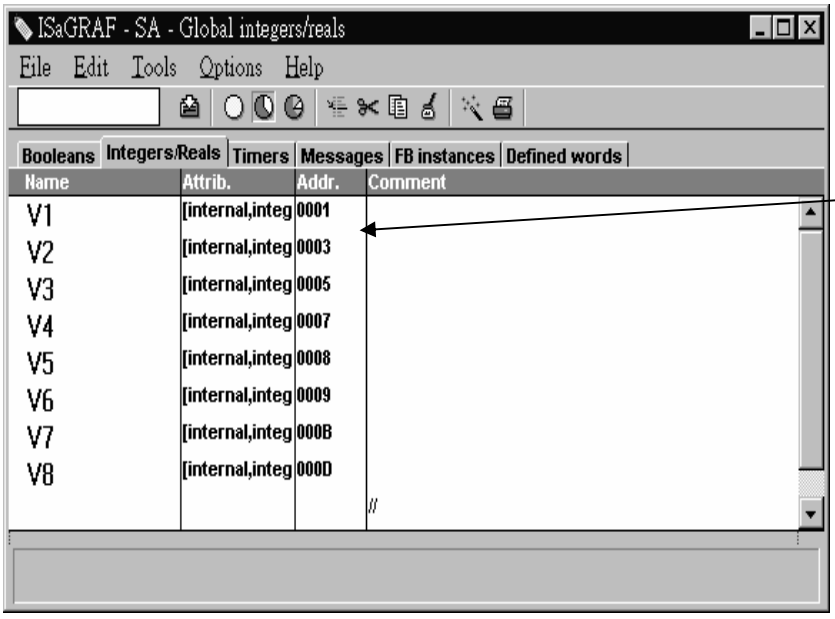
The **word** defined in the Modbus protocol of I-8xx7, I-7188EG/XG, uPAC-7186EG, iPAC-8447/8847 & W-8xx7 controllers is like a signed short integer, which occupies 2 bytes and range from –32,768 (8000 in hexa.) to +32,767 (7FFF in hexa.). It is normally used to describe the behavior of analog I/O channels. For examples, the I-87017 I/O board (please refer to section 3.2)

I-87017 :

Range ID (hexadecimal)	Electrical Range	Values on the channel (decimal)		
		-32768	0	+32767
8 (default)	± 10V	- 10V	0V	+ 10V
9	± 5V	- 5V	0V	+ 5V
A	± 1V	- 1V	0V	+ 1V
B	± 500mV	- 500mV	0mV	+ 500mV
C	± 150mV	- 150mV	0mV	+ 150mV
D	± 20mA	- 20mA	0mA	+ 20mA

The **long word** defined in the Modbus protocol of I-8xx7, I-7188EG/XG, uPAC-7186EG, iPAC-8447/8847 & W-8xx7 controllers is like a signed long integer, which occupies 4 bytes and range from -2,147,483,648 (8000 0000 in hexa.) to +2,147,483,647 (7FFF FFFF in hexa.). It is normally used to describe the value of internal integer variables declared on ISaGRAF workbench.

All integer variables declared in ISaGRAF are signed 32-bit format however the integer variable, which assigned with a network address will only, occupies 1 word (2 bytes) in the Mudbus transportation format. Since a long word occupies 2 words (4 bytes), to R / W long word through Modbus, the network address assigned to the integer variable must follow rules as below.

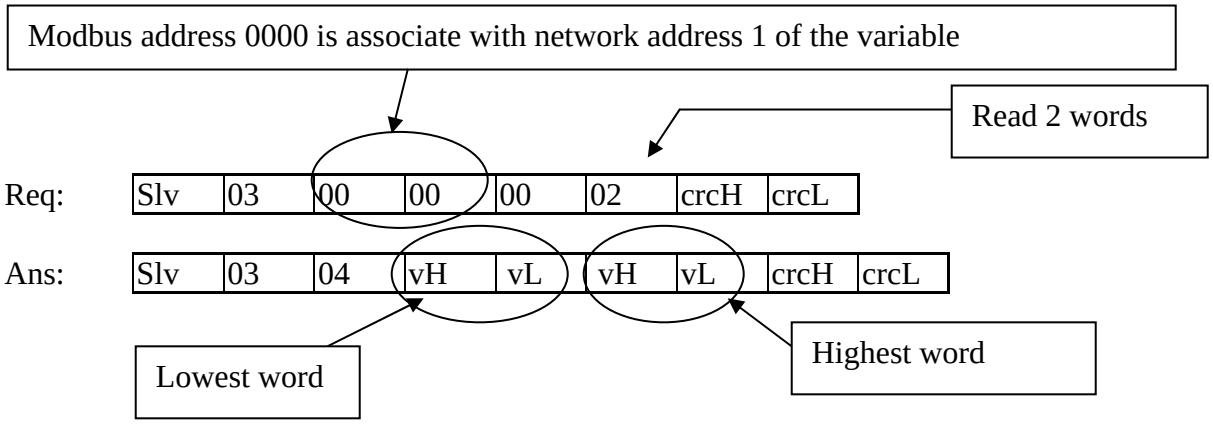


V1 is assigned to a network address “1”. If the network address “2” is not assigned to any other variable, V1 will occupy a long word (4 bytes) in the Modbus transportation formate.

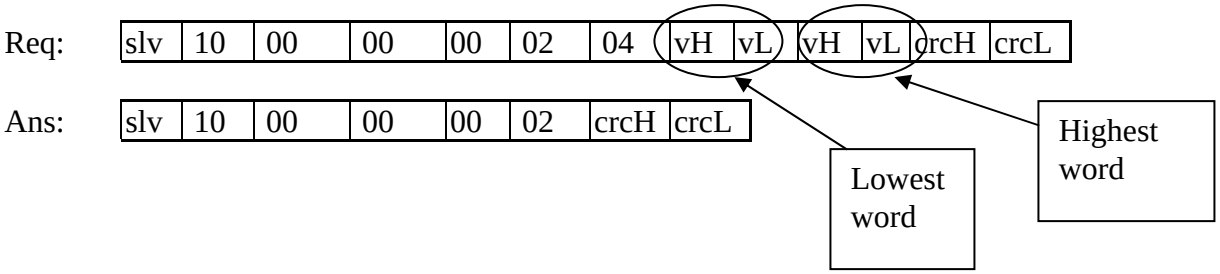
However if “2” is assigned to one another variable, V1 will only occupy one word (2 bytes) in the Modbus transportation format.

In this example, V1, V2, V3, V6, V7 and V8 will occupy 4 bytes however V4 and V5 only occupy 1 word (Lowest word) in the Modbus transportation formate.

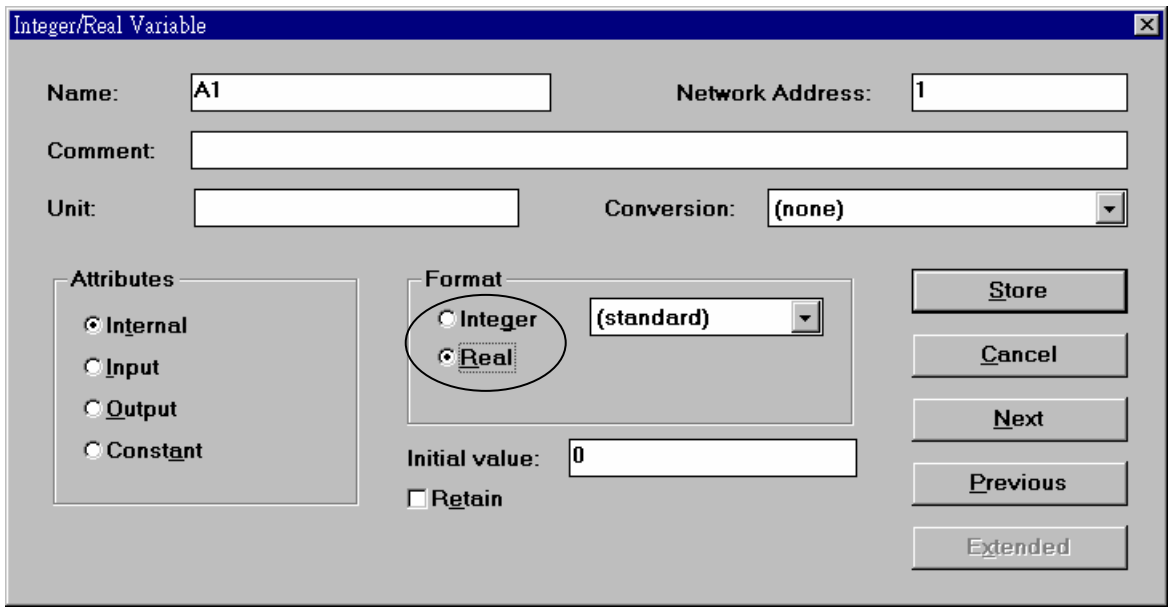
To read **long word** value of V1 is to read **2 words** by using modbus function 3 or 4 (please refer to section 5.1).



To write **long word** to V1 is to write **2 words** by using modbus function 16.



To read / write float (4 bytes) is very similar to read / write long word. The difference is the variable should be declared as “Real” type, and the next network address No. should not be assigned to any other variable.



There are much available HMI software on the market. You don’t need to care about the modbus protocol format. Just be careful to assign the correct network address on ISaGRAF.

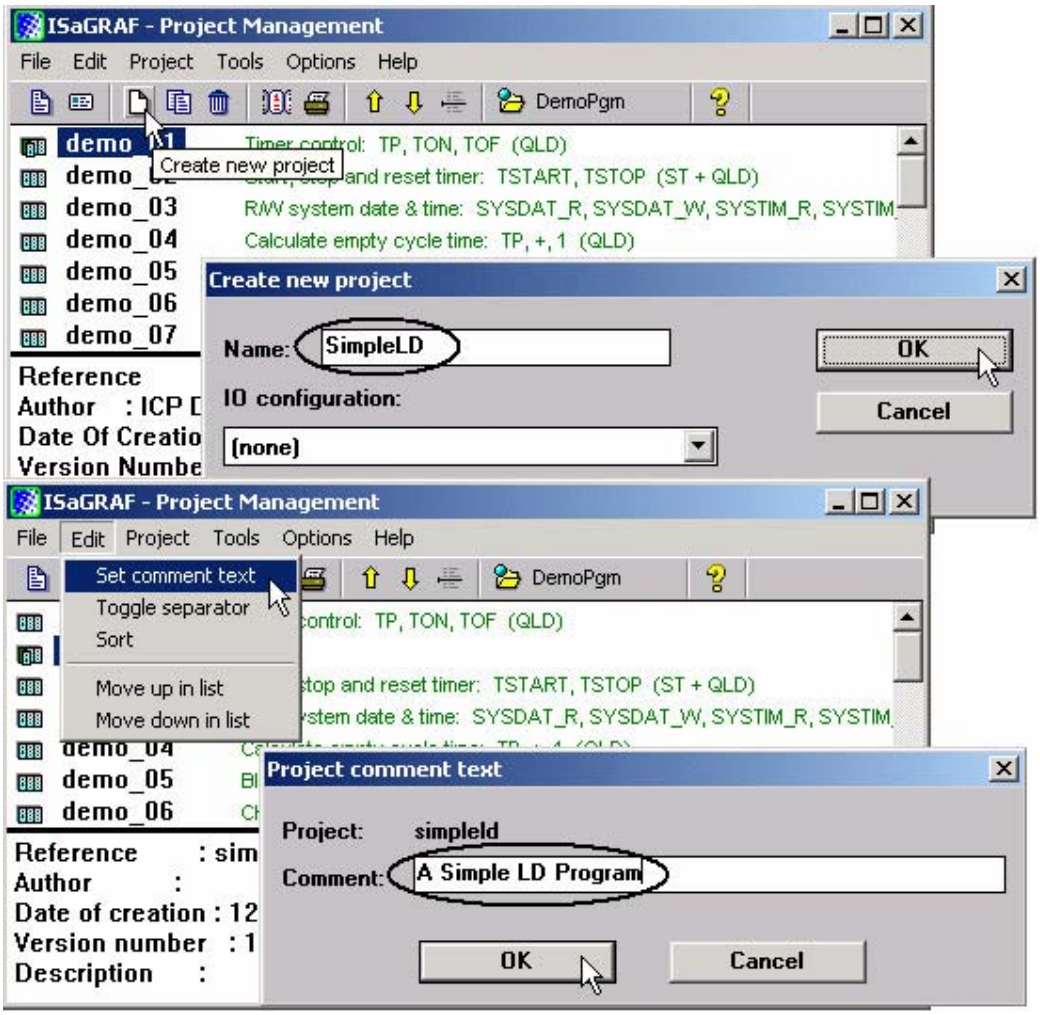
4.3: Using I-8xx7 As A Modbus I/O Or A Modbus TCP/IP I/O

There are some configurations that the HMI software gathers the I/O data from some called Modbus I/O modules. There I/O modules scan each input channels and refresh the output channels when need. Most of time there are no control logic inside these I/O modules, they are controlled by the HMI. To fit such kind of usage, the I-8417/8817/8437/8837 can be a Modbus I/O module, additionally the I-8437/8837 can be a Modbus TCP/IP I/O module. To do that, follow the following procedures (If you are not familiar with the ISaGRAF programming, recommended to review Chapter 2).

