

---

# User's Manual of ISaGRAF® PAC

---

By ICP DAS CO. , LTD. & ICP DAS-USA , January 2002, All Rights Reserved

The "User's Manual of ISaGRAF PAC" is intended for integrators, programmers, and maintenance personnel who will be installing and maintaining an I-8417/8817/8437/8837, iPAC-8447/8847, I-7188EG, uPAC-7186EG, I-7188XG , Wincon-8037/8337/8737 & Wincon-8347/8747 series controller system featuring the ISaGRAF Workbench software program.

ICP DAS CO., LTD. would like to congratulate you own your purchase of our ISaGRAF controllers. The ease to integration of the controller system and the power of the IEC 61131-3 ISaGRAF software program combine to make a powerful, yet inexpensive industrial process control system.

## Legal Liability

ICP DAS CO., LTD. assumes no liability for any and all damages that may be incurred by the user as a consequence of this product. ICP DAS CO., LTD. reserves the right to change this manual at any time without notice.

ICP DAS CO., LTD. constantly strives to provide our customers with the most reliable and accurate information possible regarding our products. However, ICP DAS CO., LTD. assumes no responsibility for its use, or for any infringements of patents or other rights of third parties resulting from its use.

## FAQ:

Please visit [www.icpdas.com](http://www.icpdas.com) - "FAQ" - "Software" - "ISaGRAF" for Frequently Asked Question, or visit <http://www.icpdas.com/faq/isagraf.htm>

## Trademark & Copyright Notice

The names of products are used for identification purposes only, and are the registered trademarks of their respective owners or companies.

Copyright January 2002, by ICP DAS CO., LTD. All Rights Reserved.

---

# Table of Contents

---

|   |            |
|---|------------|
| <b>USER'S MANUAL OF ISAGRAF® PAC .....</b>  | <b>1</b>   |
| <b>TABLE OF CONTENTS.....</b>   | <b>2</b>   |
| <b>REFERENCE GUIDE .....</b>  | <b>7</b>   |
| <b>PERFORMANCE COMPARISON TABLE 1 .....</b>   | <b>8</b>   |
| <b>PERFORMANCE COMPARISON TABLE 2 .....</b>   | <b>9</b>   |
| <b>HOW TO SELECT BETWEEN W-8X47 , W-8X37 , I-8XX7, I-7188EG , UPAC-7186EG &amp; I-7188XG.....</b> | <b>10</b>  |
| <b>SPECIFICATIONS: W-8047 / 8347 / 8747 (DUAL ETHERNET) .....</b>                                 | <b>11</b>  |
| <b>SPECIFICATIONS: W-8037 / 8337 / 8737 .....</b>   | <b>13</b>  |
| <b>SPECIFICATIONS: I-8437-80 / I-8837-80 .....</b>  | <b>15</b>  |
| <b>SPECIFICATIONS: I-8417 / 8817.....</b>   | <b>17</b>  |
| <b>SPECIFICATIONS: UPAC-7186EG.....</b>   | <b>19</b>  |
| <b>SPECIFICATIONS: I-7188EG .....</b>   | <b>21</b>  |
| <b>SPECIFICATIONS: I-7188XG .....</b>   | <b>23</b>  |
| <b>SELECTION GUIDE.....</b>   | <b>25</b>  |
| <b>CHAPTER 1. SOFTWARE &amp; HARDWARE INSTALLATION .....</b>                                      | <b>1-1</b> |
| 1.1: INSTALLING THE ISAGRAF WORKBENCH SOFTWARE PROGRAM .....                                      | 1-1        |
| 1.1.1: When closing my ISaGRAF window on windows 2000, it holds. Why ?.....                       | 1-5        |
| 1.1.2: One Quick way to avoid the “hold” problem on windows 2000. ....                            | 1-6        |
| 1.2: INSTALLING THE ICP DAS UTILITIES FOR ISAGRAF .....   | 1-7        |
| 1.3: CONNECTING YOUR PC TO THE CONTROLLER .....   | 1-8        |
| 1.3.1: Setting The NET-ID Addresses For The I-8xx7 Controller System.....                         | 1-8        |
| 1.3.2: Downloading & Communicating Via Modbus With The I-8xx7 .....                               | 1-9        |
| 1.3.3: Connecting Your PC To The I-8xx7 COM1 Port.....  | 1-9        |
| 1.3.4: Connecting Your PC To The I-8xx7 COM2 Port.....  | 1-10       |
| 1.3.5: Connecting One PC To Several I-8417/8817 Controllers .....                                 | 1-10       |
| 1.3.6: Changing The COM1 and COM2 and COM3 Modbus RTU Slave port's Baud Rate Setting .....        | 1-11       |
| 1.3.7: Deleting An ISaGRAF Project From The I-8xx7 Controller.....                                | 1-13       |
| 1.3.8: Connecting Your PC To The I-8437/8837 Ethernet Port .....                                  | 1-14       |
| 1.3.9: Multi-Clients Connection to The I-8437/8837 Ethernet Port .....                            | 1-16       |
| 1.4: CONTROLLER TO CONTROLLER DATA EXCHANGE: FBUS .....   | 1-17       |
| 1.5: LINKING I-7000 AND I-87K MODULES FOR REMOTE I/O.....   | 1-18       |
| 1.6: CREATING A MODBUS LINK WITH THE I-8XX7 CONTROLLER.....                                       | 1-19       |
| 1.7: LINKING TO AN HMI INTERFACE DEVICE .....   | 1-21       |
| 1.8: USING N-PORT COM.....  | 1-22       |
| 1.9: USING MORE MODBUS RTU SLAVE PORTS .....  | 1-23       |
| <b>CHAPTER 2. GETTING STARTED .....</b>   | <b>2-1</b> |
| 2.1: A SIMPLE LADDER LOGIC (LD) PROGRAM .....   | 2-1        |
| 2.1.1: Programming LD.....  | 2-4        |
| 2.1.2: Connecting The I/O .....   | 2-25       |
| 2.1.3: Compiling The Example LD Project.....  | 2-27       |
| 2.1.4: Simulating The LD Project.....   | 2-29       |
| 2.1.5: Download & Debugging The Example LD Project.....   | 2-31       |
| 2.2: A SIMPLE STRUCTURED TEXT (ST) PROGRAM .....  | 2-35       |
| 2.2.1: Example ST Program .....   | 2-39       |
| 2.3: A SIMPLE FUNCTION BLOCK DIAGRAM (FDB) PROGRAM.....   | 2-43       |

|   |            |
|---|------------|
| 2.3.1: Programming The Example FBD Program .....  | 2-43       |
| 2.3.2: Simulating The FBD Program.....  | 2-49       |
| 2.4: A SIMPLE INSTRUCTION LIST (IL) PROGRAM.....  | 2-51       |
| 2.5: A SIMPLE SEQUENTIAL FUNCTION CHART (SFC) PROGRAM .....   | 2-54       |
| 2.5.1: Programming The Example SFC Program .....  | 2-56       |
| 2.5.2: Editing The SFC Program .....  | 2-59       |
| 2.5.3: Simulating The SFC Program .....   | 2-65       |
| 2.6: USING VARIABLE ARRAY .....   | 2-66       |
| 2.6.1 Assign Network address No. To Variable Array .....  | 2-69       |
| 2.6.2 Setting Variable Array As Retained Variable.....  | 2-70       |
| 2.7: HOW TO DETERMINE A D/I IF IT HAS BOUNCING PROBLEM ? .....  | 2-71       |
| 2.8: HOW TO TRIGGER SOMETHING AT SOME SECONDS LATER WHEN ONE EVENT HAPPENS ? .....  | 2-72       |
| <b>CHAPTER 3. ESTABLISHING I/O CONNECTIONS .....</b>  | <b>3-1</b> |
| 3.1: LINKING I/O BOARDS TO AN ISAGRAF PROJECT .....   | 3-3        |
| 3.1.1: Linking I/O Boards.....  | 3-4        |
| 3.1.2: Linking Input & Output Board Variables.....  | 3-5        |
| 3.2: LINKING ANALOG TYPE I/O BOARDS .....   | 3-7        |
| 3.2.1: Setting “range” parameter in analog IO board .....   | 3-7        |
| 3.2.2: Setting special “range” parameter of temperature input board to get clear “Degree Celsius” or “Degree Fahrenheit” input value .....                          | 3-9        |
| 3.2.3: Using I-87017RC .....  | 3-11       |
| 3.2.4: Using I-8017H and I-8017HS .....   | 3-13       |
| 3.2.5: Using I-8080.....  | 3-14       |
| 3.2.6: Using I-87015 to measure up to 7-ch of RTD temperature inputs .....  | 3-16       |
| 3.2.7: Using I-87019R .....   | 3-17       |
| 3.2.8: Using I-8024.....  | 3-18       |
| 3.2.9: Using I-87018Z to measure up to 10-Ch. of thermo-couple inputs .....   | 3-19       |
| 3.2.10: How to measure up to +/-150 VDC voltage input in ISaGRAF controllers with the i-87017W-A5 I/O card ? .....  | 3-20       |
| 3.3: LINKING SOME SPECIAL VIRTUAL BOARD .....   | 3-22       |
| 3.3.1: Using “Push4Key” and “Show3Led” .....  | 3-22       |
| 3.3.2: Using “io_state” to test the operation state of real I/O boards.....   | 3-23       |
| 3.3.3: Using “echo_tim” to delay some milli-seconds before the Modbus RTU Slave port replying to the PC / HMI / SCADA .....   | 3-24       |
| 3.3.4: Using “RTU_Slav” to expand more Modbus RTU Slave ports in Wincon.....  | 3-25       |
| 3.3.5: Using “dis_stop” to disable / enable the ISaGRAF Download function .....   | 3-26       |
| 3.3.6: Using “tcp_para” : Why are my i-8x37, i-8x37-80, i-7188EG and uPAC-7186EG 's TCP connections disconnected and then recovered about every 20/40 hours ? ..... | 3-27       |
| 3.4: DIRECTLY REPRESENTED VARIABLES .....   | 3-28       |
| 3.5: D/I COUNTERS BUILT IN THE I-87XXX D/I MODULES .....  | 3-31       |
| 3.6: AUTO-SCAN I/O.....   | 3-33       |
| 3.7: PWM OUTPUT .....   | 3-35       |
| 3.8: COUNTERS BUILT IN PARALLEL D/I BOARDS .....  | 3-39       |
| <b>CHAPTER 4. LINKING CONTROLLERS TO AN HMI PROGRAM.....</b>  | <b>4-1</b> |
| 4.1: DECLARING VARIABLE ADDRESSES FOR NETWORK ACCESS .....  | 4-1        |
| 4.2:READ/WRITE WORD, LONG WORD & FLOAT THROUGH MODBUS .....   | 4-7        |
| 4.3: USING I-8xx7 AS A MODBUS I/O OR A MODBUS TCP/IP I/O .....  | 4-9        |
| 4.4: LINKING I-8xx7, I-7188EG/XG & W-8xx7 To TOUCH 500 .....  | 4-14       |
| 4.4.1: Program the I-8xx7, I-7188EG/XG & W-8xx7 .....   | 4-15       |
| 4.4.2: Program the Touch 510T.....  | 4-16       |
| 4.5: ACCESS TO WORD & INTEGER ARRAY VIA MODBUS.....   | 4-36       |
| <b>CHAPTER 5. MODBUS PROTOCOL .....</b>   | <b>5-1</b> |
| 5.1: MODBUS PROTOCOL FORMAT: RTU SERIAL.....  | 5-1        |
| 5.2: MODBUS PROTOCOL FORMAT: TCP/IP.....  | 5-7        |
| 5.3: ALGORITHM FOR CRC-16 CHECK .....   | 5-8        |
| <b>CHAPTER 6. LINKING RS-485 I-7000 &amp; I-87XX MODULES.....</b>   | <b>6-1</b> |
| 6.1: CONFIGURING THE I-7000 & I-87XX MODULES .....  | 6-2        |

|   |             |
|---|-------------|
| 6.2: OPENING THE "BUS7000B" FUNCTION .....  | 6-7         |
| 6.3: PROGRAMMING AN I-7000 & I-87XXX MODULE.....  | 6-9         |
| 6.3.1: Program I-7xxx or I-87xxx remote IO function blocks.....   | 6-9         |
| 6.3.2: Setting a special "ADR_" parameter of remote temperature input module to get clear "Degree Celsius" or "Degree Fahrenheit" input value .....   | 6-14        |
| 6.4: REDUNDANCY SYSTEM BY uPAC-7186EG OR I-8437-80.....   | 6-16        |
| 6.4.1: Hot-Swap Redundancy System by uPAC-7186EG plus RU-87P4 / RU-87P8 .....   | 6-16        |
| 6.4.2: Hot-Swap Redundancy System by i-8437-80 plus RU-87P4 / RU-87P8.....  | 6-19        |
| <b>CHAPTER 7. CONTROLLER TO CONTROLLER DATA EXCHANGE .....</b>  | <b>7-1</b>  |
| 7.1: BASIC FBUS RULES.....  | 7-1         |
| 7.2: CONFIGURING AN I-8XX7 TO BE A FBUS "MASTER" OR "SLAVE" .....   | 7-2         |
| 7.2.1: Configuring The Fbus Master Boolean Packages .....   | 7-4         |
| 7.3: PROGRAMMING FBUS PACKAGES.....   | 7-7         |
| 7.4: AN FBUS DATA EXCHANGE EXAMPLE.....   | 7-10        |
| 7.5: PROGRAMMING THE EBUS.....  | 7-15        |
| 7.5.1: Basic Ebus Rules .....   | 7-15        |
| 7.5.2: Configuring the Controller To Be A Ebus "Master" Or "Slave" .....  | 7-17        |
| 7.5.3: Programming Ebus Packages .....  | 7-20        |
| <b>CHAPTER 8. LINKING THE CONTROLLER TO MODBUS RTU &amp; MODBUS ASCII DEVICES .....</b>   | <b>8-1</b>  |
| 8.1: CONFIGURING THE CONTROLLER TO BE A MODBUS MASTER .....   | 8-1         |
| 8.2: PROGRAMMING A MODBUS RTU MASTER.....   | 8-3         |
| 8.3: MODBUS ASCII MASTER .....  | 8-8         |
| 8.4: MULTI-PORTS MODBUS RTU/ASCII MASTER .....  | 8-10        |
| 8.5: WHY CAN NOT CONNECT MODBUS SLAVE DEVICE CORRECTLY ?.....   | 8-13        |
| <b>CHAPTER 9. COMMONLY USED ISAGRAF UTILITIES.....</b>  | <b>9-1</b>  |
| 9.1: CREATING AN ISAGRAF PROJECT GROUPS .....   | 9-2         |
| 9.2: UPLOADING AN ISAGRAF PROJECT .....   | 9-3         |
| 9.3: SETTING AN ISAGRAF PASSWORD .....  | 9-6         |
| 9.4: CREATING AN ISAGRAF PROGRAM DIARY .....  | 9-8         |
| 9.5: BACKING UP & RESTORING AN ISAGRAF PROJECT .....  | 9-9         |
| 9.6: COPYING & RENAMING AN ISAGRAF PROJECT.....   | 9-11        |
| 9.7: SETTING COMMENT TEXT FOR AN ISAGRAF PROJECT .....  | 9-13        |
| 9.8: SETTING THE SLAVE ID FOR AN ISAGRAF CONTROLLER.....  | 9-14        |
| 9.9: OPTIMIZING THE ISAGRAF CODE COMPILER .....   | 9-15        |
| 9.10: USING THE ISAGRAF CONVERSION TABLE .....  | 9-16        |
| 9.11: EXPORT / IMPORT VARIABLE DECLARATIONS VIA MICROSOFT EXCEL .....   | 9-19        |
| 9.12: SPY LIST.....   | 9-22        |
| 9.13: HOW TO SEARCH A VARIABLE NAME IN AN ISAGRAF PROJECT ? .....   | 9-25        |
| <b>CHAPTER 10. THE RETAINED VARIABLE AND DATA BACKUP.....</b>   | <b>10-1</b> |
| 10.1: THE RETAINED VARIABLE .....   | 10-1        |
| 10.2: DATA BACKUP TO THE EEPROM.....  | 10-5        |
| 10.3: BATTERY BACKUP SRAM.....  | 10-7        |
| 10.3.1: Access to the SRAM .....  | 10-8        |
| 10.3.2: Upload data stored in the SRAM.....   | 10-8        |
| 10.3.3: Download data to the SRAM .....   | 10-10       |
| 10.3.4: Operation Functions for the battery backup SRAM .....   | 10-12       |
| 10.4: USING I-8073 - MULTIMEDIACARD TO STORE DATA.....  | 10-12       |
| 10.5: READING & WRITING FILE IN WINCON .....  | 10-13       |
| 10.5.1: Wdemo_51: Read 10 REAL values from a file. Total 10 rows, each contains one REAL value.....   | 10-14       |
| 10.5.2: Wdemo_54: Read 20 REAL values from a file. Total 4 rows, each contains 5 REAL values.....   | 10-16       |
| 10.5.3: Wdemo_55: Read 20 Integer values from a file. Total 2 rows, each contains 10 Integer values .....   | 10-18       |
| 10.5.4: Wdemo_56: Retain values of 1 to 255 Real variable in CompactFlash card.....   | 10-20       |
| 10.5.5: Record i-8017H's Ch.1 to Ch.4 voltage input in a user allocated RAM memory in the Wincon-8xx7 ? The sampling time is one record every 0.01 second. The record period is 1 to 10 minutes. Then PC can download this record and display it as a trend curve diagram by M.S. Excel. .... | 10-30       |
| 10.6: CONTROLLER FAULT DETECTION .....  | 10-31       |
| <b>CHAPTER 11. ISAGRAF PROGRAMMING EXAMPLES .....</b>   | <b>11-1</b> |



|  |             |
|--|-------------|
| 11.1: INSTALLING THE ISAGRAF PROGRAMMING EXAMPLES .....  | 11-1        |
| 11.2: ISAGRAF DEMO EXAMPLE FILES .....   | 11-3        |
| 11.3: DESCRIPTION OF SOME DEMO EXAMPLES .....  | 11-14       |
| 11.3.0 Demo_01A & Demo_03: Do something at specific time.....  | 11-14       |
| 11.3.1 Demo_02 : Start, Stop And Reset Timer .....   | 11-18       |
| 11.3.2 Demo_17 : R/W Integer Value From/To The EEPROM.....   | 11-20       |
| 11.3.3 Demo_29: Store 1200 Short Int Every 75 sec & Send To PC Via Com3.....   | 11-22       |
| 11.3.4 Demo_33 : R/W User Defined protocol Via Com3:RS232/RS485 .....  | 11-27       |
| 11.3.5 Send string to COM2 or COM3 when alarm 1 to 8 happens. (Access to variables as array) .....   | 11-35       |
| 11.3.7 Recording i-8017H 's Ch.1 to Ch.4 voltage input in S-256 / 512 in i-8437-80 or i-8837-80 . The sampling time is one record every 0.05 second. The record period is 1 to 10 minutes. Then PC can download this record and display it as a trend curve diagram by M.S. Excel.....             | 11-39       |
| 11.3.8: How to do periodic operation in ISaGRAF controllers ? .....  | 11-51       |
| 11.3.9: Connecting i-7018z to get 6 channels of 4 to 20 mA input and 4 channles of Thermo-couple temperature input. And then also display the value on PC by VB 6.0 program .....  | 11-52       |
| 11.3.10: Recording i-8017H 's Ch.1 to Ch.4 voltage input in a user allocated RAM memory in the Wincon-8xx7 . The sampling time is one record every 0.01 second. The record period is 1 to 10 minutes. Then PC can download this record and display it as a trend curve diagram by M.S. Excel. .... | 11-58       |
| <b>CHAPTER 12. HOW TO SEND EMAIL WITH ONE ATTACHED FILE BY WINCON-8XX7 OR IPAC-8447 / 8847 OR UPAC-7186EG ?.....</b>   | <b>12-1</b> |
| <b>CHAPTER 13. REMOTELY DOWNLOAD VIA MODEM_LINK.....</b>   | <b>13-1</b> |
| 13.1: INTRODUCTION .....   | 13-1        |
| 13.2: DOWNLOAD PROGRAM VIA MODEM_LINK .....  | 13-2        |
| <b>CHAPTER 14. SPOTLIGHT : SIMPLE HMI .....</b>  | <b>14-1</b> |
| 14.1 A SPOTLIGHT EXAMPLE: .....  | 14-1        |
| <b>CHAPTER 15. CREATING USER-DEFINED FUNCTIONS .....</b>   | <b>15-1</b> |
| 15.1: CREATING FUNCTIONS INSIDE ONE PROJECT .....  | 15-1        |
| 15.2: CREATING FUNCTIONS IN THE LIBRARY .....  | 15-6        |
| <b>CHAPTER 16. LINKING MMICON .....</b>  | <b>16-1</b> |
| 16.1: HARDWARE INSTALLATION .....  | 16-1        |
| 16.2: CREATE BACKGROUND PICTURE OF THE MMICON .....  | 16-2        |
| 16.3: WRITING CONTROL PROGRAM.....   | 16-2        |
| <b>CHAPTER 17. SMS: SHORT MESSAGE SERVICE .....</b>  | <b>17-1</b> |
| 17.1: HARDWARE INSTALLATION .....  | 17-1        |
| 17.2: A SMS DEMO EXAMPLE.....  | 17-2        |
| <b>CHAPTER 18. MOTION.....</b>   | <b>18-1</b> |
| 18.1: INSTALL MOTION DRIVER.....   | 18-1        |
| 18.2: INTRODUCTION .....   | 18-3        |
| 18.2.1: System Block Diagram.....  | 18-3        |
| 18.2.2: DDA Technology .....   | 18-3        |
| 18.3: HARDWARE .....   | 18-5        |
| 18.3.1: I-8000 hardware address.....   | 18-5        |
| 18.3.2: LED Indicator.....   | 18-6        |
| 18.3.3: Hardware Configuration.....  | 18-6        |
| 18.3.4: Pin assignment of connector CN2 .....  | 18-9        |
| 18.4: SOFTWARE.....  | 18-13       |
| I/O connection: .....  | 18-13       |
| Setting commands: .....  | 18-14       |
| M_regist      Register one I-8091 .....  | 18-14       |
| M_r_sys        Reset all setting.....  | 18-15       |
| M_s_var        Set motion system parameters .....  | 18-16       |
| M_s_dir        Define output direction of axes.....  | 18-17       |
| M_s_mode       Set output mode.....  | 18-17       |
| M_s_serv       Set servo ON/OFF .....  | 18-18       |
| M_s_nc         Set N.O. / N.C.....   | 18-18       |

|   |             |
|---|-------------|
| Stop commands: .....  | 18-19       |
| M_stpx      Stop X axis .....   | 18-19       |
| M_stpy      Stop Y axis .....   | 18-19       |
| M_stpall    Stop X & Y axes.....  | 18-19       |
| Simple motion commands: .....   | 18-20       |
| M_lsporg    Low speed move to ORG .....   | 18-20       |
| M_hsporg    High speed move to ORG.....   | 18-20       |
| M_lsppmv    Low speed pulse move .....  | 18-21       |
| M_hsppmv    High speed pulse move.....  | 18-21       |
| M_nsppmv    Normal speed pulse move.....  | 18-22       |
| M_lspmv     Low speed move.....   | 18-22       |
| M_hspmv     High speed move .....   | 18-23       |
| M_cspmv     Change speed move .....   | 18-23       |
| M_slwdn     Slow down to low speed.....   | 18-24       |
| M_slwstp    Slow down to stop .....   | 18-24       |
| Interpolation commands: .....   | 18-25       |
| M_intp      Move a short distance on X-Y plane .....  | 18-25       |
| M_intln     Move a long distance on X-Y plane .....   | 18-26       |
| M_intln2    Move a long distance on X-Y plane .....   | 18-27       |
| M_intcl2    Move a circle on X-Y plane.....   | 18-28       |
| M_intar2    Move a arc on X-Y plane.....  | 18-29       |
| M_intstp    Test X-Y plane moving command.....  | 18-30       |
| I-8090 encorder commands: .....   | 18-31       |
| M_r_enco    Reset I-8090's encorder value to 0 .....  | 18-31       |
| <b>CHAPTER 19. ETHERNET COMMUNICATION AND SECURITY .....</b>  | <b>19-1</b> |
| 19.1: ETHERNET SECURITY.....  | 19-1        |
| 19.2: DELIVERING MESSAGE VIA UDP .....  | 19-3        |
| 19.3: WINCON AUTOMATICALLY REPORT DATA TO LOCAL PC OR REMOTE INTERNET PC VIA TCP/IP . OR WINCON<br>COMMUNICATE WITH OTHER TCP/IP SERVER DEVICE..... | 19-5        |
| <b>CHAPTER 20. REDUNDANCY SOLUTIONS .....</b>   | <b>20-1</b> |
| 20.1: WINCON-8XX7 CPU REDUNDANCY PLUS I-87K I/O .....   | 20-2        |
| 20.2: WINCON-8XX7 CPU REDUNDANCY PLUS I-87K I/O & MODBUS RTU DEVICES.....   | 20-7        |
| 20.3: CONNECTING PC / HMI TO MODBUS RTU RS485 PORTS .....   | 20-9        |
| 20.4: CONNECTING PC / HMI TO A REDUNDANCY SYSTEM WITH A SINGLE IP ADDRESS . .....   | 20-10       |
| 20.5: SETTING UP A REDUNDANCY SYSTEM WITH ETHERNET I/O .....  | 20-12       |
| <b>CHAPTER 21. CONNECTING M-7000 SERIES I/O MODULES .....</b>   | <b>21-1</b> |
| 21.1: USING DCON UTILITY TO DO INITIAL SETTING FOR M-7000 .....   | 21-2        |
| 21.2: WRITTING PROGRAM TO CONNECT TO I-7000 MODULES.....  | 21-6        |
| <b>CHAPTER 22. CONNECTING I-8KE4-MTCP AND I-8KE8-MTCP ETHERNET I/O.....</b>   | <b>22-1</b> |
| 22.1: INDUCTION OF THE I-8KE4-MTCP AND I-8KE8-MTCP .....  | 22-1        |
| 22.2: PROGRAMMING THE I-8KE4-MTCP AND I-8KE8-MTCP .....   | 22-3        |
| <b>CHAPTER 23. WINCON AND IPAC-8X47 CONNECTING FAST FRNET REMOTE I/O.....</b>   | <b>23-1</b> |
| <b>CHAPTER 24. SOME SKILL TO OPERATE THE RS-232 / 422 / 485 SERIAL COM PORT BY COM FUNCTIONS<br/>.....</b>  | <b>24-1</b> |
| 24.1: CNTROLLER SEND 1 REQUEST AND GET 1 REPLY FROM DEVICE .....  | 24-2        |
| 24.2: CONTROLLER JUST WAIT DATA FROM THE REMOTE DEVICE.....   | 24-3        |
| 24.3: REPORT DATA TO REMOTE DEVICE PERIODICALLY .....   | 24-4        |
| 24.4: CONTROLLER SEND DATA WHEN EVENT HAPPENS .....   | 24-5        |

---

# Reference Guide

---

English manual:

"user\_manual\_i\_8xx7.pdf" and "user\_manual\_i\_8xx7\_appendix.pdf" in the following directory.

I-8000 & I-7188 CD: \napdos\isagraf\8000\english\_manu\

W-8xx7 CD: \napdos\isagraf\wincon\english\_manu\

中文 ISaGRAF 進階使用手冊:

"chinese\_user\_manual\_i\_8xx7.pdf" and "chinese\_user\_manual\_i\_8xx7\_appendix.pdf" in the following directory.

I-8000 & I-7188 CD: \napdos\isagraf\8000\chinese\_manu\

W-8xx7 CD: \napdos\isagraf\wincon\chinese\_manu\

I-8000 Hardware Manual:

Please refer to I-8000 CD\NAPDOS\8000\index.htm .

Resource on the Internet:

Newly updated ISaGRAF IO libraries, drivers and manuals can be found at

<http://www.icpdas.com/products/PAC/i-8000/isagraf.htm>

Technical Service:

Please contact local agent or email problem-report to [service@icpdas.com](mailto:service@icpdas.com)

New information can be found at [www.icpdas.com](http://www.icpdas.com)

Industrial Ethernet Switch : NS-205 / NS-208

Best choice for Industrial Ethernet Communication.

[http://www.icpdas.com/products/Switch/industrial/ethernet\\_switch.htm](http://www.icpdas.com/products/Switch/industrial/ethernet_switch.htm)



**Model: NS-205**



**Model: NS-208**

FAQ:

Please visit [www.icpdas.com](http://www.icpdas.com) - "FAQ" - "Software" - "ISaGRAF" for Frequently Asked Question, or

visit <http://www.icpdas.com/faq/isagraf.htm>

## Performance Comparison Table 1

| <i>PACs</i>                                     | <b>CPU</b>                               | <b>Normal running Speed<br/>(Normal PLC scan-time)</b> | <b>Normal Speed for floating point calculation<br/>(scan-time)</b> | <b>Ethernet</b>                      | <b>ISaGRAF code size limitation<br/>(bytes)</b> | <b>Memory for running program<br/>(bytes)</b> |
|---|--|--|--|--------------------------------------|---|---|
| <b>iPAC-8447</b><br><b>iPAC-8847</b>            | 80186- <b>80M</b> Hz or compatible       | About <b>4</b><br>(2 ~25ms)                            | About <b>0.8</b><br>(10~125ms)                                     | 1 port<br><b>10/100</b> Mbps         | 64K   | About <b>1M</b>                               |
| <b>i-8437-80</b><br><b>i-8837-80</b>            | 80186- <b>80M</b> Hz or compatible       | About <b>4</b><br>(2~25ms)                             | About <b>0.8</b><br>(10~125ms)                                     | 1 port<br>10 Mbps                    | 64K   | About 512K                                    |
| <b>i-8437</b><br><b>i-8837</b>                  | 80188-40M Hz or compatible               | About 1<br>(5~100ms)                                   | About <b>0.2</b><br>(25~500ms)                                     | 1 port<br>10 Mbps                    | 64K   | About 512K                                    |
| <b>i-8417</b><br><b>i-8817</b>                  | 80188-40M Hz or compatible               | About 1<br>(5~100ms)                                   | About <b>0.2</b><br>(25~500ms)                                     | No                                   | 64K   | About 512K                                    |
| <b>μPAC-7186EG</b>                              | 80186- <b>80M</b> Hz or compatible       | About <b>4</b><br>(2~25ms)                             | About <b>0.8</b><br>(10~125ms)                                     | 1 port<br><b>10/100</b> Mbps         | 64K   | About <b>640K</b>                             |
| <b>i-7188EG</b>                                 | 80188-40M Hz or compatible               | About 1<br>(5~100ms)                                   | About <b>0.2</b><br>(25~500ms)                                     | 1 port<br>10 Mbps                    | 64K   | About 512K                                    |
| <b>i-7188XG</b>                                 | 80188-40M Hz or compatible               | About 1<br>(5~100ms)                                   | About <b>0.2</b><br>(25~500ms)                                     | No                                   | 64K   | About 512K                                    |
| <b>W-8047</b><br><b>W-8347</b><br><b>W-8747</b> | Strong-ARM <b>206 M</b> Hz or compatible | About <b>10~20</b><br>(3~15ms)                         | About <b>10~20</b><br>(3~15ms)                                     | <b>2</b> ports<br><b>10/100</b> Mbps | <b>1M</b>                                       | About <b>20~40M</b>                           |
| <b>W-8037</b><br><b>W-8437</b><br><b>W-8837</b> | Strong-ARM <b>206 MHz</b> or compatible  | About <b>10~20</b><br>(3~15ms)                         | About <b>10~20</b><br>(3~15ms)                                     | 1 port<br>10 Mbps                    | <b>1M</b>                                       | About <b>20~40M</b>                           |

## Performance Comparison Table 2

| PACs  | μPAC Series                             |              |                           | iPAC Series                             |                                     |                            | WinPAC Series                |
|---|---|--------------|---------------------------|---|-------------------------------------|----------------------------|------------------------------|
|   | i-7188 XG                               | i-7188 EG    | μPAC-7186EG               | i-8417 / i-8817                         | i-8x37 / i-8x37-80<br><div>*1</div> | iPAC-8x47<br><div>*2</div> | W-8447/8847<br>W-8437/8837   |
| Support Ethernet I/O<br>(i-8KE4-MTCP<br>i-8KE8-MTCP)          | No                                      |              |                           | No                                      |                                     |                            | Yes                          |
| Send E-mail<br>(file attached)                                | No                                      |              | Yes<br><div>*3</div>      | No                                      |                                     | Yes                        | Yes                          |
| Max. amount for linking i-7K/ 87K remote I/Omodule            | 64 (COM2 or 3)<br>(support 1 port only) |              |                           | 64 (COM3 or 4)<br>(support 1 port only) |                                     |                            | <div>255</div><br>(COM3)     |
| Modbus Master Function Block<br>Max. amount                   | 64<br>(Total)                           |              | <div>128</div><br>(Total) | 64<br>(Total)                           |                                     | <div>128</div><br>(Total)  | <div>256</div><br>(per port) |
| Available Modbus Master COM Port<br>max. amount <div>*4</div> | 2 ports                                 |              |                           |   |                                     | <div>4 ports</div>         | <div>10 ports</div>          |
|   | COM 2, 3                                | COM 1, 2, 3  |                           | COM 1, 3, 4, 5                          |                                     | COM 1, 3~8                 | COM 2, 3, 5~14               |
| Available Modbus Slave COM Port <div>*4</div>                 | 2 ports                                 |              |                           |   |                                     |                            | 5 ports                      |
|   | COM1, 3                                 |              |                           | COM1, 2                                 | COM1, 3                             |                            | COM2 or 3, 5~8               |
| Support Modbus TCP/IP connections                             | 0                                       | <div>4</div> |                           | 0                                       | <div>4</div>                        |                            | <div>8</div>                 |
| Modbus Address Range  | 1~4095                                  |              |                           | 1~4095                                  |                                     |                            | 1~8191                       |
| Data Exchange   | Fbus                                    | Fbus & Ebus  |                           | Fbus                                    | Fbus & Ebus                         |                            | Ebus                         |
| FRNET I/O   | No                                      |              |                           | No                                      |                                     | Yes <div>*5</div>          | Yes <div>*5</div>            |

### Annotations:

- \*1. i-8x37/i-8x37-80 represents the products of i-8437/ 8837/ 8437-80/ 8837-80.
- \*2. iPAC-8x47 represents the products of iPAC-8447/ 8847 that will be available about Q1/Q2, 2008.
- \*3. μPAC-7186EG has to use an extra X-607/ 608 battery SRAM expansion card for sending Email with an attached file, or it can only send Email without attached file.
- \*4. i-8000's COM5~20 & W-8x47/ 8x37's COM5~14 resides at the i-8112/8114/8142/ 8144/8142i expansion modules; i-7188/μPAC-7186's COM3~8 resides at the X-5xx expansion boards.
- \*5. iPAC-8x47 support Max. 4 pcs. of i-8172 (Max 1024 D/I and 1024 D/O channels)  
W-8x47/ 8x37 support Max. 7 pcs. of i-8172 (Max 1792 D/I and 1792 D/O channels).

---

# How to select between W-8x47 , W-8x37 , I-8xx7, I-7188EG , uPAC-7186EG & I-7188XG

---

## Memory considerations:

1. The I-8417/8817/8437/8837 , iPAC-8447/8847, I-7188EG , uPAC-7186EG and I-7188XG has memory limitation. The ISaGRAF code size can not exceeds 64K bytes. (size of the “appli.x8m” file)
2. W-8037/8337/8737 and W-8047/8347/8747 has code size limitation of 1M bytes. The size is 16 times of the size of I-8xx7 , iPAC-8447/8847, uPAC-7186EG & I-7188EG/XG.

## CPU speed considerations:

The CPU of I-8417/8817/8437/8837 , I-7188EG and I-7188XG is 80188 (40MHz) or compatible. It is a 16-bit cpu. It is not good at doing floating point value calculation. If your application will do lots of floating point value calculation, it is better to use W-8037/8337/8737 and W-8047/8347/8747 or future advanced ISaGRAF controllers. The CPU is 32-bit and its speed is about 10 to 20 times compared with the I-8xx7 & I-7188EG/XG, especially for floating point value calculation.

The speed of I-8437-80, I-8837-80, iPAC-8447/8847 and uPAC-7186EG are about 4-times of the I-8xx7 & I-7188EG/XG.

## Redundancy considerations:

Wincon-8047/8347/8747 supports redundancy solution. Two controllers to be one redundancy system. One is redundant Master, one is redndant slave. Master handles all inputs & outputs of the remote RS-485 I/O (I-7k & I-87K) at run time. If master is dead, Slave will take over the control of the remote I/O. **All Outputs** should be configured as RS-485 remote I/O. **Inputs** can locate at slot 1 through 7 or configured as RS-485 remote I/O. Redundant Change Over Time: <= 500 ms, Synchronization: <= 75ms

## Ethernet considerations:

1. The W-8047/8347/8747 ‘s ethernet is 10/100 Mbps and dual ports. All other current ISaGRAF controllers have only one Ethernet port.

uPAC-7186EG and iPAC-8447/8847 ‘s Ethernet are 10/100 Mbps.

I-7188EG , W-8037/8337/8737 & I-8437/8837 are 10 Mbps.

All of above controllers support Modbus TCP/IP slave protocol.

I-7188XG & I-8417/8817 don’t support Ethernet.

2. W-8037/8337/8737 , W-8347/8747, uPAC-7186EG and iPAC-8447/8847 support sending email with one attached file and sending / receiving user’s defined message (string) via UDP/IP or TCP/IP to PC or other devices, however I-7188EG & I-8437/8837 no supporting them.

## Modbus RTU Slave ports:

W-8347/8747 & W-8337/8737 can support max. 5 Modbus RTU slave ports (Please refer to Appendix G, E and A.2 of its Getting Started Manual in the product box).

I-7188EG/XG and uPAC-7186EG can support max. 2 Modbus RTU slave ports (Please refer to Chapter 3.7 & 3.6 of its Getting Started Manual in the product box).

I-8xx7 and iPAC-8447/8847 can support max. 2 Modbus RTU slave ports (Please refer to Chapter 3.5 & 3.10 of its Getting Started Manual in the product box).

## Specifications: W-8047 / 8347 / 8747 (Dual Ethernet)

|                      |  |
|----------------------|--|
| Development software |  |
| ISaGRAF Version 3    | IEC61131-3 standard. Languages: LD, ST, FBD, SFC, IL & FC  |
| Max. code size       | accepts max. 1M bytes ISaGRAF code size (Appli.x8m must < 1M)  |
| Non-ISaGRAF          | Options: Microsoft EVC++4.0 or VS.NET 2003 (VB.NET, C#.NET)  |
| Web HMI              | PC running Internet Explorer can access to the Wincon-8047/8347/8747 via Local Ethernet or Internet or dial Modem, monitoring and Control.   |
| Security             | Three Level username and password protection   |
| Power supply         | 10 to 30VDC (unregulated), 20W (when I/O slots are empty )   |
| Protection           | Built-in power protection & network protection circuit   |
| General environment  |  |
| Temperature          | Operating: -25 to +75°C , Storage: -30 to +85°C,   |
| Humidity             | 5 to 95 % (non-condensed)  |
| System               |  |
| CPU                  | Intel Strong ARM CPU, 206MHz, or compatible  |
| Watchdog timer       | Yes  |
| Real time clock      | Gives hour, minute, sec, date of week, date of month, month & year   |
| SDRAM & FLASH        | SDRAM:64M bytes , FLASH Memory: 32M bytes for OS image   |
| Compact Flash Card   | One Compact Flash slot: CF memory card is 128M bytes or more   |
| EEPROM               | 16K bytes, retention > 100 years. 1,000,000 erase/write cycles   |
| I/O slots            | I/O slots: W-8047: 0 , W-8347: 3 , W-8747: 7. accept I-8K & I-87K boards   |
| VGA Port             | 1 VGA port: resolution: 320x240x16 to 1024x768x16  |
| Two USB ports        | USB 1.1 Host ports for USB drive or USB mouse or USB Key-board   |
| Reset Button & Led   | 1 reset button & 1 power Led   |
| Unique Serial Number | 64-bit hardware unique serial number   |
| NET ID               | From 1 to 255, set by software   |
| Serial ports         |  |
| COM1                 | Internal use for I-87K IO boards of W-8347/8747. W-8047 has no COM1  |
| COM2                 | RS232: full modem signals, Speed: 115200 bps max.  |
| COM3                 | RS-485, Speed: 115200 bps max. D+, D-  |
| Two Ethernet ports   | 10/100M bps, NE2000 compatible, 10 BaseT, Program download port.<br>Please use NS-205 / NS-208 Industrial Ethernet Switch.   |
| Motion               | W-8347/8747 integrate with one I-8091(2-axes) or two I-8091(4-axes) can do motion control.   |
| PWM output           | 8 channels max. 250Hz max. for Off=2 & On=2 ms . Output square curve: Off: 2 to 32766 ms, On: 2 to 32766 ms. Optional D/O boards: i-8037, 8041, 8042, 8054, 8055, 8056, 8057, 8060, 8063, 8064, 8065, 8066, 8068, 8069 (Relay boards can not generate fast square curve) |
| Counters             |  |
| Parallel D/I counter | 8 ch. max. for 1 controller. Counter value: 32 bit. 250Hz max.<br>Min. ON & OFF width must > 2ms. Optional D/I boards: i-8040, 8042, 8051, 8052, 8053, 8054, 8055, 8058, 8063, 8077  |
| Serial D/I counter   | Counter input: 100Hz max. Counter value: 0 to 65535 (16 bit)<br>Optional serial I-87K D/I boards: i-87040, 87051, 87052, 87053, 87054, 87055, 87058, 87063   |
| Remote D/I counter   | All remote I-7000 & I-87K D/I modules support counters. 100Hz max. value: 0 to 65535   |
| High speed counter   | i-87082: 100kHz max. 32 bit, i-8080: 450kHz max. 32 bit  |

|   |   |
|---|---|
| <b>Protocols</b>  |   |
| Modbus serial protocol                                  | Up to 5 COM ports (COM2, 3, 5, 6,7 or 8) can support Modbus RTU slave protocol for connecting ISaGRAF, PC/HMI/OPC Server & HMI panels.  |
| Modbus TCP/IP   | Supports Modbus TCP/IP slave protocol for ISaGRAF & PC/HMI.   |
| Web HMI protocol  | Ethernet port for connecting PC running Internet Explorer   |
| I-7000 & I-87K Remote I/O                               | COM3 supports I-7000 I/O modules & (I-87K base + I-87K serial I/O boards) as remote I/O. Max. 255 I-7000/87K remote I/O modules for one controller  |
| M-7000 series Modbus I/O                                | Max. 10 R-485 ports – COM3 & (COM5 to COM14 if I-8142/8142i/8144 are found) can support M-7000 series Modbus I/O. Each port can connect up to 32 pcs. of M-7000 Modules.  |
| Modbus master protocol (multi-port)                     | Supports multi-ports of Modbus RTU / ASCII master protocol to connect to other Modbus slave devices. COM2,COM3,(or COM5 to COM14 if I-8112/8114/8142/8142i/8144 are found)  |
| Ebus  | to exchange data between ICP DAS's ISaGRAF Ethernet controllers via Ethernet port.  |
| Email with attached                                     | Supports sending email with one attached file via Ethernet communication. Max. attached size is 2M bytes. Each sending can send to max. 10 receivers.   |
| SMS: Short Message Service                              | One of COM2 (or COM5 if I-8112/8114 is found) can link to a GSM modem to support SMS. User can request data/control the controller by cellular phone. The controller can also send data & alarms to cellular phone. Optional GSM modems: M1206 or GM29 (GSM 900/1800 )  |
| User defined protocol                                   | User can write his own protocol applied at COM2, COM3, (& COM5 to COM14 in multi-serial port boards) by serial comm. function blocks.   |
| Modem_Link  | COM2 supports PC remotely download & monitor the controller through a normal modem.   |
| MMICON / LCD  | COM2 (or COM5 if I-8112/8114 is found) supports ICP DAS's MMICON. The MMICON is featured with a 240 x 64 dot LCD & a 4 x 4 Keyboard to display picture, string, integer, float, & input a char, string, integer & float.  |
| UDP Server & UDP Client (Exchange message & AutoReport) | LAN1 or LAN2 support UDP Server and UDP Client protocol to send / receive message to / from PC/HMI or other devices. For example, to automatically report data to InduSoft 's RXTX driver.  |
| TCP Client (Exchange message & Auto-report)             | LAN1 or LAN2 support TCP Client protocol to send / receive message to / from PC/HMI or other devices which support TCP server protocol. For example, to automatically report data to InduSoft 's RXTX driver, or to connect a location camera.  |
| CPU Redundancy Solution With Two Wincon controllers     | One is Master, one is slave. Master handles all inputs & outputs of the RS-485 I/O (I-7k & I-87K modules) at run time. If master is dead, Slave will take over the control of RS-485 I/O (or I-8KE4/8-MTCP I/O). All outputs should be configured as RS-485 I/O (or I-8KE4/8-MTCP I/O). Inputs can locate at slot 1 to 7 or configured as RS-485 I/O (or I-8KE4/8-MTCP I/O). Modbus RTU slave I/O or devices can also be used in the redundancy system. At least one I-7k or I-87K I/O module must be used in this system. Change Over Time: <= 500 ms, Synchronization: <= 75ms. |
| Battery Backup SRAM                                     | W-8347/8747 supports up to 4096 retain variables with a S256/S512 plug in the socket of the new back-plane( ver. 3-slot:2.6 , 7-slot:2.8 or later version). Optional: S256: 256kbytes, S512: 512kbytes<br>Note: W-8047 doesn't support S-256 / S-512  |
| File Access   | The Compact Flash card can be used for storing run time data & any controller setting with file operation (by f_xxx function block). The CF card size default is 128Mbytes, can be more.  |
| Modbus TCP/IP IO  | W-8047/8347/8747 support connecting to Modbus TCP/IP IO modules – I-8KE4-MTCP and I-8KE8-MTCP   |



## Specifications: W-8037 / 8337 / 8737

|                      |   |
|----------------------|---|
| Development software |   |
| ISaGRAF Version 3    | IEC61131-3 standard. Languages: LD, ST, FBD, SFC, IL & FC   |
| Max. code size       | accepts max. 1M bytes ISaGRAF code size (Appli.x8m must < 1M)   |
| Non-ISaGRAF          | Options: Microsoft EVC++4.0 or VS.NET 2003 (VB.NET, C#.NET)   |
| Web HMI              | PC running Internet Explorer can access to the Wincon-8037/8337/8737 via Local Ethernet or Internet or dial Modem, monitoring and Control.  |
| Security             | Three Level username and password protection  |
| Power supply         | 10 to 30VDC (unregulated), 20W (when I/O slots are empty )  |
| Protection           | Built-in power protection & network protection circuit  |
| General environment  |   |
| <b>temperature</b>   | Operating: -25°C to +75°C , Storage : -30°C to +85°C  |
| Humidity             | 5 to 95 % (non-condensed)   |
| System               |   |
| CPU                  | Intel Strong ARM CPU, 206MHz, or compatible   |
| Watchdog timer       | Yes   |
| Real time clock      | Gives hour, minute, sec, date of week, date of month, month & year  |
| SDRAM                | SDRAM:64M bytes ,   |
| FLASH Memory         | FLASH: 32M bytes for OS image   |
| Compact Flash Card   | One Compact Flash slot: CF memory card is 128M bytes or more  |
| EEPROM               | 16K bytes, retention > 100 years. 1,000,000 erase/write cycles  |
| I/O slots            | I/O slots: W-8037: 0 , W-8337: 3 , W-8737: 7, accept I-8K & I-87K boards  |
| VGA Port             | 1 VGA port: resolution: 320x240x16 to 1024x768x16   |
| PS/2 Port & USB port | 2 PS/2 ports: keyboard and mouse. 1 USB 1.1 Host port for USB drive or USB mouse  |
| Reset Button & Led   | 1 reset button & 1 power Led  |
| Unique Serial Number | 64-bit hardware unique serial number  |
| NET ID               | From 1 to 255, set by software  |
| Serial ports         |   |
| COM1                 | Internal use for I-87K boards of W-8337/8737. W-8037 has no COM1  |
| COM2                 | RS232: full modem signals, Speed: 115200 bps max.   |
| COM3                 | RS-485, Speed: 115200 bps max. D+, D-   |
| Ethernet             | 10M bps, NE2000 compatible, 10 BaseT, Program download port.  |
| Motion               | W-8337/8737 integrate with one I-8091(2-axes) or two I-8091(4-axes) can do motion control.  |
| PWM output           | 8 channels max. 250Hz max. for Off=2 & On=2 ms. Output square curve: Off: 2 to 32766 ms, On: 2 to 32766 ms. Optional D/O boards: i-8037, 8041, 8042, 8054, 8055, 8056, 8057, 8060, 8063, 8064, 8065, 8066, 8068, 8069 (Relay boards can not generate fast square curve) |
| Counters             |   |
| Parallel D/I counter | 8 ch. max. for 1 controller. Counter value: 32 bit. 250Hz max.<br>Min. ON & OFF width must > 2ms. Optional D/I boards: i-8040, 8042, 8051, 8052, 8053, 8054, 8055, 8058, 8063, 8077   |
| Serial D/I counter   | Counter input: 100Hz max. Counter value: 0 to 65535 (16 bit)<br>Optional serial I-87K D/I boards:<br>i-87040, 87051, 87052, 87053, 87054, 87055, 87058, 87063   |
| Remote D/I counter   | All remote I-7000 & I-87K D/I modules support counters. 100Hz max. value: 0 to 65535  |

|   |  |
|---|--|
| High speed counter                                      | i-87082: 100kHz max. 32 bit, i-8080: 450kHz max. 32 bit  |
| Protocols   |  |
| Modbus serial protocol                                  | Up to 5 COM ports (COM2, 3, 5, 6,7 or 8) can support Modbus RTU slave protocol for connecting ISaGRAF, PC/HMI/OPC Server & HMI panels.   |
| Modbus TCP/IP   | Supports Modbus TCP/IP slave protocol for ISaGRAF & PC/HMI.  |
| Web HMI protocol  | Ethernet port for connecting PC running Internet Explorer  |
| I-7000 & I-87K Remote I/O                               | COM3 supports I-7000 I/O modules & (I-87K base + I-87K serial I/O boards) as remote I/O. Max. 255 I-7000/87K remote I/O modules for one controller   |
| M-7000 series Modbus I/O                                | Max. 10 R-485 ports – COM3 & (COM5 to COM14 if I-8142/8142i/8144 are found) can support M-7000 series Modbus I/O. Each port can connect up to 32 pcs. of M-7000 Modules.   |
| Modbus master protocol (multi-port)                     | Supports multi-ports of Modbus RTU / ASCII master protocol to connect to other Modbus slave devices. COM2,COM3,(or COM5 to COM14 if I-8112/8114/8142/8142i/8144 are found)   |
| Ebus  | to exchange data between ICP DAS's ISaGRAF Ethernet controllers via Ethernet port.   |
| Email with attached                                     | Supports sending email with one attached file via Ethernet communication. Max. attached size is 2M bytes. Each sending can send to max. 10 receivers.  |
| SMS: Short Message Service                              | One of COM2 (or COM5 if I-8112/8114 is found) can link to a GSM modem to support SMS. User can request data/control the controller by cellular phone. The controller can also send data & alarms to cellular phone. Optional GSM modems: M1206 or GM29 (GSM 900/1800 ) |
| User defined protocol                                   | User can write his own protocol applied at COM2, COM3, (& COM5 to COM14 if multi-serial port boards are plugged) by serial communication function blocks.  |
| Modem_Link  | COM2 supports PC remotely download & monitor the controller through a normal modem.  |
| MMICON / LCD  | COM2 (or COM5 if I-8112/8114 is found) supports ICP DAS's MMICON. The MMICON is featured with a 240 x 64 dot LCD & a 4 x 4 Keyboard to display picture, string, integer, float, & input a char, string, integer & float.   |
| UDP Server & UDP Client (Exchange message & AutoReport) | LAN1 or LAN2 support UDP Server and UDP Client protocol to send / receive message to / from PC/HMI or other devices. For example, to automatically report data to InduSoft 's RXTX driver.   |
| TCP Client (Exchange message & Auto-report)             | LAN1 or LAN2 support TCP Client protocol to send / receive message to / from PC/HMI or other devices which support TCP server protocol. For example, to automatically report data to InduSoft 's RXTX driver, or to connect a location camera.                         |
| Battery Backup SRAM                                     |  |
|   | W-8337/8737 supports up to 4096 retain variables with a S256/S512 plug in the socket of the new back-plane( ver. 3-slot:2.6 , 7-slot:2.8 or later version). Optional: S256: 256kbytes, S512: 512kbytes<br>Note: W-8037 doesn't support S-256 / S-512                   |
| File Access   | The Compact Flash card can be used for storing run time data & any controller setting with file operation (by f_xxx function block). The CF card size default is 128Mbytes, can be more.   |

## Specifications: I-8437-80 / I-8837-80

|                               |   |
|-------------------------------|---|
| <b>Power supply</b>           |   |
| Power requirements            | 10 to 30VDC (unregulated), 20W (when I/O slots are empty )  |
| Protection                    | Built-in power protection & network protection circuit  |
| <b>General environment</b>    |   |
| <b>temperature</b>            | Operating: -25°C to +75°C , Storage : -30°C to +85°C  |
| Humidity                      | 5 to 95 % (non-condensed)   |
| <b>System</b>                 |   |
| <b>CPU</b>                    | 80188, or compatible, I-8437-80 / I-8837-80: 80M Hz<br>I-8437/8837: 40M Hz  |
| Watchdog timer                | Yes   |
| Real time clock               | Gives hour, minute, sec, date of week, date of month, month & year (1980 to 2079)   |
| SRAM                          | 512Kbytes   |
| FLASH Memory                  | 512Kbytes, Erase unit is 64K bytes, 100,000 erase/write cycles  |
| NVSRAM                        | 31 bytes, battery backup, data valid up to 10 years   |
| EEPROM                        | 2048 bytes, retention > 100 years. 1,000,000 erase/write cycles   |
| SMMI                          | Five 7-Seg. Led, four push buttons & three Led on the front panel. It can display message, value, input value, simulate input & output.   |
| I/O slots                     | 4 empty slots for I-8437, 8 empty slots for I-8837. Accept parallel & serial I/O boards   |
| NET ID                        | 8 dip switch to set NET ID as 1 to 255  |
| <b>Serial ports</b>           |   |
| COM1                          | RS232: TXD,RXD,GND, Speed: 115200 bps max. Program download port.   |
| Ethernet                      | 10M bps, NE2000 compatible, 10 BaseT, Program download port.  |
| <b>COM3</b>                   | Can be configed as RS232 or S485, Speed: 115200 bps max.<br>RS232: TXD,RXD,RTS,CTS,GND, RS485: Data+, Data-   |
| <b>COM4</b>                   | RS232: Full modem signals, 115200 bps max.<br>TXD,RXD,RTS,CTS,DSR,DTR,CD,RI,GND.  |
| <b>Development software</b>   |   |
| ISaGRAF Version 3             | IEC61131-3 standard. Languages: LD, ST, FBD, SFC, IL & FC   |
| Max. code size                | I-8437/8837 accepts max. 64K byte ISaGRAF code size (Appli.x8m must < 64K)  |
| <b>Motion control</b>         |   |
|                               | I-8437/8837 can integrate with one I-8091(2-axes) or two I-8091(4-axes) to do motion control. When doing motion control, I-8437 / 8837's Ethernet communication is not available.   |
| <b>PWM output</b>             |   |
| Pulse Width Modulation output | 8 channels max. for one controller. 500Hz max. for Off=1 & On=1 ms<br>Output square curve: Off: 1 to 32767 ms, On: 1 to 32767 ms<br>Optional D/O boards: i-8037, 8041, 8042, 8054, 8055, 8056, 8057, 8060, 8063, 8064, 8065, 8066,8068, 8069 (Relay Output boards can not generate fast square curve) |
| <b>Counters</b>               |   |
| Parallel D/I counter          | 8 ch. max. for 1 controller. Counter value: 32 bit. 500Hz max.<br>Min. ON & OFF width must > 1ms. Optional D/I boards: i-8040, 8042, 8051,  |

|                            |  |
|----------------------------|--|
|                            | 8052, 8053, 8054, 8055, 8058, 8063, 8077   |
| Serial D/I counter         | Counter input: 100Hz max. Counter value: 0 to 65535 (16 bit)<br>Optional serial I-87K D/I boards: i-87051, 87052, 87053, 87054, 87055, 87058, 87063  |
| Remote D/I counter         | All remote I-7000 & I-87K D/I modules support counters. 100Hz max. value: 0 to 65535   |
| High speed counter         | i-87082: 100kHz max. 32 bit, i-8080: 450kHz max. 32 bit  |
| Protocols                  |  |
| Modbus serial protocol     | COM1 default supports Modbus RTU slave protocol for connecting ISaGRAF, PC/HMI/OPC Server & MMI panels. (COM3 can also be activated to support Modbus RTU slave protocol)  |
| Modbus TCP/IP protocol     | Ethernet port support Modbus TCP/IP slave protocol for connecting ISaGRAF & PC/HMI.  |
| Remote I/O                 | One of COM3 or COM4 supports I-7000 I/O modules & (I-87K base + I-87K serial I/O boards) as remote I/O. Max. 64 remote I/O module for one controller   |
| Modbus master protocol     | Two of COM1 or COM3 or COM4 ( or COM5 if multi-serial port boards are plugged) can support Modbus RTU / ASCII master protocol to connect to other Modbus slave devices   |
| Fbus                       | built in COM3 port to exchange data between ICP DAS's ISaGRAF controllers.   |
| Ebus                       | to exchange data between ICP DAS's ISaGRAF Ethernet controllers via Ethernet port.   |
| SMS: Short Message Service | One of COM4 or COM5 can link to a GSM modem to support SMS. User can request data/control the controller by cellular phone. The controller can also send data & alarms to user's cellular phone. Optional GSM modems: M1206 or GM29 (GSM 900/1800 )  |
| User defined protocol      | User can write his own protocol applied at COM1, COM3, COM4 (& COM5 to COM20 if multi-serial port boards are plugged) by serial communication function blocks.   |
| Modem_Link                 | Supports PC remotely download & monitor the controller through a normal modem.   |
| MMICON / LCD               | One of COM3 or COM4 supports ICP DAS's MMICON. The MMICON is featured with a 240 x 64 dot LCD and a 4 x 4 Keyboard. User can use it to display picture, string, integer, float, and input a character, string, integer and float.  |
| Battery backup SRAM        |  |
|                            | I-8437/8837 can support up to 1024 retain variables with a S256/S512 plug in the socket of the back-plane. Data can also be stored in the S256/S512, and then PC can load these data via COM1 or ethernet. PC can also download pre-defined data to the S256/S512.<br>Optional: S256: 256kbytes, S512: 512kbytes |

## Specifications: I-8417 / 8817

|                               |   |
|-------------------------------|---|
| <b>Power supply</b>           |   |
| Power requirements            | 10 to 30VDC (unregulated), 20W (when I/O slots are empty )  |
| Protection                    | Built-in power protection & network protection circuit  |
| <b>General environment</b>    |   |
| <b>temperature</b>            | Operating: -25°C to +75°C , Storage : -30°C to +85°C  |
| Humidity                      | 5 to 95 % (non-condensed)   |
| <b>System</b>                 |   |
| <b>CPU</b>                    | 80188,40MHz, or compatible  |
| Watchdog timer                | Yes   |
| Real time clock               | Gives hour, minute, sec, date of week, date of month, month & year (1980 to 2079)   |
| SRAM                          | 512Kbytes   |
| FLASH Memory                  | 512Kbytes, Erase unit is 64K bytes, 100,000 erase/write cycles  |
| NVSRAM                        | 31 bytes, battery backup, data valid up to 10 years   |
| EEPROM                        | 2048 bytes, retention > 100 years. 1,000,000 erase/write cycles   |
| SMMI                          | Five 7-Seg. Led, four push buttons & three Led on the front panel. It can display message, value, input value, simulate input & output.   |
| I/O slots                     | 4 empty slots for I-8417, 8 empty slots for I-8817. Accept parallel & serial I/O boards   |
| NET ID                        | 8 dip switch to set NET ID as 1 to 255  |
| <b>Serial ports</b>           |   |
| COM1                          | RS232: TXD,RXD,GND, Speed: 115200 bps max. Program download port.   |
| <b>COM2</b>                   | RS485: D+, D-, 115200 bps max. Self-tuner ASIC inside, Program download port.   |
| <b>COM3</b>                   | Can be configed as RS232 or S485, Speed: 115200 bps max.<br>RS232: TXD,RXD,RTS,CTS,GND, RS485: Data+, Data-   |
| <b>COM4</b>                   | RS232: Full modem signals, 115200 bps max.<br>TXD,RXD,RTS,CTS,DSR,DTR,CD,RI,GND.  |
| <b>Development software</b>   |   |
| ISaGRAF Version 3             | IEC61131-3 standard. Languages: LD, ST, FBD, SFC, IL & FC   |
| Max. code size                | I-8417/8817 accepts max. 64K byte ISaGRAF code size (Appli.x8m must < 64K)  |
| <b>Motion control</b>         |   |
|                               | I-8417/8817/8437/8837 can integrate with one I-8091(2-axes) or two I-8091(4-axes) to do motion control. When doing motion control, I-8437 / 8837's Ethernet communication is not available.   |
| <b>PWM output</b>             |   |
| Pulse Width Modulation output | 8 channels max. for one controller. 500Hz max. for Off=1 & On=1 ms<br>Output square curve: Off: 1 to 32767 ms, On: 1 to 32767 ms<br>Optional D/O boards: i-8037, 8041, 8042, 8054, 8055, 8056, 8057, 8060, 8063, 8064, 8065, 8066,8068, 8069 (Relay Output boards can not generate fast square curve) |
| <b>Counters</b>               |   |
| Parallel D/I counter          | 8 ch. max. for 1 controller. Counter value: 32 bit. 500Hz max.<br>Min. ON & OFF width must > 1ms  |

|                            |  |
|----------------------------|--|
|                            | Optional D/I boards: i-8040, 8042, 8051, 8052, 8053, 8054, 8055, 8058, 8063, 8077  |
| Serial D/I counter         | Counter input: 100Hz max. Counter value: 0 to 65535 (16 bit)<br>Optional serial I-87K D/I boards: i-87051, 87052, 87053, 87054, 87055, 87058, 87063  |
| Remote D/I counter         | All remote I-7000 & I-87K D/I modules support counters. 100Hz max. value: 0 to 65535   |
| High speed counter         | i-87082: 100kHz max. 32 bit, i-8080: 450kHz max. 32 bit  |
| Protocols                  |  |
| Modbus serial protocol     | COM1 & COM2 default supports Modbus RTU slave protocol for connecting ISaGRAF, PC/HMI/OPC Server & MMI panels.   |
| Remote I/O                 | One of COM3 or COM4 supports I-7000 I/O modules & (I-87K base + I-87K serial I/O boards) as remote I/O. Max. 64 remote I/O module for one controller   |
| Modbus master protocol     | Two of COM1 or COM3 or COM4 ( or COM5 if multi-serial port boards are plugged) can support Modbus RTU / ASCII master protocol to connect to other Modbus slave devices   |
| Fbus                       | built in COM3 port to exchange data between ICP DAS's ISaGRAF controllers.   |
| SMS: Short Message Service | One of COM4 or COM5 can link to a GSM modem to support SMS. User can request data/control the controller by cellular phone. The controller can also send data & alarms to user's cellular phone. Optional GSM modems: M1206 or GM29 (GSM 900/1800 )  |
| User defined protocol      | User can write his own protocol applied at COM1, COM3, COM4 (& COM5 to COM20 if multi-serial port boards are plugged) by serial communication function blocks.   |
| Modem_Link                 | Supports PC remotely download & monitor the controller through a normal modem.   |
| MMICON / LCD               | One of COM3 or COM4 supports ICP DAS's MMICON. The MMICON is featured with a 240 x 64 dot LCD and a 4 x 4 Keyboard. User can use it to display picture, string, integer, float, and input a character, string, integer and float.  |
| Battery backup SRAM        |  |
|                            | I-8417/8817 can support up to 1024 retain variables with a S256/S512 plug in the socket of the back-plane. Data can also be stored in the S256/S512, and then PC can load these data via COM1 or COM2. PC can also download pre-defined data to the S256/S512.<br>Optional: S256: 256kbytes, S512: 512kbytes |

# Specifications: uPAC-7186EG

|                               |   |
|-------------------------------|---|
| <b>Power supply</b>           |   |
| Power requirements            | 10 to 30VDC, $\mu$ PAC-7186EG:2W , $\mu$ PAC-7186EGD:3W (unregulated)   |
| Protection                    | Built-in power protection & network protection circuit  |
| <b>General environment</b>    |   |
| temperature                   | Operating: -25°C to +75°C , Storage : -30°C to +85°C  |
| Humidity                      | 5 to 95 % (non-condensed)   |
| ESD Protection                | 4KV Contact Discharge for RS-232, RS-485, Power and Ethernet  |
| <b>System</b>                 |   |
| CPU                           | 80186 80MHz, or compatible  |
| Watchdog timer                | Yes, default 0.8 second   |
| Real time clock               | Gives hour, minute, sec, date of week, date of month, month & year  |
| SRAM                          | 640K bytes  |
| FLASH Memory                  | 512Kbytes, Erase unit is 64K bytes, 100,000 erase/write cycles  |
| NVRAM                         | 31 bytes, battery backup, data valid up to 10 years   |
| EEPROM                        | 16K bytes, retention > 100 years. 1,000,000 erase/write cycles  |
| Display for $\mu$ PAC-7186EGD | Five 7-Seg. Led on the front. It can display message & value.   |
| Expansion I/O bus             | One optional Xxxx series I/O board can be plugged inside the $\mu$ PAC-7186EG / 7186EGD.  |
| NET ID                        | Set by software, 1 to 255   |
| <b>Serial ports</b>           |   |
| COM1                          | RS232: TXD, RXD, RTS, CTS, GND,<br>Speed: 115200 bps max. Program downloads port.   |
| COM2                          | RS485: D+, D-<br>Pull high/low resistor jumper.   |
| Ethernet                      | <b>10/100 Mbps</b> , NE2000 compatible<br>Auto-negotiating; Auto-MDIX; LED indicator  |
| <b>Development software</b>   |   |
| ISaGRAF Version 3             | IEC61131-3 standard. Languages: LD, ST, FBD, SFC, IL & FC   |
| Max. code size                | Accepts max. 64K byte ISaGRAF code size<br>(Appli.x8m must < 64K)   |
| <b>PWM output</b>             |   |
| Pulse Width Modulation output | All Xxxx series D/O boards support PWM output.<br>8 channels max. for one controller.<br>500Hz max. for Off=1 & On=1 ms<br>Output square curve: Off: 1 to 32767 ms, On: 1 to 32767 ms |
| <b>Counters</b>               |   |
| Parallel D/I counter          | All X-xxx series D/I boards support D/I counter.<br>8 ch. max. for one controller. Counter value: 32 bit<br>500Hz max. Min. ON & OFF width must > 1ms                                 |
| Remote D/I counter            | All remote i-7000 & i-87K D/I modules support counters. 100Hz max.<br>value: 0 to 65535   |
| Remote high speed counter     | Optional i-87082:100kHz max. , 32 bit   |
| <b>Protocols</b>              |   |

|                            |   |
|----------------------------|---|
| Modbus Slave protocol      | Up to 2 COM ports (COM1 and COM3-in-X-board) can support Modbus RTU slave protocol for connecting ISaGRAF, PC/HMI/OPC Server & MMI panels.  |
| Modbus master protocol     | Up to 2 COM ports (COM1, COM2 and COM3-in-X-board), support Modbus RTU Master or ASCII master protocol to connect to other Modbus slave I/O devices. 2 ports support up to 128 Mbus_xxx function blocks (same type).  |
| Modbus TCP/IP protocol     | Ethernet port supports Modbus TCP/IP slave protocol for connecting ISaGRAF & PC/HMI. Max. 4 connections   |
| Remote I/O                 | One of COM2 (or COM3:RS485 if found) supports i-7000 I/O modules & (i-87K base + i-87K serial I/O boards) as remote I/O. Max. 64 I/O modules for one controller   |
| Fbus                       | Built in COM2 port to exchange data between ICP DAS's ISaGRAF controllers.  |
| Ebus                       | To exchange data between ICP DAS's ISaGRAF Ethernet controllers via Ethernet port.  |
| Email with attached        | Send email to max.10 receivers each time via internet. If using X-607/608, it can send email with an attached file and its file size max. is about X-607:112K, X-608:488K bytes.  |
| SMS: Short Message Service | COM1 or one of (COM3:RS232 or COM4:RS232 if found) can link to a GSM modem to support SMS. User can request data/control the controller by cellular phone. The controller can also send data & alarms to user's cellular phone.<br>Optional GSM modems: M1206 or GM29 (GSM 900/1800 ) |
| User defined protocol      | User can write his own protocol applied at COM1, COM2 & (COM3 to COM8 if found) by serial communication function blocks.  |
| Modem Link                 | Supports PC remotely download & monitor the controller through COM4 of X504.  |
| MMICON / LCD               | One of (COM3:RS232 if found) supports ICP DAS's MMICON. The MMICON is featured with a 240 x 64 dot LCD and a 4 x 4 Keyboard. User can use it to display picture, string, integer, float, and input a character, string, integer and float.  |
| <b>Retained Variables</b>  |   |
|                            | With the optional X-607/608: support up to 1024 retained variables  |
| <b>Battery backup SRAM</b> |   |
|                            | With the optional X-607/608 board plugged inside the $\mu$ PAC-7186EG, data can be stored in it, and then PC can load these data via COM1 or Ethernet.<br>PC can also download pre-defined data to the X607/X608. Optional: X607: 128kbytes, X608: 512kbytes                          |



## Specifications: I-7188EG

|                               |  |
|-------------------------------|--|
| <b>Power supply</b>           |  |
| Power requirements            | 10 to 30VDC (unregulated), I-7188EG:2W , I-7188EGD:3W  |
| Protection                    | Built-in power protection & network protection circuit   |
| <b>General environment</b>    |  |
| <b>temperature</b>            | Operating: -25°C to +75°C , Storage : -30°C to +85°C   |
| Humidity                      | 5 to 95 % (non-condensed)  |
| <b>System</b>                 |  |
| <b>CPU</b>                    | 80188 40MHz, or compatible   |
| Watchdog timer                | Yes  |
| Real time clock               | Gives hour, minute, sec, date of week, date of month, month & year (1980 to 2079)  |
| SRAM                          | 512Kbytes  |
| FLASH Memory                  | 512Kbytes, Erase unit is 64K bytes, 100,000 erase/write cycles   |
| NVSRAM                        | 31 bytes, battery backup, data valid up to 10 years  |
| EEPROM                        | 2048 bytes, retention > 100 years. 1,000,000 erase/write cycles  |
| Display for I-7188EGD         | Five 7-Seg. Led on the front. It can display message & value.  |
| Expansion I/O bus             | One optional Xxxx series I/O board can be plugged inside the I-7188EG / 7188EGD.   |
| NET ID                        | Set by software, 1 to 255  |
| <b>Serial ports</b>           |  |
| COM1                          | RS232: TXD,RXD,GND, Speed: 115200 bps max. Program download port.  |
| <b>COM2</b>                   | RS485: D+, D-, 115200 bps max. Self-tuner ASIC inside, Program download port.  |
| Ethernet                      | 10M bps, NE2000 compatible, 10 BaseT, Program download port.   |
| <b>Development software</b>   |  |
| ISaGRAF Version 3             | IEC61131-3 standard. Languages: LD, ST, FBD, SFC, IL & FC  |
| Max. code size                | I-7188EG/7188EGD accepts max. 64K byte ISaGRAF code size (Appli.x8m must < 64K)  |
| <b>PWM output</b>             |  |
| Pulse Width Modulation output | All Xxxx series D/O boards support PWM output. 8 channels max. for one controller.<br>500Hz max. for Off=1 & On=1 ms<br>Output square curve: Off: 1 to 32767 ms, On: 1 to 32767 ms |
| <b>Counters</b>               |  |
| Parallel D/I counter          | All Xxxx series D/I boards support D/I counter.<br>8 ch. max. for one controller. Counter value: 32 bit<br>500Hz max. Min. ON & OFF width must > 1ms                               |
| Remote D/I counter            | All remote I-7000 & I-87K D/I modules support counters. 100Hz max. value: 0 to 65535   |
| Remote high speed counter     | Optional i-87082:100kHz max. , 32 bit  |
| <b>Protocols</b>              |  |
| Modbus serial protocol        | COM1 default supports Modbus RTU slave protocol for connecting ISaGRAF, PC/HMI/OPC Server & MMI panels. (COM3 in X5xx board can be activated to support Modbus RTU slave)          |

|                            |  |
|----------------------------|--|
| Modbus TCP/IP protocol     | Ethernet port supports Modbus TCP/IP slave protocol for connecting ISaGRAF & PC/HMI.   |
| Remote I/O                 | One of COM2 (or COM3:RS485 if found) supports I-7000 I/O modules & (I-87K base + I-87K serial I/O boards) as remote I/O. Max. 64 I/O modules for one controller  |
| Modbus master protocol     | Two of COM1 or COM2 (or COM3 if found) can support Modbus RTU / ASCII master protocol to connect to other Modbus slave I/O devices   |
| Fbus                       | built in COM2 port to exchange data between ICP DAS's ISaGRAF controllers.   |
| Ebus                       | to exchange data between ICP DAS's ISaGRAF Ethernet controllers via Ethernet port.   |
| SMS: Short Message Service | One of (COM3:RS232 or COM4:RS232 if found) can link to a GSM modem to support SMS. User can request data/control the controller by cellular phone. The controller can also send data & alarms to user's cellular phone.<br>Optional GSM modems: M1206 or GM29 (GSM 900/1800 )  |
| User defined protocol      | User can write his own protocol applied at COM1, COM2 & (COM3 to COM8 if found) by serial communication function blocks.   |
| MMICON / LCD               | One of (COM3:RS232 if found) supports ICP DAS's MMICON. The MMICON is featured with a 240 x 64 dot LCD and a 4 x 4 Keyboard. User can use it to display picture, string, integer, float, and input a character, string, integer and float.   |
| Battery backup SRAM        | I-7188EG / 7188EGD can support up to 1024 retain variables with a X607 / X608 plug in the only expansion I/O slot.<br>Data can be stored in X607/X608, and then PC can load these data via COM1 or ethernet.<br>PC can also download pre-defined data to the X607/X608<br>Optional: X607: 128kbytes, X608: 512kbytes |

## Specifications: I-7188XG

|                               |  |
|-------------------------------|--|
| <b>Power supply</b>           |  |
| Power requirements            | 10 to 30VDC (unregulated), I-7188XG:2W , I-7188XGD:3W  |
| Protection                    | Built-in power protection & network protection circuit   |
| <b>General environment</b>    |  |
| <b>temperature</b>            | Operating: -25°C to +75°C , Storage : -30°C to +85°C   |
| Humidity                      | 5 to 95 % (non-condensed)  |
| <b>System</b>                 |  |
| <b>CPU</b>                    | 80188 40MHz, or compatible   |
| Watchdog timer                | Yes  |
| Real time clock               | Gives hour, minute, sec, date of week, date of month, month & year (1980 to 2079)  |
| SRAM                          | 512Kbytes  |
| FLASH Memory                  | 512Kbytes, Erase unit is 64K bytes, 100,000 erase/write cycles   |
| NVSRAM                        | 31 bytes, battery backup, data valid up to 10 years  |
| EEPROM                        | 2048 bytes, retention > 100 years. 1,000,000 erase/write cycles  |
| Display for I-7188XGD         | Five 7-Seg. Led on the front. It can display message & value.  |
| Expansion I/O bus             | One optional Xxxx series I/O board can be plugged inside the I-7188XG / 7188XGD.   |
| NET ID                        | Set by software, 1 to 255  |
| <b>Serial ports</b>           |  |
| COM1                          | Can be used as RS232 or RS485 , Speed: 115200 bps max.<br>RS232 TXD,RXD,RTS,CTS,GND<br>RS485: D+, D-, self-tuner inside<br>Program download port.                                  |
| <b>COM2</b>                   | RS485: D+, D-, 115200 bps max. Self-tuner ASIC inside, Program download port.  |
| <b>Development software</b>   |  |
| ISaGRAF Version 3             | IEC61131-3 standard. Languages: LD, ST, FBD, SFC, IL & FC  |
| Max. code size                | I-7188XG/7188XGD accepts max. 64K byte ISaGRAF code size (Appli.x8m must < 64K)  |
| <b>PWM output</b>             |  |
| Pulse Width Modulation output | All Xxxx series D/O boards support PWM output. 8 channels max. for one controller.<br>500Hz max. for Off=1 & On=1 ms<br>Output square curve: Off: 1 to 32767 ms, On: 1 to 32767 ms |
| <b>Counters</b>               |  |
| Parallel D/I counter          | All Xxxx series D/I boards support D/I counter.<br>8 ch. max. for one controller. Counter value: 32 bit<br>500Hz max. Min. ON & OFF width must > 1ms                               |
| Remote D/I counter            | All remote I-7000 & I-87K D/I modules support counters. 100Hz max. value: 0 to 65535   |
| Remote high speed counter     | Optional i-87082:100kHz max. , 32 bit  |
| <b>Protocols</b>              |  |
| Modbus serial protocol        | COM1 default supports Modbus RTU slave protocol for connecting ISaGRAF,  |

|                            |   |
|----------------------------|---|
|                            | PC/HMI/OPC Server & MMI panels. (COM3 in X5xx board can be activated to support Modbus RTU slave protocol)  |
| Remote I/O                 | One of COM2 (or COM3:RS485 if found) supports I-7000 I/O modules & (I-87K base + I-87K serial I/O boards) as remote I/O. Max. 64 I/O modules for one controller   |
| Modbus master protocol     | Two of COM2 (or COM3 if found) can support Modbus RTU / ASCII master protocol to connect to other Modbus slave I/O devices  |
| Fbus                       | built in COM2 port to exchange data between ICP DAS's ISaGRAF controllers.  |
| SMS: Short Message Service | One of (COM3:RS232 or COM4:RS232 if found) can link to a GSM modem to support SMS. User can request data/control the controller by cellular phone. The controller can also send data & alarms to user's cellular phone.<br>Optional GSM modems: M1206 or GM29 (GSM 900/1800 )                         |
| User defined protocol      | User can write his own protocol applied at COM2 & (COM3 to COM8 if found) by serial communication function blocks.  |
| MMICON / LCD               | One of (COM3:RS232 if found) supports ICP DAS's MMICON. The MMICON is featured with a 240 x 64 dot LCD and a 4 x 4 Keyboard. User can use it to display picture, string, integer, float, and input a character, string, integer and float.  |
| Battery backup SRAM        | I-7188XG / 7188XGD can support up to 1024 retain variables with a X607 / X608 plug in the only expansion I/O slot.<br>Data can be stored in X607/X608, and then PC can load these data via COM1. PC can also download pre-defined data to the X607/X608<br>Optional: X607: 128kbytes, X608: 512kbytes |

# Selection Guide

|                           |   |
|---------------------------|---|
| <b>Power supply</b>       |   |
| ACE-540A                  | 24V/1.7A power supply(panel Mount)  |
| DIN-540A                  | 24V/1.7A power supply(DIN-Rail mount)   |
| KA-52F                    | 24V/1A power supply(no mounting)  |
| DIN-KA52F                 | 24V/1A power supply(DIN-Rail mountong)  |
| KWM020-1824F              | 24V/0.75A power supply (No-mounting)  |
| DP-1200                   | 24V/5A power supply   |
| DP-640                    | 24V/1.7A Power supply   |
| DP-660                    | 24V/2.5A 5V/0.5A Power supply   |
| DP-665                    | 24V/2.5A 5V/0.5A Power supply   |
| <b>Development tools</b>  |   |
| ISaGRAF-256-E             | ISaGRAF Ver.3, up to 256 I/O tags + one English manual  |
| ISaGRAF-256-C             | ISaGRAF Ver.3, up to 256 I/O tags + one Chinese manual  |
| ISaGRAF-32-E              | ISaGRAF Ver.3, up to 32 I/O tags + one English manual(Cannot upgrade anymore)   |
| ISaGRAF-32-C              | ISaGRAF Ver.3, up to 32 I/O tags + one Chinese manual(Cannot upgrade anymore)   |
| ISaGRAF Book-E            | User's manual of ISaGRAF controllers (English)  |
| ISaGRAF Book-C            | User's manual of ISaGRAF controllers (Chinese, traditional)   |
| <b>ISaGRAF controller</b> |   |
| I-8417                    | ISaGRAF I-8000 controller, 4 empty slots (Support battery-backup retain variables & memory if a S-256 / S-512 is plug in the back-plane)                |
| I-8817                    | ISaGRAF I-8000 controller, 8 empty slots (Support battery-backup retain variables & memory if a S-256 / S-512 is plug in the back-plane)                |
| I-8417-G                  | I-8417 gray color version   |
| I-8817-G                  | I-8817 gray color version   |
| I-8437                    | ISaGRAF I-8000 ethernet controller, 4 empty slots (Support battery-backup retain variables & memory if a S-256 / S-512 is plug in the back-plane)       |
| I-8837                    | ISaGRAF I-8000 ethernet controller, 8 empty slots (Support battery-backup retain variables & memory if a S-256 / S-512 is plug in the back-plane)       |
| I-8437-G                  | I-8437 gray color version   |
| I-8837-G                  | I-8837 gray color version   |
| I-8437-80                 | I-8437 with faster CPU (80M Hz)   |
| I-8837-80                 | I-8837 with faster CPU (80M Hz)   |
| I-8437-80-G               | I-8437-80 gray color version  |
| I-8837-80-G               | I-8837-80 gray color version  |
| I-7188XG                  | IsaGRAF I-7188 controller (Support battery-backup retain variables & memory if a X-607 / X-608 is plug in the only I/O expansion slot)                  |
| I-7188XGD                 | 7188XG controller with display  |
| I-7188EG                  | IsaGRAF I-7188 ethernet controller (Support battery-backup retain variables & memory if a X-607 / X-608 is plug in the only I/O expansion slot)         |
| I-7188EGD                 | I-7188EG with display   |
| uPAC-7186EG               | IsaGRAF I-7186 ethernet controller (Support battery-backup retain variables & memory if a X-607 / X-608 is plug in the only I/O expansion slot) (80MHz) |
| uPAC-7186EGD              | uPAC-7186EG with display  |
| W-8037                    | IsaGRAF Wincon-8000 controller, No I/O slot<br>(No support battery-backup retain variables & memory : S-256 / S-512)                                    |
| W-8337                    | IsaGRAF Wincon-8000 controller, 3 empty slots (Support battery-backup retain variables & memory if a S-256 / S-512 is plug in the back-plane)           |

|   |  |
|---|--|
| W-8737  | ISaGRAF Wincon-8000 controller, 7 empty slots (Support battery-backup retain variables & memory if a S-256 / S-512 is plug in the back-plane)                                    |
| W-8037-G  | W-8037 gray color version  |
| W-8337-G  | W-8337 gray color version  |
| W-8737-G  | W-8737 gray color version  |
| W-8036  | Indusoft + ISaGRAF Wincon-8000 controller, No I/O slot<br>(No support battery-backup retain variables & memory : S-256 / S-512)  |
| W-8336  | Indusoft + ISaGRAF Wincon-8000 controller, 3 empty slots (Support battery-backup retain variables & memory if a S-256 / S-512 is plug in the back-plane)                         |
| W-8736  | Indusoft + ISaGRAF Wincon-8000 controller, 7 empty slots (Support battery-backup retain variables & memory if a S-256 / S-512 is plug in the back-plane)                         |
| W-8036-G  | W-8036 gray color version  |
| W-8336-G  | W-8336 gray color version  |
| W-8736-G  | W-8736 gray color version  |
| <b>Recommand to use Industrial Ethernet Switch: NS-205 / NS-208 for W-8347/8747/8346/8746 and their gray color version.</b> |  |
| W-8347  | ISaGRAF Wincon-8000 controller with Dual 10/100M Ethernet ports, 3 empty slots (Support battery-backup retain variables if a S-256 / S-512 is plug in the back-plane)            |
| W-8747  | ISaGRAF Wincon-8000 controller with Dual 10/100M Ethernet ports, 7 empty slots (Support battery-backup retain variables if a S-256 / S-512 is plug in the back-plane)            |
| W-8347-G  | W-8347 gray color version  |
| W-8747-G  | W-8747 gray color version  |
| W-8346  | Indusoft + ISaGRAF Wincon-8000 controller with Dual 10/100M Ethernet ports, 3 empty slots (Support battery-backup retain variables if a S-256 / S-512 is plug in the back-plane) |
| W-8746  | Indusoft + ISaGRAF Wincon-8000 controller with Dual 10/100M Ethernet ports, 7 empty slots (Support battery-backup retain variables if a S-256 / S-512 is plug in the back-plane) |
| W-8346-G  | W-8346 gray color version  |
| W-8746-G  | W-8746 gray color version  |
|   |  |
| <b>Battery backup SRAM</b>  |  |
| S256  | 256Kbytes battery backup SRAM for I-8417 /8817/8437/8837, i-8437-80/8837-80 and W-8337/8737/8347/8747 (W-8037/8047 doesn't support S-256/512)                                    |
| S512  | 512Kbytes battery backup SRAM for I-8417 /8817/8437/8837, i-8437-80/8837-80 and W-8337/8737/8347/8747 (W-8037/8047 doesn't support S-256/512)                                    |
| X607  | 128Kbytes battery backup SRAM for<br>I-7188XG/7188EG , I-7188XGD/7188EGD & uPAC-7186EG / 7186EGD   |
| X608  | 512Kbytes battery backup SRAM for<br>I-7188XG/7188EG , I-7188XGD/7188EGD & uPAC-7186EG / 7186EGD   |
|   |  |
| <b>MMICON</b>   |  |
| MMICON / LCD  | Man-machine interface : 240 x 64 dots LCD Display  |
| MMICON / Starter Kit  | Man-machine interface : 240 x 64 dots LCD Display + 4x4 Keyboard   |
|   |  |
| <b>GSM modem</b>  |  |
| M1206   | 900/1800 GSM/GPRS External Modem   |
| GM29  | 900/1800 GSM/GPRS External Modem   |
|   |  |

|                                     |   |
|-------------------------------------|---|
| <b>I-87K expansion base</b>         | Accepts I-87K serial I/O boards   |
| RU-87P4-G                           | Remote I-87K I/O base, 4 empty slots, RS-485 signal, gray color<br>If using with high profile i-87K I/O cards, it supports hot swap and auto-config |
| RU-87P8-G                           | Same as RU-87P4 but with 8 empty slots  |
| I-87K4                              | Remote I-87K I/O base, 4 empty slots, RS-485 signal   |
| I-87K5                              | Remote I-87K I/O base, 5 empty slots, RS-485 signal   |
| I-87K8                              | Remote I-87K I/O base, 8 empty slots, RS-485 signal   |
| I-87K9                              | Remote I-87K I/O base, 9 empty slots, RS-485 signal   |
| I-87K4-G                            | I-87K4 gray color version   |
| I-87K5-G                            | I-87K5 gray color version   |
| I-87K8-G                            | I-87K8 gray color version   |
| I-87K9-G                            | I-87K9 gray color version   |
|                                     |   |
| <b>Radio I/O expansion base</b>     | Accepts I-87K serial I/O boards   |
| RF-87K1                             | Remote I-87K I/O base, 1 empty slot, Radio signal (Can work together with SST-2450)   |
| RF-87K2                             | Remote I-87K I/O base, 2 empty slots, Radio signal (Can work together with SST-2450)  |
| RF-87K5                             | Remote I-87K I/O base, 5 empty slots, Radio signal (Can work together with SST-2450)  |
| RF-87K9                             | Remote I-87K I/O base, 9 empty slots, Radio signal (Can work together with SST-2450)  |
|                                     |   |
| <b>FRNET board &amp; IO Modules</b> |   |
| I-8172                              | 2-port FRNET Master board   |
| FR-2053                             | 16-ch. Isolated Digital Input module (20-pin header)  |
| FR-2053T                            | 16-ch. Isolated Digital Input module (20-pin screw terminal connector)  |
| FR-2057                             | 16-ch. Isolated Digital Output module (20-pin header)   |
| FR-2057T                            | 16-ch. Isolated Digital Output module (20-pin screw terminal connector)   |
| FR-32P                              | 32-ch. Isolated Digital Input module  |
| FR-32R                              | 32-ch. Relay output module. Nominal load: 3A/125Vdc, 3A/270Vac  |
|                                     |   |
| <b>Ethernet I/O</b>                 |   |
| I-8KE4-MTCP                         | 4-slot Modbus/TCP I/O Unit  |
| I-8KE8-MTCP                         | 8-slot Modbus/TCP I/O Unit  |
|                                     |   |
| <b>Motion control board</b>         |   |
| I-8091                              | 2-axes stepping/servo motor control card  |
| I-8090                              | 3-axes encoder card   |
|                                     |   |
| <b>Timer/Counter board</b>          |   |
| I-8080                              | 4-ch. counter/frequency, 32 bit , 450K Hz max.  |
| I-8080-G                            | I-8080 gray color version   |
| I-87082                             | 2 channel counter/Frequency, 32 bit, 100K Hz max.   |
| I-87082-G                           | I-87082 gray color version  |
|                                     |   |
| <b>RS-232/485/422 board</b>         |   |
| I-8112                              | 2-port RS232  |
| I-8114                              | 4-port RS232  |
| I-8142                              | 2-port RS485/422  |
| I-8142i                             | 2-port isolated RS485/422   |
| I-8144                              | 4-port RS485/422  |

|                                   |   |
|-----------------------------------|---|
| I-8112-G                          | I-8112 gray color version   |
| I-8114-G                          | I-8114 gray color version   |
| I-8142-G                          | I-8142 gray color version   |
| I-8142i-G                         | I-8142i gray color version  |
| I-8144-G                          | I-8144 gray color version   |
|                                   |   |
| <b>Parallel analog I/O board</b>  |   |
| I-8017H                           | 8-ch. 14-bit analog input, each ch. can be different input type (V, mA) & range.<br>Range of +/- 20 mA need external resistor of 125 ohms.<br>Differential input.   |
| I-8017H-G                         | I-8017H gray color version  |
| I-8017HS (high-profile)           | 8-ch. 14-bit analog input, each ch. can be different input type (V, mA) & range.<br>Range of +/- 20 mA need external resistor of 125 ohms.<br>Differential or Single-ended input, module is taller than I-8017H |
| I-8017HS-G (high-profile)         | I-8017HS gray color version   |
| I-8024                            | 4-ch. 14-bit analog output, each ch. can be different output type (V,mA) & range  |
| I-8024-G                          | I-8024 gray color version   |
|                                   |   |
| <b>Parallel digital I/O board</b> |   |
| I-8037                            | 16-ch. isolated open-source output  |
| I-8037-G                          | I-8037 gray color version   |
| I-8040                            | 32-ch. isolated digital input   |
| I-8040-G                          | I-8040 gray color version   |
| I-8041                            | 32-ch. isolated digital output  |
| I-8041-G                          | I-8041 gray color version   |
| I-8042                            | Isolated digital 16-ch. input & 16-ch. output   |
| I-8042-G                          | I-8042 gray color version   |
| I-8050                            | 16-ch. universal Digital I/O, each channel can be config as input or output (isolated)  |
| I-8050-G                          | I-8050 gray color version   |
| I-8051                            | 16-ch. non-isolated digital input   |
| I-8051-G                          | I-8051 gray color version   |
| I-8052                            | 8-ch. isolated digital input (differential)   |
| I-8052-G                          | I-8052 gray color version   |
| I-8053                            | 16-ch. isolated digital input   |
| I-8053-G                          | I-8053 gray color version   |
| I-8054                            | Isolated digital 8-ch. input & 8-ch. output   |
| I-8054-G                          | I-8054 gray color version   |
| I-8055                            | Non-isolated digital 8-ch. input & 8-ch. output   |
| I-8055-G                          | I-8055 gray color version   |
| I-8056                            | 16-ch. non-isolated O.C. output   |
| I-8056-G                          | I-8056 gray color version   |
| I-8057                            | 16-ch. isolated O.C. output   |
| I-8057-G                          | I-8057 gray color version   |
| I-8058                            | 8-ch. isolated digital input, AC/DC input, max. 250V  |
| I-8058-G                          | I-8058 gray color version   |
| I-8060                            | 6-ch. relay output , AC: 125V @0.6A; 250V @0.3A, DC: 30V @2A; 110V@0.6A   |
| I-8060-G                          | I-8060 gray color version   |
| I-8063                            | Isolated digital 4-ch. input & 4-ch. relay , AC: 125V @0.6A; 250V @0.3A   |
| I-8063-G                          | I-8063 gray color version   |
| I-8064                            | 8-ch. power relay output , AC: 250V @5A, DC: 30V @5A  |
| I-8064-G                          | I-8064 gray color version   |
| I-8065                            | 8-ch. SSR-AC output, AC: 24 to 265Vrms @1.0Arms, Max. load current: 1.0 Arms  |



|                            |   |
|----------------------------|---|
| I-8065-G                   | I-8065 gray color version   |
| I-8066                     | 8-ch. SSR-DC output, DC: 3~30VDC@1.0A, Max. load current: 1.0A  |
| I-8066-G                   | I-8066 gray color version   |
| I-8068                     | 8-ch. relay output, AC: 120V @0.5A, DC:30V @ 1A   |
| I-8068-G                   | I-8068 gray color version   |
| I-8069                     | 8-ch. Photo Mos relay output  |
| I-8069-G                   | I-8069 gray color version   |
| I-8077                     | Simulation board: 8-ch. digital input (DIP switch) & 8-ch. output (Leds)  |
| I-8077-G                   | I-8077 gray color version   |
| Serial analog I/O board    |   |
| I-87013W (high-profile)    | 4-ch. RTD input (temperature measurement with broken-line detect)   |
| I-87015 (high-profile)     | 7-ch. RTD input (temperature measurement with broken-line detect)   |
| I-87016W (high-profile)    | 2-ch. Isolated Strain Gauge Input (Will be available)   |
| I-87017R (high-profile)    | I-87017 with Over voltage protection: +/- 240 Vrms  |
| I-87017RC (high-profile)   | 8-ch. analog input. Accept 4-20mA, 0-20mA or +/- 20mA. No external resistor needed  |
| I-87017W-A5 (high-profile) | 8-ch. analog input. Can measure up to +/- 50VDC , +/- 150VDC. with Over voltage protection. Gray color  |
| I-87018Z (high-profile)    | 10-ch. Input, thermocouple or 4-20mA, 0-20mA or +/- 20mA . or +/- 15mV , +/-50mV, +/- 100 mV, +/- 500 mV , +/- -1V or +/- 2.5 V (temperature measurement with broken-line detection) each ch. can be different input type (T/C or V or mA) & range i-87018z is better than i-87018, i-87018R, i-87018-G, i-87018R-G |
| I-87018R (high-profile)    | 87018 with Over voltage protection: +/- 240 Vrms (temperature measurement with broken-line detection) (I-87018Z is better)  |
| I-87019R (high-profile)    | 8-ch. universal Input with Over voltage protection: +/- 240 Vrms (V, mA, Thermocouple), each channel can be different input type and range (temperature measurement with broken-line detection) Range of +/- 20 mA need to set jumper on board.   |
| I-87022W (high-profile)    | 2-ch. 12-bit analog output, each ch. can be different output type (V,mA) & range, support channel to channel isolation  |
| I-87024W (high-profile)    | 4-ch. 14-bit analog output  |
| I-87026W (high-profile)    | 2-ch. 16-bit analog output, each ch. can be different output type (V,mA) & range, support channel to channel isolation  |
| Serial digital I/O board   |   |
| I-87040W (high-profile)    | 32-ch. isolated digital input   |
| I-87041W (high-profile)    | 32-ch. isolated digital output  |
| I-87046W (high-profile)    | 16-ch. isolated digital input   |
| I-87051W (high-profile)    | 16-ch. non-isolated digital input   |
| I-87052W (high-profile)    | 8-ch. isolated digital input (differential)   |
| I-87053W (high-profile)    | 16-ch. isolated digital input   |
| I-87053W-A5 (high-profile) | 16-ch. isolated digital input (ON: +68V to +150VDC , OFF: +48VDC max.)  |
| I-87054W (high-profile)    | Isolated digital 8-ch. input & 8-ch. output   |
| I-87055W (high-profile)    | Non-isolated digital 8ch. input & 8ch. output   |
| I-87057W (high-profile)    | 16-ch. isolated O.C. output   |
| I-87058W (high-profile)    | 8-ch. isolated digital input, AC/DC input, max. 250V  |
| I-87063W (high-profile)    | Isolated digital 4-ch. input & 4-ch. relay, AC: 125V @0.6A; 250V @0.3A DC: 30V @2A; 110V@0.6A   |
| I-87064W (high-profile)    | 8-ch. power relay output, AC: 250V @5A, DC:30V @5A  |
| I-87065W (high-profile)    | 8-ch. SSR-AC output, AC: 24 to 265Vrms @1.0Arms, Max. load current: 1.0 Arms  |

|                                 |   |
|---------------------------------|---|
| I-87066W (high-profile)         | 8-ch. SSR-DC output, DC: 3-30VDC @ 1A, Max. load current: 1.0A  |
| I-87068W (high-profile)         | 8-ch. relay output, AC: 120V @0.5A, DC:30V @1A  |
| I-87069W (high-profile)         | 8-ch. photo Mos relay   |
| <b>Conveter &amp; Repeater</b>  |   |
| PCISA-7520R                     | PCI / ISA bus RS-232 to RS-485 card   |
| PCISA-7520AR                    | PCI / ISA bus RS-232 to RS-485/422 card   |
| I-7520                          | RS-232 to RS-485 converter  |
| I-7520R                         | I-7520 with 3000V DC isolation at RS-485 side   |
| I-7520A                         | RS-232 to RS-422/RS-485 converter   |
| I-7520AR                        | I-7520A with 3000V DC isolation at RS-422/485 side  |
| I-7560                          | USB to RS-232 Converter   |
| I-7561                          | USB to RS-232/422/485 Converter   |
| I-7563                          | USB to 1-ch RS-485 conveter with a three way RS485 Hub (isolated)   |
| I-7510                          | RS-485 isolated repeater  |
| I-7510A                         | RS485/RS422 isolated repeater   |
| I-7510AR                        | Three way Isolated RS-422/485 Repeater  |
| <b>RS-485 Hub</b>               |   |
| I-7513                          | 3-way isolated RS485 to 3 ports RS485 hub   |
| <b>Man Machine Interface</b>    |   |
| Touch506L                       | 5.7" 4-Gray STN Panel display with touch  |
| Touch506T                       | 5.6" 256-color TFT panel LCD Display with Touch   |
| Touch510T                       | 10.4" Color TFT Panel Display With Touch  |
| <b>Wireless Modem</b>           |   |
| SST-2450                        | Wireless Modem Module with RS-232/RS-485 Interface  |
| <b>I-7000 analog I/O module</b> |   |
| I-7011P                         | 1-ch. thermo-couple input (16-bit), 1-ch. D/I & 2-ch. D/O, enhanced version of I-7011   |
| I-7011PD                        | I-7011P with display  |
| I-7012                          | 1-ch. analog input (16-bit), 1-ch. D/I & 2-ch. D/O  |
| I-7012D                         | I-7012D with display  |
| I-7012F                         | High speed version of I-7012 (12-bit), normal 16-bit  |
| I-7012FD                        | I-7012F with display  |
| I-7013                          | ch. RTD input (16-bit)<br>temperature measurement with broken-line detection  |
| I-7013D                         | I-7013 with display   |
| I-7014D                         | 1-ch. analog/transmitter input (16-bit) with display, 1-ch. D/I & 2-ch. D/O   |
| I-7015                          | 6-ch. 2-or-3 wire RTD input (16-bit)<br>temperature measurement with broken-line detection  |
| I-7016                          | 2-ch. strained gauge input (16-bit), 1-ch. D/I & 4-ch. D/O  |
| I-7016D                         | I-7016 with display   |
| I-7016P                         | 1-ch. strained gauge input for longer cable length (16-bit), 1-ch. D/I & 4-ch. D/O  |
| I-7016PD                        | I-7016P with display  |
| I-7017                          | 8-ch. analog input (16-bit), support voltage input and current input. Range of +/- 20 mA need external resistor of 125 ohms (I-7017R is better) |
| I-7017C                         | I-7017 with current input only (+/- 20mA). No external resistor needed. (I-7017RC is better)  |
| I-7017R                         | 8-ch. analog input (16-bit), support voltage input and current input,   |

|                                  |   |
|----------------------------------|---|
|                                  | Robust & high speed version of I-7017, higher Over voltage protection: (240Vrms).<br>Range of +/- 20 mA need external resistor of 125 ohms  |
| I-7017RC                         | I-7017R with current input only (+/- 20mA). No external resistor needed   |
| I-7017R-A5 (will be available)   | 8-ch. analog input (16-bit), can measure +/- 50VDC , +/-150VDC<br>With Over voltage protection  |
| I-7018Z                          | 10-ch. Input, thermocouple or 4-20mA, 0-20mA or +/- 20mA . or +/- 15mV ,<br>+/-50mV, +/- 100 mV, +/- 500 mV , +/-1V or +/- 2.5 V<br>(temperature measurement with broken-line detection)<br>each ch. can be different input type (T/C or V or mA) & range<br>i-7018z is better than i-7018, i-7018R, i-7018P & their “G” : Gray color version |
| I-7018                           | 8-ch. thermocouple input (16-bit). (without broken-line detection) (7018z is better)  |
| I-7018P                          | 8-ch. thermocouple input (16-bit), add 2 thermo-input type: L, M<br>(without broken-line detection) (7018z is better)   |
| I-7018R                          | Enhanced version of I-7018P with temperature broken-line detection & higher Over<br>voltage protection: (240Vrms) (7018z is better)   |
| I-7019R                          | 8-ch. universal analog input, (voltage, current & Thermocouple),<br>higher Over voltage protection: (240Vrms),<br>each channel can be different input type and range<br>(temperature measurement with broken-line detection)<br>Range of +/- 20 mA need to set jumper on board  |
| I-7021                           | 1-ch. analog output (12-bit)  |
| I-7021P                          | 1-ch. analog output (16-bit)  |
| I-7022                           | 2-ch. analog output (12-bit), each ch. can be different output type (V,mA) & range  |
| I-7024                           | 4-ch. analog output (14-bit)  |
| I-7033                           | 3-ch. RTD input (16-bit)<br>temperature measurement with broken-line detection  |
| I-7033D                          | I-7033 with display   |
| <b>I-7000 digital I/O module</b> |   |
| I-7041                           | 14-ch. isolated digital input   |
| I-7041D                          | I-7041 with LED display   |
| I-7042                           | 13-ch. isolated O.C. output   |
| I-7042D                          | I-7042 with LED display   |
| I-7043                           | 16-ch. non-isolated O.C. output   |
| I-7043D                          | I-7043 with LED display   |
| I-7044                           | Isolated digital 4-ch. input & 8-ch. output   |
| I-7044D                          | I-7044 with LED display   |
| I-7045                           | 16-ch. Isolated digital output  |
| I-7045D                          | I-7045 with LED display   |
| I-7050                           | Non-isolated 7-ch. digital input (sink) & 8-ch. output  |
| I-7050D                          | I-7050 with LED display   |
| I-7050A                          | 7 digital input & 8 output (current source)   |
| I-7050AD                         | I-7050A with LED display  |
| I-7052                           | 8-ch. isolated digital input (6 differential + 2 single end)  |
| I-7052D                          | I-7052 with LED display   |
| I-7053                           | Non-isolated 16-ch. digital input   |
| I-7053D                          | I-7053 with LED display   |
| I-7055                           | 8-ch. isolated input & 8-ch. isolated output  |
| I-7055D                          | I-7055 with LED display   |
| I-7060                           | 4-ch. isolated input & 4-ch. relay output, AC: 125V @0.6A; 250V @0.3A, DC: 30V<br>@2A; 110V @0.6A   |
| I-7060D                          | I-7060 with LED display   |
| I-7063                           | 8-ch. isolated input & 3-ch. power relay, AC: 250V @5A, DC: 30V @5A   |