Create a new project

You may refer to section 2.1.1.2

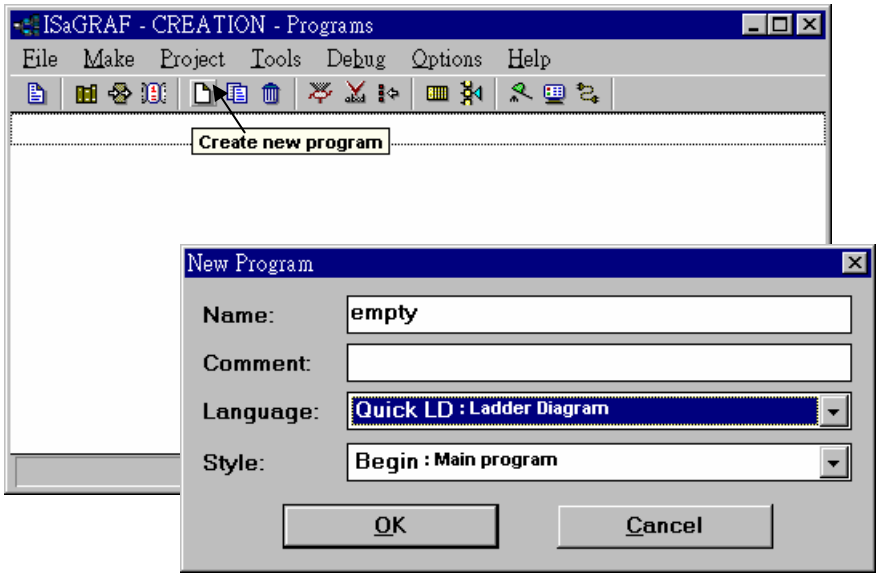
Example:



Create an empty program

No logic need.

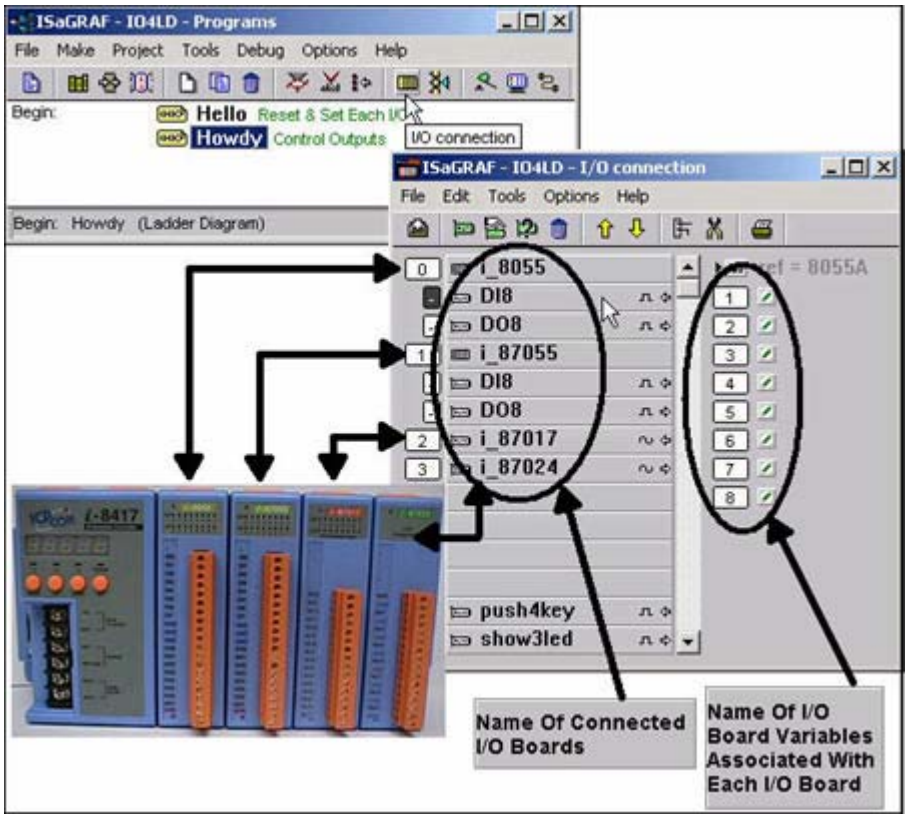
Example:



Connect I/O modules

You may refer to section 3.1

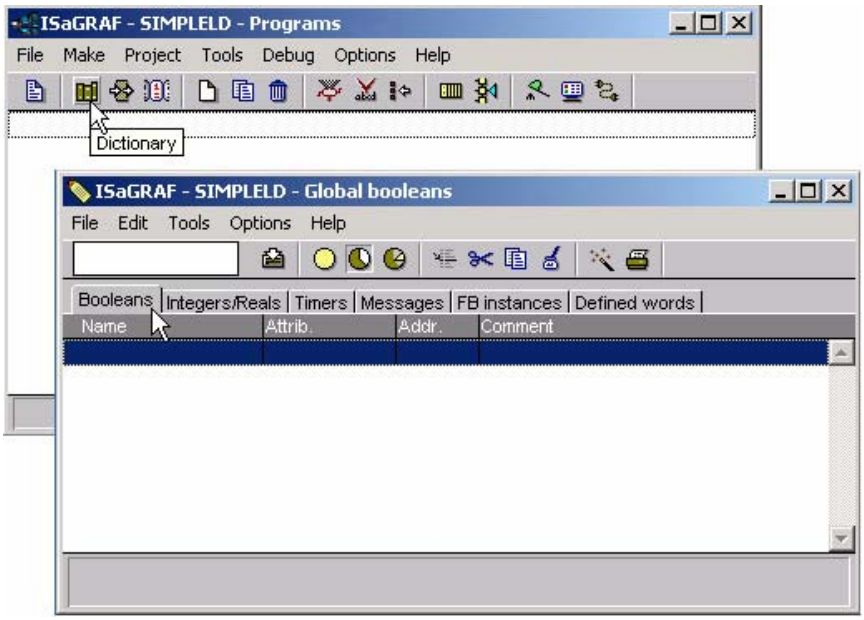
Example:

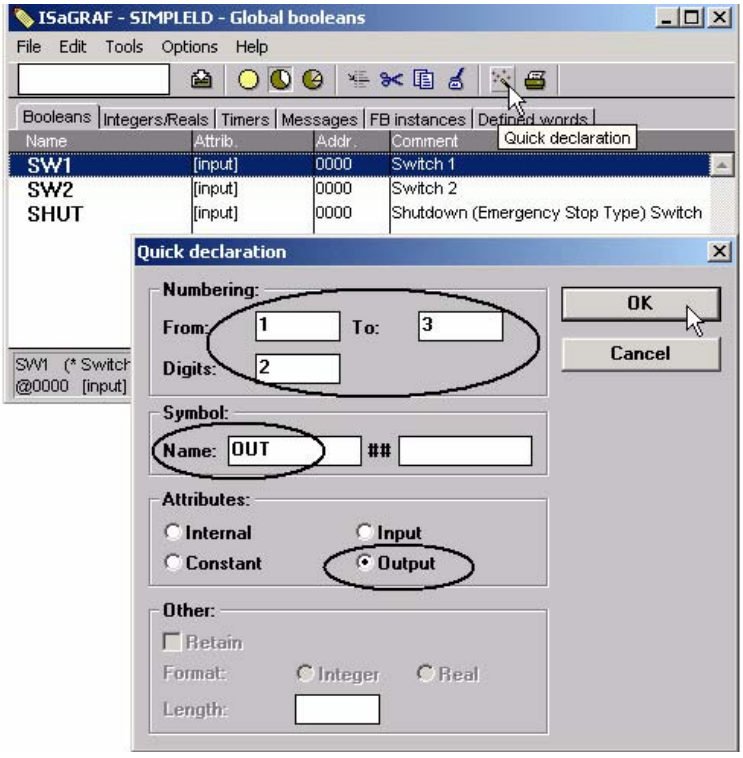


Declare Variables associated with the channels of connected I/O modules.

You may refer to section 2.1.1.3

Example:

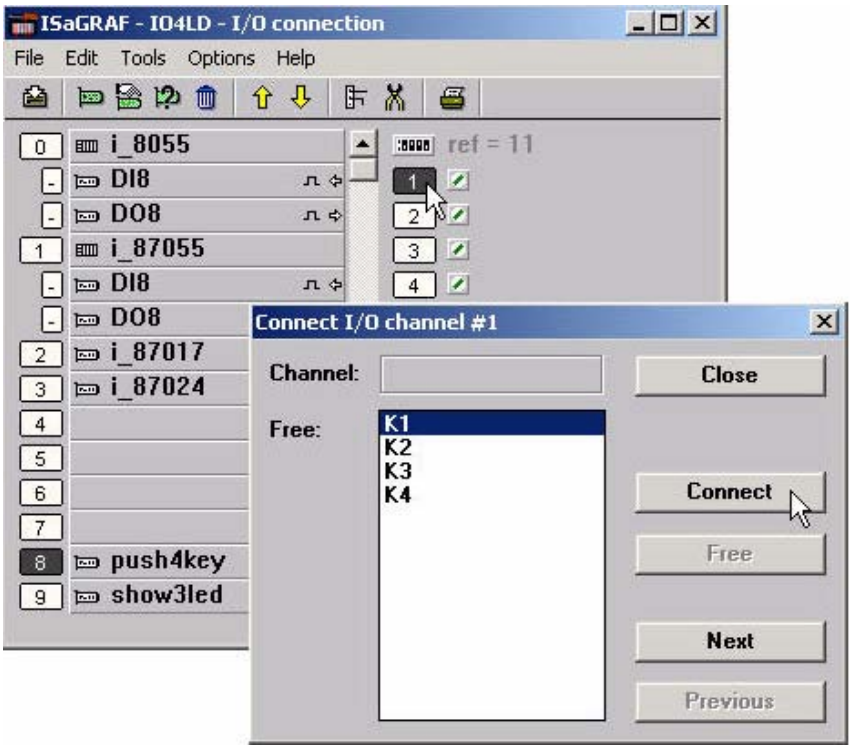




Link Variables to the associated channels of connected I/O modules.

You may refer to section 3.1.2

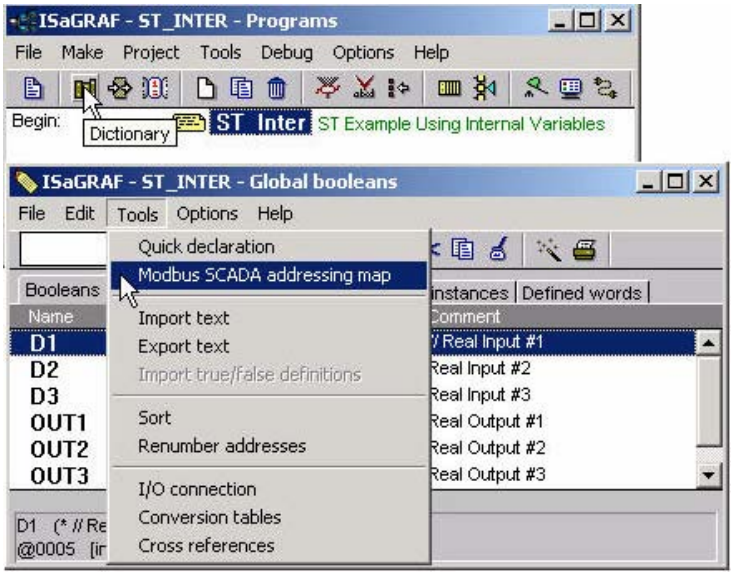
Example:



Assign the linked Variable a network address No.

You may refer to section 4.1

Example:



Compile & download the project

You may refer to section 2.1.3 & 2.1.5

Note:

Make sure the Net ID is set to the proper No. (section 1.3.1) For I-8437/8837, make sure the IP and Mask address is well set (appendix B).

The HMI can access to I/O channels through the associated network address now!

4.4: Linking I-8xx7, I-7188EG/XG & W-8xx7 To Touch 500

Touch500 series HMI support below protocols to link to ICP DAS ISaGRAF controllers.

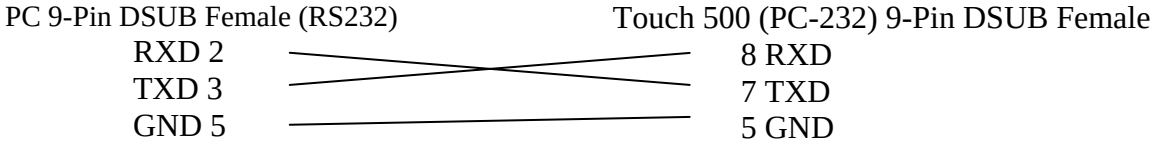
Item	Protocols
Touch-506L	Modbus RTU RS-232 , Modbus RTU RS-485
Touch-506T	Modbus RTU RS-232 , Modbus RTU RS-485
Touch-510T	Modbus RTU RS-232 , Modbus RTU RS-485

Please install ”EasyBuilder 500” software (Ver. 2.7.1 or later version) first before you can program the Touch 506L, 506T, 510T HMI. You may download the new released software and manual from below web site

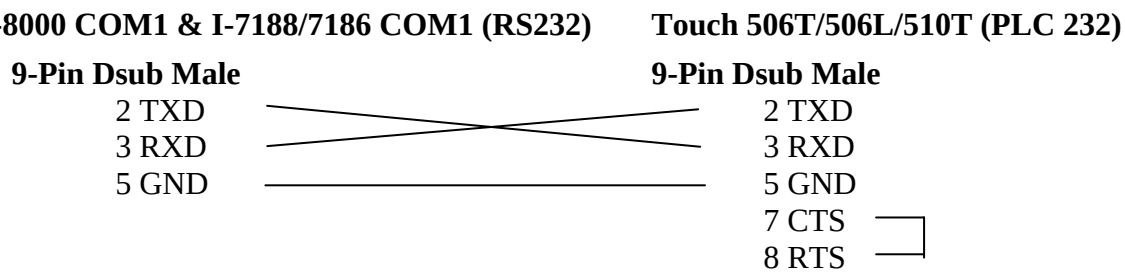
http://www.icpdas.com/download/others/touch/touch.htm “**setup.zip**”

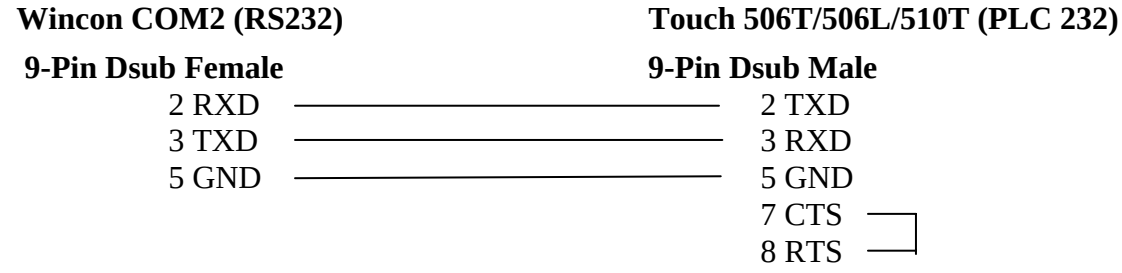
or run “setup.exe” at I-8000 CD-ROM:\napdos\others\touch\500series\setup\

RS-232 Cable Pin assignment of PC to Touch 500 series (For PC to download HMI screen).

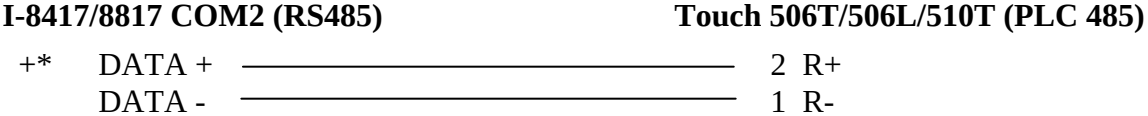


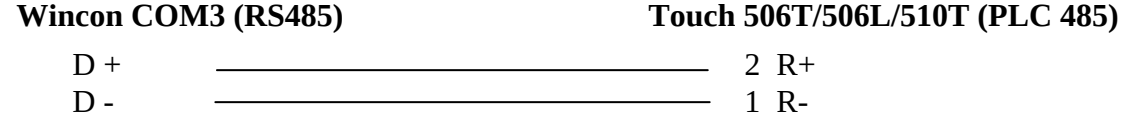
RS-232 Cable Pin assignment between controllers and Touch 500 series.





RS-485 Cable Pin assignment between controllers and Touch 500 series





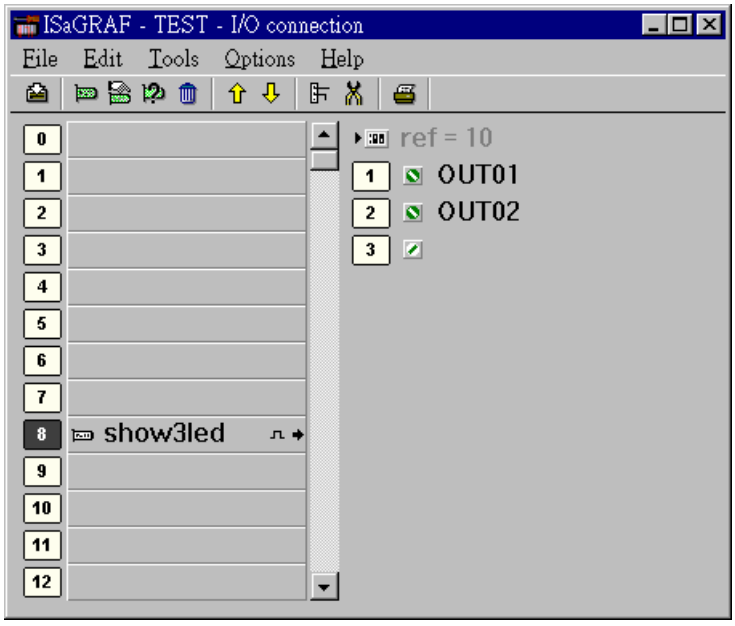
4.4.1: Program the I-8xx7, I-7188EG/XG & W-8xx7

To make data of the I-8xx7, I-7188EG/XG, uPAC-7186EG, iPAC-8447/8847 & W-8xx7 controller to be accessible to the Touch 510T, variables in the controller should be assigned a network address. Please refer to section 4.1, 4.2. If you are not familiar with the ISaGRAF programming, recommended to review Chapter 2.

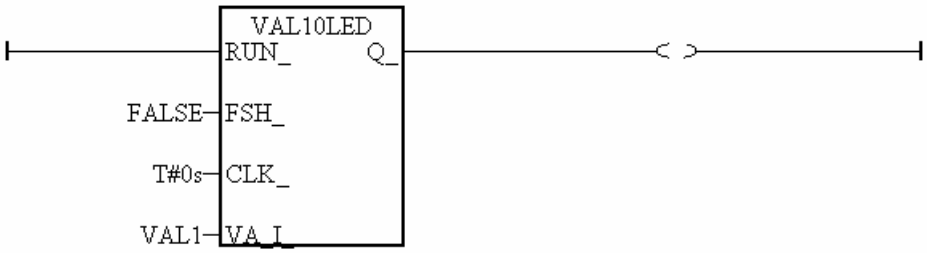
Variables used in this example.

Name	Type	Attribute	Network address	Others
OUT01	Boolean	Output	**0001**	-
OUT02	Boolean	Output	**0002**	-
VAL1	Integer	Internal	000A (10)	-

IO connection:



A simple LD program to show the “VAL1” to 7-segment LED:

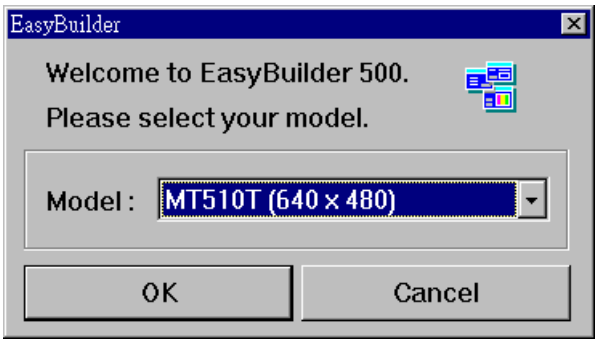


After you finish this project, compile and download it to the I-8xx7 controller.

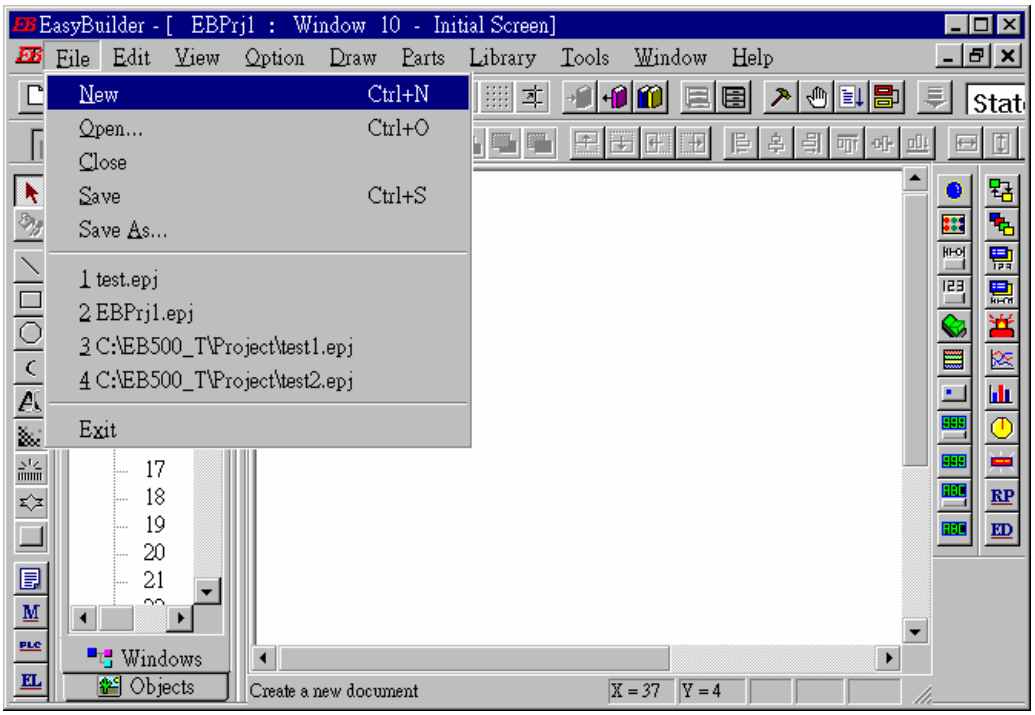
4.4.2: Program the Touch 510T

The “EasyBuilder 500” software can be used to designe many useful pictures for Touch 500 series. This section illustrates a simple example to program a Touch 510T. For more information about programming on the Touch series, please refer to the user manual which is provided with the “Touch” series hardware.

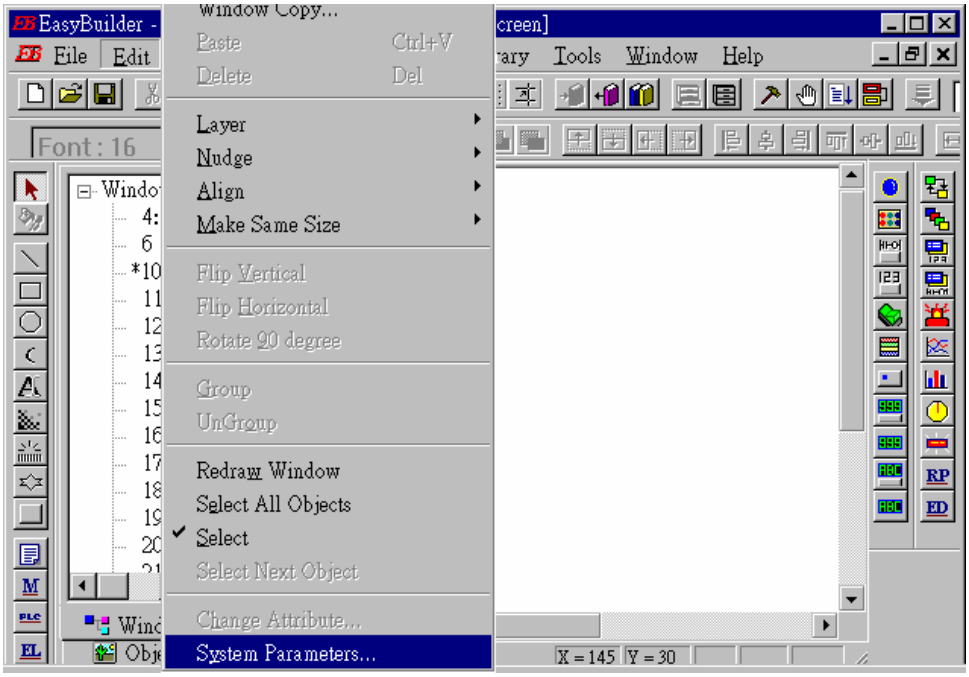
Click on the Windows "Start" button, then click on the "Program" button, then click on the "EasyBuilder" – “EasyBuilder 500” button. The following window will be displayed. Select the proper model for your application.



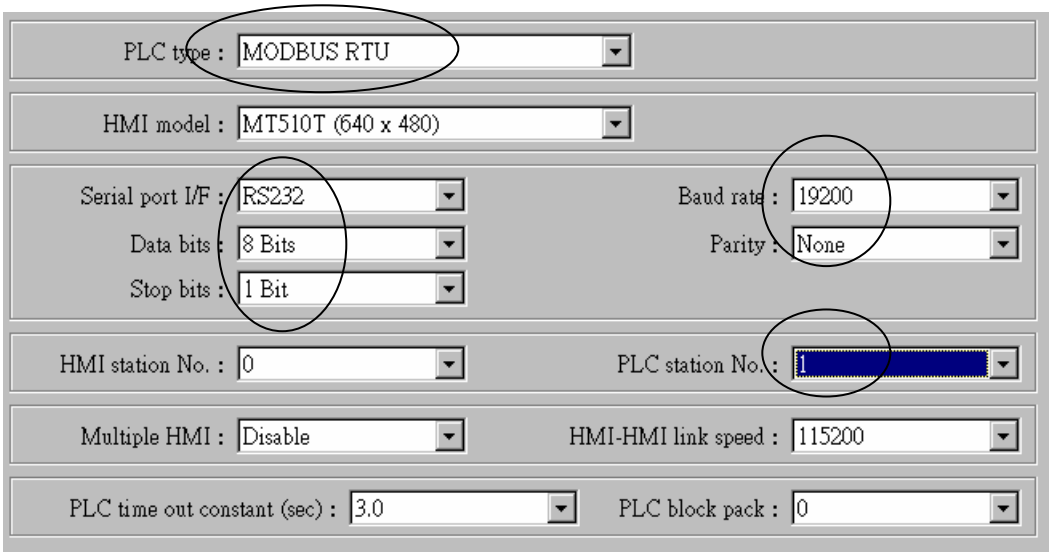
Click “File” – “New” to create a new project.



Click “Edit” – “System Parameters” to set the communication parameter between the Touch 510 and the ISaGRAF controller.