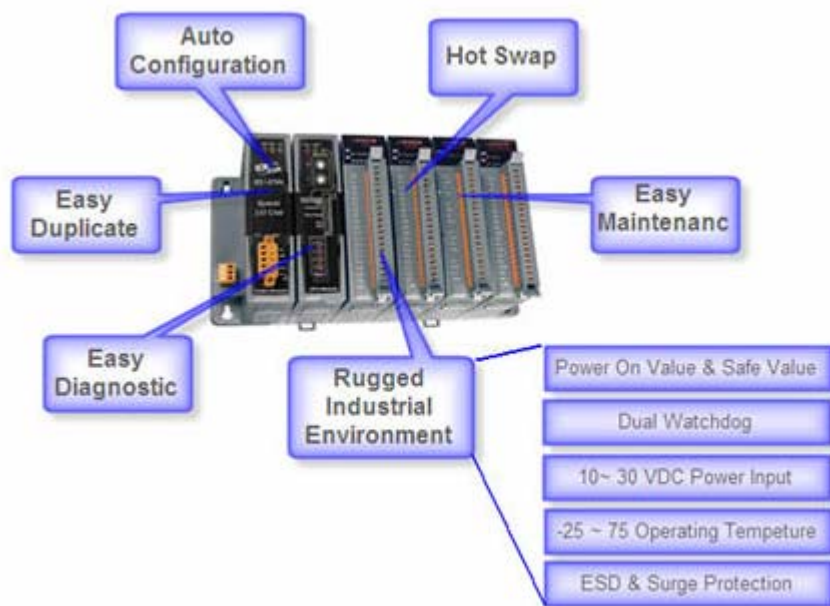
|                         |  |
|-------------------------|--|
| I-7063D                 | I-7063D with LED display   |
| I-7063A                 | 8-ch. isolated input & 3-ch. AC-SSR output, AC:24~ 265Vrms @1.0Arms, Max. Load current 1.0Arms                                 |
| I-7063AD                | I-7063A with LED display   |
| I-7063B                 | 8-ch. isolated input & 3ch. DC-SSR output, DC:3~ 30Vdc @1.0Arms, Max. Load current 1.0Arms                                     |
| I-7063BD                | I-7063B with LED display   |
| I-7065                  | 4-ch. isolated input & 5-ch. power relay, AC: 250VAC @5A, DC: 30V @5A  |
| I-7065D                 | I-7065 with LED display  |
| I-7065A                 | 4-ch. isolated input & 5ch. AC-SSR relay, AC:24~ 265Vrms @1.0Arms, Max. Load current 1.0Arms                                   |
| I-7065AD                | I-7065A with LED display   |
| I-7065B                 | 4-ch. isolated input & 5ch. DC-SSR relay, DC:3~ 30Vdc @1.0Arms, Max. Load current 1.0Arms                                      |
| I-7065BD                | I-7065B with LED display   |
| I-7066                  | 7-ch. Photo Mos relay output   |
| I-7066D                 | I-7066 with LED display  |
| I-7067                  | 7-ch. relay output, AC 120V@0.5A, DC 24V@1A  |
| I-7067D                 | I-7067 with LED display  |
|                         |  |
| I-7000 counter module   |  |
| I-7080                  | 2 high speed counter/frequency input (32-bit), 100K Hz max.  |
| I-7080D                 | I-7080 with display  |
|                         |  |
| X-board (parallel I/O)  | For I-7188XG & I-7188EG  |
| X107                    | Non-isolated 6-ch. D/I and 7-ch. D/O   |
| X109                    | Isolated 7-ch. PhotoMos Relay  |
| X110                    | Non-isolated 14-ch. D/I  |
| X111                    | Non-isolated 13-ch. D/O  |
| X116                    | Isolated 4-ch. D/I and 6-ch. Relay (without case)  |
| X119                    | Non-isolated 7-ch. D/O and 7-ch. D/I (without case)  |
| X202                    | Non-isolated 7-ch. A/D (0~20mA, 12-bit)  |
| X203                    | Non-isolated 2-ch. A/D (0~20mA, 12-bit), 2-ch. D/I, 6-ch. D/O  |
| X303                    | Non-isolated 1-ch. A/D (+/-5V, 12-bit), 1-ch. D/A (+/-5V, 12-bit), 4-ch. D/I, 6-ch. D/O  |
| X304                    | Non-isolated 3-ch. A/D (+/-5V, 12-bit), 1-ch. D/A (+/-5V, 12-bit), 4-ch. D/I, 4-ch. D/O  |
| X305                    | Non-isolated 7-ch. A/D (+/-5V, 12-bit), 1-ch. D/A (+/-5V, 12-bit), 2-ch. D/I, 2-ch. D/O  |
| X307                    | Non-isolated 8-ch. A/D (+/-10V, 12-bit), 2-ch. D/I, 2-ch. D/O (will be available)  |
| X308                    | Non-isolated 4-ch. A/D (+/-10V, 12-bit), 6-ch. D/O   |
| X310                    | Non-isolated<br>2-ch. A/D, ch.0:(0~20mA, 12-bit), ch.1: A/D (0~10V, 12-bit)<br>2-ch. D/A (0~10V, 12-bit), 3-ch. D/I, 3-ch. D/O |
|                         |  |
| X-board (RS232/422/485) | For I-7188XG & I-7188EG  |
| X503                    | 1-Port RS-232 (5-Pin)  |
| X504                    | 2-Port RS-232, one is 5-Pin , one is 9-Pin   |
| X505                    | 3-Port RS-232 (5-Pin)  |
| X506                    | 6-Port RS-232 (3-Pin)  |
| X507                    | 1-Port RS-422/485, Non-isolated 4-ch. D/I, 4-ch. D/O   |
| X508                    | 1-Port RS-232 (5-Pin), Non-isolated 4-ch. D/I, 4-ch. D/O   |
| X509                    | 2-Port RS-232 (3-Pin), Non-isolated 4-ch. D/I, 4-ch. D/O   |

|                               |   |
|-------------------------------|---|
| X510                          | 1-Port RS-232 (3-Pin ), Non-isolated 5-ch. D/I, 5-ch. D/O, EEPROM 128K x2 |
| X511                          | 3-Port RS-485   |
| X518                          | 1-Port RS-232 (5-Pin), Non-isolated 8-ch. D/O                             |
| X560                          | 3-Port RS-232 (3-Pin), 8M bytes Flash memory (without case)               |
| X-board (Battery backup SRAM) | For I-7188XG & I-7188EG   |
| X607                          | 128Kbytes battery backup SRAM   |
| X608                          | 512Kbytes battery backup SRAM   |
| Industrial Ethernet Switch    |   |
| NS-205                        | Unmanaged 5-Port Industrial 10/100 Base-T Ethernet Switch                 |
| NS-208                        | Unmanaged 8-Port Industrial 10/100 Base-T Ethernet Switch                 |

### New Products – RU-87P4 (4 slots) , RU-87P8 (8 slots) : RS-485 expansion unit

- Better than i-87K4/ i-87K8
- If using with i-87K high-profile I/O cards, it supports hot-swap
- Auto-configuration: Only need to config it once at design time, then at run time, just plug and play
- Communication Addr, Baud rate & Checksum are set by rotary and dip switch.
- All ISaGRAF controllers support them just like supporting the i-87K4/8 (please refer to Chapter 6)

More at [http://www.icpdas.com/products/io\\_expansion\\_unit/ru-87p/ru-87p4.htm](http://www.icpdas.com/products/io_expansion_unit/ru-87p/ru-87p4.htm)



---

# Chapter 1. Software & Hardware Installation

---

## NOTE:

The I-8xx7 abbreviation is for the I-8417, I-8437, I-8817 and I-8837 controllers, while W-8xx7 is the abbreviation for the Wincon-8037/8337/8737 and Wincon-8047/8347/8747 controller.

## 1.1: Installing The ISaGRAF Workbench Software Program

---

Chapter 1 of the "User's Manual of ISaGRAF PAC" manual details how to properly setup and run the I-8xx7, I-7188EG/XG , uPAC-7186EG, iPAC-8447/8847 & W-8xx7 controller system and the ISaGRAF Workbench software program.

Numerous illustrations and pictures are provided in this chapter to assist the integrator and programmer with the basics of how to properly setup the hardware and software for their system.

If you are not familiar with the setup of either the I-8xx7, I-7188EG/XG , uPAC-7186EG, iPAC-8447/8847 & W-8xx7 controller system or the ISaGRAF Workbench software program, please take the time to thoroughly read Chapter 1. The procedures detailed in this chapter are easy to understand, and will assist the user to quickly and easily setup and start running the controller and the ISaGRAF software program.

For the I-8xx7, I-7188EG/XG , uPAC-7186EG, iPAC-8447/8847 & W-8xx7 controller system and the ISaGRAF Workbench software to operate properly, it is imperative that each is setup correctly. This chapter covers the details of how to setup the controller system and the ISaGRAF Workbench software in a minimum of time.

Before you can start programming the I-8xx7, I-7188EG/XG , uPAC-7186EG, iPAC-8447/8847 & W-8xx7 PAC system with the ISaGRAF software program, you must first install the ISaGRAF Workbench software program on a target PC.

### Hardware Requirements

- A Personal Computer With At Least A Pentium, 133 MHz Or Faster Processor
- 32 Mbytes Memory (Preferably 64 Mbytes RAM)
- A Hard Drive With At Least 128 Mbytes Of Storage Space (Preferably Larger)
- At Least One RS-232 Serial Port

### Software Requirements

One of the following computer operating systems must be installed on the target computer system before you can install the ISaGRAF Workbench software program.

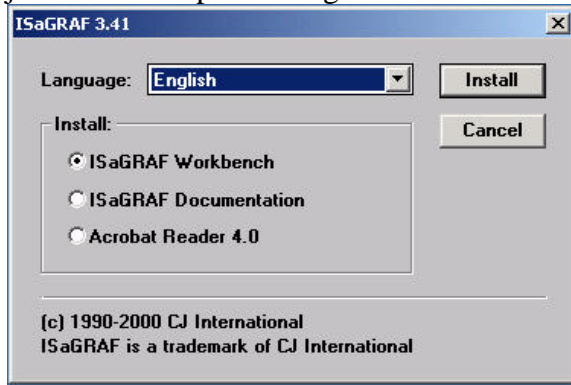
- Windows 95
- Windows 98
- Windows NT Version 3.51 or Windows NT Version 4.0
- Windows 2000 Or Windows XP

### Steps To Installing The ISaGRAF Workbench Program

Insert the ISaGRAF Workbench CD into your CD-ROM drive. Normally the auto-start program will activate the "install.bat" file automatically. If your computer does not have the auto-start feature active,

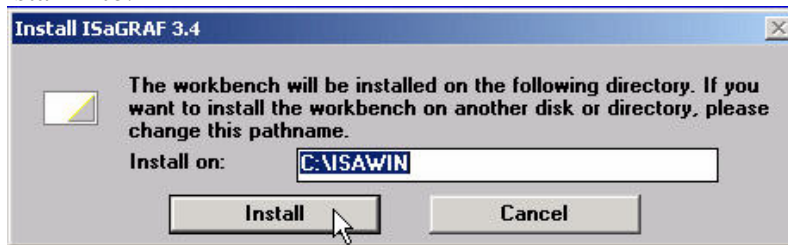
use the Windows Explorer and go to the CD-ROM drive where the Workbench CD is installed, then double-click on the "install.bat" file listed on the ISaGRAF CD. If the "install.bat" file is not found on your ISaGRAF CD, then double-click on the "ISaGRAF.exe" file to start the installation process.

Once you have started the "install.bat" file, a dialog box will appear as shown on the next page. Select the language version of the ISaGRAF software program you would like to use. The English version is used on all subjects and examples throughout this manual.



Once you have selected to install the ISaGRAF Workbench program and selected the desired language, just press the "Install" button, and follow the step-by-step directions of each dialog box as they appear to complete the installation process.

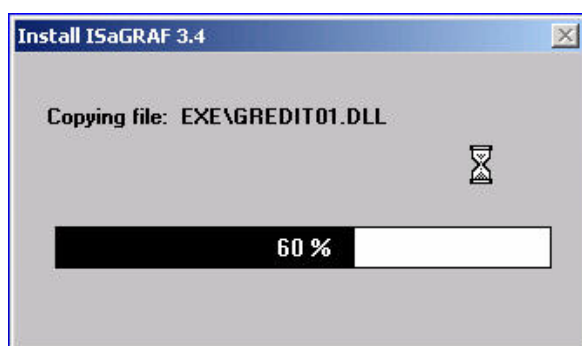
The first dialog box to appear allows the user to define what drive and subdirectory the ISaGRAF program will install into.



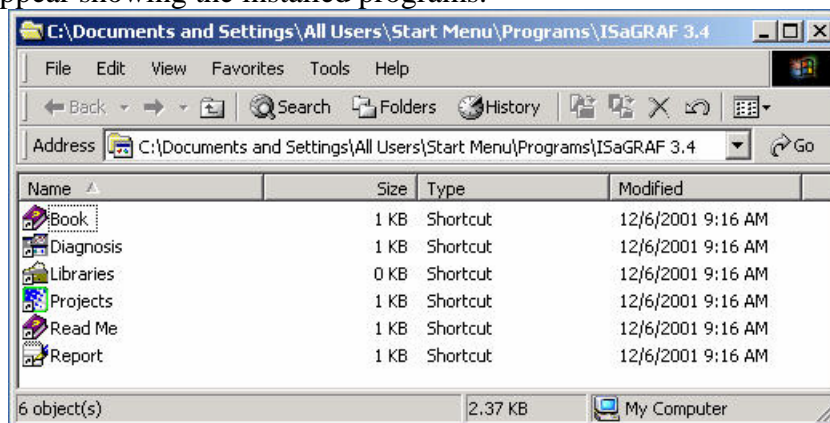
The next dialog box asks the user how much of the ISaGRAF program to you wish to install. By default, it is best to allow all of the ISaGRAF programs to install.



Once you have selected which programs and applications are to be installed, the installation process begins, and an installation progress dialog box will appear showing the installation progress.



Once the ISaGRAF Workbench software installation process has been completed, a Windows Explorer window will appear showing the installed programs.



The installation process is now complete, and you can begin to use the ISaGRAF software program.

To begin the ISaGRAF 3.x software program, click on the Windows "Start" button, then on "Programs", and you should see the ISaGRAF program group as illustrated below.



You will see that six program icons are now associated with the ISaGRAF 3.x software group. You can select any of the icons to learn more about the ISaGRAF Workbench software program.

If your ISaGRAF Key-Pro is USB type, please follow below steps to install the proper USB driver.

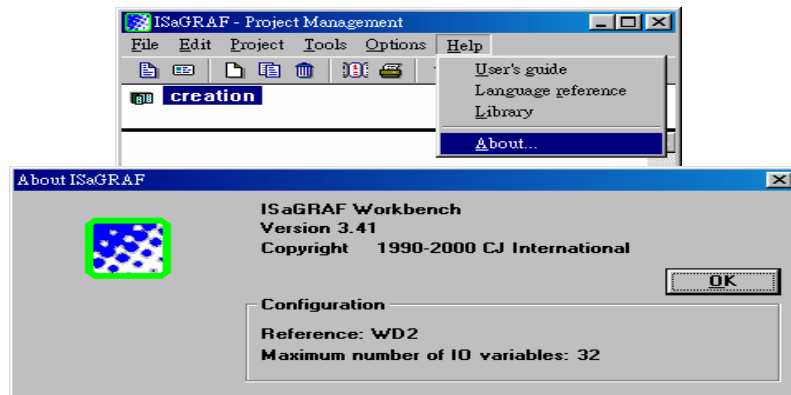
1. To make your PC recognize the ISaGRAF USB protection-key, please **un-plug** the USB protection-key from your USB port first, then run “\Sentinel\SSD5411-32bit.exe“ in the ISaGRAF 3.55 CD-ROM after you have installed the ISaGRAF. Then please reset your PC.

**2. To run ISaGRAF Ver. 3.5x, please always plug the USB protection-key in the PC's USB port.**

**NOTE:** You must install the hardware protection device (dongle) provided with the ISaGRAF software on your computers parallel port to for the ISaGRAF program to achieve fully authorized functionality.



While using ISaGRAF and the dongle is plugged well, if the “Help” – “About” says “Maximum number of IO variables: 32”, it means ISaGRAF workbench cannot find the dongle well. Please reset your PC and then check the “Help” – “About” again. If it still displays “Maximum number of IO variables: 32”,



the dongle driver may not be installed well. Please execute the ISaGRAF CD\_ROM  
 \Sentinel5382\setup.exe for ISaGRAF-80 or  
 \Sentinel\setup.exe for other ISaGRAF version or  
 follow the former section if your protection-key is USB type.  
 and then reset the PC again.

#### Important Notice For Window NT Users

If your computer is using the Windows NT operating system, you will need to add one line to the "isa.ini" file in the ISaGRAF Workbench "EXE" subdirectory. If the ISaGRAF program is installed on your computers "C" hard drive, you will find the required file in the following path:

C:\isawin\exe\isa.ini

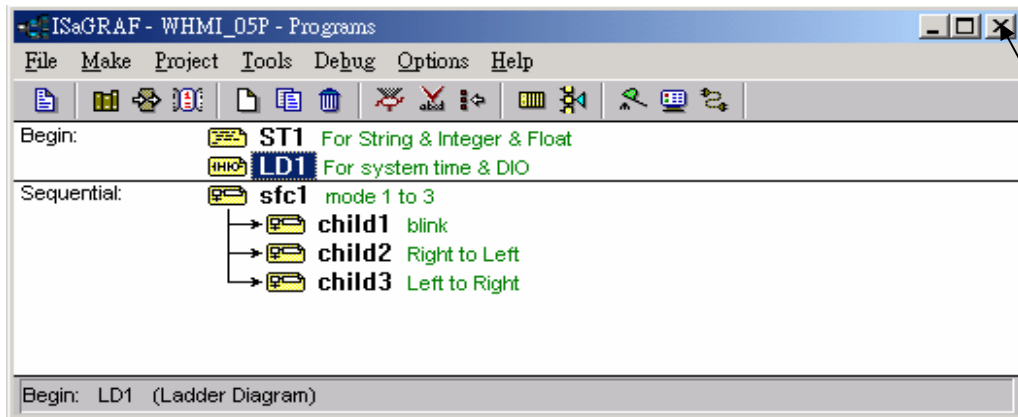
You can use any ASCII based text editor (such as Notepad or UltraEdit32) to open the "isa.ini" file. Locate the [WS001] header in the "isa.ini" initialization file (it should be at the top of the file). Anywhere within the [WS001] header portion of the "isa.ini" initialization file, add the entry shown below within the [WS001] header:

```
[WS001]
NT=1
Isa=C:\ISAWIN
IsaExe=C:\ISAWIN\EXE
Group=Samples
IsaApl=c:\isawin\smpl
IsaTmp=C:\ISAWIN\TMP
```

The [WS001] header should now look like the above example. The **NT=1** entry addition is absolutely required for the RS-232 communications to operate properly in the Windows NT operating environment.

### 1.1.1: When closing my ISaGRAF window on windows 2000, it holds. Why ?

This problem usually happens on the windows 2000. When you close some ISaGRAF windows by clicking on the “X”, it holds about 20 to 40 seconds (No response).



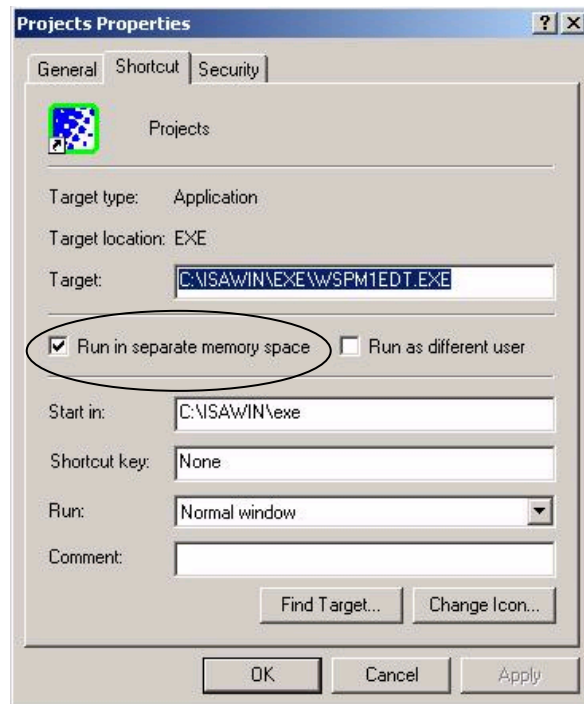
This “hold” behavior is caused by the “CTFMON.EXE” process. We still don’t know the reason yet. You may stop this process by click on the “Ctrl” & “Alt” & “Del” at the same time to open the window Task Manager, and then stop it as next page.

However you will find the “CTFMON.EXE” still load to run when you reboot your PC or run Microsoft Office. So you need to stop it every time when your windows 2000 is rebooted. If you want to know more about the “CTFMON.EXE”, please visit [www.microsoft.com](http://www.microsoft.com) & search “CTFMON.EXE”.



### 1.1.2: One Quick way to avoid the “hold” problem on windows 2000.

You may create a shortcut for the “ISaGRAF project manager. And then check on "run in separate memory space" option in the shortcut property.



## 1.2: Installing The ICP DAS Utilities For ISaGRAF

---

The “ICP DAS Utilities For ISaGRAF” consists of 3 major items.

I/O library definition

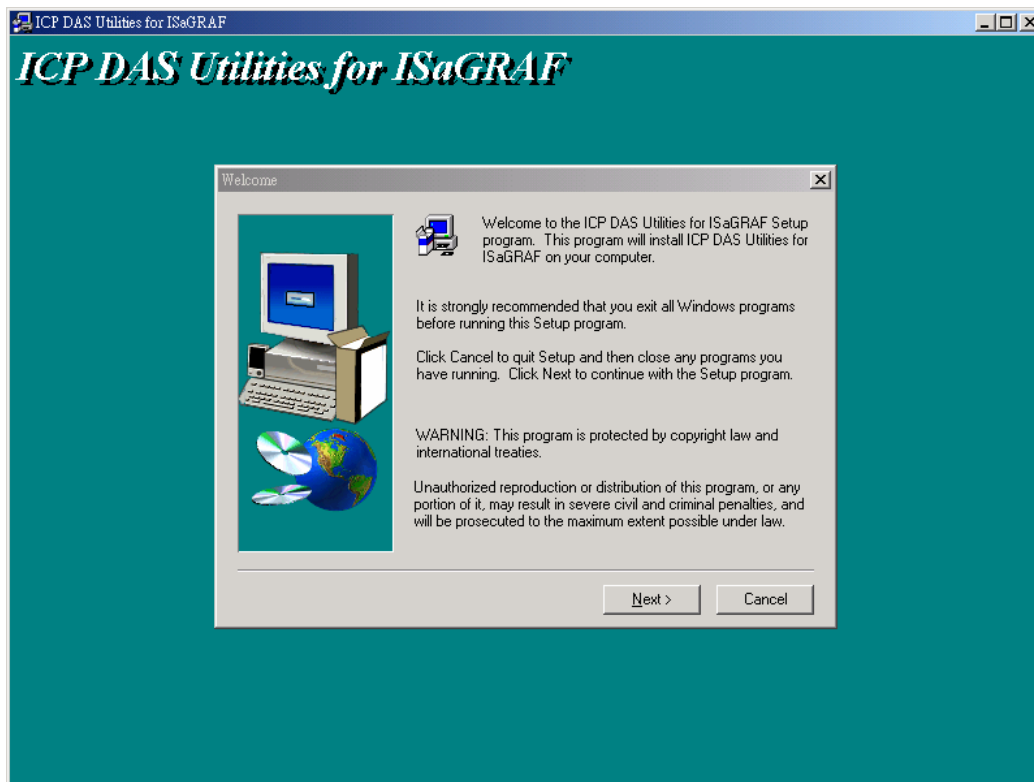
Modem\_Link utility (Chapter 13)

Auto-scan I/O utility (Section 3.6)

The ISaGRAF Workbench software program must be installed before attempting to install the “ICP DAS Utilities for ISaGRAF”. If you have not already installed the ISaGRAF Workbench program, please refer to section 1.1 before continuing.

When the ISaGRAF Workbench program is first installed, it contains only the basic I/O libraries from CJ International - the authors of the ISaGRAF software program. Users will have to install the appropriate I/O library files and some utilities before you can properly program the ISaGRAF controller.

There is a CD-ROM supplied with each of the ISaGRAF controllers with the “ICP DAS Utilities for ISaGRAF”. Please insert the CD-ROM into your CD-ROM drive. Then run “setup.exe” in the folder of CD-ROM: \napdos\isagraf\ . Follow the steps to install it.



Note:

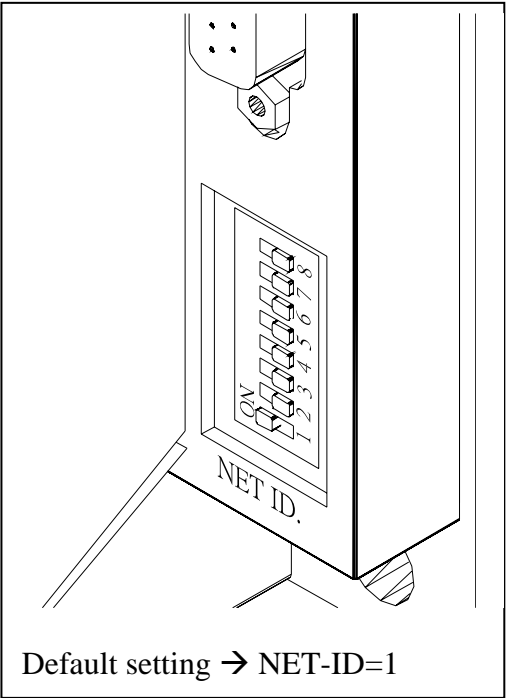
If “setup.exe” is not in your CD-ROM, please download “ICP DAS Utilities For ISaGRAF” from <http://www.icpdas.com/products/PAC/i-8000/isagraf.htm>

# 1.3: Connecting Your PC To The Controller

**Note:** Below sections are for the I-8417/8817/8437/8837 controller only, please refer to the respective “Getting Started” Manual which delivered with the controller for connecting PC to the I-7188EG/XG, uPAC-7186EG, iPAC-8447/8847 or W-8xx7 controller.

## 1.3.1: Setting The NET-ID Addresses For The I-8xx7 Controller System

For the I-8xx7 controller to properly operate, it must first be addressed correctly.

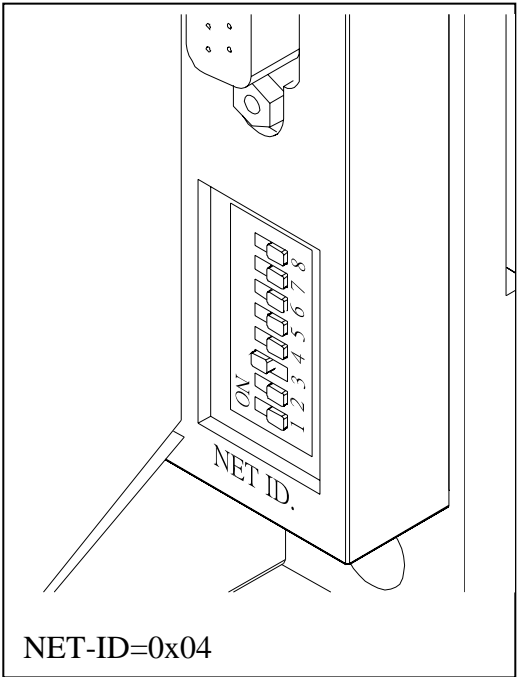
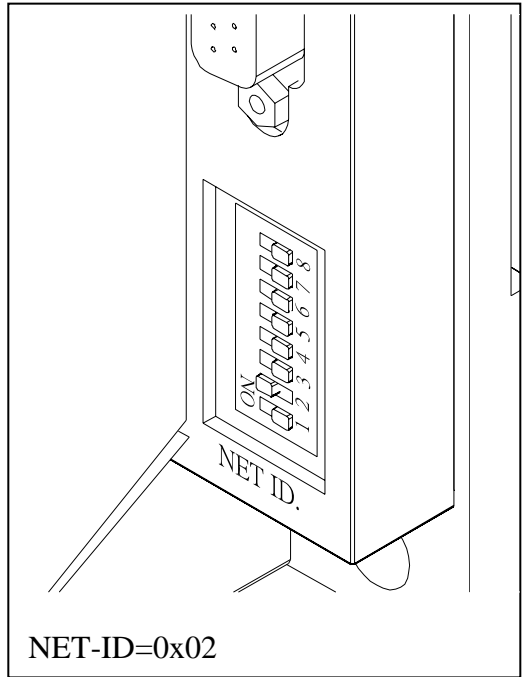


|           | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  |
|-----------|----|----|----|----|----|----|----|----|
| NET-ID=00 |    |    |    |    |    |    |    |    |
| NET-ID=01 | ON |    |    |    |    |    |    |    |
| NET-ID=02 |    | ON |    |    |    |    |    |    |
| NET-ID=03 | ON | ON |    |    |    |    |    |    |
| NET-ID=04 |    |    | ON |    |    |    |    |    |
|           |    |    |    |    |    |    |    |    |
|           |    |    |    |    |    |    |    |    |
|           |    |    |    |    |    |    |    |    |
| NET-ID=FF | ON | ON | ON | ON | ON | ON | ON | ON |

Default setting → NET-ID=01

For ISaGRAF workbench , it can only recognize NET-ID from 01 to FF (1~255).

The NET-ID of every Main Control Unit in the same network must be unique (different from each other).



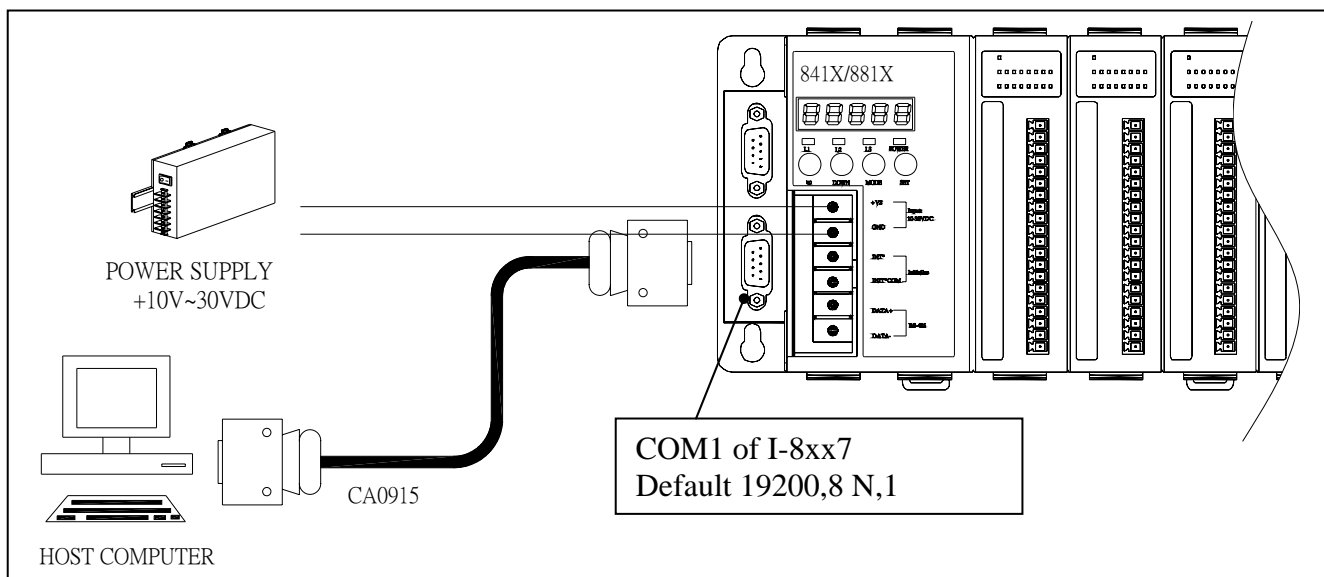
### 1.3.2: Downloading & Communicating Via Modbus With The I-8xx7

The I-8xx7 controller provides two COM ports standard for downloading the ISaGRAF program and debugging your application. The COM1 port is an RS-232 port and the COM2 port is an RS-485 port for the I-8417/8817 controller system, and the I-8437/8837 features an Ethernet port connection instead of a second COM port.

Both of the COM1 and COM2 ports of the I-8417/8817 controllers support the Modbus communications protocol. For I-8437/8837 and iPAC-8447/8847 controllers, COM1 support Modbus protocol while COM2 is an ethernet port support Modbus TCP/IP protocol. There are an abundant number of Human Machine Interface (HMI) and Man Machine Interface (MMI) software programs and additional hardware devices that support the Modbus or/and Modbus TCP/IP communications protocols. All of these programs and devices can access data from the I-8xx7 controller system through the two COM ports using the Modbus / Modbus TCP/IP protocol.

### 1.3.3: Connecting Your PC To The I-8xx7 COM1 Port

When you receive your I-8xx7 controller system, there is one (1) RS-232 communications cable provided with the system. The cable is used to connect your PC to the I-8xx7 controller or to an I-7520 RS-232/RS-485 converter that can be purchased from ICP DAS.



The communication parameters for the I-8xx7 COM1 port defaultly be set to 19200-baud rate, 8 data bits, no stop bits, and one parity bit ("19200, 8, N, 1").

Normal RS-232 Pin Wiring Assignments

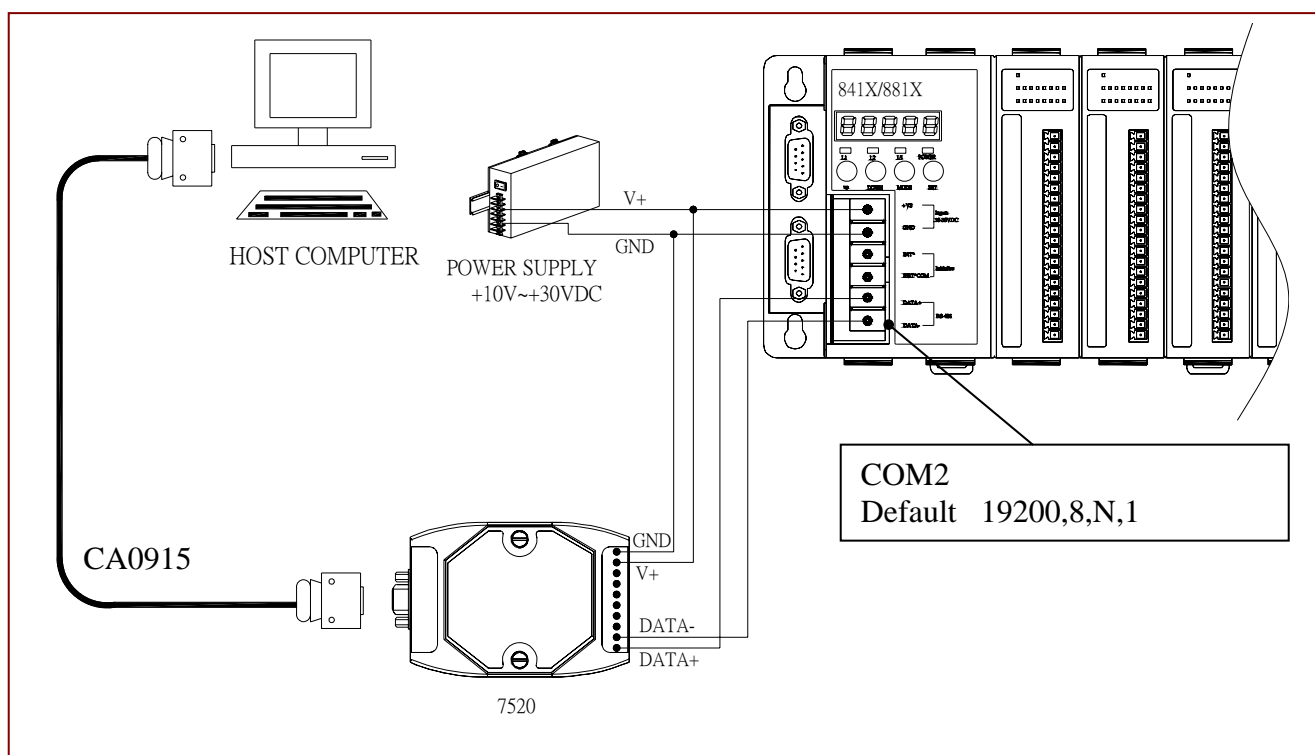
| PC          |       | I-8xx7 |
|-------------|-------|--------|
| 9-Pin D-Sub |       | COM1   |
| RXD 2       | ===== | TXD 2  |
| TXD 3       | ===== | RXD 3  |
| GND 5       | ===== | GND 5  |

For the ISaGRAF Workbench RS-232 communications to operate properly, only the RXD, TXD, and the GND signals are used. If your PC is running a hardware device or software program that uses the CTS and DSR signals, you will need to wire the RTS-CTS and DTR-DSR signals together as shown below.

| PC          |                          | I-8xx7 |  |
|-------------|--------------------------|--------|--|
| 9-Pin D-Sub |                          | COM1   |  |
| RXD 2       | _____                    | TXD 2  |  |
| TXD 3       | _____                    | RXD 3  |  |
| GND 5       | _____                    | GND 5  |  |
| DTR 4       | <input type="checkbox"/> |        |  |
| DSR 6       | <input type="checkbox"/> |        |  |
| RTS 7       | <input type="checkbox"/> |        |  |
| CTS 8       | <input type="checkbox"/> |        |  |

### 1.3.4: Connecting Your PC To The I-8xx7 COM2 Port

If your PC is connecting to an I-8417/8817's COM2 port (RS-485), the maximum distance between the I-7520 (the RS-232/RS-485 converter) and an I-8xx7 controller is up to 1,200 meters (4,000 feet). The distance between the two is dependent on the baud rate; the rule to follow is the lower you set the baud rate, the longer the distance can be.

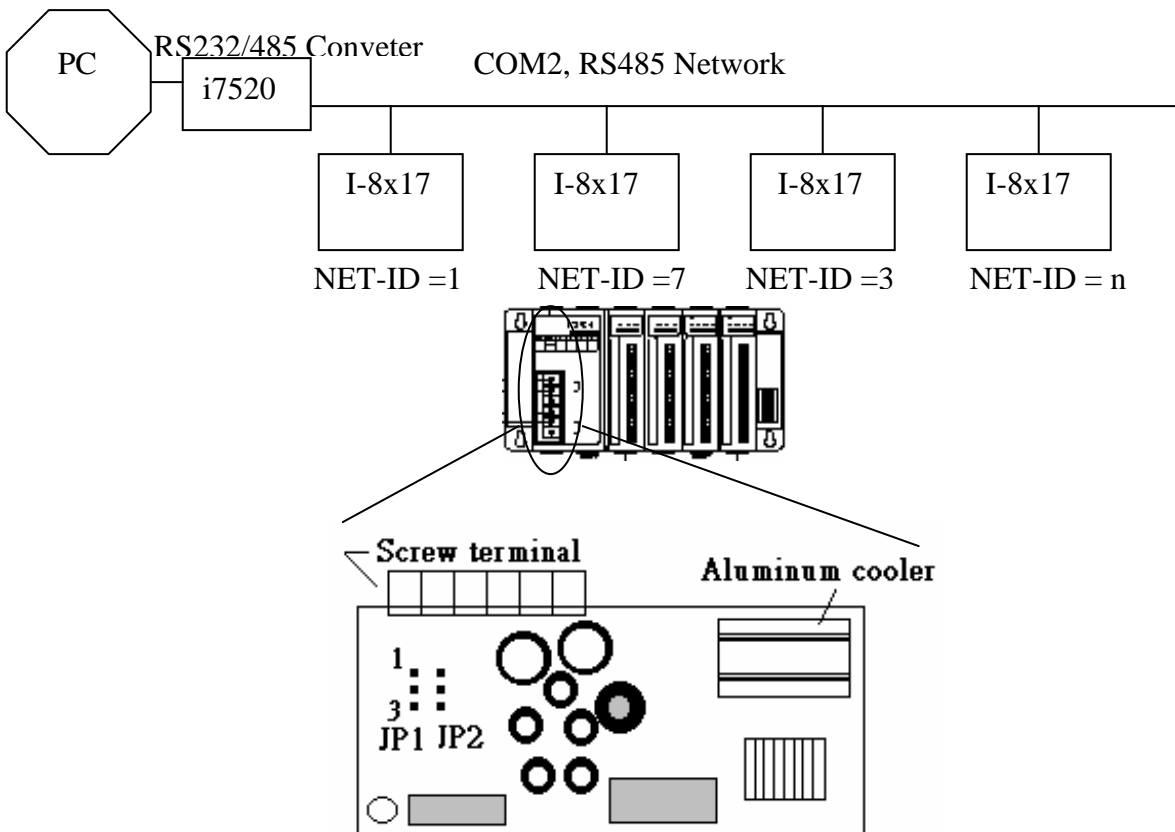


### 1.3.5: Connecting One PC To Several I-8417/8817 Controllers

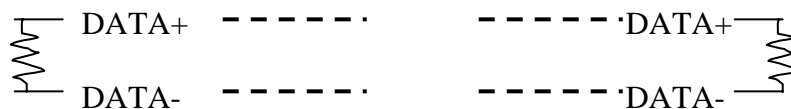
An additional feature of using the COM2 port of the I-8417/8817 is that you can configure an RS-485 network from one PC to link to numerous I-8417/8817 controllers. The PC can download ISaGRAF applications to each I-8417/8817 controller system on the RS-485 network. The maximum number of I-8417/8817 controllers that can be networked via the RS-485 network is 255 (Not recommended to use so many).

To create an RS-485 network you must first insure that each I-8417/8817 controller has a unique NET-ID address, and each of the controllers link the "DATA+" to the "DATA+" signal, and the "DATA-" to the "DATA-" signals.

Lastly, you must plug ONE of the I-8417/8817's JP-1 and JP-2 on the power board to position 1 to 2, (resistance applied to the network). The other I-8417/8817's JP-1 and JP-2 plugs should be left at the default setting of connecting 2 to 3 (no resistance).



It is recommended to add two terminal resistors (try 220Ω, then 110Ω, and then 330Ω) on the nearest I-8417/8817 and farthest I-8417/8817 for long distance RS485 network.



### 1.3.6: Changing The COM1 and COM2 and COM3 Modbus RTU Slave port's Baud Rate Setting

Modbus slave port's default setting:

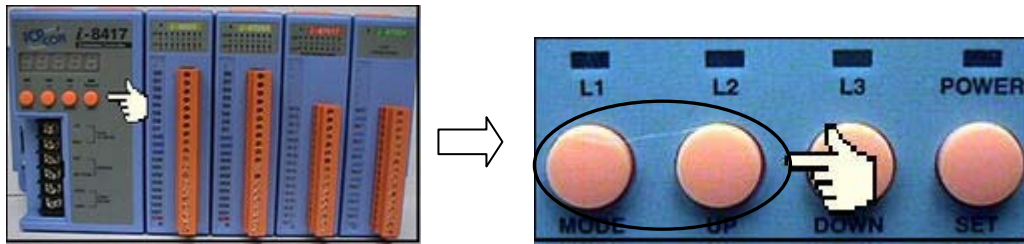
**I-8417 / I-8817: COM1 port(RS-232,19200 ) and COM2 port(RS-485,19200)**

**I-8437 / I-8837 & i-8437-80/8837-80: COM1 port(RS-232,19200 )**

The baud rate can be set between 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bps(bit per second).



To change the baud rate setting on the COM1 & I-8417/8817 's COM2 port, first power off the controller. Then press in and hold in the **first two buttons** on the front panel of the controller and then power back up the controller system as shown below.



The first read out to appear is the "SEL 0" or "SEL 1"

("SEL 0" is to set COM1's baud rate, while "SEL 1" is to set I-8417/8817's COM2 or I-8437 / 8837 's COM3 Modbus RTU Slave baud rate).

|                       |                       |                       |                       |   |
|-----------------------|-----------------------|-----------------------|-----------------------|---|
| S                     | E                     | L                     |                       | 1 |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |   |
| (Up)                  | (Dn)                  | (Ok)                  | (Cancel)              |   |

Press the "Up" or "Dn" to change selection, then press the "OK" button (third button on the panel), and the "BAU x" setting will appear.

|                       |                       |                       |                       |   |
|-----------------------|-----------------------|-----------------------|-----------------------|---|
| B                     | A                     | U                     |                       | 4 |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |   |
| (Up)                  | (Dn)                  | (Ok)                  | (Cancel)              |   |

You can now change the baud rate setting by pressing the "UP" or "Down" button to the desired baud rate setting.

The settings for the **COM1/COM2 /COM3** baud rate are as follows:

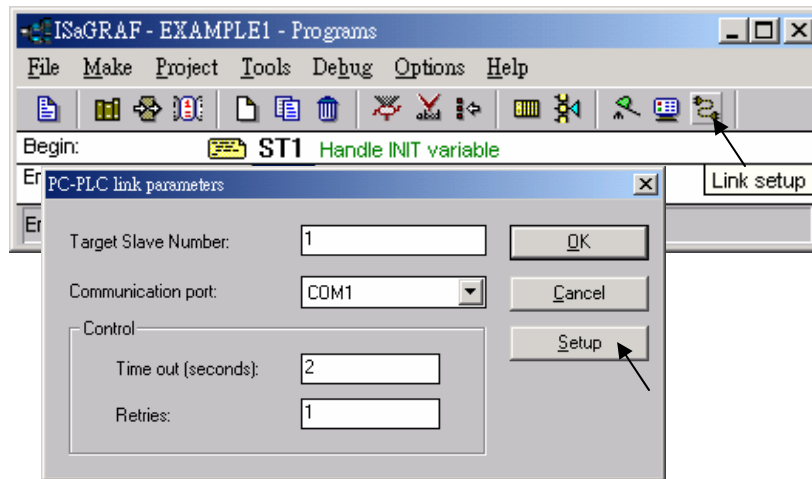
**(0)1200, (1) 2400, (2) 4800, (3) 9600, (4) 19200, (5) 38400, (6) 57600, (7) 115200, (8) 300 (9) 600.**

**The setting (-) None-Modbus slave port is only for the COM3 !**

Press "OK" to save the selected setting. And then press some "Cancel" to exit the hardware setting.

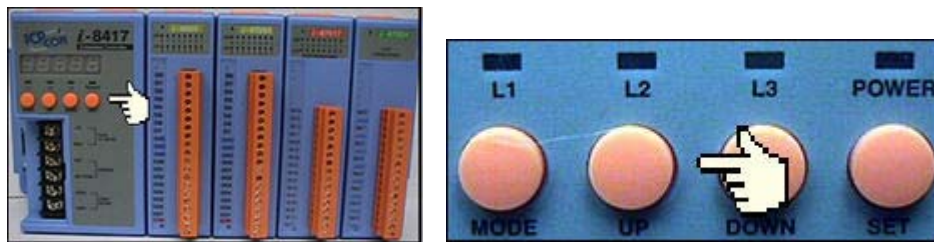
## **IMPORTANT NOTICE:**

The ISaGRAF workbench's default setting for PC's COM1 & COM2 is 19200, 8, N, 1. If you have changed the I-8417/8817/8437/8837 COM1/COM2's baud rate to other value, You should change your ISaGRAF Workbench's COMM to the same setting before they can link to each other. (Please refer to Section 2.5)

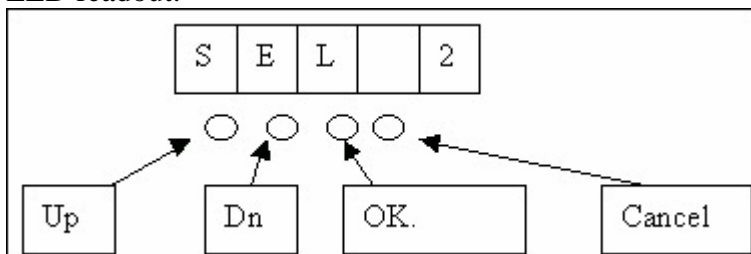


### 1.3.7: Deleting An ISaGRAF Project From The I-8xx7 Controller

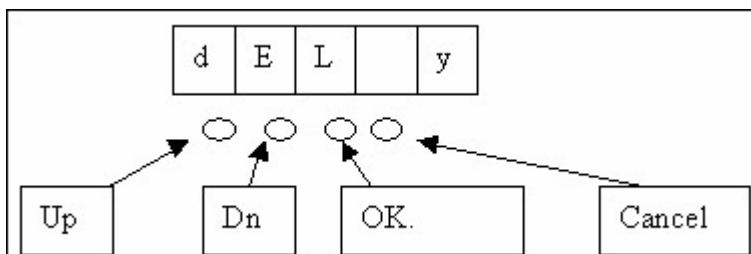
There may be occasions when you will need to delete the ISaGRAF project from the controller system. To begin this, you follow the same control start up routine as changing the baud rate. You first press in and hold in the **first two buttons** on the front panel of the controller and then power back up the I-8417/8817/8437/8837 controller to gain the ability to change the parameters.



When the first display appear, press the "Up" or "Down" button until "SEL 2" (Select 2) appears in the LED readout.



Press the "Up" or "Down" buttons until "dEL" appears in the LED read out.

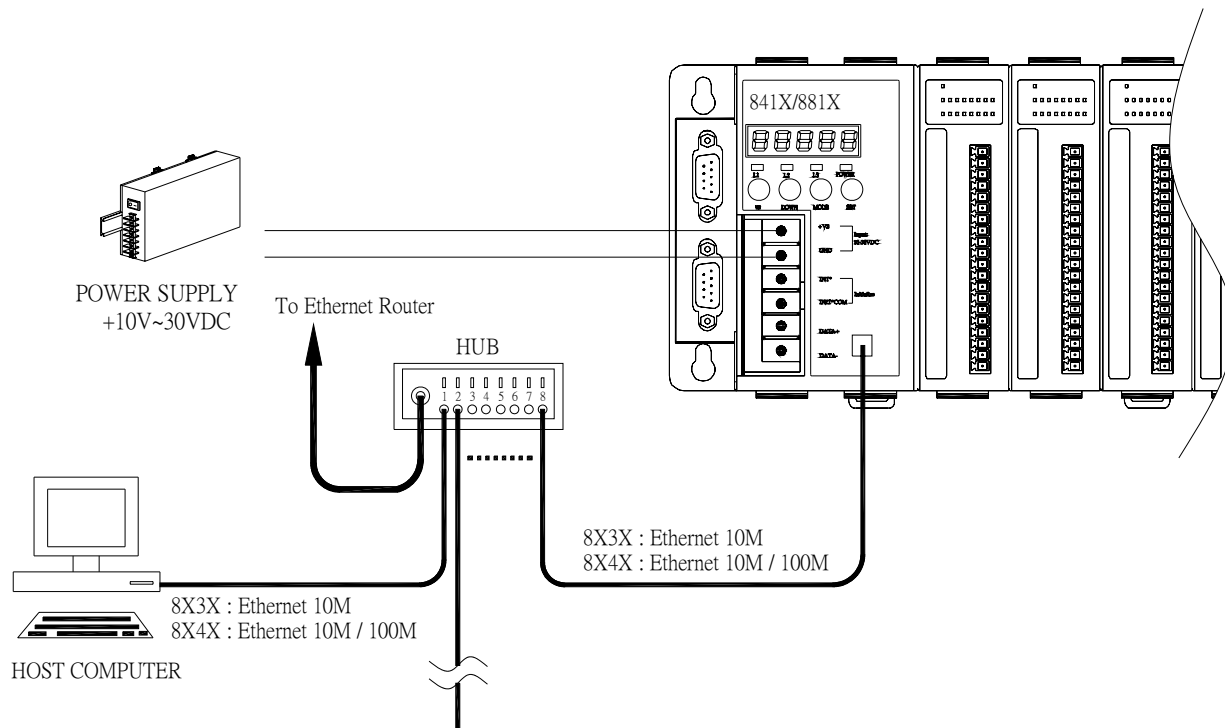


Press the "Up" or "Down" buttons until "y" appears in the LED readout then press the "OK" button. This will delete the currently installed ISaGRAF project from the controller system. After that press some "Cancel" to exit the hardware setting.

### 1.3.8: Connecting Your PC To The I-8437/8837 Ethernet Port

**Note:** If the controller is W-8x47/8x46. There are two Ethernet ports built in the W-8x47/8x46 controllers. Please connect your PC to W-8x47/8x46's "LAN1" port. And please using "NS-205" or "NS-208" Ethernet switch.

The I-8437 and I-8837 controller systems feature a built in Ethernet port. The COM2 port is replaced from an RS-485 to Ethernet.

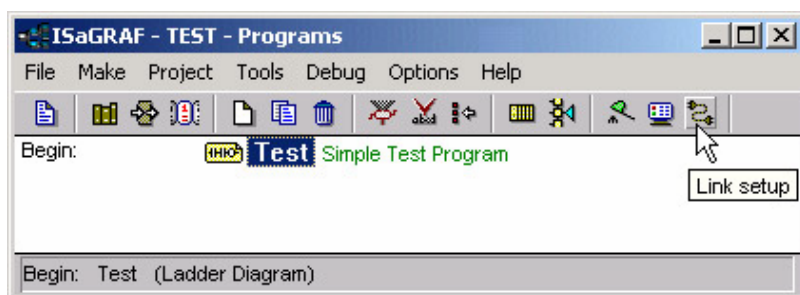


Before you can download an ISaGRAF application to the I-8437/8837 controller system using the Ethernet port, you must first setup the Ethernet port to properly communicate with the host PC.

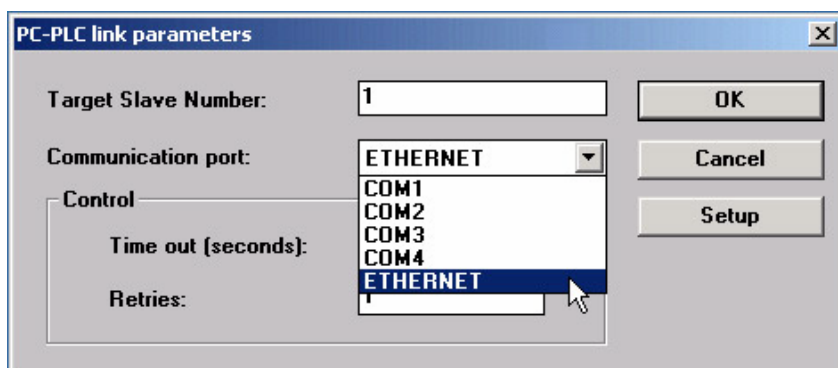
On the I-8437/8837, iPAC-8447/8847, Set IP, Mask and Gateway address:  
Refer to **Appendix B** or CD\_ROM:\NAPDOS\ISaGRAF\8000\driver\setip.txt

On your PC:

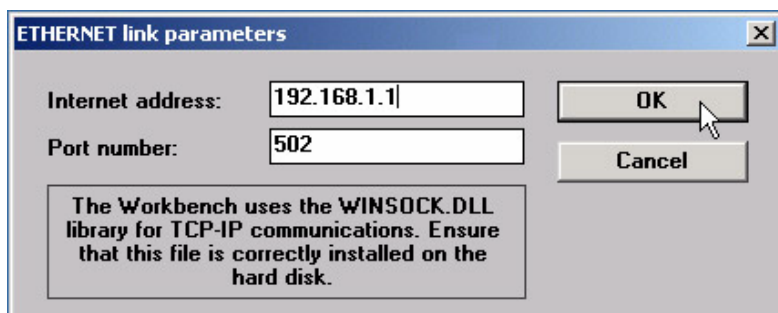
First open an ISaGRAF project and select a program you wish to communicate between your PC and the I-8437/8837 controller system. Next, select the "Link Setup" button on the project screen as shown below.



A "PC-PLC Link Parameters" dialog box will appear as shown below. From here select the "Ethernet" communications option and click on the "Setup" button.



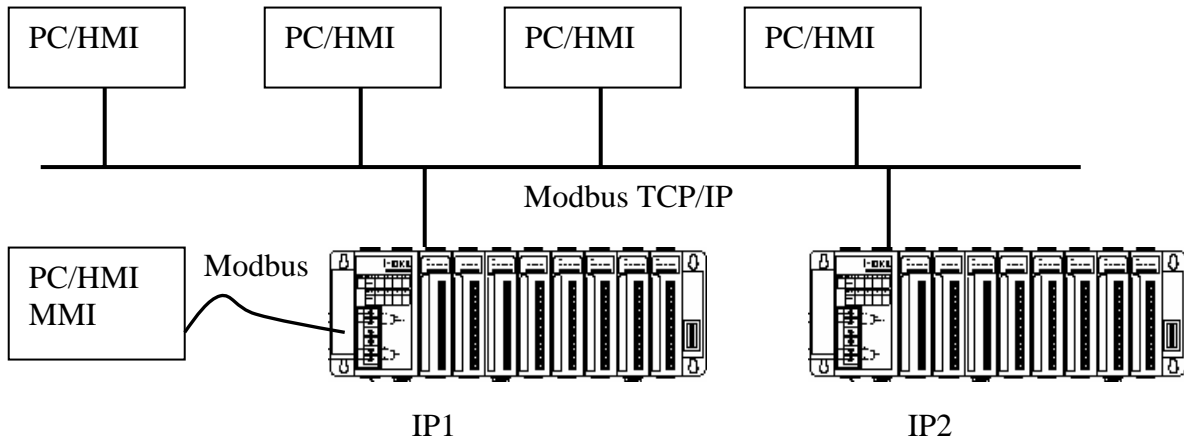
Once you have clicked on the "Setup" button, an "Ethernet Link Parameters" dialog box will appear. Set the "Port Number" to "502" and enter in the **Internet address (IP) of the I-8437/8837** controller.



Once you have entered the appropriate information, click on the "OK" button, and now you have configured your PC to communicate with the I-8437/8837 through the Ethernet port.

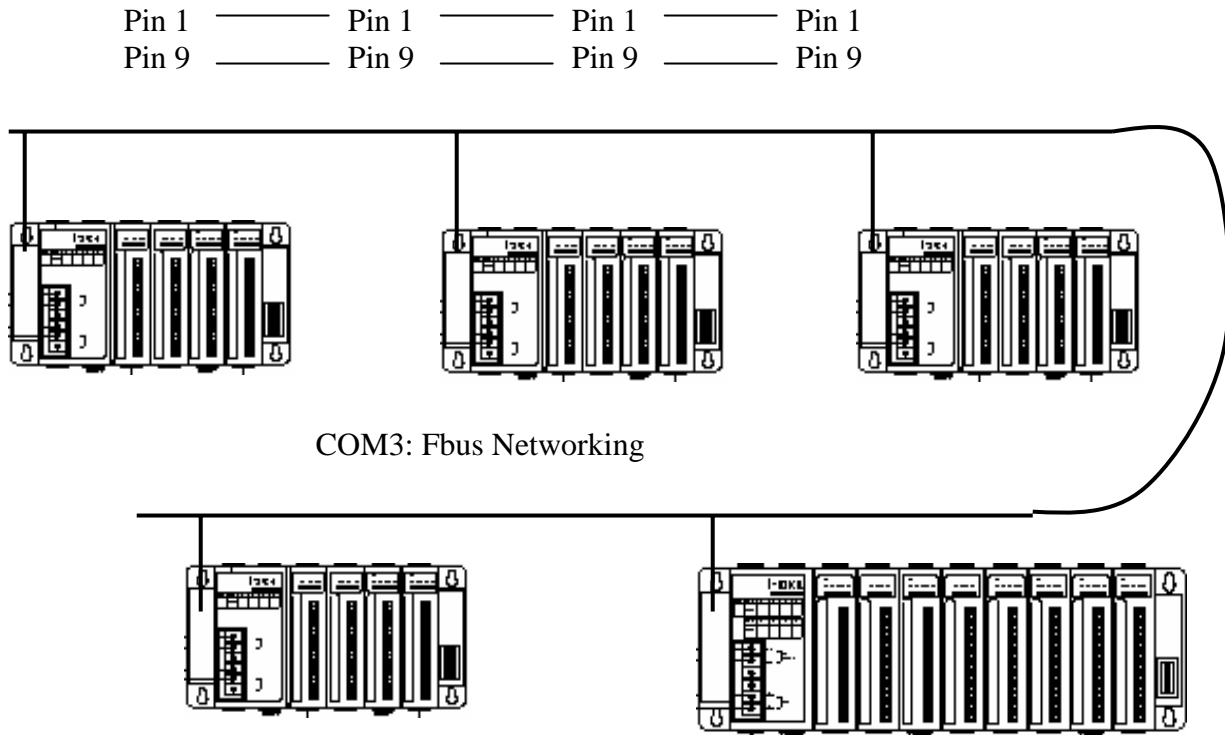
### 1.3.9: Multi-Clients Connection to The I-8437/8837 Ethernet Port

Each I-8437 / 8837 & i-8437-80 / 8837-80 has an IP address and with a fixed Ethernet port No. **502**. Up to 4 PCs can link to one I-8437 / 8837 throughout Ethernet (Modbus TCP/IP protocol). Two other PC or HMI can link to COM1: RS232 port or COM3:RS-232/485 ( refer to 1.3.6 ) by Modbus slave protocol in the I-8437/8837 & i-8437-80 / 8837-80. Therefore the maximum number of clients can be linked is 6.



## 1.4: Controller to Controller Data Exchange: Fbus

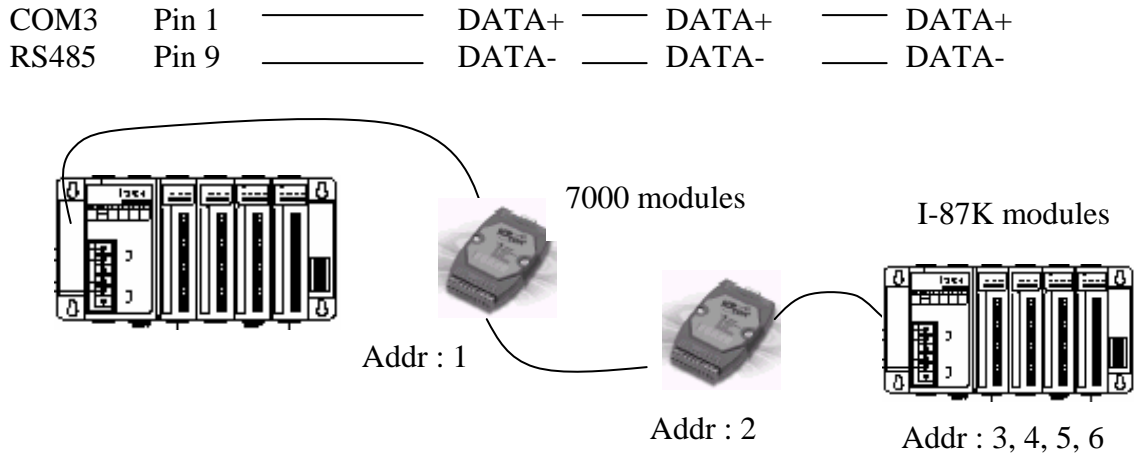
If your controllers have Ethernet port, you may use one another way – Ebus (please refer to section 7.5) Connect all COM3's Pin 1 together and Pin 9 together and then one of these I-8xx7 controllers should set its **JP1 and JP2 of the power board to position "1 to 2"** (refer to section 1.3.5). The maximum distance for the Fbus data exchange network is 1200 meters (4,000 feet) depending on the communication baud rate. The distance between the PC and the I-8xx7 controller system is dependent on the baud rate; the rule to follow is the lower you set the baud rate, the longer the distance can be.



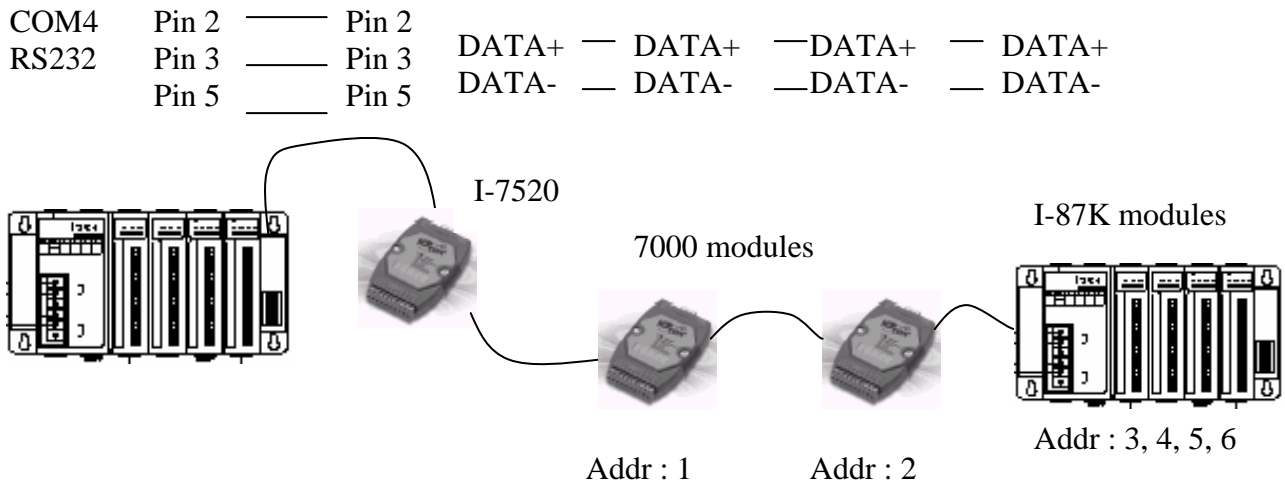
## 1.5: Linking I-7000 and I-87K Modules For Remote I/O

The I-8xx7 controller system can use one of its COM3 or COM4 ports, while COM2 or COM3 for I-7188EG/XG, to link to ICP DAS's "I-7000" and "I-87K" series of remote I/O modules. This configuration can be very useful in applications that require distributed remote I/O throughout the system. (RU-87P4 / RU-87PN is better than I-87K4/5/8/9. Please visit below web site

[http://www.icpdas.com/products/io\\_expansion\\_unit/ru-87p/ru-87p4.htm](http://www.icpdas.com/products/io_expansion_unit/ru-87p/ru-87p4.htm))



If you choose to utilize the COM4 port, connect the COM4 port to the I-7520's RS-232 port, and also connect the "DATA+" to the "DATA+" signal, and the "DATA-" to the "DATA-" signal as shown below.



You can link up to 64 I-7000 or I-87K series remote modules to one I-8xx7 controller system. You must remember to set each I-7000 and I-87K remote module must have a unique address, and be set to the same baud rate as the I-8xx7 controller system.

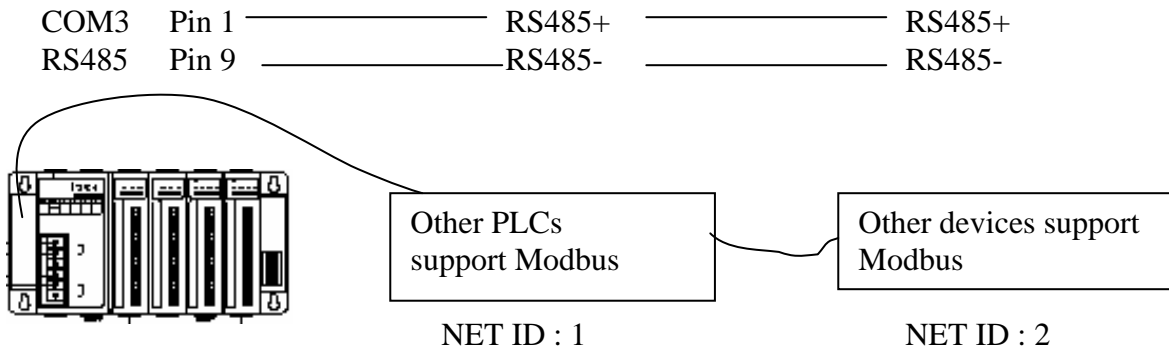
For more information regarding setting up and programming an I-7000 / I-87K remote module, please refer to Chapter 6 - "Linking To I-7000 and I-87K Modules".

## 1.6: Creating A Modbus Link With The I-8xx7 Controller

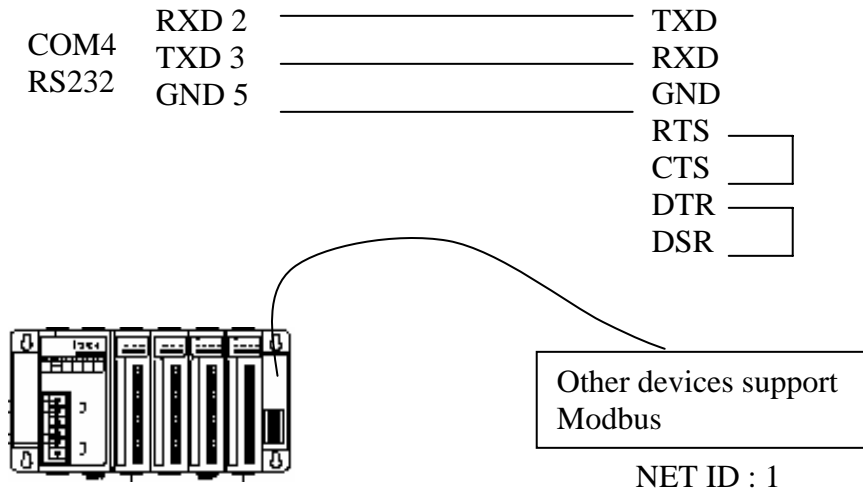
The I-8xx7 controller system can be a Modbus "Slave" and/or a Modbus "Master" controller depending on the application. Through this method you can use the COM1 and COM2 ports of the I-8xx7 controller system to link to a PC or other HMI products. In this type of configuration, the I-8xx7 controller system becomes a Modbus slave controller system. For more information about setting up and programming for Modbus slave, please refer to Chapter 4 – "Linking The I-8xx7 To An HMI Program".

If COM3 or COM4 is used to link to other devices that support the Modbus protocol, the I-8xx7 controller system will be the Modbus master controller. For more information about setting up and programming for Modbus master, please refer to Chapter 8 - "Linking To A Modbus RTU Or Other Devices".

If the COM3:RS485 port is used for Modbus master, one I-8xx7 can connect to many other devices. Each device on the link must have a unique NET ID (1 ~ 255) address, and communicate at same baud rate settings.

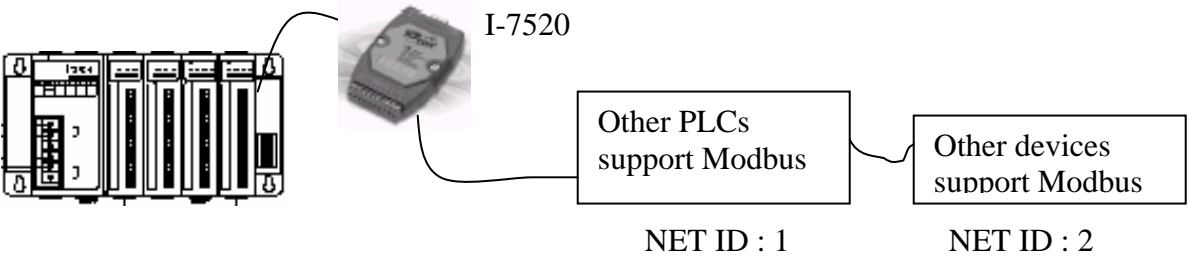
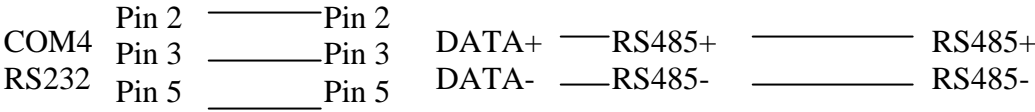


If COM4 is used, you can only link one I-8xx7 to one other Modbus device.





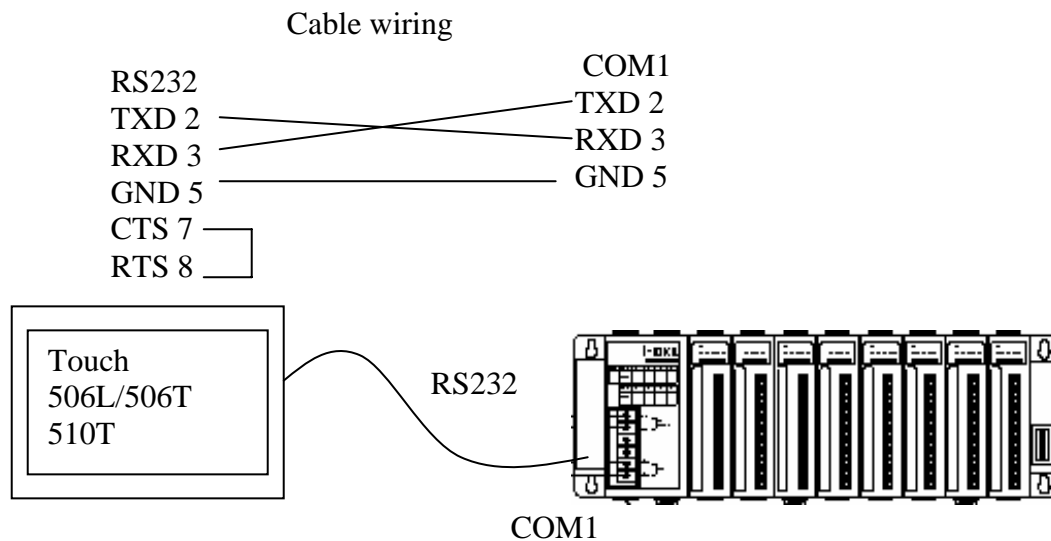
If the COM4 port of the I-8xx7 controller system is used to connect to one I-7520 remote device, then the I-8xx7 controller can network to numerous Modbus devices.



## 1.7: Linking To An HMI Interface Device

The COM1 (RS-232) & COM2 (RS-485) ports of the I-8417/8817 and COM1 (RS-232) & COM3 (RS-232/485, section 1.3.6) ports of the I-8437/8837 & i-8437-80 / 8837-80 controller can be used to interface with additional Man Machine Interface (MMI) devices such as touch screen displays. ICP DAS provides a full line of touch screen displays, such as the "Touch" series screens. The models in the product line include the Touch 506L/506T and Touch 510T HMI products.

If you are using any of the "Touch" series of MMI devices to connect to an I-8xx7 controller, you can only interface the devices to the COM1 port on the I-8xx7 controller.



For more information regarding interfacing the Touch series of HMI devices to the I-8xx7 controller system, please refer to Chapter 4- "Linking The I-8xx7 To HMI Devices".

## 1.8: Using N-Port COM

There are some N-Port COM boards that can be used to extend communication ability of the I-8xx7 controller. The model No. available are as below.

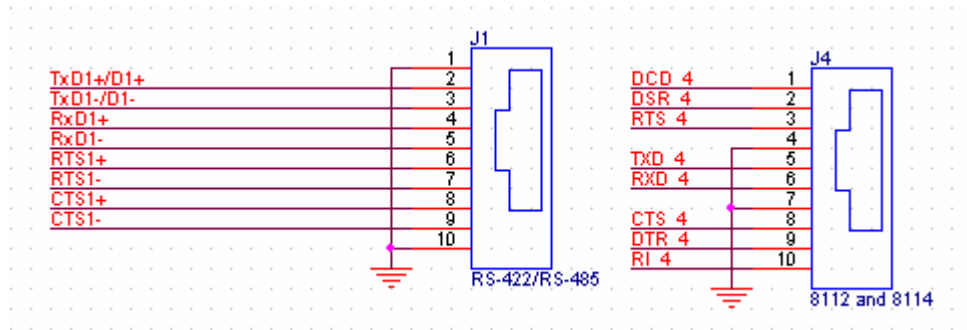
I-8112:2-channel RS232 Module  
I-8114:4-channel RS232 Module  
I-8142:2-channel RS422/485 Module  
I-8144:4-channel RS422/485 Module

Note:

These N-Port COM boards can only be plugged into slot 0 to slot 3. It doesn't support slot 4 to slot 7. That means user can use only Com5 to Com20 of N-Port COM boards.

Some functions can be used to read/write these COM ports. Please refer to Appendix A.4 for "COMOPEN", "COMOPEN2", "COMCLOSE", "COMREADY", "COMARY\_R", "COMARY\_W", "COMREAD", "COMSTR\_W", "COMWRITE" and "COMCLEAR".

Pin assignment:



\*\*\*\*\*

1. To use Wincon-8xx7 's expansion COM5 to COM14, please refer to Appendix E of its "Getting started Manual" delivered with the Wincon-8xx7 product. Or  
W-8xx7 CD-ROM:\napdos\isagraf\wincon\english\_manu\ "getting\_started\_w8337.pdf"

2. To use I-7188EG / 7188XG 's expansion COM3 to COM8, please purchase one X-5xx board and then plug it inside the I-7188EG/XG. [http://www.icpdas.com/products/PAC/i-o\\_expansion/x\\_list.htm](http://www.icpdas.com/products/PAC/i-o_expansion/x_list.htm)

\*\*\*\*\*

## 1.9: Using more Modbus RTU Slave ports

---

W-8337/8737, W-8347/8747 can support max. 5 communication ports as Modbus RTU Slave ports. Please refer Appendix G, E and A.2 of its “Getting Started Manual” or  
W-8xx7 CD-ROM: \napdos\isagraf\wincon\english\_manu\

I-7188EG/XG can support max. 2 communication ports as Modbus RTU Slave ports. Please refer Chapter 3.7 and 3.6 of its “Getting Started Manual” or  
I-8000 CD-ROM: \napdos\isagraf\7188eg\chinese\_manu\

I-8437-80 / 8837-80 can support max. 2 communication port as Modbus RTU Slave ports. Please refer Chapter 3.5 and 3.10 of its “Getting Started Manual” or  
I-8000 CD-ROM: \napdos\isagraf\8000\chinese\_manu\

You may also download manuals at  
[http://www.icpdas.com/products/PAC/i-8000/getting\\_started\\_manual.htm](http://www.icpdas.com/products/PAC/i-8000/getting_started_manual.htm)

---

## Chapter 2. Getting Started

---

For more extensive information regarding all of the capabilities of the ISaGRAF programming system, please refer to **Appendix E: “Language Reference”** of this manual or the **“ISaGRAF USER’S GUIDE”** manual which can be found from the CD\_ROM of the ISaGRAF workbench. Its file name is either “ISaGRAF.pdf” or “ISaGRAF.doc”.

This manual provides some program examples and its description, please refer to Chapter 11.

### NOTE:

The I-8xx7 abbreviation is for the I-8417, I-8437, I-8817 and I-8837 controllers, while W-8xx7 is the abbreviation for the Wincon-8037/8337/8737 and Wincon-8047/8347/8747 controller.

## 2.1: A Simple Ladder Logic (LD) Program

---

### Ladder Logic Basics

"Ladder Logic" programming (LD) is a graphical representation of Boolean equations, combining **contacts** (input arguments) and **coils** (output results). Ladder Logic most closely resembles the electrical schematics that an electrician or technician may use to diagnose and troubleshoot an industrial process controller system.

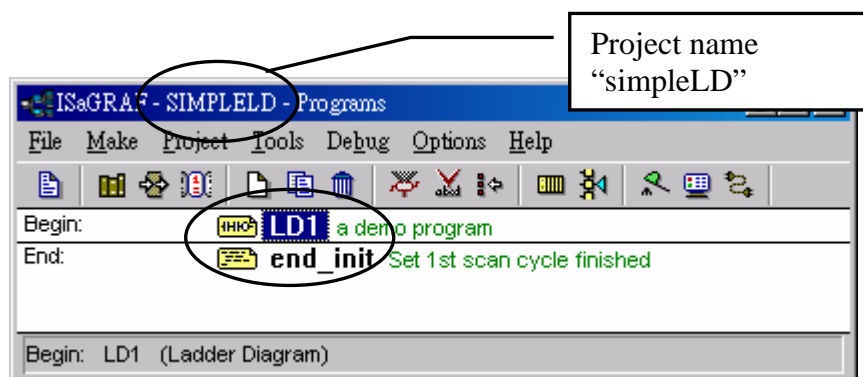
The LD language enables the programmer to describe the conditions and modifications to Boolean data by placing "graphical symbols" to represent hardware devices used in a process control application.

### A Simple Ladder Example Program

The following is a step-by-step example on how to create a ladder logic (hence forth referred as "LD") program using the ISaGRAF Workbench software program provided with the ISaGRAF controller system.

We will create one another Structured Text (hence forth referred as “ST”) program to indicate the first PLC scan cycle. That means in this example ISaGRAF project, we have two programs inside it. One is written in LD and the other is written in ST.

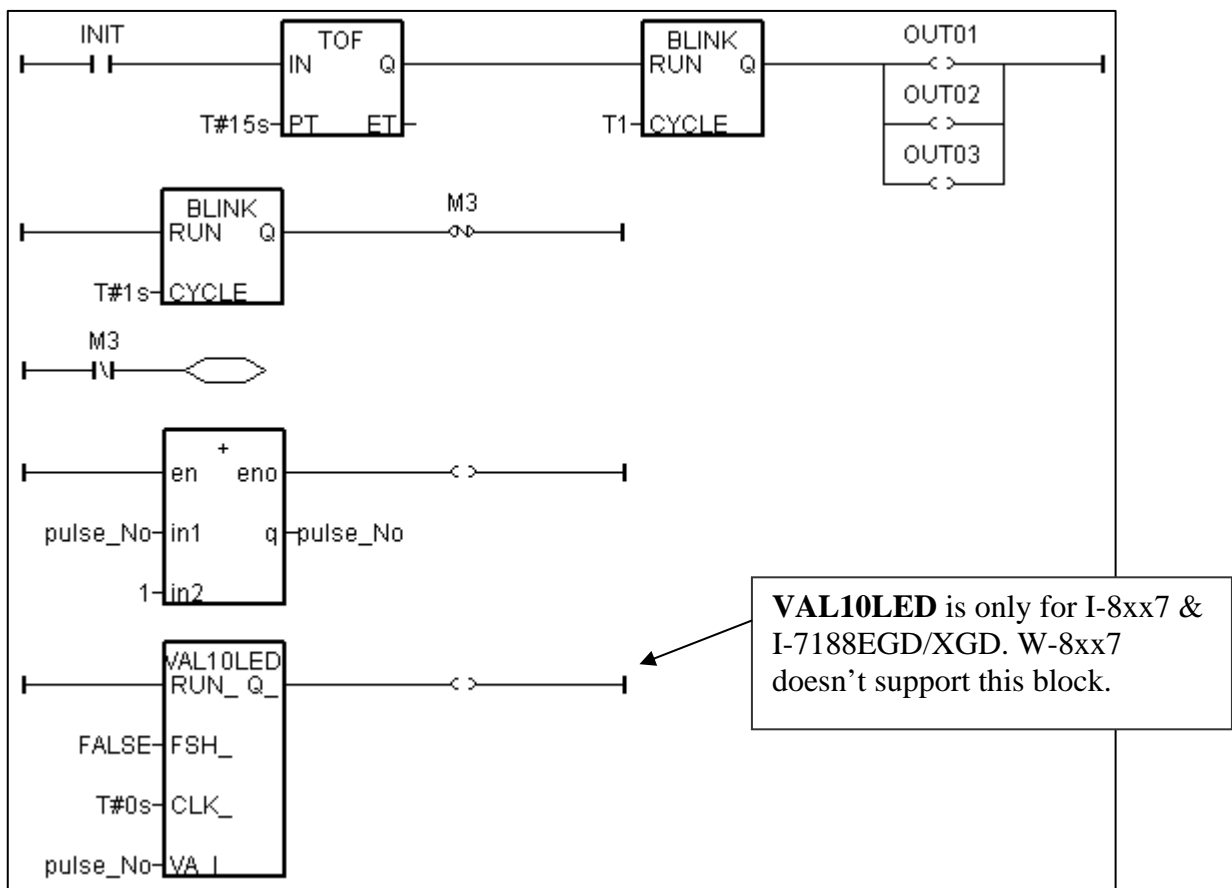
The example project name is “simpleLD”. The name of the LD program of this example project is “LD1” and “end\_init” is the name of the ST program .



Variables Used In The Example LD Program:

| Name     | Type    | Attribute       | Description  |
|----------|---------|-----------------|--|
| INIT     | Boolean | <b>Internal</b> | <b>initial value at “TRUE”</b> . TRUE means 1 <sup>st</sup> scan cycle       |
| M3       | Boolean | <b>Internal</b> | Indicate a pulse is generated or not.  |
| OUT01    | Boolean | <b>Output</b>   | Output 1   |
| OUT02    | Boolean | <b>Output</b>   | Output 2   |
| OUT03    | Boolean | <b>Output</b>   | Output 3   |
| T1       | Timer   | <b>Internal</b> | Time Period of blinking, <b>initial value is set at "T#1s"</b>               |
| Pulse_No | Integer | <b>Internal</b> | To puls one when M3 pulse is generated<br><b>initial value is set at "0"</b> |

Ladder Logic Program “LD1” Outline:



ST program “end\_init” Outline:

```
INIT := FALSE ;
```

### Process Operation Actions:

Ladder Logic Program “LD1” :

Blink Outputs 1, 2, & 3 with a period of “T1” in the first 15 seconds, “T1” has initial value equal to 1 second. After these 15 seconds, Outputs 1, 2, & 3 will be turned OFF.

Generate a pulse output every 1 second to the internal boolean variable “M3”.

Plus integer variable “pulse\_No” by 1 every time when “M3” pulse is generated.

Display the value of “pulse\_No” to the 7-Seg leds of the I-8xx7 or I-7188EG/XG controller.

ST Program “end\_init” :

Set boolean variable “INIT” to FALSE at the end of the PLC scan cycle. So that “INIT” will be TRUE only at the first scan cycle.