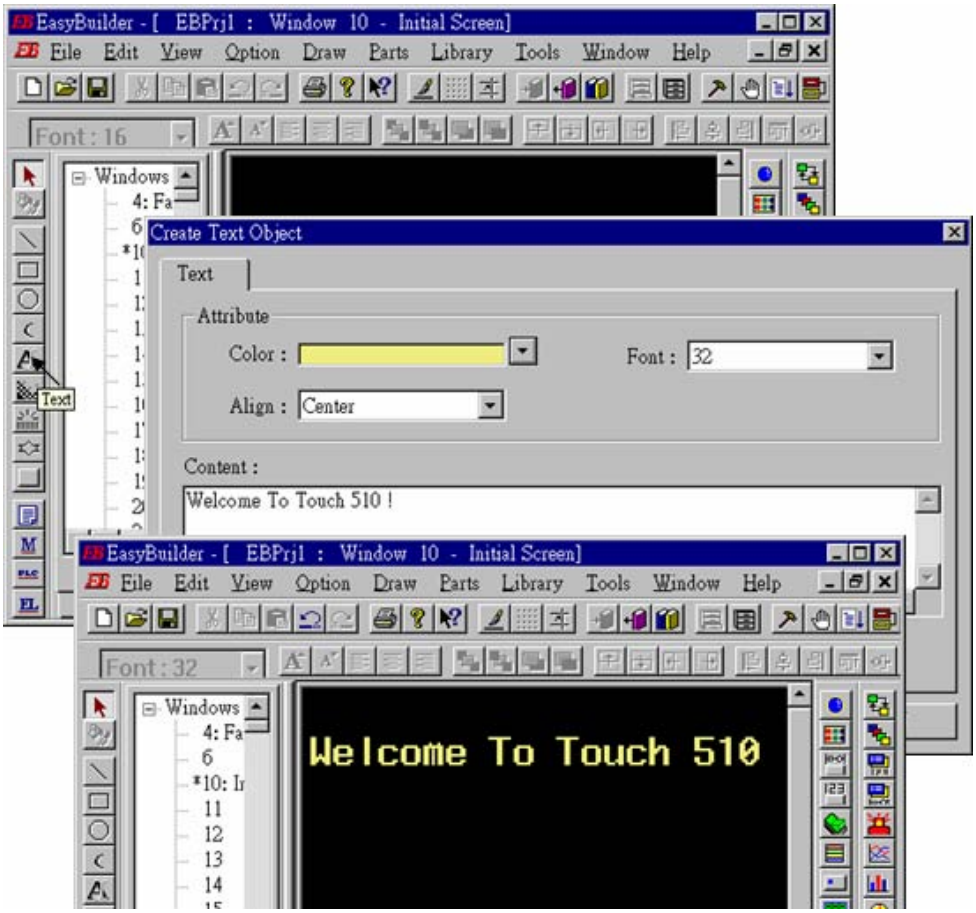
PLC type should be set to “**MODBUS RTU**”, Serial port set to “RS232”, Data bits set to “8 Bits”, Stop bits set to “1 Bit”, Baud rate set to “19200”, Parity set to “None”, PLC station No. set to be equal to the Net-ID of the I-8xx7 (set to 1 in this example).



Note:

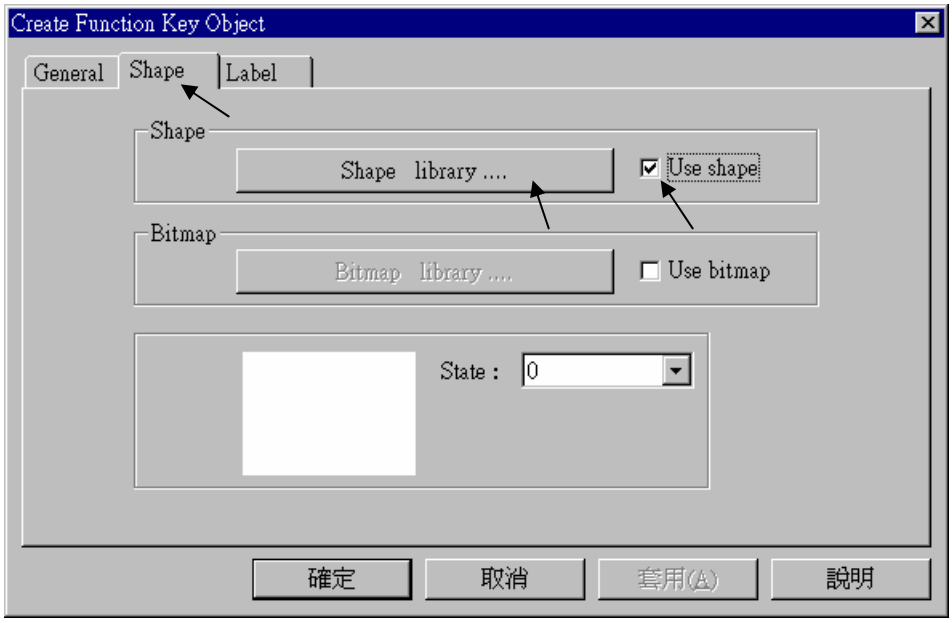
1. If using Touch506TE ‘s Ethernet to link to controller, please set PLC type as “MODBUS RTU TCP/IP” , PLC I/F port as “Ethernet” , Local IP address as Touch506TE ’s IP, Server IP address as controller ’s IP, PLC station No. as the same Net-ID No. of the controller (default is 1)
2. If the cable between the Touch 500 series and the controller is 2-wire RS-485, please set PLC type as “RS-485 2W”. Other setting is the same as RS-232.

Click on “Text” to add a text. Select the prefered “Color”, “Font”, “Align” for the text and then enter the “Content”. And then place it to the proper position.

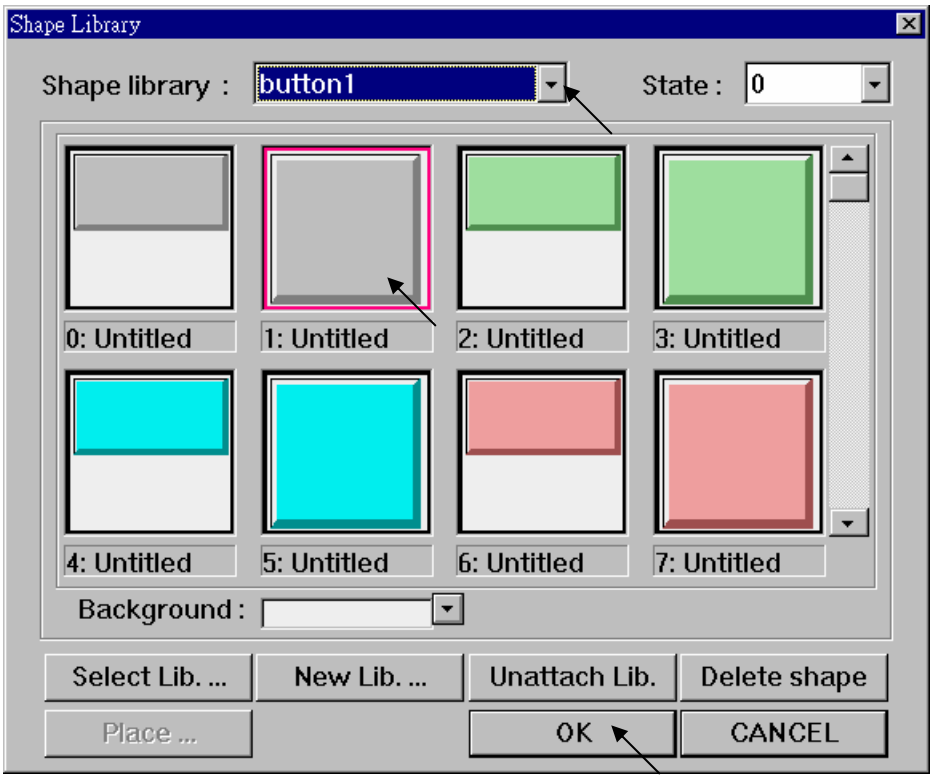


Click on “Function Key” to add a change-window button. Click on “General”, then select “Change Window” and set “Window No.” to 11.

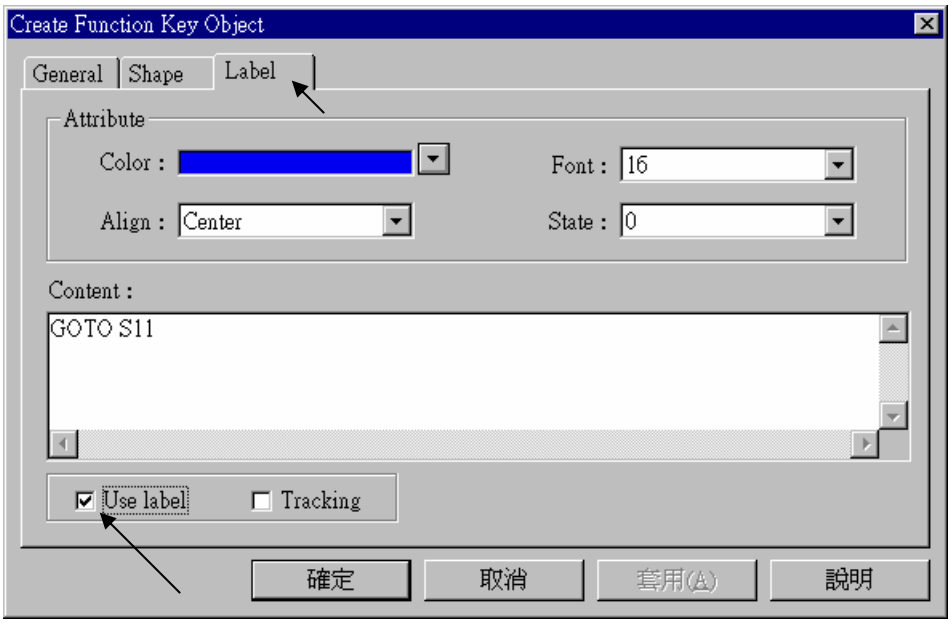
Click on “Shape”, then select “Use shape” and the click on “Shape library …”



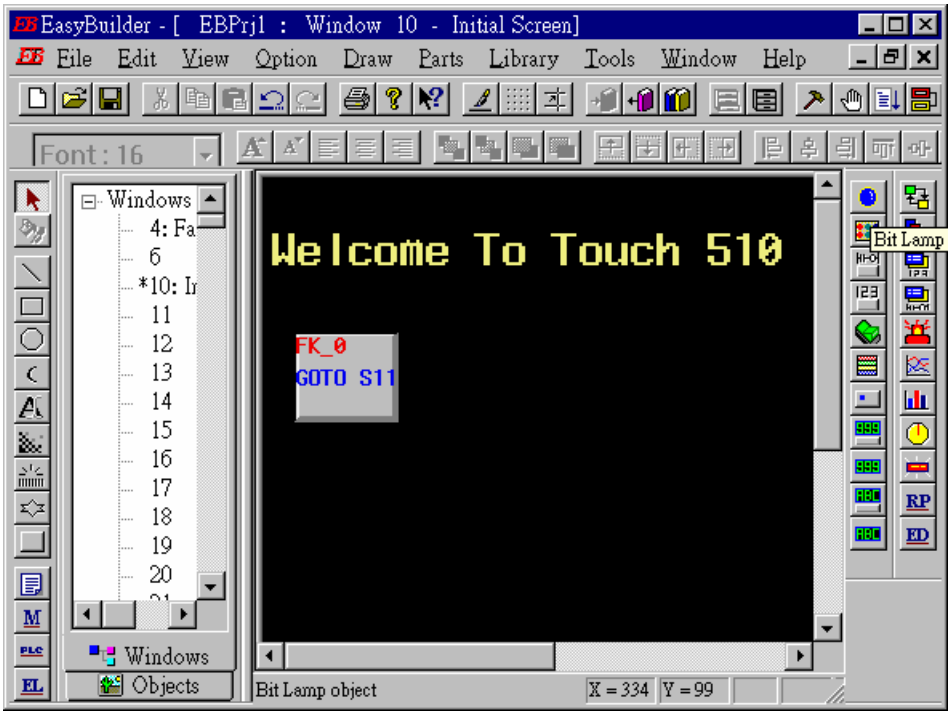
Select the prefered “Shape library” and then select one item and click on “OK”.



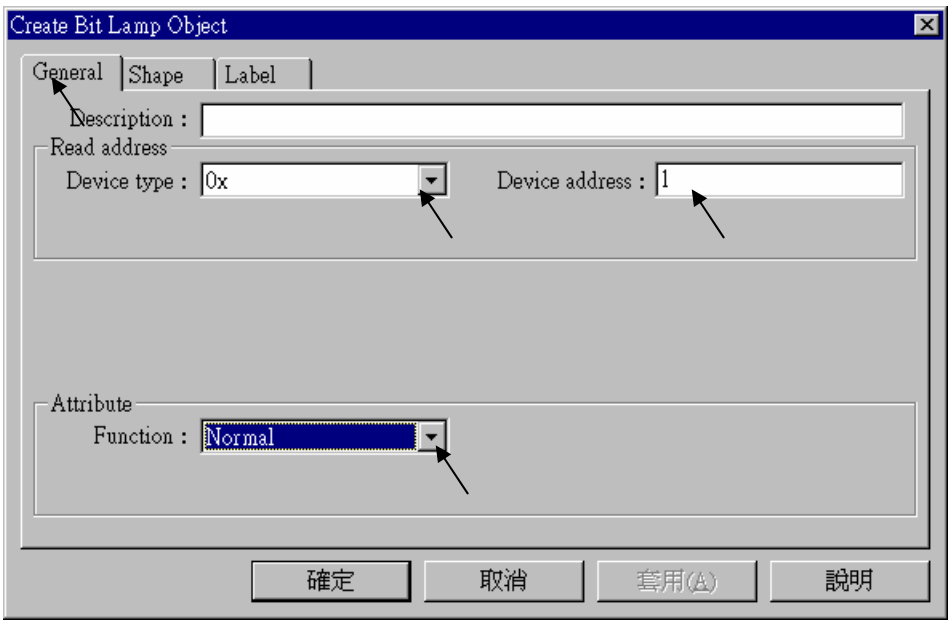
Click on “Label”, then select the prefered “Color”, “Font”, “Align” and set “Content” to “GOTO S11”, and **make sure “Use label” is selected**.