### Description of block and some basic LD item:

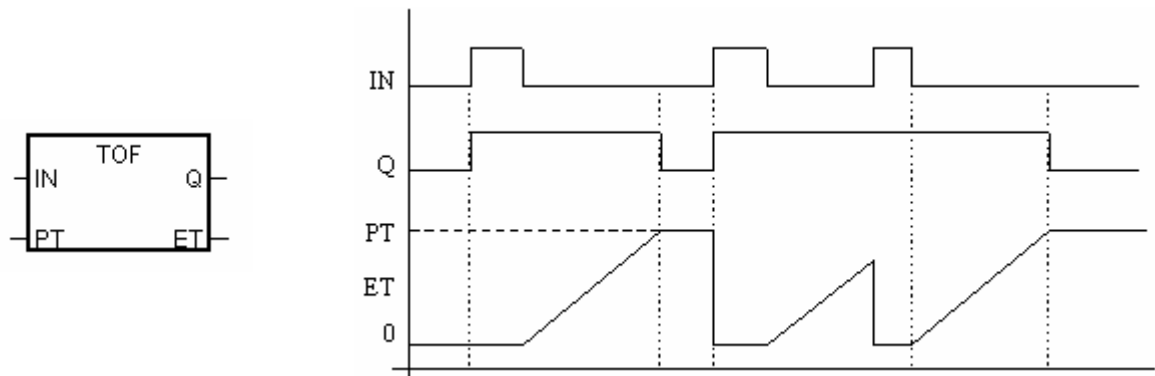
TOF: To turn off a boolean however delay a time of “PT”.

“IN” is a boolean parameter, if falling from TRUE to FALSE. The timer ticks from 0 to “PT”

“PT” is a timer parameter, it defines the delay time of output.

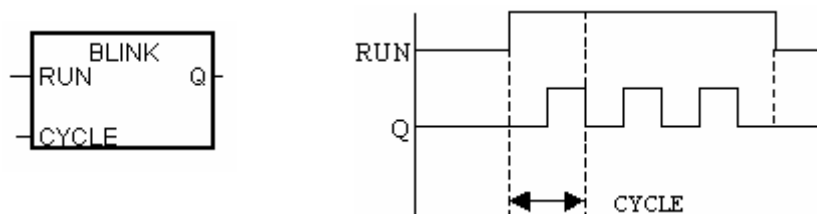
“Q” is the boolean output of this block. It will be turned OFF when “PT” is reached.

“ET” is the timer output of this block. (We don’t use it in this example)



BLINK: To blink a boolean with a period of “CYCLE”.

“RUN” is a boolean parameter, if it is TRUE, the boolean output “Q” will be blinking at period of the timer parameter “CYCLE”.



VAL10LED: Display a interger value to the 7-Seg leds of the controller.

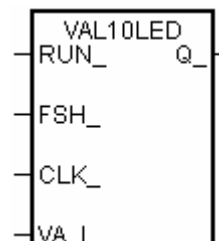
Only for I-8xx7, I-7188EGD & I-7188XGD.

“RUN\_” is a boolean parameter. TRUE to display.

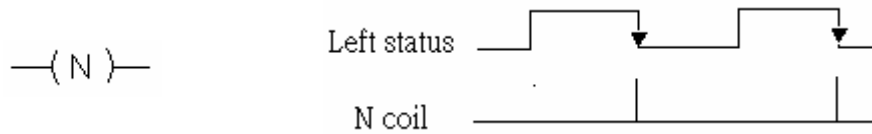
“FSH\_” is a boolean parameter. TRUE to blink the display.

“CLK\_” is a timer parameter. It defines the blinking period.

“VA\_I\_” is the integer to display.



“N” coil : Coil with N type means it will be set to a pulse TRUE when the left status is just falling from TRUE to FALSE.



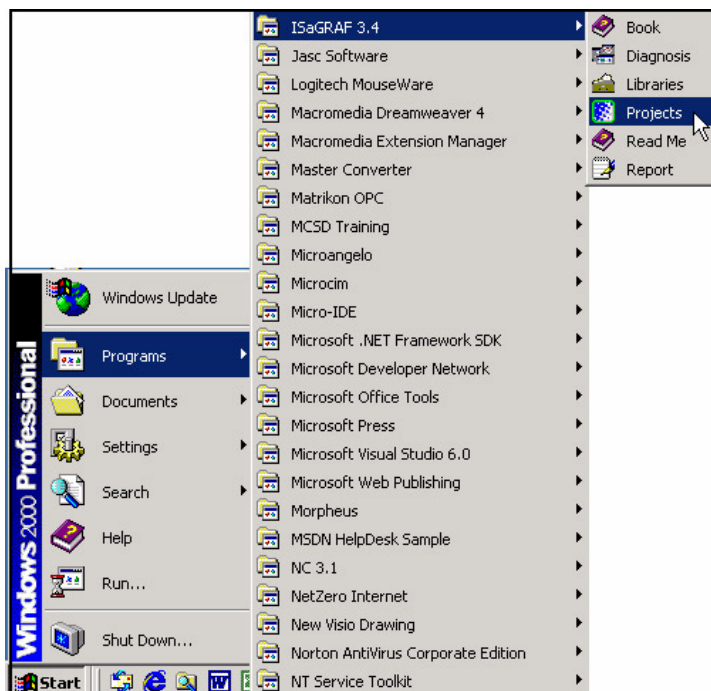
“Retrun” : To return from the excution if the left status is TRUE, that is, the reset LD rungs of the program below this “return” will not excute when the left status is TRUE.



### 2.1.1: Programming LD

Starting & Running The ISaGRAF Workbench Program

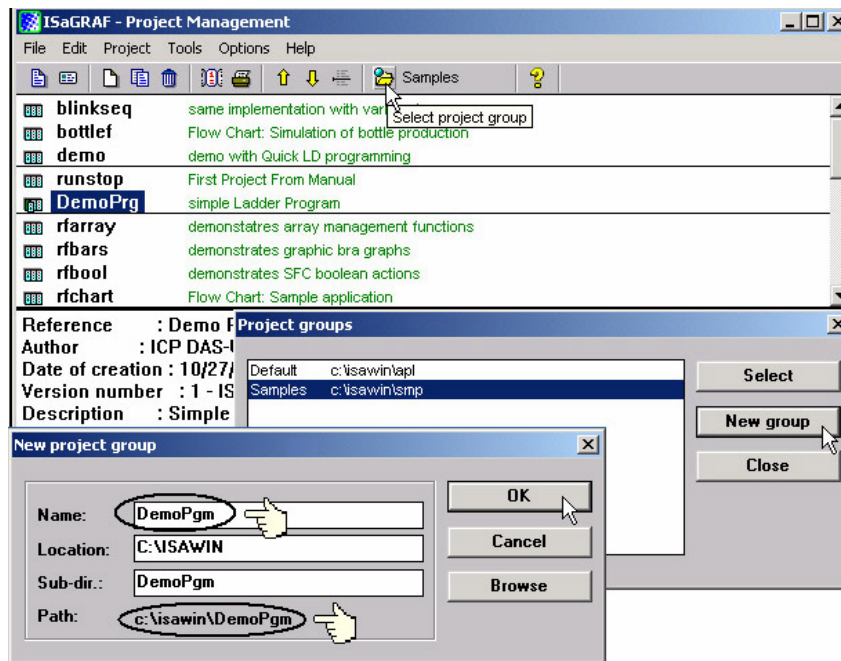
Click on the Windows "Start" button, then click on "Programs", then click on "ISaGRAF 3.4", then click on "Projects" as shown below.





### 2.1.1.1: 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".

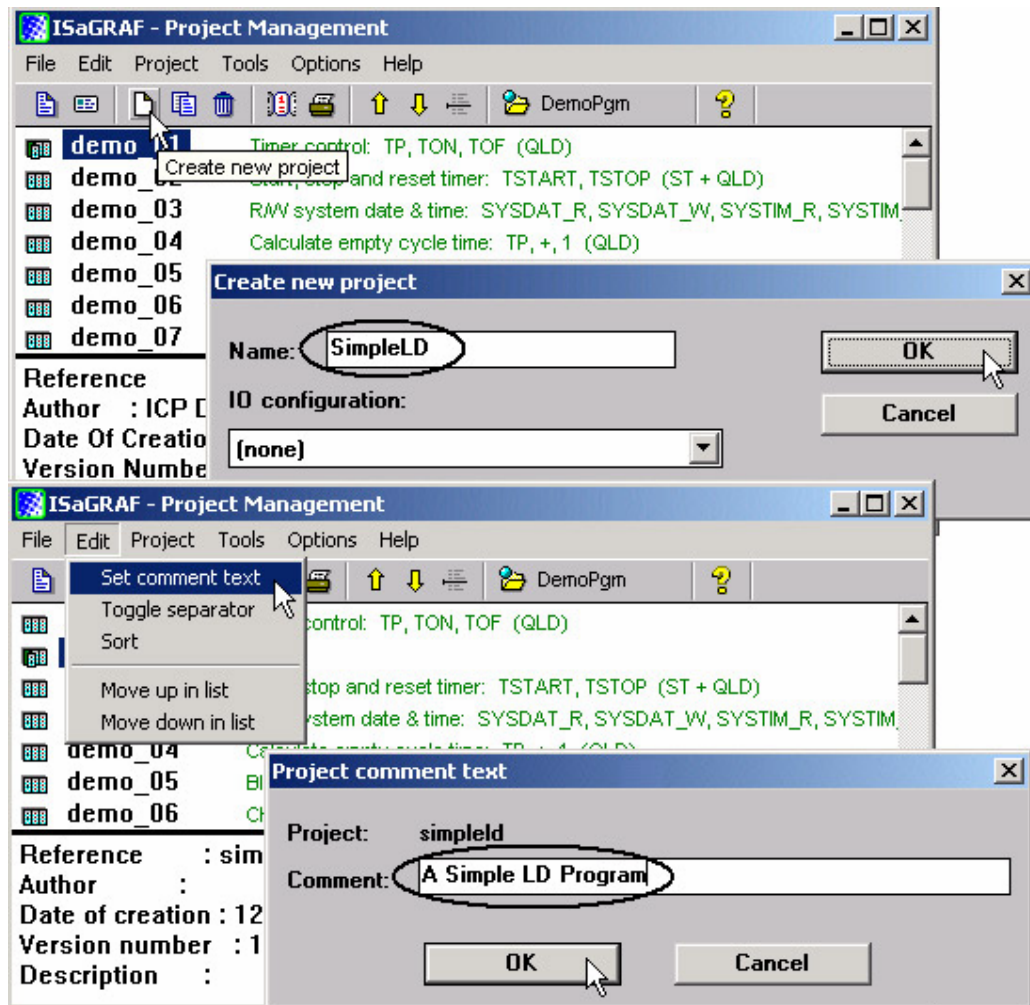


Note that the name that you give the "New Project Group" also creates a new sub-directory corresponding to the project group name in the "c:\isawin" sub-directory.

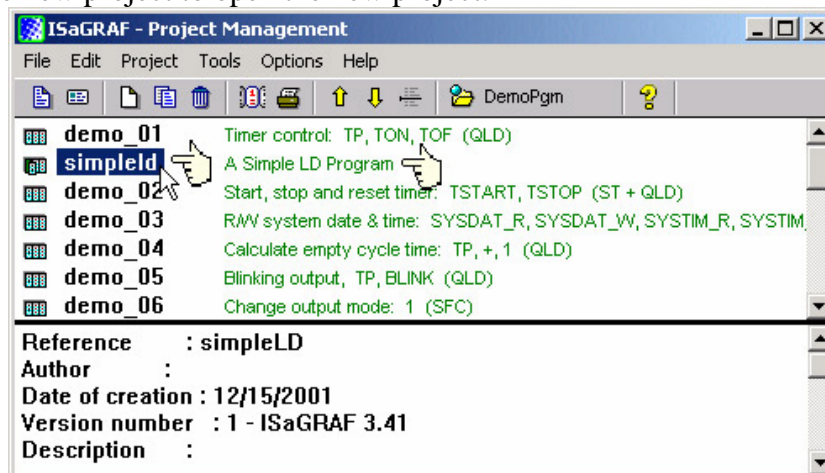
To get into the new project group, either double click on the new group name, or click on the new group name (the name will be highlighted) to select the new project group and click on the "Select" button.

### 2.1.1.2: 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.

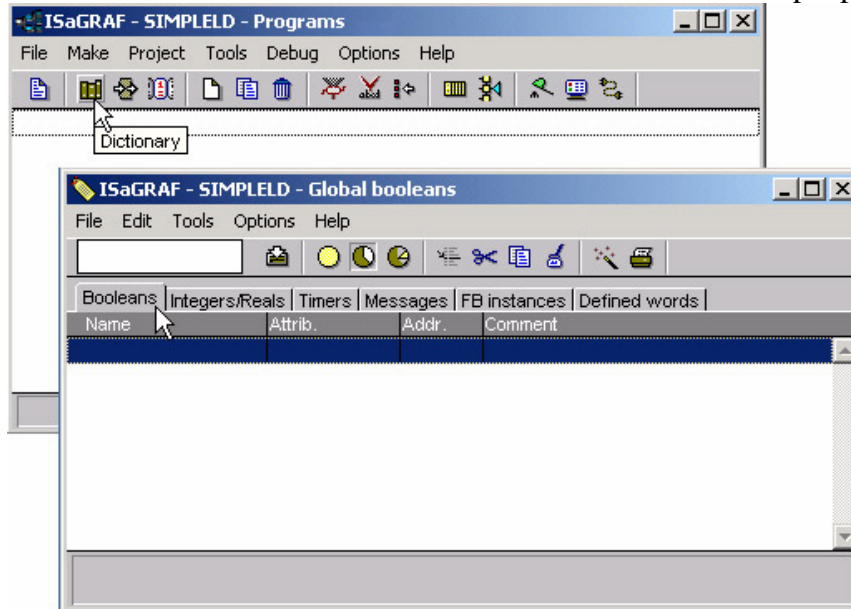


You will now see the name of the new project in the "Project Management" window. Double click on the name of the new project to open the new project.

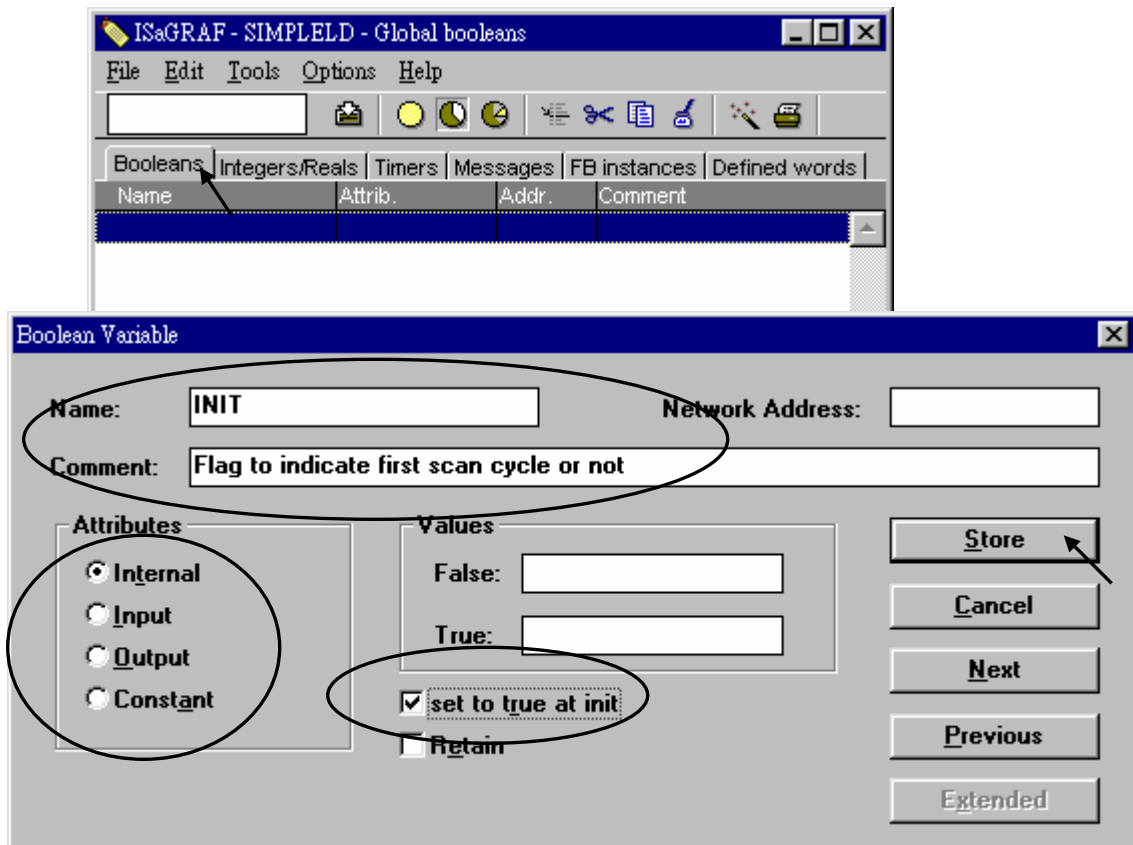


### 2.1.1.3: Declaring The ISaGRAF Project Variables

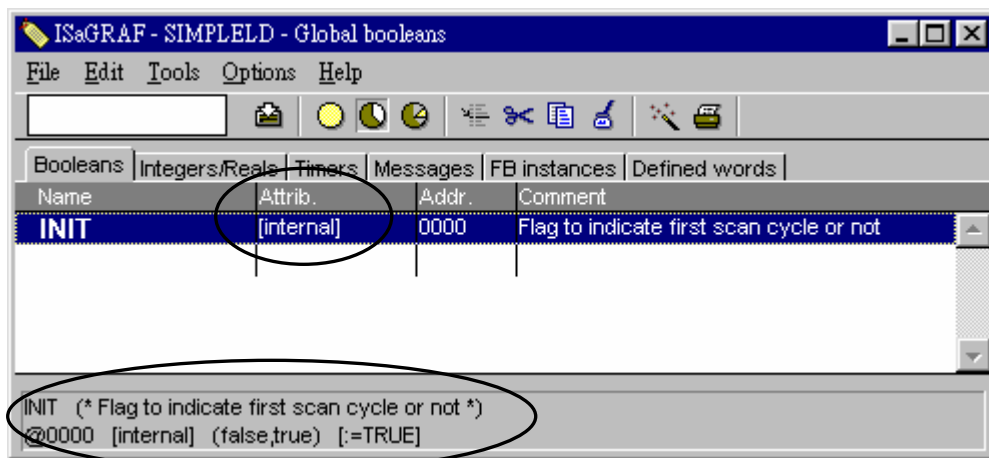
Before you can start creating an ISaGRAF program, you must first declare the variables that will be used in the ISaGRAF program. To begin this process, first click on the "Dictionary" icon and then click on the "Boolean" tab to declare the Boolean variables that will be used in our example program.



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 "INIT", and "Flag to indicate first scan cycle or not" 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, INIT's attribute will be an "Internal". Lastly, check on the "set to true at init" since we need INIT has its initial value as TRUE when the project is just power up to run. Then press the "Store" button to save the Boolean variable that has been created.

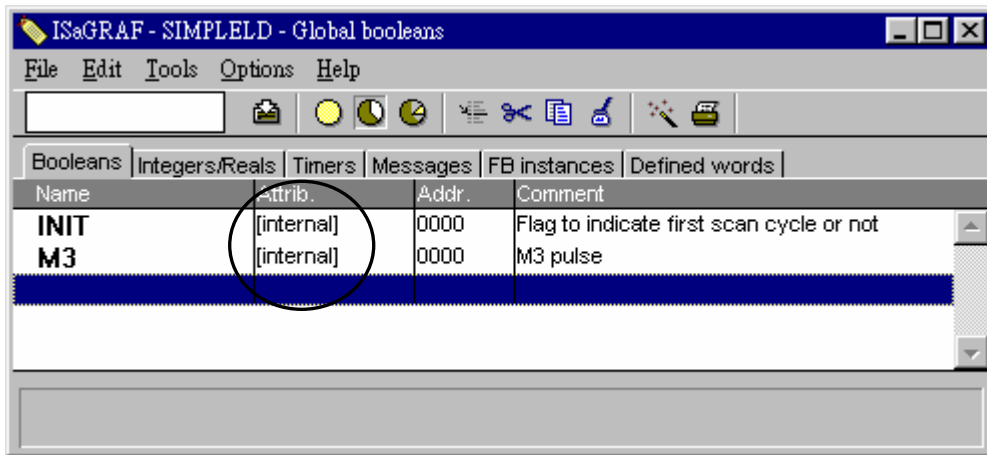


The new Boolean variable has now been declared. Note the other information areas that are provided for the programmer to fully explain how the variable will be handled.

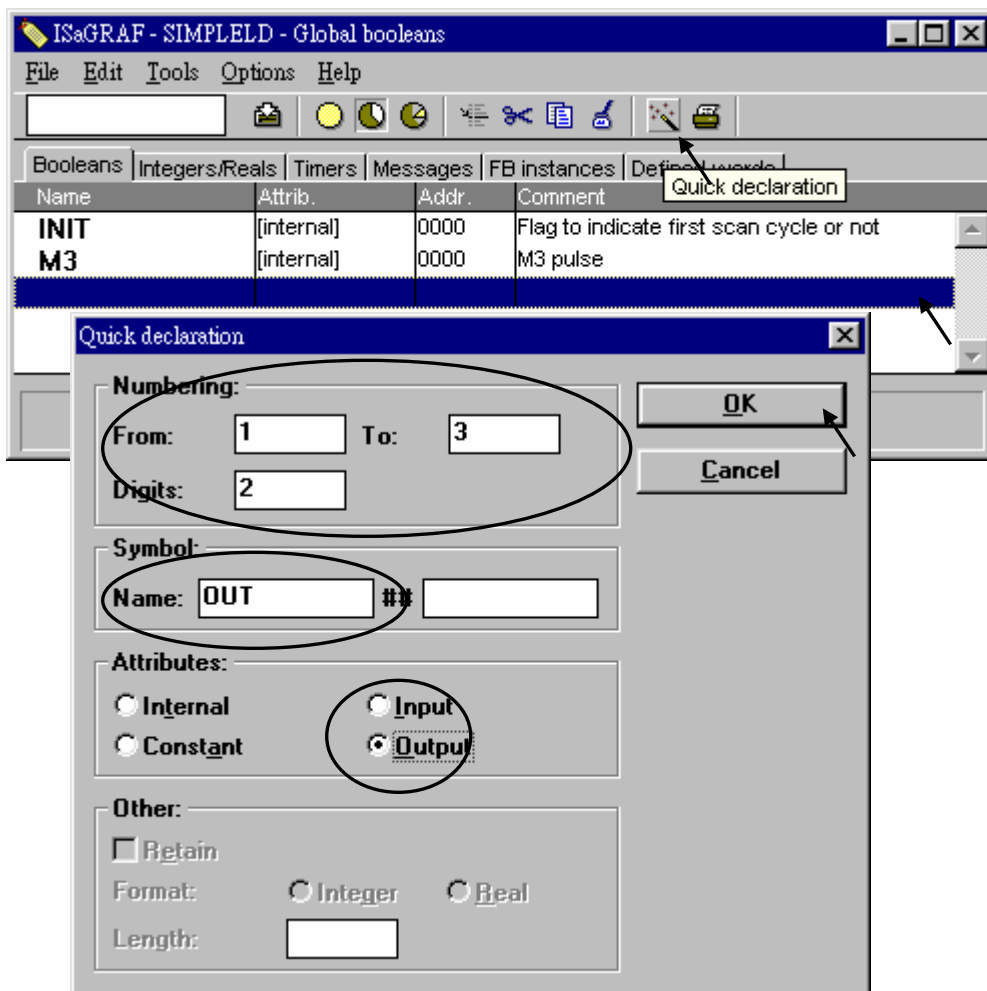


**NOTE:** You MUST make sure that the variable you have declared has the desired Attribute assigned. If you decide that you want to change a project variable's attribute, just double click on the variable name and you can reassign the attribute for the variable.

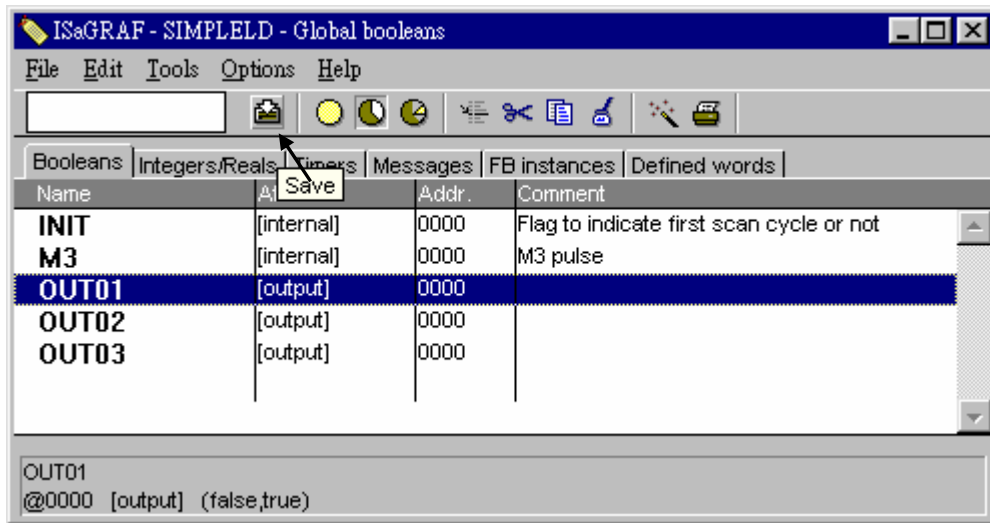
Using the same method described above, declare the additional Boolean variables for this example program, "M3". When you have completed the Boolean variable assignments, the Global Boolean window should look like the example below.



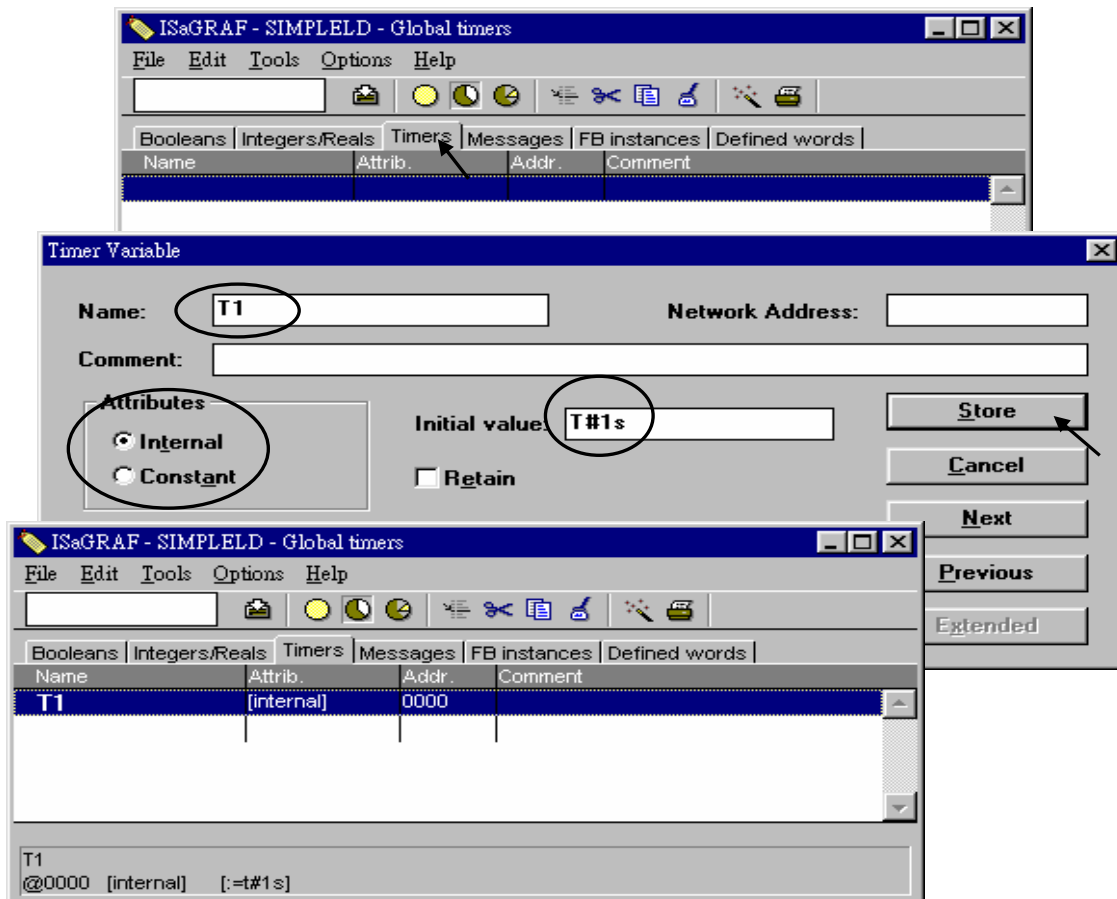
There are three outputs used in this example program named "OUT01, OUT02, and OUT03". ISaGRAF provides a quick and easy way to declare variables that are sequentially ordered. To begin this process, click on the "Quick Declaration" icon, and enter in the output number that you will start with in the "Numbering" from and "To" field (this example uses from 1 to 3). Enter the "Symbol" name for the output variables being declared, and lastly, set the attribute to "Output".



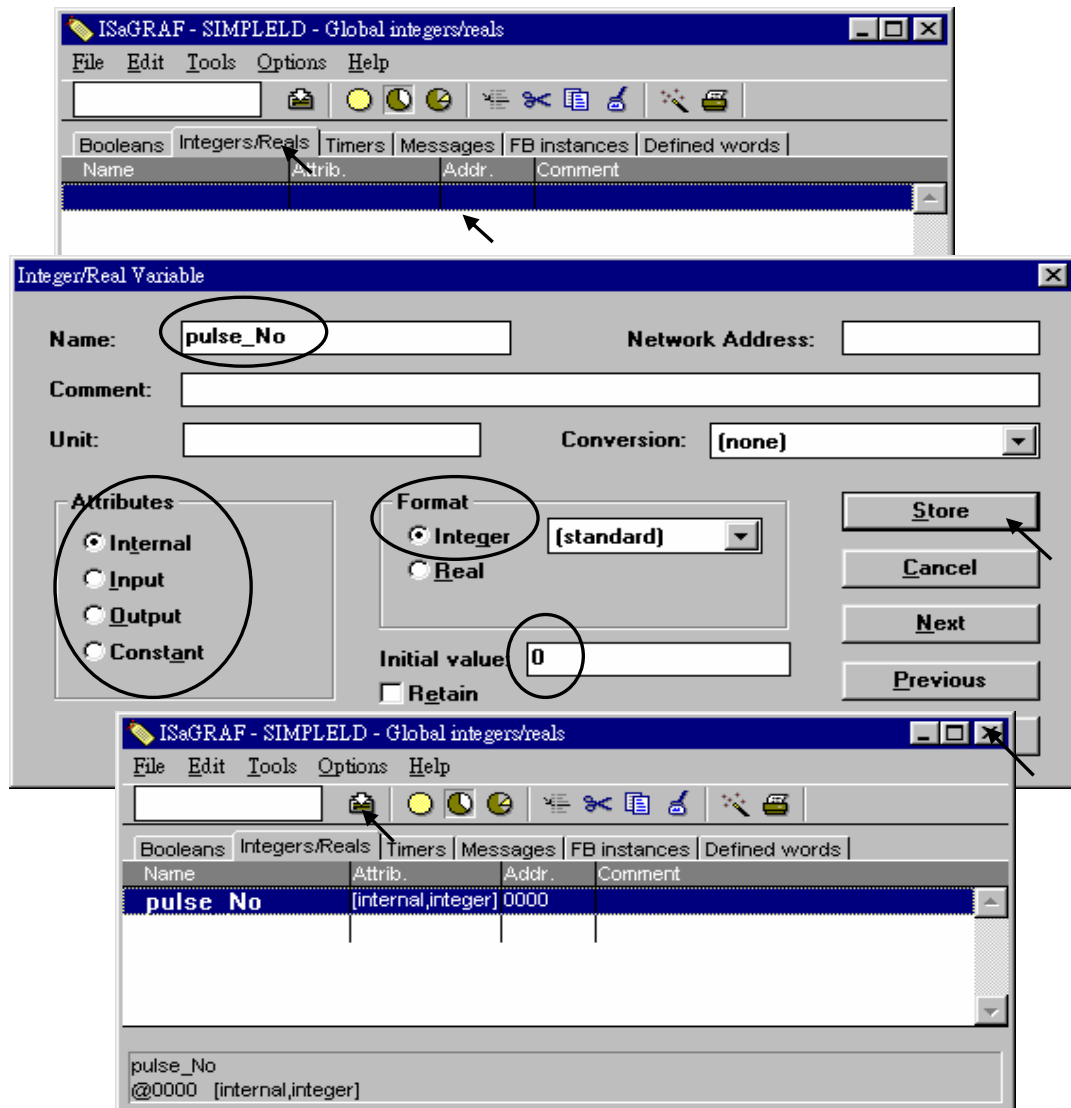
When you click on the "OK" button, all three outputs will be immediately added to the "Global Boolean" window.



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#1s", then click on the "Store" button.



To declare the Integer (pulse\_No) variable used in this example program, click on the "Integers/Reals" tab in the setup screen. Double click on the colored area and enter the Name as "pulse\_No", set the "Attributes" to "Internal", the "Format" to "Integer", and the "Initial Value" to "0", then click on the "Store" button.



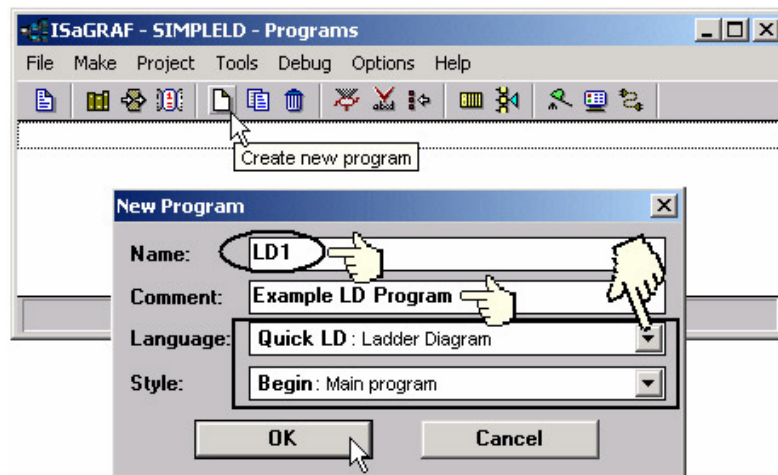
Once all of the variable characteristics have been properly setup, click on "save" and then click on "X" at the top right of the setup window to close the variable dictionary for this example project.



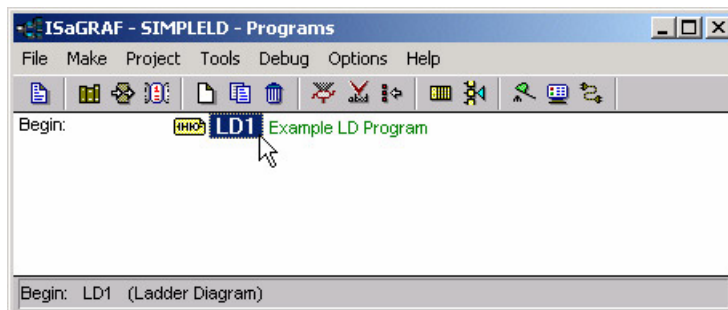
#### 2.1.1.4: Creating The Example LD Program

Once all of the variables have been properly declared, you are now ready to create the example LD program. To start this process, click on the "Create New Program" icon and the "New Program" window will appear.

Enter the "Name" as "LD1" (the name of our example program), 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.



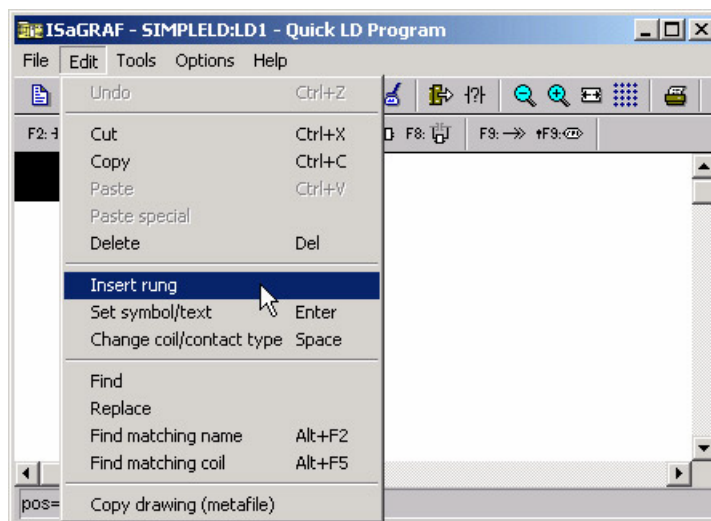
The "LD1" program has now been created. To open the "LD1" program, double click on the "LD1" name.



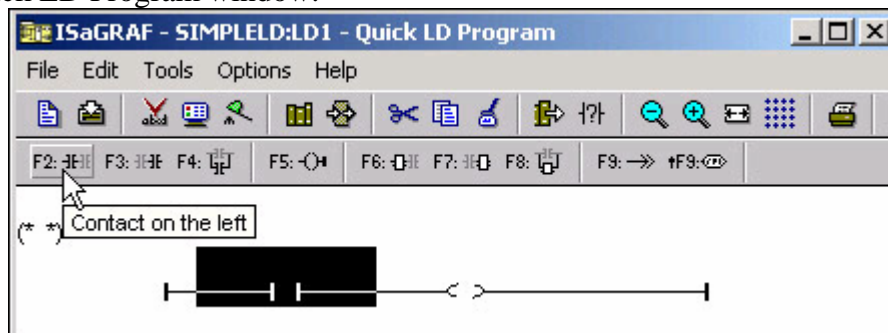
#### 2.1.1.5: Editing The Example "LD1" Program

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" as shown below. "Insert Rung" means to insert a basic LD rung just above the current position.

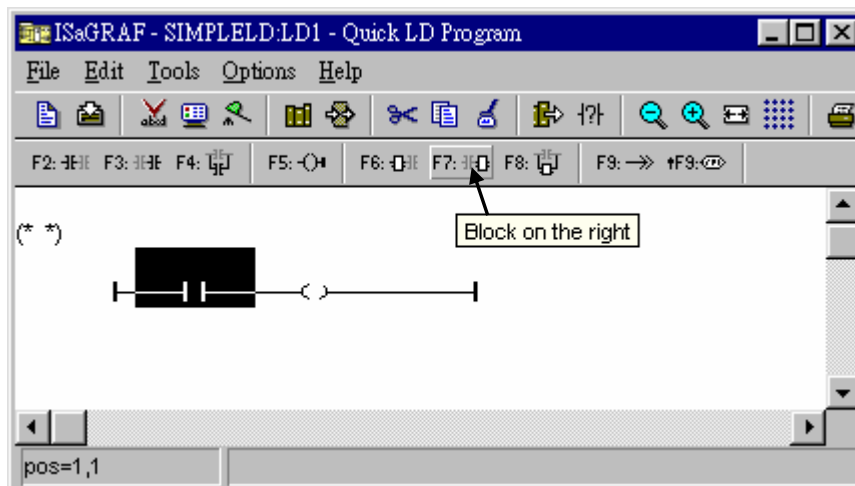




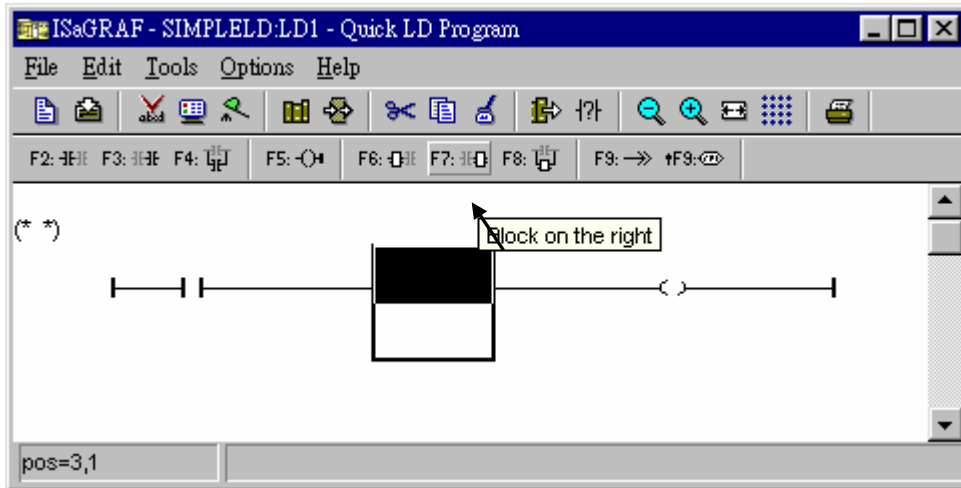
Or, you may just simply click on the "F2 (Contact On The Left)" icon, and the following will appear within the Quick LD Program window.



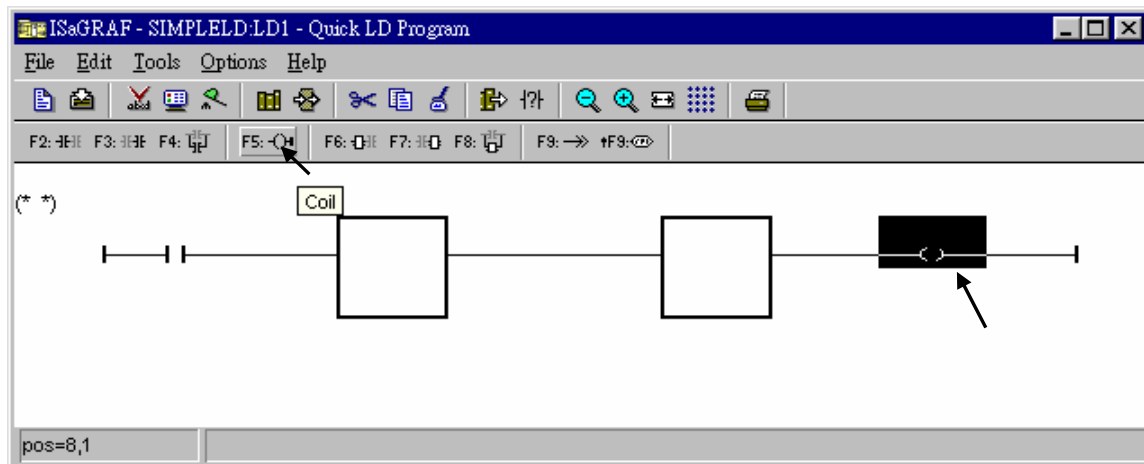
Click on the "F7 (Block on the right)" icon and you will create a block on the right of the first input contact.



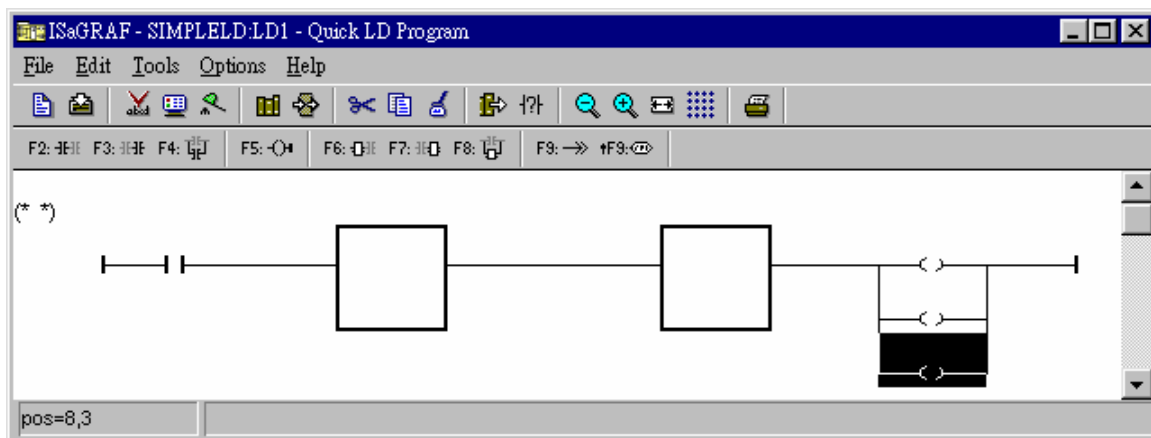
Click on "F7 (Block on the right)" icon again to create one another block on the right of the first block.



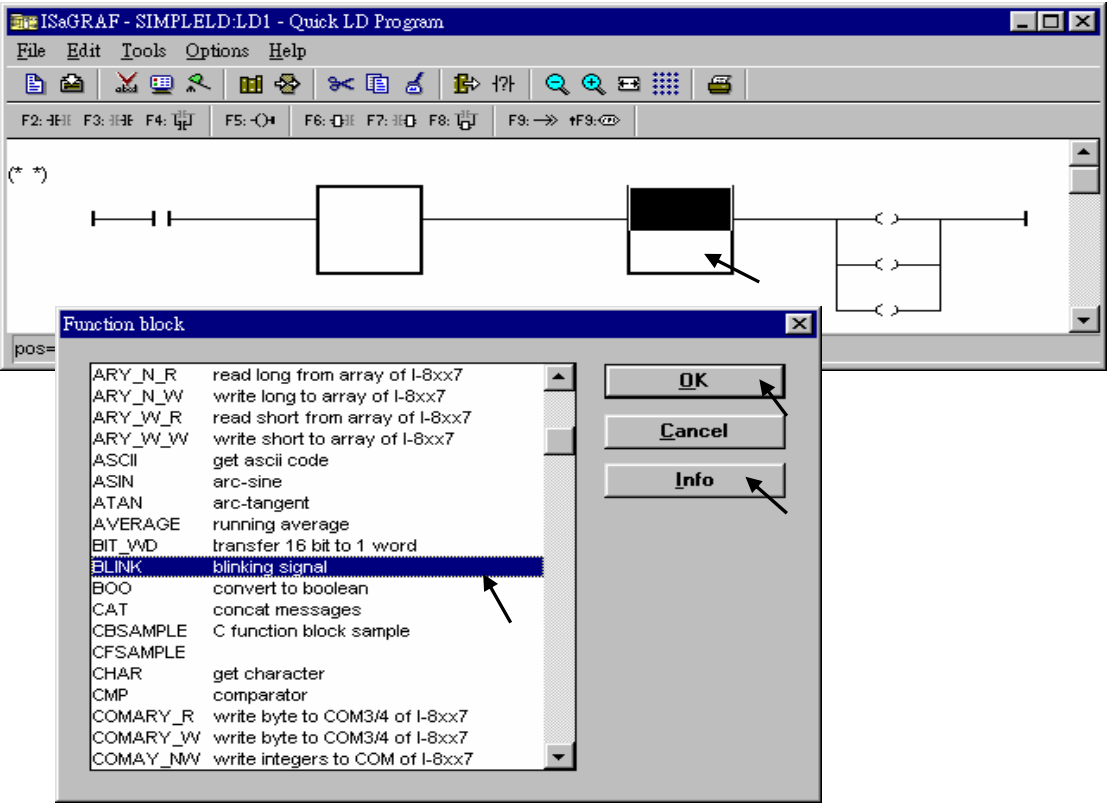
Then you will get the window as below. Move the cursor to the Coil on the right. Then click on “F5 (Coil)” to add one coil just below the first coil. And then click on “F5 (Coil)” again to add the third coil.



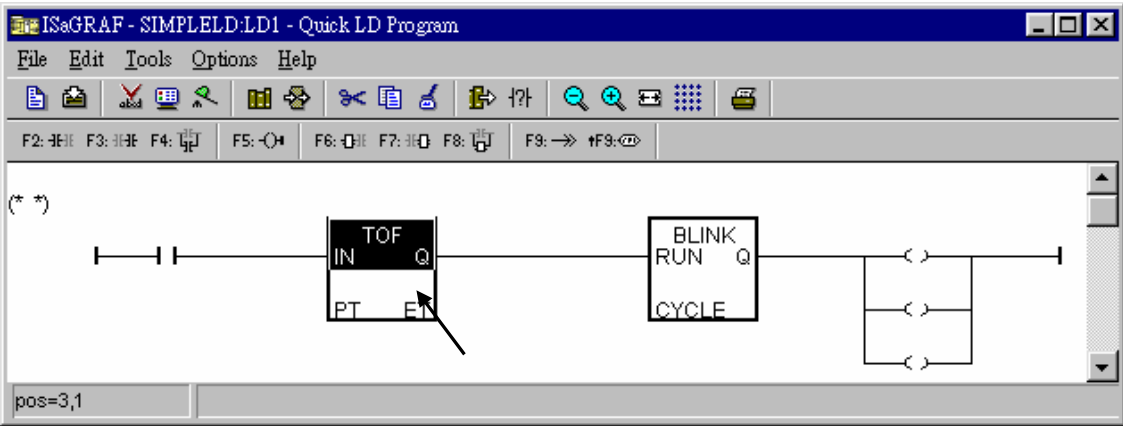
Then the window will look like below.



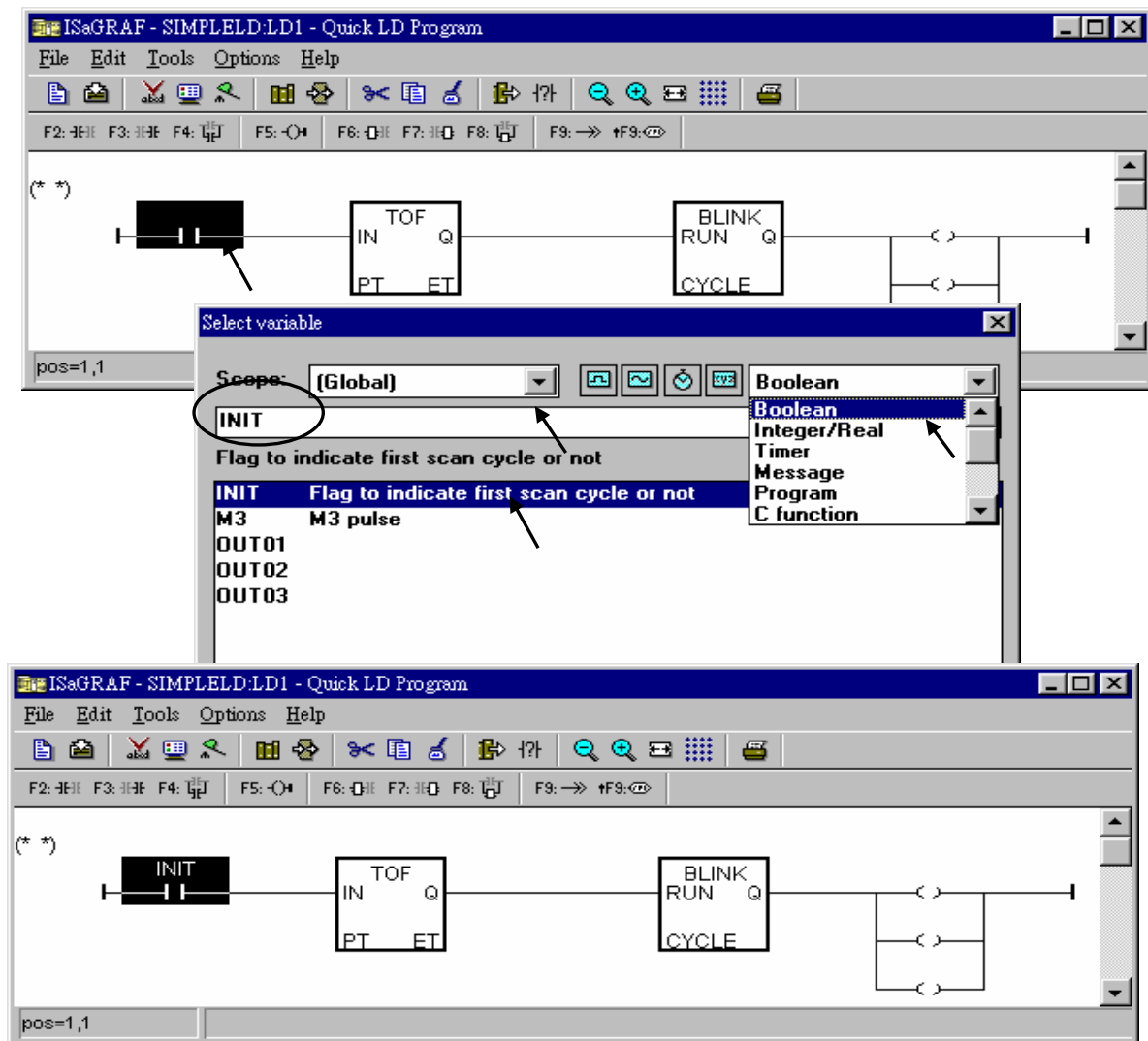
Double click anywhere inside of the second block and the "Function Block" assignment window appears. Select the "BLINK" type function block are using in our example program. To learn how the "BLINK" function operates you can click on the "Info" button for a detailed explanation of its functionality.



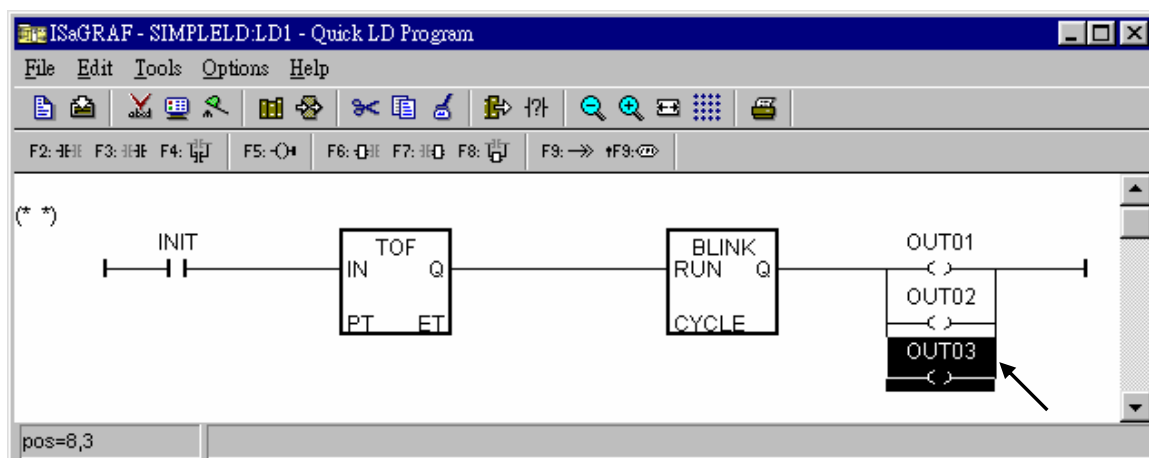
Using the same procedure to assign the first block to “TOF” as below.



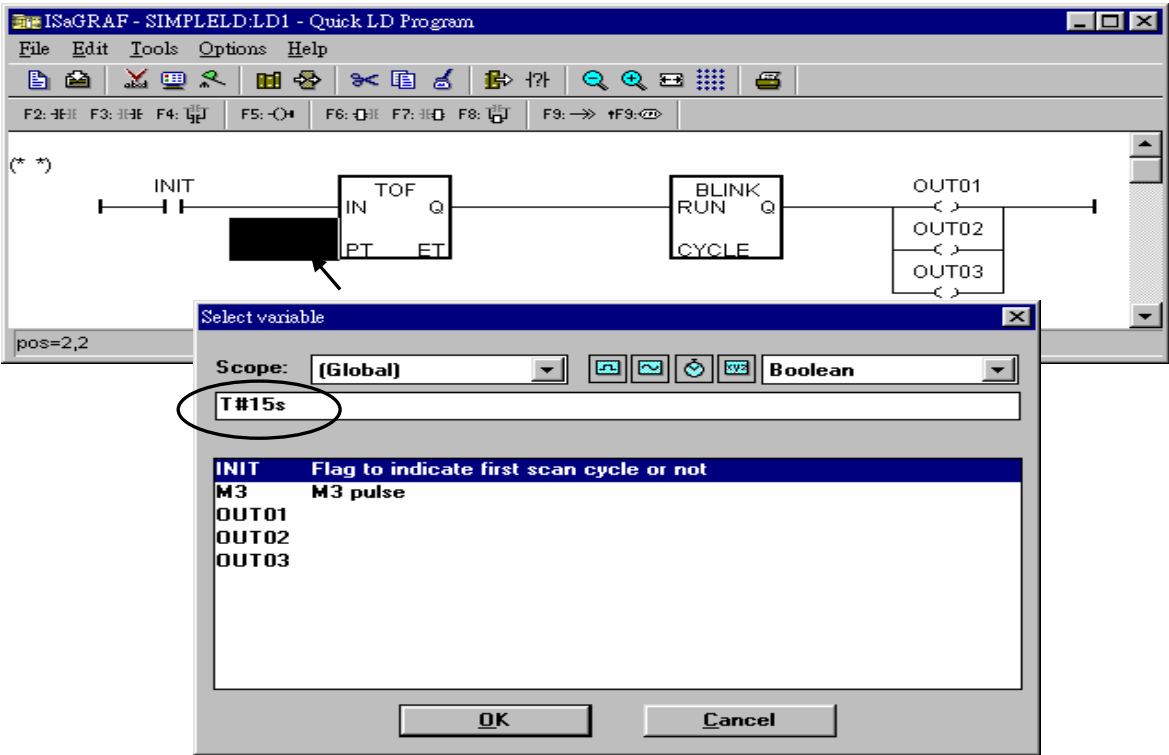
Now we are going to assign the associated variable & constant to each item. Double click on the first contact, a “Select variable” screen appeared. First select the “Scope” to “(Global)” and the proper type to “Boolean”. Then double click on “INIT” or you may use the keyboard to type “INIT”.



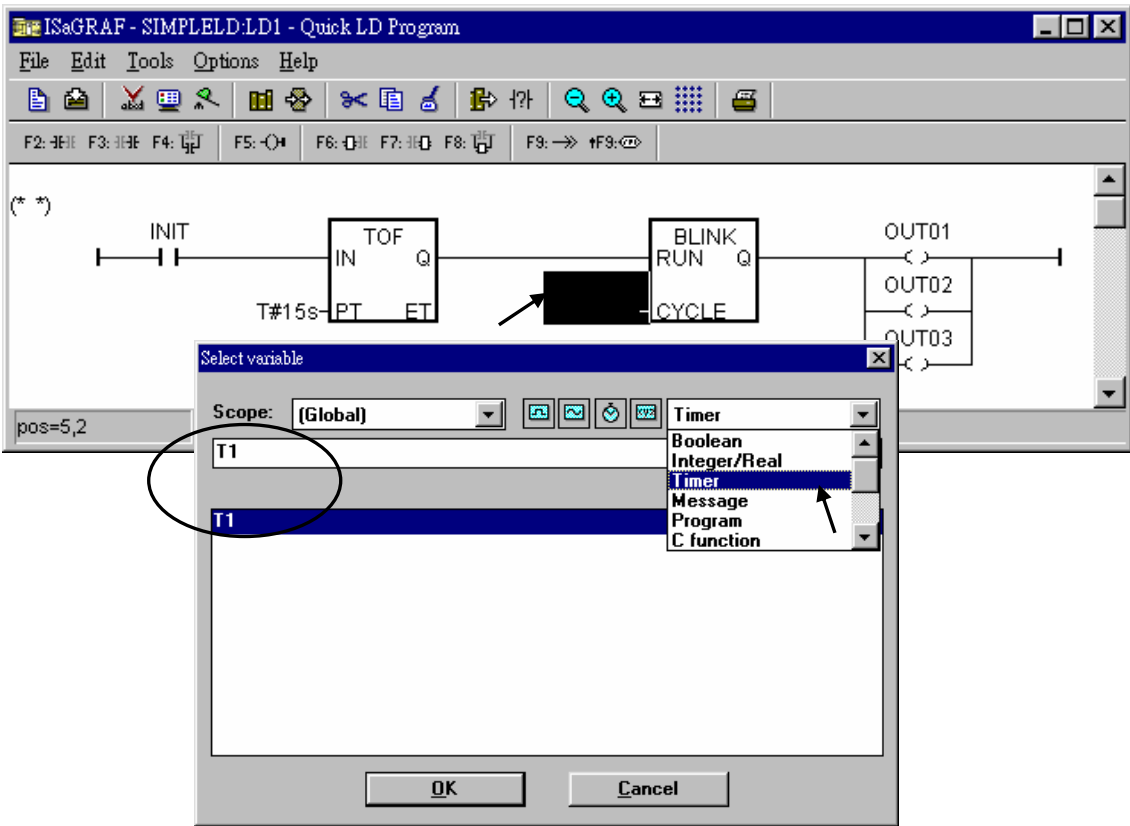
Using the same procedure to assign OUT01 thru. OUT03 to the associated coil.



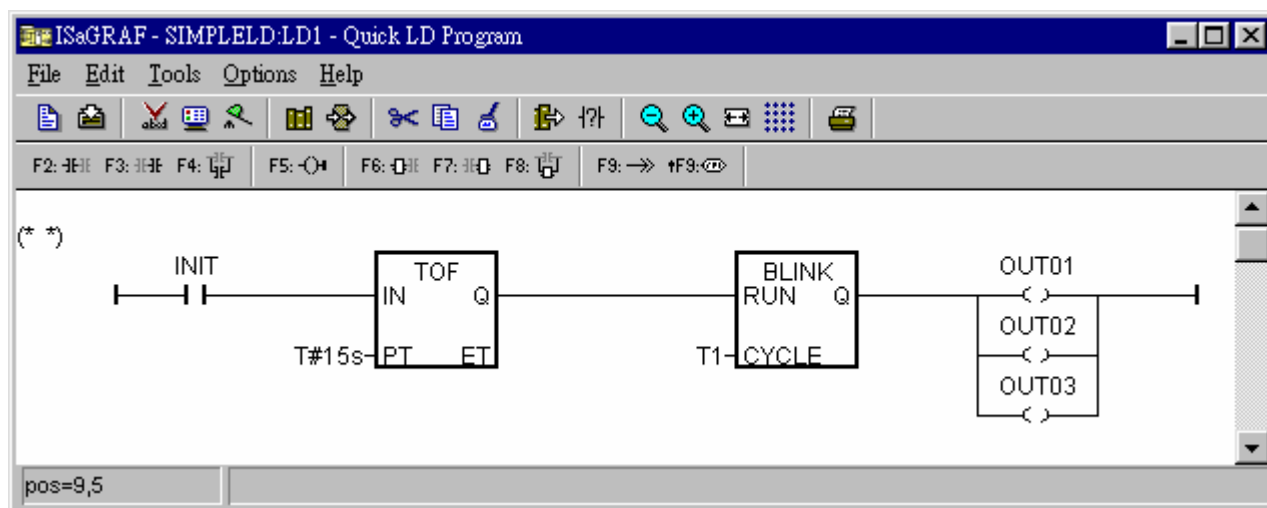
Now move your cursor to the left of the parameter “PT” of the “TOF” block. Double click on it, type “T#15s” (it means 15 second), then press “OK”.



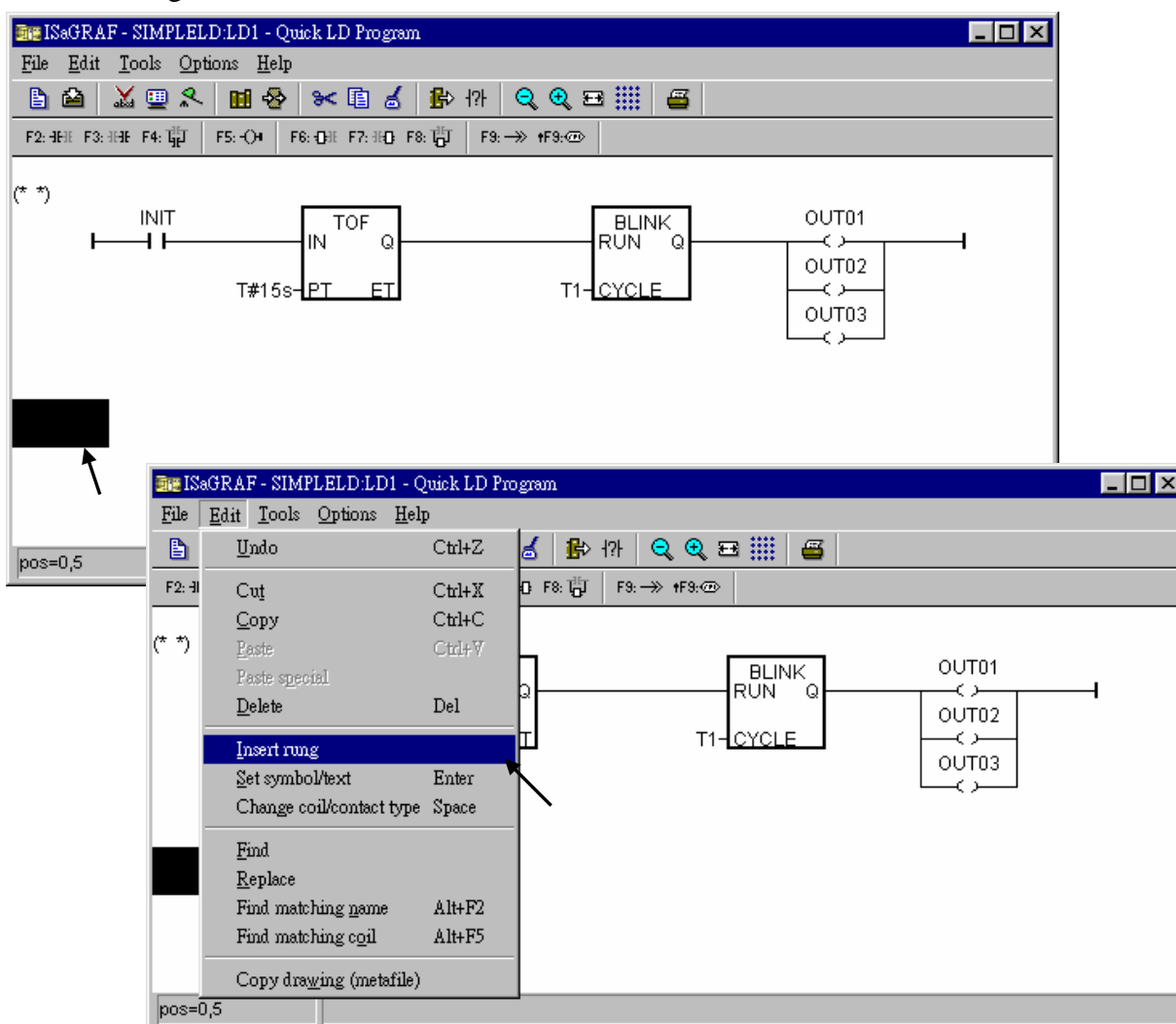
Do the same way to assign “T1” to the left of the parameter “CYCLE” of the “BLINK” block.



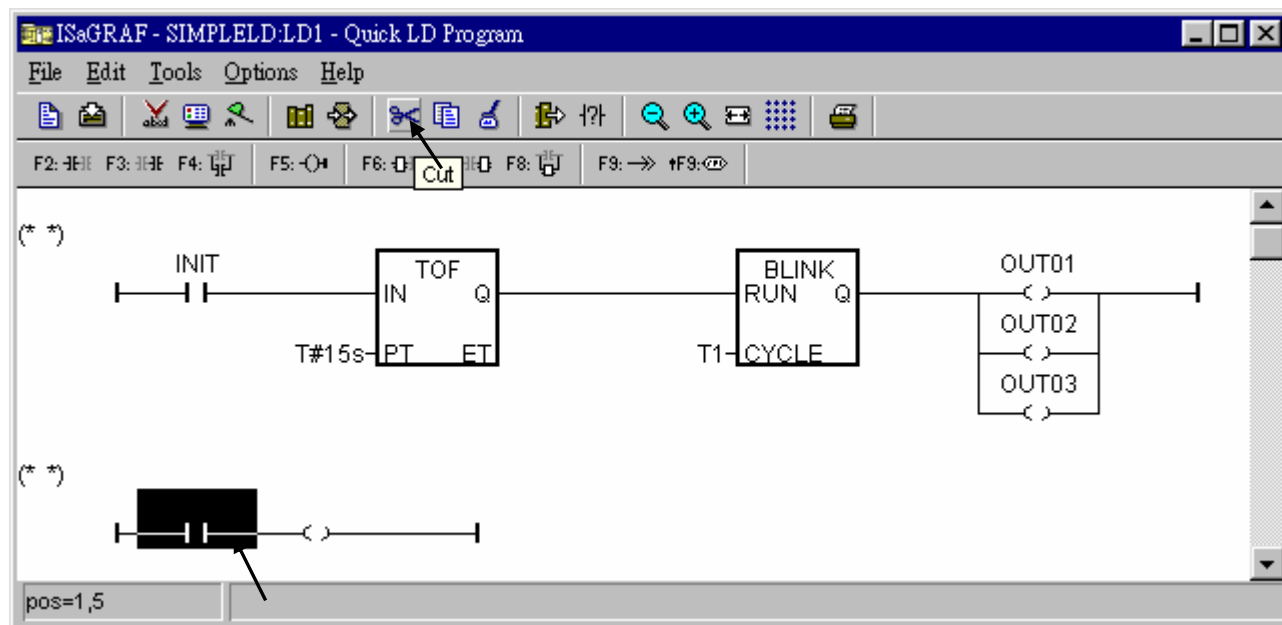
Now the window will look like below.



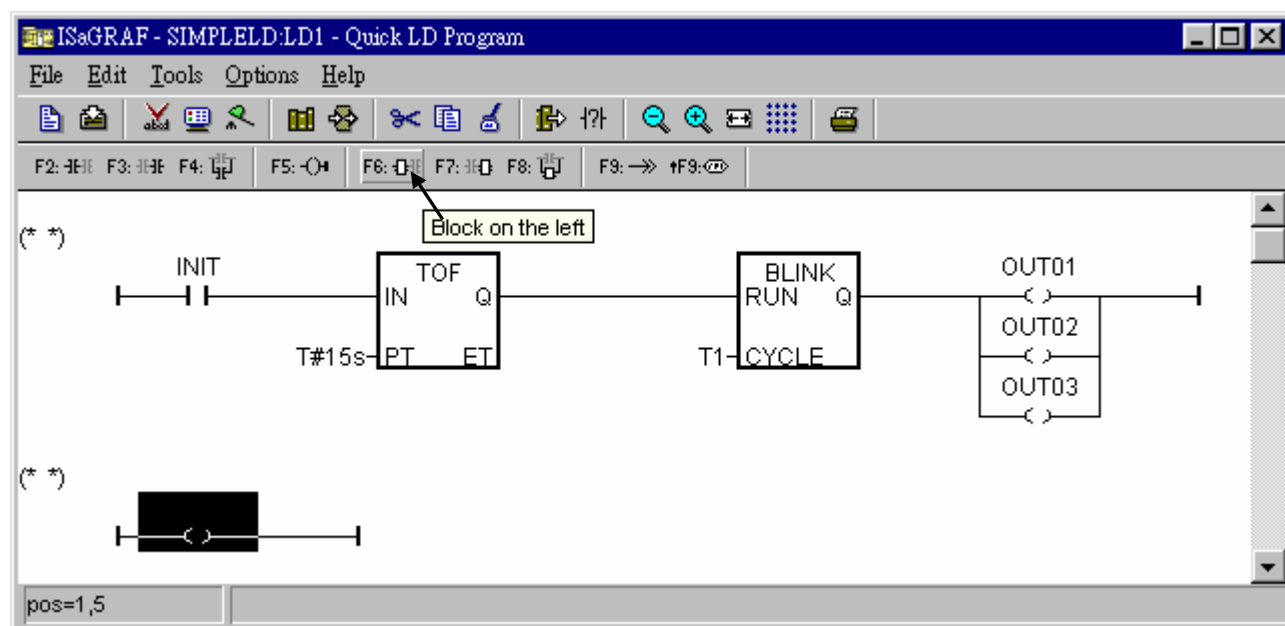
To add a new LD rung, first move the cursor to the proper position below the first rung. Then click on “Edit – Insert rung”



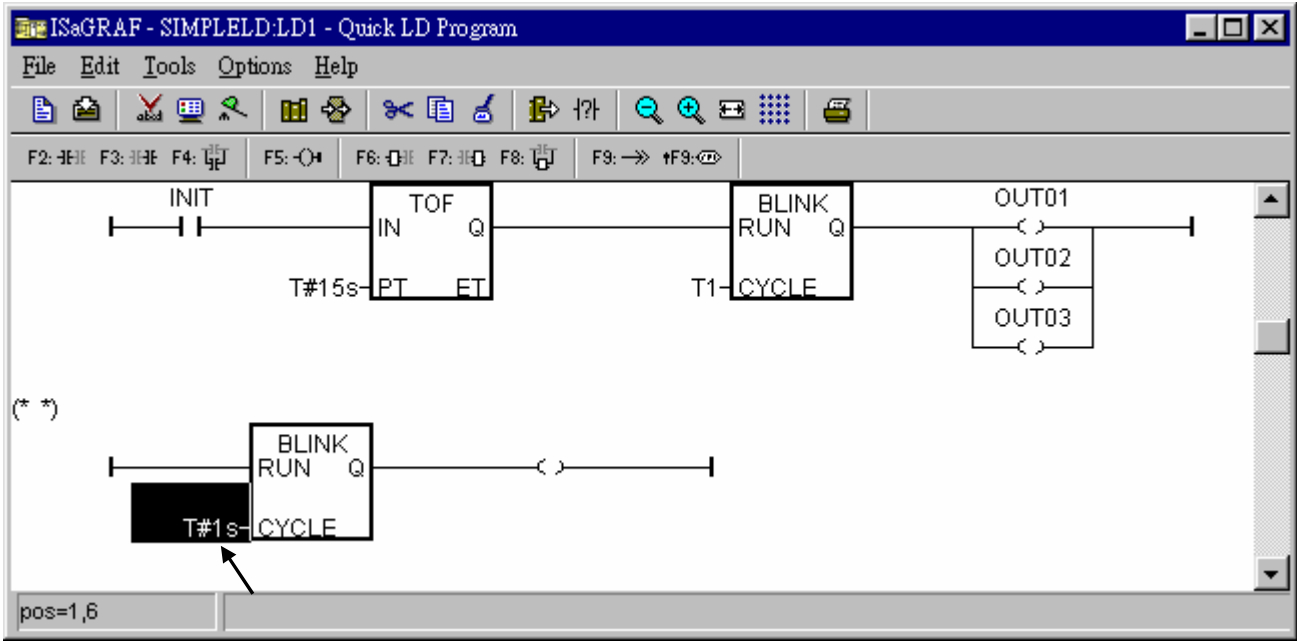
We don't need the contact in the new rung, move cursor to it, then click on "Cut".



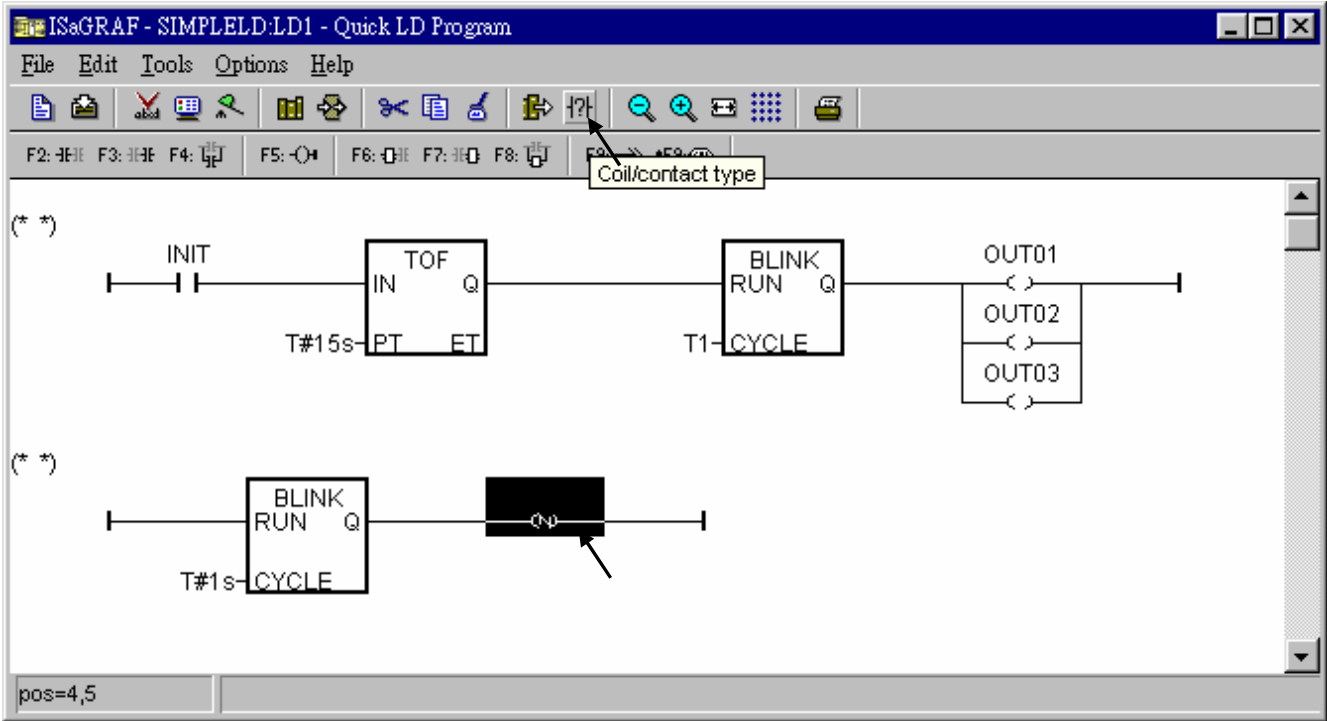
Now click on "F6 (Block on the left)", and then double click on inside the block to create an "BLINK" block.



Assign “T#1s” to the parameter of “CYCLE”, then we got the below window.

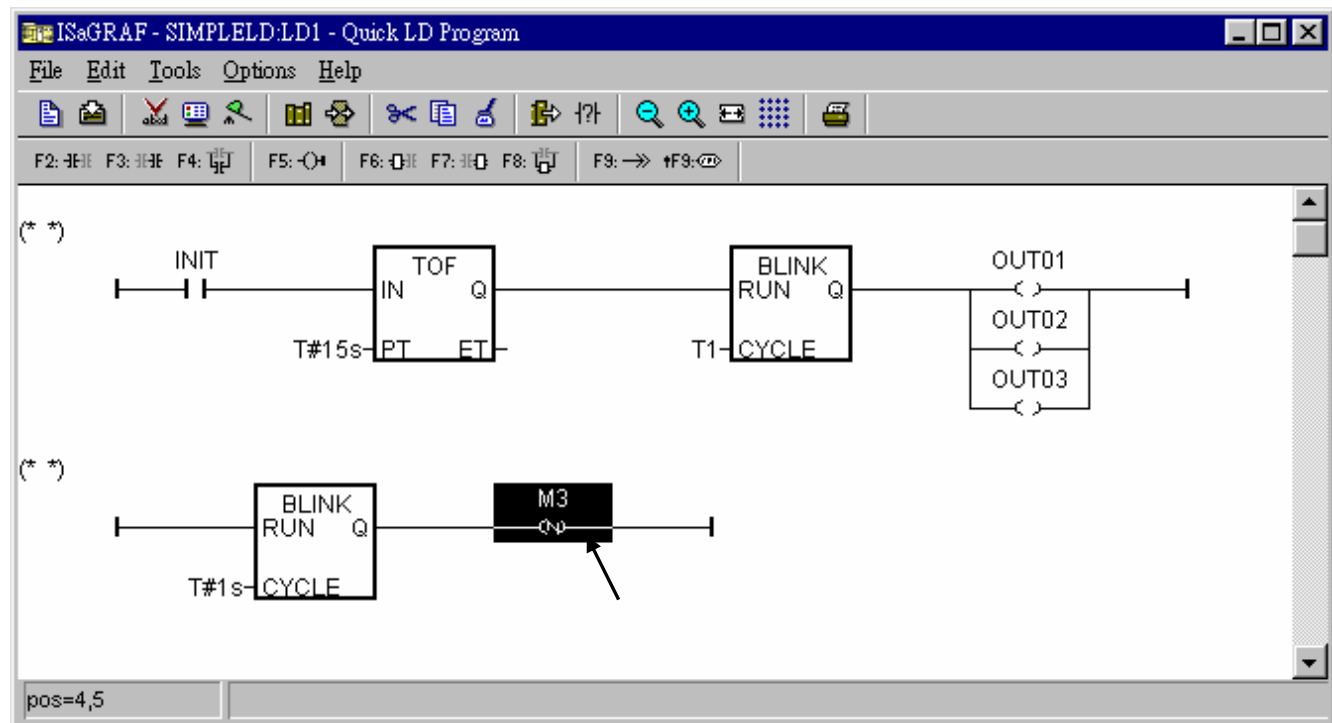


Move the cursor to the right coil, then click on “Coil/contact type” some times to assign the type to “N”.

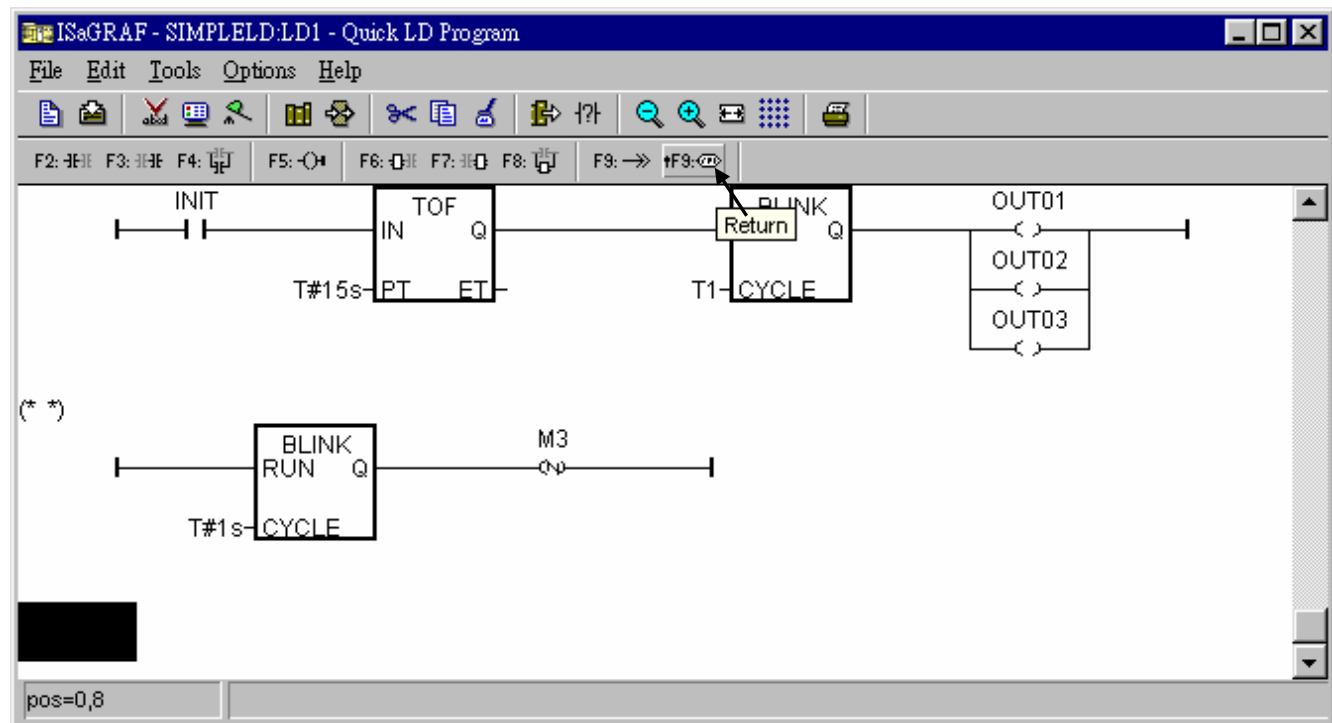




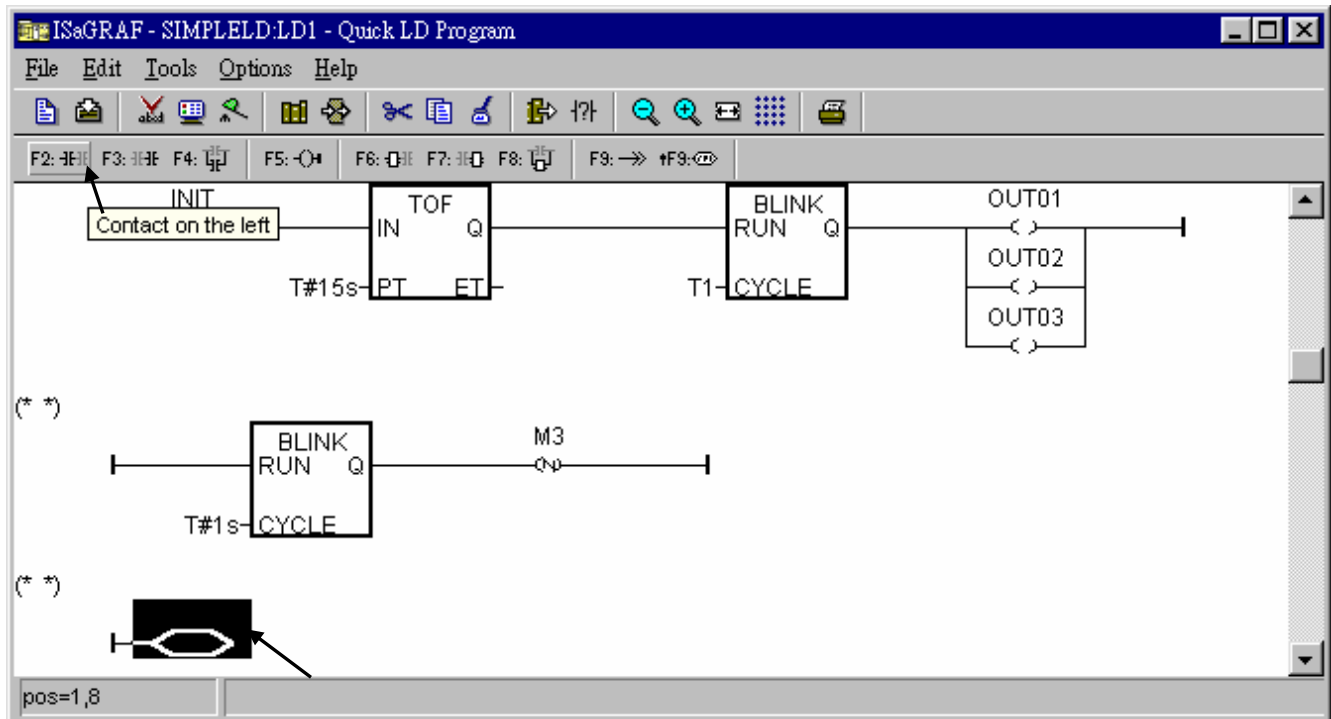
Double click on the N coil to assign “M3” to it.



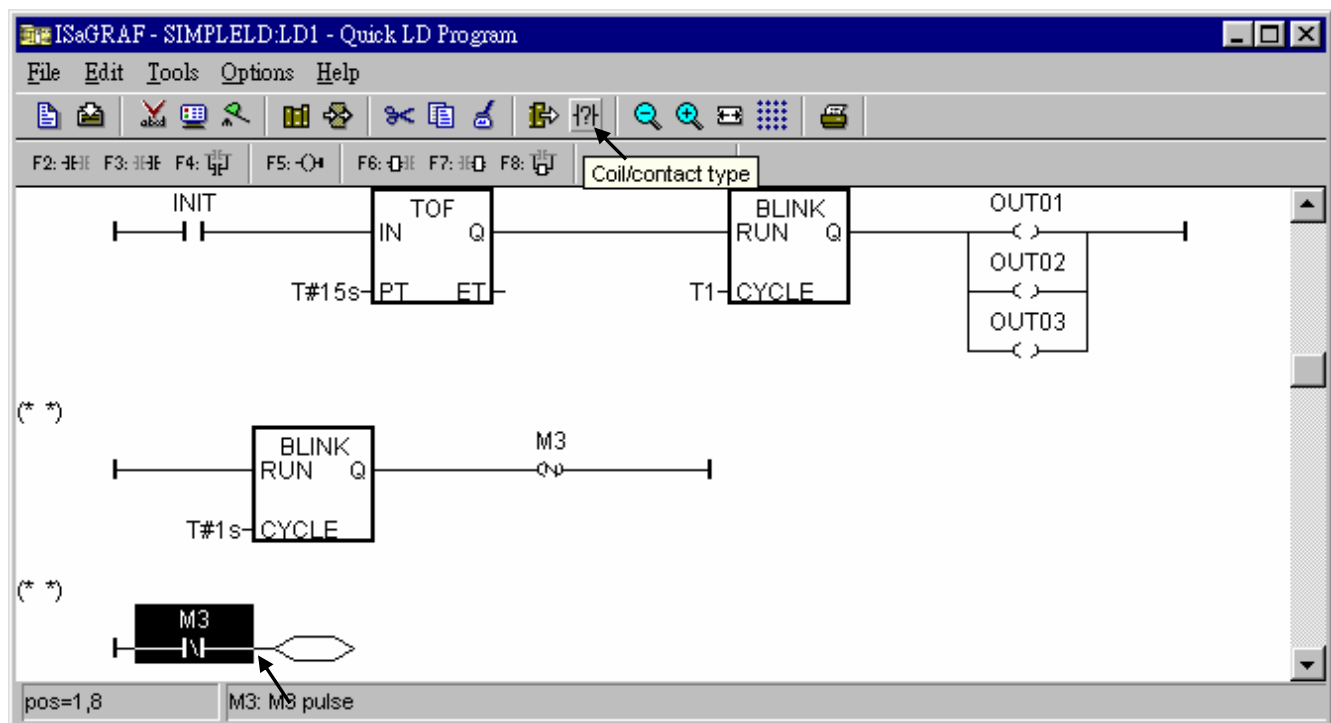
Now we are going to add another LD rung. Move the cursor to the below position of the second rung. And click on “F9 (Return)”.



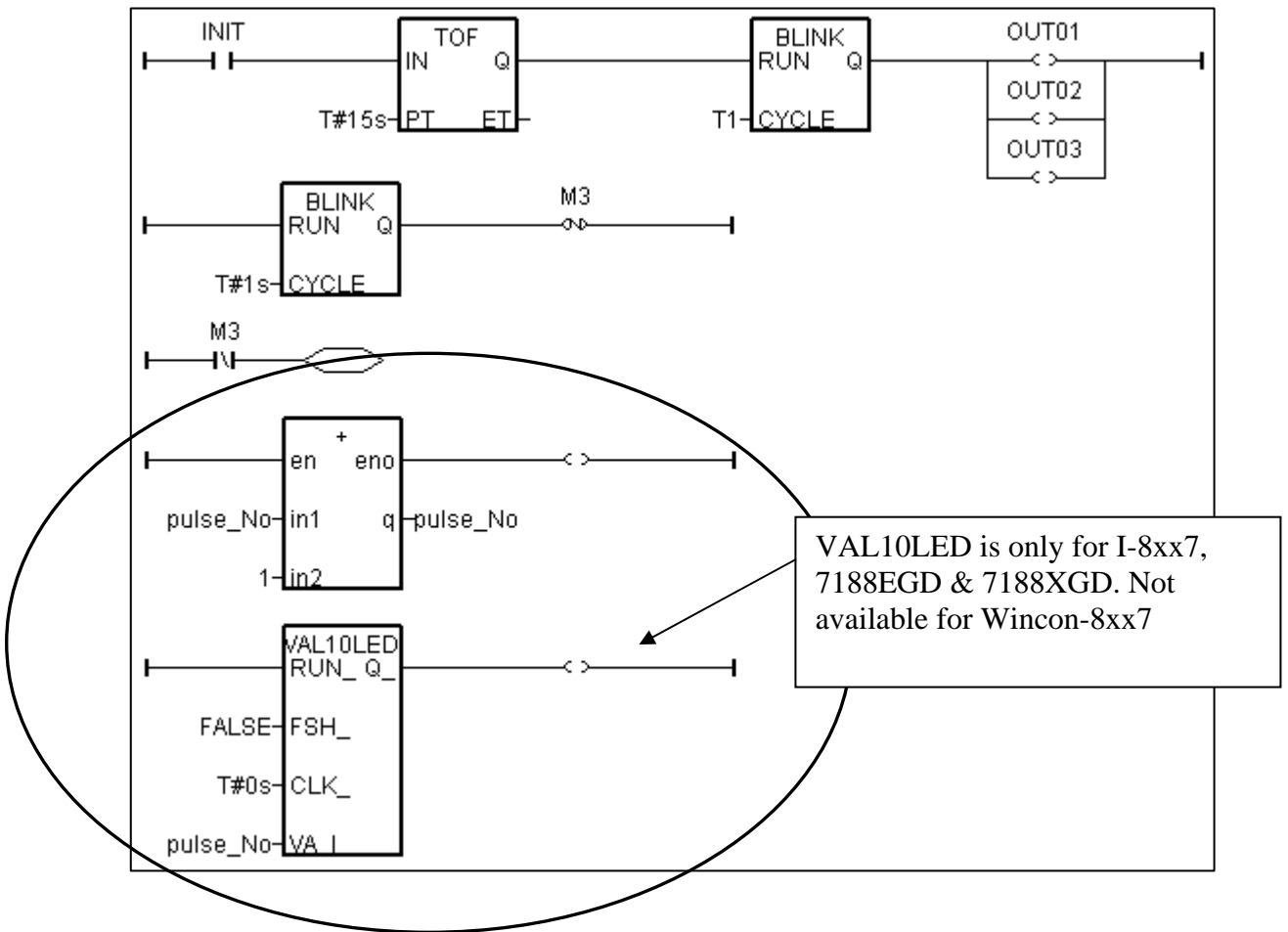
Move the cursor to “return” and then click on “F2 (Contact on the left)” to add a contact on the left.



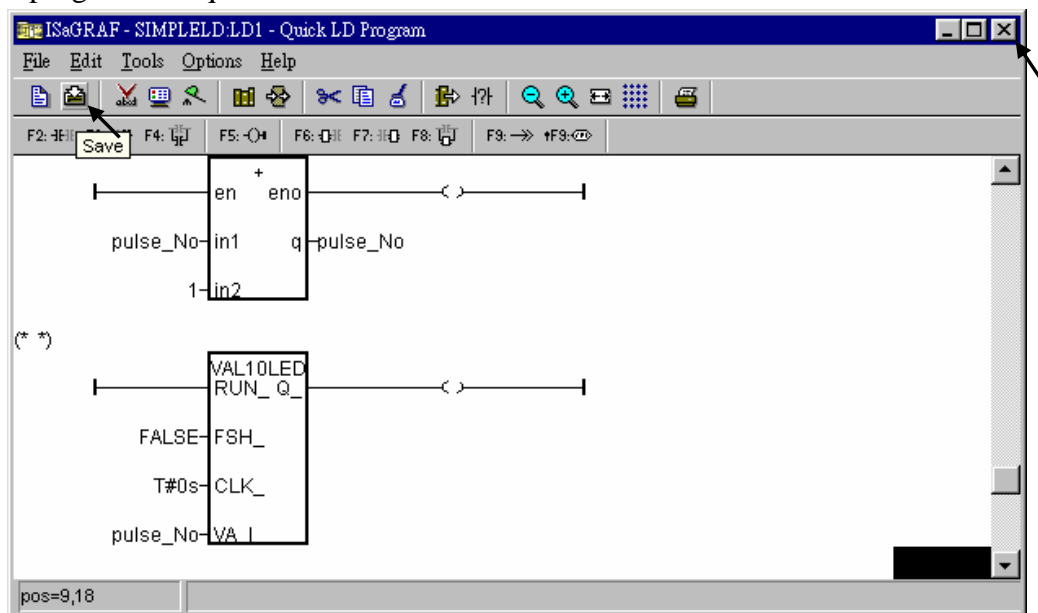
Then double click on the contact to assign “M3” to it. And change its type to “\” (inverted contact).



The procedure to create the forth & the last LD rung is similar as former steps. Please do it by yourself.  
The final LD program should look like the below.



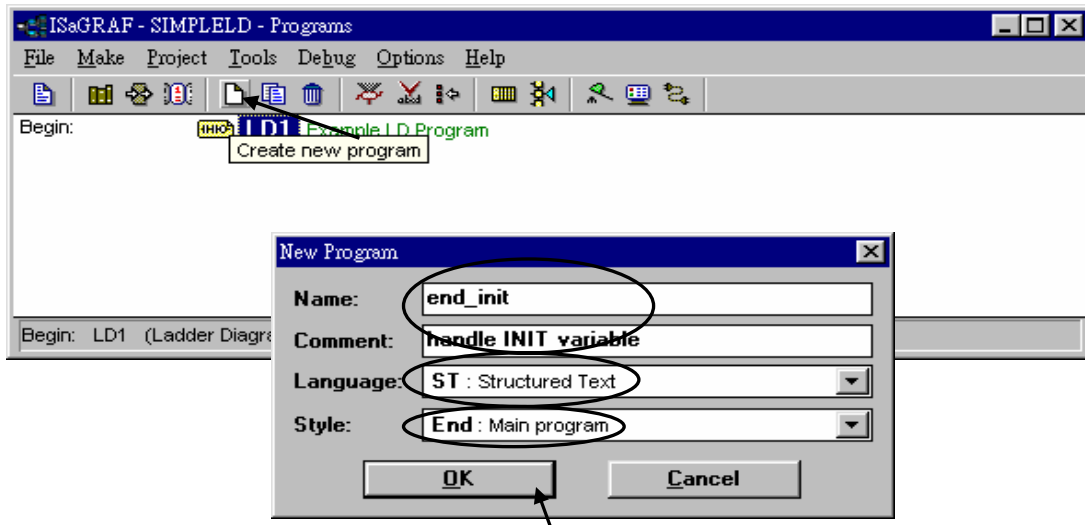
Save this LD program and quit.



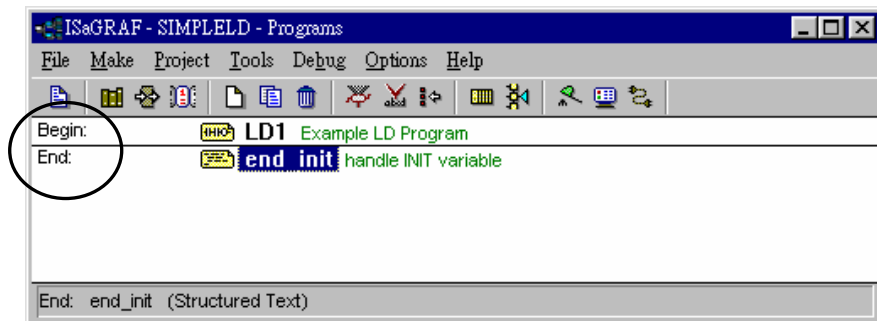
### 2.1.1.6: Create The ST "end\_init" Program

In this project we need an extra ST program to handle the "INIT" variable.

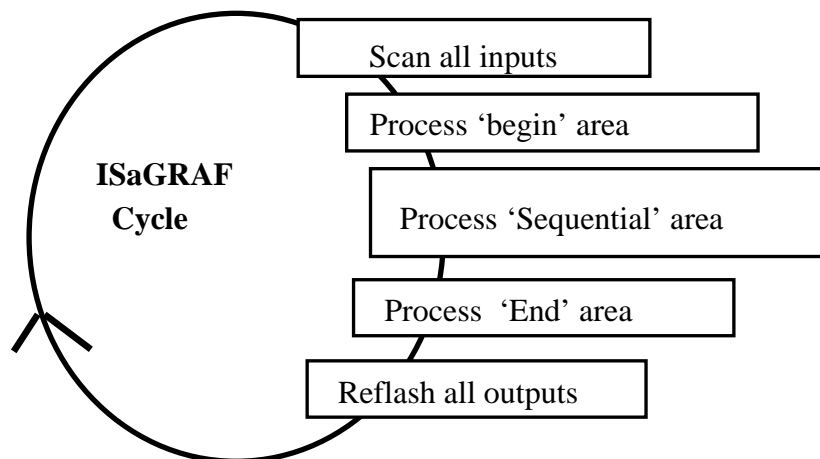
Click on "Create new program" in the "... - Programs" window to add a ST program. Given the Name as "end\_init", Comment as "Handle INIT variable", Language as "ST: Structured Text", & Style as "End: Main program". Then click on "OK".



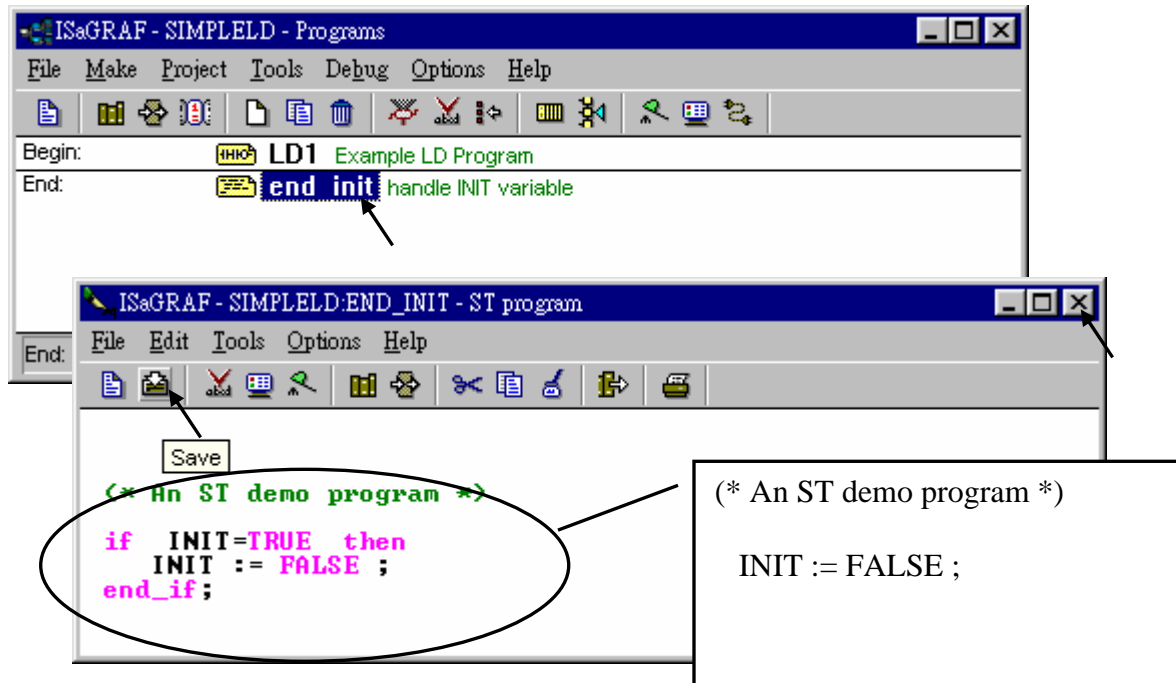
Now we have two programs inside this project.



ISaGRAF will run these two programs 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 run in the way as the below scheme.



Double click on “end\_init” program to edit it. Click on “save” and then exit when you finish it.  
(Any character inside between “(” and “)” is the comment.)



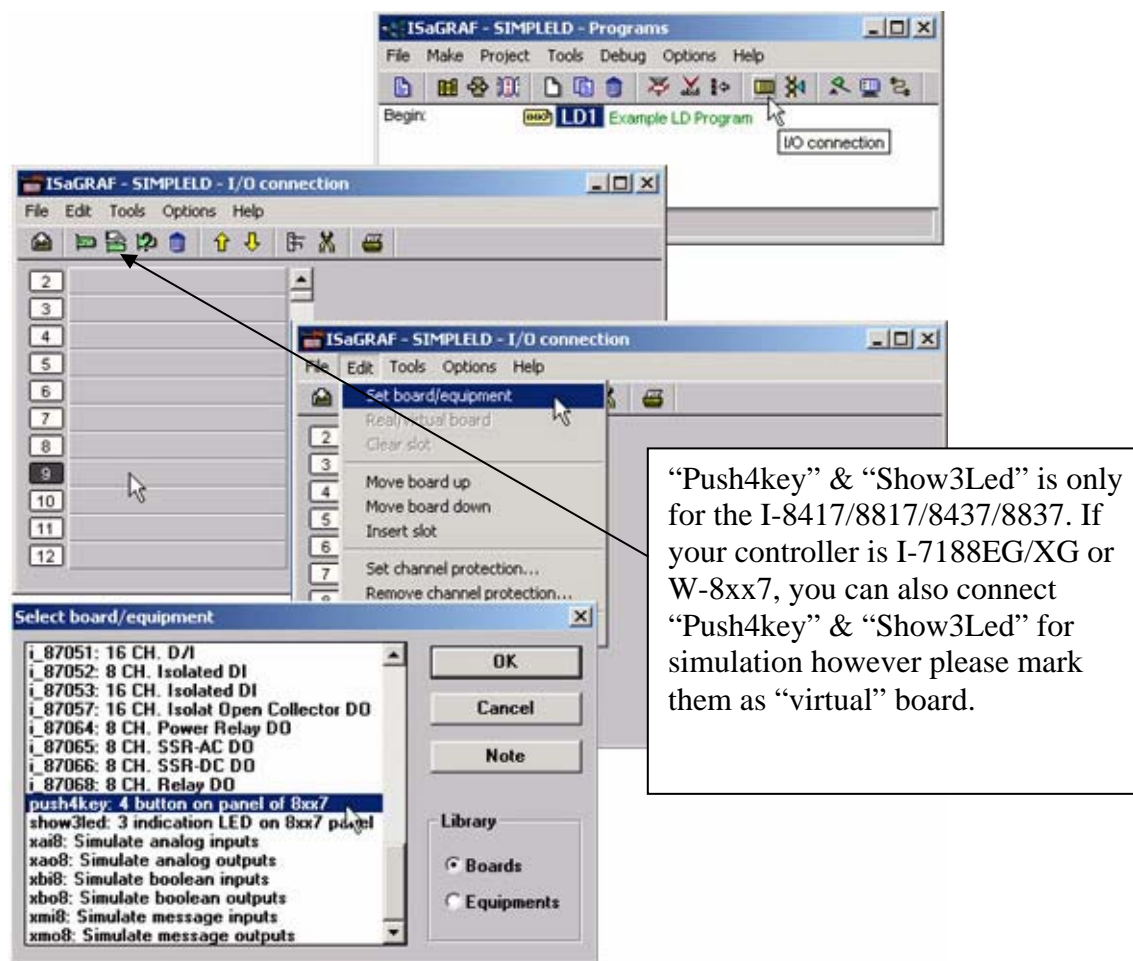
Since “INIT” is declared with an initial value “TRUE”, this ST program will let “INIT” set to “FALSE” at the end of the first scan cycle. In other word, “INIT” will indicate this project is running in the first scan cycle or not (TRUE: first scan cycle, FALSE: other cycles).

Now we have finished the programming, now we are going to the next step – “Connect the I/O”.

### 2.1.2: Connecting The I/O

The ISaGRAF Workbench software program is an open programming system. This allows the user to create an ISaGRAF program that can operate a large number of different PLC controller systems. It is the responsibility of the PLC hardware manufacturer to embed the ISaGRAF "driver" in their respective controller for the ISaGRAF program to operate properly. The ICP DAS line of I-8417/8817/8437/8837, I-7188EG, I-7188XG , W-8037/8337/8737 & W-8047/8347/8747 series of controllers have the ISaGRAF driver embedded, creating a powerful and flexible industrial controller system.

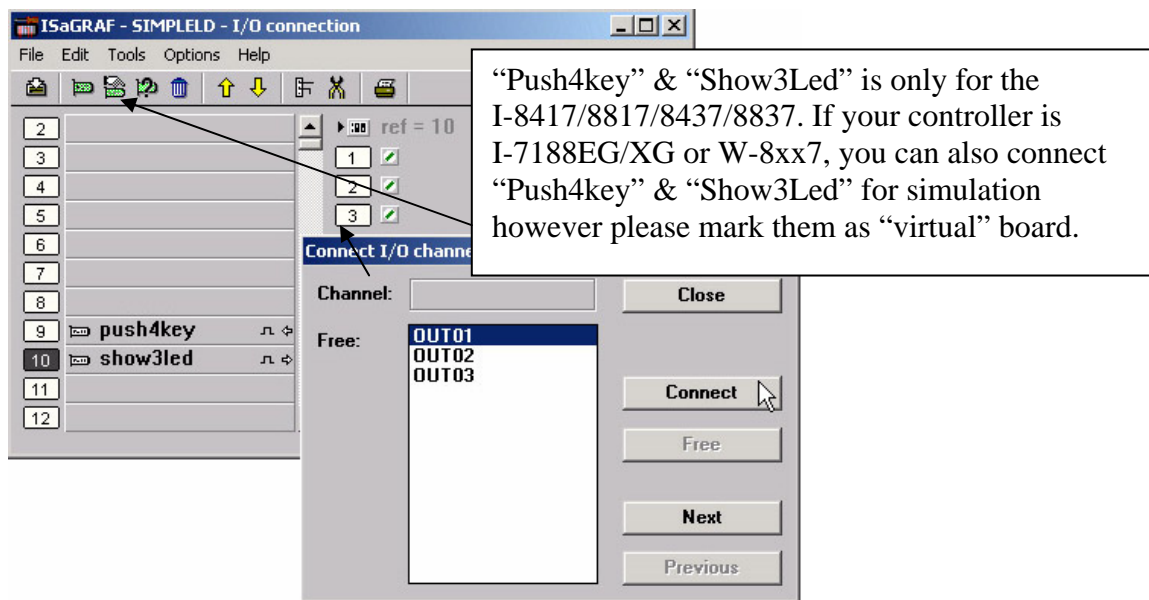
Now that you have created the ISaGRAF example program, now you must connect the I/O to the controller system. A useful feature of the I-8xx7 controller system is that you can run program we have created WITHOUT having any I/O boards plugged into the I-8xx7 controller system. The four pushbuttons on the I-8xx7 controller system can be used as four digital inputs, and the three left LED's above the control panel pushbuttons can be used as outputs.



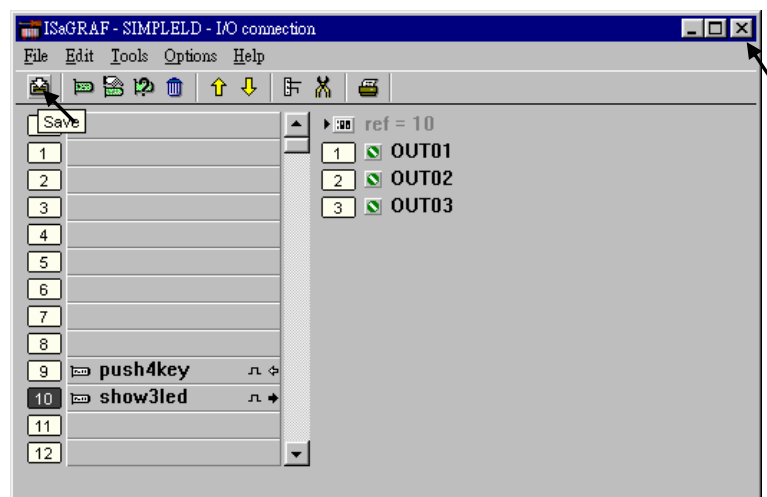
Click on the "I/O Connection" icon as shown in the top picture and the "I/O Connection" window will appear as shown in the next illustration. For the purpose of this example, you can either double click on the "9" slot, or just click on the "9" slot, then click on "Edit" and then "Set Board/Equipment" and then the "I/O Connection" window will appear. This now associates the four control panel pushbuttons - “push4key” as four digital inputs. (We don’t use it in this example program since there is no boolean variable declared with “Input” attribution).

**IMPORTANT NOTICE:** I/O Slots 0 through 7 are reserved for REAL I/O boards that will be used in the I-8xx7 controller (W-8337/8737 doesn’t have slot 0). You can use slots 8 and above for additional functionality as illustrated by the example program.

To create the I/O connections for the outputs, double click on the "10" slot, then click on the "Show3led: 3 indication LED on 8xx7 panel" selection. This will now associate the three LED’s above the four control panel pushbuttons as the three outputs for the example program. Your "I/O Connection" window should now look like the screen below.



Remember to click on the "SAVE" icon to save the I/O connections that have been created for the example program. And click on the "X" to exit the window.



**IMPORTANT NOTE:** 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 3 boolean output variables, they are OUT01, OUT02 & OUT03.

### 2.1.3: Compiling The Example LD Project

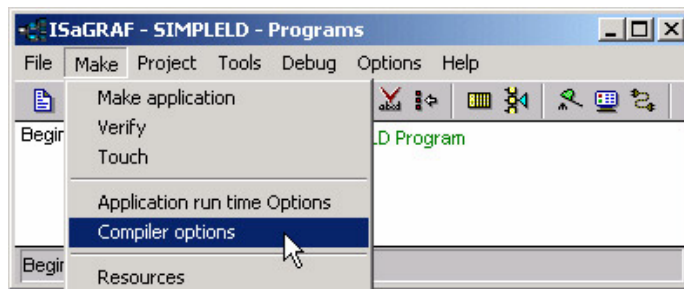
For ANY AND EVERY ISaGRAF program to work properly with any of the I-8xx7, I-7188EG, 7188XG & W-8xx7 controller systems, it is the responsibility of the programmer to properly select the correct "Compiler Options". You MUST select the "ISA86M: TIC Code For Intel" option as described below.

#### Note:

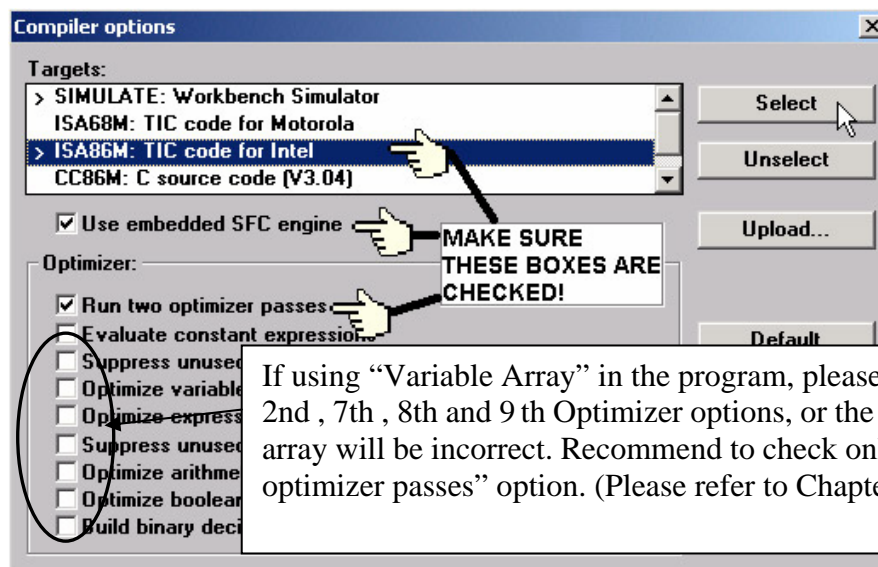
1. The I-8417/8817/8437/8837, I-7188EG and I-7188XG has memory limitation. The ISaGRAF code size can not exceeds 64K bytes. (size of the "appli.x8m" file)

2. W-8037/8337/8737 and W-8047/8347/8747 has code size limitation of 1M bytes. It is 16 times of I-8xx7 & I-7188EG/XG.
3. The CPU speed of the W-8037/8337/8737 and W-8047/8347/8747 is about 10 to 20 times of the I-8xx7 & I-7188EG/XG. Especially for floating point value calculation. The CPU speed of I-8437-80 / I-8837-80 is about 4 times of I-8xx7 & I-7188EG/XG. Please refer to "Performance Comparison Table of ISaGRAF Controllers" in this manual.

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.



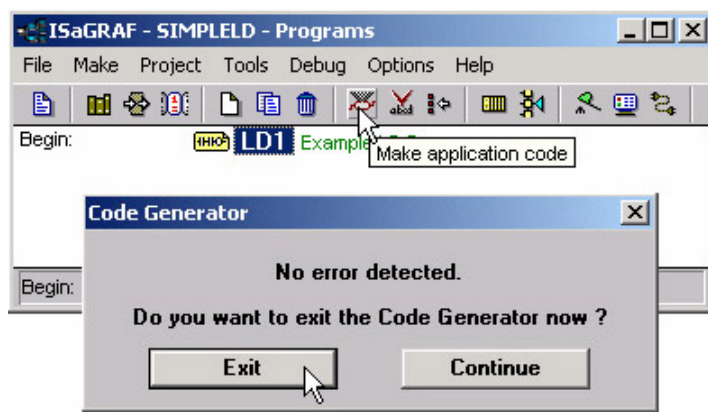
If using "Variable Array" in the program, please DO NOT check the 2nd , 7th , 8th and 9 th Optimizer options, or the value of the Variable array will be incorrect. Recommend to check only the 1st – "Run two optimizer passes" option. (Please refer to Chapter 2.6)

## 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 LD project. If there are no compiler errors detected during the compilation process, CONGRATULATIONS, you have successfully created our example LD program.

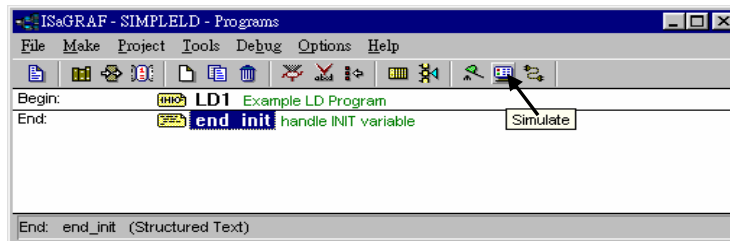
If errors are detected during the compilation process, just click on the "CONTINUE" button to review the error messages. Return to the Project Editor and correct the errors as outlined in the error message window.





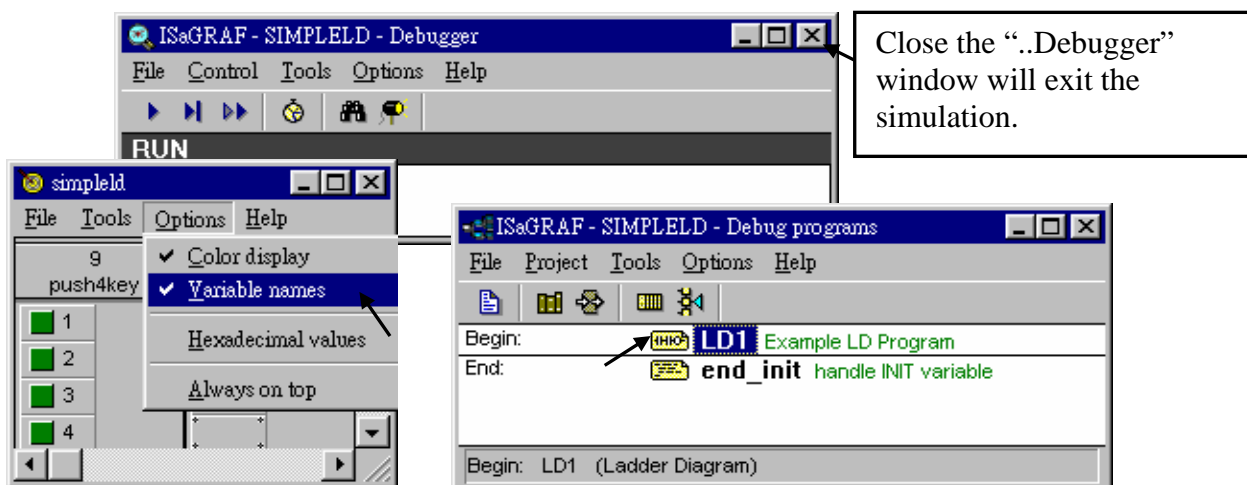
### 2.1.4: Simulating The LD Project

A powerful program-debugging feature of the ISaGRAF software program is the ability to "SIMULATE" the program you have developed before loading it into the ISaGRAF controller system. After successfully compiling the example LD program, click on the "SIMULATE" icon as shown below.



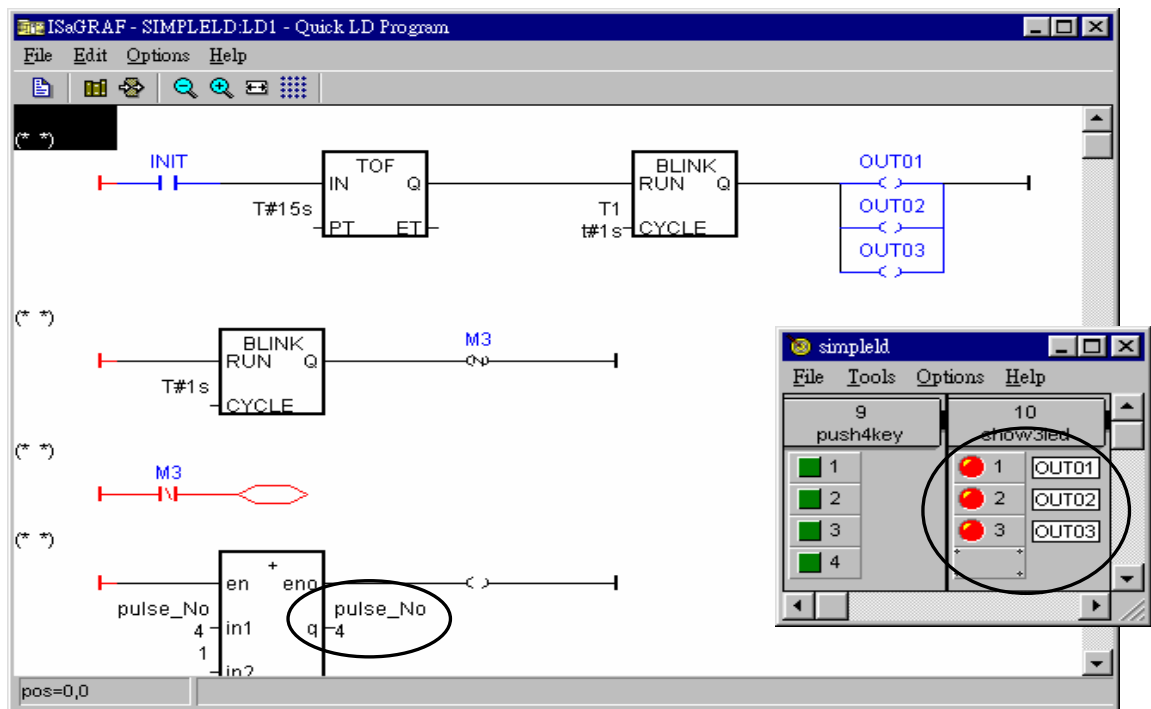
When you click on the "Simulate" icon three windows will appear. The windows are the "ISaGRAF Debugger", the "ISaGRAF Debug Programs", and the "I/O Simulator" windows. If the I/O variable names you have created DO NOT appear in the I/O simulator window, just click on the "Options" and "Variable Names" 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.



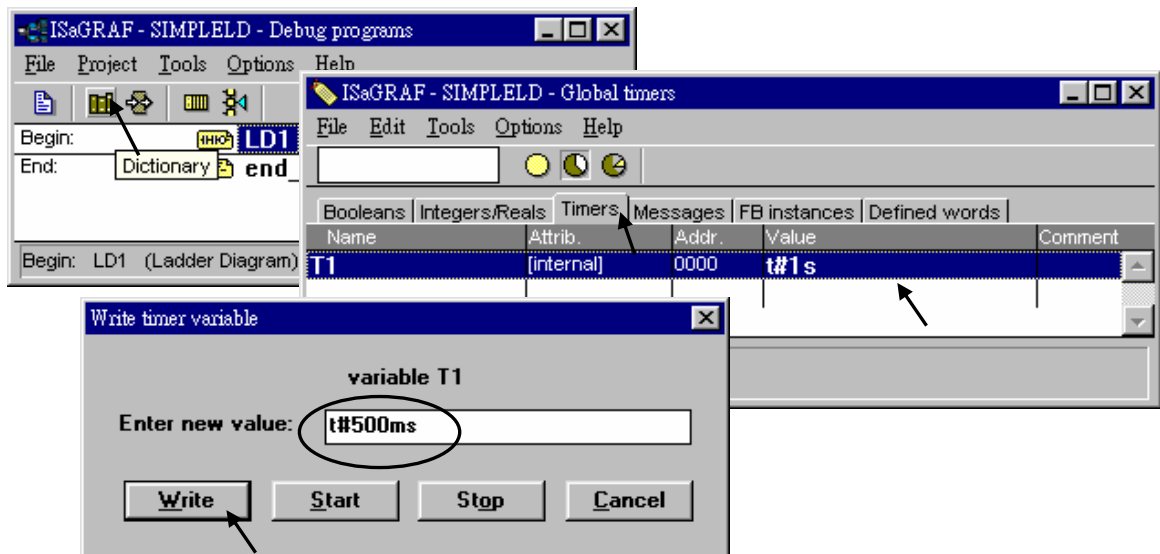
### Running The Simulation Program

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



You can see outputs “OUT01” thru. “OUT03” will blink in the first 15 seconds. And the “pulse\_No” continuously plus one every second.

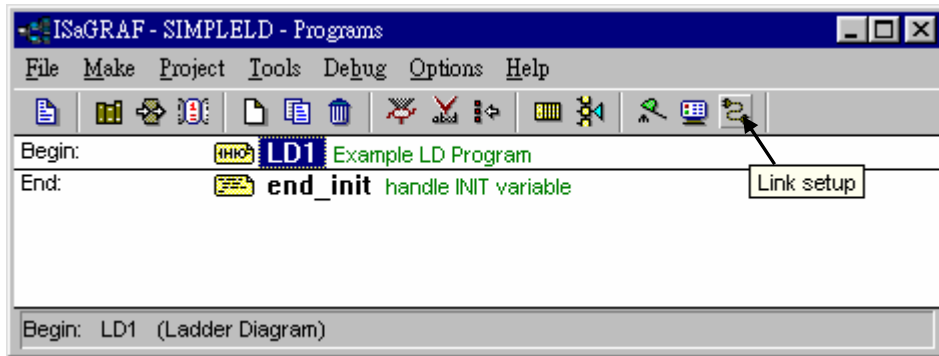
You can adjust the "T1" variable while the program is running. To accomplish this, click on the "Dictionary" icon which will open the "ISaGRAF Global Variables" window as shown in the first two pictures below. Click on “Timer” tab and then double click on “T1” to change the timer value to “T#500ms” (this means 0.5 second). Then click on “Write”.



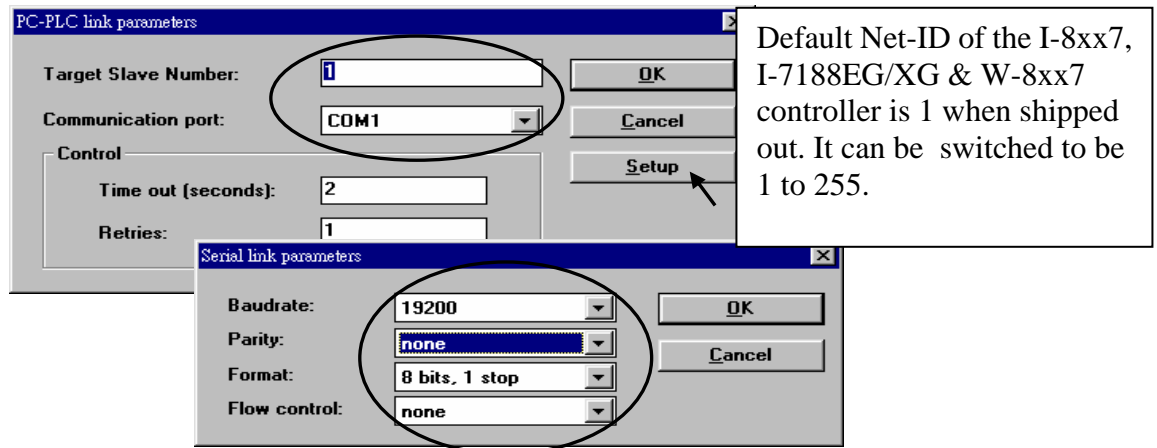
## 2.1.5: Download & Debugging The Example LD Project

The last step required to running the example LD program on the ISaGRAF controller system is to download the project to the controller (frequently referred to as the "Target" platform"). Before this download can be accomplished you must first establish communications between your development PC and the controller.

To begin this process, click on the "Link Setup" icon in the "ISaGRAF Programs" window. When you click on the "Link Setup" icon, the following window will appear.



The "Target Slave Number" is the Node-ID address for the I-8xx7 controller as defined by the dipswitch settings outlined in Chapter 1, Section 1.3.1. The Node-ID dipswitch is located in the bottom right portion of the I-8xx7 controller. If your I-8xx7 controller is the first one, the Node-ID address should be set to "1". The "Communication Port" is the serial port connection on your development PC, and this is normally either COM1 or COM2.



The communication parameters for the target I-8xx7 controller MUST be set to the same serial communication parameters for the development PC. For I-8417 and I-8817 controllers (serial port communications), the default parameters for COM1 (RS232) and COM2 (RS485) ports are:

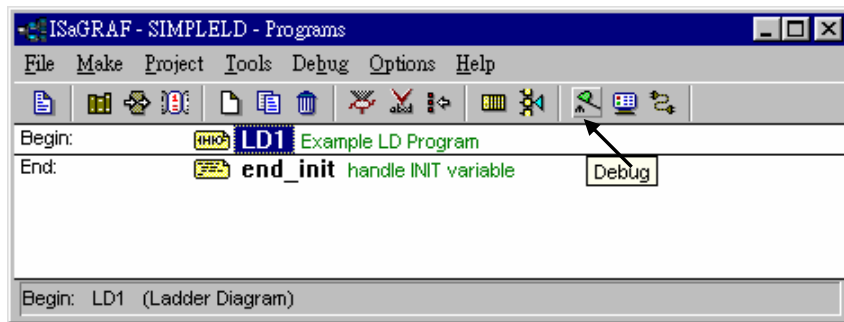
|               |                |
|---------------|----------------|
| Baudrate:     | 19200          |
| Parity:       | none           |
| Format:       | 8 bits, 1 stop |
| Flow control: | none           |

## IMPORTANT NOTE

It may be necessary to change the COM port settings for the development PC. Depending on which computer operating system you are using, you will need to make sure that the COM port can properly communicate to the I-8xx7, I-7188EG/XG & W-8xx7 controller system.

## DOWNLOADING THE EXAMPLE PROJECT

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



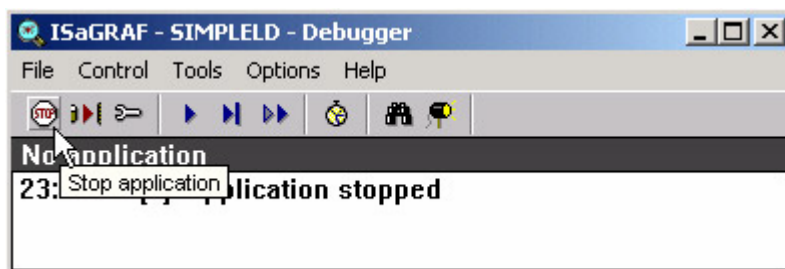
If the development PC and the I-8xx7, I-7188EG/XG & W-8xx7 controller system are communicating properly with each other, the following window displayed below will appear (or if a program is already loaded in the controller system, the name of the project will be displayed with the word "Active" following 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 controller system.

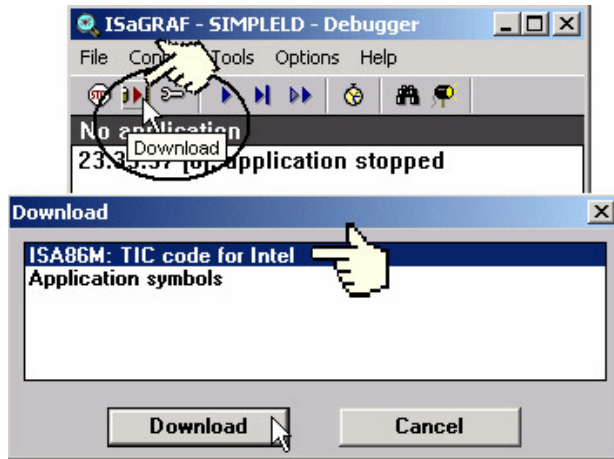
You may have to either change the serial port communication settings for the development PC (which may require changing a BIOS setting) or change the "Serial Link Parameters" in the ISaGRAF program.

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.

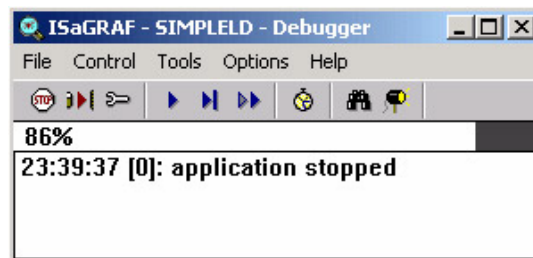


## STARTING THE DOWNLOADING PROCESS

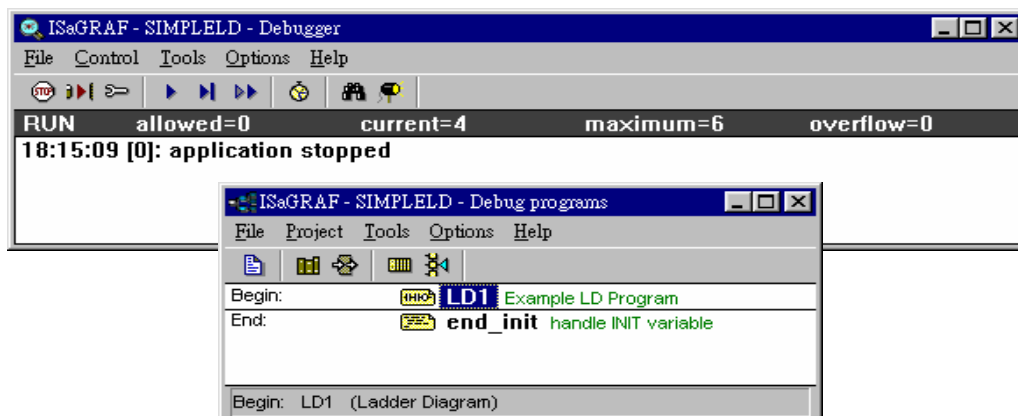
From the "ISaGRAF Debugger" window click on the "Download" icon, then click on "ISA86M: TIC Code For Intel" from the "Download" window as shown below.



The example project will now start downloading to the I-8xx7, I-7188EG/XG & W-8xx7 controller system. A progress bar will appear in the "ISaGRAF Debugger" window showing the project downloading progress.



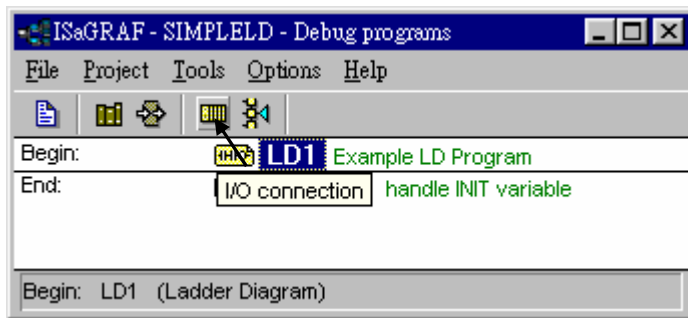
When the example project has successfully completed the downloading process to the controller system the following two windows will appear.



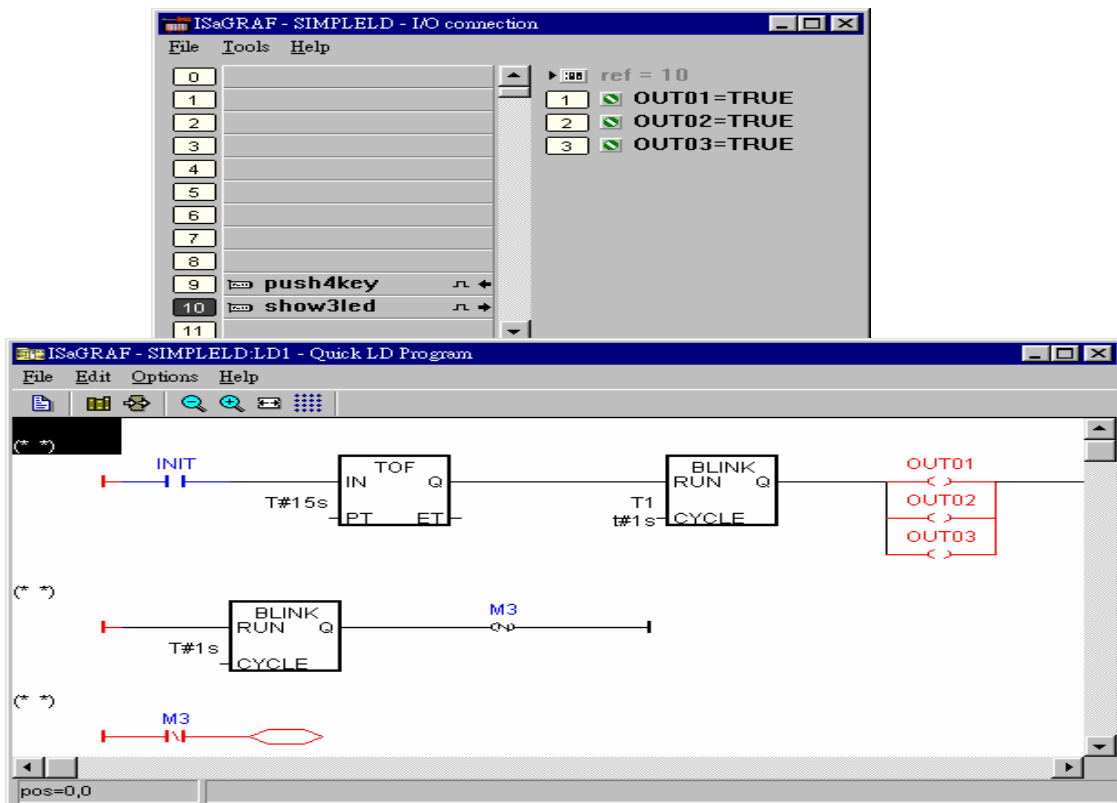
## RUNNING THE EXAMPLE LD PROGRAM

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. 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.



In the window below, the OUT01 thru. OUT03 is blinking in the first 15 seconds. The "Quick LD Program" window shows the entire ladder logic program in REAL TIME and is an excellent diagnostic tool for development and troubleshooting.



Though there are numerous steps involved in creating and downloading an ISaGRAF program, each step is quick and easy to accomplish, and the end result is a powerful and flexible control development environment for the ISaGRAF controller systems.

**PRACTICE, PRACTICE, PRACTICE!**

Now that you have successfully created and ran your first ISaGRAF program with the I-8xx7, I-7188EG/XG & Wincon-8xx7 controller system, you should practice creating more elaborate and powerful programs. Like any other computer development environment, practice and experimentation is the key to understanding and success, **GOOD LUCK!**

## 2.2: A Simple Structured Text (ST) Program

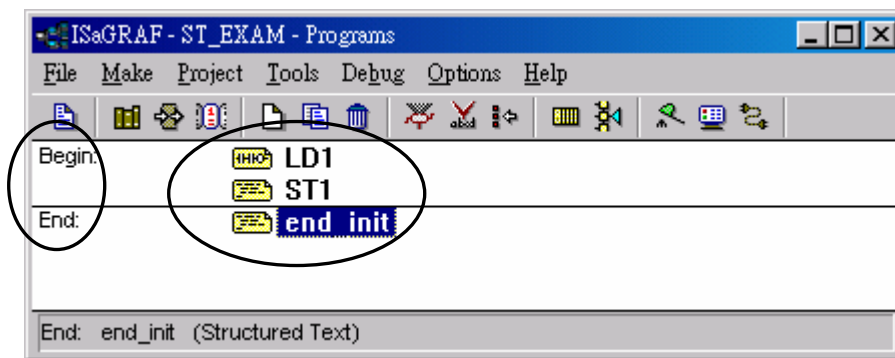
A "Structured Text" program is a high-level program language that is designed for automation process control applications. The "Structured Text (henceforth referred to as "ST") is primarily used to implement complex procedures that cannot be easily expressed by a graphical language such as LD or FBD.

An ST program is comprised by a list of "ST Statements", and each "ST Statement" MUST end with a semi-colon ";". All characters inside between "(" and ")" is comment.

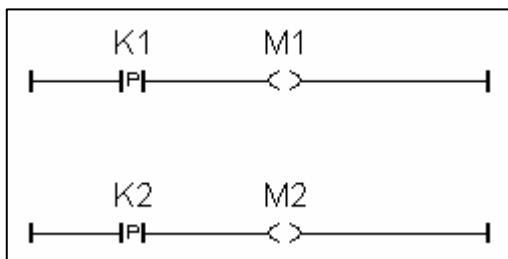
Variables Used In The Example ST Project:

| Name  | Type    | Attribute | Description  |
|-------|---------|-----------|--|
| INIT  | Boolean | Internal  | initial value at "TRUE". TRUE means 1 <sup>st</sup> scan cycle           |
| K1    | Boolean | Input     | The first pushbutton on the front panel of the I-8xx7                    |
| K2    | Boolean | Input     | The second pushbutton on the front panel of the I-8xx7                   |
| M1    | Boolean | Internal  | Indicate pushbutton K1 is just pushed.                                   |
| M2    | Boolean | Internal  | Indicate pushbutton K2 is just pushed.                                   |
| TEMP  | Boolean | Internal  | A boolean variable for temporary use                                     |
| COUNT | Integer | Internal  | A integer value generated by push K1 & K2<br>initial value is set at "0" |

Three programs are used in this example. One is LD program named "LD1", The other two are ST programs named respectively as "ST1" & "end\_init".



LD program "LD1" Outline:



ST program “ST1” Outline:

```
(* Open Com3 with 9600 baud rate, 8 char. size, no parity, 1 stop bit at first scan cycle *)
if INIT = TRUE then
  TEMP := comopen( 3 , 9600 , 8 , 0 , 1 ) ;
end_if ;

(* Do something when K1 or K2 is pushed *)
if ( M1 = TRUE ) or ( M2 = TRUE ) then

  (* COUNT plus 1 when K1 is pushed *)
  if M1 = TRUE then
    COUNT := COUNT + 1 ;
  end_if ;

  (* COUNT plus 10 when K2 is pushed *)
  if M2 = TRUE then
    COUNT := COUNT + 10 ;
  end_if ;

  (* save COUNT value to the 5th Pos. of No.2 integer array *)
  TEMP := ARY_N_W( 2 , 5 , COUNT ) ;

  (* write one byte = 2 (hex.) to Com3 *)
  TEMP := COMWRITE( 3 , 16#2 ) ;

  (* write 1 integer (1 long integer contains 4 bytes) of Pos. 5 inside No.2 array to Com3 *)
  TEMP := COMAY_NW( 3 , 2 , 1 , 5 ) ;

  (* write one byte = 3 (hex.) to Com3 *)
  TEMP := COMWRITE( 3 , 16#3 ) ;

end_if ;
```

ST program “end\_init” Outline:

```
INIT := FALSE ;
```



### Process Operation Actions:

#### LD Program “LD1” :

Catch the rising edge status when pushbutton K1 is just pushed and save it into a internal boolean variable “M1”

Catch the rising edge status when pushbutton K2 is just pushed and save it into a internal boolean variable “M2”

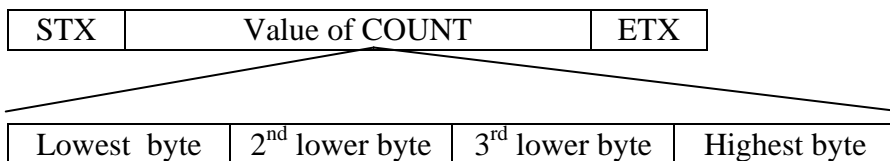
#### ST Program “ST1” :

Open Com3 of the I-8xx7 controller with 9600 baud rate, 8 char. size, no parity, 1 stop bit at the first scan cycle.

Plus “COUNT” value by 1 every time when pushbutton K1 is pushed.

Plus “COUNT” value by 10 every time when pushbutton K2 is pushed.

Send “Count” value to a PC via Com3 of the I-8xx7 controller in the below frame format.



STX : Start of frame, byte value = 2

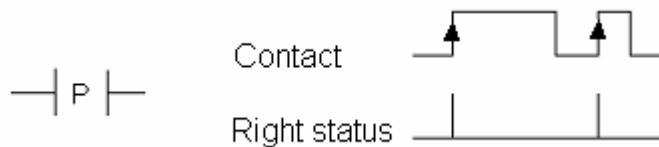
ETX : End of frame, byte value = 3

#### ST Program “end\_init” :

Set boolean variable “INIT” to FALSE at the end of the PLC scan cycle. So that “INIT” will be TRUE only at the first scan cycle.

### Function description:

“P” contact : Contact with P type means the right status will be set to a pulse TRUE when the contact is just rising from FALSE to TRUE.



Comopen(PORT, BAUD, CHAR, PARI, STOP) : To open a Com port of the I-8xx7 controller

#### Parameter

|        |         |  |
|--------|---------|--|
| PORT : | Integer | 3:COM3 ,4:COM4, ..., 20:COM20                            |
| BAUD : | Integer | baud rate, 2400, 4800, 9600, 19200, 38400, 57600, 115200 |
| CHAR : | Integer | char. size, 7 or 8                                       |
| PARI : | Integer | parity, 0:none, 1:even, 2:odd                            |
| STOP : | Integer | stop bit, 1 or 2   |

Return :    boolean            ok.: TRUE , fail: FALSE

Ary\_N\_W(NUM, ADR, DATA) : Save one long integer into an integer array.

Parameter

|        |         |  |
|--------|---------|--|
| NUM :  | Integer | save to which array (1-6)                |
| ADR :  | Integer | save to which Pos. in this array (1-256) |
| DATA : | Integer | the integer value to save                |

Return :    boolean        ok.: TRUE , fail: FALSE

ComWrite(PORT, DATA) : Write one byte to a Com port

Parameter

|        |         |                                   |
|--------|---------|-----------------------------------|
| PORT : | Integer | 3:COM3 ,4:COM4, ..., 20:COM20     |
| DATA : | Integer | the byte value (0 - 255) to write |

Return :    boolean        ok.: TRUE , fail: FALSE

ComAy\_NW(PORT, ARY\_NO, NUM, POS) : Write an integer array to a Com port

Parameter

|          |         |  |
|----------|---------|--|
| PORT :   | Integer | 3:COM3 ,4:COM4, ..., 20:COM20                    |
| ARY_NO : | Integer | the array No. to write (1-6)                     |
| NUM :    | Integer | number of integers to write (0-256)              |
| POS :    | Integer | start position inside the array to write (1-256) |

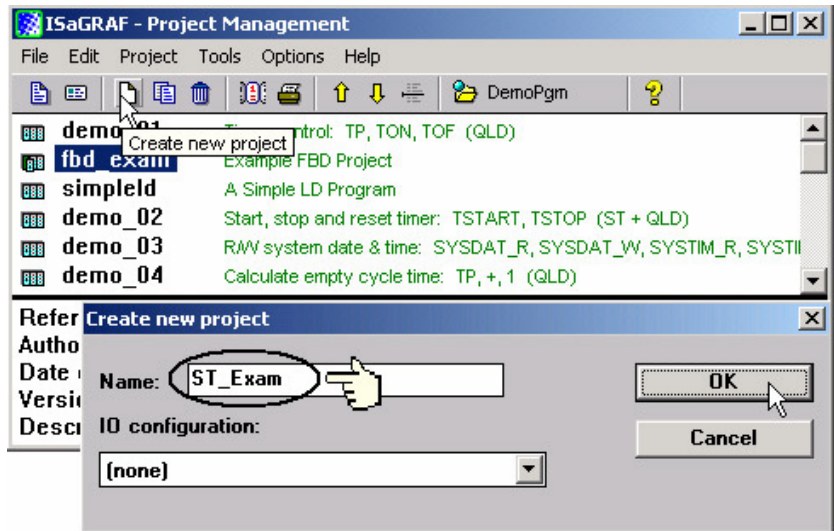
Return :    boolean        ok.: TRUE , fail: FALSE

### 2.2.1: Example ST Program

The first step is to create a new project for the example ST program.

#### Creating The Example ST Project

From the "ISaGRAF Project Management" window click on the "Create New Project" icon and enter "ST\_Exam" for the name for the example ST project.



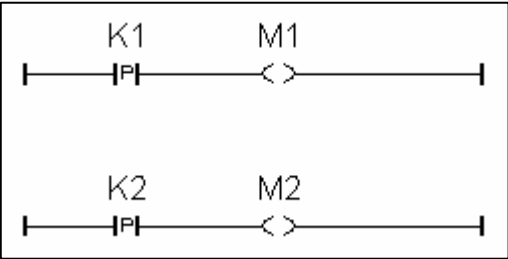
Declaring The Example ST Variables as below content

Refer to Section 2.1.1.3. "Declaring The Variables" for assistance.

| Name  | Type    | Attribute | Description  |
|-------|---------|-----------|--|
| INIT  | Boolean | Internal  | initial value at "TRUE". TRUE means 1 <sup>st</sup> scan cycle           |
| K1    | Boolean | Input     | The first pushbutton on the front panel of the I-8xx7                    |
| K2    | Boolean | Input     | The second pushbutton on the front panel of the I-8xx7                   |
| M1    | Boolean | Internal  | Indicate pushbutton K1 is just pushed.                                   |
| M2    | Boolean | Internal  | Indicate pushbutton K2 is just pushed.                                   |
| TEMP  | Boolean | Internal  | A boolean variable for temporary use.                                    |
| COUNT | Integer | Internal  | A integer value generated by push K1 & K2<br>initial value is set at "0" |

Creating a LD program "LD1" with the below content.

Refer to Section 2.1.1.4. and 2.1.1.5 for assistance.



Follow the same steps as 2.1.1.6. to create a ST program “end\_init” with the below content.

**INIT := FALSE ;**

Creating a ST program “ST1” with the below content.  
Refer to Section 2.1.1.6. for assistance.

```
(* Open Com3 with 9600 baud rate, 8 char. size, no parity, 1 stop bit at first scan cycle *)
if INIT = TRUE then
  TEMP := comopen( 3 , 9600 , 8 , 0 , 1 ) ;
end_if ;

(* Do something when K1 or K2 is pushed *)
if ( M1 = TRUE ) or ( M2 = TRUE ) then

  (* COUNT plus 1 when K1 is pushed *)
  if M1 = TRUE then
    COUNT := COUNT + 1 ;
  end_if ;

  (* COUNT plus 10 when K2 is pushed *)
  if M2 = TRUE then
    COUNT := COUNT + 10 ;
  end_if ;

  (* save COUNT value to the 5th Pos. of No.2 integer array *)
  TEMP := ARY_N_W( 2 , 5 , COUNT ) ;

  (* write one byte = 2 (hex.) to Com3 *)
  TEMP := COMWRITE( 3 , 16#2 ) ;

  (* write 1 integer (1 long integer contains 4 bytes) of Pos. 5 inside No.2 array to Com3 *)
  TEMP := COMAY_NW( 3 , 2 , 1 , 5 ) ;

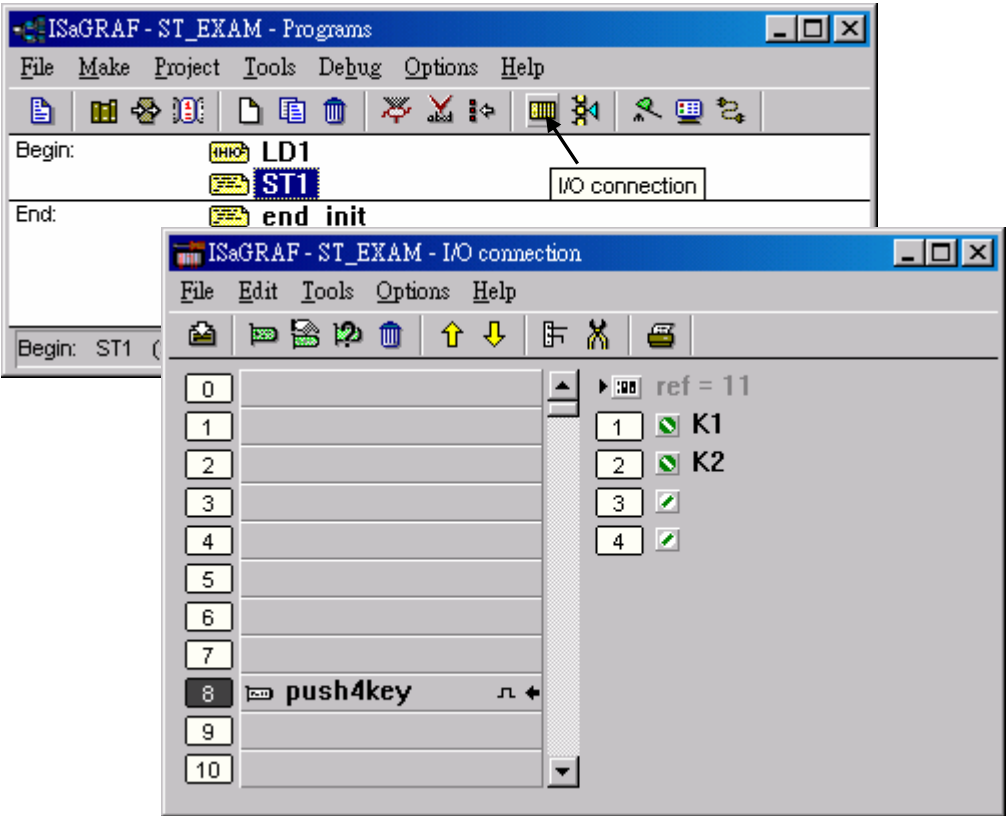
  (* write one byte = 3 (hex.) to Com3 *)
  TEMP := COMWRITE( 3 , 16#3 ) ;

end_if ;
```

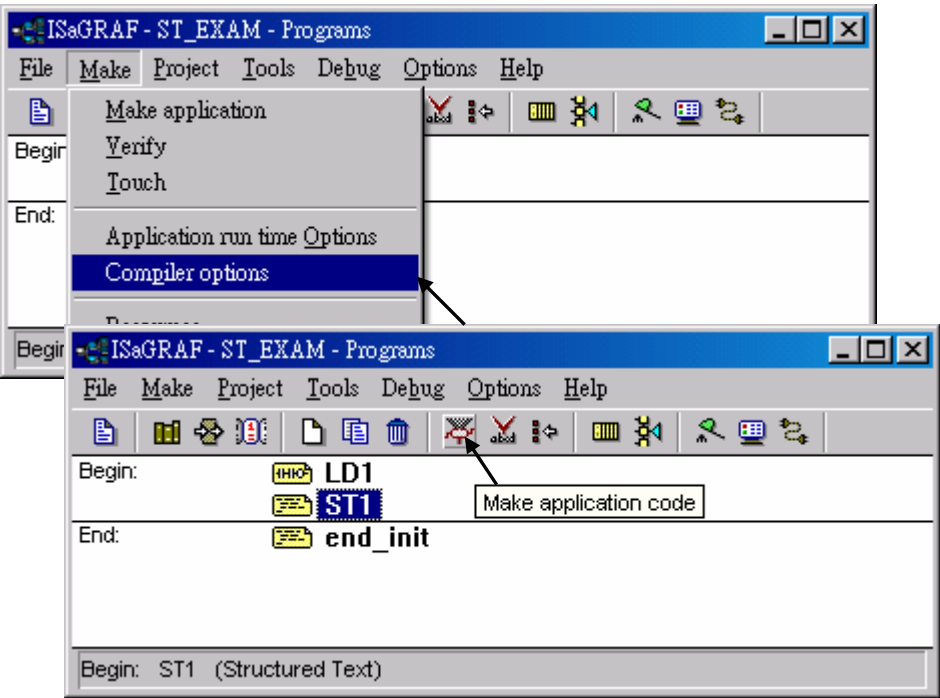
#### **IMPORTANT NOTE**

Each ST statement line **MUST** end with a semi-colon ";" as shown above. After entering in the above example program remember to click on the "Save" icon to save the program, then click on "Exit".

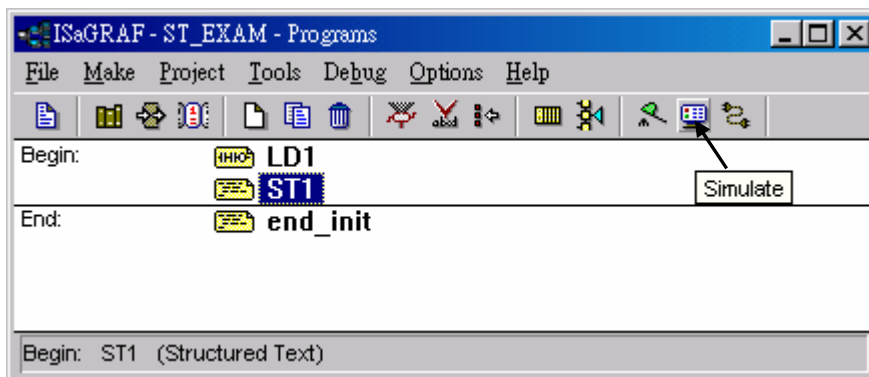
Use the similar procedure for the "Connecting I/O" as detailed in Section 2.1.2



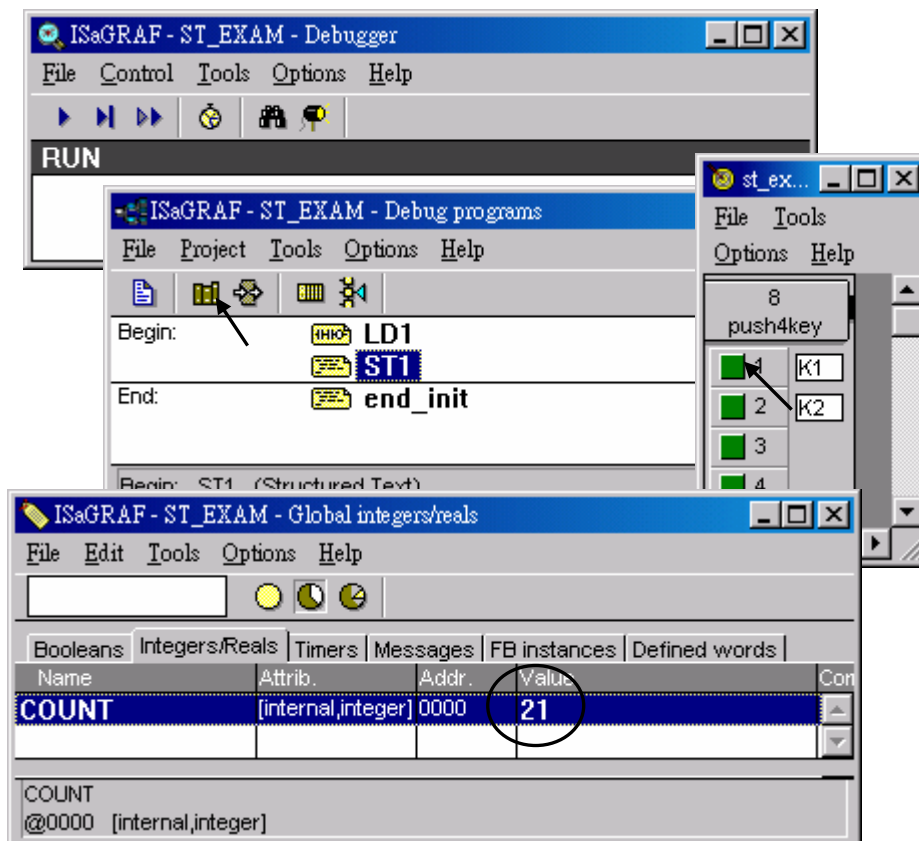
Use the similar procedure for the "Compilling the project" as detailed in Section 2.1.3



After compiling the example ST project click on the "Simulate" icon to observe the ST program running.



You may open the dictionary window to see the "COUNT" value. Click on "K1" or "K2", you will see the "COUNT" value is changed.



You can now download this example project to the I-8xx7 controller system. Please follow the same procedure as outlined in Section 2.1.5 .

After downloading to the controller, the program will send 6 bytes via Com3 of the controller whenever K1 or K2 is pushed. If you have your RS232 monitoring program running on your PC, you can connect Com3 to your PC to see how it works.

## 2.3: A Simple Function Block Diagram (FBD) Program

The "Function Block Diagram (FBD) is a graphical programming language that allows a programmer to build complex procedures by taking existing "Functions" from the ISaGRAF library and "Wiring" them together graphically to create powerful process control applications.

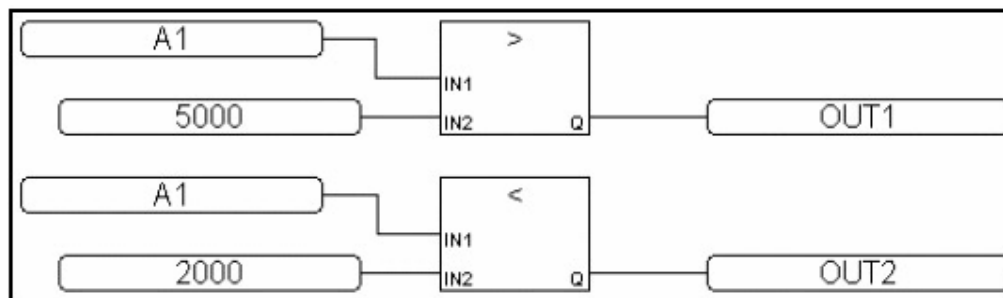
The following section details how to build a "Function Block Diagram" program with ISaGRAF. Function Block Diagram programs are extremely useful for managing several control process programs from a single source.

Example FBD Control Specification:

The following details the variables that will be used in our example Function Block Diagram program.

| Name | Type    | Attribute | Description   |
|------|---------|-----------|---|
| OUT1 | Boolean | Output    | High alarm  |
| OUT2 | Boolean | Output    | Low alarm   |
| A1   | Integer | Internal  | Simulate a temperature input, <b>initial value is 0</b> |

FBD Program Outline:



FBD Program Action:

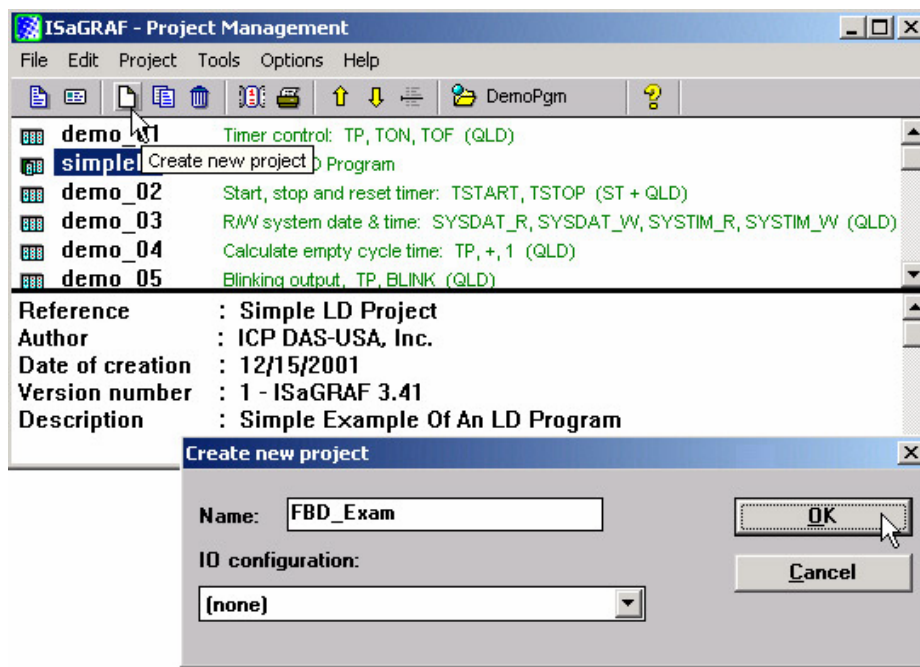
If "A1" > 5000, output "OUT1" is "TRUE".  
If "A1" < 2000, output "OUT2" is "TRUE".  
Other situation, output "OUT1" and "OUT2" are "FALSE"

### 2.3.1: Programming The Example FBD Program

Creating a Function Block Diagram (henceforth referred to as "FBD") program is very similar to creating a LD program as outlined in Section 2.1. The following steps detail how easy it is to create a FBD program.

Creating a New FBD Project

From the "ISaGRAF Project Management" window click on the "Create New Project" icon and enter the name "FBD\_Exam".

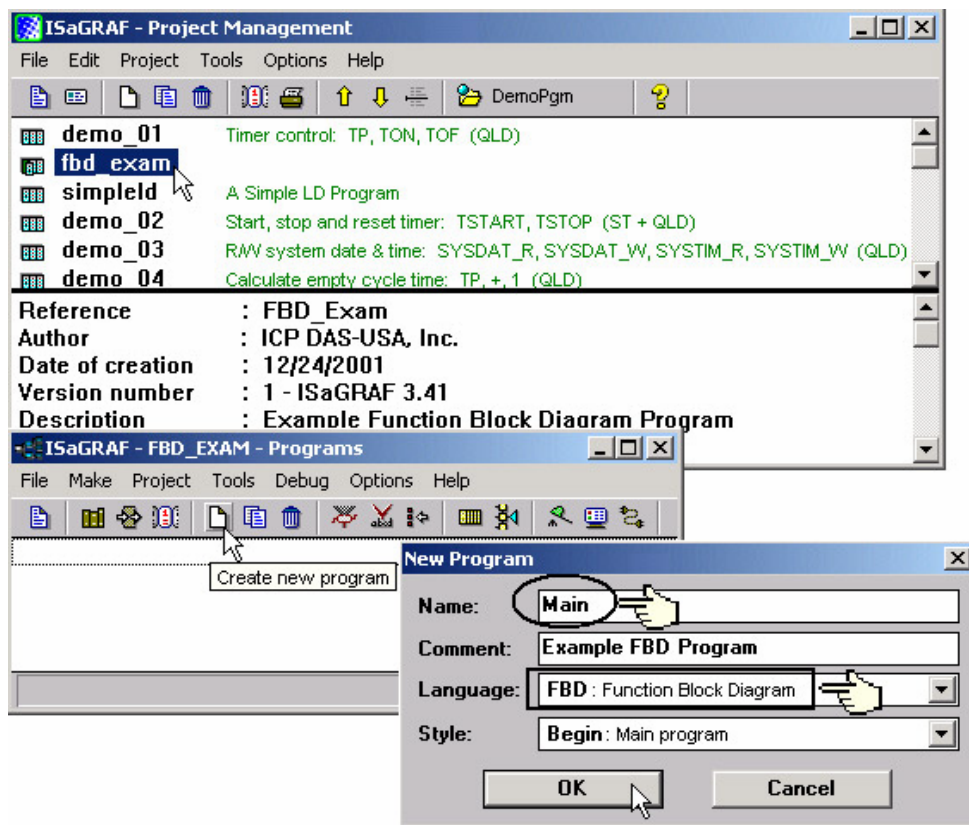


After you have created the new FBD project, double click on the "FBD\_Exam" name in the "ISaGRAF Project Management" window to open the new FBD project. Click on the "Create New Program" icon in the "ISaGRAF Programs" window, which will open the "New Programs" window.

In the "New Programs" window enter in the name field "Main", and for "Language" make sure the "FBD – Function Block Diagram" is selected. You can add a comment about your program also while in the "New Program" window, but it is not mandatory.

Once you have entered in all the information in the "New Programs" window click on the "OK" button.



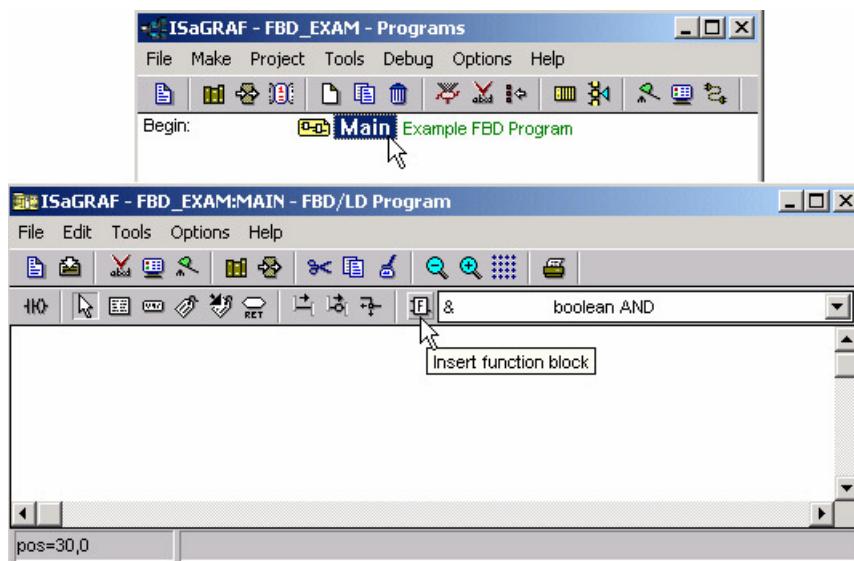


### Declaring The Variables

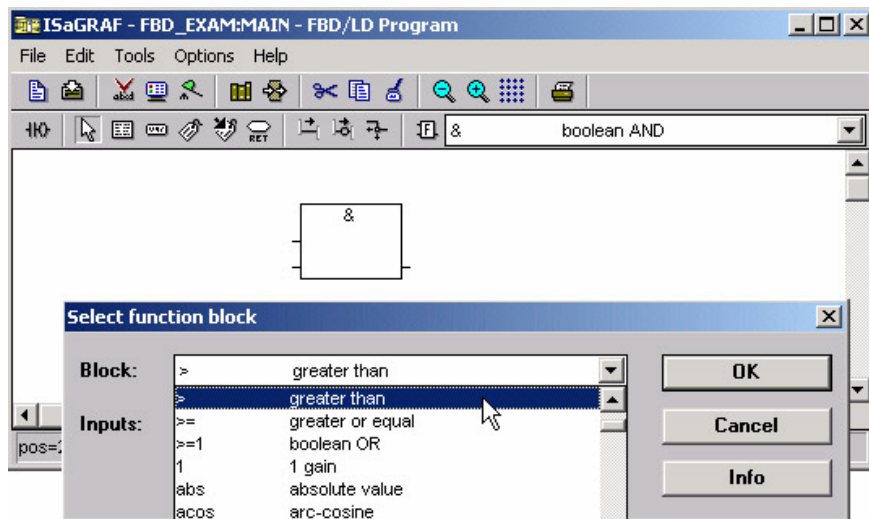
For our example FBD program we are going to declare three variables. The variables to be used are "OUT1", "OUT2", and an integer variable called "A1". Declaring variables for the FBD program is like declaring variables for the LD program. Refer to Section 2.1.1.3 – "Declaring The Variables" to review the variable declaration process.

### Editing The FBD Program

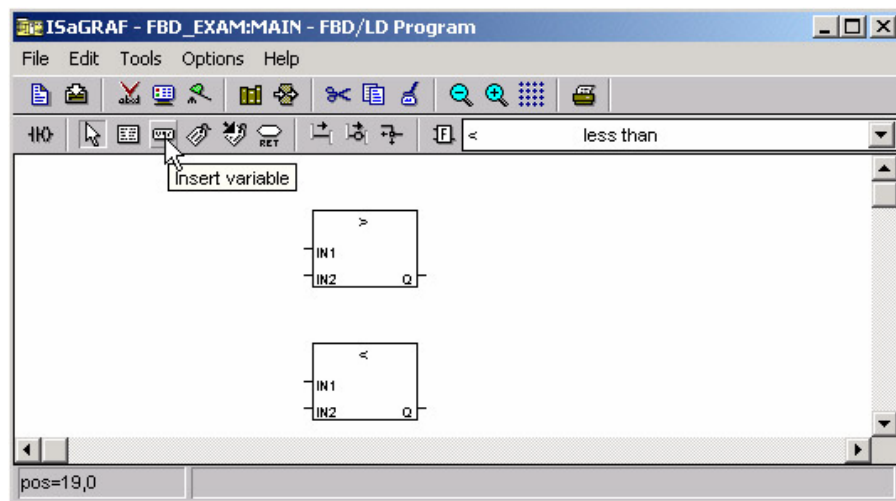
To create and edit the example FBD program, double click on word "MAIN" in the "ISaGRAF Programs" window, and then click on the "Insert Function Block" icon as shown below.



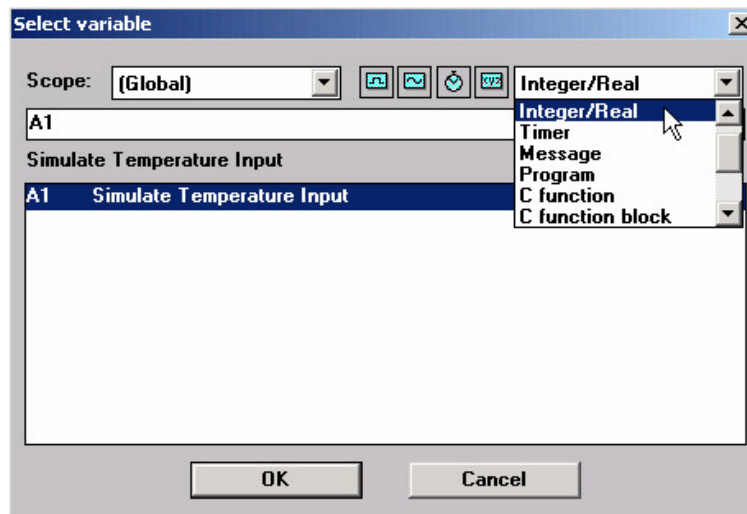
Move the cursor to approximately the middle of the "ISaGRAF FBD/LD Program" window and click the mouse one time to add the first function block. Next, double click on the block to select "> Greater Than". For more information regarding any of the function blocks available in the ISaGRAF program just click on "Info" button.



Using the same procedure as described above, add a "< Less Than" function block below the "Greater Than" function block.



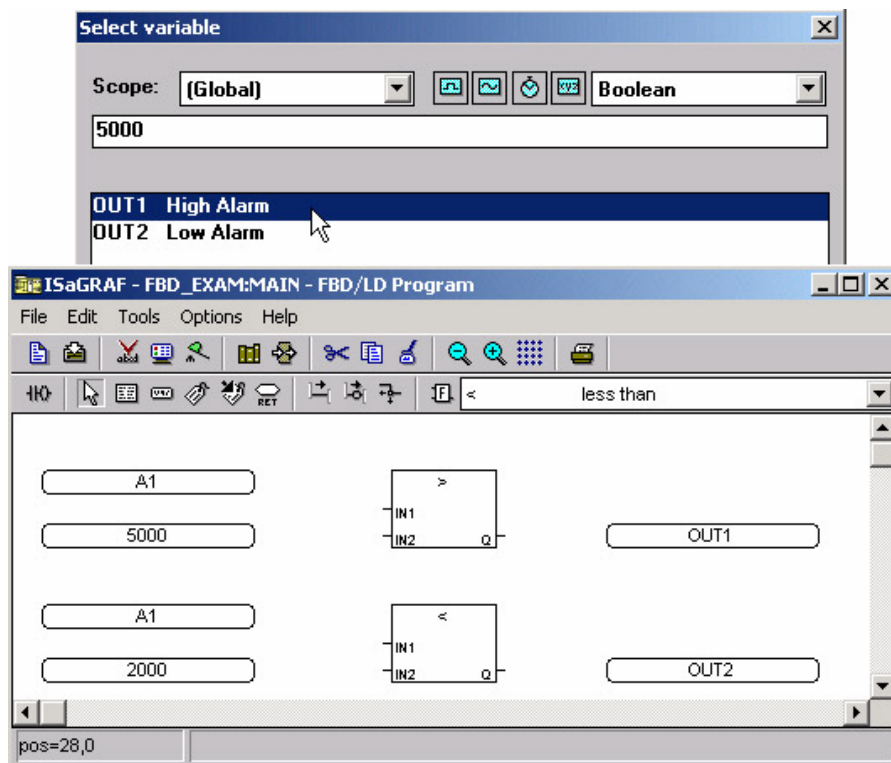
Now it is time to add the program variables to the FBD example program. Click on the "Insert Variable" icon as shown above, and then click on "Integer/Real" from the "ISaGRAF Select Variable" window. This will cause the variable "A1" to appear in the "ISaGRAF Select Variable" to appear.



Double click on the highlighted "A1 Simulate Temperature Input" which will then place the variable "A1" inside of the "ISaGRAF FBD/LD Program" window. Repeat the same process to add a second "A1" variable.

Click on the "Insert Variable" icon to add the "OUT1" and "OUT2" variables to the right of the function blocks as shown below.

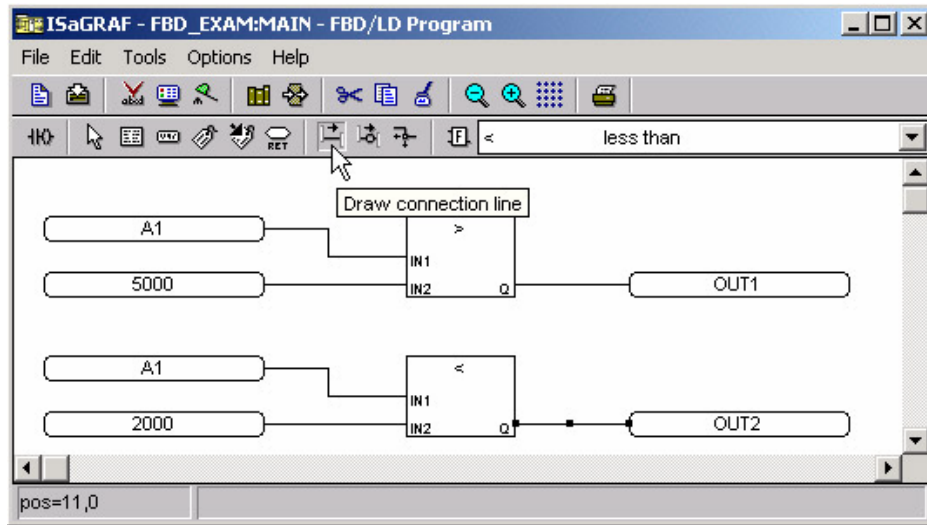
Lastly, add two additional variables, the first is a constant of "5000" and place it below the first "A1" variable, then create a second constant of "2000", and place it below the second "A1" variable.



Your "ISaGRAF FBD/LD Program" window should now look like the above example. Remember, we have added a total of six variables to the program. We have added the "A1" variable twice, the "OUT1"

variable, the "OUT2" variable, one constant called "5000" and another constant called "2000" to the FBD example program.

The last task to accomplish is making the connection between each of the variables (and constants) and the function blocks. Click on the "Draw Connection Line" icon and draw a line between each of the variables and function blocks as shown below.

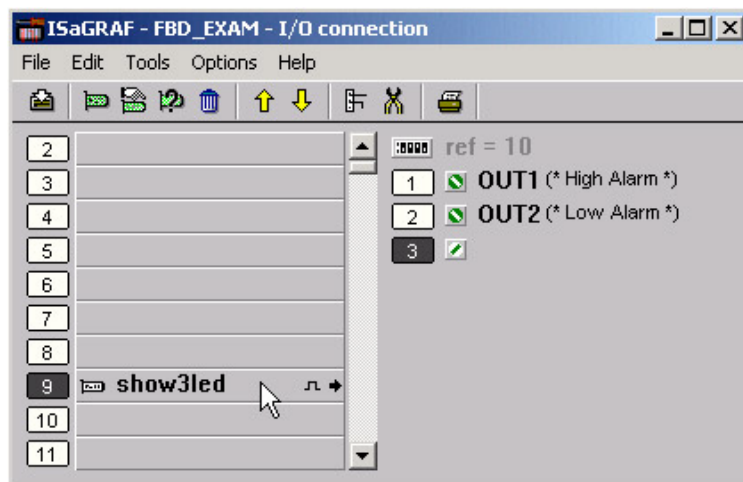


The top "A1" variable should connect to the "IN1" of the "> Greater Than" function block, the "5000" constant to the "IN2" of the "> Greater Than" function block, the bottom "A1" variable to the "IN1" of the "< Less Than" function block, and the "2000" constant to the "IN2" of the "< Less Than" function block.

Lastly, connect the "Q" of the "> Greater Than" function block to the "OUT1" variable, and the "Q" of the "< Less Than" function block to the "OUT2" variable.

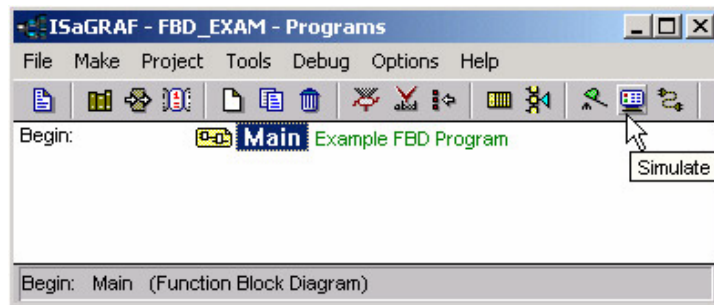
#### Connecting The I/O & Compiling The Project

Follow the same procedure as outlined in Section 2.1.2 and 2.1.3 for connecting the I/O and compiling the FBD example program. The "ISaGRAF I/O Connection" window should look like the example below.



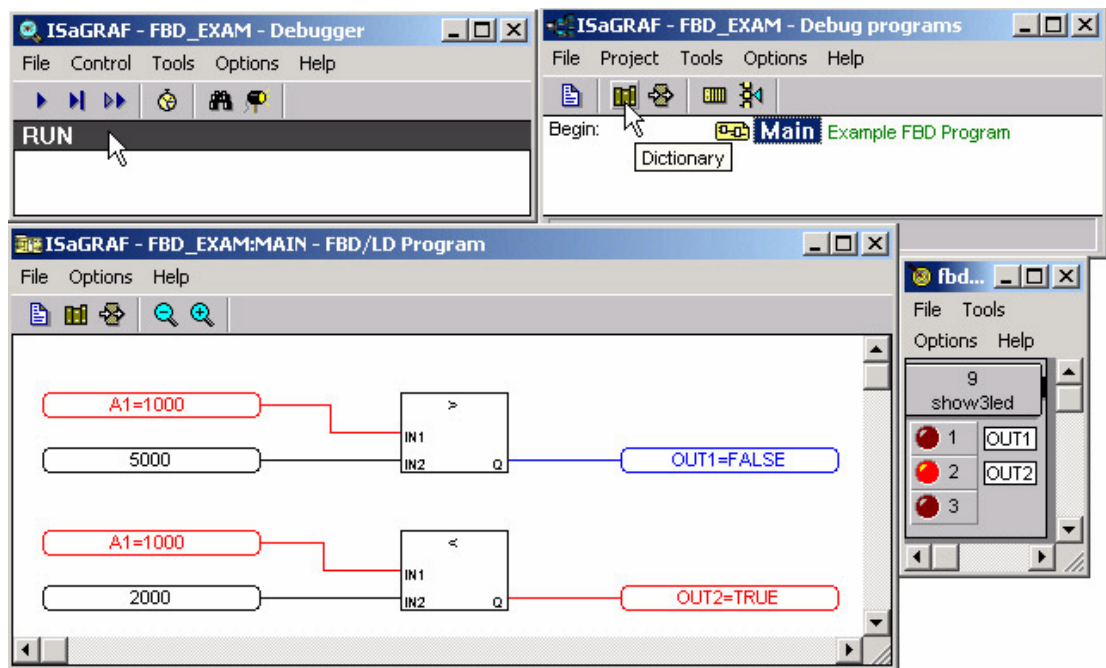
### 2.3.2: Simulating The FBD Program

You can now run the "Simulate" on the example FBD program by clicking on the "Simulate" icon in the "ISaGRAF Programs" window.

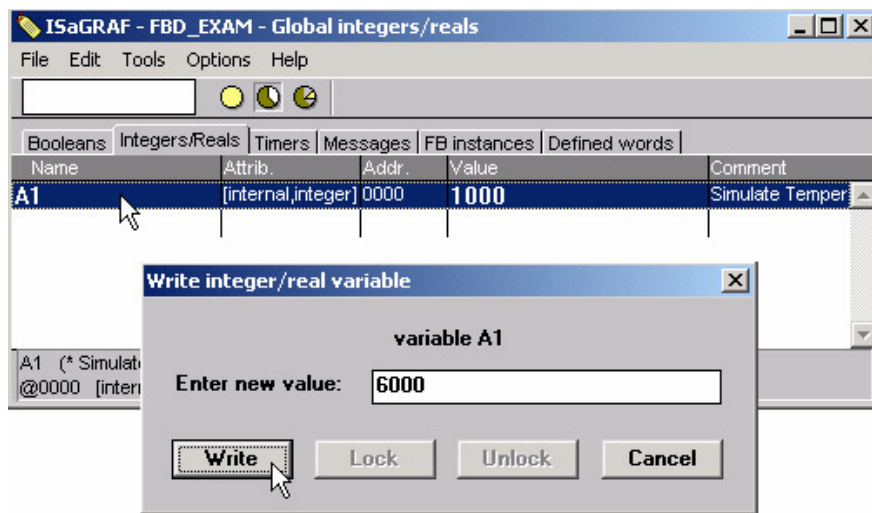


When you click on the "Simulate" icon the "ISaGRAF Debugger" window, the "ISaGRAF Debug Programs", and the "I/O Simulator" window will now open. If you double click on "MAIN" in the "ISaGRAF Debug Programs" window the "ISaGRAF FBD/LD Program" window will open showing the state of the program.

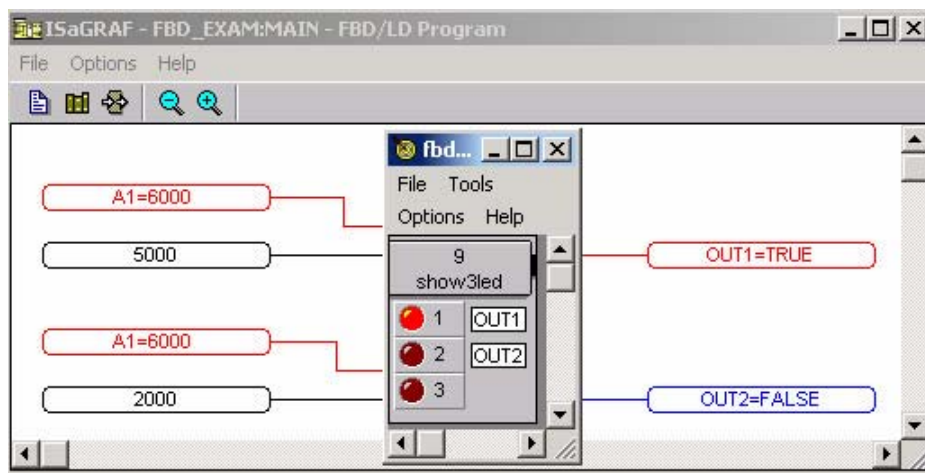
Notice that because the "A1" variable is less than 2000 (currently set to 1000 in the example below) that the "OUT2" output is currently true and the "OUT1" output is false.



To further test the example FBD program, click on the "Dictionary" icon in the "ISaGRAF Debug Programs" window to open the "Global Dictionary" window, and click on the "Integer/Real" tab. Click on the highlighted "A1" and the "Write Integer/Real Variable" will open.



Type in "6000" in the "Enter New Value" field and click on the "Write" button. Now the following changes will be observed.



You can now download the example FBD program to the I-8xx7 controller system. Follow the same procedure as outlined in Section 2.1.5 for downloading the program to the I-8xx7 controller system.



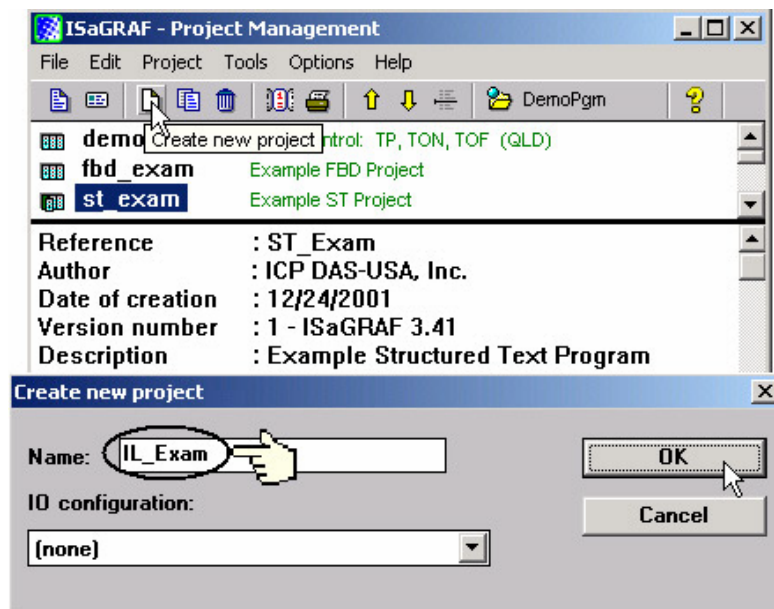
## 2.4: A Simple Instruction List (IL) Program

Instruction List (IL) programming is a low level programming language consisting of a list of instructions. Each instruction always relates to the **current result** (or **IL register**) and must begin on a new line and must contain an **operator**. The operator indicates the operation that must be made between the current value and the **operand**. The result of the operation is stored again in the result.