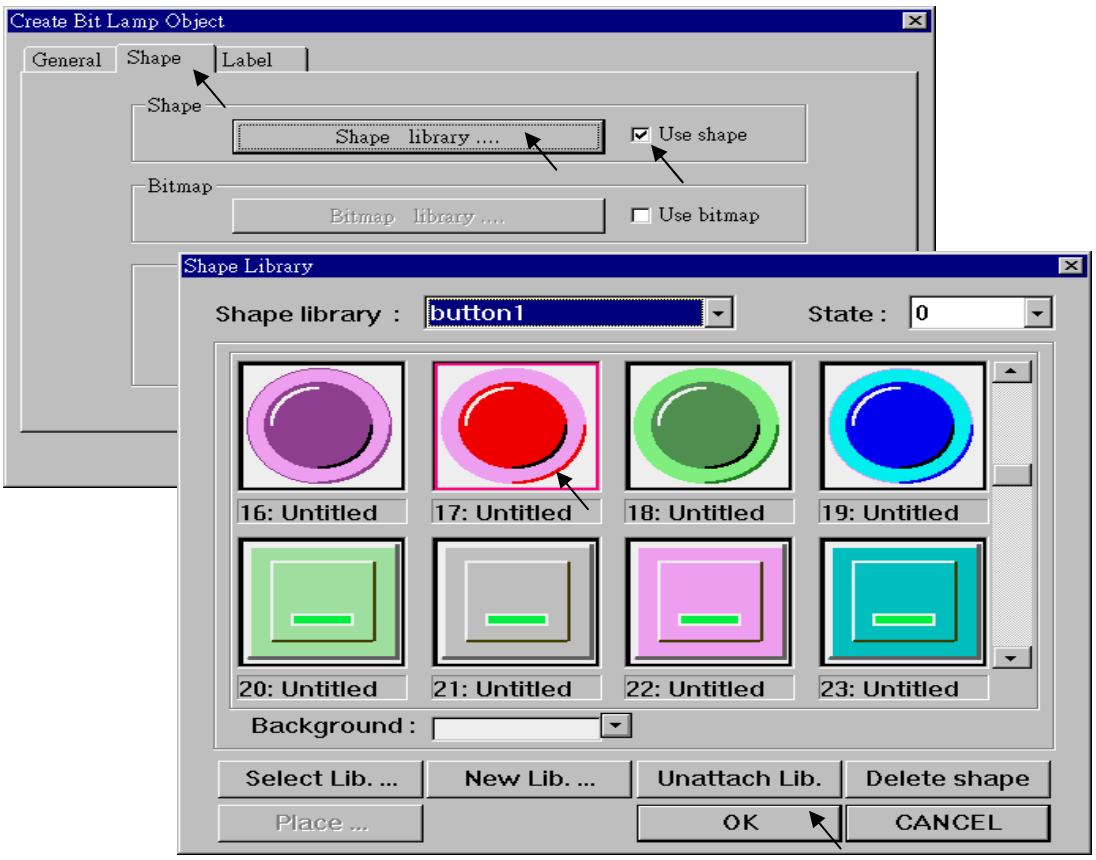
Click on “Bit Lamp”



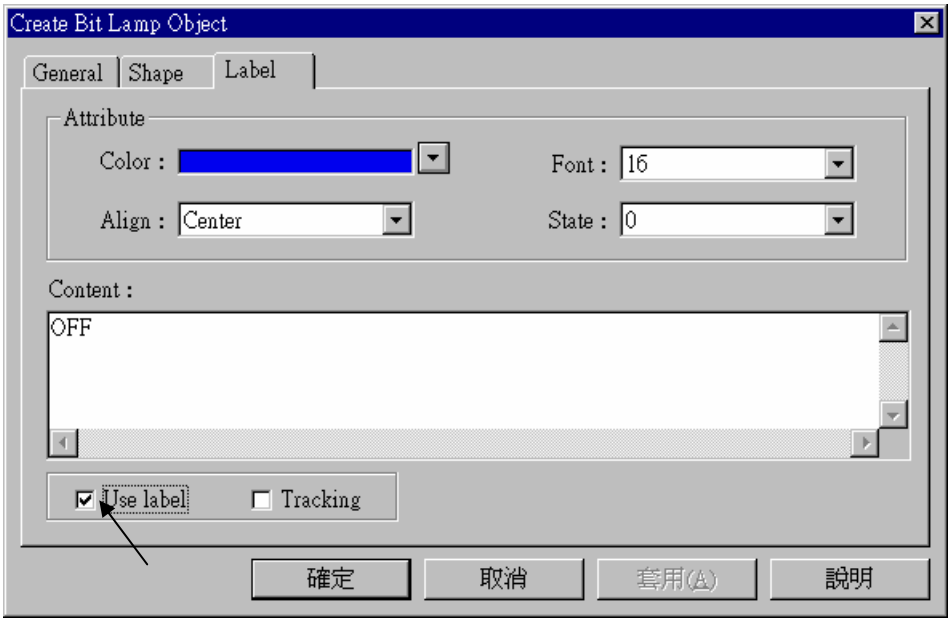
Click on “General”, then select “Device type” to “**0x**” (**0x is for boolean variables**), then set “Device address” to 1 (this value is associated with the network address value of the variable in the I-8xx7). And then set “Function” to “Normal”.



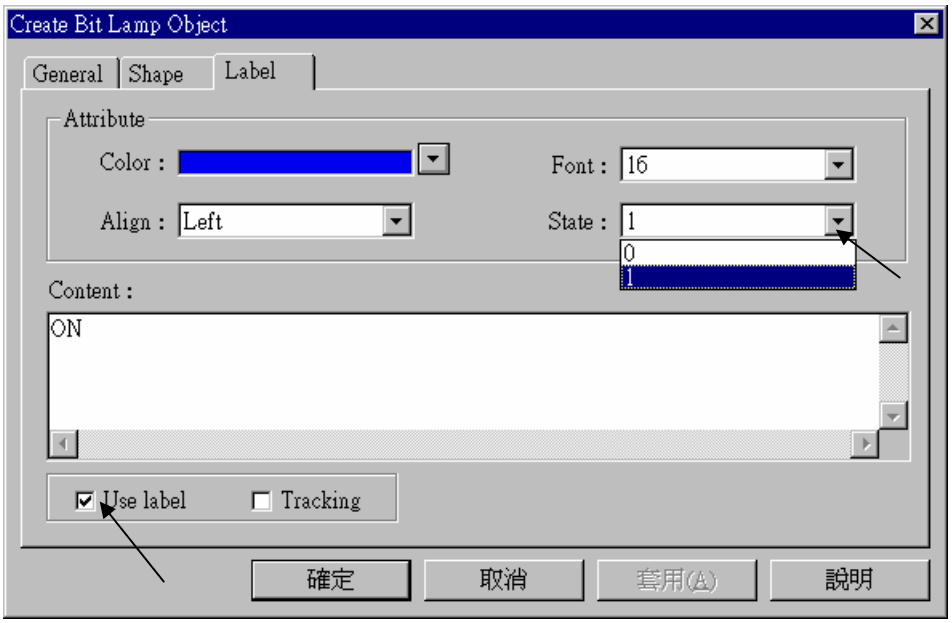
By the same way as former, select prefered “Shap library”.



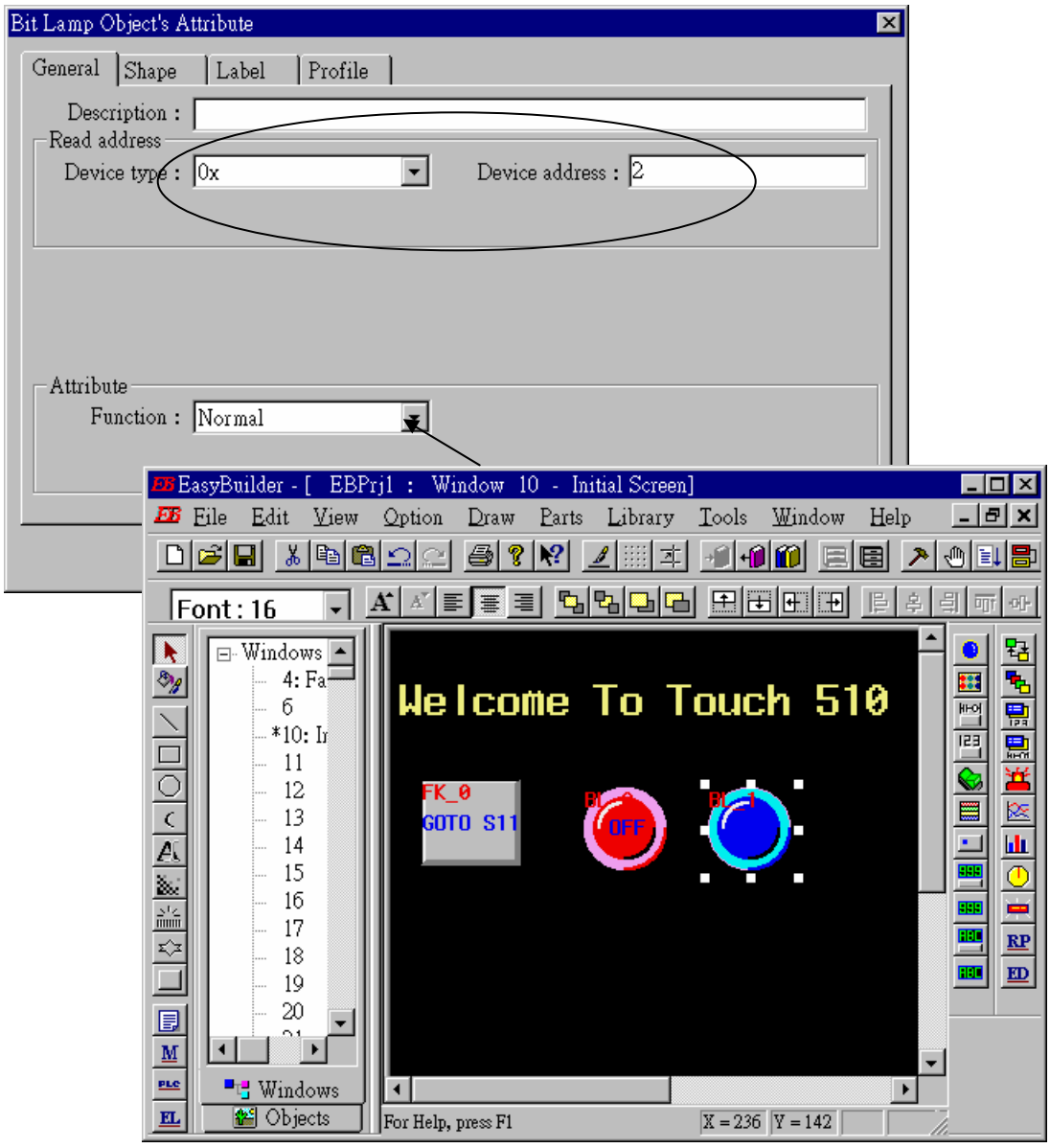
And then select “Label”, given a “OFF” to “Content” for “State : 0”. **Make sure “Use label” is choosed**.



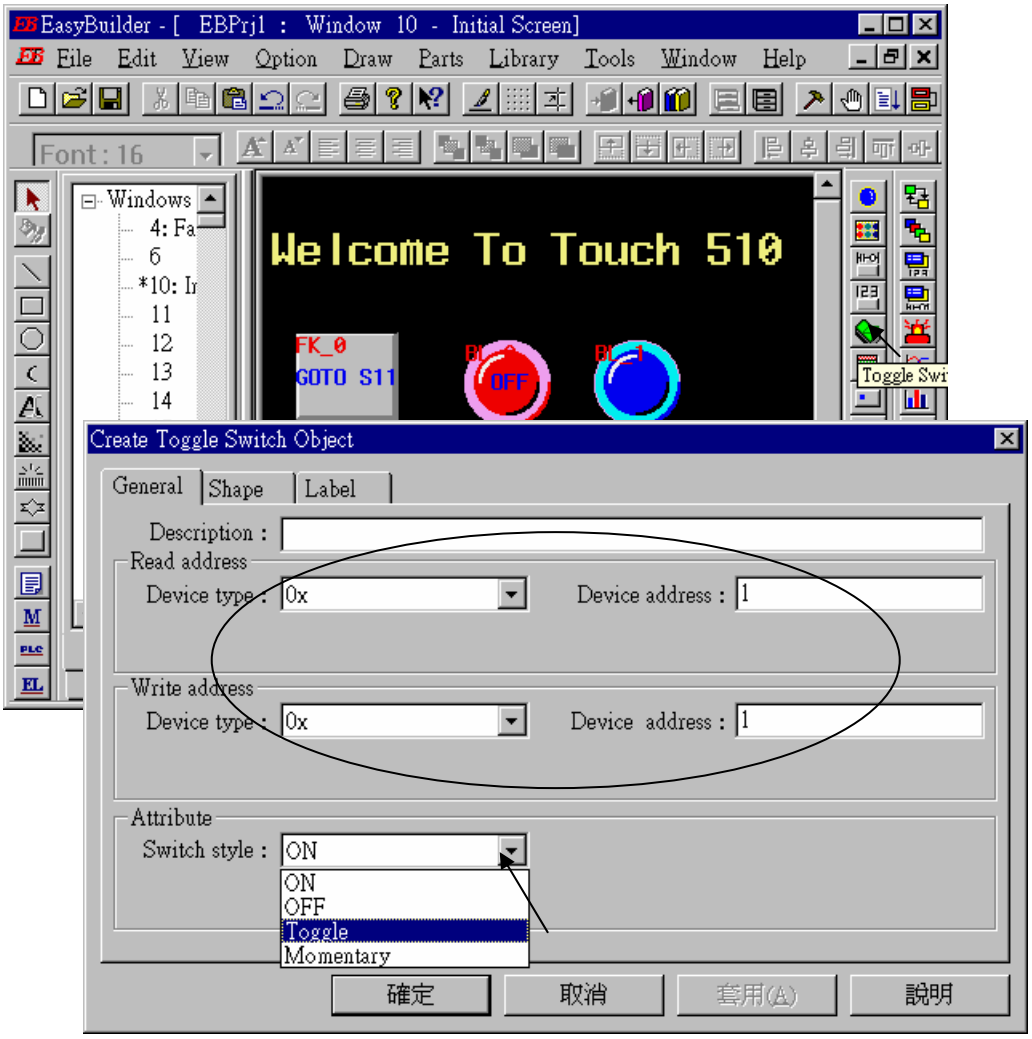
And then change “State” to 1, and given a “ON” to “Content”. **Make sure “Use label” is choosed.**