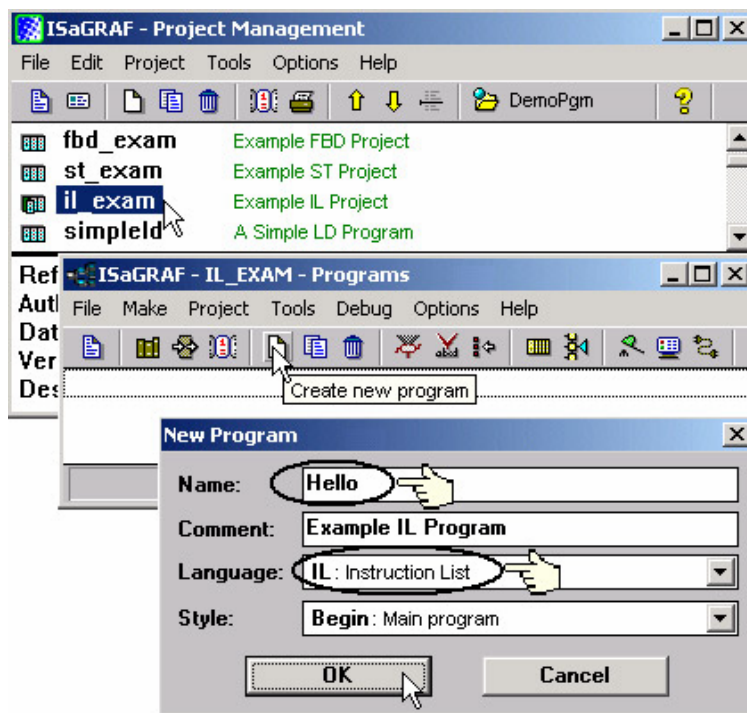
Instruction List (IL) programming requires adherence to a strict programming format that must be followed. Each instruction must begin on a new line, it must contain an **operator**, completed with optional modifiers and if necessary, for the specific operation, one or more operands, separated with commas (","). A **label** followed by a colon (":") may precede the instruction. If a comment is attached to an instruction, it must be the last component of the line. Comments must always begin with (\*) and end with (\*). The following is an example of a comment in IL; (\* **place comment here** \*).

This section describes how to program an Instruction List (henceforth referred to as IL) program. This IL program has the same program specification as the FBD program as outlined in Section 2.3.

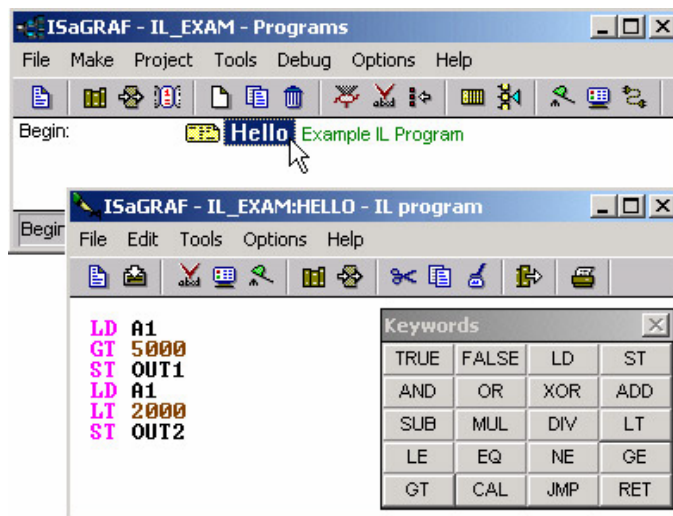
The first step to creating an IL program is to create an IL project. This is accomplished in the same manner as creating any other ISaGRAF project.



For the purpose of this example IL program I have created a new IL project name of "IL\_Exam". Click on the "OK" button and the "ISaGRAF Project Management" window will appear with the new project name. Double click on the "IL\_Exam" name and the "ISaGRAF Programs" window will appear. Click on the "Create New Program" icon and the "New Program" window will appear. Enter "Hello" in the name field (and you can add a program comment if desired) and make sure to select "IL: Instruction List" from the language field, click on the "OK" button when you are done.



When you click on the "OK" button the "ISaGRAF Programs" window will open. Double click on "Hello" and the "ISaGRAF IL Program" window will open.

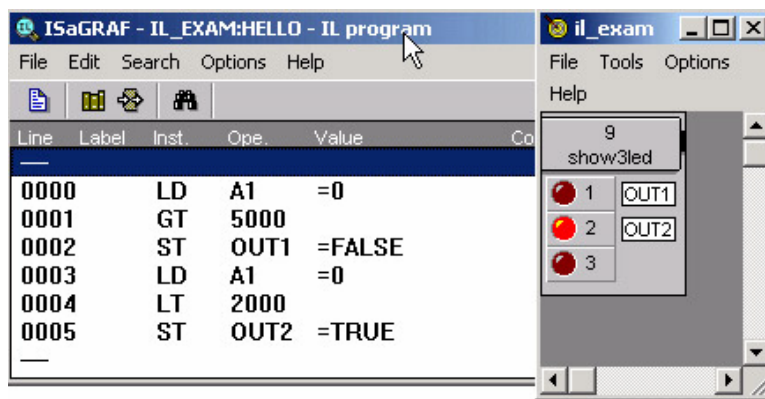


### Declaring The Example IL Variables

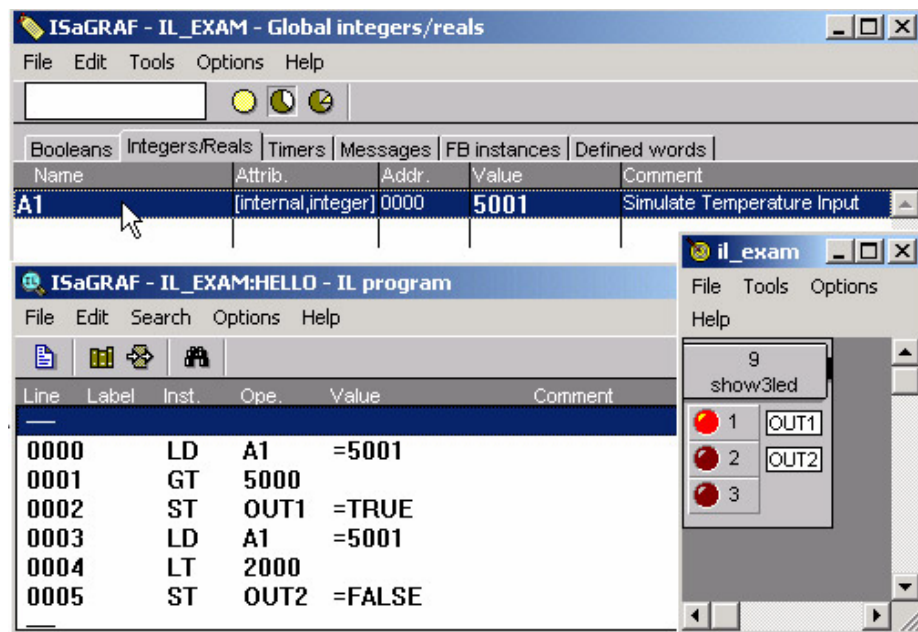
This example IL program uses the same variables as the example FBD program, "OUT1", "OUT2" and the integer variable "A1". Refer to Section 2.1.1.3 "Declaring The Variables" for assistance. Use the same procedure for the "Connecting I/O" and "Compiling" the program as detailed in Section 2.1.2 and 2.1.3, and use the same procedure to "Simulate" the program as detailed in Section 2.3.2.

When you have connected the I/O and compiled the example IL program, click on the "Simulate" icon and the following window will appear.





Because the variable "A1" value is 0, "OUT1" is set to false and "OUT2" is set to true. Change the value of "A1" to a value greater than 5001 and you will see that "OUT1" is set to true and "OUT2" is set to false.



## 2.5: A Simple Sequential Function Chart (SFC) Program

A Sequential Function Chart (SFC) program is a graphical programming language used to describe **sequential operations**. The process is represented as a set of defined **steps**, linked by **transitions**. A **Boolean condition** is attached to each transition, and **actions** with the steps are detailed by using other languages such as ST, IL, LD and FDB.

An SFE program is a graphical set of **steps** and **transitions**, linked together by **oriented links**. Multiple connection links are used to represent divergences and convergences. Some parts of the complete program may be separated and represented in the main chart by a single symbol, call **macro steps**. The basic graphic rules for an SFC program are:

1. A Step CANNOT Be Followed By Another Step
2. A Transition CANNOT Be Followed By Another Transition

The basic components (graphical symbols) of the SFC programming language are: steps and initial steps, transitions, oriented links, and jumps to a step.

This section details how to build a Sequential Function Chart (henceforth referred to as SFC) program.

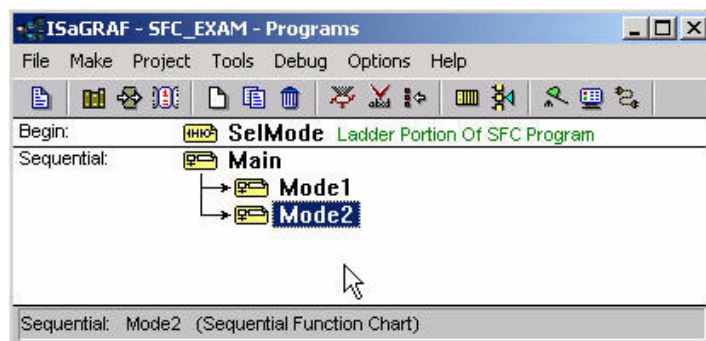
Example SFC Control Specification:

The following details the variables that will be used in our example SFC program.

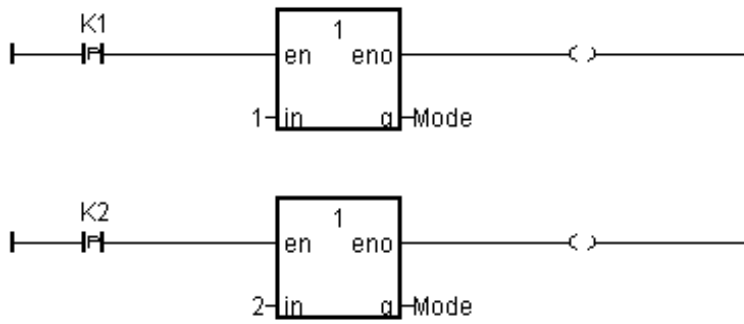
| Name | Type    | Attribute | Description  |
|------|---------|-----------|--|
| OUT1 | Boolean | Output    | Output 1   |
| OUT2 | Boolean | Output    | Output 2   |
| K1   | Boolean | Input     | Mode 1 button input                                      |
| K2   | Boolean | Input     | Mode 2 button input                                      |
| TMR1 | Timer   | Internal  | Switch time of output, initial value is "T#1s"           |
| Mode | Integer | Internal  | 1 means mode1 , 2 means mode2, <b>initial value is 1</b> |

The SFC Program Outline:

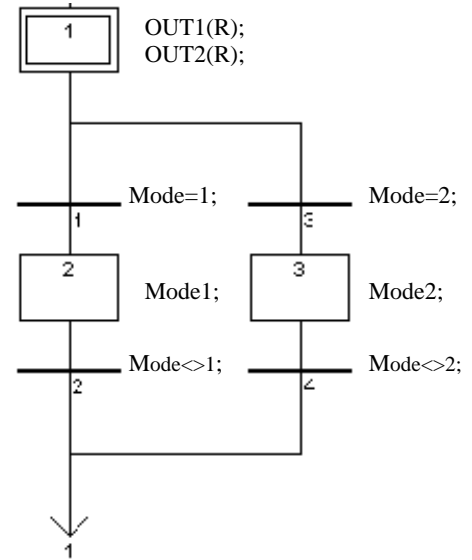
When you have completed the "ISaGRAF Programs" window, it should look like the following:



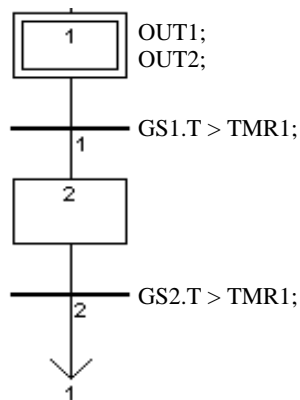
### LD Program "SelMode"



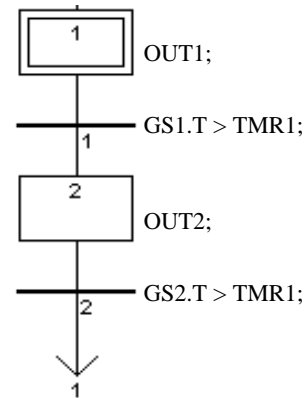
### SFC Program "Main"



### SFC Child Program "Mode1"



### SFC Child Program "Mode2"

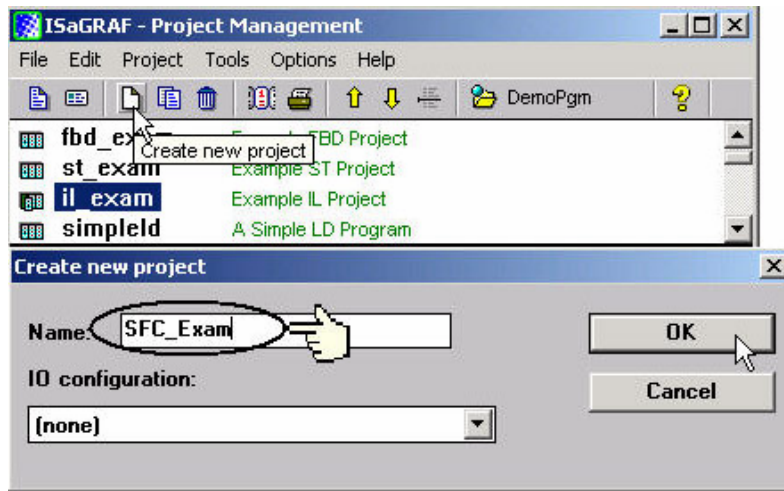


### SFC Program Action:

1. When "K1" is pressed, run the "Mode1" program.
2. When "K2" is pressed, run the "Mode2" program.

### 2.5.1: Programming The Example SFC Program

The procedure for creating the example SFC program is the same as outlined in Section 2.1. You must remember to declare the variables "K1", "K2", "OUT1", "OUT2", "TMR1" and "MODE". The following illustrates creating the new SFC project.



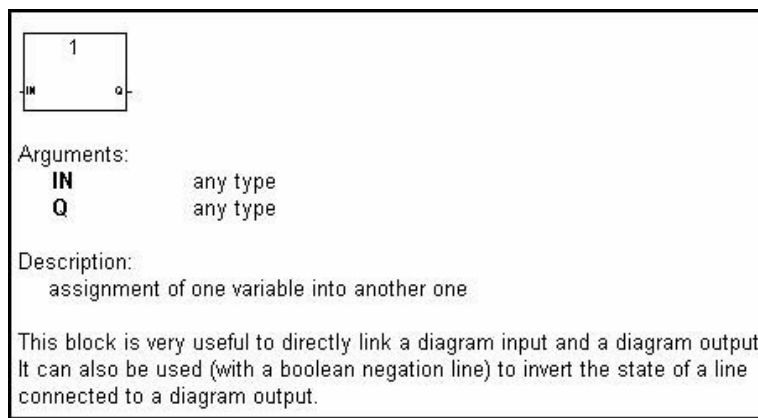
After creating the new SFC project, the next step is to create an LD program named "SelMode" as illustrated below.



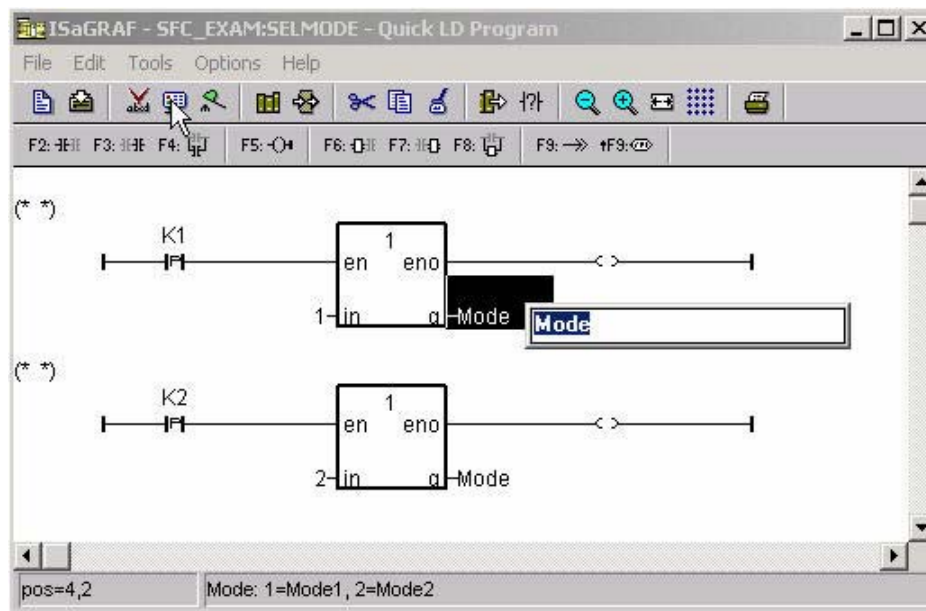
When you click on the "OK" button the "ISaGRAF Quick LD Program" window will open. Add the instructions as shown in the example below.

#### IMPORTANT NOTE:

The example SFC program uses a function block that has not been used throughout the manual. We will be adding the "1 Gain" function block to our LD program.

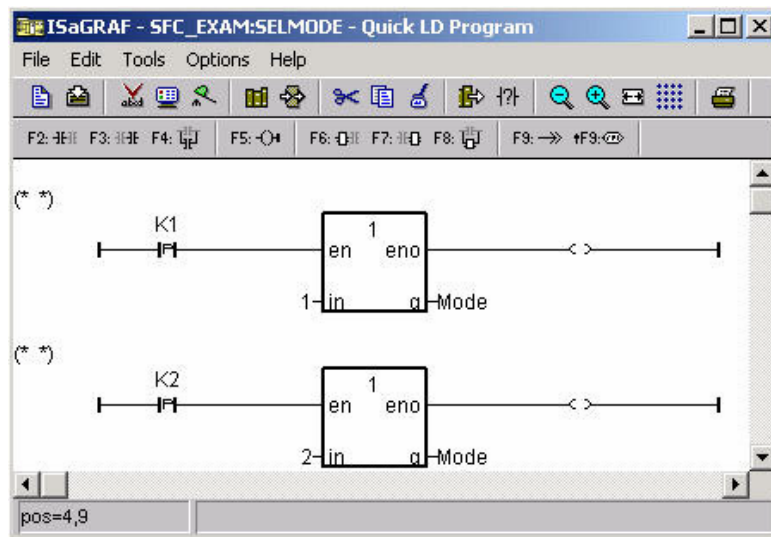


Even though the "EN" (input) and "ENO" (output) arguments are not shown in the above example, they will be added when you place the "1 Gain" function block in the program.

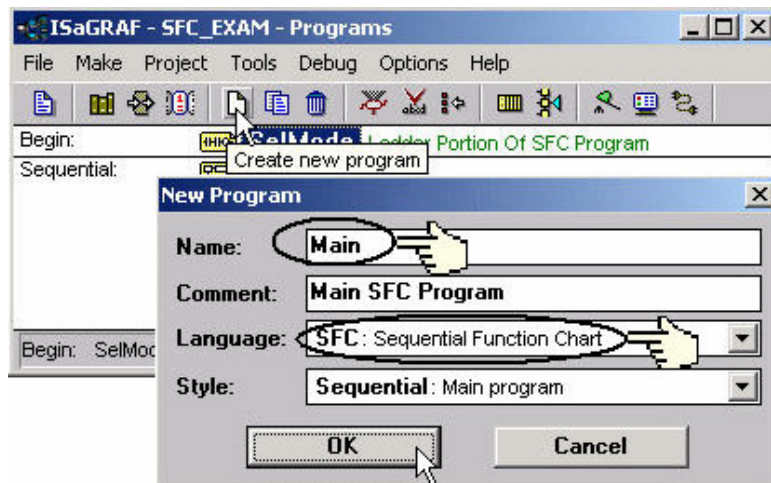


You will need to change the "K1" and "K2" contacts type to "P". The "P" contact (Positive) enables a Boolean operation between a connection line state and the rising edge of a Boolean variable. Place the cursor to the right of the "Q" and click once, then type in "Mode" for both lines of logic. Place the cursor to the left of the "IN" on the top "1 Gain" function block, click once and enter a "1". Do the same for the second LD line and enter a value of "2", then click once on the "Q" and enter in "Mode".

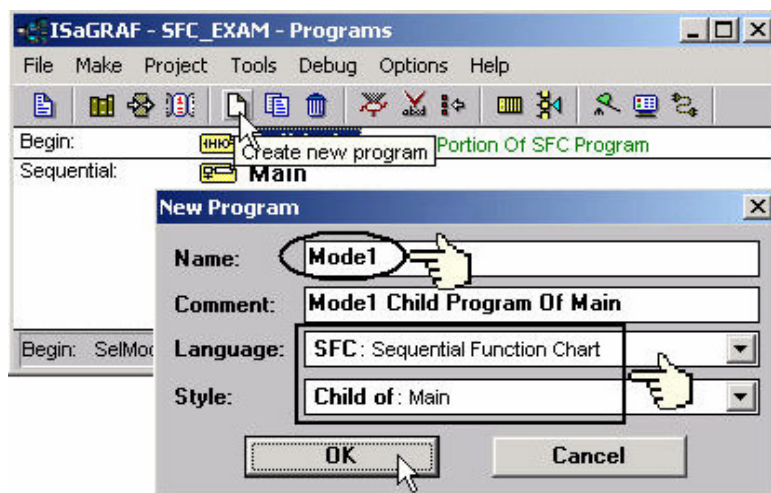
When you are finished editing the "ISaGRAF Quick LD Program" window it should look like the below example.



The next step is to create a new SFC program called "Main".

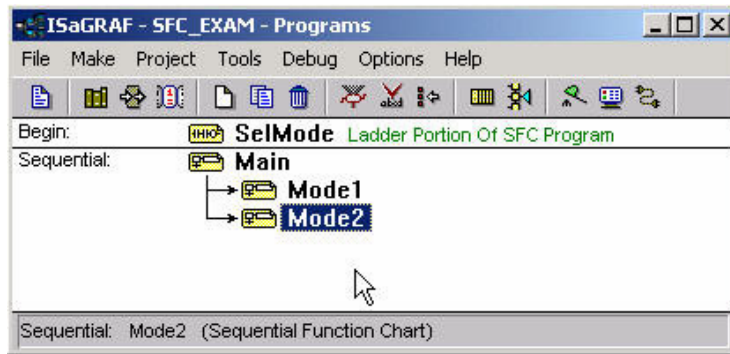


The next step is to create a "CHILD" program called "Model".



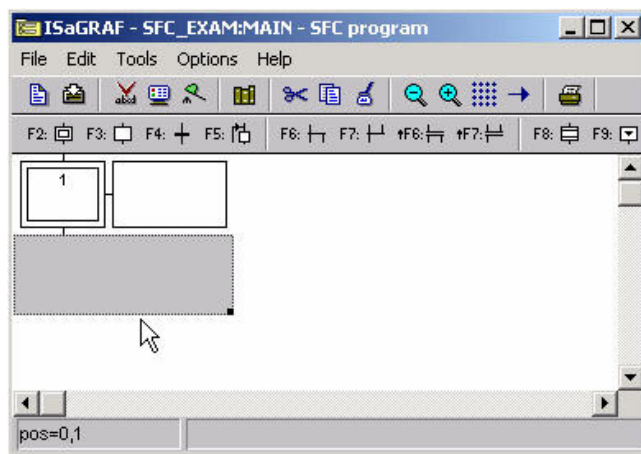


Follow the same procedure to create a second "CHILD" program named "Mode2". When you are completed the "ISaGRAF Programs" window should look as follows.

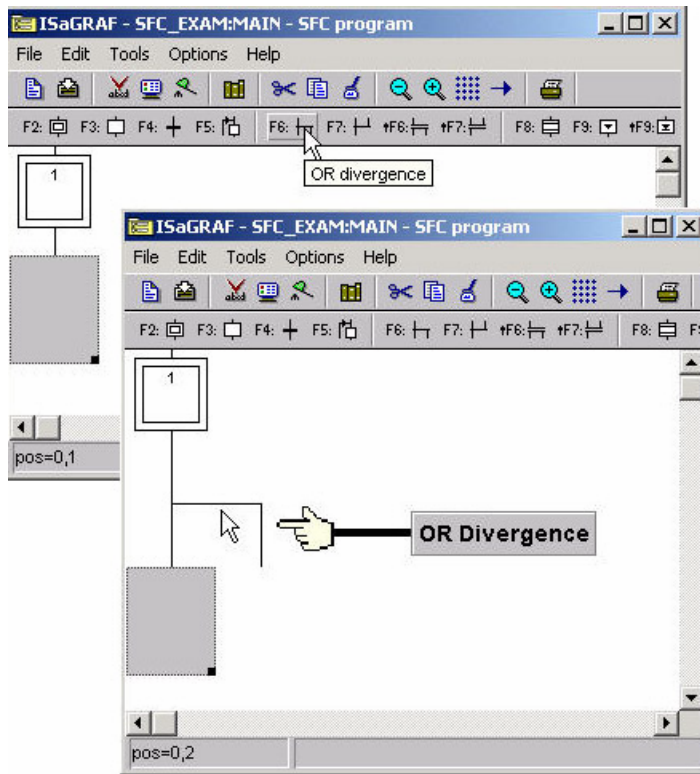


### 2.5.2: Editing The SFC Program

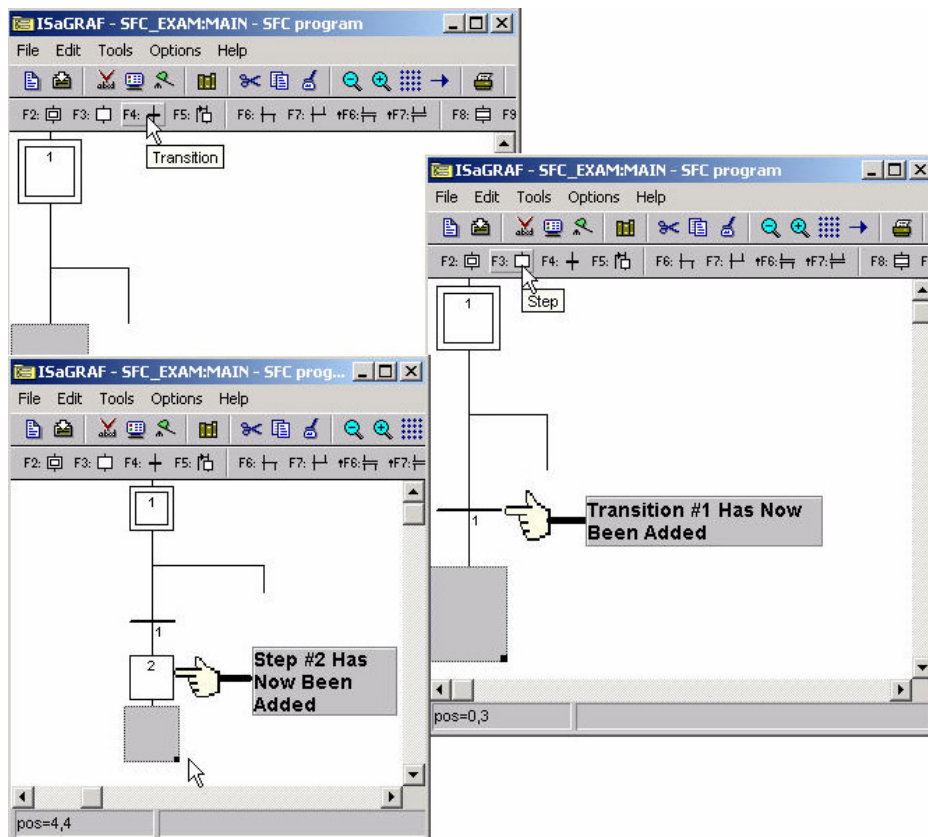
To begin editing the example SFC program double click on "Main" in the sequential portion of the "ISaGRAF Programs" window and the "ISaGRAF SFC Program" window will appear.



You will note an additional box to the right of the initial step box. This box will contain the code for each of the steps and transitions in the example SFC program. The "code box" is not required during the initial programming so you can get rid of it temporarily by clicking on the black dot in the gray box area below the initial step and resize the window to approximately the size of the initial step box.



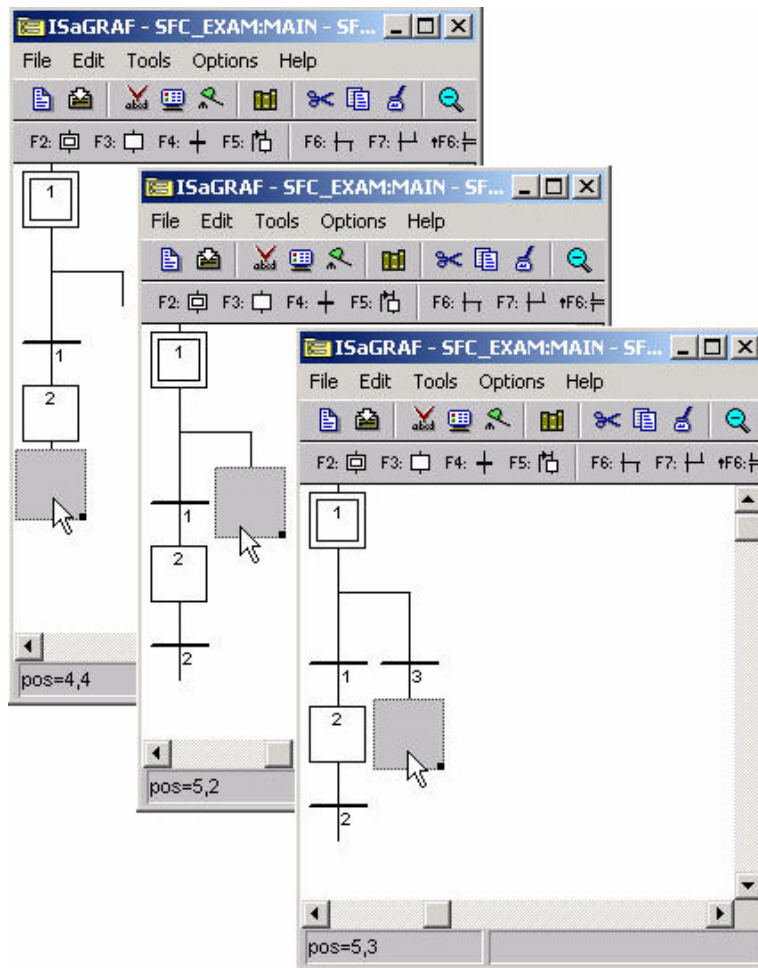
The gray box will move down automatically when you click on the "OR Divergence" icon. The next step is to click on the "Transition" icon to create "Transition 1" and then the "Step" icon to create "Step 2" as shown below.



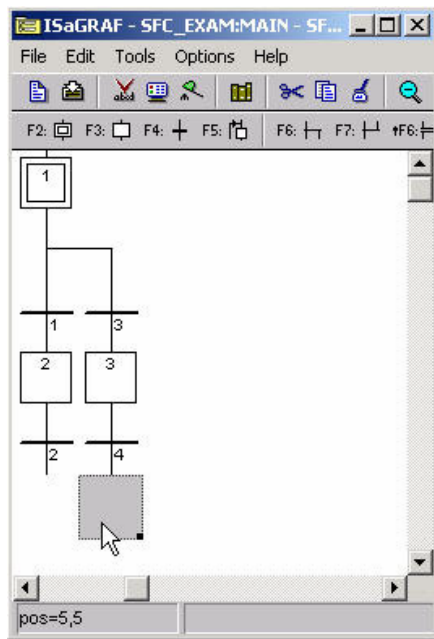


With the gray box below "Step 2" click on the transition button to add a second transition (transition #2) to the example SFC program. After adding the second transition below "Step 2", click directly below the "OR Divergence" so that the gray box is now placed there. Click on the transition icon again with the gray box below the "OR Divergence" to add a third transition (transition #3).

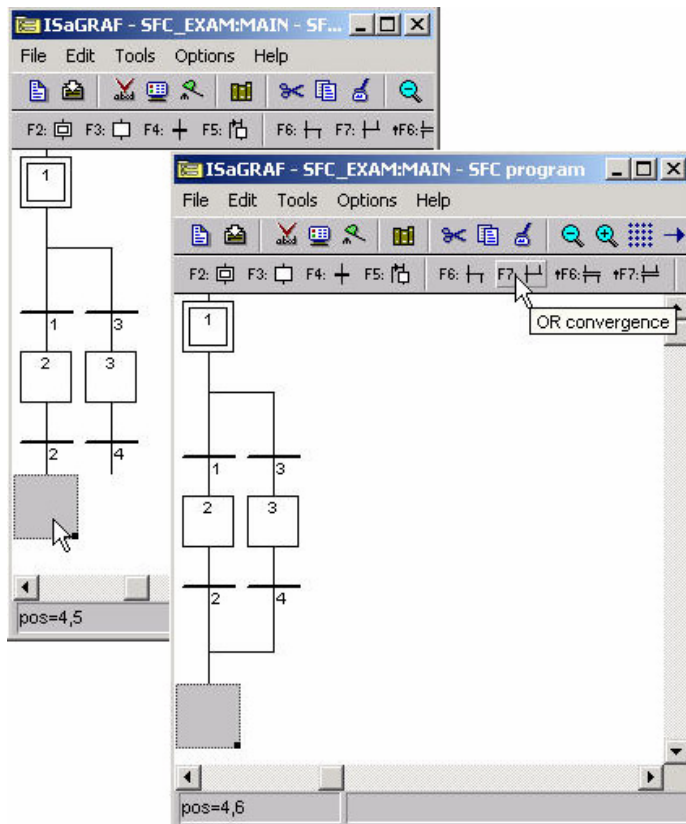
When you have completed these tasks your SFC program should now look like the third SFC picture below.



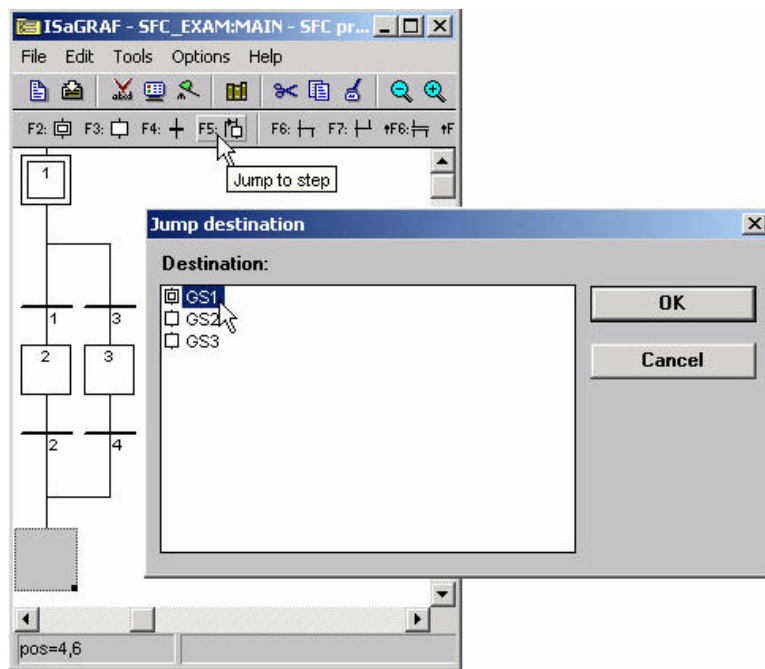
From where the gray box is currently click on the "Step" icon to add Step #3, and then with the gray box below the newly created step #3 click on the transition icon to add a fourth transition (transition #4) to the example SFC program. Your SFC program should now look like the below example.



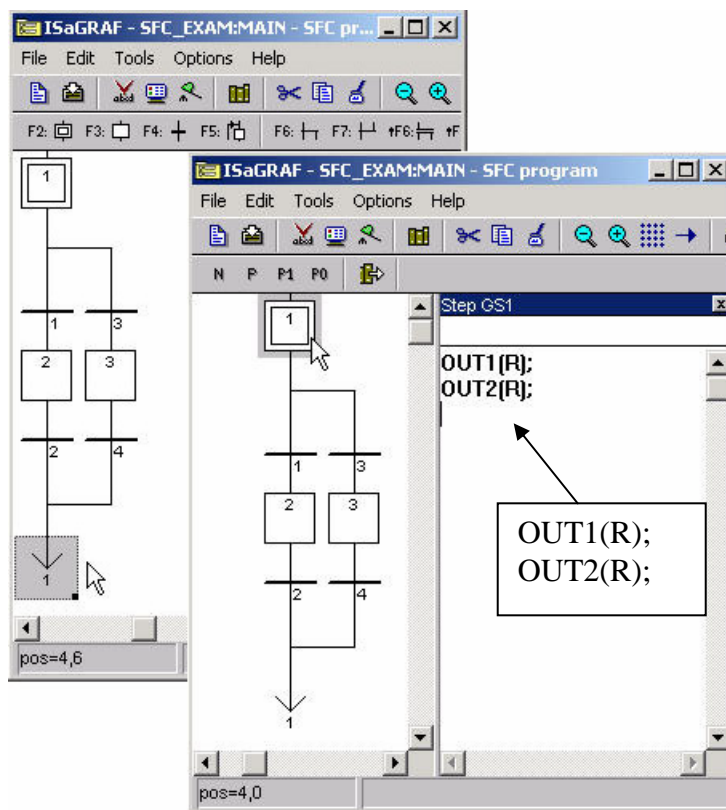
Now click the gray box below transition #2 and click on the "OR Convergence" (F7) icon.



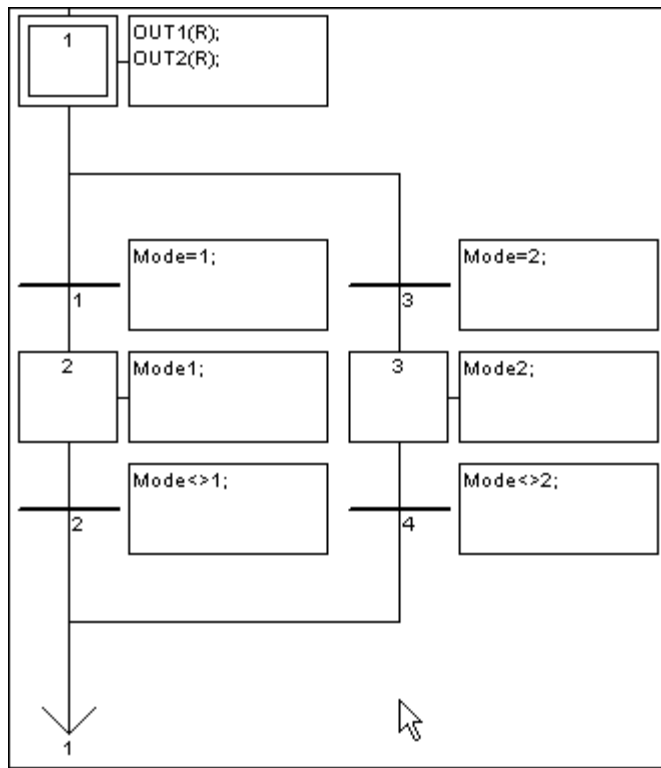
Now click on the "Jump To Step" (F5) icon, this will open the "Jump Destination" window. Double click on the "GS1" label in the "Jump Destination" window.



We have now finished programming the "Main" portion of the example SFC program. The next detail is to add the code for each of the steps and transitions. Double click on step #1 (initial step) and the "ISaGRAF SFC Program" window will open. Type the displayed text into the area shown below. This will associate the typed in code with the step #1. **REMEMBER** to type a semi-colon (":") at the end of each line of code.



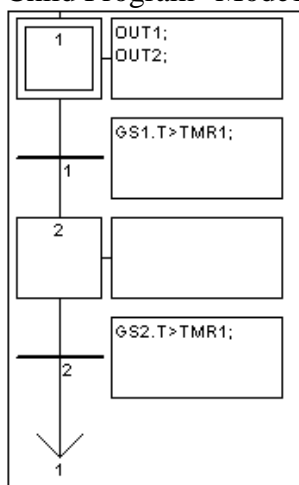
Using the same method as described above, double click on each transition and step and add the code for each item as shown below.



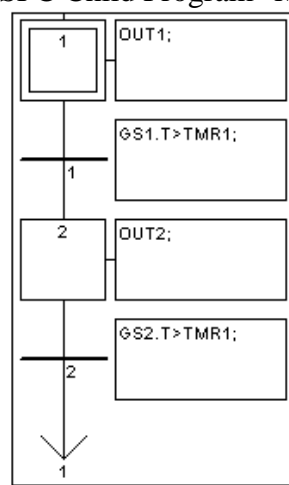
**CONGRATULATIONS!** You have now successfully programmed the "Main" section of the example SFC program (and the most time consuming).

The last portion of creating the example SFC program requires the creation and editing of the two "CHILD" programs. You program the "CHILD" programs using the exact same method as required for creating the "MAIN" program. When you are finished creating and editing the "CHILD" programs your two windows should look like the examples below.

SFC Child Program "Mode1"



SFC Child Program "Mode2"

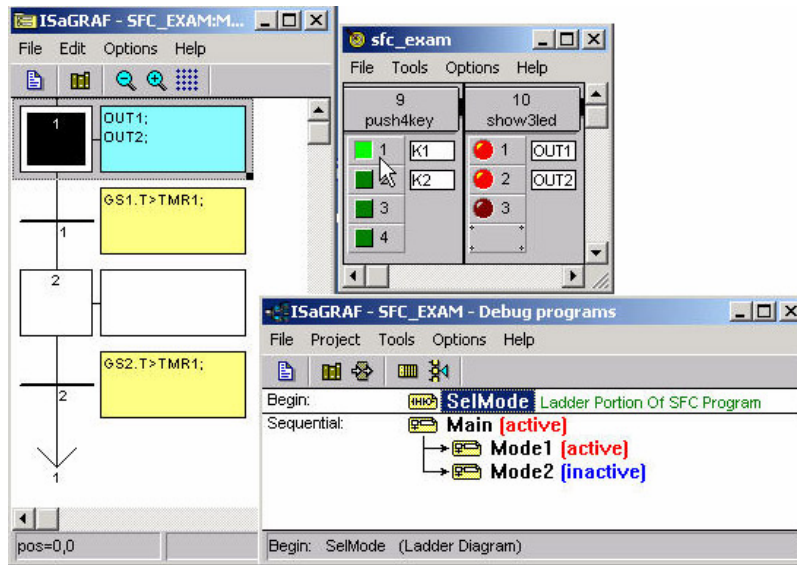


#### Final Details

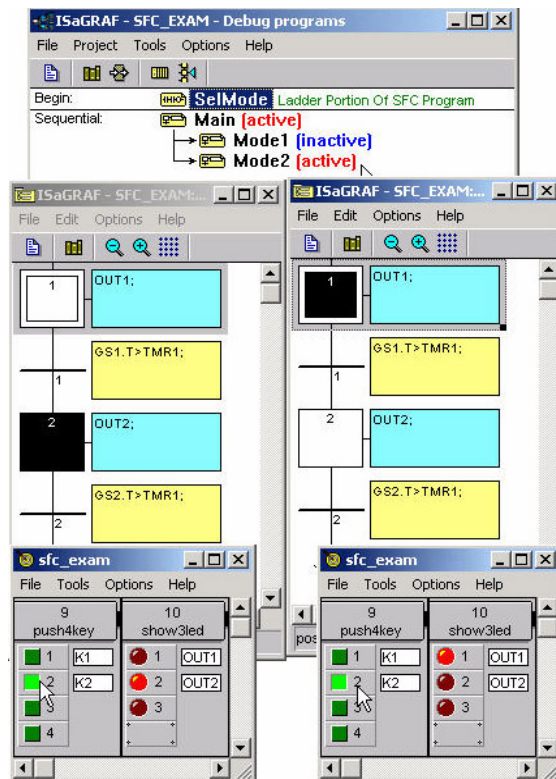
Remember that you must follow the same procedure for "Connecting I/O's" and "Compiling The Project" as detailed in Section 2.1.2 and Section 2.1.3.

### 2.5.3: Simulating The SFC Program

After you have successfully compiled the SFC program, you can now run the example SFC program in "Simulate" mode to observe how the two "CHILD" programs work within the "MAIN" SFC program. When "K1" is on, "Mode1" is true and both "OUT1" and "OUT2" turn on and off together, and "Mode2" is false.



When "K2" is on "Mode2" is true "OUT1" will turn on while "OUT2" is off and then they will alternate where "OUT2" will turn on and "OUT1" will be off, and "Mode1" is false.



## 2.6: Using Variable Array

If your ISaGRAF Workbench is version of 3.4 or 3.5, you can declare variable array in the ISaGRAF dictionary, And then program them in each language (ST, LD, FBD, SFC, IL & FC).

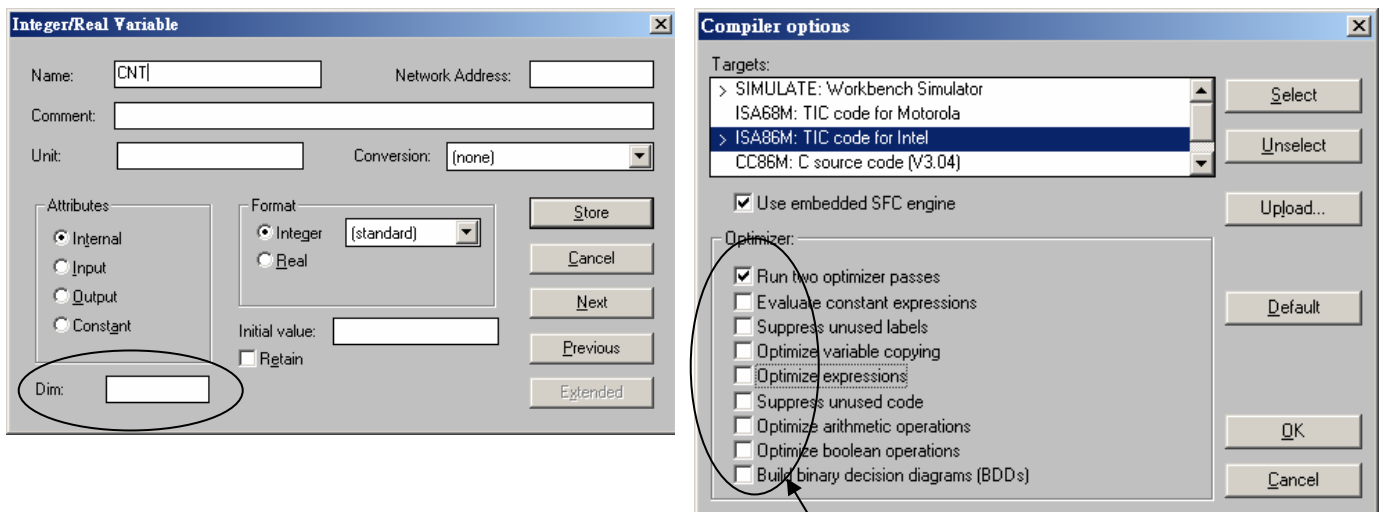
Please close all ISaGRAF windows first, and then add two extra lines in your ISaGRAF workbench root “EXE” directory, normally in the c:\isawin\exe.

In the “C:\ISAWIN\EXE\ISA.INT”, adds two extra raws on the top of this file.

**[DEBUG]**

**arrays=1**

And then re-open the ISaGRAF workbench, you will find there is one more “Dim” column in the ISaGRAF dictionary. The number entered can be 1 to 512. However **it is very important, please always declare the proper number you want. The larger “Dim” number, the larger memory is consumed.**



If using “Variable Array” in the program, please DO NOT check the 2nd , 7th , 8th and 9 th Optimizer options, or the value of the Variable array will be incorrect. Recommend to check only the 1st – “Run two optimizer passes” option.

The index of the variable array is always starting from 0. For example, if you declare an integer “CNT” with “Dim” = 10 , the variable array will be CNT[0..9] , that is the item can be used is CNT[0], CNT[1], ..., CNT[9].

## How to program variable array ?

For example, the below ST code can assign an initial value of 100 to 109 to CNT[0] to CNT[9]

(\* INIT is declared as an internal Boolean with initial value of TRUE \*)  
(\* CNT is declared as an integer array with “Dim” = 10 \*)  
(\* ii is declared as an internal integer \*)

### IF INIT THEN

INIT := FALSE ; (\* only do it once at 1<sup>st</sup> PLC scan cycle \*)

For ii := 0 to 9 do

CNT[ii] := 100 + ii ;

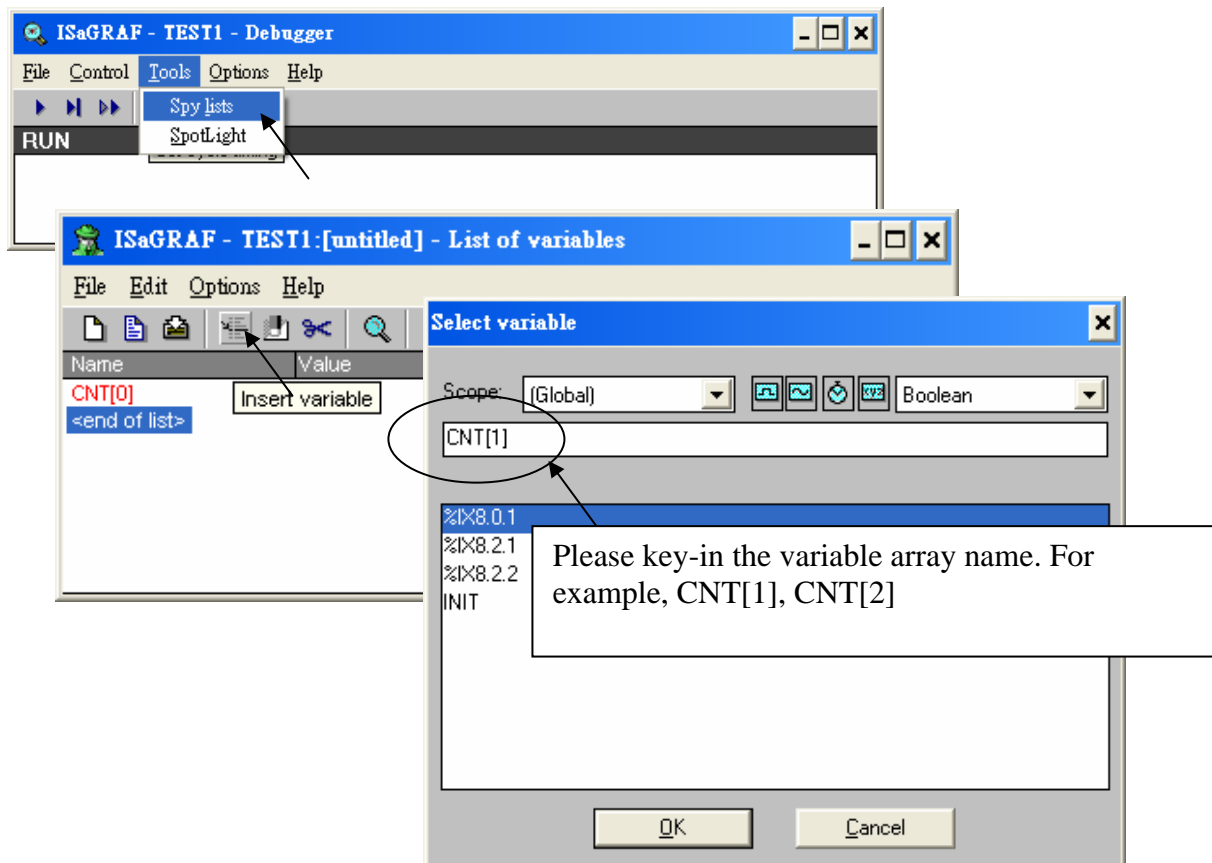
End\_For ;

END\_IF ;

**Note: Please do not exceed the “Dim” number of variable array. For example please do not program at CNT[10] or CNT[11] in the above example since the CNT ‘s dimension is only 10 , CNT[0], CNT[1], ..., CNT[9]. There is no CNT[10], CNT[11], ...**

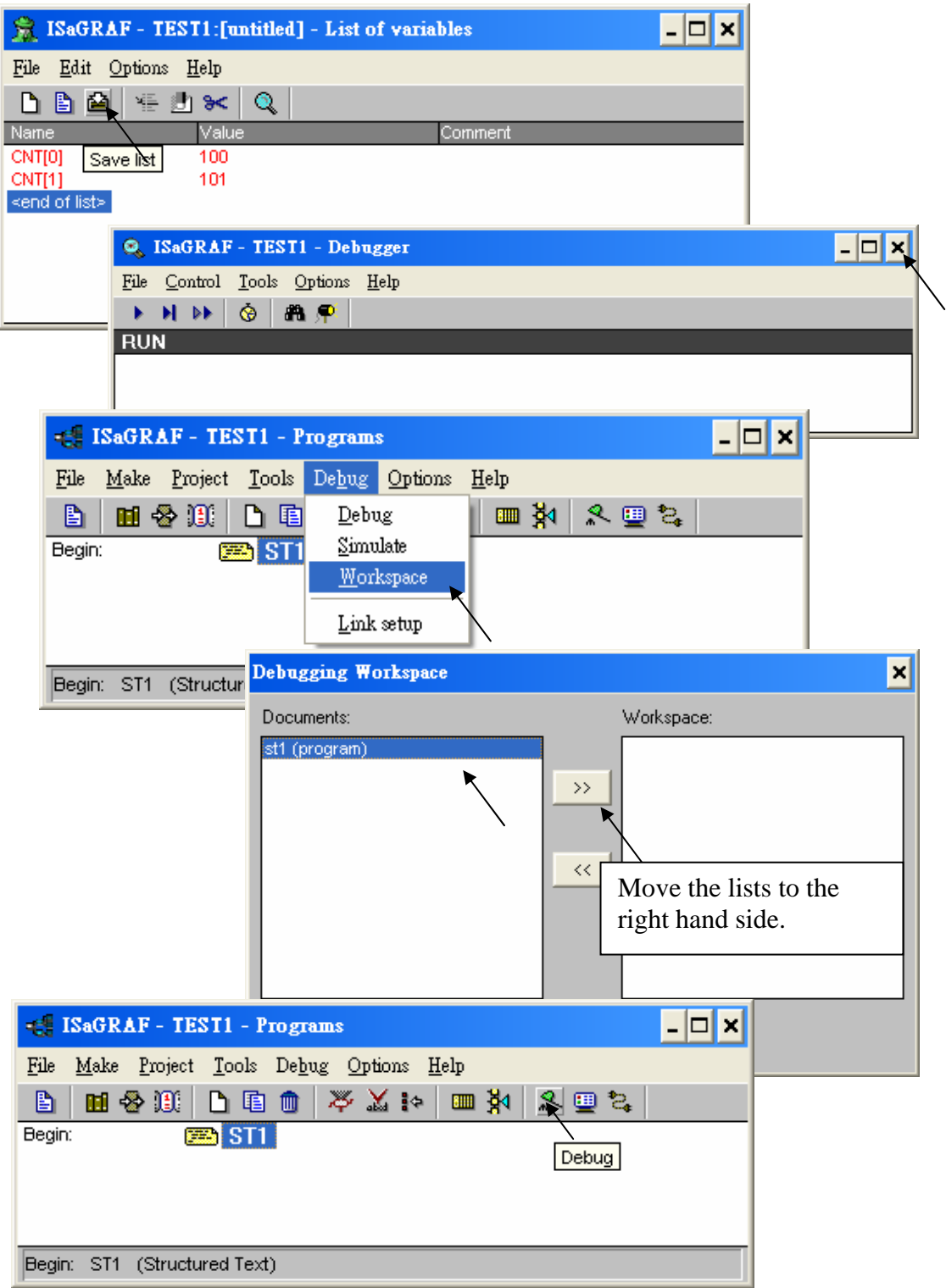
## How to debug variable array ?

After you compile your ISaGRAF project, you may download the project to the controller or simulate it. Please open the “Tools” - “Spy lists” on the “Debugger” windows. (please refer to section 9.12 for more information about “Spy lists”). Insert the name you want to debug.





Please remember to save the “spy list” to a name, for example – “list1” and then put it into the workspace. You will find the “list1” will automatically pop-up when you open the debugger.





### 2.6.1 Assign Network address No. To Variable Array

To assign Modbus Network address number to the variable array. Please assign the network address number to the first element, For example, No. = 1 assigned to CNT[0]. And the using “S\_MB\_ADR( )” function as below.

The screenshot shows the 'Integer/Real Variable' dialog box. The 'Name' field is 'CNT', 'Network Address' is '1', and 'Dim' is '10'. A callout box points to the 'Network Address' field with the text: 'Please assign the first number for the first element. (please refer to section 4.1 for more information) The number enter here is always in Hex. format.' The 'Attributes' section has 'Internal' selected. The 'Format' section has 'Integer' selected with '(standard)' in the dropdown. The 'Initial value' is '0' and 'Retain' is unchecked. Buttons for 'Cancel', 'Next', 'Previous', and 'Extended' are on the right.

And then using “S\_MB\_ADR” to assign the other network address number for each element.  
For example,

1. **Assign continuous Network No = 1 ,2 ,3, ...,10 to CNT[0], CNT[1], CNT[2], ..., CNT[9]**

(\* INIT is declared as internal Boolean with initial value at TRUE \*)

(\* TMP is declared as internal Boolean \*)

**IF INIT THEN**

**INIT := FALSE ;** (\* only do it at 1<sup>st</sup> PLC scan \*)

**TMP := S\_MB\_ADR(1, 10, 0) ;** (\* assign 10 elements starting at No.=1, continuous No. \*)

**END\_IF ;**

2. **Assign Jumping Network No = 1 , 3 , 5, ...,19 to CNT[0], CNT[1], CNT[2], ..., CNT[9]**

(\* INIT is declared as internal Boolean with initial value at TRUE \*)

(\* TMP is declared as internal Boolean \*)

**IF INIT THEN**

**INIT := FALSE ;** (\* only do it at 1<sup>st</sup> PLC scan \*)

**TMP := S\_MB\_ADR(1, 10, 1) ;** (\* assign 10 elements starting at No.=1, jumping No. \*)

**END\_IF ;**

## 2.6.2 Setting Variable Array As Retained Variable

To set “variable array” as retained data, please assign the network address number to the first element, For example, No. = 1 assigned to CNT[0]. And then using “Retain\_A( )” function as below. Please refer to chapter 10.1 for more information about the “New retained function”.

**Integer/Real Variable**

Name:  Network Address:

Comment:

Unit:

Attributes:

- ☒ Internal
- ☐ Input
- ☐ Output
- ☐ Constant

Format:

- ☒ Integer
- ☐ Real

Initial value:

☐ Retain

Buttons: Cancel, Next, Previous

Dim:

Callout 1: Please assign the first number for the first element. (please refer to section 4.1 for more information) The number enter here is always in Hex. format.

Callout 2: Do Not check the “Retain” option for variable array.

For example, setting integer variable array CNT[0..9] as retained data in the integer retained memory starting from 20 , 21 , ... to 29.

(\* INIT is declared as internal Boolean with initial value at TRUE \*)

(\* TMP is declared as internal Boolean \*)

**IF INIT THEN**

**INIT := FALSE ;** (\* only do it at 1<sup>st</sup> PLC scan \*)

**TMP := Retain\_A( 'N' , 1 , 10 , 20 ) ;**

(\*

1st parameter : 'B' : boolean , 'N' : Integer , 'F' : Real , 'T' : Timer

2nd parameter : Network address No. for the 1st element of the "Variable Array".

3rd parameter : 1 - 255 , number of element in the “variable array” to be assigned as retained data.

4th parameter : starting retained address for this “variable array”.

7188EG/XG+X607/608, I-8xx7+S256/512 : 'B' & 'T' is 1 to 256 , 'N' & 'F' is 1 to 1024 .

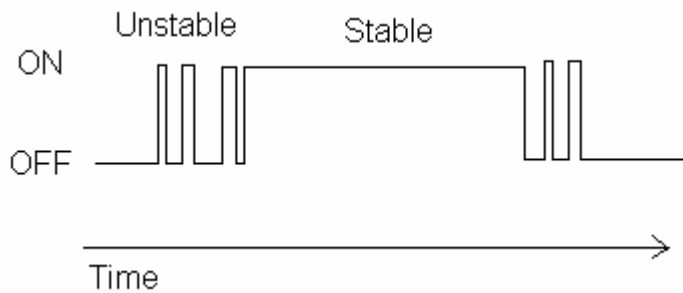
Wincon-8xx7/8xx6+S256/512 : 'B' & 'T' is 1 to 1024 , 'N' & 'F' is 1 to 4096

\*)

**END\_IF ;**

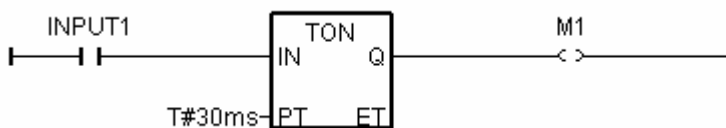
## 2.7: How to determine a D/I if it has bouncing problem ?

Some Digital Input devices have bouncing problem. They act similar as below.

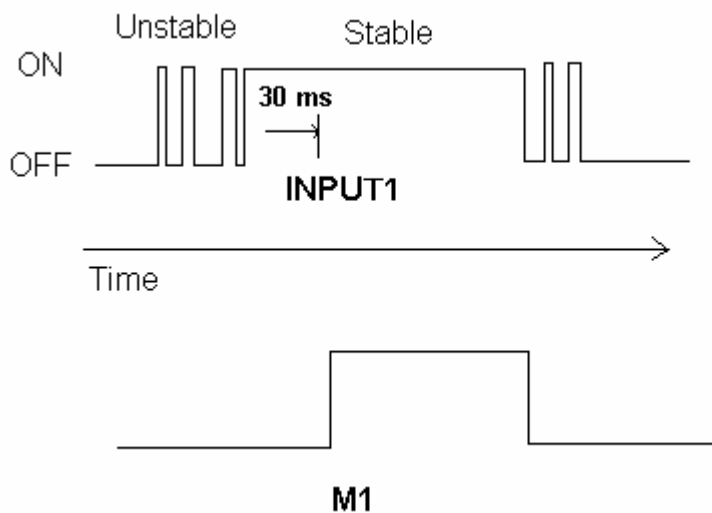


To determine the stable state of this D/I. User can write LD program as below.

Assume 30ms (0.03 second) to be defined as True.



Then M1 is the “ON” stable state of Input1.



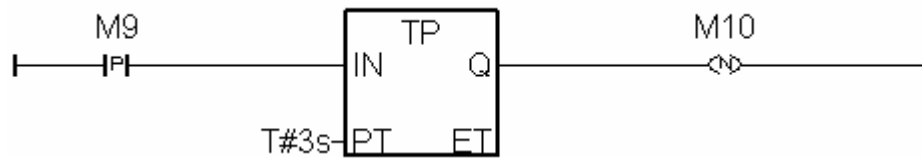
There is one another function block – “TOF” can be used to define the “OFF” stable state.

Note: M1 & Input1 are declared as Boolean variable.

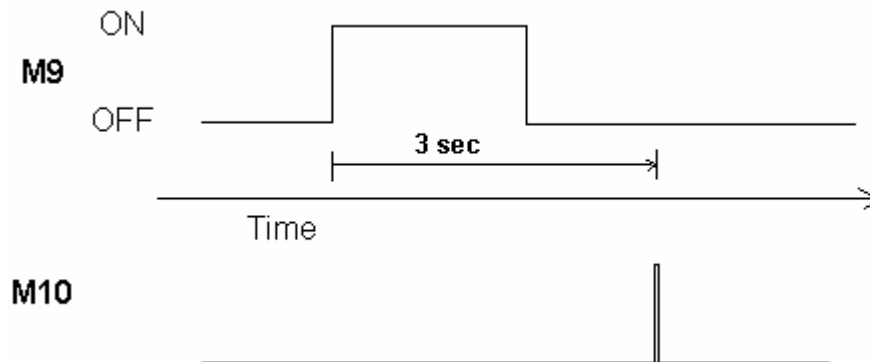
## 2.8: How to trigger something at some seconds later when one event happens ?

---

For ex. If users want to trigger one action at 3 seconds later when one event happens, please use the below LD logic. (M9 pulse is the event, M10 is the action to be triggered)



Then M10 is the triggered action at 3 seconds later.



Note: M9 & M10 are declared as Boolean variable.

## Chapter 3. Establishing I/O Connections

Before you can operate an ISaGRAF program with the I-8xx7, I-7188EG/XG & Wincon-8xx7 controller, you must make sure that the I/O Library has been installed. If you haven't done so already, install it as outlined in Section 1.2 "Installing The ICP DAS Utilities For ISaGRAF".

Please visit below Web site to get more information about I-8K and I-87K I/O boards.

[http://www.icpdas.com/products/PAC/i-8000/8000\\_IO\\_modules.htm](http://www.icpdas.com/products/PAC/i-8000/8000_IO_modules.htm)

I-8K parallel I/O boards can be plug in I-8xx7 's slot 0 through slot 7 and slot 1 through slot 7 of W-8xx7 controller. Parallel I/O boards CAN NOT be used in RU-87P4 , RU-87P8 and I-87K4 / 5 / 8 / 9 expansion unit.

|   |   |
|---|---|
| Digital Input / Output                  | I-8037 (16 out), I-8040 (32 in) , I-8041 (32 out) , I-8042 (16 in + 16 out)<br>I-8050 (16 in / out) , I-8051 (16 in) , I-8052 (8 in) , I-8053 (16 in)<br>I-8054 (8 in + 8 out) , I-8055 (8 in + 8 out) , I-8056 (16 out) , I-8057 (16 out)<br>I-8058 (8 in) , I-8060 (6 Relay out) , I-8063 (4 in + 4 Relay out)<br>I-8064 (8 Relay out) , I-8065 (8 SSR-AC out) , I-8066 (8 SSR-DC out)<br>I-8068 (8 Relay out) , I-8069 (8 PhotoMos relay out)<br>I-8077 (8 simulation in + 8 simulation out) |
| Analog Input (V , mA)                   | I-8017H (8 Ch.) , I-8017HS (16 Ch.)<br>I-8xx7: max. I-8017H Sample rate is about <b>125Hz</b> ,<br>while I-8017HS is about 62 Hz<br>W-8xx7: max. I-8017H Sample rate is about <b>200Hz</b> ,<br>I-8017HS is about 100 Hz<br>It depends on the PLC scan time. The bigger scan time, the samller sample rate.   |
| Analog Output<br>(V , mA)               | I-8024 (4 Ch.)  |
| Counter / Frequency<br>Input            | I-8080 (4/8 Ch counter or Frequency)<br>(If using I-8xx7, please update its driver as I-8080 version)<br><a href="http://www.icpdas.com/products/PAC/i-8000/isagraf.htm">http://www.icpdas.com/products/PAC/i-8000/isagraf.htm</a>  |
| Motion control<br>(I-8091 & I-8090)     | I-8091 (2 Axes. Pulse output) , I-8090 (3 Axes Encorder)<br>(If using I-8xx7 driver, please update its driver as motion version)<br><a href="http://www.icpdas.com/products/PAC/i-8000/isagraf.htm">http://www.icpdas.com/products/PAC/i-8000/isagraf.htm</a><br>If using W-8xx7,the general driver version already support 8091 & 8090.  |
| FRNET Master card                       | I-8172 (2-port FRNET)<br>Only W-8xx7 and iPAC-8447 / 8847 support I-8172. One W-8xx7 can control 7 pcs. of I-8172. One iPAC-8447 / 8847 can control 4 pcs. of i-8172. Please refer to Chapter 23 or visit below web site for FRNET I/O<br><a href="http://www.icpdas.com/products/Remote_IO/frnet/frnet_list.htm">http://www.icpdas.com/products/Remote_IO/frnet/frnet_list.htm</a> or<br><a href="http://www.icpdas.com/faq/isagraf.htm">http://www.icpdas.com/faq/isagraf.htm</a> FAQ-082     |
| Expansion serial<br>communication ports | I-8112 (2-Ch. RS-232) , I-8114 (4-Ch. RS-232)<br>I-8142 (2-Ch. RS-485 / 422) , I-8144 (4-Ch. RS-485 / 422)<br>I-8142I (2-Ch. RS-485 / 422)<br>(I-8xx7 can expan max. 16 ports at slot 0 through slot 3)<br>(W-8xx7 can expan max. 10 ports at slot 1 through slot 7)  |

I-87K serial I/O boards can be used in I-8xx7 's slot 0 through 7 and slot 1 through slot 7 of W-8xx7. They also can be used in I-87K4 / 5 / 8 / 9 expansion unit.

|                           |   |
|---------------------------|---|
| Digital Input / Output    | I-87040 (32 in) , I-87040W (32 in), I-87041 (32 out) , I-87041W (32 out)<br>I-87051 (16 in), I-87051W (16 in) , I-87052 (8 in) , I-87052W (8 in)<br>I-87053 (16 in) , I-87053W (16 in)<br>I-87054 (8 in + 8 out) , I-87054W (8 in + 8 out)<br>I-87055 (8 in + 8 out) , I-87055W (8 in + 8 out)<br>I-87057 (16 out) , I-87057W (16 out) , I-87058 (8 in) , I-87058W (8 in)<br>I-87063 (4 in + 4 Relay out) , I-87063W (4 in + 4 Relay out)<br>I-87064 (8 Relay out) , I-87064W (8 Relay out)<br>I-87065 (8 SSR-AC out) , I-87065W (8 SSR-AC out)<br>I-87066 (8 SSR-DC out) , I-87066W (8 SSR-DC out)<br>I-87068 (8 Relay out) , I-87068W (8 Relay out)<br>I-87069 (8 PhotoMos Relay out) , I-87069W (8 PhotoMos Relay out) |
| Anlog Input (V , mA)      | I-87017R (8-Ch.) , I-87017RC (8-Ch.) : Sample rate is 1.25Hz (10/8=1.25)<br>I-87019R (8-Ch. Universal input or T/C input) : Sample rate is 1.25Hz<br>I-87017W-A5 (8-Ch.)  |
| Analog Output (V , mA)    | I-87022 (2-Ch.) , I-87022W (2-Ch.) , I-87024 (4-Ch.) , I-87024W (4-Ch.)<br>I-87026 (2-Ch.) , I-87026W (2-Ch.)   |
| Temperature input         | I-87013 (4-Ch. RTD) , I-87013W (4-Ch. RTD)<br>I-87015 (7-Ch. RTD) , I-87018R (8-Ch. T/C)<br>I-87018Z(10-Ch. T/C or 0~20mA, 4~20mA or +/-2.5V)<br>I-87019R (8-Ch. Universal input or T/C input)  |
| Counter / Frequency Input | i-87082 (2-Ch. Counter or Frequency)<br>I-87082W (2-Ch. Counter or Frequency)   |

**Important:**

1. The I-87017R and I-87017RC is better than I-87017 and I-87017C in industrial application.
2. The I-87018Z is better than I-87018R and I-87018 in industrial application. (I-87018Z has 10-channels. The precision is better than I-87018R and I-87018 . And each channel can configure to be different Input type and range. For example, using Ch.1 to 4 to measure 4 to 20 mA , using Ch.5 to 8 as Thermo-Couple K-Type, using Ch.9 to measure +/- 2.5 V, and using Ch.10 as Thermo-Couple R-Type.)
3. The I-7018Z is better than I-7018R and I-7018. (The reason is the same as I-87018Z)

i-7018z: [http://www.icpdas.com/products/Remote\\_IO/i-7000/i-7018z-g.htm](http://www.icpdas.com/products/Remote_IO/i-7000/i-7018z-g.htm)

i-87018z: [http://www.icpdas.com/products/Remote\\_IO/i-87k/i-87018z.htm](http://www.icpdas.com/products/Remote_IO/i-87k/i-87018z.htm)

For more description about using I-87018Z and I-7018Z, please refer to Chapter 11.3.9 or [www.icpdas.com](http://www.icpdas.com) – FAQ – Software – ISaGRAF - 055.

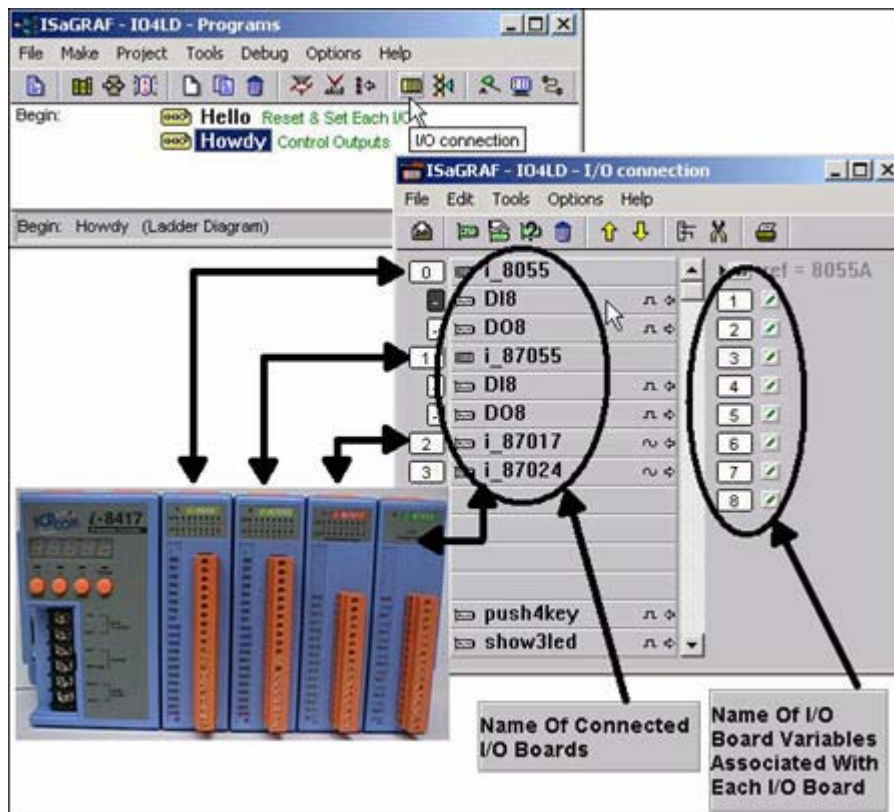
Please refer to “Getting Started Manual” of I-7188EG/XG for information about X-xxx board. or [http://www.icpdas.com/products/PAC/i-8000/getting\\_started\\_manual.htm](http://www.icpdas.com/products/PAC/i-8000/getting_started_manual.htm)  
[http://www.icpdas.com/products/PAC/i-o\\_expansion/x\\_list.htm](http://www.icpdas.com/products/PAC/i-o_expansion/x_list.htm)

### 3.1: Linking I/O Boards To An ISaGRAF Project

To begin connecting I/O boards to an ISaGRAF project you must first link the I/O boards to the ISaGRAF program. The numbers on the left of the "I/O Connections" window indicate the slot number. Slots 0 through 7 are used ONLY for **real** I-8000 series I/O boards (**Slot 1 through 7 for W-8xx7**). Slots 8 and above can be used for "virtual" I/O boards such as the "Push4Key" and "Show3Led" functions for the I-8xx7. For I-7188EG/XG & uPAC-7186EG, slot 0 is for Xxxx serial I/O boards, slot 1 & above are for others.

In this example I/O connection we are using the I-8417 controller system that has the following boards installed:

- Slot 0: I-8055 Board (8 digital inputs & 8 digital outputs)
- Slot 1: I-87055 Board (8 serial inputs & 8 serial outputs)
- Slot 2: I-87017 Board (8 channel analog input)
- Slot 3: I-87024 Board (4 channel analog output)
- Slot 8: "Push4Key"
- Slot 9: "Show3Led"

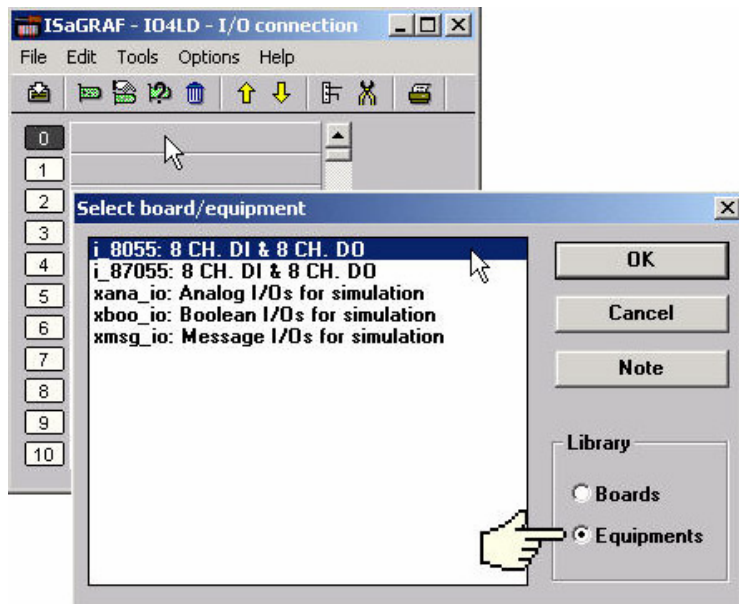


A powerful feature of the I-8xx7 controller system is that you can intersperse "real" I/O boards with "virtual" I/O boards.

### 3.1.1: Linking I/O Boards

With the "I/O Connection" window open double click on the slot that you want to connect an I/O board to. The "Select Board/Equipment" window will open, scroll to the name of the I/O board that you want to associate with the particular slot.

The ISaGRAF controller library defines two basic types of real I/O boards, "Boards" and "Equipments". The "Boards" selection is for I/O boards that are "single type", meaning that all of the channels on that board are of a single type and attribute. The "Equipments" selection is for I/O boards that are "multi-type", which means boards that have multiple types (such as the I-8055 digital I/O board that has 8 digital inputs and 8 digital outputs all on the same board). To begin the linking I/O board process, double click on the slot that you want to associate an I/O board to.



If you link an I/O board to an incorrect slot, first click on the slot number you wish to correct, then just click on the "Clear Slot" icon to delete the connection. The connection is now cleared, and now you can make a connection to the desired slot location.



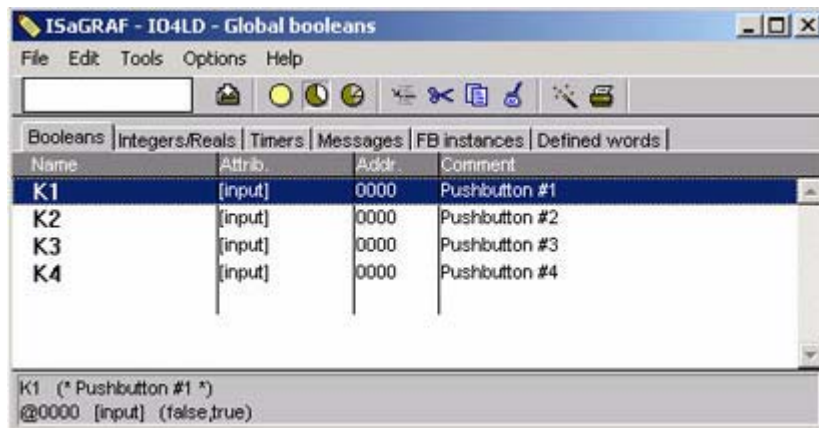


### 3.1.2: Linking Input & Output Board Variables

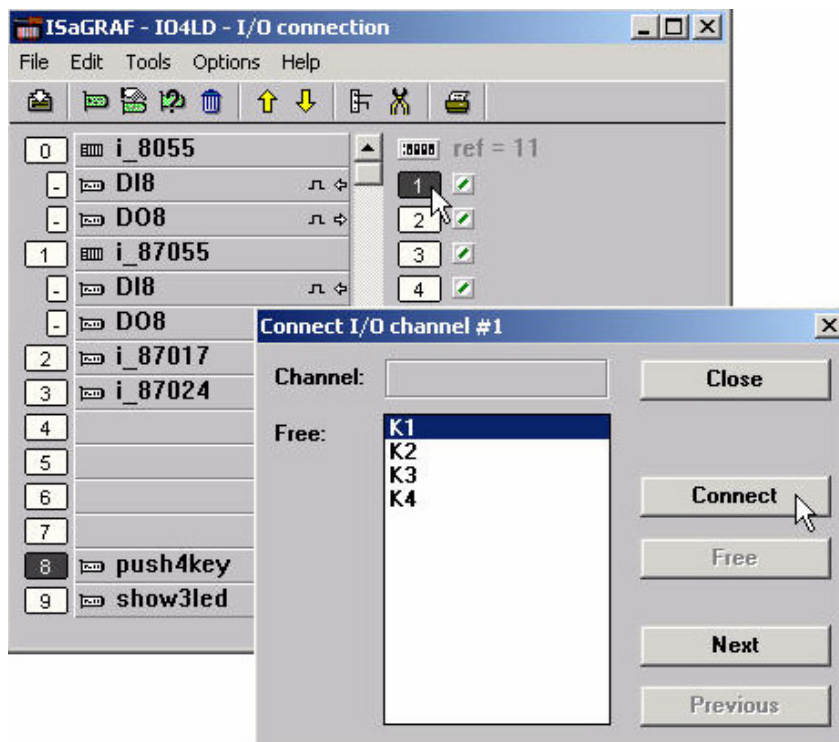
All of the input and output board "variables (or names)" must be linked (connected) in the "I/O Connection" window. Click on the slot you wish to link the attribute to, then double click on the channel (or I/O point name) number on the right hand portion of the "I/O Connection" window. Lastly, choose the variable name you wish to link to and then click on the "Connect" button.

#### IMPORTANT NOTE

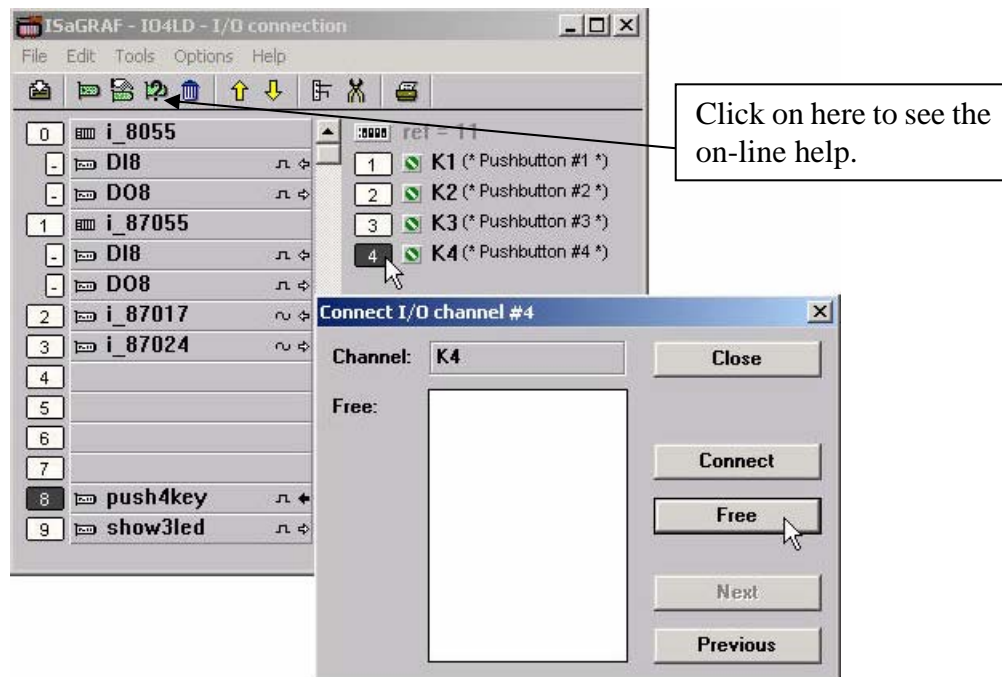
Remember that before you can assign any input or output, you must FIRST declare the variable in the "ISaGRAF Global Variables" window as shown below.



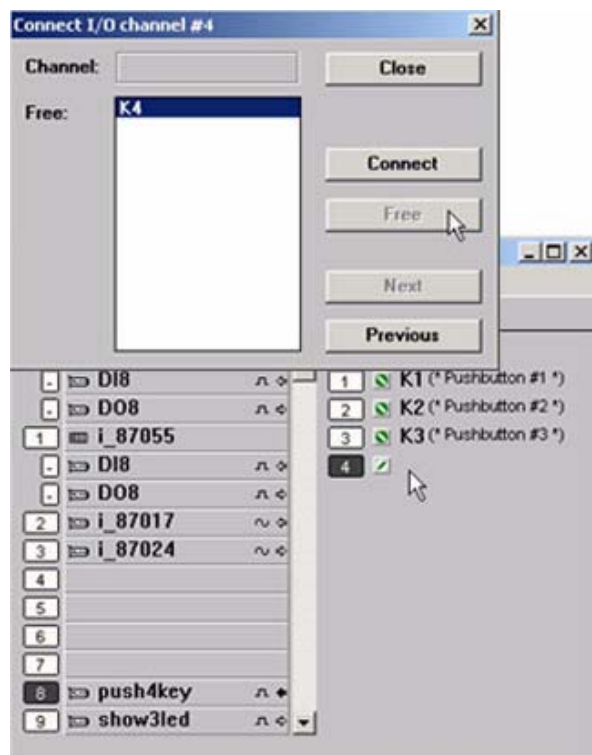
Click once on slot 8, then double click on "1" on the right hand side of the "ISaGRAF I/O Connection" window. With the "Connect I/O Channel #1" window now open, click on the "Connect" button to create the link between the variable "K1" and channel number 1 of the "Push4Key" input.



If you connect an input or an output variable to the wrong (or undesired) I/O location, double click on the I/O point you wish to remove. The "Connect I/O Channel #x" will open then click on the "Free" button to remove that variable from the I/O point.



When you click on the "Free" button you will see that the variable is removed from the I/O point in the "ISaGRAF I/O Connection" window and the variable is placed in the "Free" portion of the "Connect I/O Channel #x" window.



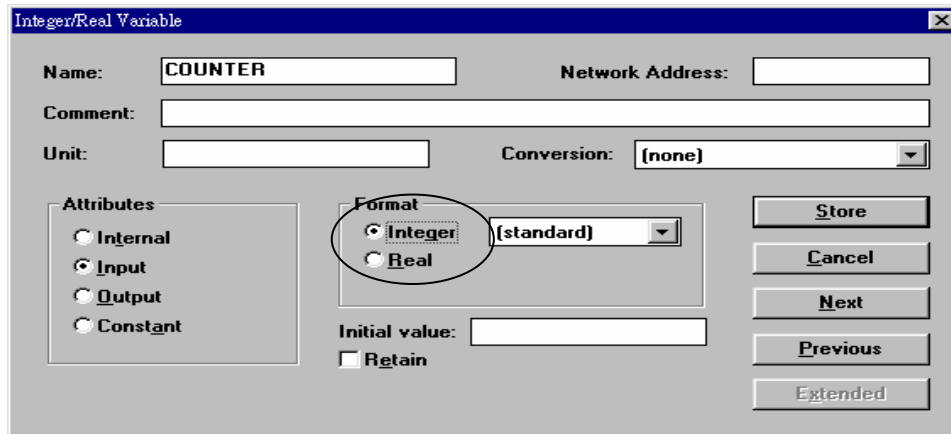
## 3.2: Linking Analog Type I/O Boards

Note:

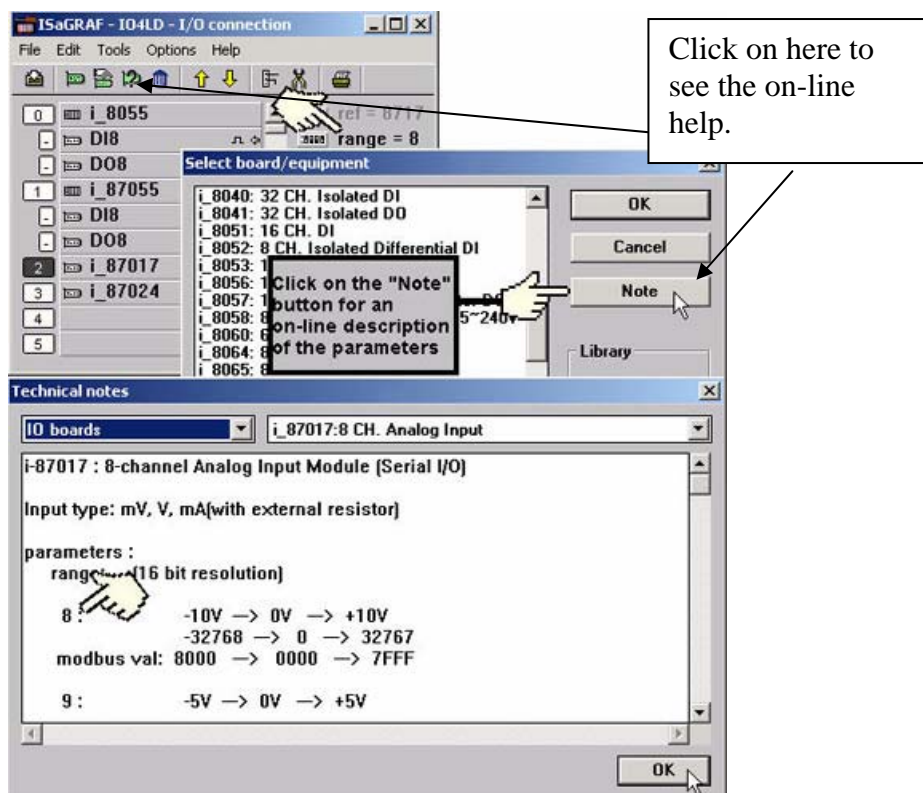
1. The I-87017R and I-87017RC is better than I-87017 and I-87017C in industrial application.
2. The I-87018Z is better than I-87018R and I-87018 in industrial application.

### 3.2.1: Setting “range” parameter in analog IO board

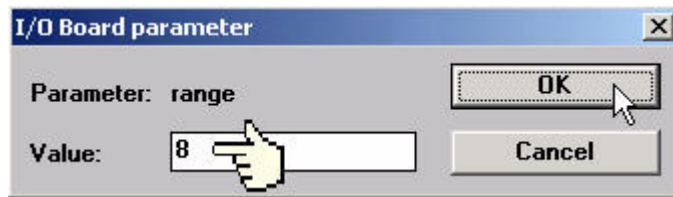
The method to connect analog type I/O boards to the controller system is very similar to that of connecting digital I/O boards. First, variables which are connected to analog type I/O boards should be declared as “Integer” format.



The ONE main difference is that you MUST define one parameter that defines the “range” for the analog board so it will operate as expected.



To modify the analog board "Range" parameter, click on the word "Range" in the "ISaGRAF I/O Connection" window and the "I/O Board Parameter" window will open. Enter in the correct "Range" parameter for your particular analog board application.



The below table provides information on several of the possible options for the "Range" parameter. Note that the default value is set to "8", which means you can interface to a –10v to +10v signal with a range value of –32768 to 32767. Changing the value of "Range" parameter to "9" means you can interface to a –5v to +5v signal with a range value of –32768 to 32767.

Note that if you set the "Range" parameter to "A" you will be interface to a –1v to +1v signal with a range value of –32768 to 32767. This range value can be very helpful in analog applications that require a great deal of resolution over a very small range (typically temperature) control.

| range: (16 bit resolution) |                    |  |
|----------------------------|--------------------|--|
| 8 :                        | -10V → 0V → +10V   |  |
|                            | -32768 → 0 → 32767 |  |
| modbus val:                | 8000 → 0000 → 7FFF |  |
| 9 :                        | -5V → 0V → +5V     |  |
|                            | -32768 → 0 → 32767 |  |
| modbus val:                | 8000 → 0000 → 7FFF |  |
| A :                        | -1V → 0V → +1V     |  |
|                            | -32768 → 0 → 32767 |  |

Please refer to **Appendix D - "Table of The Analog IO Value"** for more information for several different types of analog boards and their respective ranges.

### Note: Analog conversion functions:

The below functions are useful if user want to convert analog I/O value to application engineering value. For example, to convert 4 to 20 mA to become 0 to 1000 Psi, user can use function - "A4\_20\_To" to do it. Please refer Appendix A.4 for more description.

1. **A4\_20\_To** : to convert 4 to 20 mA analog value to become engineering value (32-bit Real)
2. **V0\_10\_To** : to convert 0 to 10 V analog value to become engineering value (32-bit Real)
3. **To\_A4\_20** : to convert engineering to become 4 to 20 mA analog output Value (Integer).
4. **To\_V0\_10** : to convert engineering to become 0 to 10 V analog output Value (Integer).
5. **BIN2ENG** : to convert Analog value (Integer, value should be in –32768 to +32767) to become engineering value (Integer, value should be in –32767 to +32768)

### 3.2.2: Setting special “range” parameter of temperature input board to get clear “Degree Celsius” or “Degree Fahrenheit” input value

ICPDAS provides many temperature input modules as below.

With “broken-line detection” or called “wire opening detection”

Thermocouple type: I-87018Z, I-87018R, 87019R, 7018Z, 7018R, 7018BL, 7019, 7019R

RTD type: I-87013, 87015, 7013, 7015, 7033

Thermister type: I-87005, 7005

Without “broken-line detection”

Thermocouple type: I-87018, 7018, 7018P

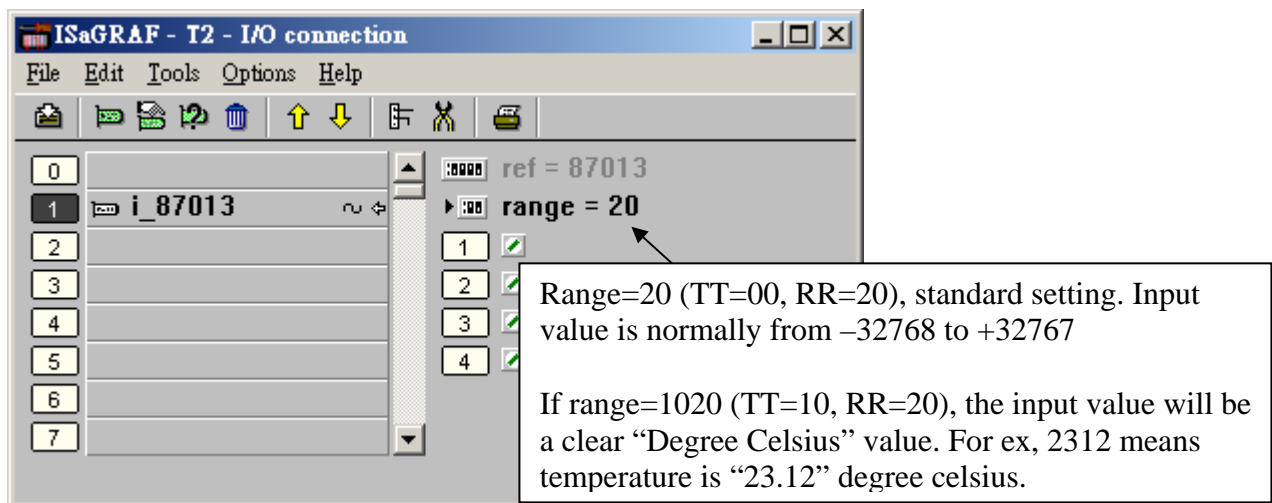
The “range” parameter of temperature IO board can be “standard setting” or “special setting”.

For example, I-87013: 4 channel RTD input module. Its range can be

20 : Platinum 100,  $\alpha=0.00385$ , -100 ~ +100 degree Celsius

...