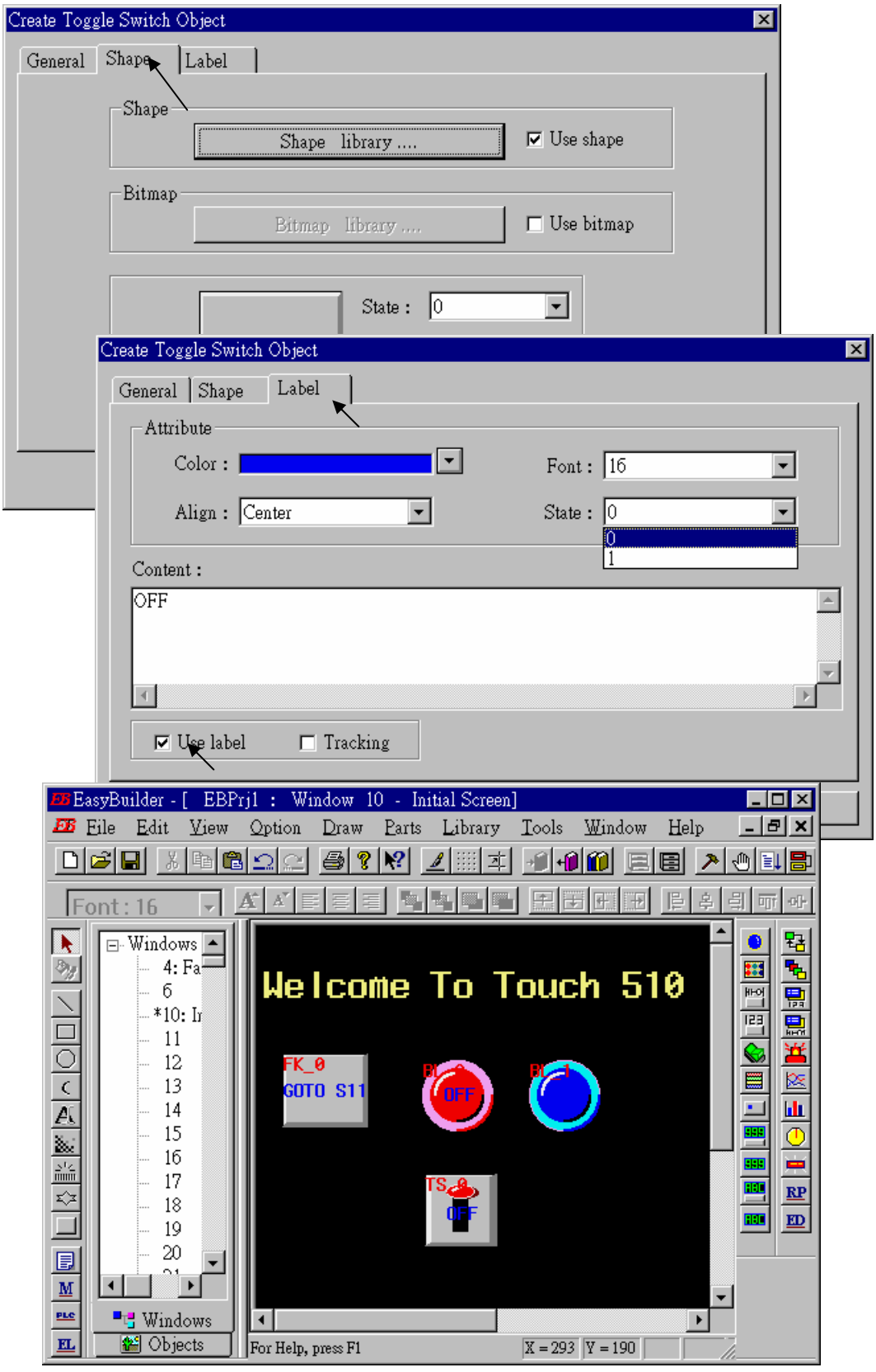
By the same way as former, create one another Bit Lamp with a “Device address” = 2.



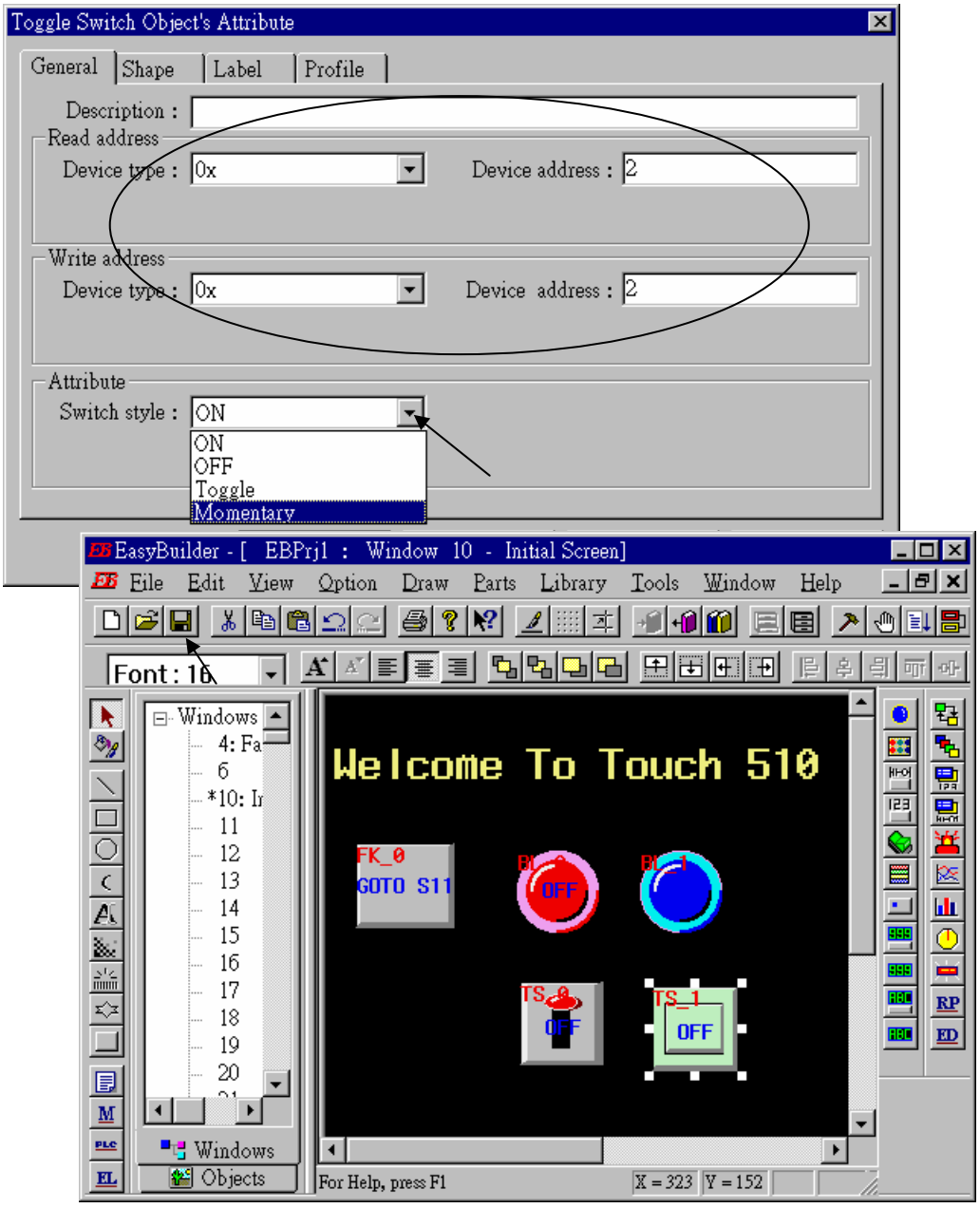
Click on “Toggle Switch”, then set all “Device Type” to “**0x**”, all “Device address” to 1 and select “Switch Type ” to “Toggle”.



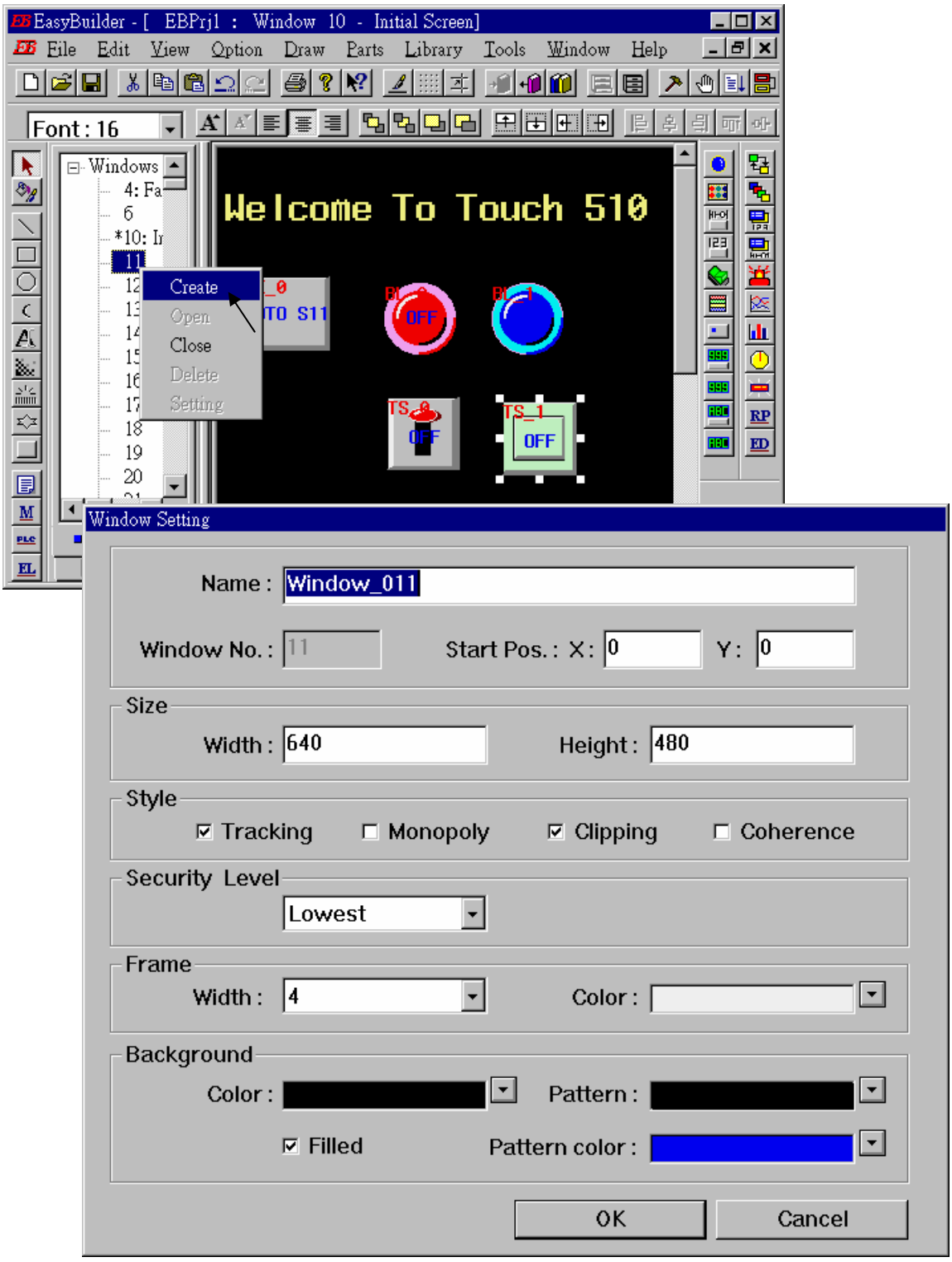
By the same way as former to choose a prefered “shape” and “label”.



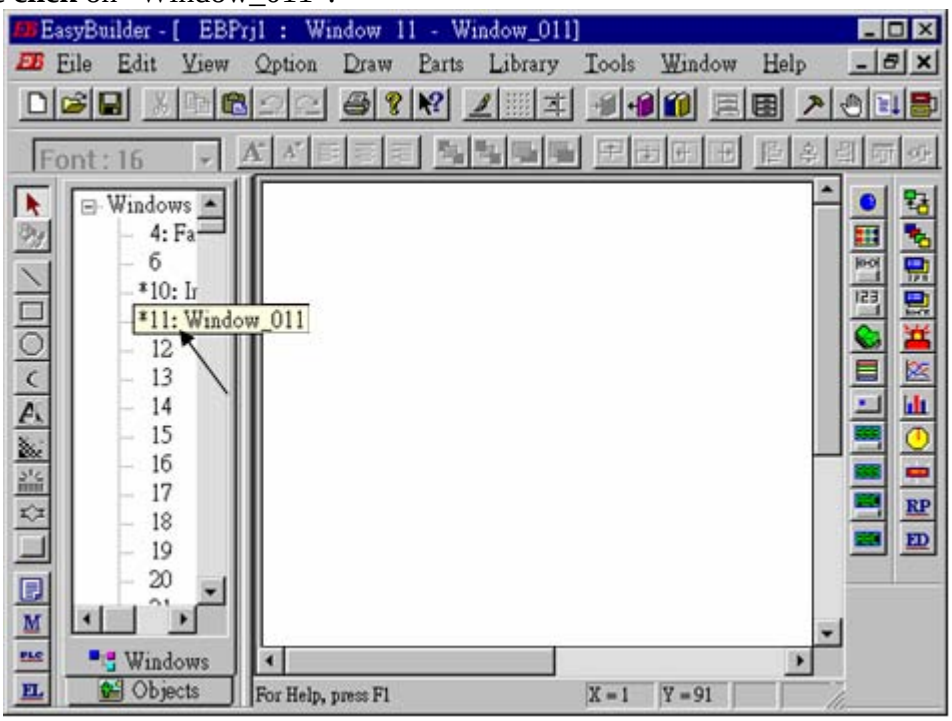
By the same way as former, create one another “Toggle Switch” however set all “Device address” to 2 and “Switch style” to “Momentary”. Click on “save” to save the project.