2F : Platinum 100,  $\alpha=0.003916$ , -200 ~ +200 degree Celsius



If setting range as 20 (or 21 to 2F), then it is “standard setting”. The temperature input value is 2’s complement value from -32768 to +32767 depends on the “range” value. For example, setting range as 20, value of -32767 means temperature is about -100 Degree, +32766 is about +100 Degree. Value of 16383 means +50 Degree (**Note:** Normally value of -32768 or +32767 means wire “broken-line”)

If user want to get a clear temperature input value, for example, value of 2312 means “23.12” Degree Celsius. Then please set “range” to a special value defined as below.

#### Important:

Special “range” is supported since driver version of I-8xx7:3.11 , W-8xx7:3.24. It is better to update your controller hardware driver to become I-8xx7:3.19 , W-8xx7:3.36 or later version.

<http://www.icpdas.com/products/PAC/i-8000/isagraf.htm>

Format: TTRR (Hex. Value)

TT=10 (Convert to "Degree Celsius")

TT=20 (Convert to "Degree Fahrenheit")

TT=00 (Default value, -32768 to +32767, this is "standard setting")

RR: original "range" setting

For example, setting I-87013's "range" as

**A. 1020 :** (TT=10, RR=20) the input value will be "Degree Celsius", unit is 0.01 degree, range="20 : Platinum 100, a=0.00385, degree Celsius". That results input value of "2356" = 23.56 Degree Celsius, "-489" = -4.89 Degree Celsius, "999990" = sensor broken line.

**B. 202A :** (TT=20, RR=2A) the input value will be "Degree Fahrenheit", unit is 0.01 degree, range="2A: Platinum 1000, a=0.00385, degree Celsius". That results input value of "4512" = 45.12 Degree Fahrenheit, "500" = 5.00 Degree Fahrenheit, "999990" = sensor broken line.

**C. 21 :** (TT=00, RR=21) the input value will be Default value (standard "range" setting), -32768 to +32767, range = "21 : Platinum 100, a=0.00385, degree Celsius"

### 3.2.3: Using I-87017RC

I-87017RC is a 8-Channel current input board. Un-like the I-8017H , I-8017HS and I-87017R, it doesn't need to plug an external resistor of 125 ohms to measure current input.

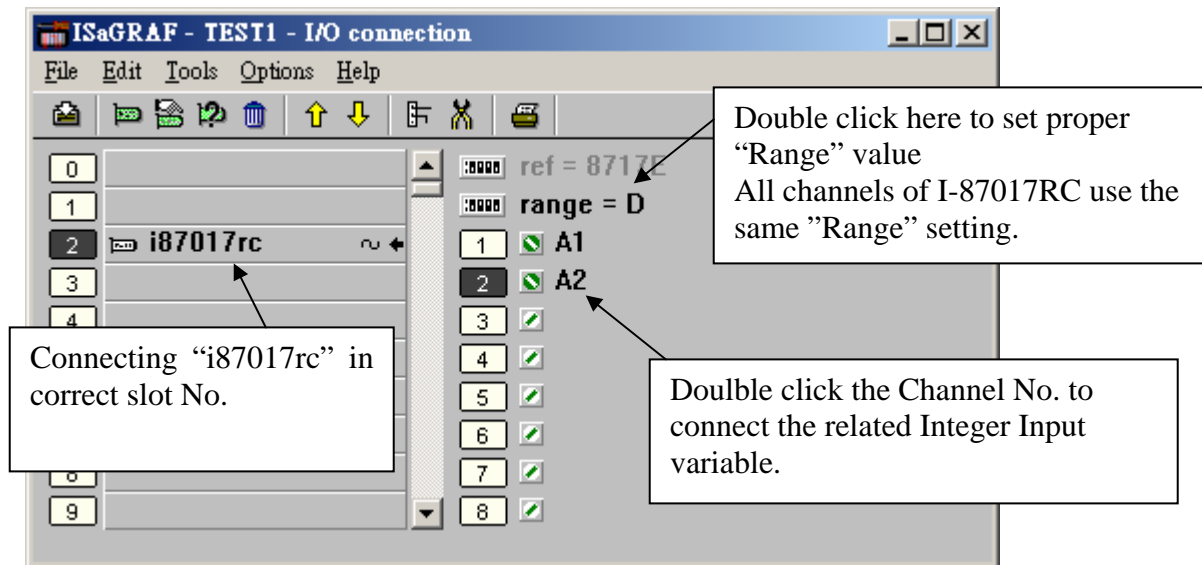
[http://www.icpdas.com/products/Remote\\_IO/i-87k/i-87017rc.htm](http://www.icpdas.com/products/Remote_IO/i-87k/i-87017rc.htm)

I-87017RC 's sample rate is about  $10/8 = 1.25$  Hz. If your application need faster sample rate, Please use I-8017H or I-8017HS.

The analog input value of I-87017RC is an integer between  $-32768$  to  $+32767$  as below table.

| Range Code (Hex) | Item                         | Max.     | Minimum  |
|------------------|------------------------------|----------|----------|
| 7                | Physical                     | +20.0 mA | +4.0 mA  |
|                  | Analog Input value (Decimal) | +32767   | 0        |
|                  | Analog Input value (Hex.)    | 7FFF     | 0        |
| D                | Physical                     | +20.0 mA | -20.0 mA |
|                  | Analog Input value (Decimal) | +32767   | -32768   |
|                  | Analog Input value (Hex.)    | 7FFF     | 8000     |
| 1A               | Physical                     | +20.0 mA | 0 mA     |
|                  | Analog Input value (Decimal) | +32767   | 0        |
|                  | Analog Input value (Hex.)    | 7FFF     | 0        |

To use I-87017RC in slot 0 to 7 of I-8xx7 or slot 1 to 7 of W-8xx7, please connect "i87017rc" in the related slot No. Then set the proper "Range" setting, then connecting related Integer Input variables in Channel 1 to 8.



**Note:**

If the current input sensor is 4 to 20 mA, user may better set I-87017RC 's range type to "[D] : +/- 20 mA", or "[1A] : 0 ~ 20 mA". (set as "[7] : 4 to 20 mA" is not good )

The reason is :

If setting I-87017RC 's range type as "[7] : 4 to 20 mA", analog Input value of 0 or close to 0 could mean the Sensor input is 4 mA , and also possible the Sensor is broken-line. So it is not easy to distinguish these two situation by software.

Howevr, if setting I-87017RC 's range type as "[D] : +/- 20 mA" or "[1A] : 0 ~ 20 mA", analog input value of 0 or close to 0 only means the Sensor is broken-line . If the Sensor input is 4 to 20mA , the analog value should be 6553 to 32767 , not close to 0.



### 3.2.4: Using I-8017H and I-8017HS

I-8017H and I-8017HS are parallel analog input boards. I-8017H has 8-channels, while I-8017HS has 16-channels. If using them in I-8xx7 controller, the I-8017H 's max. sample rate is about 125Hz , I-8017HS is about 62 Hz. If using in W-8xx7 controller, the I-8017H 's max. sample rate is about 200Hz , I-8017HS is about 100 Hz. However the sample rate depends on the PLC scan time, the bigger PLC scan time, the smaller sample rate will be. For ex, if the PLC scan time is 50 ms, that will make the max. I-8017 sample rate only about  $1000 / 50 = 20$  Hz.

[http://www.icpdas.com/products/PAC/i-8000/8000\\_IO\\_modules.htm](http://www.icpdas.com/products/PAC/i-8000/8000_IO_modules.htm)

Un-like the I-87017R and I-87017RC, the I-8017H and I-8017HS can set each channel to become different range type. The “CHx\_rang” setting can be :

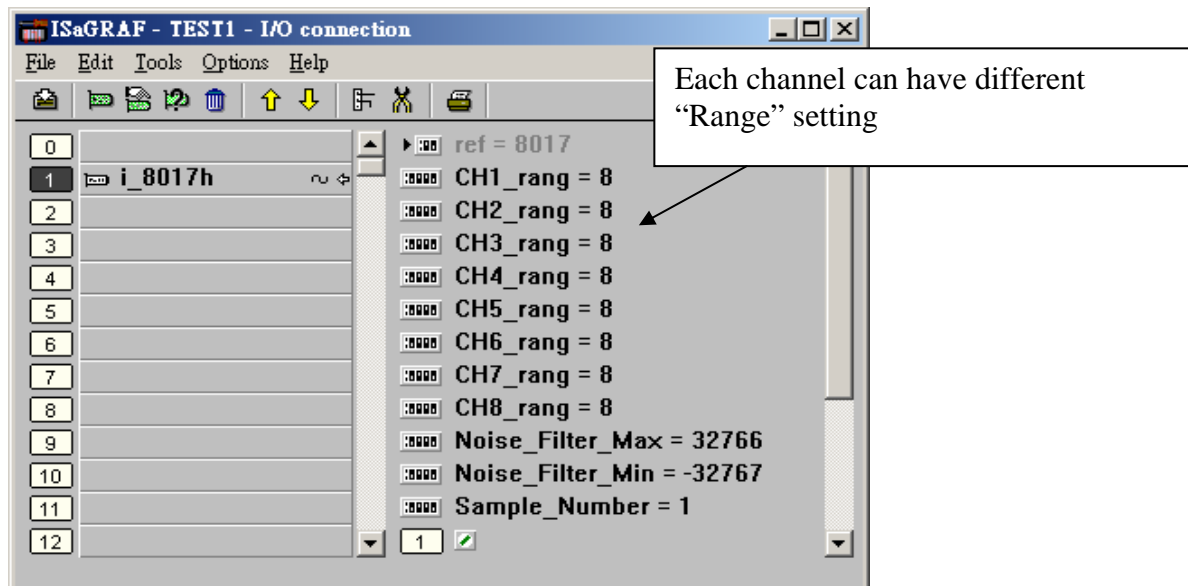
| Range setting                             | Minimum<br>Physical (analog val.) | Physical (analog val.) | Maximum<br>Physical (analog val.) |
|---|-----------------------------------|------------------------|-----------------------------------|
| 5   | -2.5 V (-32768)                   | 0 V (0)                | + 2.5 V (+32767)                  |
| 6 (need external resistor<br>of 125 ohms) | -20 mA (-32768)                   | 0 mA (0)               | + 20 mA (+32767)                  |
| 7   | -1.25 V (-32768)                  | 0 V (0)                | + 1.25 V (+32767)                 |
| 8   | -10 V (-32768)                    | 0 V (0)                | + 10 V (+32767)                   |
| 9   | -5 V (-32768)                     | 0 V (0)                | + 5 V (+32767)                    |

Parameters:

**Noise\_Filter\_Max** : The maximum allowed analog input value. -32468 to +32767. If the analog input value is larger than this value, it will be modified to become this value.

**Noise\_Filter\_Min** : The minumum allowed analog input value. -32768 to +32467. If the analog input value is smaller than this value, it will be modified to become this value.

**Sample\_Number** : The sampling number to be averaged as an Analog input value. Default is 1. Can be modified to become 1 to 500. Setting as 1 using the max. sampling speed. However setting as 500 means every 500 samples to be averaged as one analog input value.



### 3.2.5: Using I-8080

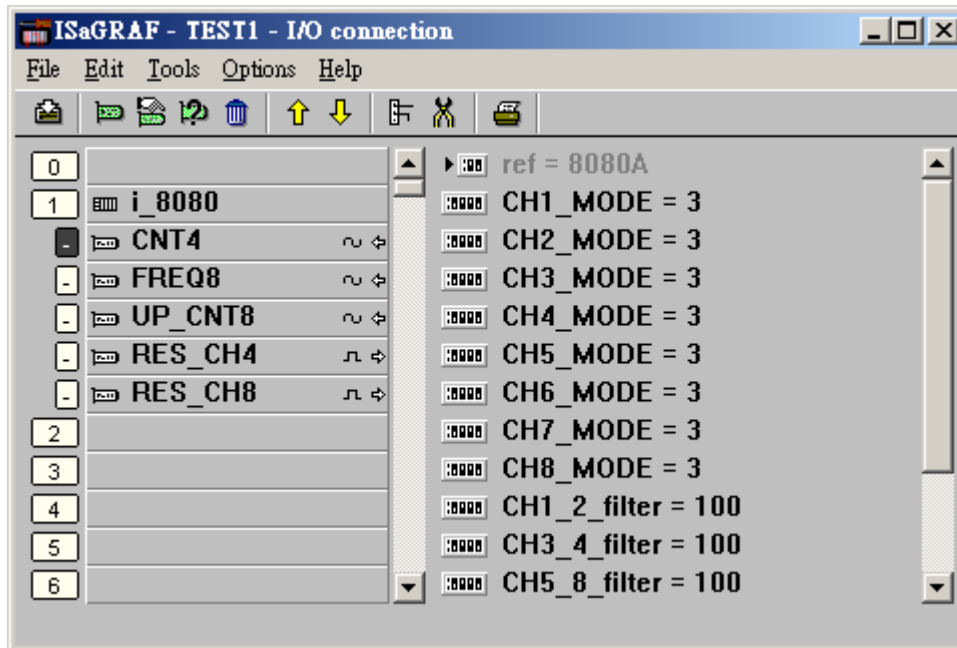
If using in I-8xx7, please update I-8xx7 driver to become i-8080 version (Ver. 3.19 or later).

If using in Wincon-8xx7, please update driver to become 3.36 or later.

<http://www.icpdas.com/products/PAC/i-8000/isagraf.htm>

I-8080 can measure 8-channel of “Up-Counter” or 4-channel of “Dir / Pulse Counter” or 4-channel of “Up / Down Counter” . It can also measure 8-channel of “Frequency input”

[http://www.icpdas.com/products/Remote\\_IO/i-8ke/i-8080.htm](http://www.icpdas.com/products/Remote_IO/i-8ke/i-8080.htm)



The default "CHx\_x\_filter" setting is 100. It is for filtering, the input signal with smaller time-width (that is, larger input frequency) will be filtered out.

CH1\_2\_filter: for Ch.1 and Ch.2 of “8-Ch. Up Counter” and “8-Ch. Frequency” or  
Ch.1 of “4-Ch. Dir/Pulse Counter” and “4-Ch. Up/Down Counter”

CH3\_4\_filter: for Ch.3 and Ch.4 of “8-Ch. Up Counter” or “8-Ch. Frequency” or  
Ch.2 of “4-Ch. Dir/Pulse Counter” and “4-Ch. Up/Down Counter”

CH5\_8\_filter: for Ch.5, 6, 7, 8 of “8-Ch. Up Counter” and “8-Ch. Frequency” or  
Ch.3 and Ch.4 of “4-Ch. Dir/Pulse Counter” and “4-Ch. Up/Down Counter”

Please set a proper filter value according to the physical input signal.

| Max. allowed input signal (Hz) | CHx_x_filter value |
|--------------------------------|--------------------|
| 0 ~ 1K                         | 200                |
| 0 ~ 2K                         | 100                |
| 2K ~ 5K                        | 40                 |
| 5K ~ 10K                       | 20                 |
| 10K ~ 20K                      | 10                 |
| 20K ~ 100K                     | 2                  |
| 100K ~ 450K                    | 1                  |

“CHx\_MODE” setting is to set the signal input type of each channel as below.

“CH1\_MODE” to “CH8\_MODE” is for Ch.1 to Ch.8 of “8-Ch Up Counter” and “8-Ch Frequency” .

If setting as “4-Ch. DIR / Pulse Counter” or “4-Ch. Up / Down Counter” mode,  
CH1\_MODE and CH2\_MODE must set as the same value. It is for Ch1.  
CH3\_MODE and CH4\_MODE must set as the same value. It is for Ch2.  
CH5\_MODE and CH6\_MODE must set as the same value. It is for Ch3.  
CH7\_MODE and CH8\_MODE must set as the same value. It is for Ch4.

For example,

1. if setting CH1\_MODE as 0 : “Dir / Pulse” (4-Ch), then CH2\_MODE should be also set as 0.
2. if setting CH1\_MODE as 3 : “Up Count” (8-Ch), then CH2\_MODE can be set as 83 , 2, 6, A, 82, 86 or 8A

Below value is for Counter input type.

- 0 : Dir / Pulse (4-Ch.)
- 1 : Up / Down (4-Ch.)
- 3 : Up Count (8-Ch)
- 80 : Dir / Pulse (4-Ch. Inverse input signal)
- 81 : Up / Down (4-Ch. Inverse input signal)
- 83 : Up Count (8-Ch. Inverse input signal)

Below value is for Frequency input type.

- 2 : Frequency (apply the “AutoTT” setting)
- 6 : Frequency (apply the “LowTT” setting)
- A : Frequency (apply the “HighTT” setting)
- 82 : Frequency (apply the “AutoTT” setting, Inverse input signal)
- 86 : Frequency (apply the “LowTT” setting, Inverse input signal)
- 8A : Frequency (apply the “HighTT” setting, Inverse input signal)

Note:

1. “DIR / Pulse” mode and “Up / Down Counter” mode are similar as Encoder Input.

The Counter value should be controlled in between -2,147,483,648 to 2,147,483,647. Or it will be overflow.

2. The input value of “Up Counter” mode is a 32-bit integer. It starts at 0, then increasing by the signal input, 1, 2, ... to max. value of +2,147,483,647, then if one more signal input, the value will suddenly drop to -2,147,483,648. Then increasing ... to -2, -1, 0, 1, 2, ... to +2,147,483,647.

The ISaGRAF integer value is a signed 32-bit integer , It can not get a positive value larger than +2,147,483,647.

If user apply SCADA software which can handle unsigned 32-bit integer, then the value displayed in the SCADA software can be 0, 1, ..., +2147483647 , +2147483648 , +2147483649 , ... , +4294967295 , then back to 0, 1, ...

### 3.2.6: Using I-87015 to measure up to 7-ch of RTD temperature inputs

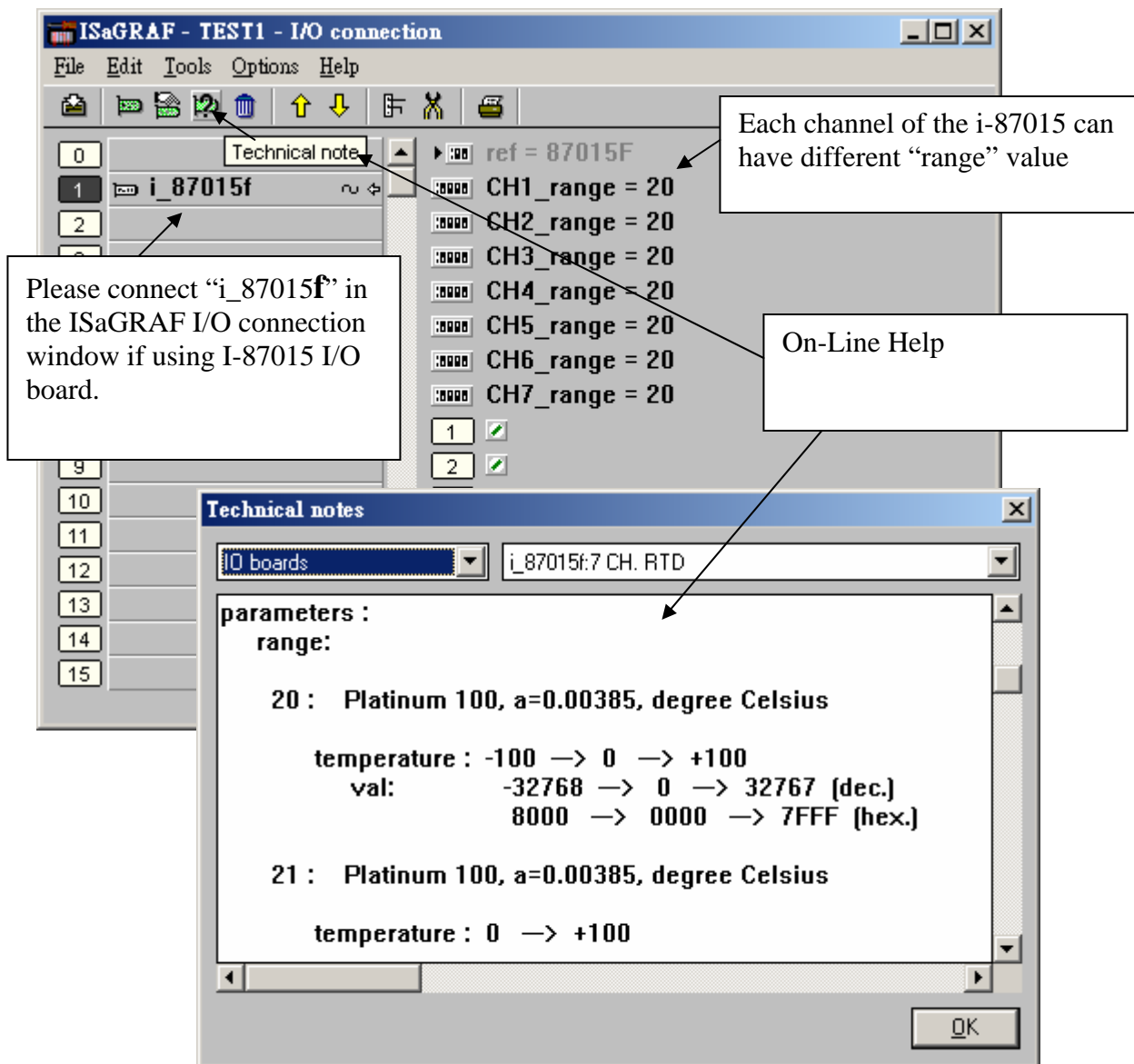
The i-87K I/O board (like I-87015) can plug in the slot 0 through slot 7 of the I-8xx7 controller, and in slot 1 through slot 7 of the Wincon-8xx7 controller. It can also be used as RS-485 remote I/O module. (That is plug in the RS-485 I-87K4 / 5 / 8 / 9 expansion unit. Please refer to Chapter 6).

[http://www.icpdas.com/products/Remote\\_IO/i-87k/i-87015.htm](http://www.icpdas.com/products/Remote_IO/i-87k/i-87015.htm)

I-87015 's Sample Rate is about  $10/8 = 1.25$  Hz.

The below figure is using the I-87015 in slot 1.

The I-87015 has 7-channel of RTD temperature input. Please connect “i\_87015F” in the ISaGRAF I/O connection window if using I-87015 I/O board. Each channel can have different “Range” setting. Please refer to the On-Line Help as below for more information. (If user want to convert to degree Celsius or Degree Fahrenheit, please refer to Chapter 3.2.2 to set proper “Range” value)



### 3.2.7: Using I-87019R

I-87019R is a 8-channel of Thermo-Couple temperature input board or can be set as normal analog input board, like “±10V” or “±20mA”. Each channel can have different “range” type setting.

[http://www.icpdas.com/products/PAC/i-8000/8000\\_IO\\_modules.htm](http://www.icpdas.com/products/PAC/i-8000/8000_IO_modules.htm)

[http://www.icpdas.com/products/Remote\\_IO/i-87k/i-87019r.htm](http://www.icpdas.com/products/Remote_IO/i-87k/i-87019r.htm)

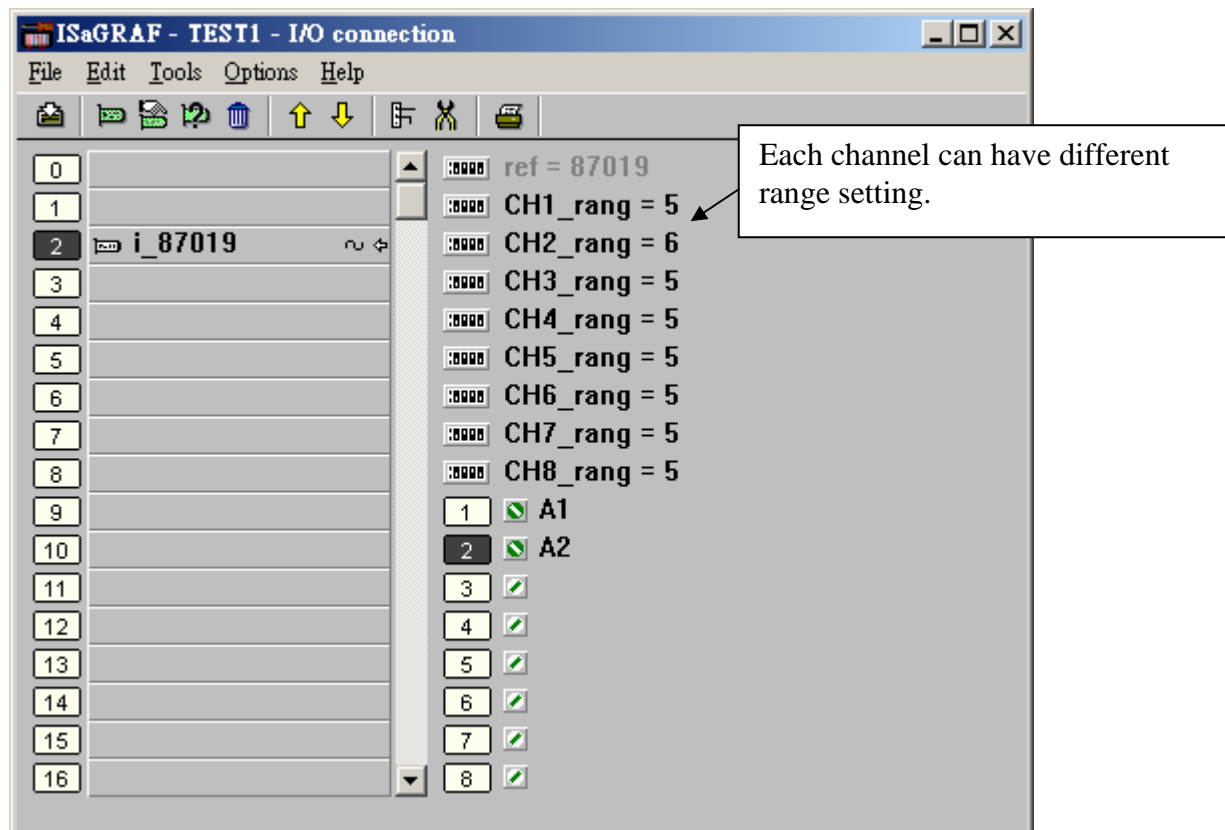
The analog input value of I-87019R is between –32768 to +32767 (refer to Appendix D).

When applying as temperature input , please refer to Chapter 3.2.2 to get direct temperature value.

If applying I-87019R to measure current input (“range” setting is 06 or 0D), it doesn’t need to plug external resistor of 125 ohms. (There is “Jumper” to be switched on the board).

I-87019R ‘s Sample Rate is about  $10/8 = 1.25$  Hz. (If your application need faster Sample Rate, please use I-8017H or I-8017HS)

To program I-87019R, please connect “i\_87019” in the related slot in the ISaGRAF I/O connection window. Then set proper “range” setting. Then connecting related Integer Input variable in Ch.1 to Ch. 8 .



### 3.2.8: Using I-8024

The I-8024 is a 4-channel analog output board which can output signal of “ $\pm 10V$ ” and “0 to 20mA”. Each channel can have different “range” setting.

[http://www.icpdas.com/products/PAC/i-8000/8000\\_IO\\_modules.htm](http://www.icpdas.com/products/PAC/i-8000/8000_IO_modules.htm)

The output speed of the I-8024 depends on the PLC Scan Time. For example, if PLC scan time is 10 ms, then the output value will be output in about 10 ms. However if PLC scan time is 100ms, then the output value will be output in about 100 ms.

The speed of the I-8024 is faster than I-87024. Besides, unlike the I-87024, the I-8024 can have different “range” setting in each channel. (All channels of I-87014 use the same “range” setting)

However I-87024 can use in slot 0 to 7, and also in the RS-485 remote I-87K4 / 5 / 8 / 9 expansion unit. (I-8024 can only plug in slot 0 to 7 of the I-8xx7 or slot 1 to slot 7 in Wincon-8xx7 controller)

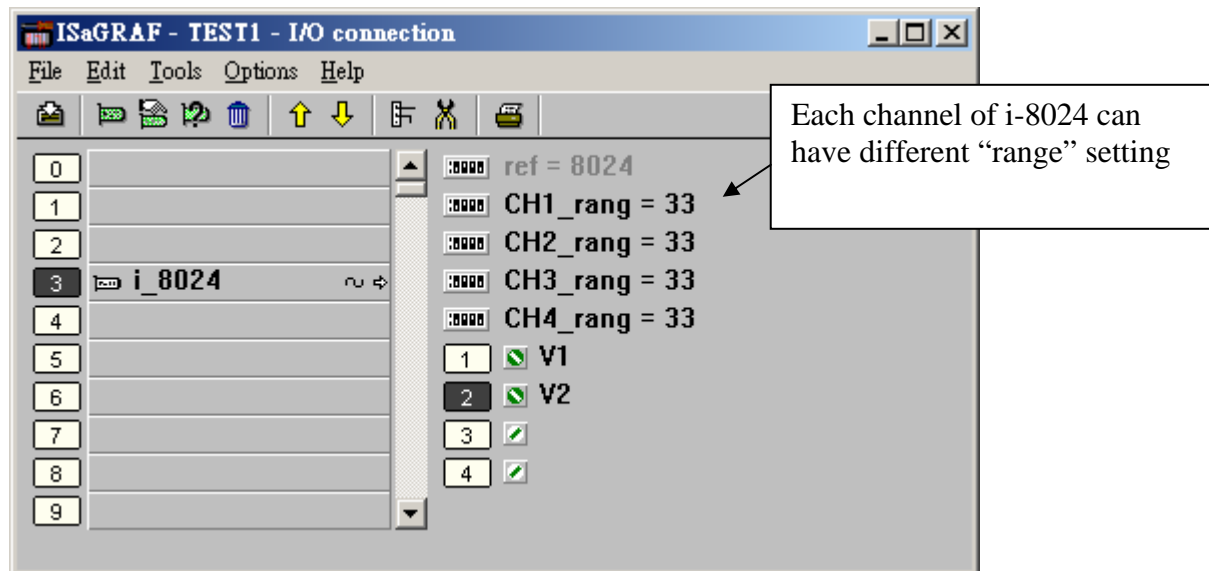
The analog output value of the I-8024 is :

Current : (0 to 32767) means (0 to 20 mA)

Voltage : (-32768 to 32767) means (-10 V to 10 V)

| Range Code | Item                | Max.     | Minimun |
|------------|---------------------|----------|---------|
| 30         | Physical signal     | +20.0 mA | +0.0 mA |
|            | Analog output value | +32767   | +0      |
| 33         | Physical signal     | +10.0 V  | -10.0 V |
|            | Analog output value | +32767   | -32768  |

To program I-8024, please connect “i\_8024” in the related slot No. on the I/O connection window. Then set proper “range” setting. Then connect related Integer Output variable to Ch.1 to Ch. 4.



### 3.2.9: Using I-87018Z to measure up to 10-Ch. of thermo-couple inputs

The I-87018Z is better than I-87018R and I-87018 in industrial application. I-87018Z has 10-channels. The precision is better than I-87018R and I-87018. And each channel can configure to be different Input type and range. For example, using Ch.1 to 4 to measure 4 to 20 mA, using Ch.5 to 8 as Thermo-Couple K-Type, using Ch.9 to measure +/- 2.5 V, and using Ch.10 as Thermo-Couple R-Type.

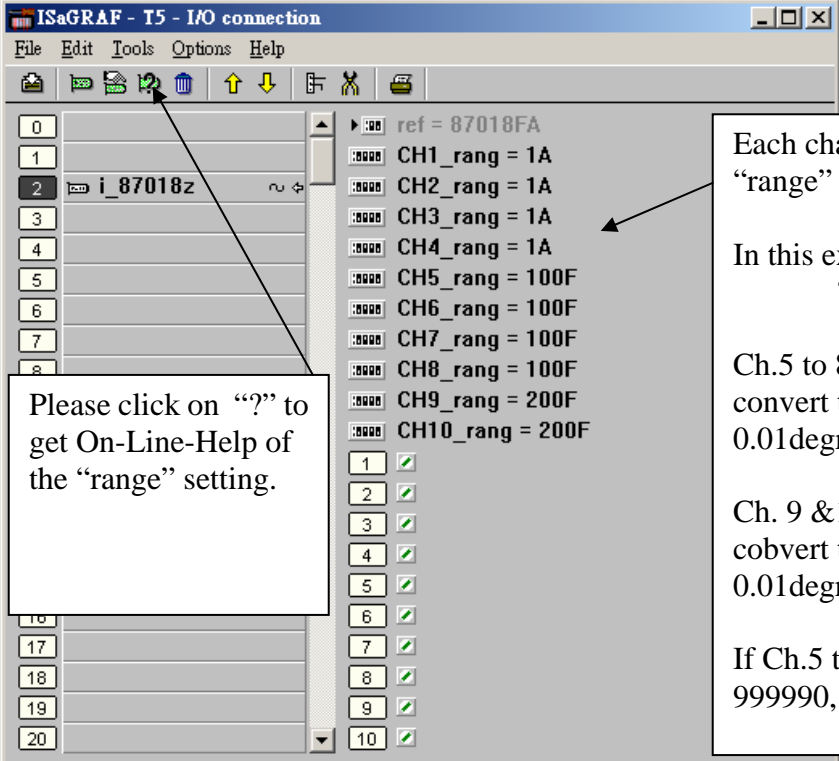
i-87018z: [http://www.icpdas.com/products/Remote\\_IO/i-87k/i-87018z.htm](http://www.icpdas.com/products/Remote_IO/i-87k/i-87018z.htm)

The analog input value of I-87018Z is between -32768 to +32767. (refer to Appendix D)  
When applying as temperature input, please refer to Chapter 3.2.2 to get direct temperature value.

Please connect an external resistor of 125 ohms if using I-87018Z to measure current input ("range" setting is 06:  $\pm 20$ mA or 07: 4 ~ 20 mA or 1A: 0 ~ 20 mA).

I-87018Z's Sample Rate is about  $10/10 = 1$  Hz. (If your application is to measure faster signal of " $\pm 10$ V" or " $\pm 20$ mA", please use I-8017H or I-8017HS)

To program I-87018Z, please connect "i\_87018z" in the related slot No. on the I/O connection window. Then set proper "range" setting. Then connect related Integer Input variable to Ch.1 to Ch. 10.



ISaGRAF - T5 - I/O connection

File Edit Tools Options Help

0 1 2 i\_87018z 3 4 5 6 7 8

ref = 87018FA

- CH1\_range = 1A
- CH2\_range = 1A
- CH3\_range = 1A
- CH4\_range = 1A
- CH5\_range = 100F
- CH6\_range = 100F
- CH7\_range = 100F
- CH8\_range = 100F
- CH9\_range = 200F
- CH10\_range = 200F

1 2 3 4 5 6 7 8 9 10

Please click on "?" to get On-Line-Help of the "range" setting.

Each channel of i-87018z can have different "range" setting.

In this example, Ch. 1 to 4 are using "1A: 0 ~ 20mA".

Ch.5 to 8 are using "100F: T/C K-Type, convert to Degree Celsius, unit is 0.01degree".

Ch. 9 &10 are using "200F T/C K-Type, convert to Degree Fahrenheit, unit is 0.01degree.

If Ch.5 to Ch.10 in this example return 999990, it means Sensor broken-line.

### 3.2.10: How to measure up to +/-150 VDC voltage input in ISaGRAF controllers with the i-87017W-A5 I/O card ?

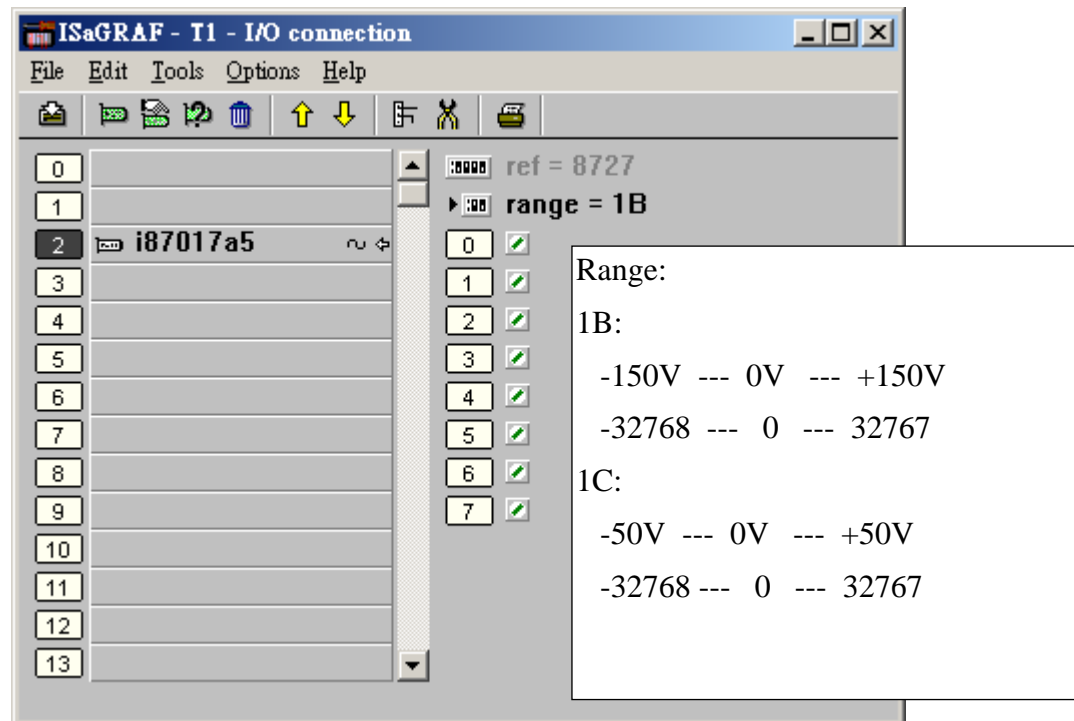
ISaGRAF controllers support the i-87017W-A5 I/O module to measure +/-150VDC or +/-50VDC since below driver version.

|                               |      |
|-------------------------------|------|
| Wincon-8xx7 / 8xx6 :          | 3.41 |
| i-8417 / 8817 / 8437 / 8837 : | 3.23 |
| i-8437-80 / 8837-80 :         | 3.23 |
| i-7188EG :                    | 2.21 |
| i-7188XG :                    | 2.19 |

Please download the new released driver and the “ICP DAS Utilities For ISaGRAF” at <http://www.icpdas.com/products/PAC/i-8000/isagraf-link.htm> if yours is old.

Please visit [http://www.icpdas.com/products/Remote\\_IO/i-87k/i-87017w-a5.htm](http://www.icpdas.com/products/Remote_IO/i-87k/i-87017w-a5.htm) for More information about the i-87017W-A5.

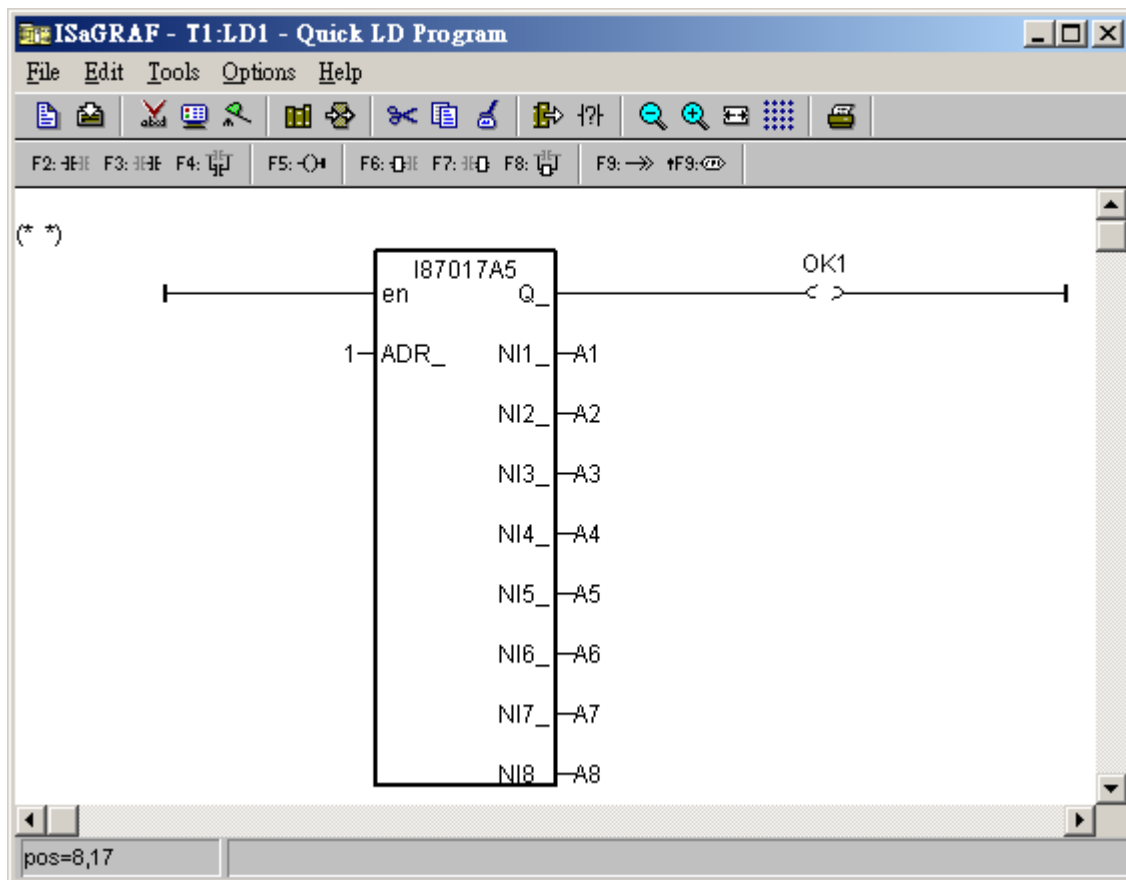
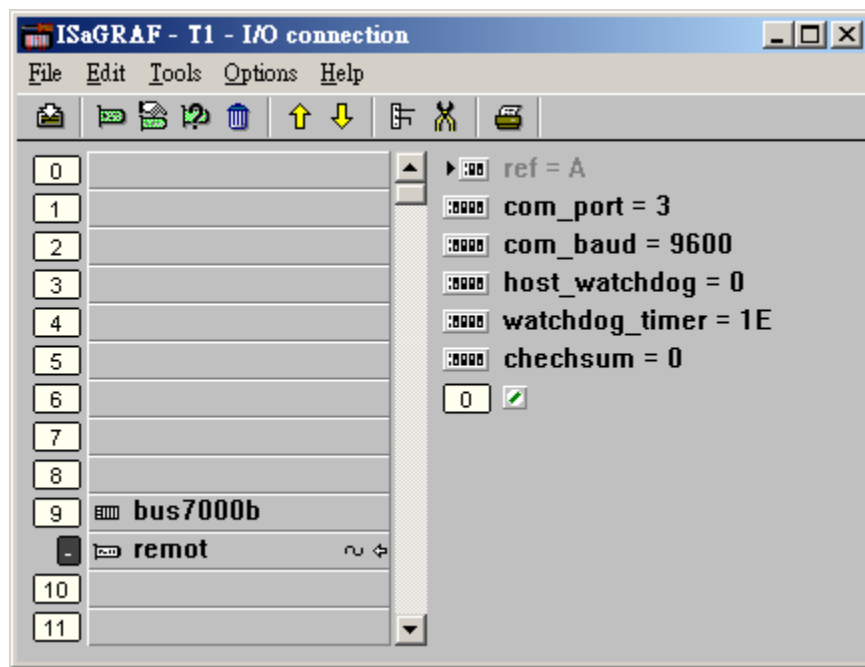
To use the i-87017W-A5 in the Wincon-8xx7 's slot 1 through 7 or in the i-8xx7 's slot 0 through 7. Please connect “i87017a5” in the ISaGRAF I/O connection window.



If the i-87017W-A5 is plugged in the RS-485 remote expansion base like the i-87K4/5/8/9 or the Ru-87P1/2/4/8, please connect “bus7000b” in the ISaGRAF I/O connection window first. Set proper communication parameters, then using the “i87017a5” function block in the programs . **(Please refer to chapter 6** for more information about programming the RS-485 remote I/O). The i-87017W-A5 's DCON utility setting should be **2's compliment format**. Type code can be 1B or 1C.

|      |       |     |   |     |       |                          |     |   |     |         |
|------|-------|-----|---|-----|-------|--------------------------|-----|---|-----|---------|
| 1B : | -150V | --- | 0 | --- | +150V | (Channel value is -32768 | --- | 0 | --- | +32767) |
| 1C : | -50V  | --- | 0 | --- | +50V  | (Channel value is -32768 | --- | 0 | --- | +32767) |





## 3.3: Linking Some Special Virtual board

### 3.3.1: Using “Push4Key” and “Show3Led”

**Note:**

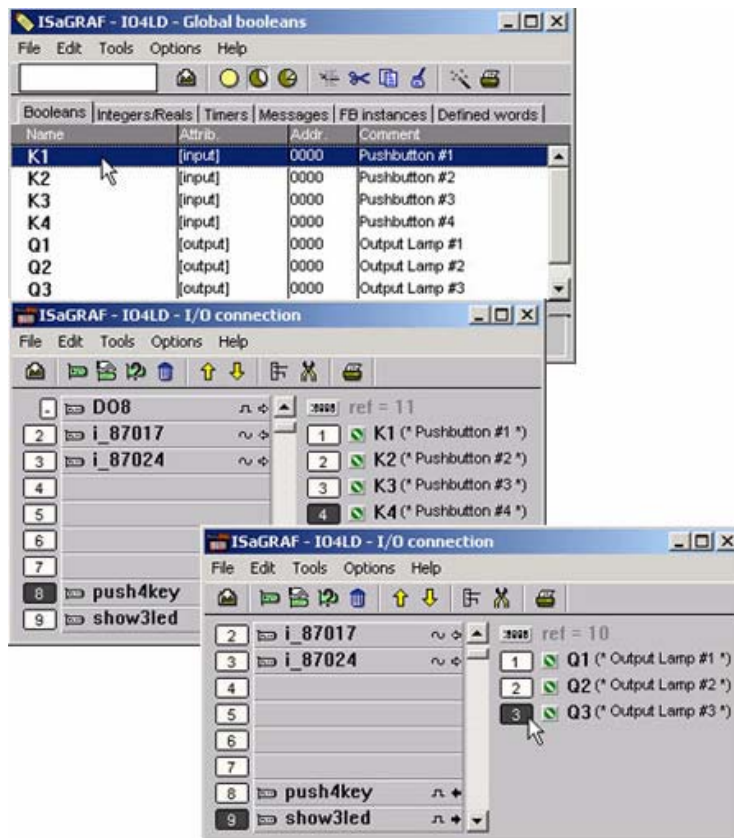
**I-7188EG/XG & Wincon-8037/8337/8737 doesn't support “Push4Key” & “Show3Led”**

The "Push4Key" are the four pushbuttons on the I-8xx7 control front panel and they are handled as digital inputs. The "Show3Led" are three of the four LED's on the I-8xx7 control front panel (the first three from left to right, the fourth LED is strictly to show if the power is turned on the I-8xx7 controller system) and they are handled as digital outputs.

Both of these can be linked to an ISaGRAF program through the "I/O Connection" window and can be used to interface with Man Machine Interface (MMI) programs or for program debugging. It is recommended that you assign these functions to slot 8 or higher (remember, slots 0 through 7 are reserved for real I/O boards).

**IMPORTANT NOTE:**

As with any real digital input or real digital output, you **MUST** declare a variable name for each of the "Push4Button" inputs and "Show3Led" outputs in the "ISaGRAF Global Variables" window **BEFORE** they can be assigned to an ISaGRAF program.



### 3.3.2: Using “io\_state” to test the operation state of real I/O boards

I-8xx7 and W-8xx7 / W-8xx6 controllers support “io\_state” to test the operation state of real I/O board in slot 0 through slot 7. The failure state are as below.

1. Wrong I/O board plugged in the slot.
2. I/O board is absent.
3. I/O board is damaged which cause the ID byte can not be read by the controller.

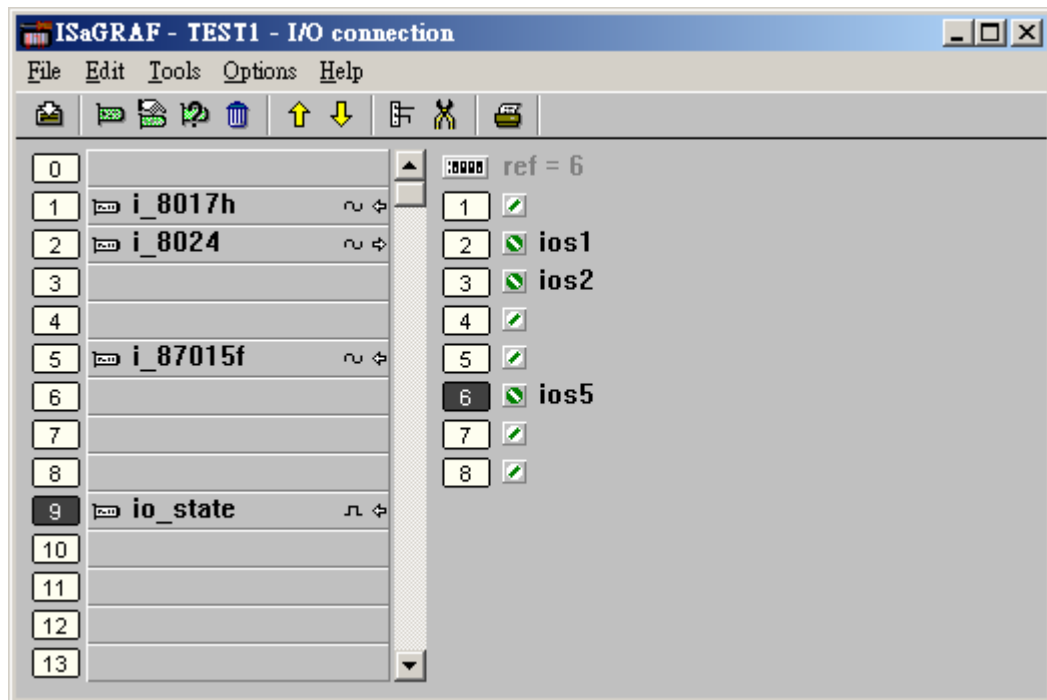
#### Note:

1. “io\_state” can not detect i-8112 / 8114 / 8142 / 8144 / 8142i, i-8072 , i-8073 , i-8074, i-8120, i-8172 , i-87120 and i-8090 / 8091 / 8092 / 8093 / 8094 .
2. I-8xx7 and W-8xx7 / 8xx6 doesn't support “Hot-Swap” in slot 0 to slot 7. Please **DO NOT** plug in IO boards when controller power is ON. This action may make the IO board and backplane damaged.

“io\_state” can only detect the real I/O boards in slot 0 through 7 (Wincon is slot 1 through 7). If the slot No. in the ISaGRAF I/O connection window is empty, it is not detectable.

To use “io\_state” , please connect it at slot 8 or larger slot No in the I/O connection window.

The below example will detect if slot 1 : i-8017H and slot 2: i-8024 and slot 5: i-87015 working normal or failure. It means working normal if the related channel is TRUE, False means failure or can not find the related I/O board (Wincon doesn't have channel 1 – “Slot 0” )



### 3.3.3: Using “echo\_tim” to delay some milli-seconds before the Modbus RTU Slave port replying to the PC / HMI / SCADA

I-8xx7 and W-8xx7 / 8xx6 controllers support “echo\_tim” to delay some milli-seconds before the Modbus RTU slave port to replying to the PC / HMI / SCADA. I-8xx7 can control only one port to delay the reply. However Wincon can control max. 5 RTU slave port to delay reply (Please refer to Chapter 3.3.4).

The “echo\_tim” can control the below Modbus RTU Slave port.

I-8xx7 : One of COM1 and COM2 and COM3

W-8xx7 / 8xx6 : One of COM2 and COM3.

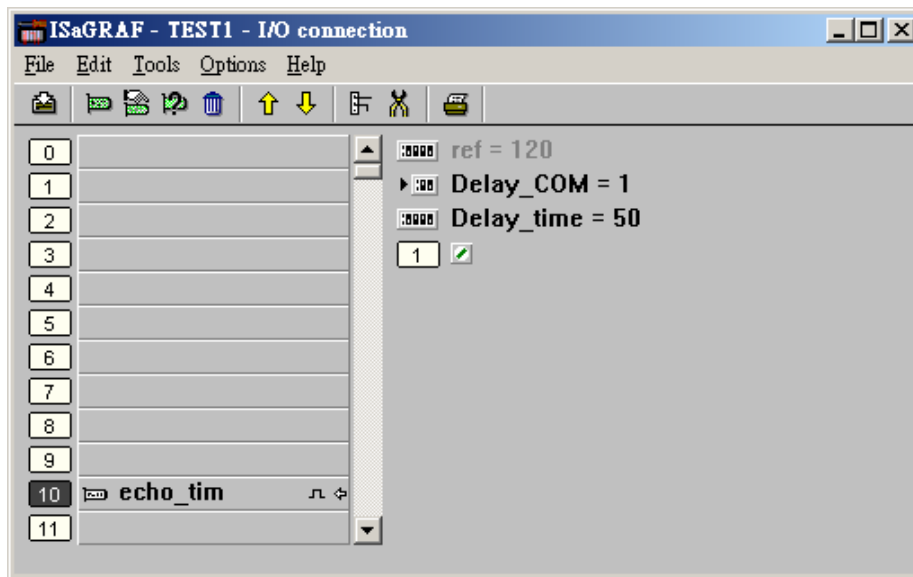
(Please refer to Chapter 3.3.4 for COM5 to COM8 in W-8xx7)

To use “echo\_tim”, please connect it at slot 8 or larger slot No.

The first “Delay\_COM” setting is the COM port No. to be delay. I-8xx7 can be 1, 2 or 3. W-8xx7/8xx6 can be 2 or 3. The second “Delay\_time” is the milli-seconds to delay. It can be 1 to 10000 (unit is 0.001 second)

It means Ok if Channel 1 returns FALSE. If it returns TRUE, something wrong, for example, setting wrong COM port No.

The below example set I-8xx7 ‘s COM1 Modbus RTU Slave port to delay 50 ms before responding to PC / HMI .



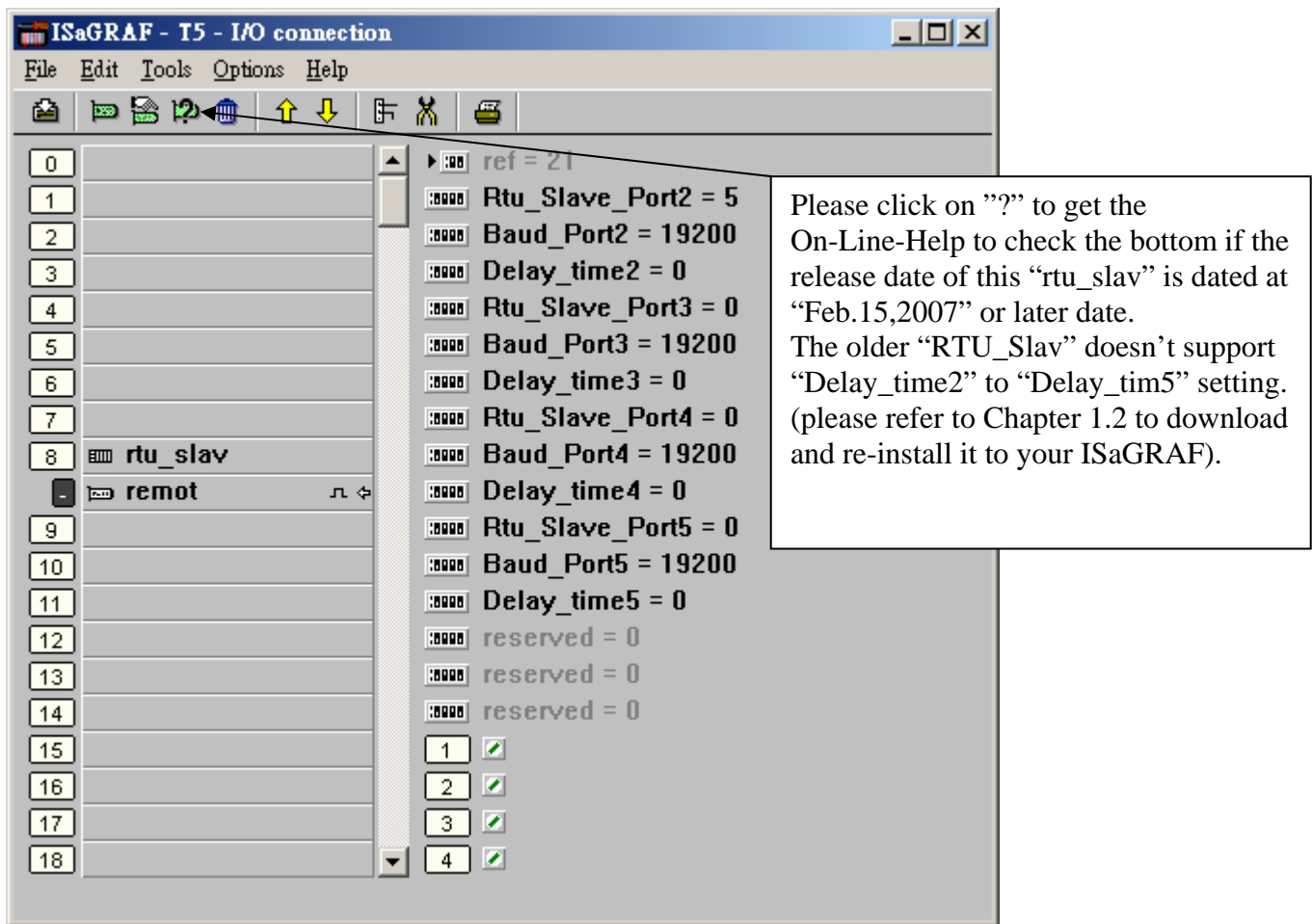
#### Why Modbus RTU Slave port need to delay ?

For example, there is some application applying some communication equipment (like radio modem) which need some time to switch the sending / receiving state. When PC / HMI send Modbus RTU request to I-8xx7 and Wincon-8xx7 via this “radio modem”, the I-8xx7 and W-8xx7 should not reply immediately. They should delay some milli-seconds to wait this “radio modem” to complete its switching to send / receive state, then reply.

### 3.3.4: Using “RTU\_Slav” to expand more Modbus RTU Slave ports in Wincon

Wincon -8xx7 / 8xx6 can configure one of its COM2:RS-232 and COM3:RS-485 to become a Modbus RTU slave port. To expand more Modbus RTU slave ports, please plug one or two i-8112 / 8114 / 8142 / 8144 serial communication board in slot 1 or 2 (COM5 to COM8). Then please refer to Appendix E and G of the W-8xx7 “Getting Started Manual” to setup COM5 to other COM No. first .

Please connect “RTU\_Slav” at slot 8 or larger slot No as below.



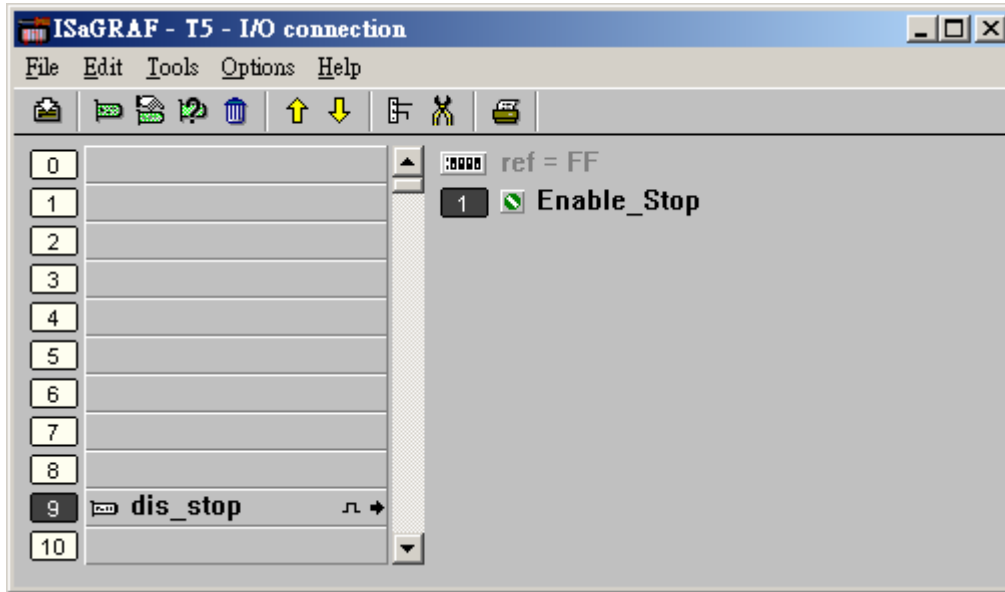
“Rtu\_Slave\_Port2” to “Rtu\_Slave\_Port5” setting can be 0, 5, 6, 7 or 8 . Setting as 0 means not enable the Modbus RTU Slave port.

“Baud\_Port2” to “Baud\_Port5” setting can be 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 or 115200.

“Delay\_time2” to “Delay\_time5” setting can be 0 or 1 to 10000. unit is 0.001 second. It specifies the delaying milli-seconds before replying to the PC / HMI / SCADA software. Why dealy ? Please refer to the description of Chapter 3.3.3.

### 3.3.5: Using “dis\_stop” to disable / enable the ISaGRAF Download function

For some reason, to prevent someone to use ISaGRAF software to stop or to download a different controller project already running in the i-7188EG , I-8437/8837 and W-8xx7, the “Dis\_stop” can be applied . Please connect “dis\_stop” at a slot No. larger than 8 and init the channel value to become TRUE. Then stop / download command is not allowed in this controller.



To disable “Dis\_stop” to accept stop / download command, please run the original ISaGRAF project to link to this controller and set the channel value to become False.

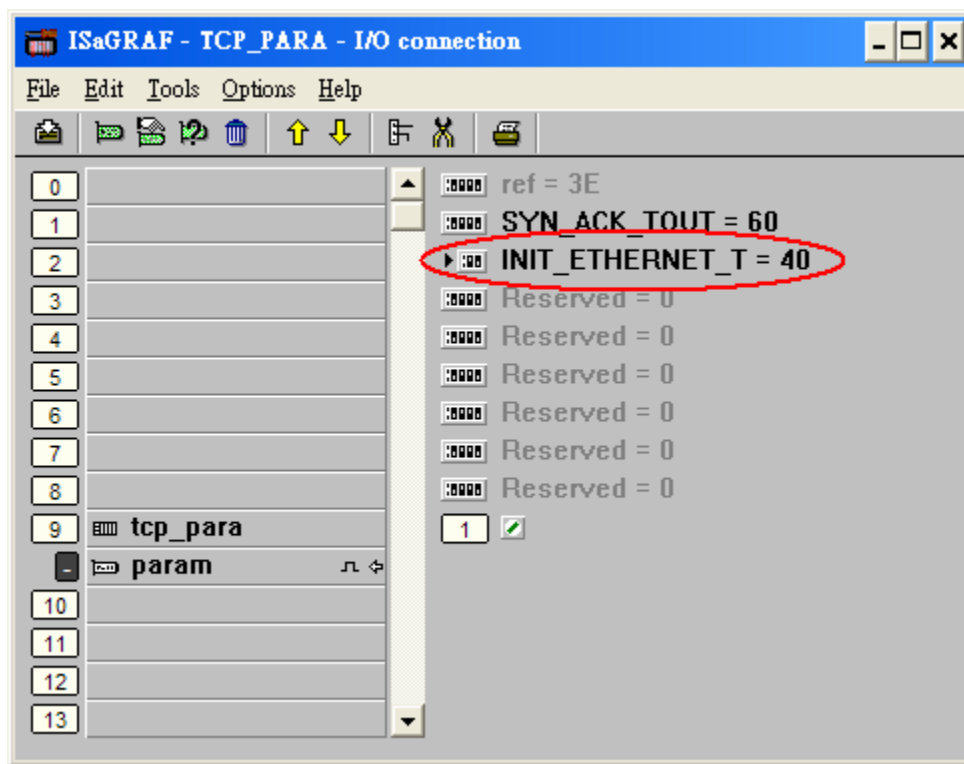
Please refer Chapter 19 for more information about the Internet security.

### 3.3.6: Using “tcp\_para” : Why are my i-8x37, i-8x37-80, i-7188EG and uPAC-7186EG 's TCP connections disconnected and then recovered about every 20/40 hours ?

In order to avoid the TCP sockets staying idle for a longer time, these controllers will default reset all TCP connections about every 20/40 hour .

If this feature is not necessary and you want to disable it, please link the “tcp\_para” in the I/O connection window as below (The "tcp\_para" is belong to "IO complex equipment" .)

The "INIT\_ETHERNET\_T" parameter can be set to configure the TCP recycling time. The value of this parameter could be set as “0” or any value between 10 to 300(hours). To disable this recycling TCP socket feature, please set the "INIT\_ETHERNET\_T" parameter as “0”.



## 3.4: Directly Represented Variables

If you have an ISaGRAF-256 or ISaGRAF-L workbench (Version 3.4x or 3.5x ) with a dongle, you don't need to use the skill described in this section.

A very useful feature of the ISaGRAF Workbench program is the ability to create "directly represented (or internal)" variables. Internal variables are program variables that can be used in an ISaGRAF program, but they are not physically connected to any of the input or output variables. There are four versions of the ISaGRAF Workbench program available with the I-8xx7 controller system: ISaGRAF-32, ISaGRAF-80, ISaGRAF-256, and ISaGRAF-L. The number after "ISaGRAF" represents the number of I/O variables that are allowed with that particular ISaGRAF Workbench program.

The ISaGRAF Workbench program comes with a hardware protection device (dongle) that plugs directly into your development computers parallel port. Every time you compile a program in ISaGRAF the hardware protection device is read to make sure that you are not trying to connect to more program variables than are allowed with your particular copy of the ISaGRAF Workbench program that you purchased with your I-8xx7 controller system.

These "directly represented (henceforth called "internal") variables can be used in lieu of your real world inputs and outputs so you can create additional program variables that do not count against the amount of ISaGRAF program variables. The only "caveat emptor" to these internal variables is that you must follow a strict programming scheme to program and access these internal variables, and they are more complicated to create than the regular input and output variables. **For a professional programmer, recommend to purchase an ISaGRAF-256 workbench rather than an ISaGRAF-80 or ISaGRAF-32 workbench for programming on I-8xx7, I-7188EG/XG & Wincon-8xx7 controllers.**

Single Type Internal Variable Programming Scheme:

For single-typed board: "s" is the slot No, "c" is the channel No.

|        |   |            |
|--------|---|------------|
| %IXs.c | free channel of a <b>boolean input</b> board,   | ex. %IX2.3 |
| %QXs.c | free channel of a <b>boolean output</b> board,  | ex. %QX0.2 |
| %IDs.c | free channel of an <b>integer input</b> board,  | ex. %ID3.1 |
| %QDs.c | free channel of an <b>integer output</b> board, | ex. %QD2.4 |
| %ISs.c | free channel of a <b>message input</b> board,   | ex. %IS3.1 |
| %QSs.c | free channel of a <b>message output</b> board,  | ex. %QS2.4 |

Complex Type Internal Variable Programming Scheme:

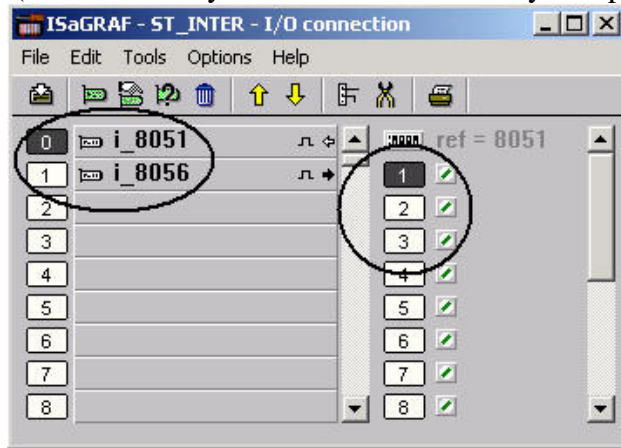
For complex board: "s" is the slot No, "b" is the index of the single board within the complex equipment. "c" is the channel No.

|          |   |              |
|----------|---|--------------|
| %IXs.b.c | free channel of a <b>boolean input</b> board,   | ex. %IX2.3.2 |
| %QXs.b.c | free channel of a <b>boolean output</b> board,  | ex. %QX0.2.1 |
| %IDs.b.c | free channel of an <b>integer input</b> board,  | ex. %ID3.1.3 |
| %QDs.b.c | free channel of an <b>integer output</b> board, | ex. %QD2.4.3 |
| %ISs.b.c | free channel of a <b>message input</b> board,   | ex. %IS3.3.1 |
| %QSs.b.c | free channel of a <b>message output</b> board,  | ex. %QS2.1.4 |

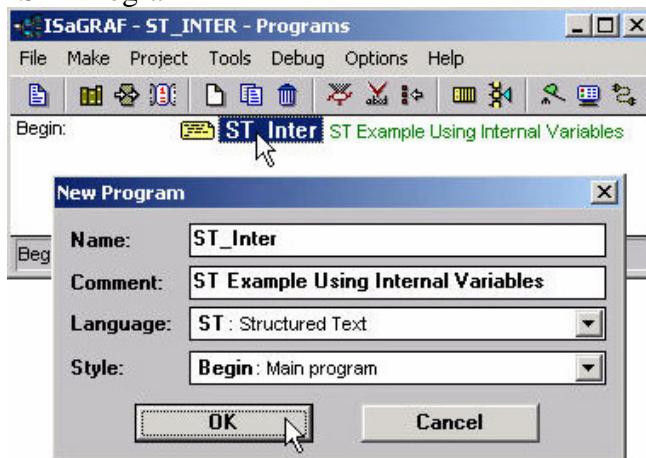


## An Internal Variable Program Example

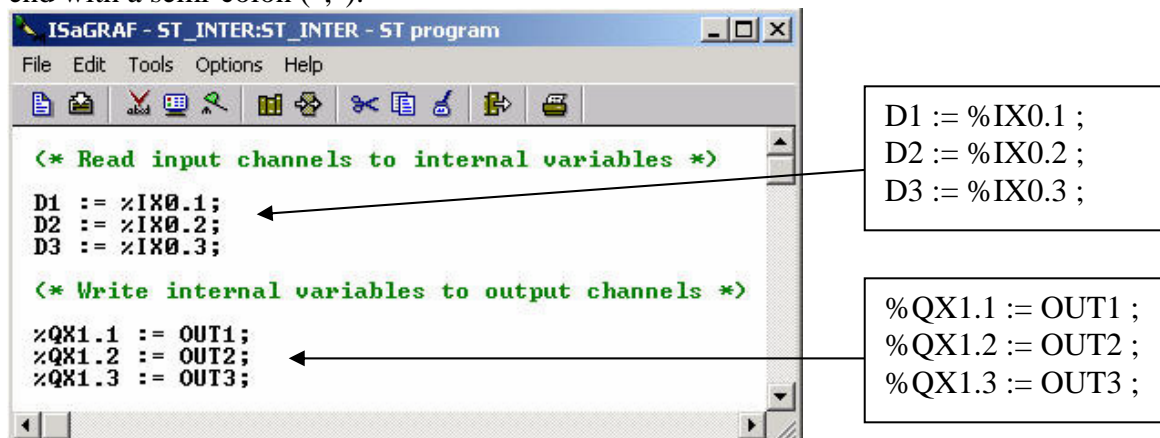
Create a new project for an ISaGRAF ST program, and then create a link to the I/O boards that are specified in the window below. Declare three input variables called "D1", "D2", & "D3" for the I-8051 board located at slot 0, and then create three output variables called "OUT1", "OUT2", & "OUT3" for an I-8056 board located at slot 1. This time set each of their respective attributes to "internal" instead of input or output (this means they are not connected to any real physical I/O).



## Create A New "ST" Program



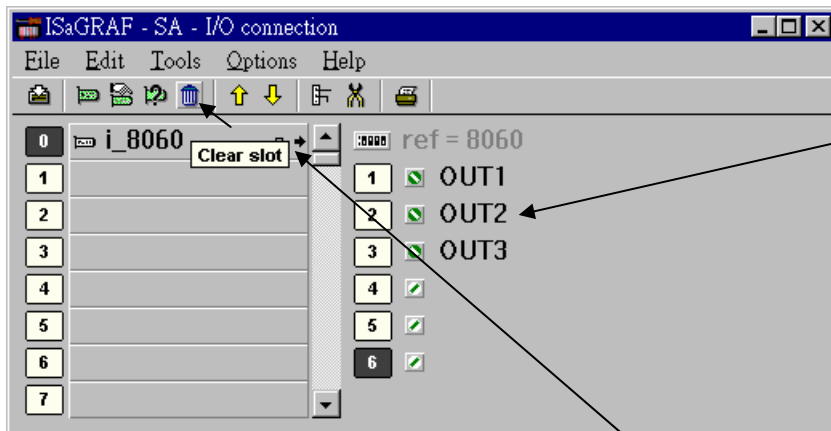
Double click on the "ST\_Inter" that is highlighted and the "ISaGRAF ST Program" window will open. Type in the program code displayed in the window below EXACTLY as shown. Remember, each line MUST end with a semi-colon (";").



Now we can use the internal variables D1 through D3 and OUT1 through OUT3 that have been created in other programs in the same project. The newly created internal variables will generate input and output actions to the associated channels in this ST program.

#### IMPORTANT NOTE:

If once the input or output attributed variables have been connected to an connected IO board or complex equipment, and if they would like to be replaced by Directly represented variables, these input or output attributed variables have to be re-attributed to “internal” and the board or equipment **must be re-connected to the slot**.



If you wish to replaced these variables by directly represented variables, re-attributed them to “internal” attribution in the “dictionary” window.

Clear slot and re-connect again.

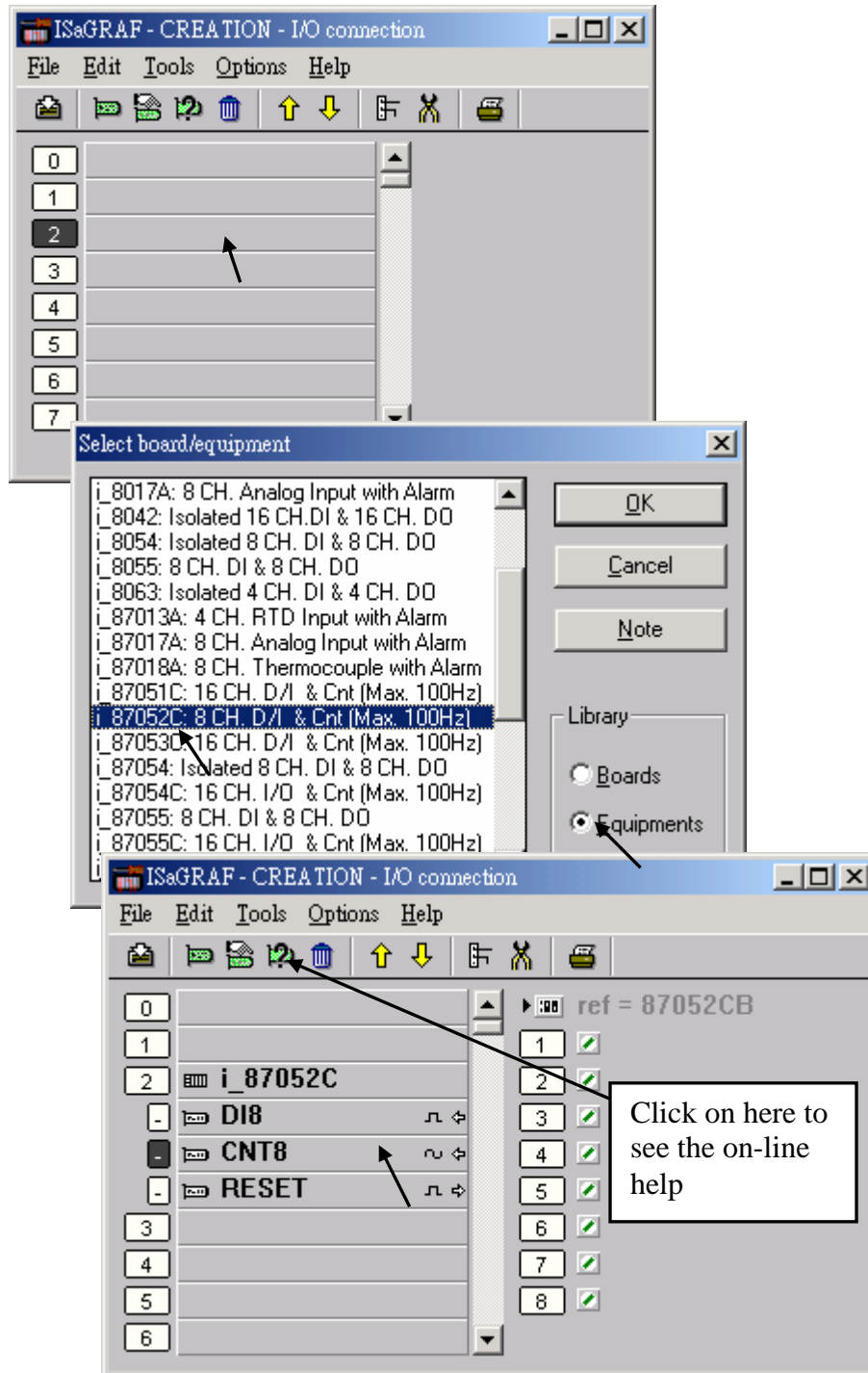
#### IMPORTANT NOTE

If you enable the compiler option of upload, option “**Comments for not connected I/O channels**” must be choosed if “Directly represented variables” is used in this project (refer to section 9.2).

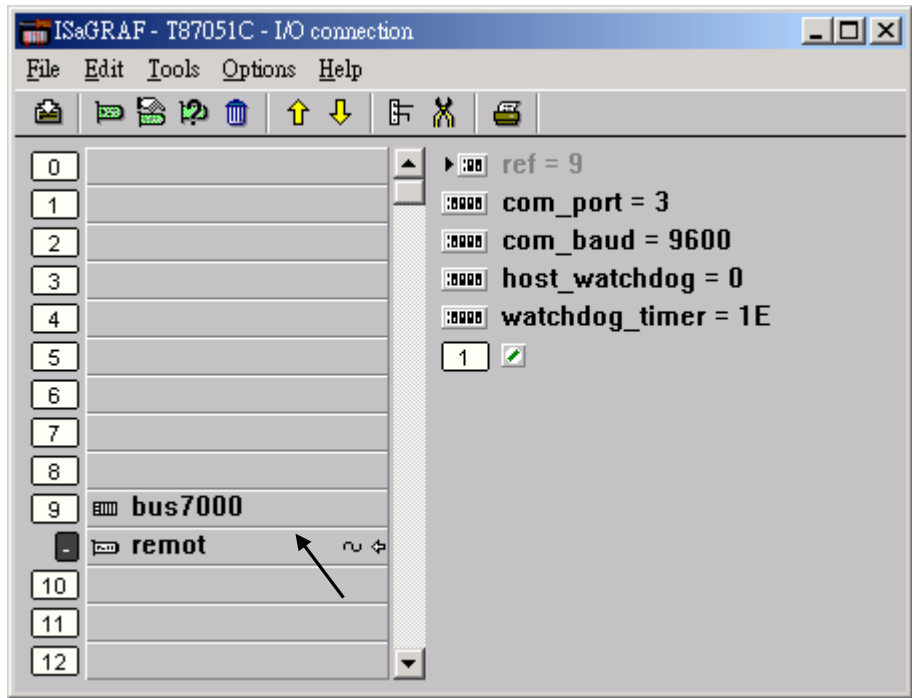
### 3.5: D/I Counters Built in The I-87xxx D/I Modules

87051, 87052, 87053, 87054, 87055, 87058 & 87063 have built-in low speed D/I counters associated with each D/I channel. The max counter speed of these modules is 100Hz. The counter value is ranging from 0 to 65535 and can be reset to 0.

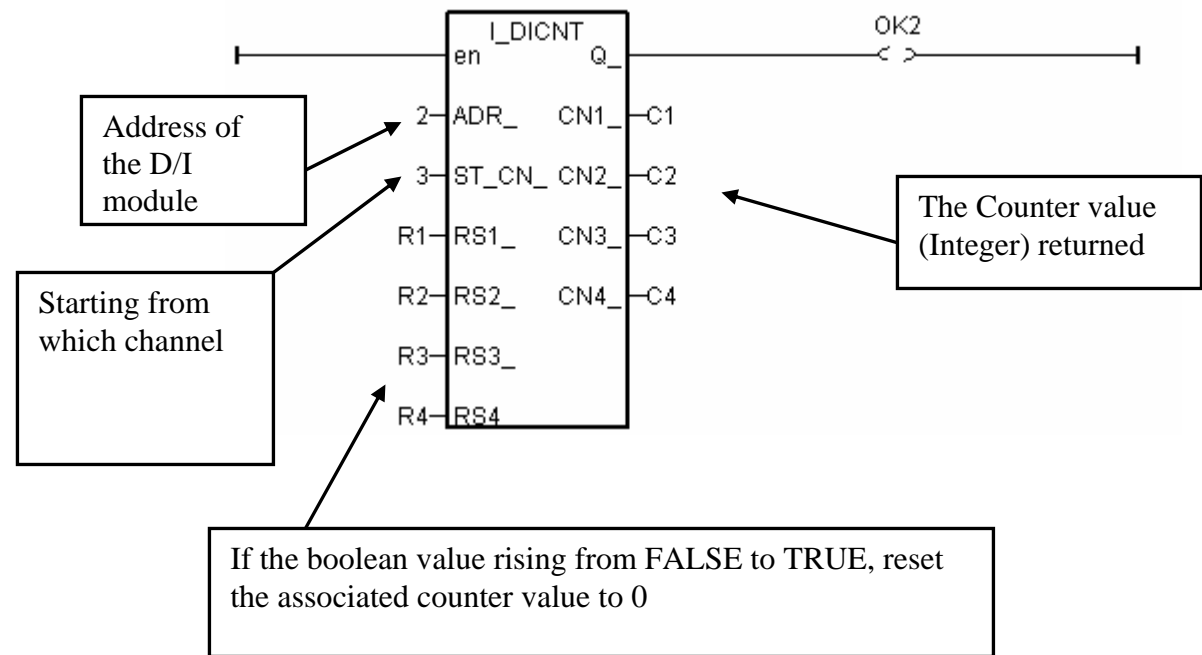
To use these D/I counters, connect these I/O modules with a last character – “C”. For ex. “i\_87052C”.



If the I-87xxx D/I Module is plugged in the 87K4, 87K5, 87K8 & 87K9 extension base module, or the I-7000 D/I module is used, Please refer to Chapter 6 to use “7000 utility” to set the appropriate address, baud rate , then connect “Bus7000” on the ”I/O connection” window.



Then using “I\_DiCnt” block to get the “D/I Counter” value. Each “I\_DiCnt” can get 4 counters.



## 3.6: Auto-Scan I/O

Before you can use Auto-scan I/O utility, make sure the “ICP DAS Utilities For ISaGRAF” has been installed. (please refer to section 1.2)

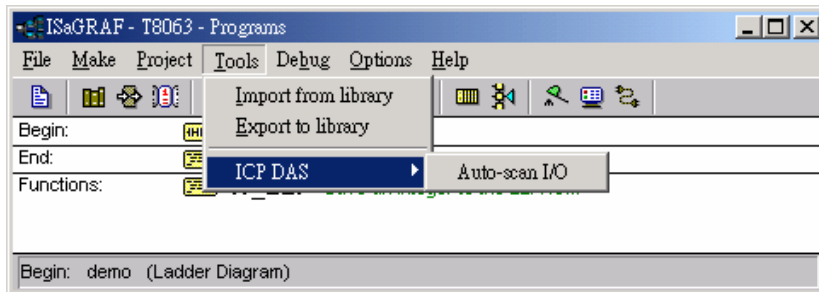
What is Auto-scan I/O :

It's a tool for ISaGRAF to easily configure your I/O connection and automatically declare variables for each I/O channel in I-8xx7 and Wincon-8xx7 controllers. (**I-7188EG/XG and uPAC-7186EG don't support Auto-scan I/O**)

How to use ?

A. Open your ISaGRAF program.

B. Click on “Tools/ICP DAS/Auto-scan I/O” to run Auto-scan.

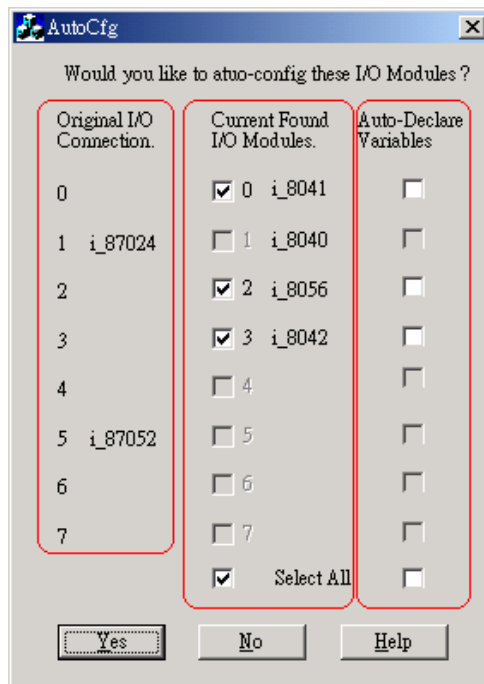


C. The Auto-scan I/O is divided into three area.

**Original I/O Connection** shows the modules that already exist in your I/O connection at the first eight slots of your ISaGRAF project.

**Current Found I/O Modules** shows the I/O modules that detected in your controller (By RS232 or TCP/IP).

**Auto-Declare Variables** shows what modules that you want Auto-scan to automatically declare variables for you also.



D. In the “Current Found I/O Modules.” area:

The check box will be enable only when an I/O module is detected in the controller and the slot is **not used** by original I/O connection.

E. In the “Auto-Declare Variables”:

The check box can be enable only when one I/O module **is checked** in the current found area.

F. You can check the “Select All” to check all available boxes in the respective area.

What is necessary for Auto-scan I/O ?

A. Make sure the “Link setup” parameter is correct.

B. Plug in I/O boards first before your ISaGRAF can detect them.

Naming rules of automatically declared variables

Name format : Type\_Slot\_Channel

Type:

Digital Input : DI

Digital Output : DO

Analog Input : AI

Analog Output : AO

Slot : one digital slot number.

Channel : two digital channel number.

For ex. :

DI\_0\_02 , Digital Input channel at channel No.2 of slot 0.

AI\_5\_06 , Analog Input channel at channel No.6 of slot 5.

DO\_2\_12, Digital Output channel at channel No.12 of slot 2.

AO\_1\_03, Analog Output channel No. 3 of slot 1.

**Note:**

I-8xx7 & Wincon-8xx7 supports “Auto-Scan”, however I-7188EG/XG doesn’t support it.

## 3.7: PWM Output

The scan time of the ISaGRAF controller depends on the ISaGRAF program and the hardware driver. For normal usage, the scan time is about 5 to 40 ms. It may go up to 100 ms sometime when the user's ISaGRAF program is very complicated. It is not easy to generate a precise periodic pulse output because the scan time of ISaGRAF is always varying, for example, a square curve of 2 ms OFF & then 1 ms ON. To achieve this kind of application, ISaGRAF provide PWM output functions.

To use PWM output (Pulse Width Modulation) in **I-8417/8817/8437/8837**, please update the driver to version of **2.43** or higher. Only parallel Output boards are supported, not for serial boards. The following output boards are available with the PWM function.

I-8037, 8041, 8042, 8054, 8055, 8056, 8057, 8060, 8063, 8064, 8065, 8066, 8068, 8069

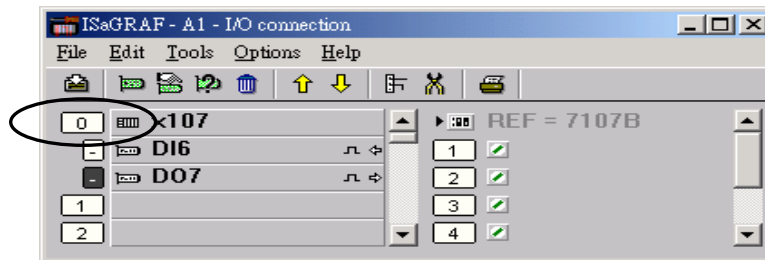
**Wincon** supports PWM functions since its driver version of 3.23.

To support PWM function in **I-7188EG**, please update the driver to version of **1.35** or higher, while **1.32** or higher for **I-7188XG**

Only the Xxxx boards with digital output channels are available with PWM function.

### Note:

1. Max 8 digital outputs can call PWM\_en, PWM\_en2, pwm\_ON & pwm\_OFF at the same time.
2. I-7188EG/XG must connect the Xxxx board at slot 0, or the PWM function will not work.



**PWM\_dis**      Disable PWM output

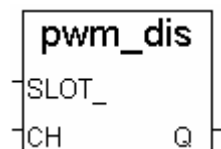
### Parameters:

SLOT\_      integer      Which slot ? 0 ~ 7 for I-8xx7, only 0 for I-7188EG/XG. Wincon: 1 to 7

CH\_      integer      Which channel ? 1 ~ 32.

### Return:

Q\_      boolean      TRUE: Ok .  
FALSE: wrong input parameters, too many PWM outputs been enable, or the associate output channel is not found.



### Note:

1. After calling PWM\_dis, the associate output will then be controlled by the ISaGRAF cycle engine.
2. Max 8 output channels can call PWM\_en, PWM\_en2, pwm\_ON, pwm\_OFF at one controller.

Example: I-8xx7: demo\_63 , Wincon: Wdemo\_22

**PWM\_en** Enable PWM to output until PWM\_dis is called

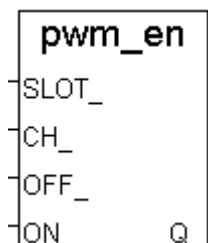
Parameters:

|       |         |  |
|-------|---------|--|
| SLOT_ | integer | Which slot ? 0 ~ 7 for I-8xx7, only 0 for I-7188EG/XG. Wincon: 1 to 7                                |
| CH_   | integer | Which channel ? 1 ~ 32.  |
| OFF_  | integer | Off time, I-8xx7 & I-7188EG/XG: 1 ~ 32,767, scale is 1 ms. Wincon: 2 ~ 32766 (Wincon's scale is 2ms) |
| ON_   | integer | On time, I-8xx7 & I-7188EG/XG: 1 ~ 32,767, scale is 1 ms. Wincon: 2 ~ 32766 (Wincon's scale is 2ms)  |

Return:

|    |         |  |
|----|---------|--|
| Q_ | boolean | TRUE: Ok .<br>FALSE: wrong input parameters, too many PWM outputs been enable, or the associate output channel is not found. |
|----|---------|--|

Example: I-8xx7: demo\_63 , Wincon: Wdemo\_22



**PWM\_en2** Enable PWM to output a given number of pulse

Parameters:

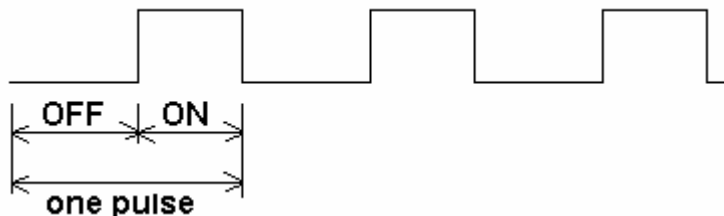
|       |         |  |
|-------|---------|--|
| SLOT_ | integer | Which slot ? 0 ~ 7 for I-8xx7, only 0 for I-7188EG/XG. Wincon: 1 to 7  |
| CH_   | integer | Which channel ? 1 ~ 32.  |
| OFF_  | integer | Off time, I-8xx7 & I-7188EG/XG: 1-32,767, scale is 1ms. Wincon: 2 ~ 32766 (Wincon's scale is 2ms)  |
| ON_   | integer | On time, I-8xx7 & I-7188EG/XG: 1-32,767, scale is 1 ms. Wincon: 2 ~ 32766 (Wincon's scale is 2ms)  |
| NUM_  | integer | number of pulse to output, 1 - 2,147,483,647. If gives a negative value to NUM_, for ex. -1, it will ouput indefinitely until pwm_dis been called. |

Return:

|    |         |   |
|----|---------|---|
| Q_ | boolean | TRUE: Ok . FALSE: wrong parameters, too many PWM outputs been enable, or the associate output channel is not found. |
|----|---------|---|

Example: I-8xx7:demo\_63, Wincon: Wdemo\_22

PWM output curve:



Note:

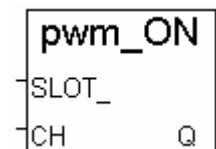
1. Every time the PWM\_en or PWM\_en2 is called, it will reset its internal tick to 0, and re-start ticking to OFF, ON, OFF, ON, ...
2. If the given number of pulse of pwm\_en2 is reached, it will stop & disable PWM auomatically (Calling PWM\_dis for pwm\_en2 is not necessary).
3. PWM\_sts can be used to test if pwm\_en2 reaches its given number of pulse or not.
4. Max 8 output channels can call PWM\_en, PWM\_en2, pwm\_ON, pwm\_OFF at one controller.
5. Do not enable the channel that is already enable. Please disable it first.



**pwm\_ON**      Set parallel D/O to TRUE immediately

Parameters:

|       |         |   |
|-------|---------|---|
| SLOT_ | integer | Which slot ? 0 ~ 7 for I-8xx7, only 0 for I-7188EG/XG. Wincon: 1 to 7 |
| CH_   | integer | Which channel ? 1 ~ 32.   |



Return:

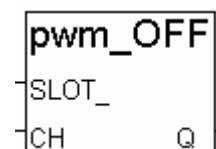
|    |         |  |
|----|---------|--|
| Q_ | boolean | TRUE: Ok .<br>FALSE: wrong input parameters, too many PWM outputs been enable, or the associate output channel is not found. |
|----|---------|--|

Example: demo\_63

**pwm\_OFF**      Set parallel D/O to FALSE immediately

Parameters:

|       |         |   |
|-------|---------|---|
| SLOT_ | integer | Which slot ? 0 ~ 7 for I-8xx7, only 0 for I-7188EG/XG. Wincon: 1 to 7 |
| CH_   | integer | Which channel ? 1 ~ 32.   |



Return:

|    |         |  |
|----|---------|--|
| Q_ | boolean | TRUE: Ok .<br>FALSE: wrong input parameters, too many PWM outputs been enable, or the associate output channel is not found. |
|----|---------|--|

Example: demo\_63

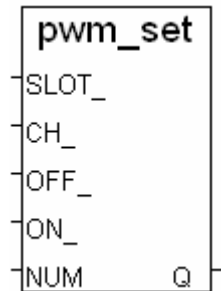
Note:

1. Max 8 output channels can call PWM\_en, PWM\_en2, pwm\_ON, pwm\_OFF at one controller.
2. pwm\_ON will set the associate parallel D/O to TRUE immediately.
3. pwm\_OFF will set the associate parallel D/O to FALSE immediately.
4. If users wish to enable one D/O as PWM output by PWM\_en or PWM\_en2 after pwm\_ON & pwm\_OFF has been called, please disable it first by PWM\_dis, then call PWM\_en or PWM\_en2.