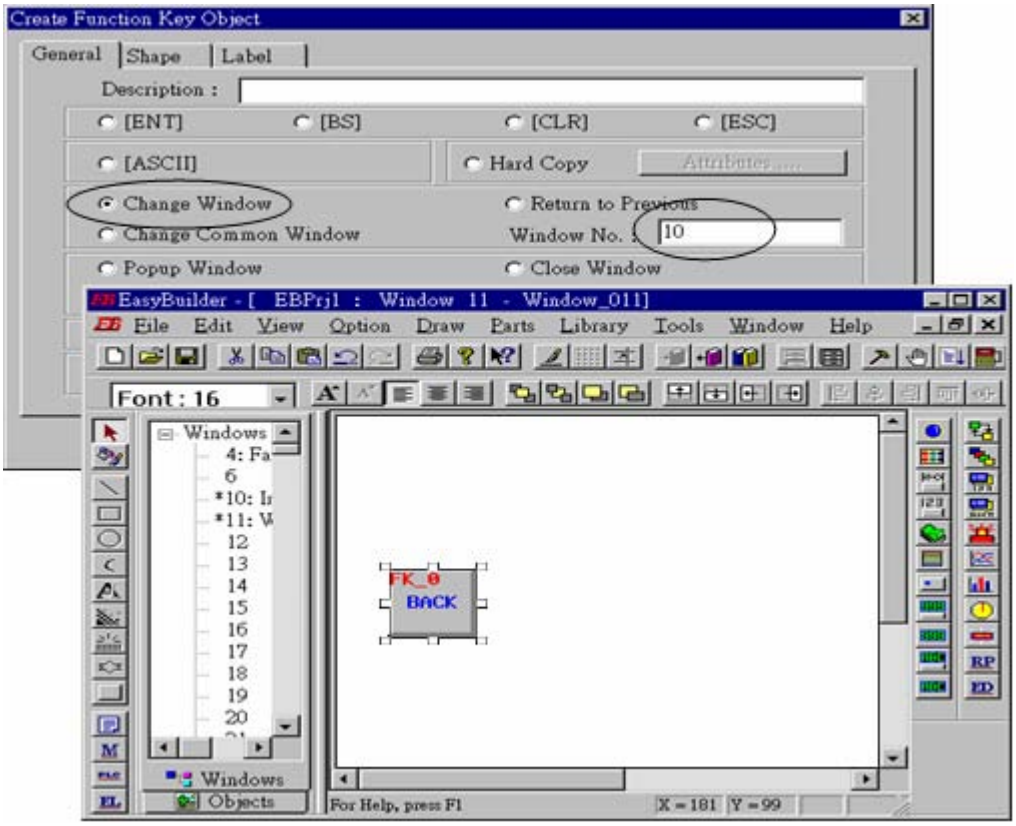
We are going to design another window. Click on “Windows” – “11”, then click and hold on the right button of the mouse and drag to “Create”.



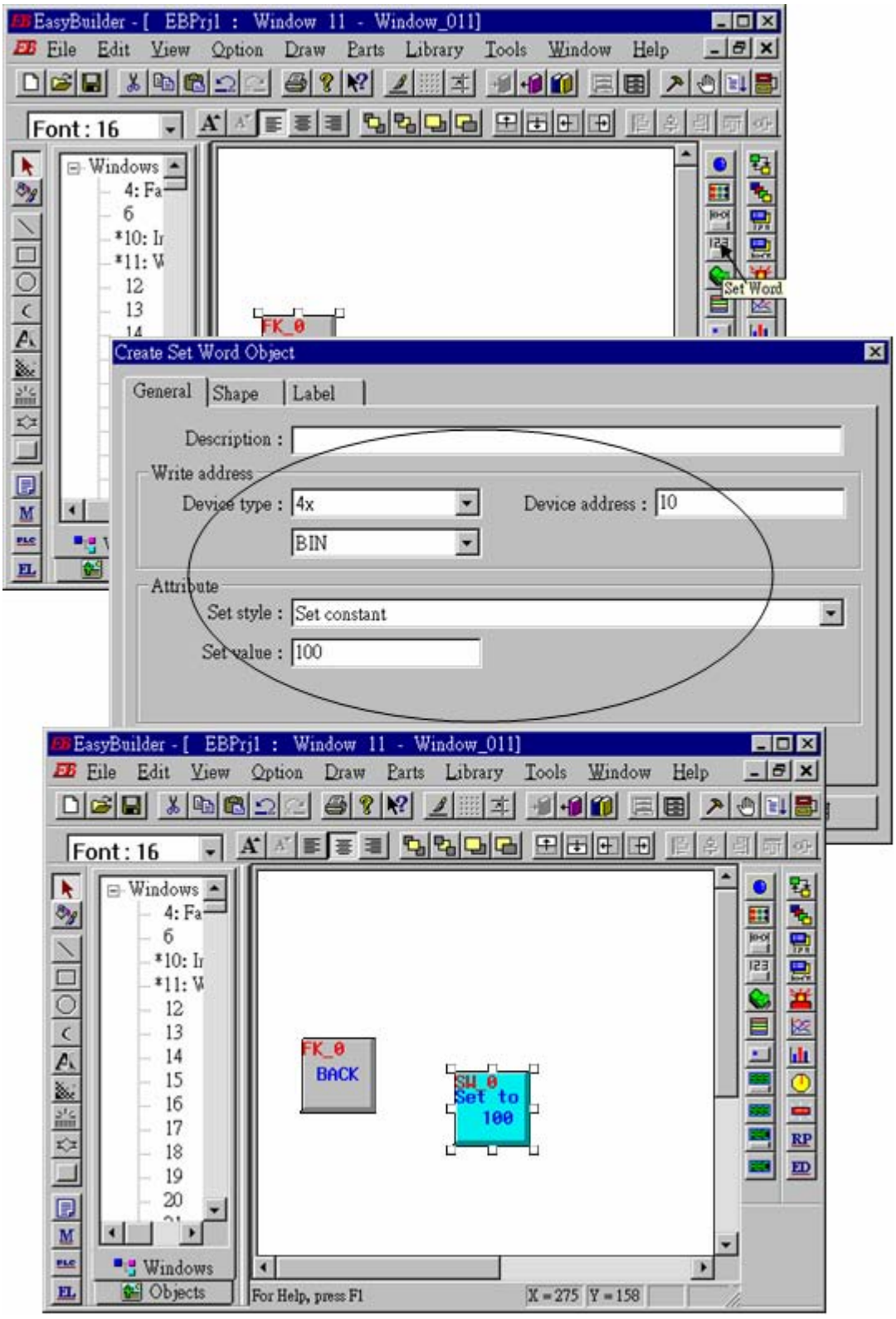
Double click on “Window_011”.



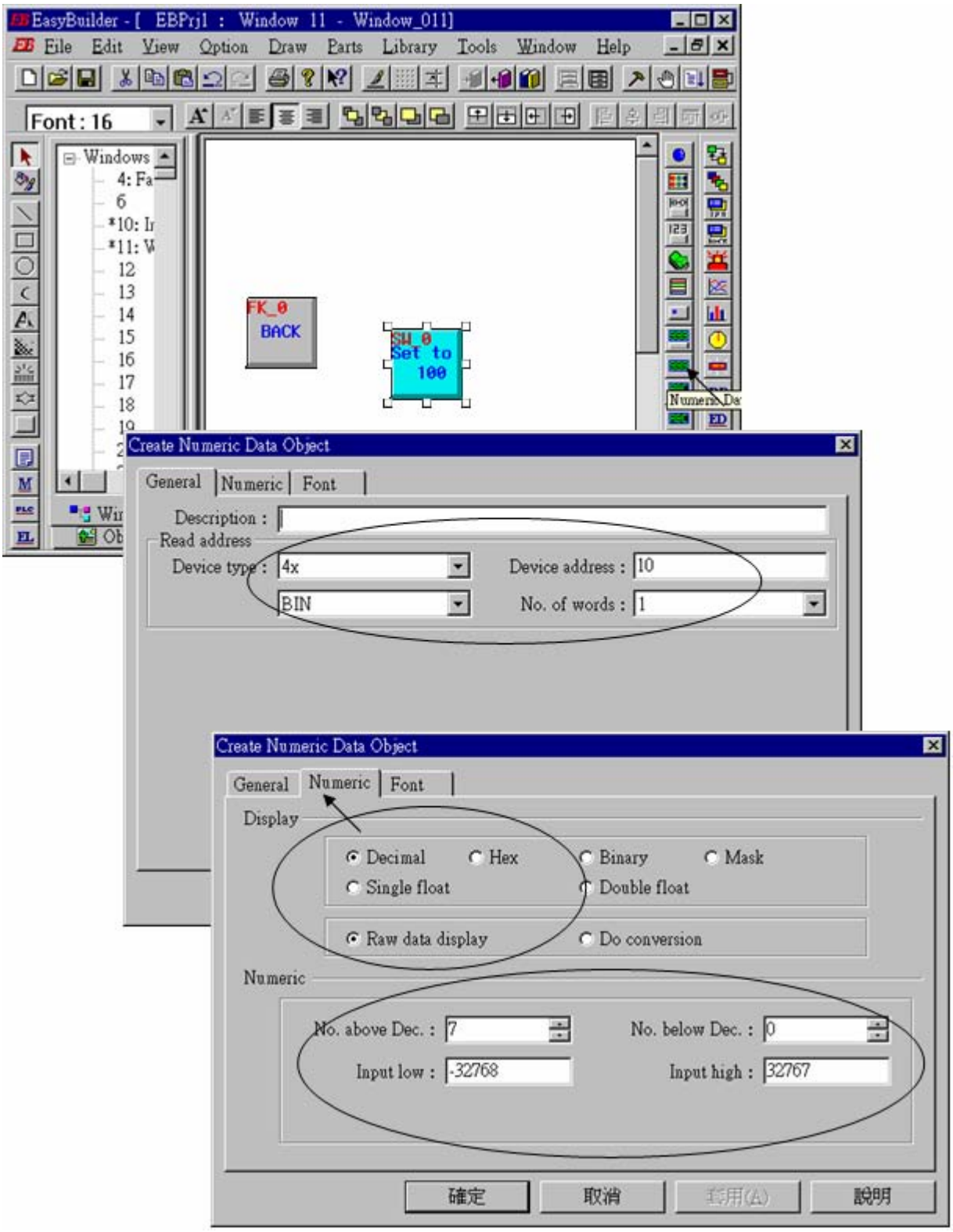
Create a change-window “Function Key” as former method to change to “Window No.” = 10, and Labeled as “BACK”.



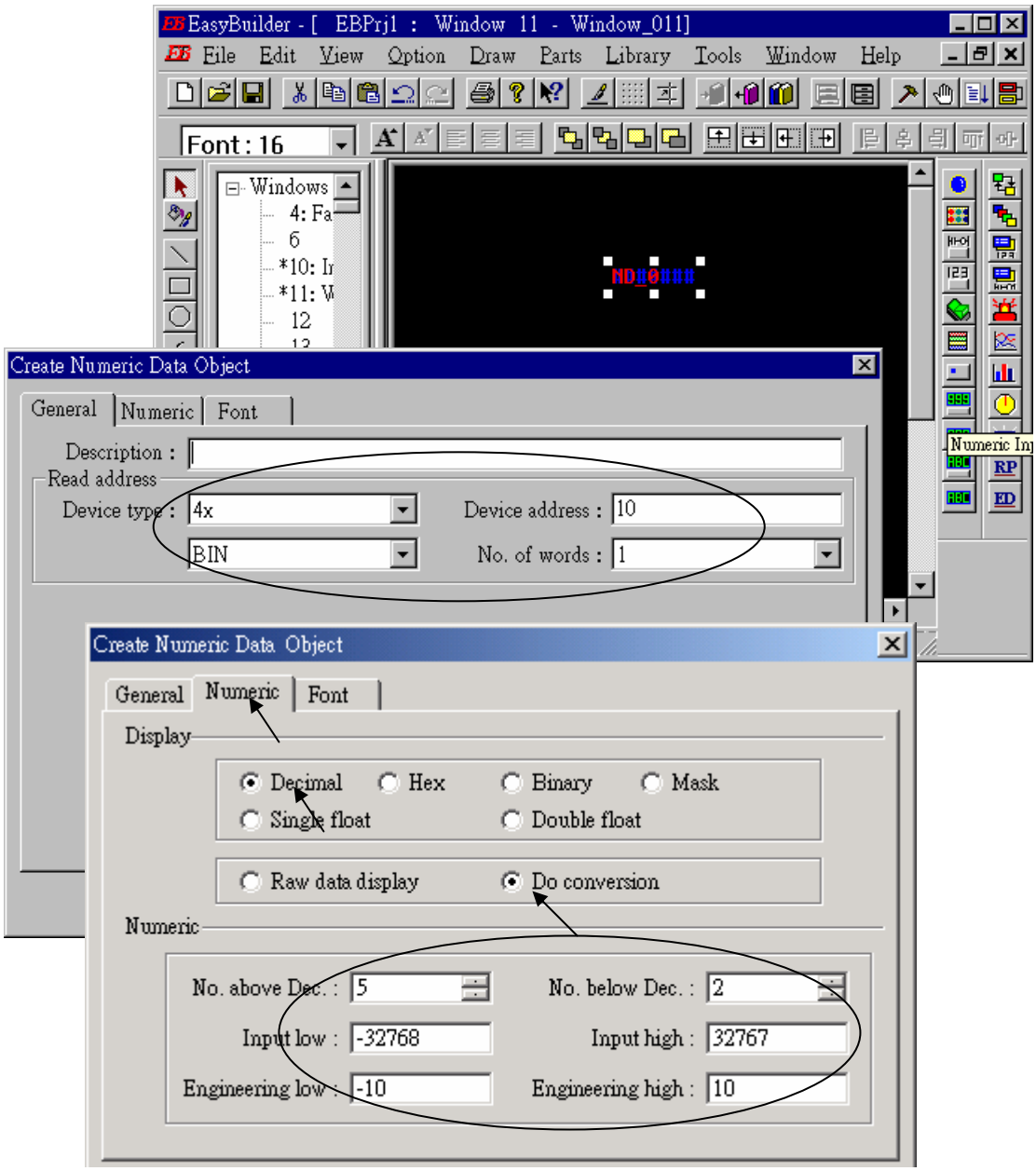
Click on “Set Word”, then set “Device Type” as “**4x**” (**4x is for short integer, 4L is for long integer**), set “Device address” to 10, “BIN”, and “Set style” to “Set Constant”, and “Set value” = 100. And then select the prefered “shape”, and set “label” to “Set to 100”.