**PWM\_set** Dynamically change the ON\_, OFF\_ & NUM\_ setting

Parameters:

|       |         |  |
|-------|---------|--|
| SLOT_ | integer | Which slot ? 0 ~ 7 for I-8xx7, only 0 for I-7188EG/XG. Wincon: 1 to 7  |
| CH_   | integer | Which channel ? 1 ~ 32.  |
| OFF_  | integer | Off time, I-8xx7 & I-7188EG/XG:1-32,767, scale is 1ms. Wincon: 2 ~ 32766 (Wincon's scale is 2ms)   |
| ON_   | integer | On time, I-8xx7 & I-7188EG/XG:1-32,767, scale is 1 ms. Wincon: 2 ~ 32766 (Wincon's scale is 2ms )  |
| NUM_  | integer | number of pulse to output, 1 - 2,147,483,647. If gives a negative value to NUM_, for ex. -1, it will ouput indefinitely until pwm_dis been called. |



Return:

|    |         |   |
|----|---------|---|
| Q_ | boolean | TRUE: Ok . FALSE: wrong parameters, too many PWM outputs been enable, or the associate output channel is not found. |
|----|---------|---|

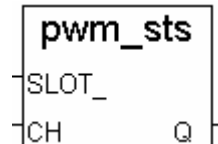
**PWM\_sts** Get PWM status

Parameters:

|       |         |   |
|-------|---------|---|
| SLOT_ | integer | Which slot ? 0 ~ 7 for I-8xx7, only 0 for I-7188EG/XG. Wincon: 1 to 7 |
| CH_   | integer | Which channel ? 1 ~ 32.   |

Return:

|    |         |   |
|----|---------|---|
| Q_ | boolean | TRUE: this channel has been enable<br>FALSE: disable (for pwm_en2 been called, it means the given pulse number is reached). |
|----|---------|---|



**Note:**

1. Max 8 output channels can call PWM\_en, PWM\_en2, pwm\_ON, pwm\_OFF at one controller.
2. This function can be used to test if "PWM\_en2" reaches its given pulse number or not.

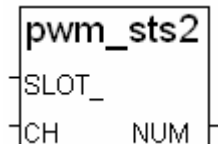
**PWM\_sts2** Get pulse number been output by pwm\_en2 and pwm\_en

Parameters:

|       |         |   |
|-------|---------|---|
| SLOT_ | integer | Which slot ? 0 ~ 7 for I-8xx7, only 0 for I-7188EG/XG. Wincon: 1 to 7 |
| CH_   | integer | Which channel ? 1 ~ 32.   |

Return:

|      |         |   |
|------|---------|---|
| NUM_ | integer | the pulse number already been output by pwm_en2 |
|------|---------|---|



**Note:**

1. This function only works when "pwm\_sts" return True.
2. The returned pulse number may less than the given number in "pwm\_en2" when it reach the destination. For example, gives 20000 pulse in "pwm\_en2", however when reach the end, the "pwm\_sts2" may return only 19998.
3. If the ouput number given in the "pwm\_en2" is a negative value, the pulse output will never stop unless the "pwm\_dis" command is given. Then the returned number of "pwm\_sts2" will become 0 , 1, 2, ... to 2,147,483,647 and then go back to 0, 1, 2, ...

Example: I-8xx7: demo\_63 , Wincon: Wdemo\_22

### 3.8: Counters Built in Parallel D/I Boards

I-8417/8817/8437/8837 supports D/I counters since its driver version of 2.43. Only parallel input boards plug **at slot 0** are supported, not for serial boards. The following input boards are available with D/I counters.

I-8040, 8042, 8051, 8052, 8053, 8054, 8055, 8058, 8063, 8077

**Wincon** supports Di\_Cnt since its driver version of 3.23. **Only its slot 1** can use Di\_Cnt.

I-7188EG supports D/I counters since its driver version of 1.35 while I-7188XG since 1.32. Only the X??? boards with digital input channels are available with D/I counters.

**The max channel amount of parallel D/I counter available in one controller is up to 8. And the max frequency of counter input for I-8xx7 & I-7188EG/XG is up to 500 Hz with minimum NO and OFF width > 1 ms. While Wincon is up to 250 Hz with minimum ON & OFF width > 2 ms.**

The below c function block is for getting/reset D/I counters at slot 0.

Parameters:

RS1\_ ~ RS8\_    boolean    Reset the associated D/I counter when rising from False to True

Return:

Q\_                boolean    work ok. : TRUE. If Q\_ is FALSE , it means "No parallel D/I module found at slot 0 "

CN1\_ ~ CN8\_    integer    DI Counter value of channel No. 1 to 8. Valid value is ranging from 0 to 2,147,483,647. If value is over 2,147,483,647, it    restarts at 0.

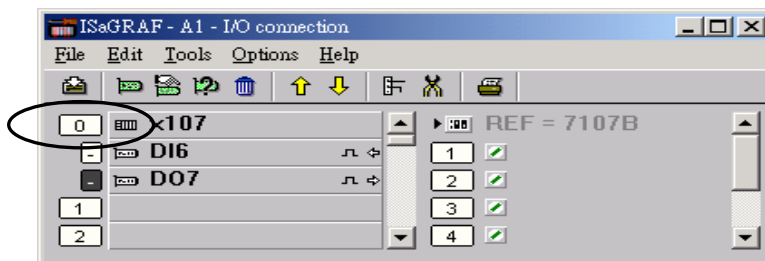
| Di_Cnt |      |
|--------|------|
|        | Q_   |
| RS1_   | CN1_ |
| RS2_   | CN2_ |
| RS3_   | CN3_ |
| RS4_   | CN4_ |
| RS5_   | CN5_ |
| RS6_   | CN6_ |
| RS7_   | CN7_ |
| RS8_   | CN8_ |

**Note:**

I-8xx7 & I-7188EG/XG: Only Parallel D/I board plug in slot 0 support "Di\_Cnt", W-8xx7 is slot 1.

Only the first 8 D/I channel support "Di\_Cnt".

I-7188EG/XG must connect the X??? board at slot 0, or the "Di\_Cnt" will not work.



**Demo:**

Please refer to I-8417/8817/8437/8837's demo\_63.

Wincon: Wdemo\_22

---

## 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 default 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.

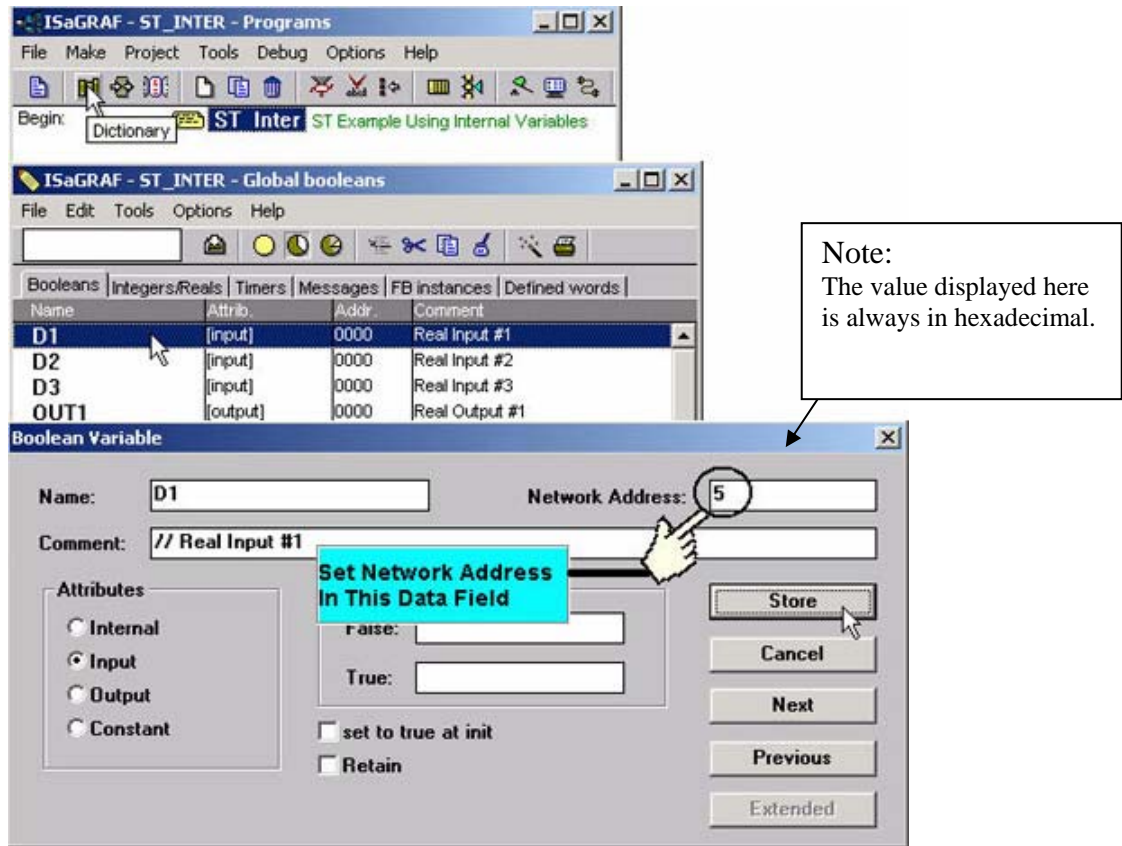
There are two methods available to declare a variable for network address access. The first method is described below. Open an "ISaGRAF Programs" windows and click on the "Dictionary" icon, then double click on the variable to assign a network address number.

**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.

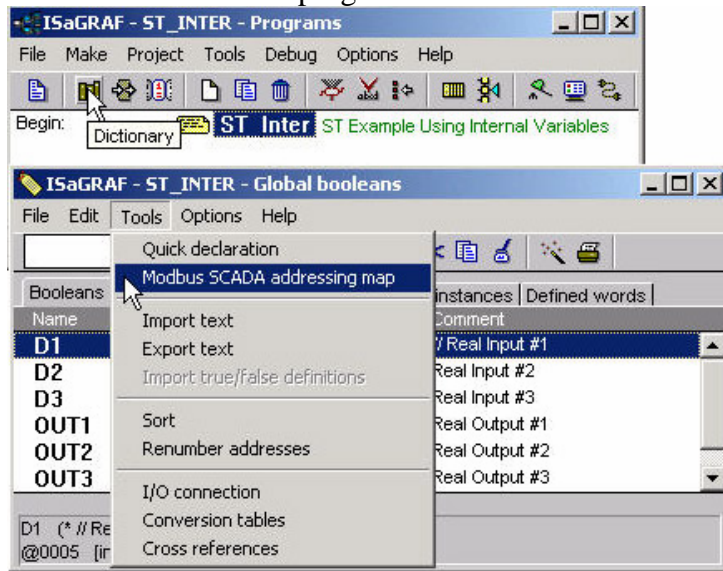
There are two ways to assign a Modbus network address No. to a variable. One is as below figure. (To assign many Modbus Network address No. to the “Variable Array”, please refer to Chapter 2.6)



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.

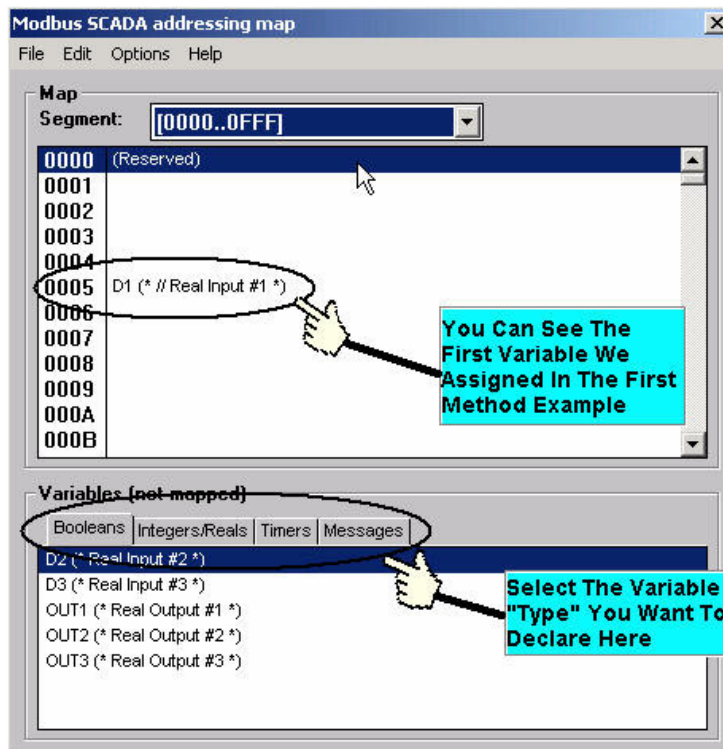


The second method for assigning network addresses to variables requires that you declare the variables BEFORE you assign them. This method allows you to assign numerous network address variables before you link them to an ISaGRAF program.

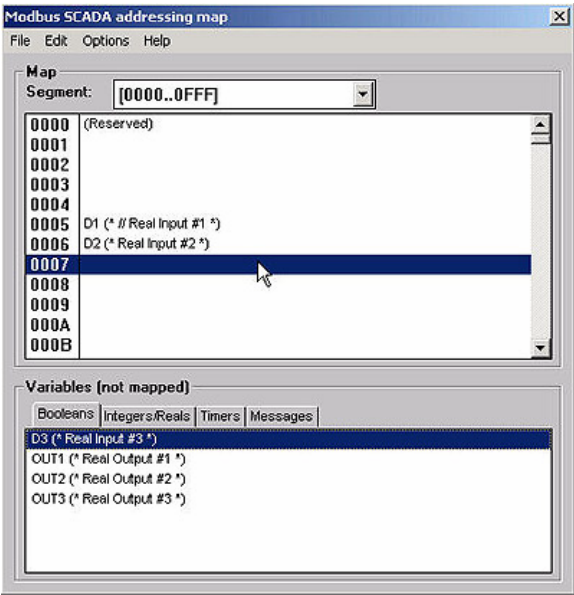


When you click on "Modbus SCADA Addressing Map" (SCADA is an industrial process control acronym that stands for "Supervisory Control And Data Acquisition") the "Modbus SCADA Addressing Map" window will open.

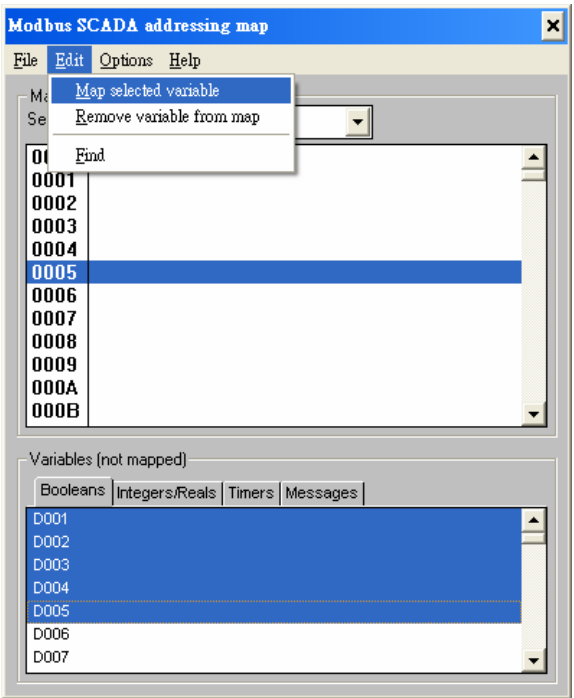
Note that one of the variables (D1) is already assigned from our previous example. You will note that the other variables that are not yet mapped are displayed in the lower portion under the "Variables (Not Mapped)" portion of the "Modbus SCADA Addressing Map" window.



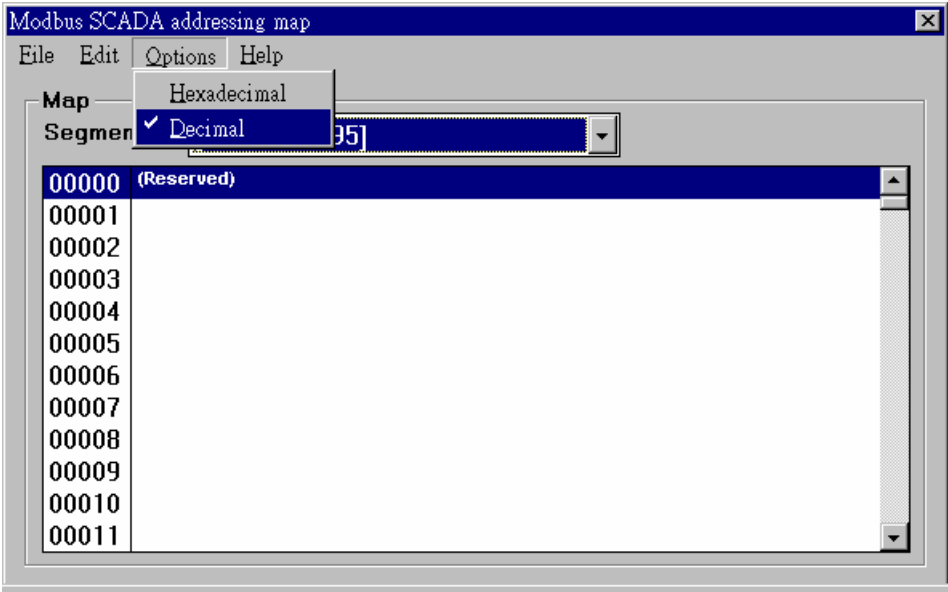
To assign the other variable address click on an unassigned "Map Segment" number, and then double click on the variable you want to assign to the address and the variable will automatically assign itself to the "Map Segment".



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 mistaken 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.

| <u>iFix(Decimal)</u>         | <u>I-8xx7 (Decimal)</u> |
|------------------------------|-------------------------|
| <b>0</b> 0001 (R/W Boolean)  | 1                       |
| ...                          | ...                     |
| <b>1</b> 0010 (Read Boolean) | 10                      |
| ...                          | ...                     |
| <b>3</b> 1000(Read Word)     | 1000                    |
| ...                          | ...                     |
| <b>4</b> 2101(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 Modbus 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.

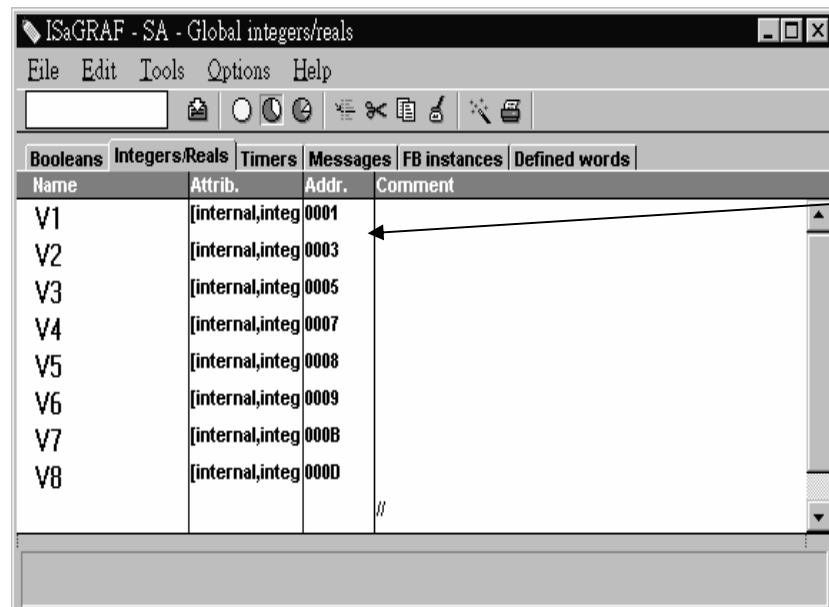
The **word** defined in the Modbus protocol of I-8xx7, I-7188EG/XG, uPAC-7186EG, iPAC-8447/8847 & W-8xx7 controllers is like a signed short integer, which occupies 2 bytes and range from -32,768 (8000 in hexa.) to +32,767 (7FFF in hexa.). It is normally used to describe the behavior of analog I/O channels.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

I-87017 :

| Range ID<br>(hexadecimal) | Electrical<br>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.

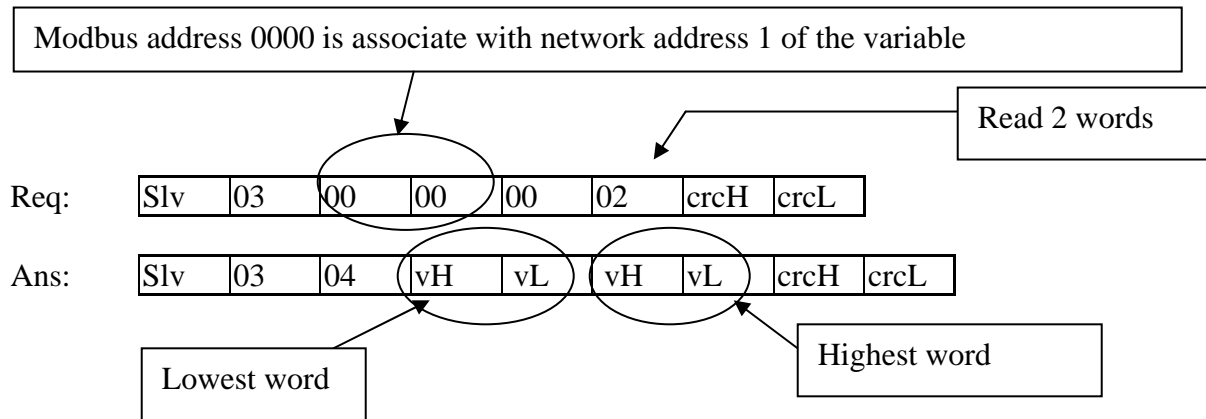


V1 is assigned to a network address “1”. If the network address “2” is not assigned to any other variable, V1 will occupy a long word (4 bytes) in the Modbus transportation formate.

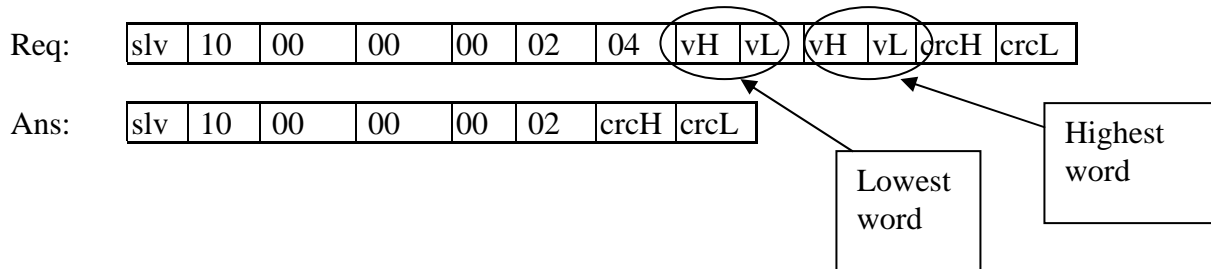
However if “2” is assigned to one another variable, V1 will only occupy one word (2 bytes) in the Modbus transportation format.

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.

Integer/Real Variable

Name: A1 Network Address: 1

Comment:

Unit: Conversion: (none)

Attributes

- ☒ Internal
- ☐ Input
- ☐ Output
- ☐ Constant

Format

☐ Integer (standard)

☒ Real

Initial value: 0

☐ Retain

Store

Cancel

Next

Previous

Extended

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.

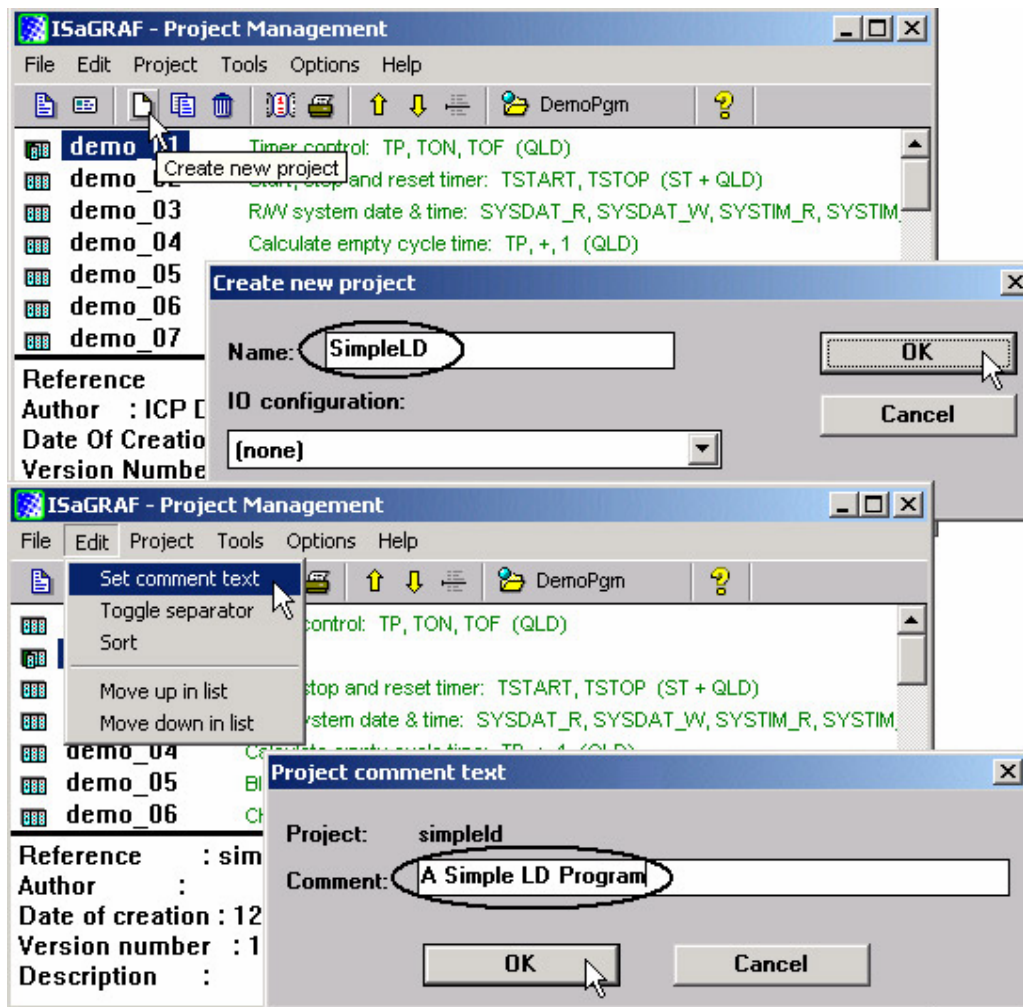
## 4.3: Using I-8xx7 As A Modbus I/O Or A Modbus TCP/IP I/O

There are some configurations that the HMI software gathers the I/O data from some called Modbus I/O modules. These 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).

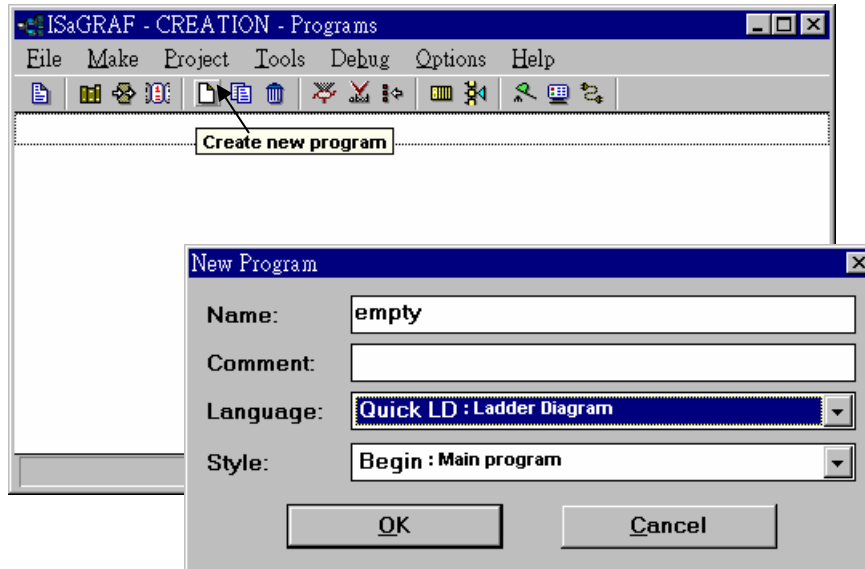
Create a new project

You may refer to section 2.1.1.2

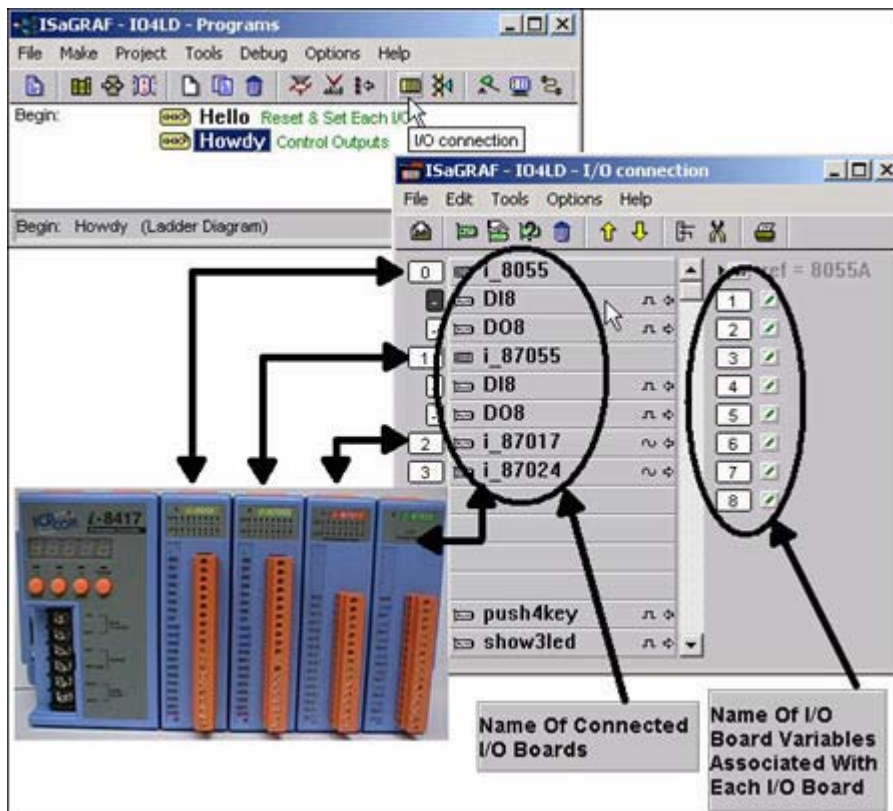
Example:



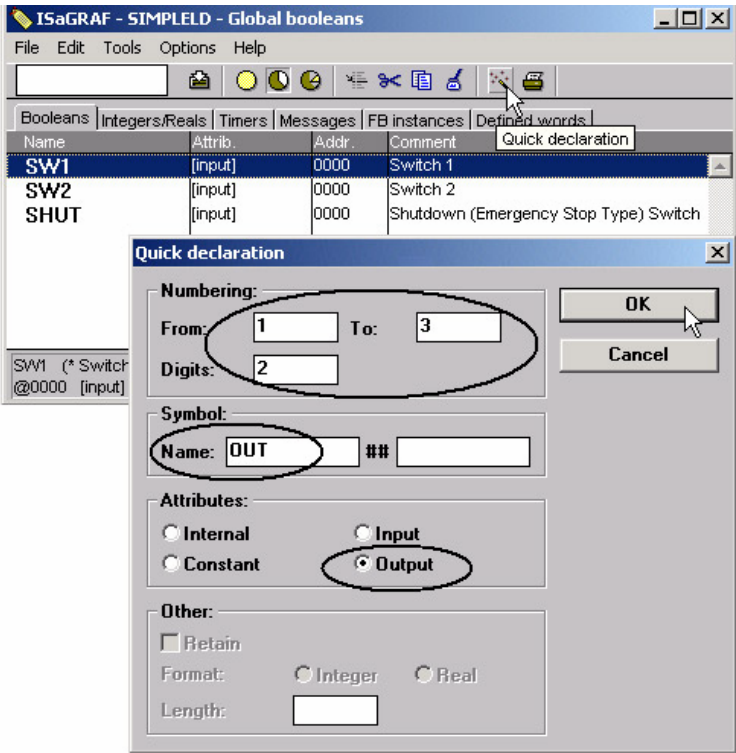
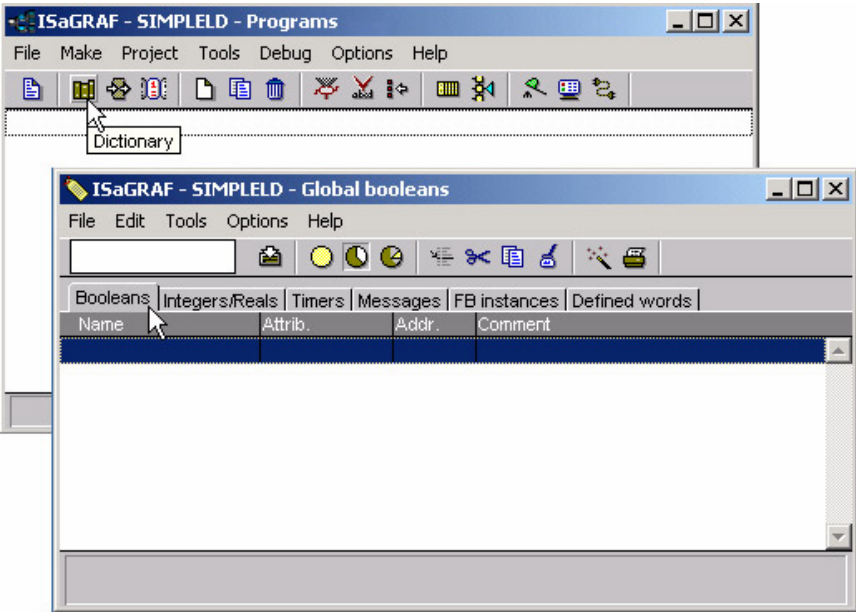
Create an empty program  
No logic need.  
Example:



Connect I/O modules  
You may refer to section 3.1  
Example:



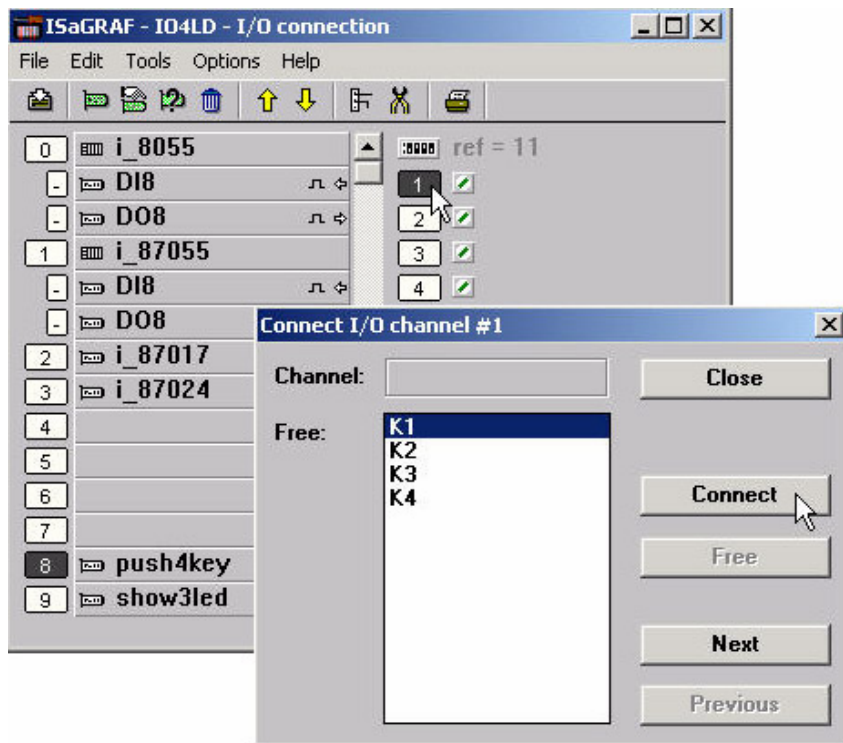
Declare Variables associated with the channels of connected I/O modules.  
You may refer to section 2.1.1.3  
Example:



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

You may refer to section 3.1.2

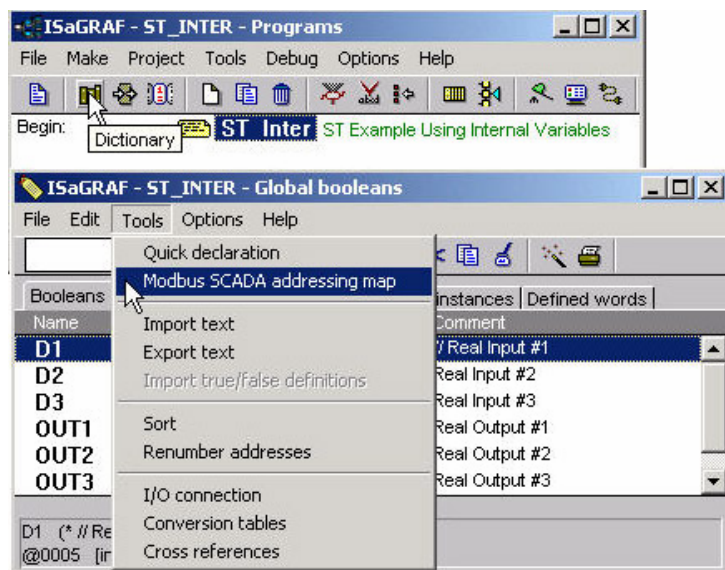
Example:



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.

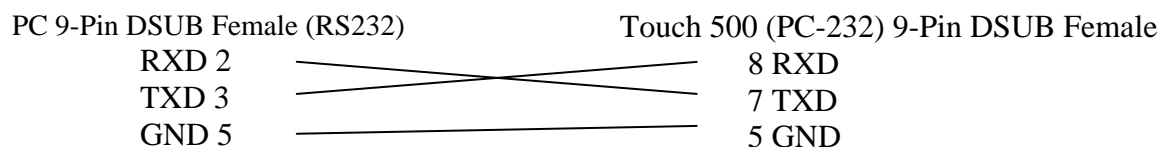
| Item              | Protocols                             |
|-------------------|---------------------------------------|
| Touch-506L        | Modbus RTU RS-232 , Modbus RTU RS-485 |
| <b>Touch-506T</b> | Modbus RTU RS-232 , Modbus RTU RS-485 |
| Touch-510T        | Modbus RTU RS-232 , Modbus RTU 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"

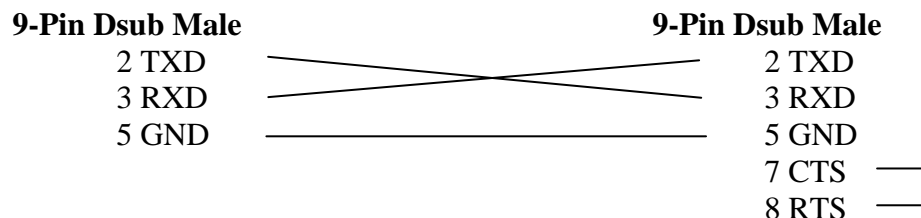
or run "setup.exe" at I-8000 CD-ROM:\napdos\others\touch\500series\setup\

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

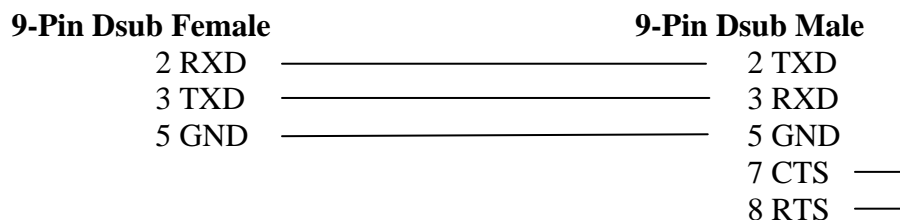


RS-232 Cable Pin assignment between controllers and Touch 500 series.

**I-8000 COM1 & I-7188/7186 COM1 (RS232)      Touch 506T/506L/510T (PLC 232)**

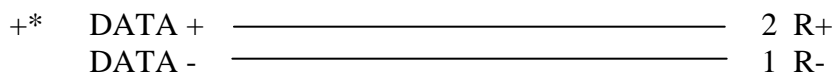


**Wincon COM2 (RS232)      Touch 506T/506L/510T (PLC 232)**

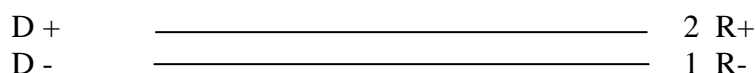


RS-485 Cable Pin assignment between controllers and Touch 500 series

**I-8417/8817 COM2 (RS485)      Touch 506T/506L/510T (PLC 485)**



**Wincon COM3 (RS485)      Touch 506T/506L/510T (PLC 485)**



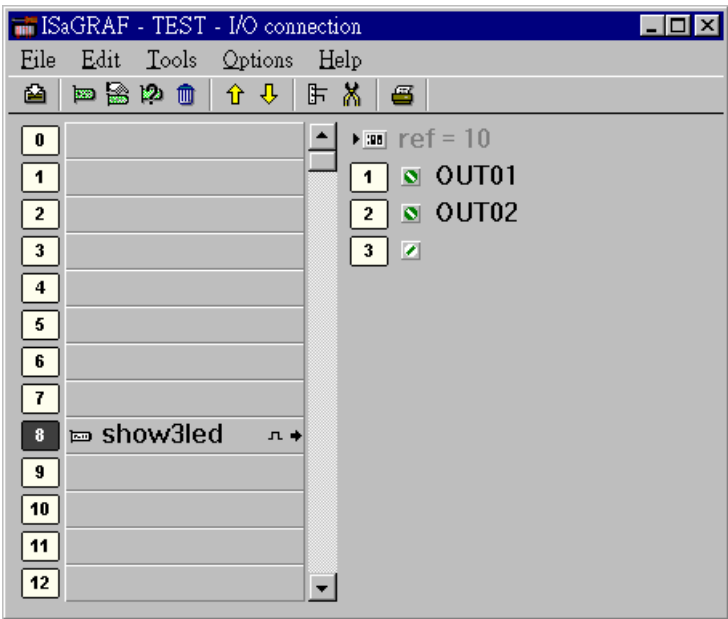
### 4.4.1: Program the I-8xx7, I-7188EG/XG & W-8xx7

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.

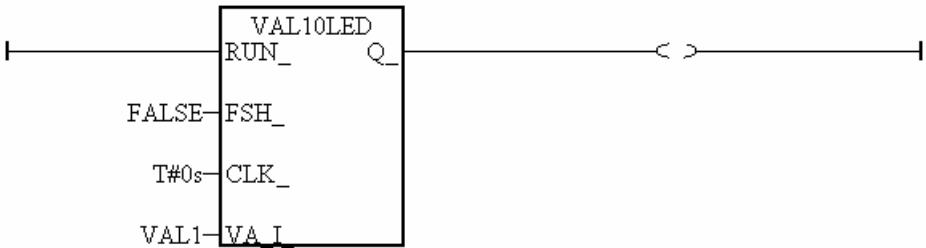
Variables used in this example.

| Name  | Type    | Attribute | Network address | Others |
|-------|---------|-----------|-----------------|--------|
| OUT01 | Boolean | Output    | <b>0001</b>     | -      |
| OUT02 | Boolean | Output    | <b>0002</b>     | -      |
| VAL1  | Integer | Internal  | 000A (10)       | -      |

IO connection:



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

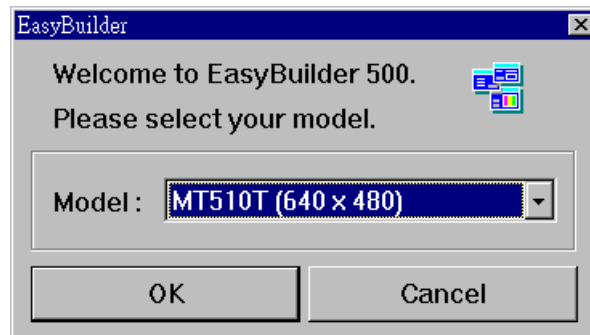


After you finish this project, compile and download it to the I-8xx7 controller.

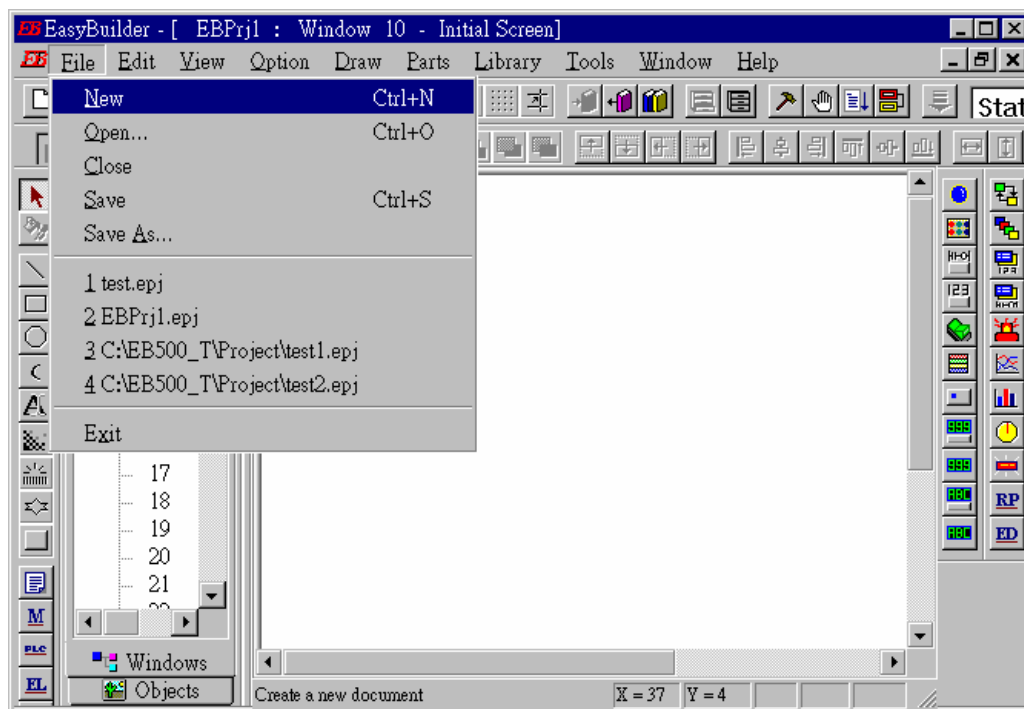
#### 4.4.2: Program the Touch 510T

The “EasyBuilder 500” software can be used to design 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.

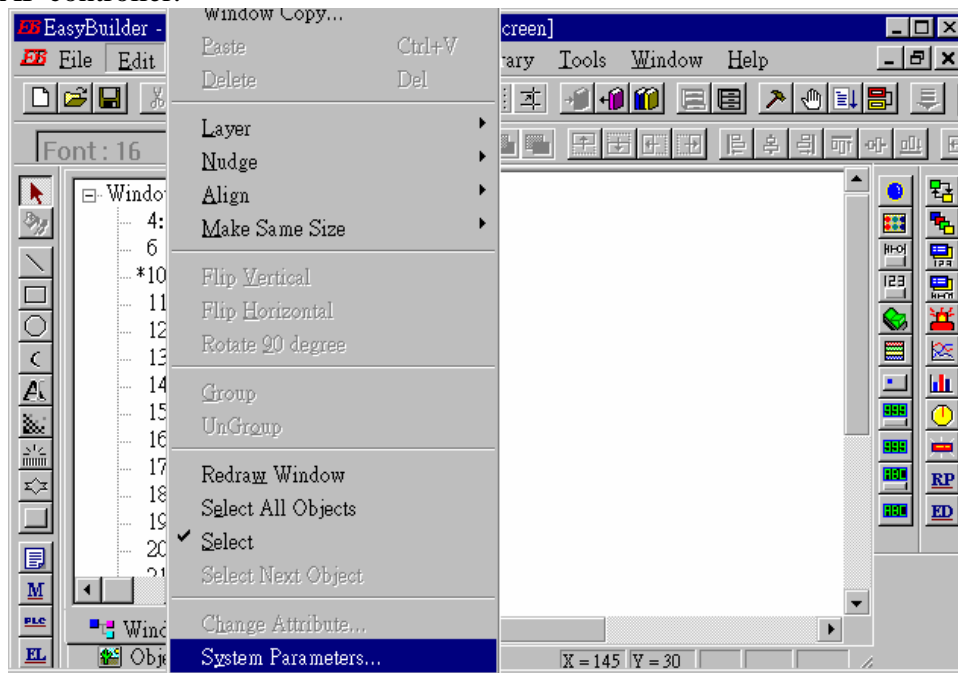
Click on the Windows "Start" button, then click on the "Program" button, then click on the "EasyBuilder" – “EasyBuilder 500” button. The following window will be displayed. Select the proper model for your application.



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.



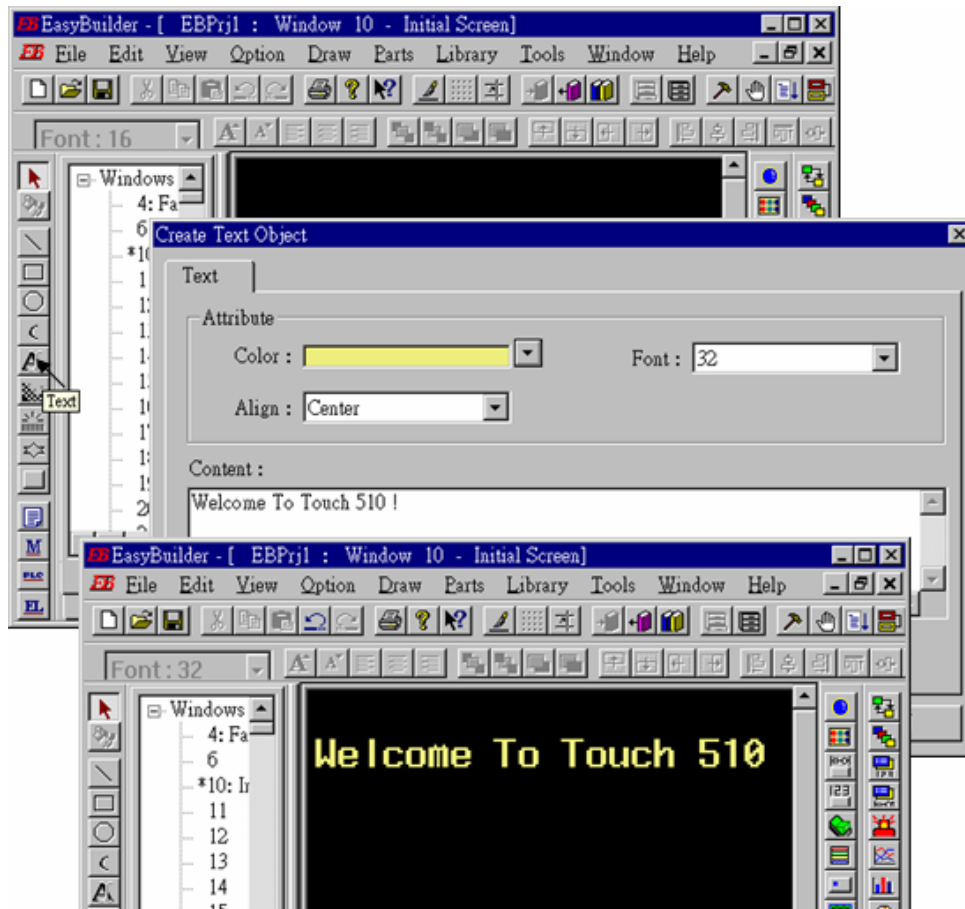
PLC type should be set to “**MODBUS RTU**”, Serial port set to “RS232”, Data bits set to “8 Bits”, Stop bits set to “1 Bit”, Baud rate set to “19200”, Parity set to “None”, PLC station No. set to be equal to the Net-ID of the I-8xx7 (set to 1 in this example).

|                                   |                             |
|-----------------------------------|-----------------------------|
| PLC type : MODBUS RTU             |                             |
| HMI model : MT510T (640 x 480)    |                             |
| Serial port I/F : RS232           | Baud rate : 19200           |
| Data bits : 8 Bits                | Parity : None               |
| Stop bits : 1 Bit                 |                             |
| HMI station No. : 0               | PLC station No. : 1         |
| Multiple HMI : Disable            | HMI-HMI link speed : 115200 |
| PLC time out constant (sec) : 3.0 | PLC block pack : 0          |

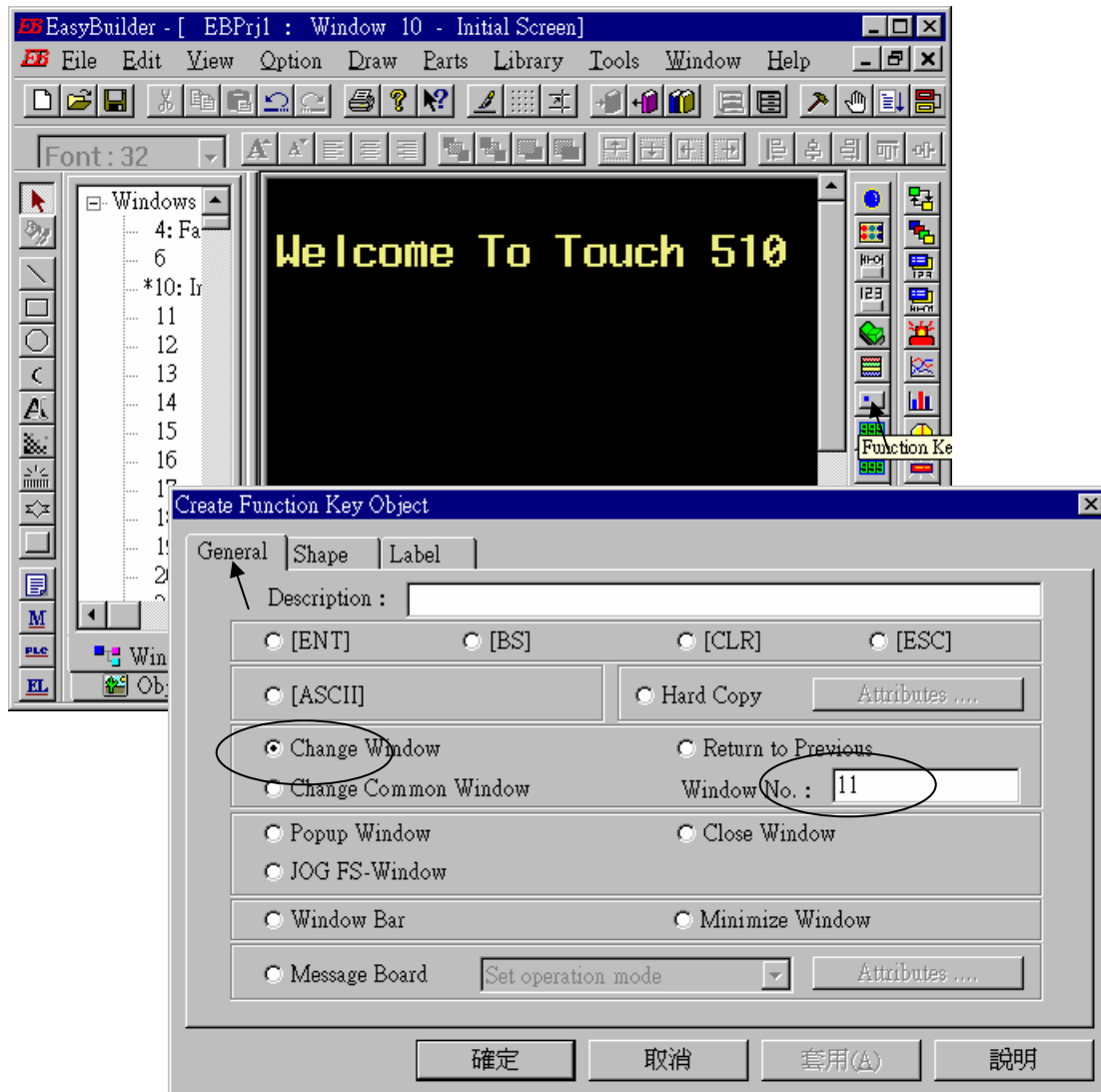
Note:

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.

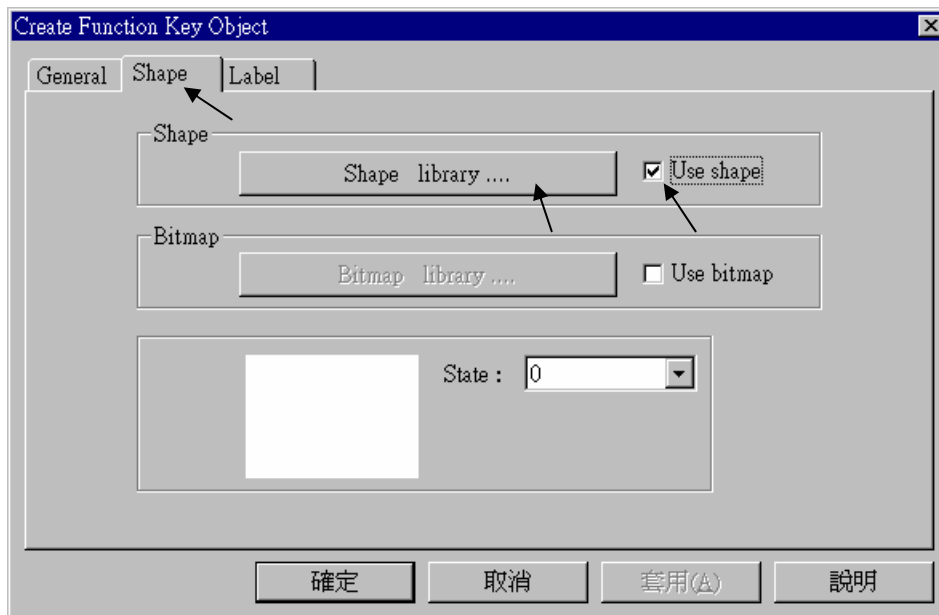
Click on “Text” to add a text. Select the preferred “Color”, “Font”, “Align” for the text and then enter the “Content”. And then place it to the proper position.



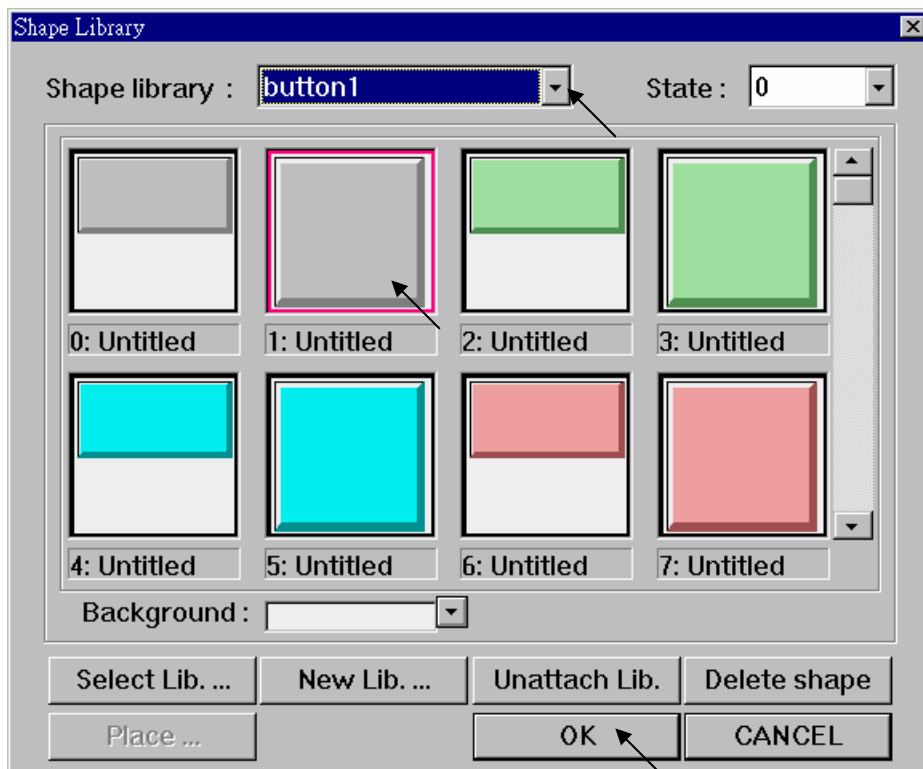
Click on “Function Key” to add a change-window button. Click on “General”, then select “Change Window” and set “Window No.” to 11.



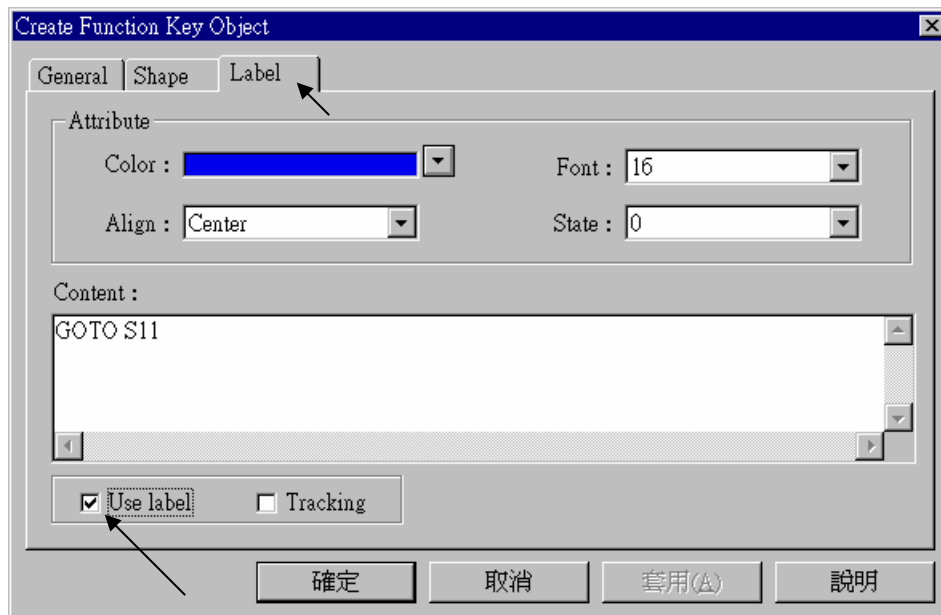
Click on “Shape”, then select “Use shape” and then click on “Shape library ...”



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



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

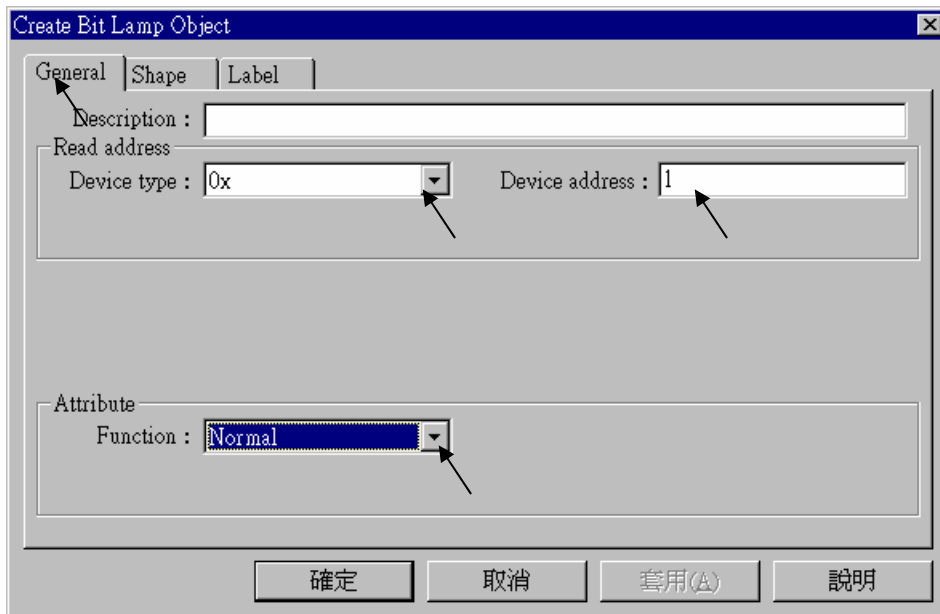


Click on “Bit Lamp”

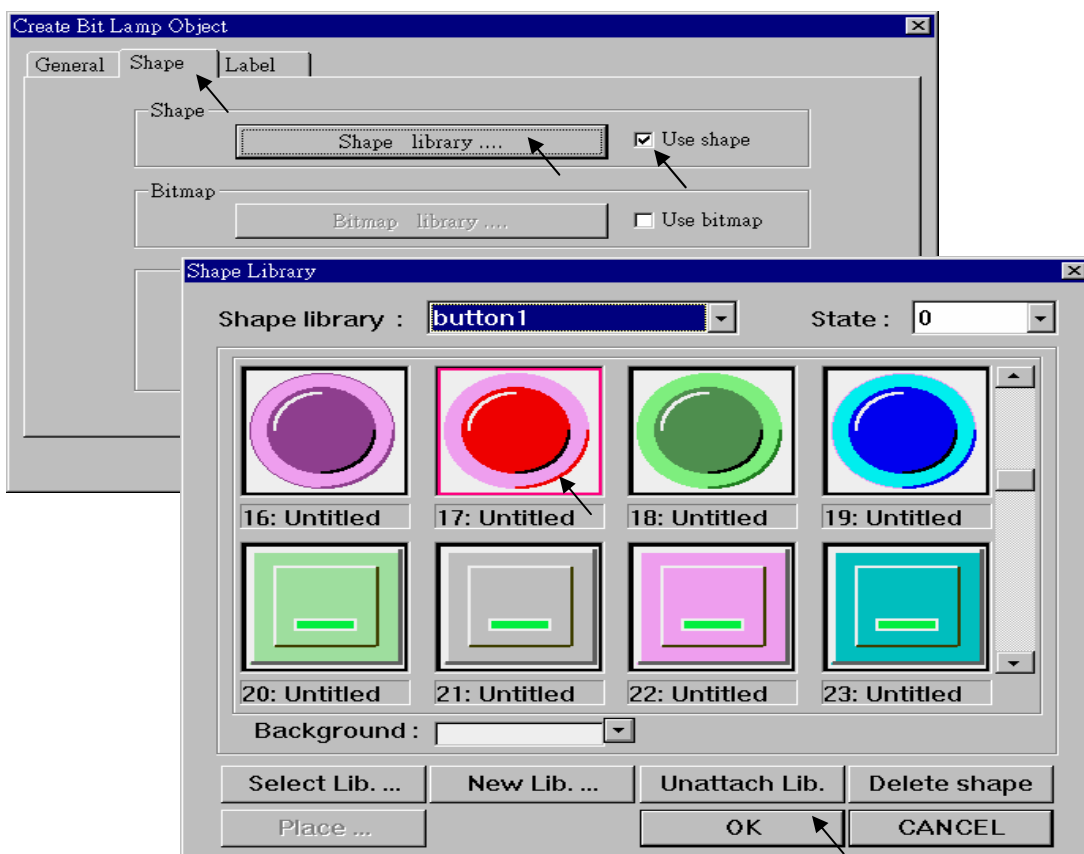




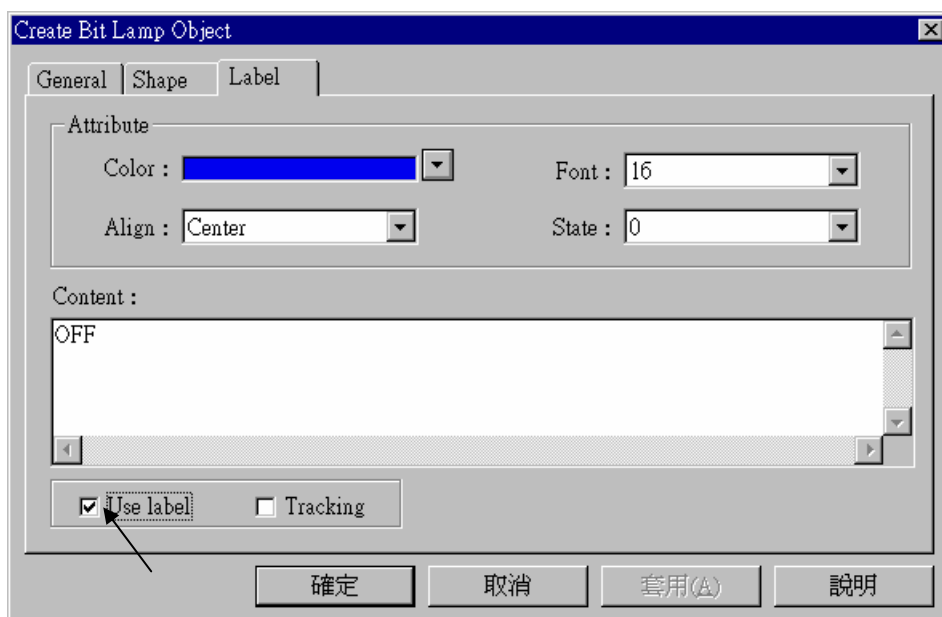
Click on “General”, then select “Device type” to “0x” (**0x is for boolean variables**), then set “Device address” to 1 (this value is associated with the network address value of the variable in the I-8xx7). And then set “Function” to “Normal”.



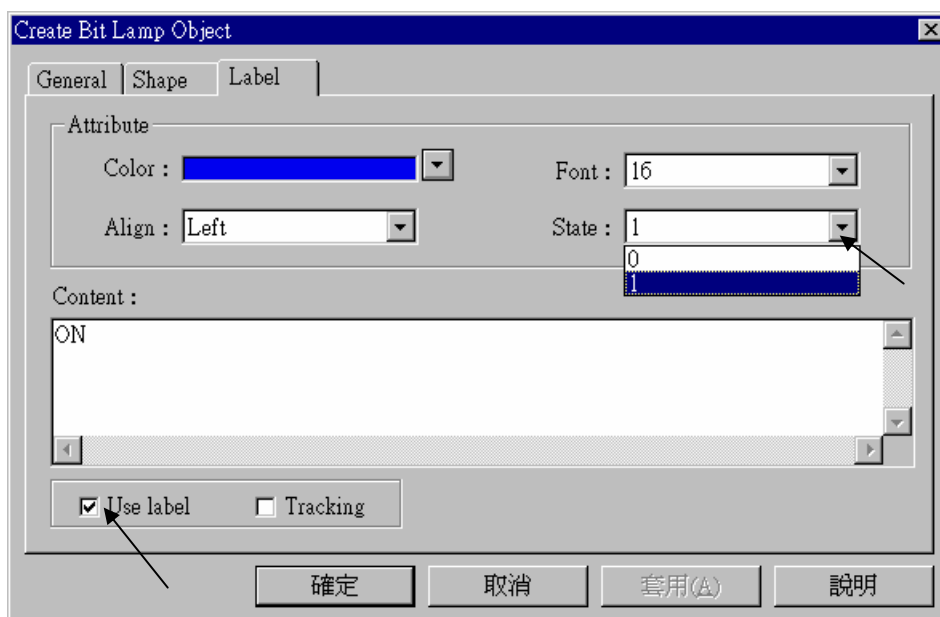
By the same way as former, select preferred “Shap library”.



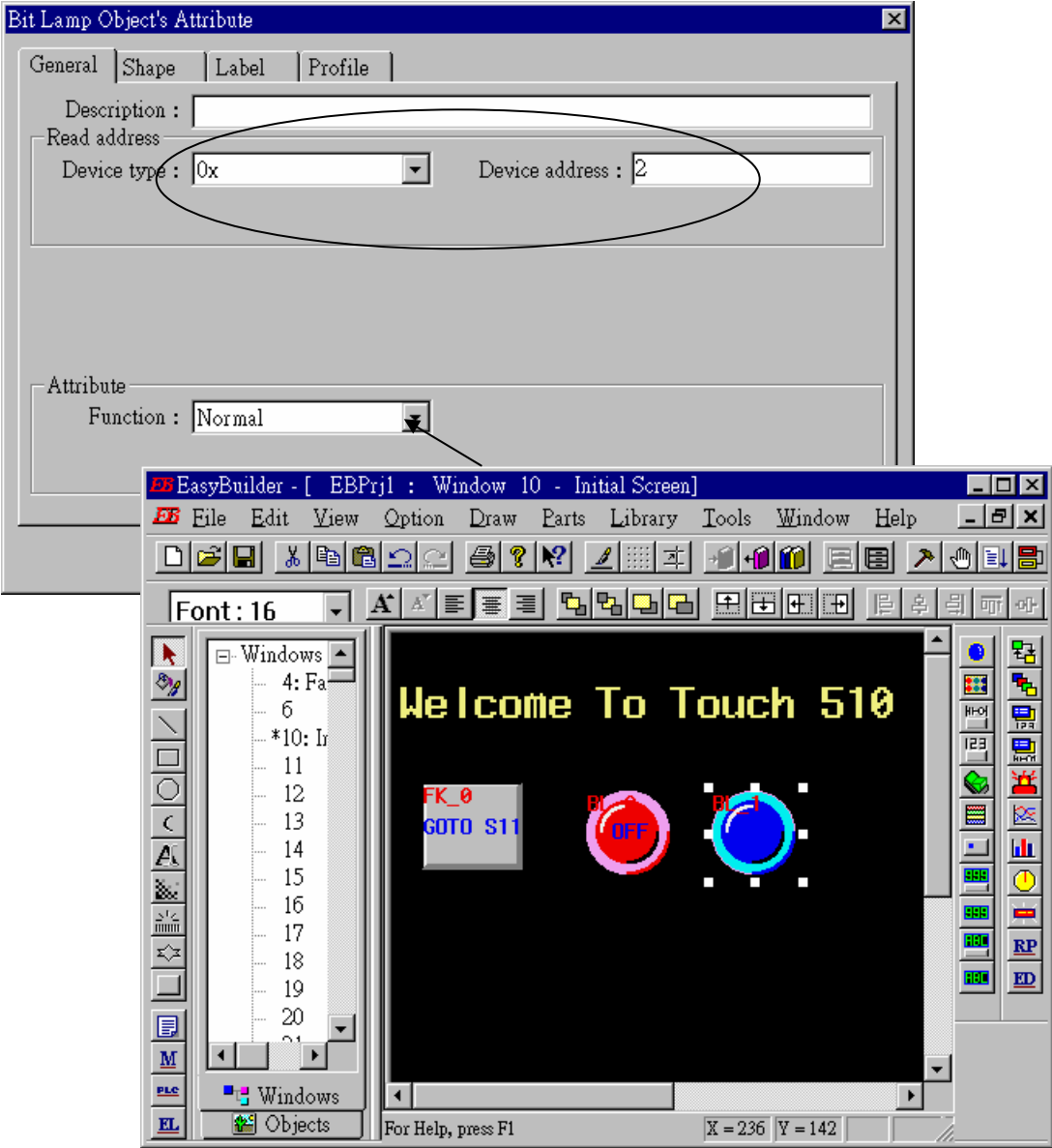
And then select “Label”, given a “OFF” to “Content” for “State : 0”. **Make sure “Use label” is choosed.**



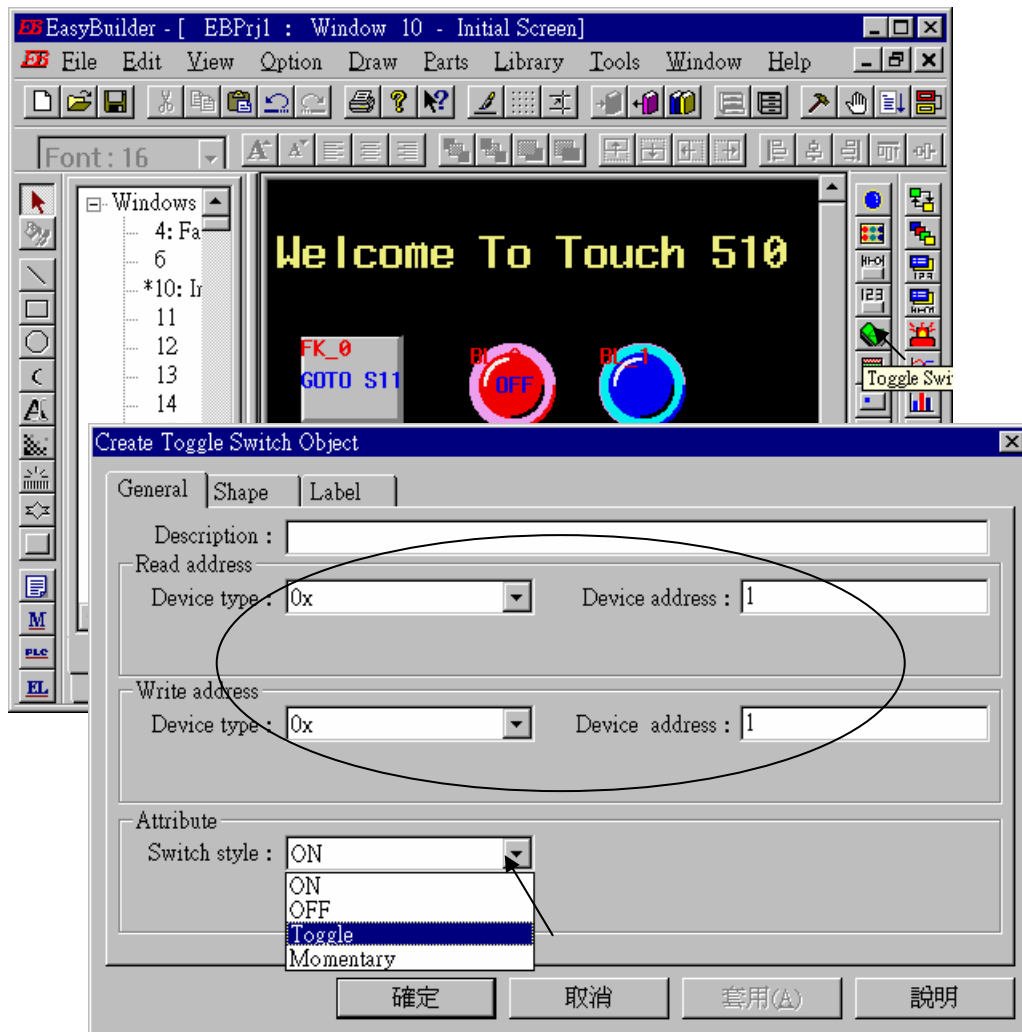
And then change “State” to 1, and given a “ON” to “Content”. **Make sure “Use label” is choosed.**



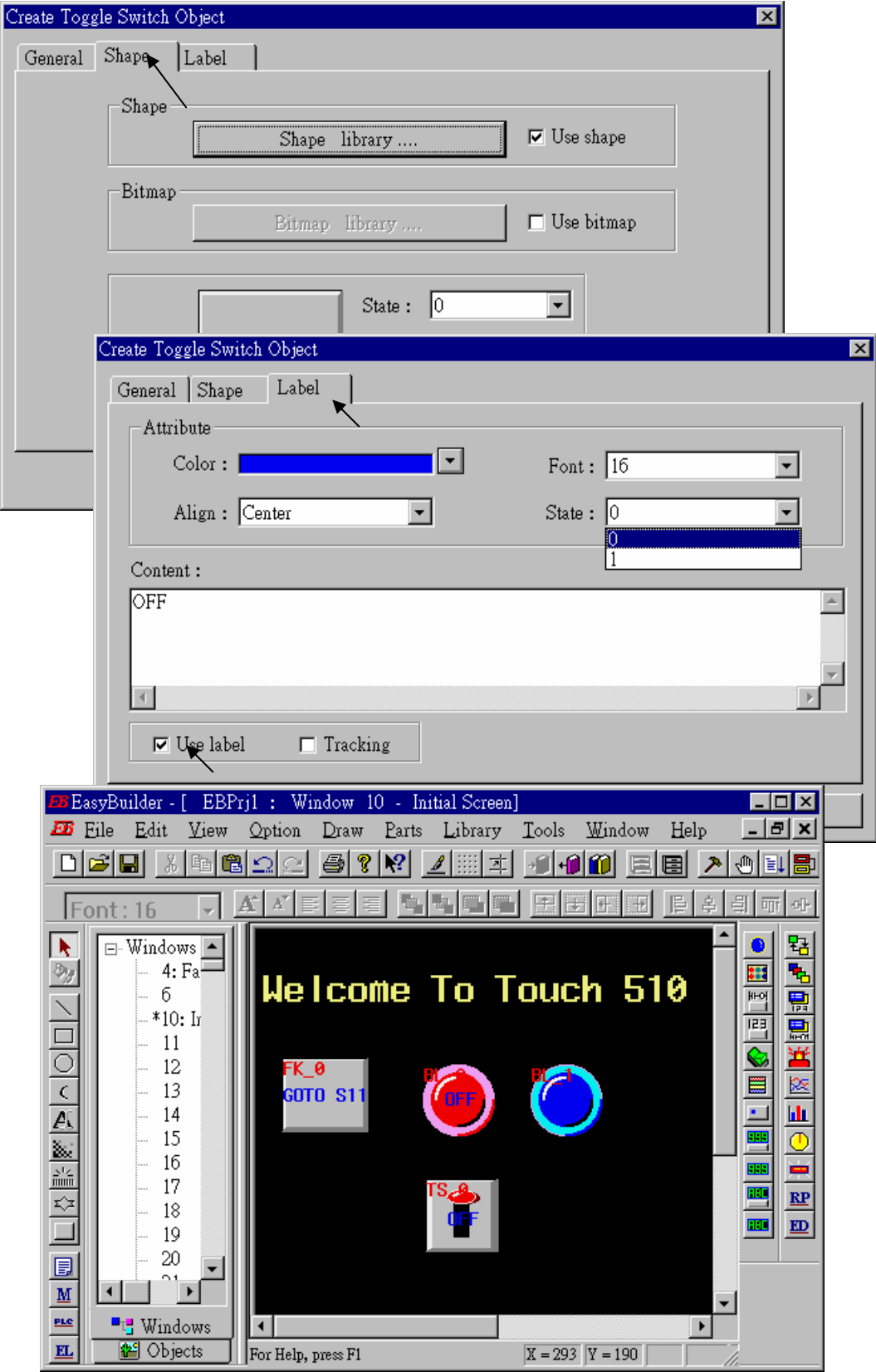
By the same way as former, create one another Bit Lamp with a “Device address” = 2.



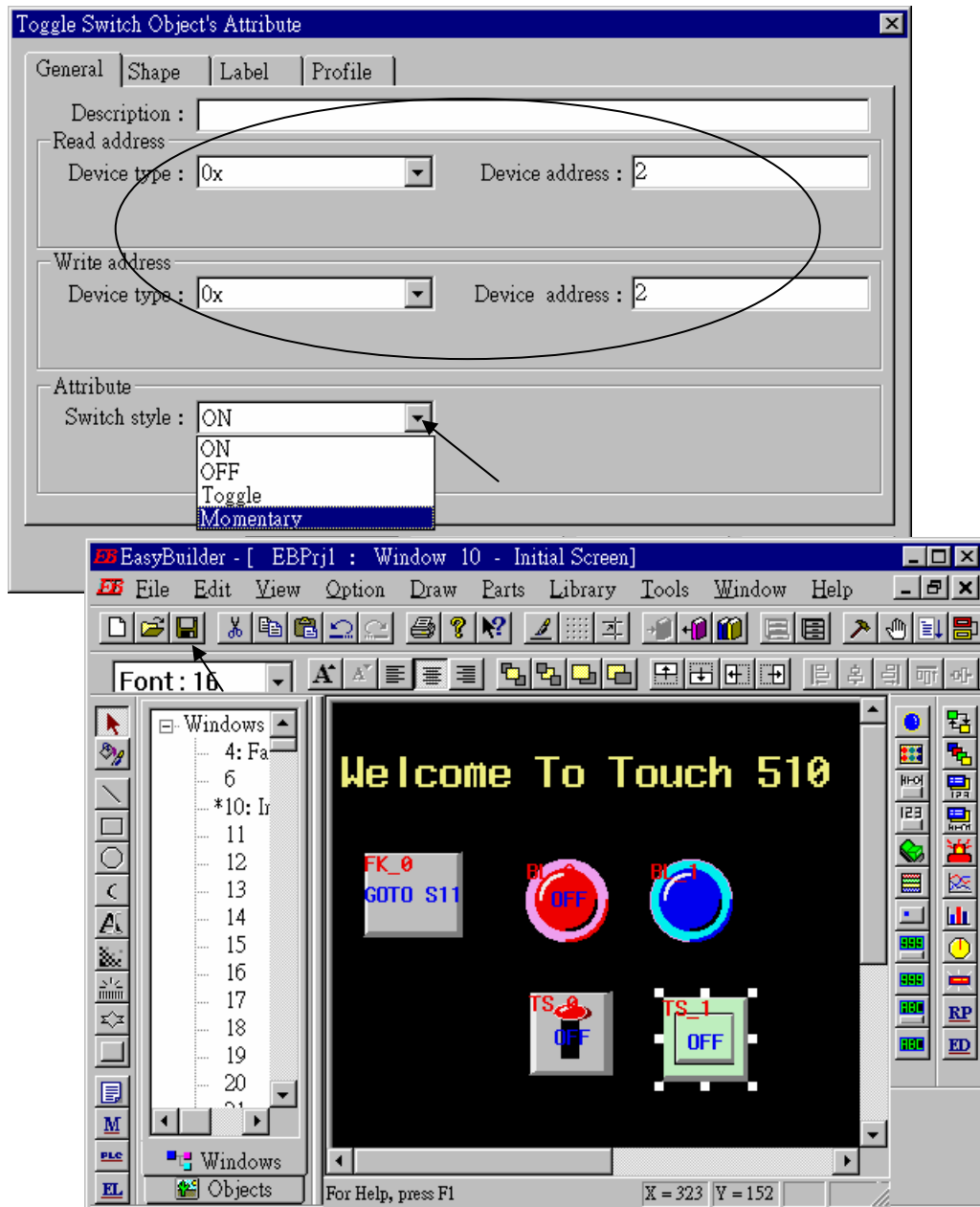
Click on “Toggle Switch”, then set all “Device Type” to “0x”, all “Device address” to 1 and select “Switch Type ” to “Toggle”.



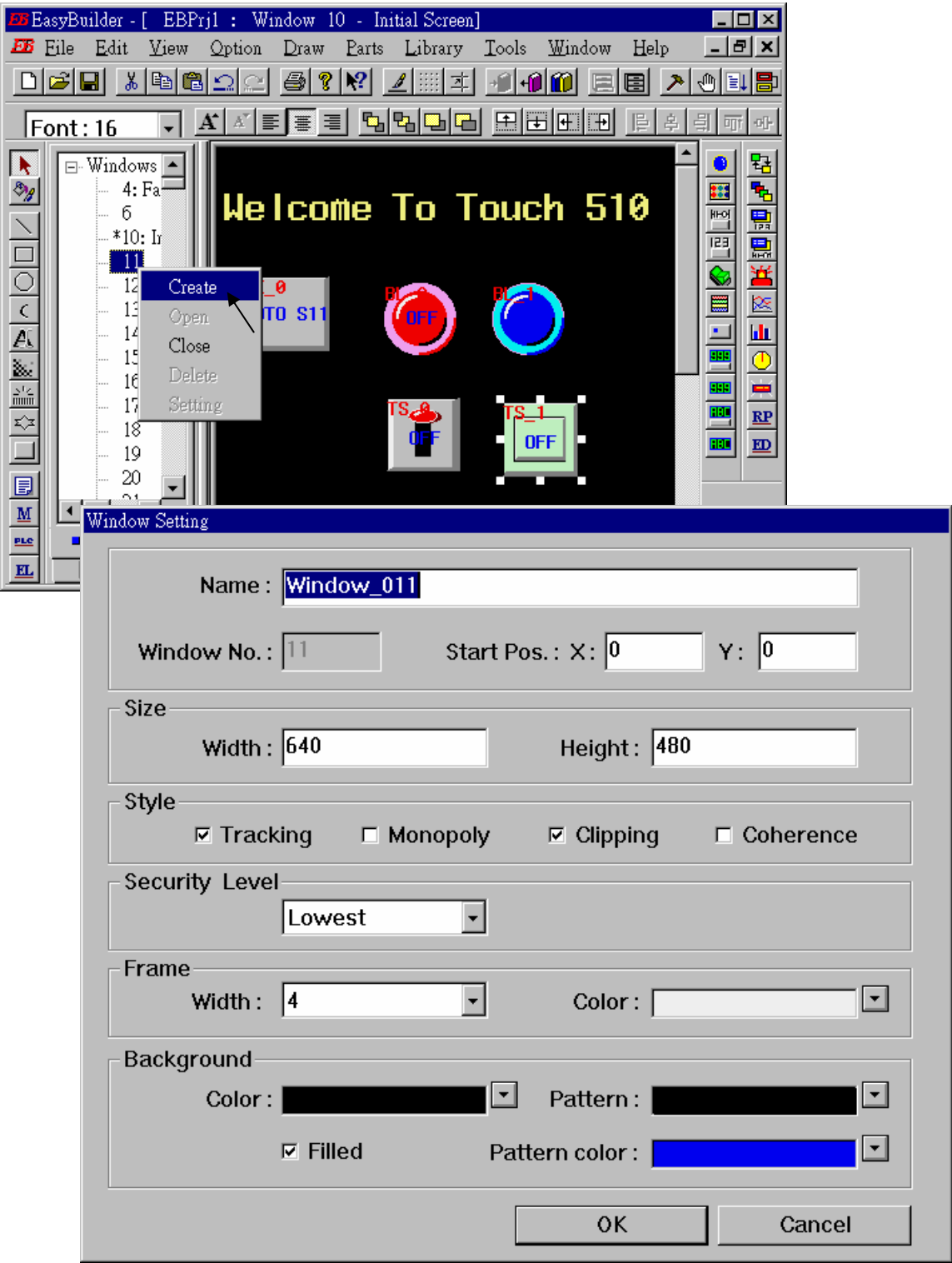
By the same way as former to choose a preferred “shape” and “label”.



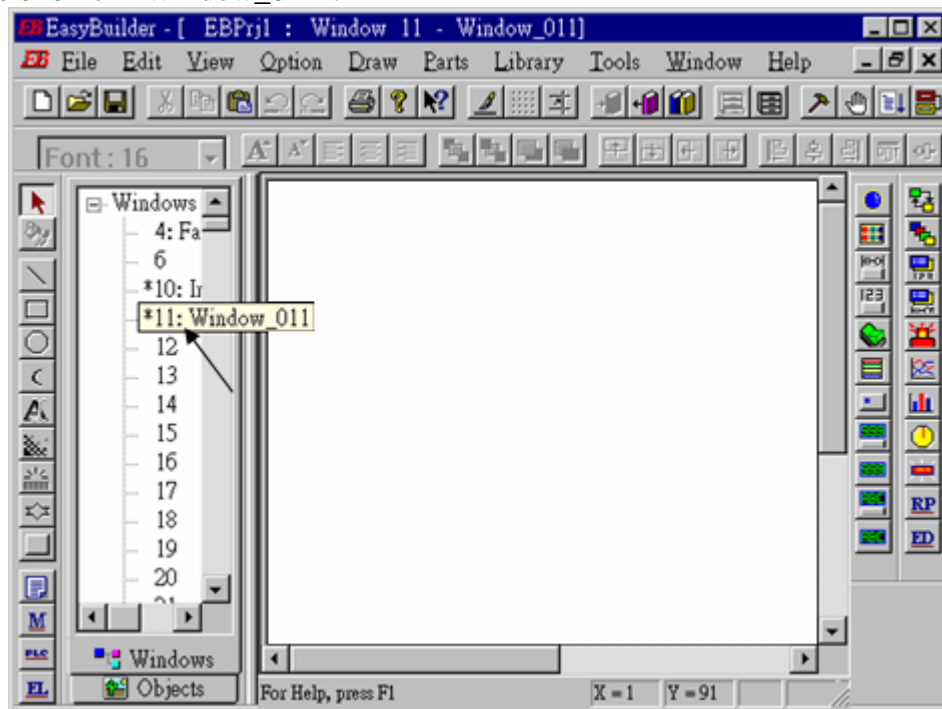
By the same way as former, create one another “Toggle Switch” however set all “Device address” to 2 and “Switch style” to “Momentary”. Click on “save” to save the project.



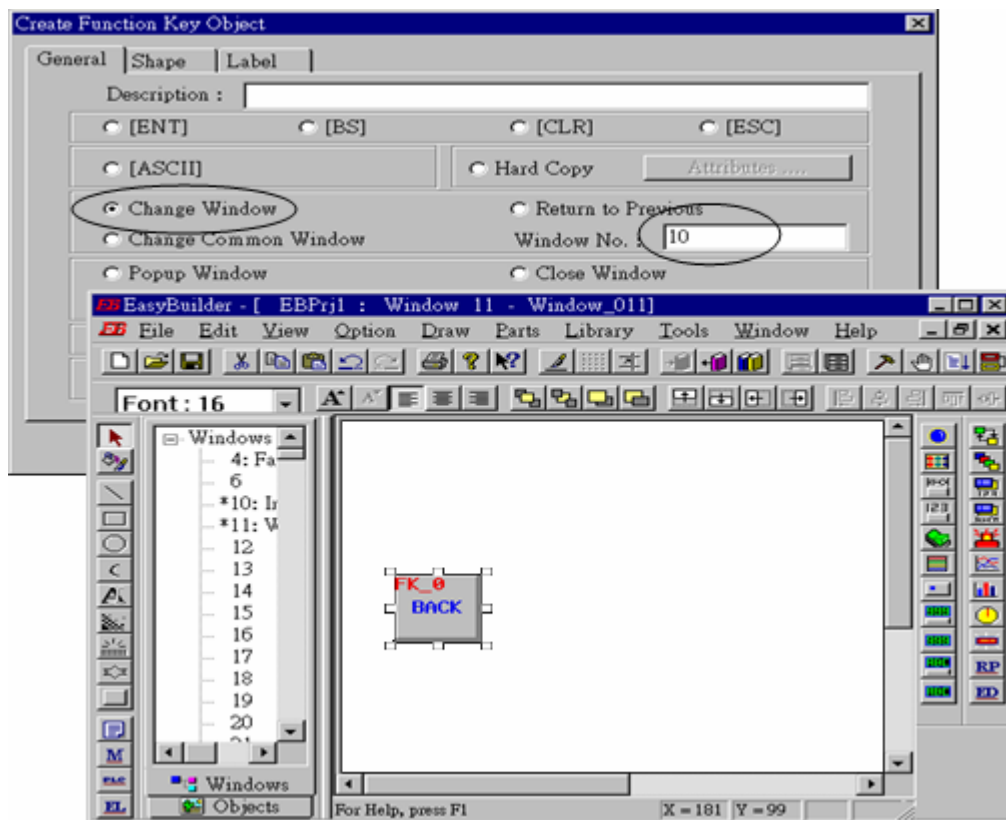
We are going to design another window. Click on “Windows” – “11”, then click and hold on the right button of the mouse and drag to “Create”.



Double click on “Window\_011”.