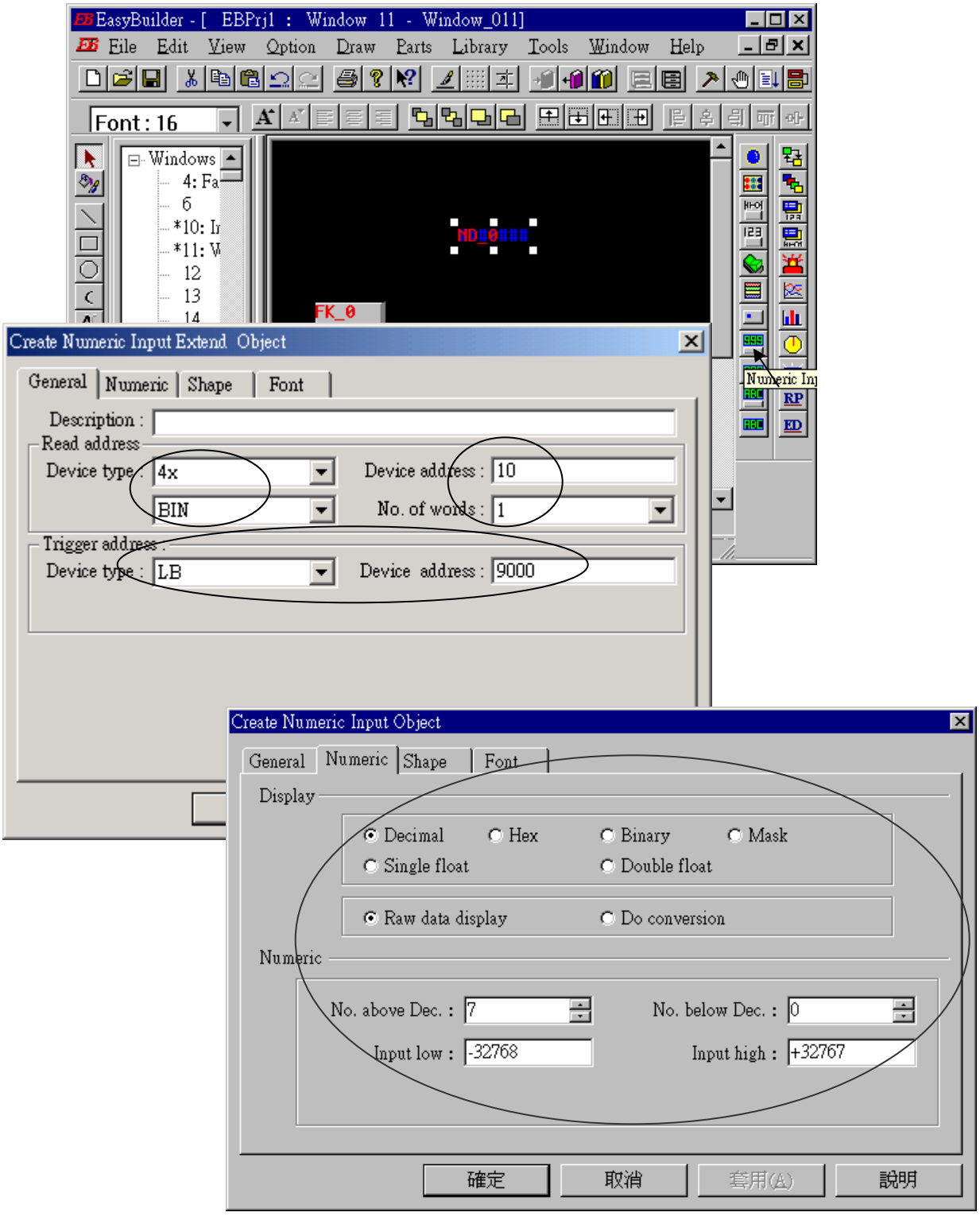
Click on “Numerical Data”, set “Device Type” to “4x” (**4x is for short integer, 4L is for long integer**), “Device address” to 10, “BIN”, “Number of words” to 1, “No. above Dec” to 7, “No. below Decimal” to 0, “Input low” to –32768, “Input high” to +32767. And then select the prefered shape.



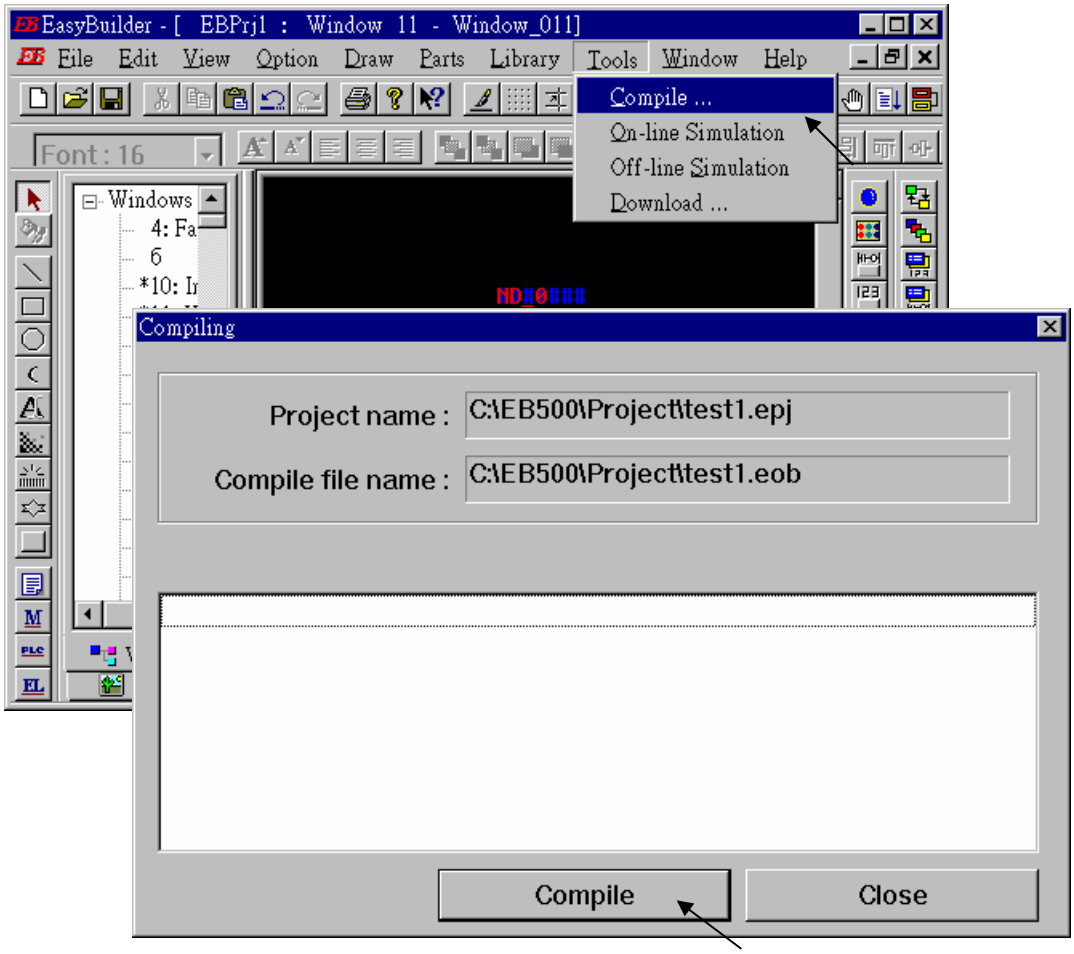
Now we are going to add one another “Numerical Data” with **conversion**.
Click on “Numerical Data”, set “Device Type” to “4x” , “Device address” to 10, “BIN”, “Number of words” to 1, “No. above Dec” to 5, “No. below Decimal” to 0, “Input low” to –32768, “Input high” to +32767, check “Do conversion”, set “engineering low” to –10, “engineering high” to +10 (**Convert [-32768,+32767] to [-10,+10]**). And then select the prefered font.



Click on “Numerical Input”, set “Device Type” to “4x”, “Device address” to 10, “BIN”, “Number of words” to 1, **“Trigger Device Type” to “LB”, “Trigger Device address” to “9000”,** “No. above Dec” to 7, “No. below Decimal” to 0, “Input low” to –32768, “Input high” to +32767. And then select the prefered shape.

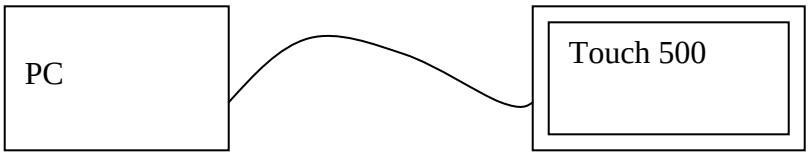


Click “Tools” – “Compile …” to compile this project.

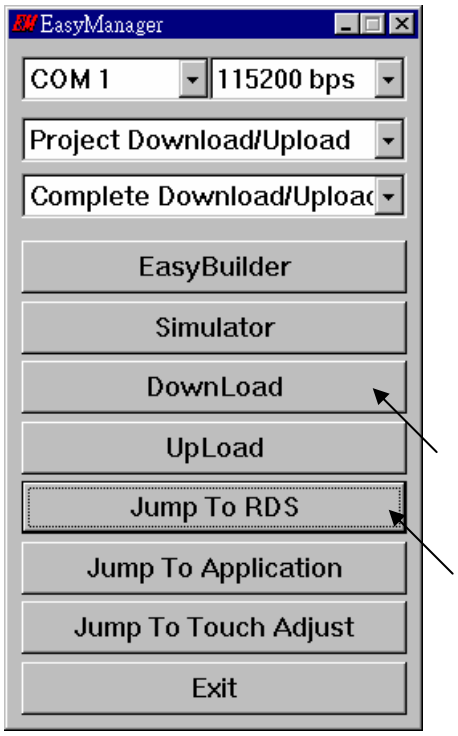


To download the project to the Touch 510, click on the Windows "Start" button, then click on the "Program" button, then click on the "EasyBuilder" – “EasyManager” button. The following window will be displayed. Choose the correct COM No. on your PC (Normally is COM1), “115200 bps”.

Connect the RS232 download cable (refer to section 4.4) between PC and Touch 500.

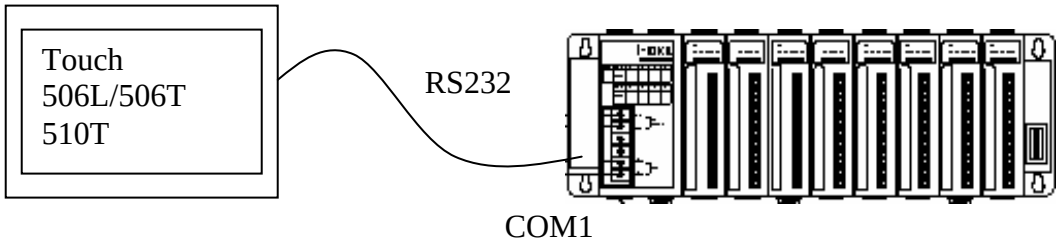


Click on “Jump To RDS” first, if OK., you can see the screen of the Touch 500 will change and wait for project download. Click on “Download” to start to download the MMI picture to the Touch 500.



If downloading is OK, You may choose to click on “Jump To Application” or reset the Touch 510T , and then connect another RS232 cable between Touch 510 and the I-8xx7 (refer to section 4.4).

Now, you may touch each icon on the Touch 510 to test. Have a good luck !



4.5: Access To Word & Integer Array Via Modbus

User can use the below functions to read/write word & integer arrays inside the ISaGRAF project. For more information about these functions, please refer to Appendix A.4.

ARY_N_R	Read one integer(4 byte, signed) from an integer array
ARY_N_W	Write one integer(4 byte, signed) to an integer array
ARY_W_R	Read one word(2 byte, signed) from an word array
ARY_W_W	Write one word(2 byte, signed) to an word array

Word and integer arrays built in the I-8xx7, I-7188EG, I-7188XG, uPAC-7186EG, iPAC-8447/8847 & Wincon-8xx7 controller occupy the same memory area, please use them carefully. Other softwares (HMI, OPC server, …) running on the PC can access to these word and integer arrays via **Modbus** protocol. The valid **network address** for these arrays is from **5001 to 8072 for I-8xx7, I-7188EG & I-7188XG**, while **10,001 to 19,216 for the W-8xx7** and their relation is listed in below table.

For the I-8xx7, I-7188EG, I-7188XG, uPAC-7186EG, iPAC-8447/8847:

Network Address (Decimal)	Word Array	Integer Array
5001	(1,1)	(1,1)
5002	(1,2)	
5003	(1,3)	(1,2)
5004	(1,4)	
…	…	…
…	…	
8071	(12,255)	(6,256)
8072	(12,256)	

For the W-8xx7:

Network Address (Decimal)	Word Array	Integer Array
10001	(1,1)	(1,1)
10002	(1,2)	
10003	(1,3)	(1,2)
10004	(1,4)	
…	…	…
…	…	
19215	(36,255)	(18,256)
19216	(36,256)	

Note:

1. **Network address 1 to 4095 for I-8xx7, I-7188EG/XG, uPAC-7186EG & iPAC-8447/8847, while 1 to 8191 for W-8xx7,** can be defined by users, please refer to Section 4.1.
2. **Modbus address** in the physical transmission format is equal to **Network address** minus one (please refer to Chapter 5). So the valid Modbus address for word & integer arrays is from 5000 to 8071 for I-8xx7, I-7188EG/XG, uPAC-7186EG and iPAC-8447/8847, and 10000 to 19215 for W-8xx7.