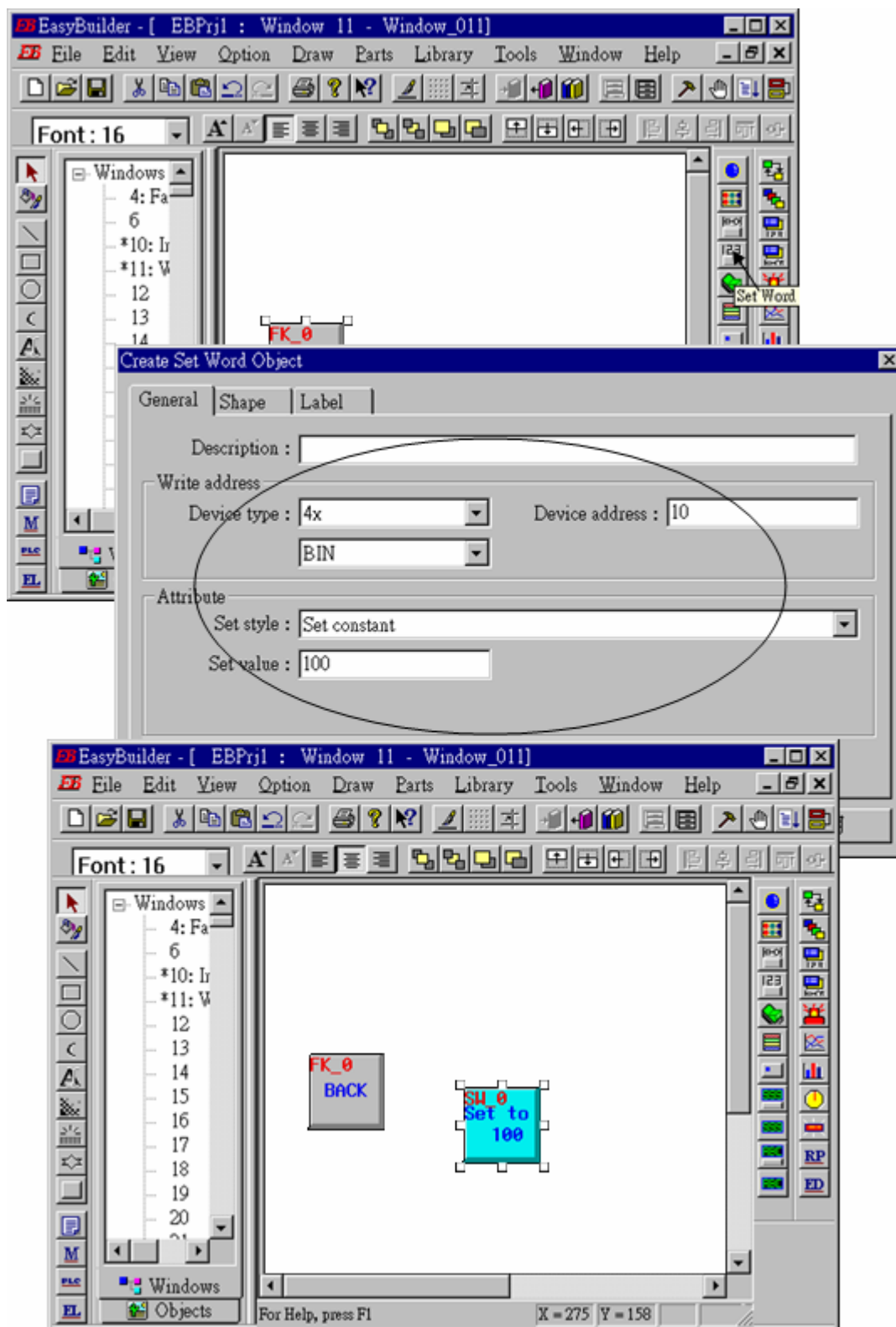


Create a change-window “Function Key” as former method to change to “Window No.” = 10, and Labeled as “BACK”.

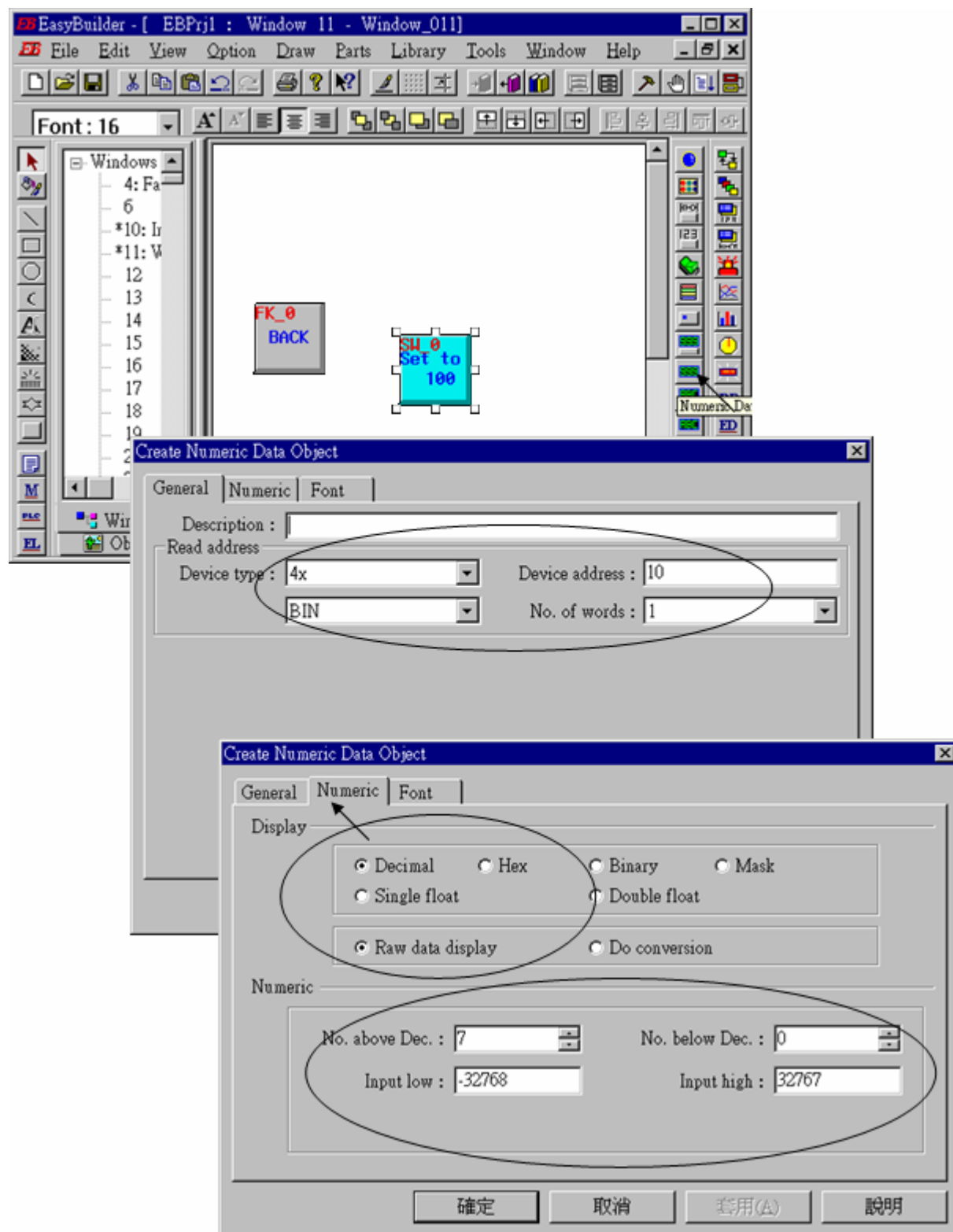




Click on “Set Word”, then set “Device Type” as “4x” (4x is for short integer, 4L is for long integer), set “Device address” to 10, “BIN”, and “Set style” to “Set Constant”, and “Set value” = 100. And then select the preferred “shape”, and set “label” to “Set to 100”.

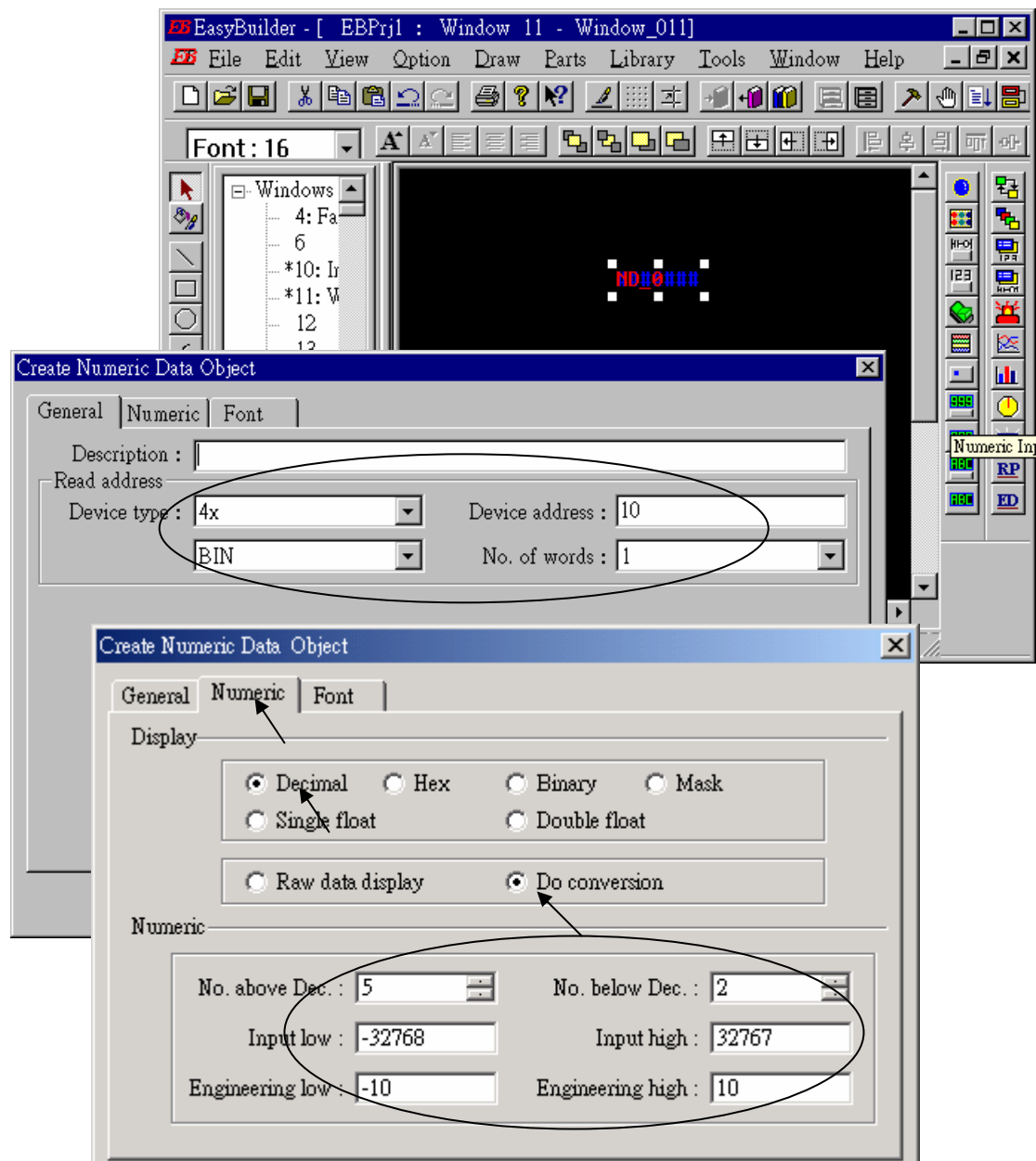


Click on “Numerical Data”, set “Device Type” to “4x” (**4x is for short integer, 4L is for long integer**), “Device address” to 10, “BIN”, “Number of words” to 1, “No. above Dec” to 7, “No. below Decimal” to 0, “Input low” to -32768, “Input high” to +32767. And then select the preferred shape.

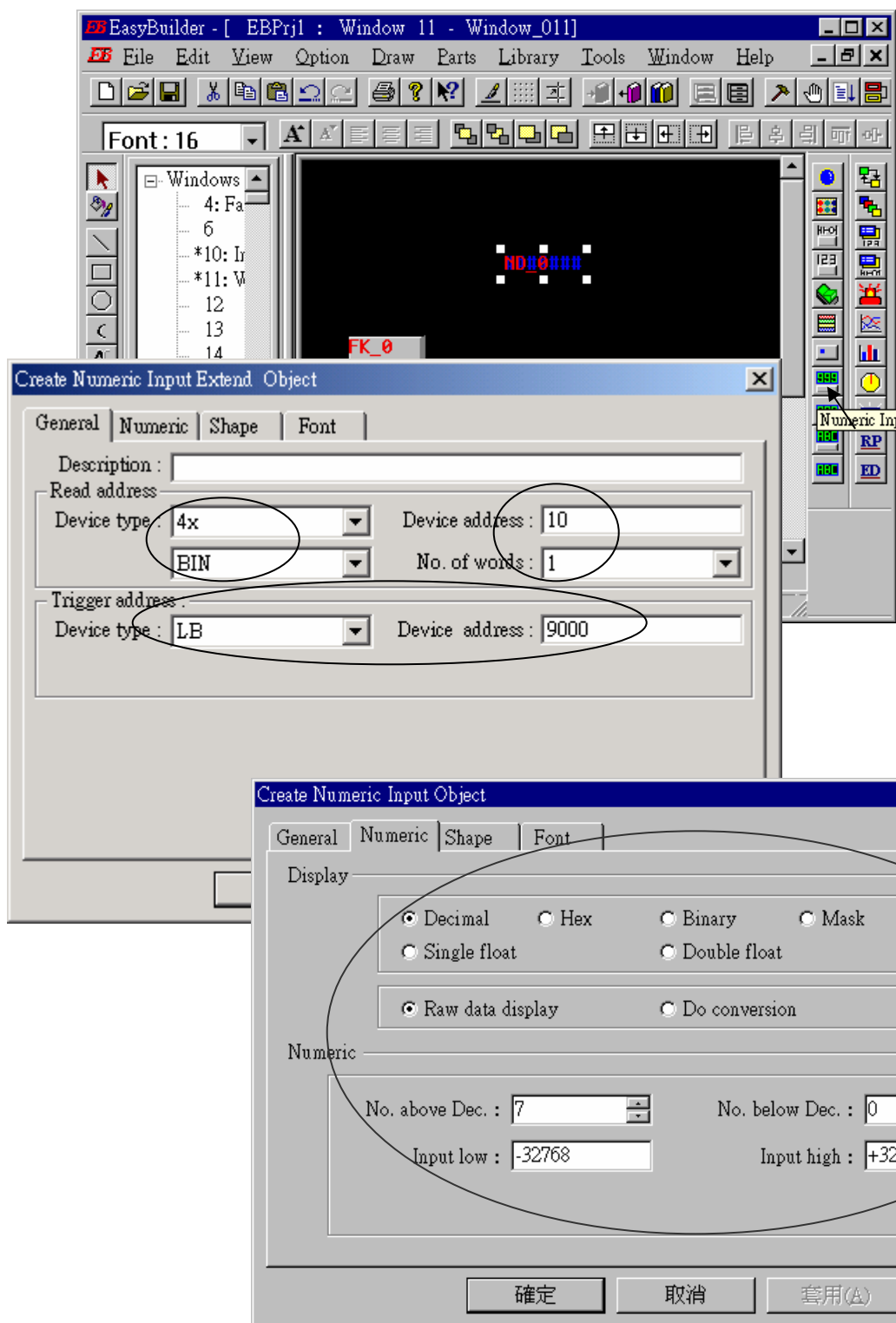


Now we are going to add one another “Numerical Data” with **conversion**.

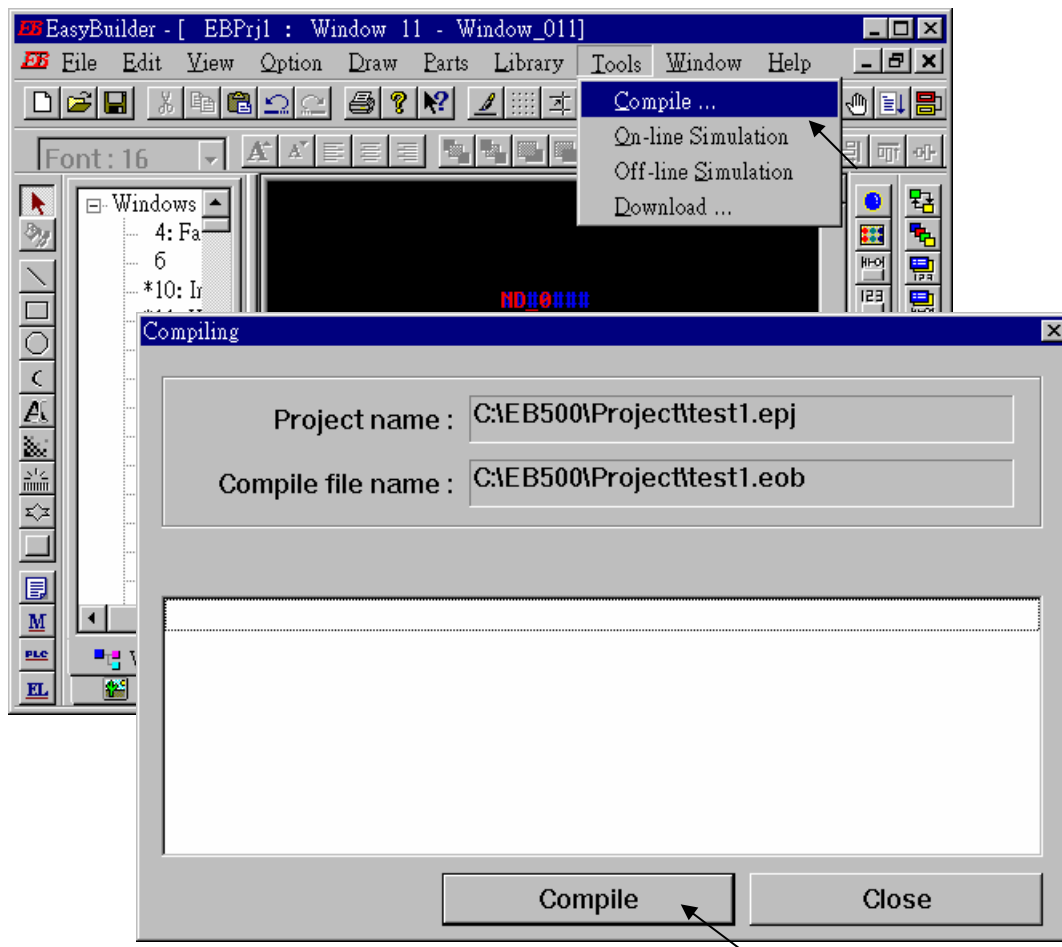
Click on “Numerical Data”, set “Device Type” to “4x”, “Device address” to 10, “BIN”, “Number of words” to 1, “No. above Dec” to 5, “No. below Decimal” to 0, “Input low” to –32768, “Input high” to +32767, check “Do conversion”, set “engineering low” to –10, “engineering high” to +10 (**Convert [-32768,+32767]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

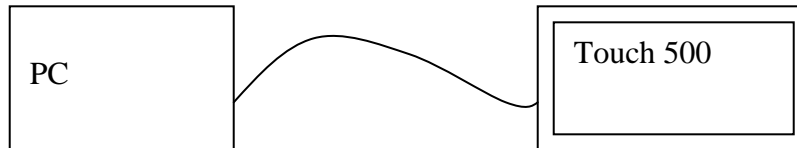


Click “Tools” – “Compile ...” to compile this project.

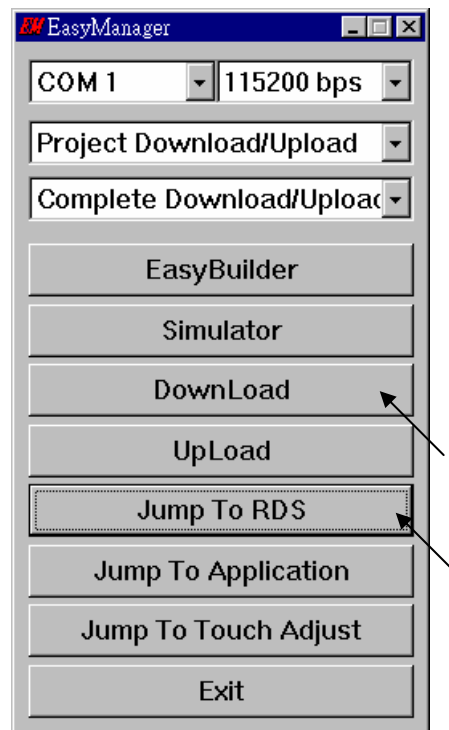


To download the project to the Touch 510, click on the Windows "Start" button, then click on the "Program" button, then click on the "EasyBuilder" – "EasyManager" button. The following window will be displayed. Choose the correct COM No. on your PC (Normally is COM1), "115200 bps".

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

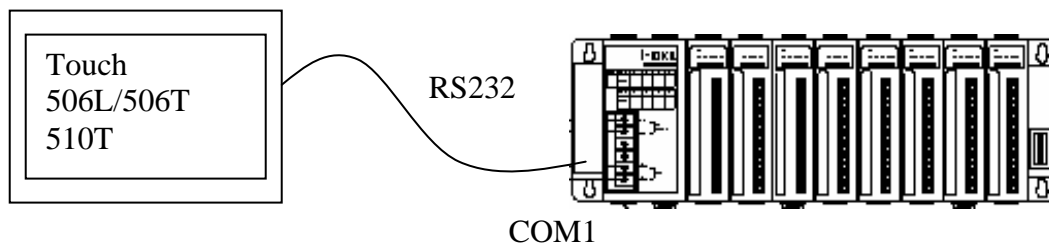


Click on "Jump To RDS" first, if OK., you can see the screen of the Touch 500 will change and wait for project download. Click on "Download" to start to download the MMI picture to the Touch 500.



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.

---

## Chapter 5. Modbus Protocol

---

The Modbus protocol is a powerful and flexible communications protocol that allows numerous software programs and hardware devices to communicate with each other. Any I-8xx7, I-7188EG/XG, uPAC-7186EG, iPAC-8447/8847 & W-8xx7 variable that will be used to communicate through the Modbus protocol **MUST** have a unique network address before it can communicate through a Modbus link (please refer to section 4.1).

### 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.

### 5.1: Modbus Protocol Format: RTU Serial

The Modbus "RTU Serial" format is supported by the I-8417 and I-8817 controller systems through both COM1 or COM2 communications ports, and the I-8437, I-8837, I-7188EG, I-7188XG, uPAC-7186EG and iPAC-8447/8847 controller systems through the COM1 communications port, and the Wincon-8x37 & Wincon-8x47 controller systems through the COM2 (or COM3) communications port.

PC software programs and HMI hardware devices can access data from the variables in the ISaGRAF controller system **ONLY** after that variable is assigned a unique network address (please refer to Chapter 4). For more information regarding connecting a PC to an I-8xx7 controller system, please refer to Section 1.3.3 through 1.3.5 for details on how to properly connect these devices.

It is **CRITICAL** that you must program the Modbus format **EXACTLY** as described to make a proper connection between the Modbus device and the ISaGRAF controller system. The I-8xx7, I-7188EG/XG, uPAC-7186EG and iPAC-8447/8847 & W-8xx7 controllers support the following Modbus functions.

| Modbus function | Action                                |
|-----------------|---------------------------------------|
| 1               | Read N bits (booleans)                |
| 2               | Read N bits (booleans)                |
| 3               | Read N words (signed short integers)  |
| 4               | Read N words (signed short integers)  |
| 5               | Write 1 bit (boolean)                 |
| 6               | Write 1 word (signed short integer)   |
| 15              | Write N bits (booleans)               |
| 16              | Write N words (signed short integers) |



To read boolean variables, both of function 1 or 3 may be used. If using function 1, values are stored in a bit field while using function 3, variable TRUE means 0xFFFF.

To write boolean variables, both of function 5, 15 could be used. If using function 5, writing bit 0 of byte-vH to 1 will set the Boolean variable to TRUE. For ex, writing vH=1 or 3, or 255 will set Boolean variable to TRUE.

To read analog variables, function 3 should be used.

To write analog variables, both of function 6, 16 could be used.

To read long words (signed long integers, float), function 3 should be used. To write long words, function 16 should be used. Please refer to section 4.2 for the definition of network address of long words.

To assist you with the naming conventions used throughout the Modbus protocol-addressing chapter, the following table describes the notations used in this chapter.

|      |   |
|------|---|
| Slv  | Slave number (Net ID address of the I-8xx7) |
| Nbw  | Number of words                             |
| Nbb  | Number of bytes                             |
| Nbi  | Number of bits                              |
| AddH | <b>Modbus address</b> , high byte , 0 ~ 0F  |
| AddL | <b>Modbus address</b> , low byte , 0 ~ FE   |
| VH   | Word value, high byte                       |
| VL   | Word Value, low byte                        |
| V    | Byte value                                  |
| CrcH | Checksum, high byte , CRC-16                |
| CrcL | Checksum, low byte , CRC-16                 |

## IMPORTANT NOTE

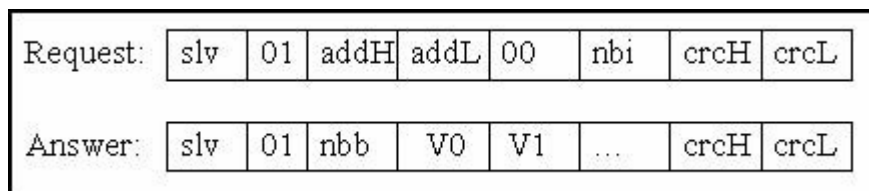
All of the values used in the request and answer frames are **hexadecimal** values.

Modbus address described in this chapter is equal to Network address of the variable minus one.

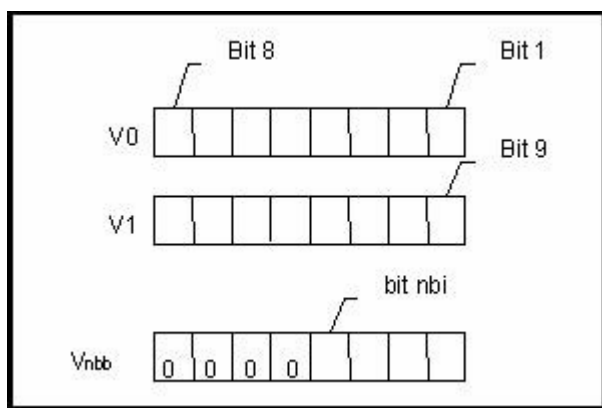
For ex., Modbus address 0 is associate with Network address 1. Modbus address FFE (4094) is associate with Network address FFF (4095).

Function 1: Read "N" Bits

Function 1 reads "n" number of bits (nbi) in Boolean starting from Modbus address addH/addL.



V0, V1 ... are the bit fields of number of bytes (nbb) using the following format.



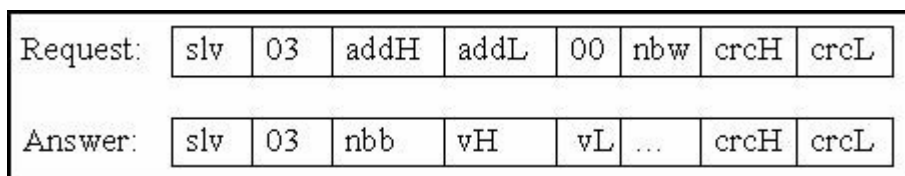
Bit 1 corresponds to the Boolean value of the variables with the Modbus address addH/addL. Bit nbi corresponds to the Boolean value of the variable with the Modbus address addH/addL + nbi – 1. If the value of the Boolean variable is "True", then the corresponding bit will be set to a "1". If the value is "False", the corresponding bit will be set to a "0".

#### Function 2: Read N Bits

Function 2 has the same exact same format as function 1.

#### Function 3: Read N Words

Function 3 reads the number of words (nbw), in signed 16-bit integer format, starting from the Modbus address addH/addL.



The number of bytes (nbb) is the total number of bytes from word value high byte (vH) to word value low byte (vL) inclusive.

#### IMPORTANT NOTE About Function 3

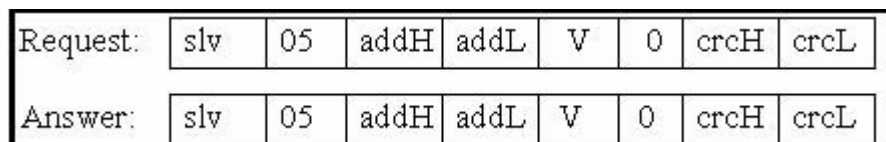
Integer values can be read by function 3. A word in the modbus protocol is a 16-bit value (signed short integer), and an integer variable is a 32-bit value, so only the lower 16 bits of the integer variable are returned. If users would like to read a 32-bit integer (signed long integer) of I-8xx7 controller, the proper network address of the variable should be set as described in section 4.2.

#### Function 4: Read N Words

Function 4 has the same exact format as function 3.

#### Function 5: Write 1 Bit

Function 5 writes one (1) bit to the Boolean variable with the Modbus address addH/addL.



Writing a 0xFF value to the byte value (V) will set the Boolean variable to "True". Writing a zero to the byte value (V) is set the Boolean variable to "False".

Function 6: Write 1 Word

Function 6 writes one (1) word (16 bits) to the integer variable with the Modbus address addH/addL.

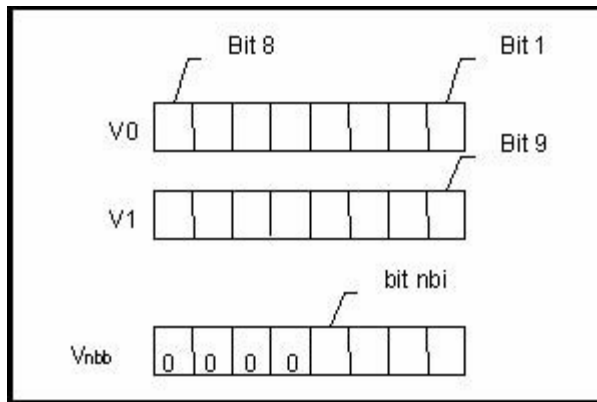
|          |     |    |      |      |    |    |      |      |
|----------|-----|----|------|------|----|----|------|------|
| Request: | slv | 06 | addH | addL | vH | vL | crcH | crcL |
| Answer:  | slv | 06 | addH | addL | vH | vL | crcH | crcL |

### Function 15: Write N Bits

Function 15 writes a number of bits (nbi) to the Boolean variables starting from the Modbus address addH/addL to addH/addL + nbi – 1. The total number of bytes (nbb) is the total amount of bytes occupied by nbi bits, that means  $nbb = (nbi+7)/8$ . For ex. nbi=1~8, nbb=1; nbi=9~16, nbb=2.



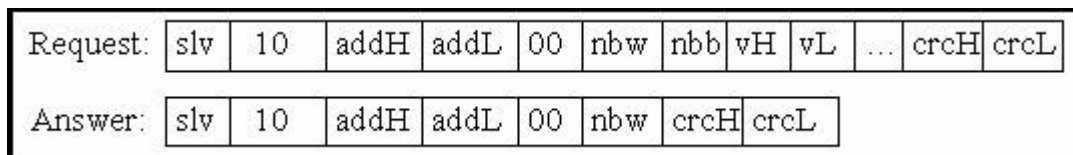
V0, V1 ... are the bit fields of number of bytes (nbb) using the following format.



Bit 1 corresponds to the Boolean value of the variables with the Modbus address addH/addL. Bit nbi corresponds to the Boolean value of the variable with the Modbus address addH/addL + nbi – 1. Writing a 1 to a bit will set the value of the corresponding Boolean variable to "True", and writing a 0 to a bit will set the corresponding Boolean variable to "False".

### Function 16: Write N Words

Function 16 writes a number of words (nbw) to the integer variables starting from the Modbus address addH/AddL to addH/addL + nbw – 1. The number of bytes (nbb) is the total amount of bytes occupied by number of words (nbw), that is  $nbb = 2 * nbw$ .



## Examples Of Modbus Function Formats

**Function 1:** Read 15 bits starting from **Modbus address 0x1020**. The NET ID address is 1.

|          |    |    |    |    |    |    |    |    |
|----------|----|----|----|----|----|----|----|----|
| Request: | 01 | 01 | 10 | 20 | 00 | 0F | 79 | 04 |
| Answer:  | 01 | 01 | 02 | 00 | 12 | 39 | F1 |    |

In this example function 1 returns 2 bytes, the value is 0x0012. This means variables with a **network address** of 0x102A and 0x102D are "True" (**Modbus address** is 0x1029 and 0x102C), the rest of the variables are set to "False".

**Function 5:** Write 1 bit to the Boolean variable with the **Modbus address 0x0006**. The NET ID address is 1. The value to write to is 0xFF.

|          |    |    |    |    |    |    |    |    |
|----------|----|----|----|----|----|----|----|----|
| Request: | 01 | 05 | 00 | 06 | FF | 00 | 6C | 3B |
| Answer:  | 01 | 05 | 00 | 06 | FF | 00 | 6C | 3B |

In this example of function 5 the Boolean variable is set to "True".

**Function 16:** Write 2 words (4 bytes) to the integer variables with the **Modbus address** starting from 0x2100. The first word value to write to is 0x1234. The second word value to write to is 0x5678. The NET ID address is 1.

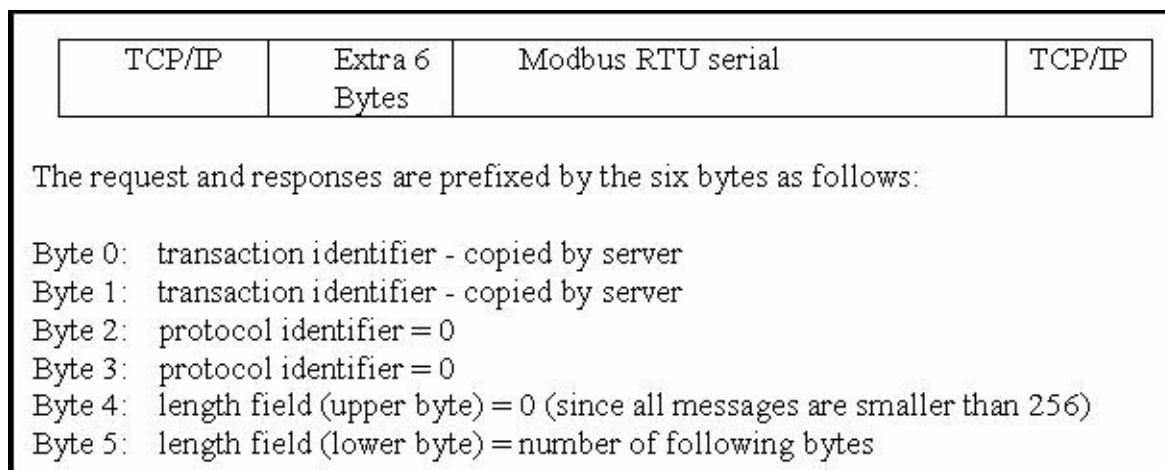
|          |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Request: | 01 | 10 | 21 | 00 | 00 | 02 | 04 | 12 | 34 | 56 | 78 | 1C | CA |
| Answer:  | 01 | 10 | 21 | 00 | 00 | 02 | 4B | F4 |    |    |    |    |    |

## 5.2: Modbus Protocol Format: TCP/IP

The I-8437 / I-8837 , iPAC-8447/8847, I-7188EG, uPAC-7186EG, W-8x37 and W-8x47 controller systems support the Modbus "TCP/IP" communications protocol.

ALL requests are sent via TCP on port number **502**.

The Modbus TCP/IP protocol adds 6 extra bytes before the Modbus RTU serial protocol, and these 6 extra bytes and the Modbus RTU serial protocol are all packed inside the TCP/IP protocol.



The rest of the Modbus TCP/IP protocol is the same as the Modbus RTU Serial protocol after byte No. of 6 except that the CRC-16 is not need for the Modbus TCP/IP protocol.

### Example TCP/IP Transactions

The first example of a TCP/IP transaction is reading one (1) word at Modbus address 4 from slave number 9 returning a value of 8; the transaction would be as follows:

|           |    |    |    |    |    |    |    |    |    |    |    |    |
|-----------|----|----|----|----|----|----|----|----|----|----|----|----|
| Request:  | 01 | 02 | 00 | 00 | 00 | 06 | 09 | 03 | 00 | 04 | 00 | 01 |
| Response: | 01 | 02 | 00 | 00 | 00 | 05 | 09 | 03 | 02 | 00 | 08 |    |

The second example of a TCP/IP transaction is reading 8 bits starting from Modbus address 2 from slave number 7, returning a value of 0x49 (bit field: 01001001) would be as follows:

|           |    |    |    |    |    |    |    |    |    |    |    |    |
|-----------|----|----|----|----|----|----|----|----|----|----|----|----|
| Request:  | 03 | 29 | 00 | 00 | 00 | 06 | 07 | 01 | 00 | 02 | 00 | 08 |
| Response: | 03 | 29 | 00 | 00 | 00 | 04 | 07 | 01 | 01 | 49 |    |    |

## 5.3: Algorithm For CRC-16 Check

---

The following C language algorithm is for Modbus RTU Serial ONLY!! This CRC (Cyclic Redundancy Check) program provides a checksum that can be used to validate information being passed through Modbus RTU Serial protocol.

This CRC-16 check program first calls "crc\_init()" one time at the beginning of the communication to initialize the checksum table. Then you can call "crc\_make()" to calculate a checksum whenever you want to.

```
#define POLY_CRC16 0xA001
static BYTE TABLE1[256];
static BYTE TABLE2[256];

void crc_init(void) /* set crc table */
{
    WORD mask,bit,crc,mem;
    for(mask=0;mask<0x100;mask++)
    {
        crc=mask;
        for(bit=0;bit<8;bit++)
        {
            mem=crc & 0x0001;
            crc/=2;
            if(mem!=0) crc ^= POLY_CRC16;
        }
        TABLE2[mask]=crc & 0xff;
        TABLE1[mask]=crc >> 8;
    }
}

void crc_make(WORD size, BYTE *buff, BYTE *hi, BYTE *lo) /* calculate crc */
{
    BYTE car,i;
    BYTE crc[2];
    crc[0]=0xff;
    crc[1]=0xff;
    for(i=0;i<size;i++)
    {
        car = buff[i];
        car ^= crc[0];
        crc[0]=crc[1] ^ TABLE2[car];
        crc[1]=TABLE1[car];
    }
    *hi=crc[0];
    *lo=crc[1];
}
```

---

## Chapter 6. Linking RS-485 I-7000 & I-87xx Modules

---

### Note:

1. The I-87017R and I-87017RC is better than I-87017 and I-87017C in industrial application.
2. The I-87018Z is better than I-87018R and I-87018 in industrial application. (I-87018Z has 10-channels. The precision is better than I-87018R and I-87018 . And each channel can configure to be different Input type and range. For example, using Ch.1 to 4 to measure 4 to 20 mA , using Ch.5 to 8 as Thermo-Couple K-Type, using Ch.9 to measure +/- 2.5 V, and using Ch.10 as Thermo-Couple R-Type.)
3. The I-7018Z is better than I-7018R and I-7018. (The reason is the same as I-87018Z)

i-7018z: [http://www.icpdas.com/products/Remote\\_IO/i-7000/i-7018z-g.htm](http://www.icpdas.com/products/Remote_IO/i-7000/i-7018z-g.htm)

i-87018z: [http://www.icpdas.com/products/Remote\\_IO/i-87k/i-87018z.htm](http://www.icpdas.com/products/Remote_IO/i-87k/i-87018z.htm)

For more description about using I-87018Z and I-7018Z, please refer to Chapter 11.3.9 or [www.icpdas.com](http://www.icpdas.com) – FAQ – Software – ISaGRAF - 055.

Please refer to Section 1.5 for connection instructions between the I-8xx7 controller system to the I-7000 and I-87xx series modules.

### Very Important:

Please wire an terminal resistor around 110 to 330 ohms at I-7188EG/XG, I-8xx7 or W-8xx7 controller 's RS-485 port , between the D+ and the D- pin (or between the RS-485+ and RS-485- pin).

This will ensure the host watchdog of I-7000 and I-87K output modules to work correctly when the communication between the controller and the I-7000 / I-87K output modules is broken.

(You can try 110 ohms first, then try 220 ohms, then others)

For example, if you don't wire any terminal resistor and enable the host watchdog function at "bus7000b" (Section 6.2 , the "host\_watchdog" parameter set as 1) , when you just un-plug the I-7000's "DATA+" pin (keep "Data-" pin connected with the controller) , you will see the watchdog doesn't work in this i-7000. If you wire a resistor about 110 ohms between the controller's RS-485 D+ and D- pin, if you un-plug anyone of I-7000's "Data+" or "Data-" pin, the watchdog will work correctly.



## 6.1: Configuring The I-7000 & I-87xx Modules

---

### Note:

**A. If RS-485 remote I-7000 and I-87xxx I/O module's type is Analog Input, please configure the format as "2's complement" by DCON utility.** Like : I-7005, I-7013, I-7015, I-7016, I-7017, I-7017R, I-7017RC, I-7018, I-7018R, I-7018Z, I-7019, I-7019R, I-7033, I-87013, I-87015, I-87016, I-87017, I-87017R, I-87017RC, I-87018, I-87018R, I-87018Z and I-87019R.

**B. If RS-485 remote I-7000 and I-87xxx I/O module's type is Analog Output, please configure the format as "Engineer Unit" by DCON utility.**

Like : I-7021, I-7022, I-7024, I-87022, I-87024 and I-87026 .

To begin configuration of the I-7000 and I-87xx series modules to the controller system, use the "DCON Utility" program to set up the I-7000 and I-87xx modules. Please use DCON utility of version 4.4.3 or later version. (at [http://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon\\_utility/](http://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon_utility/) or I-8000 CD-ROM:\napdos\driver\dcon\_utility\setup\ )

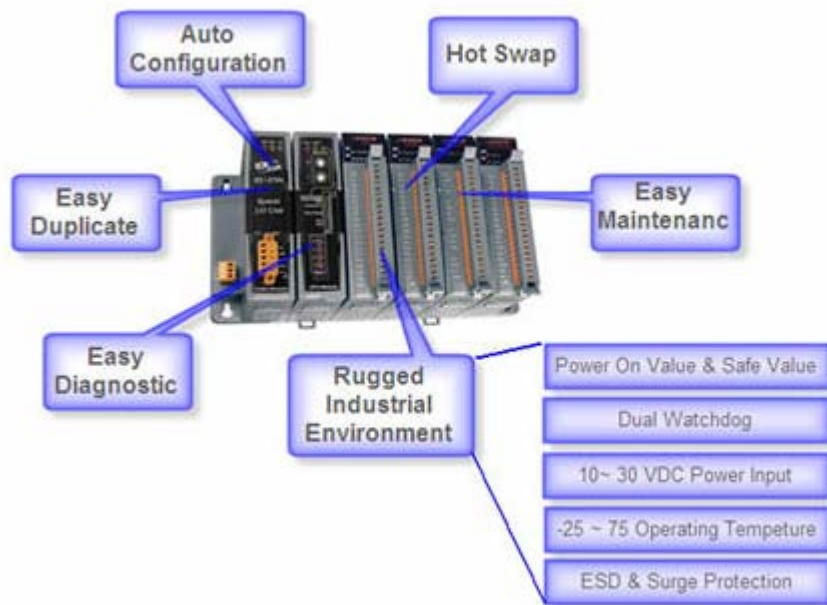
- Notes:**
1. Make sure the hardware connection is correct.
  2. Search and configure the modules one by one.
  3. Connect the module's INIT\* to GND and Power on the module.

---

### New Products – RU-87P4 (4 slots) , RU-87P8 (8 slots) : RS-485 expansion unit

- Better than i-87K4/ i-87K8
- If using with i-87K high-profile I/O cards, it supports hot-swap
- Auto-configuration: Only need to config it once at design time, then at run time, just plug and play
- Communication Addr, Baud rate & Checksum are set by rotary and dip switch.
- All ISaGRAF controllers support them just like supporting the i-87K4/8 (please refer to Chapter 6)

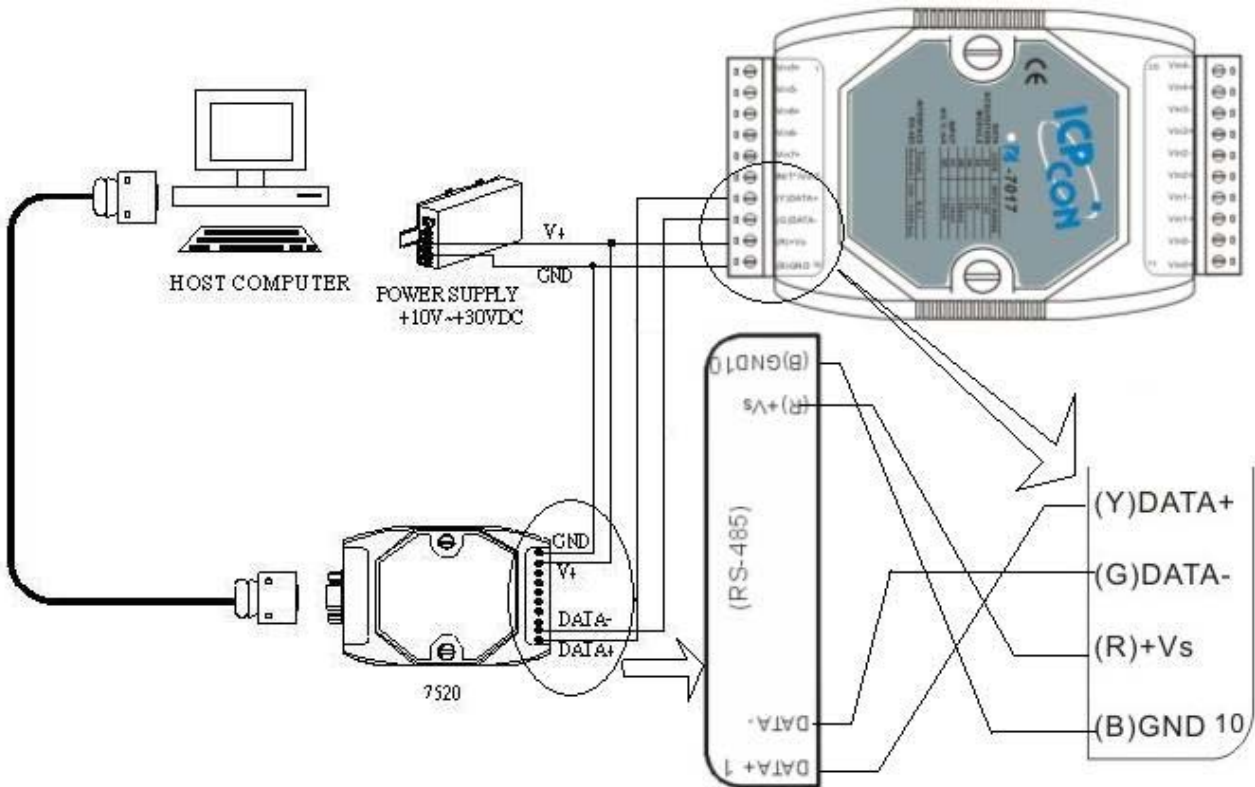
More at [http://www.icpdas.com/products/io\\_expansion\\_unit/ru-87p/ru-87p4.htm](http://www.icpdas.com/products/io_expansion_unit/ru-87p/ru-87p4.htm)



## Step 1: Hardware connection

A. The power supply must be DC power between +10V to +30V.

B. Wiring diagram for connecting to I-7000: (one module for each time)



**Note:** For configuring I-7000 and I-87xxx I/O module, you have to prepare an I-7520R converter. For other wiring diagram please refer to “[DCON Utility User’s Manual](#)”.

## Step 2: Set I/O module to initial state

If the module is a new one, factory have set a default settings for user’s convenient. If you don’t know the configuration of the module, please set the I/O module to initial state.

\*\*\* To set I-7000 module to initial state is to wire connect the INIT\* to GND and Power on the module. Then the module will become initial state. **(Some new designed I-7000 modules have a Dip-switch at its back. Please switch it to the “INIT” position , then power up the module)**

\*\*\* I-87K module’s initial state is set by I-87K4 / 5 / 8 / 9 ’s dip switch. For example, setting dip-2 to “ON”, and then re-cycle the power, it means the second slot is in initial state. (If using **I-87K5** and **I-87K9**, please do not plug I-87xxx board in its left-most slot for initial configuration . Please plug at 2nd to 9th slot for initial configuration. The dip-1 is for 2nd slot of **I-87K5** and **I-87K9**, ..., dip-4 is for 5th slot, Dip-8 is for 9th slot of I-87K9). **Some new designed I-87K I/O modules have dip-switch built-in. Their “INIT / Normal” state is controlled by its own dip-switch not by the dip-switch of I-87K4 / 5 / 8 / 9 .**

**The default state from factory:**

|            |               |                 |               |
|------------|---------------|-----------------|---------------|
| I/O Module | i-7000        | M-7000          | 87K series    |
| Address    | 1             | 1               | 1             |
| Baud rate  | 9600          | 9600            | 115200        |
| Checksum   | Disabled      | Not defined     | Disabled      |
| Protocol   | DCON Protocol | Modbus Protocol | DCON Protocol |



**The initial state after initiation:**

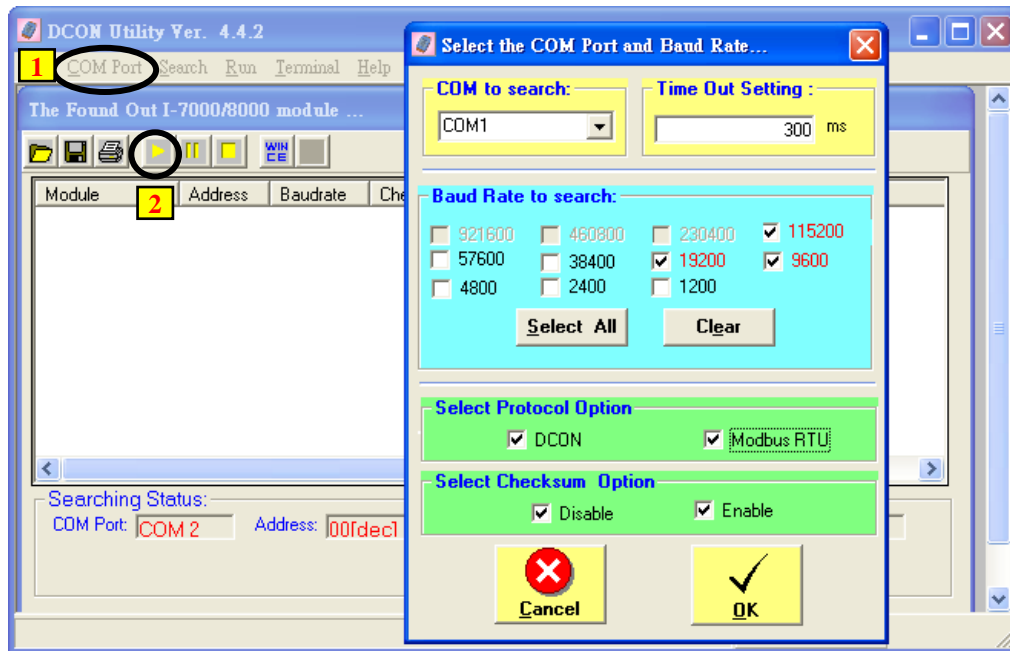
|            |                                 |               |
|------------|---------------------------------|---------------|
| I/O Module | 7000 series (i-7000 and M-7000) | 87K series    |
| Address    | 0                               | 0             |
| Baud rate  | 9600                            | 115200        |
| Checksum   | Disabled                        | Disabled      |
| Protocol   | DCON Protocol                   | DCON Protocol |

**Step 3: Select COM port and baud rate to search**

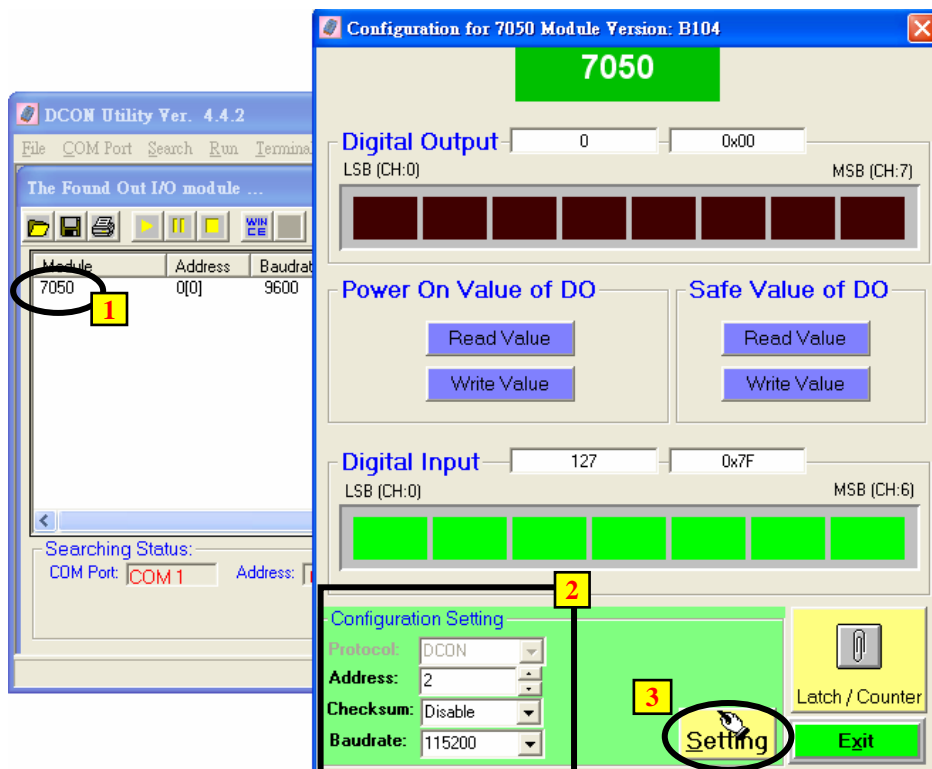
Execute the DCON Utility from “Start/programs/DAQPro/DCON Utility/”.

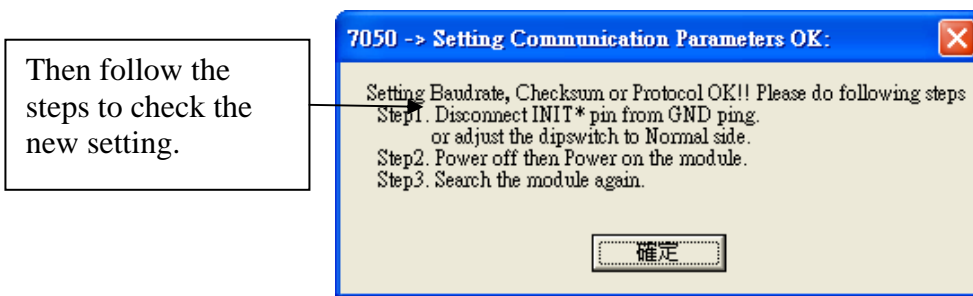


1. Click “COM Port” menu to select the COM port and baud rate to search. You can select multi-baud rate, protocol or checksum conditions if you do not know the module’s setting, but it will spend more time to scan the network. After selection, click “OK”.
2. Click  “Start Search” icon to begin search module. Click  when it is found.



**Step 4:** Click Searched module ID and give the new configuration





**Note:** Remember to remove the connection of I-7000's INIT\* and GND after the setting is well configured. Then recycle its power.  
For I-87K I/O, remember to switch the related Dip to "OFF", then recycle its power.

### IMPORTANT NOTES Regarding remote I-7000 & I-87xx Modules:

One I-8xx7, I-7188EG/XG controller system can link up to a maximum of 64 pcs. of I-7000 and I-87xx modules (**However 255 pcs for W-8xx7**). It is recommended though that you do not link more than 24 modules to a single I-8xx7 and 7188EG/XG, while 64 modules for a W-8xx7 controller system. **Each I-7000 and I-87xx module MUST have it's own unique address to properly link to an ISaGRAF controller system. The default "Checksum" setting is disabled** (If set as enabled, please connect "bus7000b" listed in section 6.2 and set "checksum" parameter to 1), **and make sure that all of the I-7000 and I-87xx modules are set to the same baud rate and same checksum setting as the controller system** (9600 baud by default).

When you receive any of the I-7000 series modules or I-87xxx modules you will receive documentation called "Getting Started With I-7000 Series Modules" that provides instructions on how to properly configure these modules. If you need assistance on changing the baud rate or checksum, please refer to the "Change Baud Rate & Checksum" section in the "Getting Started With I-7000 Series Modules". You can find all of the documentation on the CD provided with your I-7000 series module from ICP DAS in a file titled "getstart.pdf".

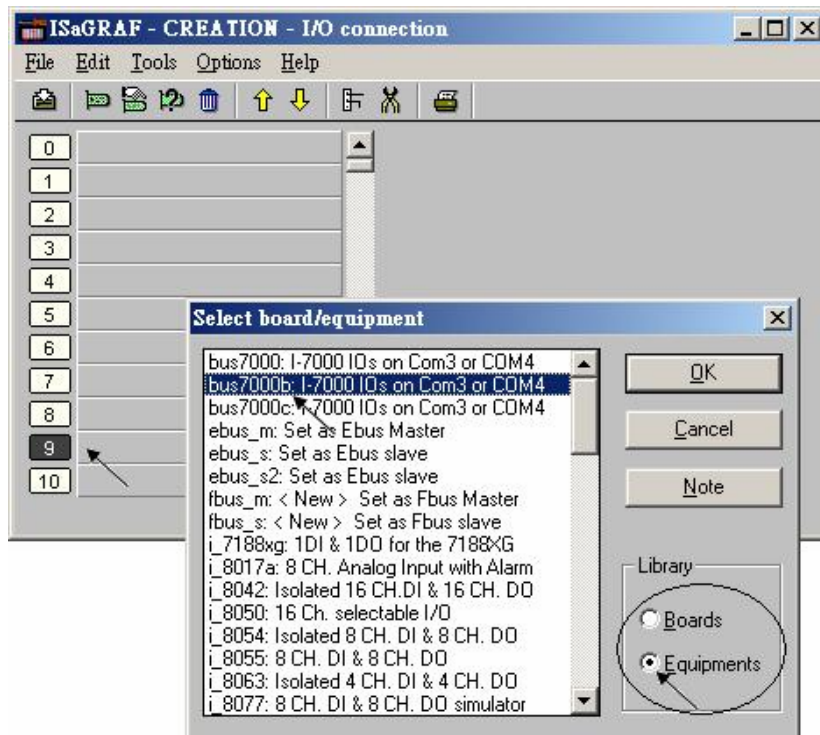
**If RS-485 remote I-7000 and I-87xxx I/O module's type is Analog Input, please configure the format as "2's complement" by DCON utility.** Like : I-7005, I-7013, I-7015, I-7016, I-7017, I-7017R, I-7017RC, I-7018, I-7018R, I-7018Z, I-7019, I-7019R, I-7033, I-87013, I-87015, I-87016, I-87017, I-87017R, I-87017RC, I-87018, I-87018R, I-87018Z and I-87019R.

**If RS-485 remote I-7000 and I-87xxx I/O module's type is Analog Output, please configure the format as "Engineer Unit" by DCON utility.**  
Like : I-7021, I-7022, I-7024, I-87022, I-87024 and I-87026 .

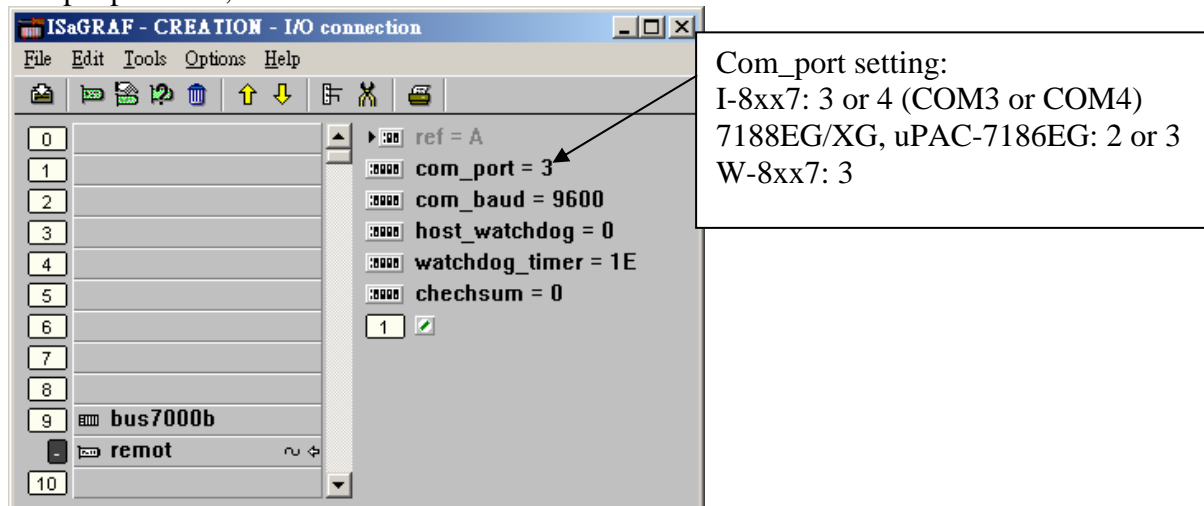
## 6.2: Opening The "Bus7000b" Function

To create a link between the I-8xx7, iPAC-8447/8847, I-7188EG/XG, uPAC-7186EG & W-8xx7 controller system and an I-7000 and I-87xx module, you need to connect the "Bus7000b" function through the "ISaGRAF I/O Connection" window. The "Bus7000b" function is considered a "virtual board", and must be selected from the "Equipments" section of the "Select Board/Equipment" window.

The "Bus7000b" MUST be connected to slot number 8 or higher on the "ISaGRAF I/O Connection" window (since slot 0 through 7 are used to connect to real I-8000 boards). **Only one "Bus7000b" can be linked to one I-8xx7, iPAC-8447/8847, I-7188EG/XG, uPAC-7186EG & W-8xx7 controller system!** If you attempt to connect more "Bus7000b" to an ISaGRAF controller, it will not work.



In the example provided, set the slot below number 9 to "Bus7000b: Remote".



The "**com\_port**" parameter can have a value of 3 (for COM3) or 4 (for COM4) for the I-8xx7 controller, while 2 (COM2) or 3 (COM3) for the I-7188EG/XG & uPAC-7186EG, and 3 (COM3) for the W-8xx7. This parameter defines which COM port ID the controller system will communicate with the I-7000 / I-87xx module. The default value for the "com\_port" parameter is 3.

The "**com\_baud**" parameter defines the baud rate that the I-8xx7, I-7188EG/XG, uPAC-7186EG & W-8xx7 will communicate with the I-7000 / I-87xx module. The possible values are 2400, 4800, 9600, 19200, 38400, 57600, and 115200. You must make sure that the controller system and the I-7000 / I-87xx modules are all set to the same "com\_baud" value.

The "**host\_watchdog**" parameter enables or disables the watchdog function for the I-7000 and I-87xx module. Setting the "host\_watchdog" parameter to a non-zero value will enable the "host\_watchdog" feature.

The "**watchdog\_timer**" parameter defines the amount of time before a "host\_watchdog" will occur. The value for the "watchdog\_timer" is defined in a **hexadecimal** value with the units defined in 0.1-second increments. For example, if the "watchdog\_timer" is set to a value of 1E, the "watchdog\_timer" is set for 3 seconds. If the "watchdog\_timer" value is set to 2A, the "watchdog\_timer" is set for 4.2 seconds.

The "**checksum**" parameter defines the remote IO is using "0: No checksum" or "1:with checksum". (The old "bus7000" not supporting "checksum" options. **Bus700b** does support it)

If the host watchdog feature is active and the watchdog timer is exceeded on the controller system (it means the connection is break between the controller and I-7000 / I-87xx modules), the I-7000 / I-87xx modules will go to a "safe" predetermined value by DCON utility. Normally for Digital Output channel, the "safe" state is D/O=OFF or D/O=False.

There is an analog input channel available on the "Bus7000b: Remote" virtual board. This analog input channel will return a value equal to the currently set baud rate.

## 6.3: Programming an I-7000 & I-87xxx Module

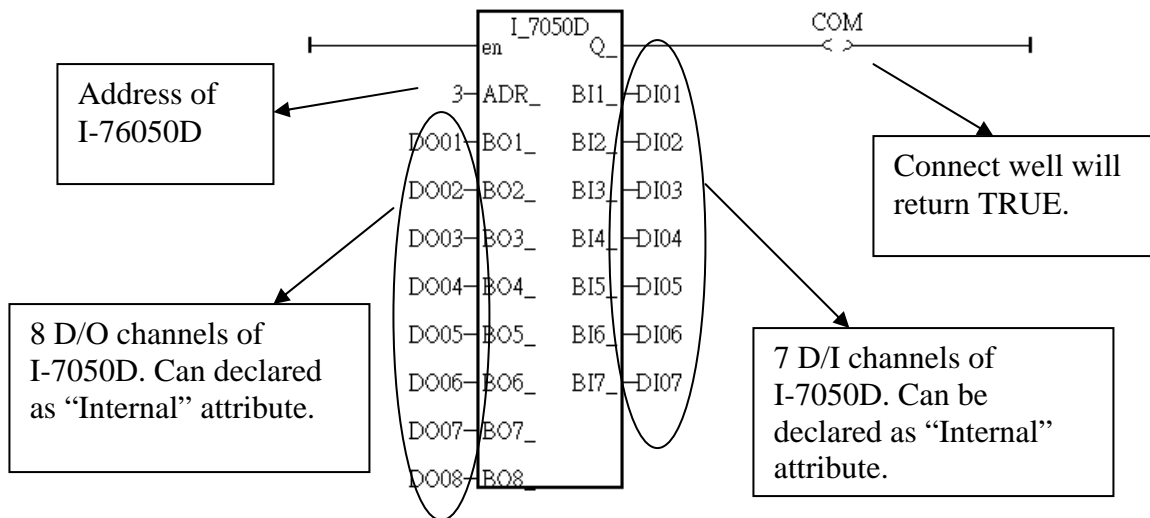
### 6.3.1: Program I-7xxx or I-87xxx remote IO function blocks

To link any I-7000 and I-87xx module to the I-8xx7, I-7188EG/XG & W-8xx7 controller system, the "Bus7000b" module MUST be opened first. Once the "Bus7000b" is opened, the "I\_7xxx" / "I-87xx" function block can now be programmed and you can access all of the I/O channels available from that function block, and that data can now be used in a LD program.

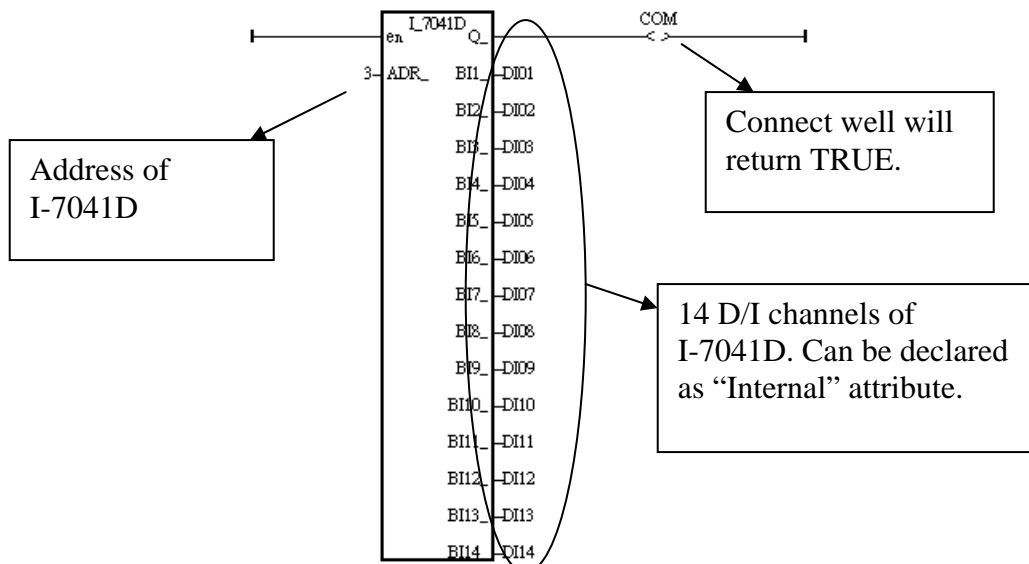
#### NOTE:

Please declare all variables which connect to the I-7xxx / I-87xx block as **"Internal"** attribution.

#### Example 1: Programming An I-7050D Module



#### Example 2: Programming An I-7041D Module

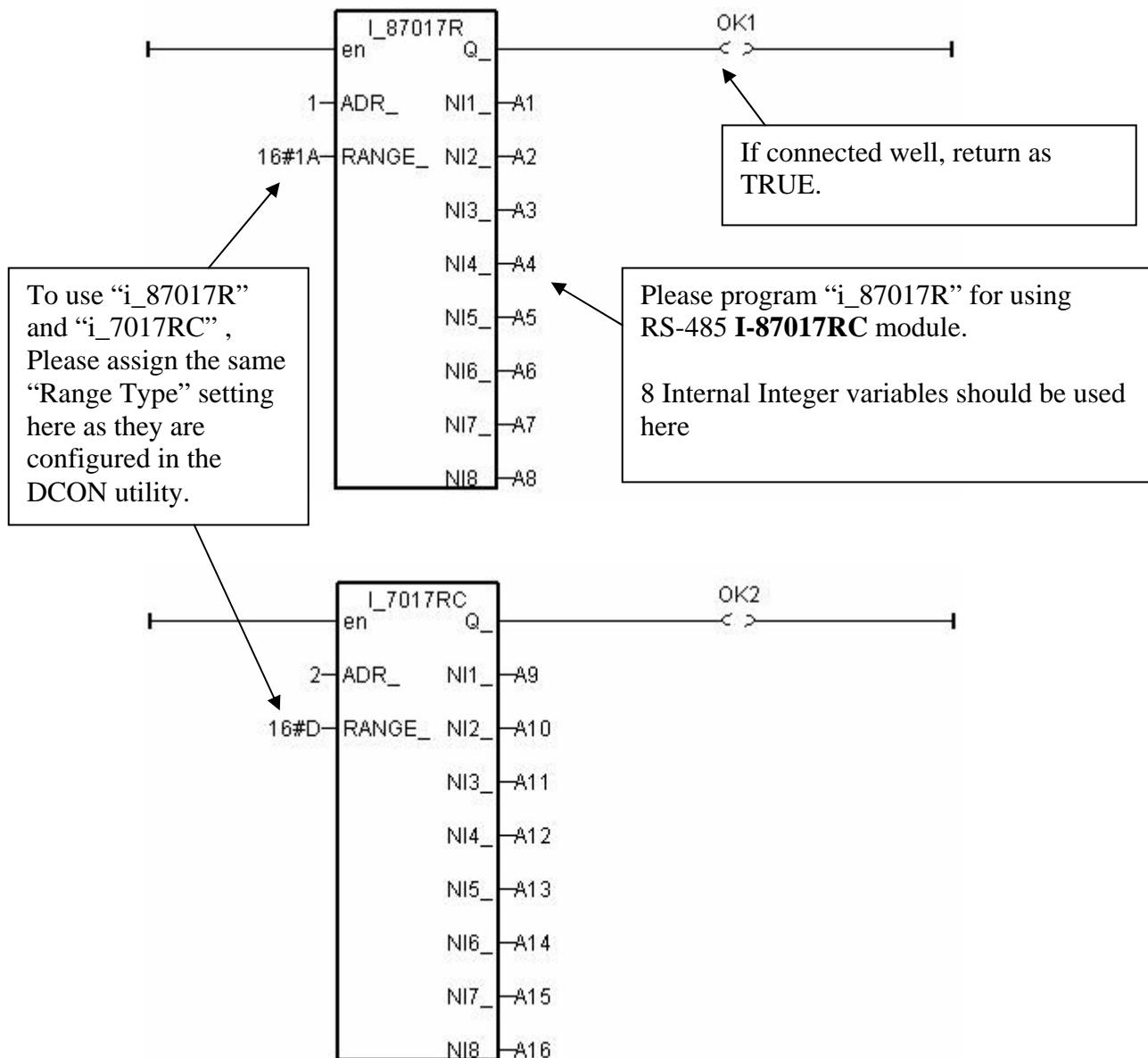




### Example 3: Programming a I-87017R or I-7017RC function block (current input measurement)

I-87017RC and I-7017RC can measure current input of  $\pm 20$  mA, 0 ~ 20mA and 4 ~ 20mA. Unlike I-87017, I-87017R and I-7017, no external 125 ohm resistor required. Please configure their format as “2’s complement” by DCON utility. (The “A4\_20\_to” function can be used to convert the analog input value to user’s engineering value, please refer to Appendix A.4)

| Range type<br>(by “DCON Utility”) | Physical value | I-7017RC /87017RC<br>Analog Input value |      |        |
|-----------------------------------|----------------|---|------|--------|
|                                   |                | - 32768                                 | 0    | +32767 |
| 7                                 | 4 ~ 20 mA      |   | 4 mA | 20 mA  |
| D                                 | $\pm 20$ mA    | - 20mA                                  | 0 mA | 20mA   |
| 1A                                | 0 ~ 20 mA      |   | 0 mA | 20 mA  |



**Note:**

If the current input sensor is 4 to 20 mA, user may better set I-7017RC or I-87017RC 's range type to "[D] : +/- 20 mA" , or "[1A] : 0 ~ 20 mA" . (set as "[7] : 4 to 20 mA" is not good )

The reason is :

If setting RS-485 I-7017RC or I-87017RC 's range type as "[7] : 4 to 20 mA", analog Input value of 0 or close to 0 could mean the Sensor input is 4 mA , and also possible the Sensor is broken-line. So it is not easy to distinguish these two situation by software.

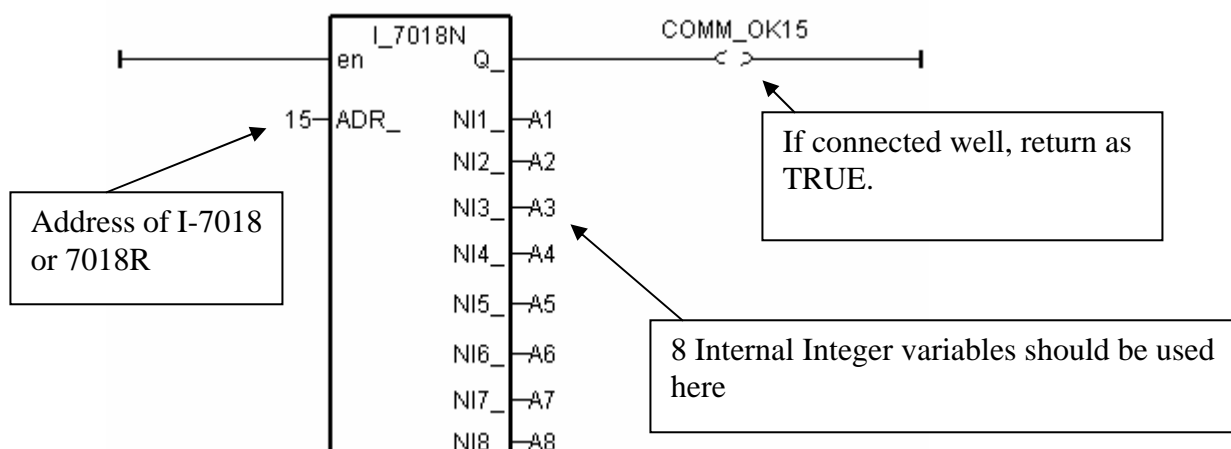
Howevr, if setting I-7017RC or I-87017RC 's range type as "[D] : +/- 20 mA" or "[1A] : 0 ~ 20 mA", analog input value of 0 or close to 0 only means the Sensor is broken-line . If the Sensor input is 4 to 20mA , the analog value should be 6553 to 32767 , not close to 0.

(Of course, the communication state of the RS-485 I-7017RC and I-87017RC should be Ok. The "Ok1" and "OK2" variable in the above example 3 can indicate the communication is Ok or not. If the communication is False, it means the controller can not link to the RS-485 I/O well. You need to handle this situation in your ISaGRAF program).

Example 4: Program I-7018 block for i-7018R and i-7018 (Please use new “i\_7018n” block)

( **i-7018z is a better hardware than i-7018 and i-7018R. Please refer to Chapter 11.3.9 or [www.icpdas.com](http://www.icpdas.com) – FAQ – Software – ISaGRAF – 055 for demo example**)

Please configure I-7018 and I-7018R ‘s format as “2’s complement” by DCON utility. Then please program a “I\_7018n” block (The “I\_7018n” block request all 8-channels by one single command, however the “I\_7018” block need to send 8 commands for 8-channels)



The other RS-485 I-7000 and I-87K I/O all use the similar way.

**Note:**

**If RS-485 remote I-7000 and I-87xxx I/O module’s type is Analog Input, please configure the format as “2’s complement” by DCON utility.** Like : I-7005, I-7013, I-7015, I-7016, I-7017, I-7017R, I-7018, I-7018R, I-7018Z, I-7019, I-7019R, I-7033, I-87013, I-87015, I-87016, I-87017, I-87017R, I-87018, I-87018R, I-87018Z and I-87019R.

**If RS-485 remote I-7000 and I-87xxx I/O module’s type is Analog Output, please configure the format as “Engineer Unit” by DCON utility.**

Like : I-7021, I-7022, I-7024, I-87022, I-87024 and I-87026 .

Below table is for the I-7017, 7017R, 87017, 87017R.

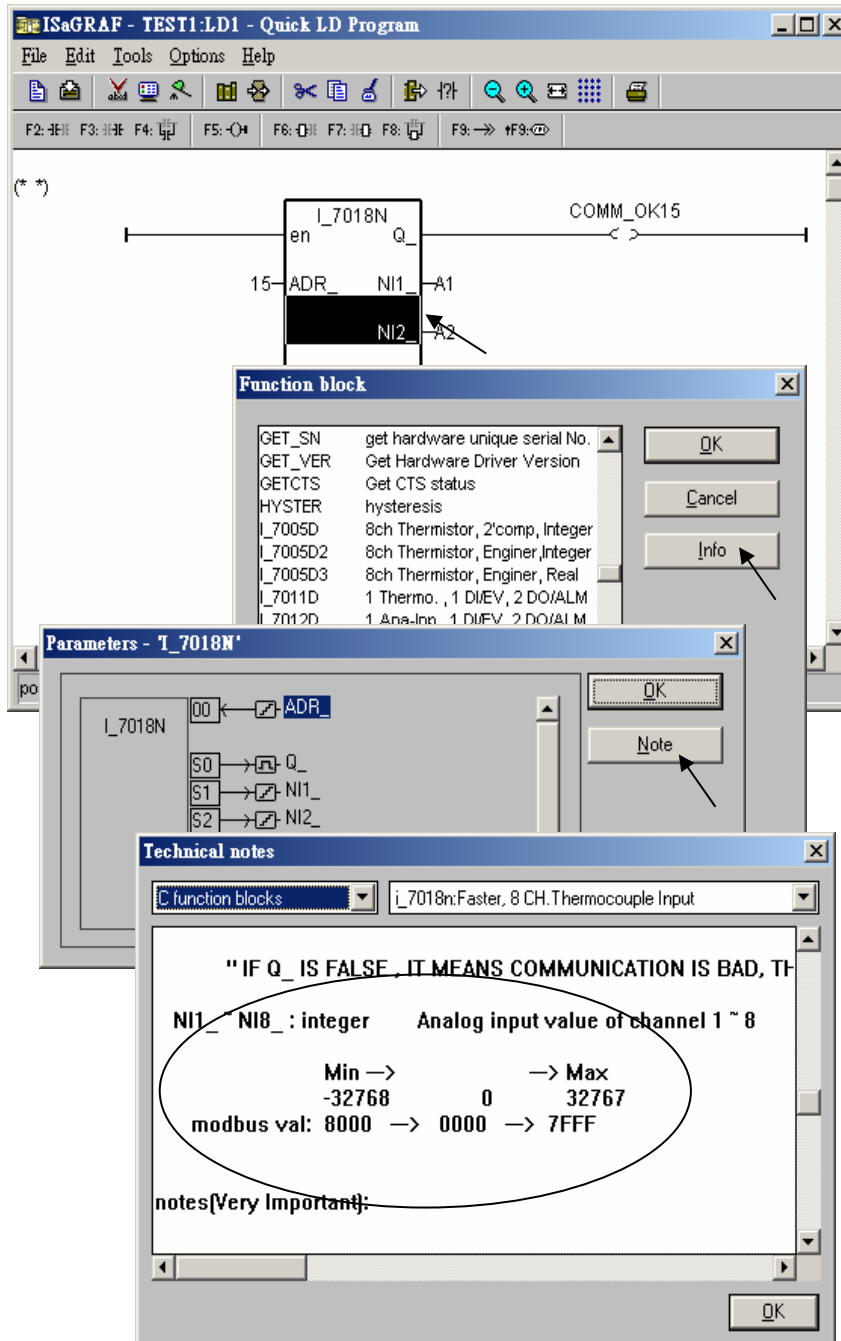
(These modules need external 125 ohm resistor if using “D:  $\pm 20\text{mA}$ ”)

| Range tyep<br>(by DCON Utility) | Physical value     | I-7017 / 87017<br>Analog Input value |     |         |
|---------------------------------|--------------------|--------------------------------------|-----|---------|
|                                 |                    | -32768                               | 0   | +32767  |
| 8                               | $\pm 10\text{V}$   | - 10V                                | 0V  | + 10V   |
| 9                               | $\pm 5\text{V}$    | - 5V                                 | 0V  | + 5V    |
| A                               | $\pm 1\text{V}$    | - 1V                                 | 0V  | + 1V    |
| B                               | $\pm 500\text{mV}$ | - 500mV                              | 0mV | + 500mV |
| C                               | $\pm 150\text{mV}$ | - 150mV                              | 0mV | + 150mV |
| D                               | $\pm 20\text{mA}$  | - 20mA                               | 0mA | + 20mA  |

Below table is for the i-7017RC and i-87017RC (no external 125 ohm resistor required)

| Range type<br>(by DCON Utility) | Physical value | I-7017RC / 87017RC<br>Analog Input value |      |        |
|---------------------------------|----------------|--|------|--------|
|                                 |                | - 32768                                  | 0    | +32767 |
| 7                               | 4 ~ 20 mA      |  | 4 mA | 20 mA  |
| D                               | ± 20mA         | - 20mA                                   | 0 mA | 20mA   |
| 1A                              | 0 ~ 20 mA      |  | 0 mA | 20 mA  |

Please refer to the on-line help for each I/O module's table or refer to the Appendix D.



### 6.3.2: Setting a special “ADR\_” parameter of remote temperature input module to get clear “Degree Celsius” or “Degree Fahrenheit” input value

ICPDAS provides many temperature input modules as below.

With “broken-line detection” or called “wire opening detection”

Thermocouple type: I-87018Z, I-87018R, 87019R, 7018Z, 7018R, 7018BL, 7019, 7019R

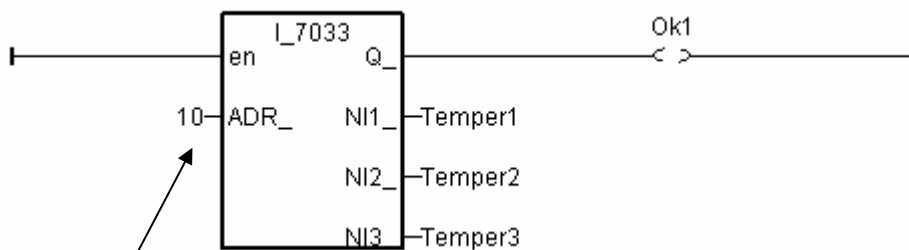
RTD type: I-87013, 87015, 7013, 7015, 7033

Thermister type: I-87005, 7005

Without “broken-line detection”

Thermocouple type: I-87018, 7018, 7018P

The “ADR\_” parameter of temperature IO function block can be “standard setting” or “special setting”. For example setting “ARD\_” of the “I\_7033” function block to 1 to 255 (Dec. value) means “standard setting”, the value of 1 to 255 indicates the address of the remote I-7033. The temperature input value is normally  $-32768$  to  $+32767$  in the case. It depends on the IO module’s “Type code” setting (Set by DCON utility). (normally value of  $-32768$  &  $+32767$  means wire “broken-line”)



ADR\_ = 10 (TT=00, RR=00, AA=0A, Hex.) means “standard setting”, address=10, the temperature input value is normally  $-32768$  to  $+32767$

If ADR\_ = 16#10201A (TT=10, RR=20, AA=1A, Hex) means “special setting”, “Degree Celsius”, “type code=20 of this I-7033 module set by DCON utility”, address=26, the temperature input value is a clear “Degree Celsius” value, for example, value of 4556 means “45.56” degree. “-500” means “-5.00” degree.

If user want to get a clear temperature input value, for example, value of 2312 means “23.12” Degree Celsius. Then please set “ADR\_” to a special value defined as below.

**Important:** Special “ADR\_” setting is supported since driver version of I-8xx7:3.11 , I-7188EG:2.09 , I-7188XG:2.07 , W-8xx7:3.24

Format: TTRRAA (Hex.)

TT=10 (Convert to "Degree Celsius")

TT=20 (Convert to "Degree Fahrenheit")

TT=00 (standard setting, -32768 to +32767. RR should be set as 00 if TT=00)

RR: "type code" setting of the related temperature input module

AA: address of the related temperature input module

For example, setting "ADR\_" as

- A. 16#102011 : (TT=10, RR=20, AA=11, Hex) the input value will be "Degree Celsius", unit is 0.01 degree, range= "20 : Platinum 100, a=0.00385, degree Celsius", address=17(Dec.). That results input value of "2356" = 23.56 Degree Celsius, "-489" = -4.89 Degree Celsius, "999990" = sensor broken-line.
- B. 16#202A03 : (TT=20, RR=2A, AA=03, Hex)) the input value will be "Degree Fahrenheit", unit is 0.01 degree, range= "2A : Platinum 1000, a=0.00385, degree Celsius", address=3(Dec.). That results input value of "4512" = 45.12 Degree Fahrenheit, "500" = 5.00 Degree Fahrenheit, "999990" = sensor broken line.
- C. 16#01 : (TT=00, RR=00, AA=1) standard setting, the input value will be , -32768 to +32767, address=1

## 6.4: Redundancy System by uPAC-7186EG or i-8437-80

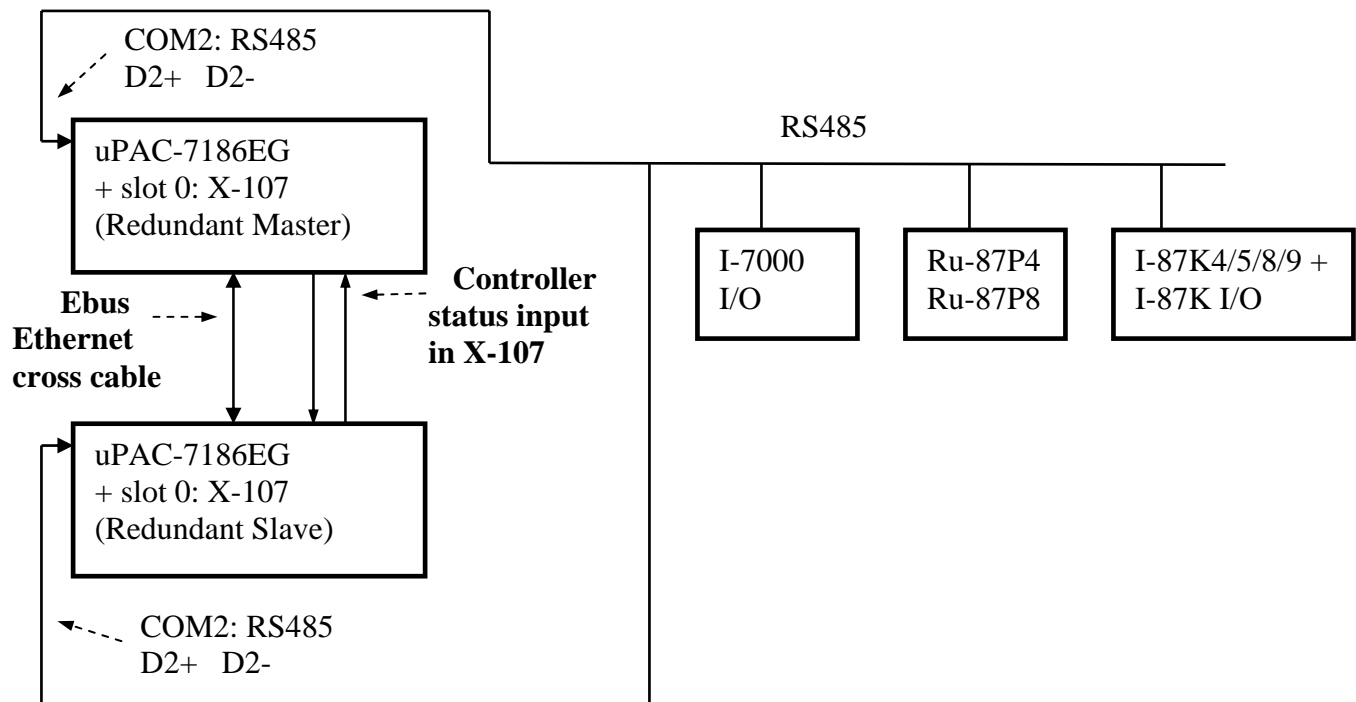
### Note:

1. W-8347 and W-8747 support best CPU redundancy solutions. Please refer to Chapter 20 and [www.icpdas.com](http://www.icpdas.com) – FAQ – Software – ISaGRAF - 062. The CPU speed of Wincon is about 10 to 20 times faster compared with the 40MHz i-8417/8817/8437/8837 and i-7188EG/XG 's CPU.
2. The uPAC-7186EG and i-8437-80 and future iPAC-8447 can setup a Bus7000b redundancy system. Their CPUs are 80MHz. The CPU speed is about 2 to 4 times of the 40MHz i-7188EG/XG 's CPU.
3. The 40 MHz i-8417/8817/8437/8837 and i-7188EG and i-7188XG are not good for Bus7000b redundancy system. Please use the best solution of Item (1) or the solution of Item (2)

### 6.4.1: Hot-Swap Redundancy System by uPAC-7186EG plus RU-87P4 / RU-87P8

uPAC-7186EG (Driver since it is released) supports Redundant Bus7000b. The configuration is listed as the following. It supports hot-swap I/O if using RS-485 expansion unit of RU-87P4 and RU-87P8 (Please refer to section 6.1). The Ebus are for exchanging data between the “Redundant Master” & “Redundant Slave”. The controller status inputs in X-107 's D/I channel are to indicate the other controller – “I am still alive”. Please wire Ch. 1 output of the redundant master 's X-107 to Ch.1 input of the redundant slave 's X-107. And also wire Ch. 1 output of the redundant slave 's X-107 to Ch.1 input of the redundant master 's X-107.

**uPAC-7186EG : Bus7000 redundancy system**

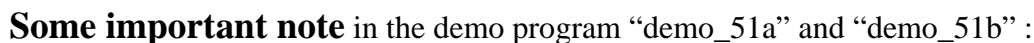


### Operations Principle:

1. When the system is powered up, the control of Bus7000b belong to “Redundant Master”.
2. If “Redundant Master” is damaged( or Power off), “Redundant Slave” takes the control of Bus7000b.
3. If “Redundant Master” is alive from damaged (or power up again), it takes the control of Bus7000b again. The change over time of (2) and (3) is about 5 seconds.
4. Control data is exchanging via Ebus (if using a cross cable, no need any ethernet switch) .
5. All I/O should be RS-485 I/O except the status I/O in the slot 0: X-107

<ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/7188eg/demo/>

PC / HMI / SCADA can connect to this bus7000 redundancy system with only one IP of the i-7188EX-MTCP



6-17



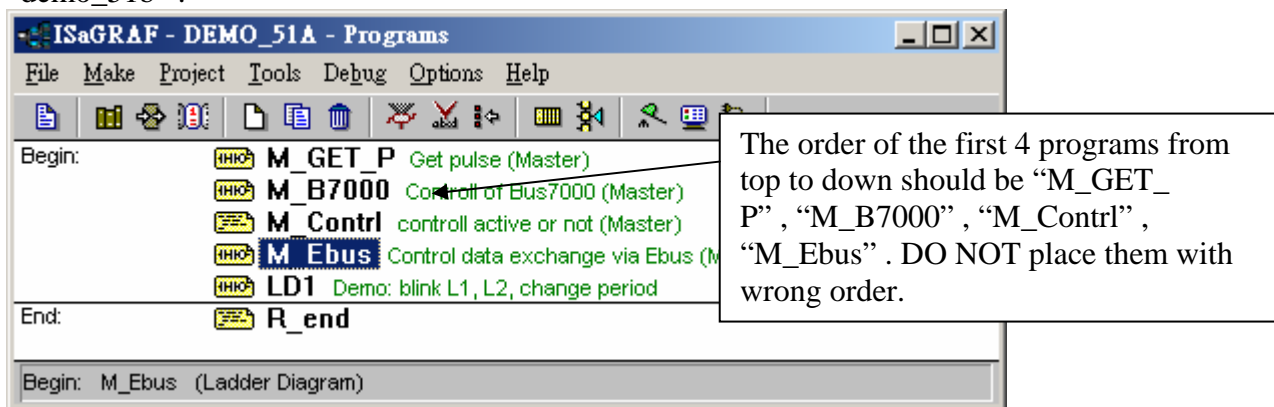
2. There are three useful variables which show “who is the current active controller” and the “controller (or Ebus communication or DI / DO) status” . Your HMI and SCADA can request these three values to know the current working status if assigning a proper Modbus network number to them. They are listed as following.

| Name      | Type    | Description  | NetWork number   |
|-----------|---------|--|------------------|
| i_am      | Integer | 1 : means I am Master controller<br>2 : means I am Slave controller  | Not assigned yet |
| Master_Ok | Boolean | True : Master controller is working OK<br>Flase: Master controller is not working OK<br>(for ex, power OFF, Ebus cable broken, DI/DO broken) | Not assigned yet |
| Slave_Ok  | Boolean | True : Slave controller is working OK<br>Flase: Slave controller is not working OK<br>(for ex, power OFF, Ebus cable broken, DI/DO broken)   | Not assigned yet |

3. There are three programs should to be modified in “demo\_51a” and “demo\_51b” to fit your real application.

- (1) In the “M\_B7000” and “S\_B7000” programs:  
search the “Add YOUR RS-485 I/O CONTROL of BUS7000 Here”
- (2) In the “M\_Contrl” and “S\_Contrl” programs:  
search the “Add YOUR CONTROL data here”
- (3) In the “M\_Ebus” and “S\_Ebus” programs:  
search the “ADD YOUR EBUS CONTROL in the Following”

4. All the other programs below the “M\_Ebus” and “S\_Ebus” in the “demo\_51a” and “demo\_51b” should be identical programs. Only the first 4 - “M\_GET\_P”, “M\_B7000”, “M\_Contrl” & “M\_Ebus” in the “demo\_51a” are different as the first 4 - “S\_GET\_P”, “S\_B7000”, “S\_Contrl” & “S\_Ebus” in the “demo\_51b” .

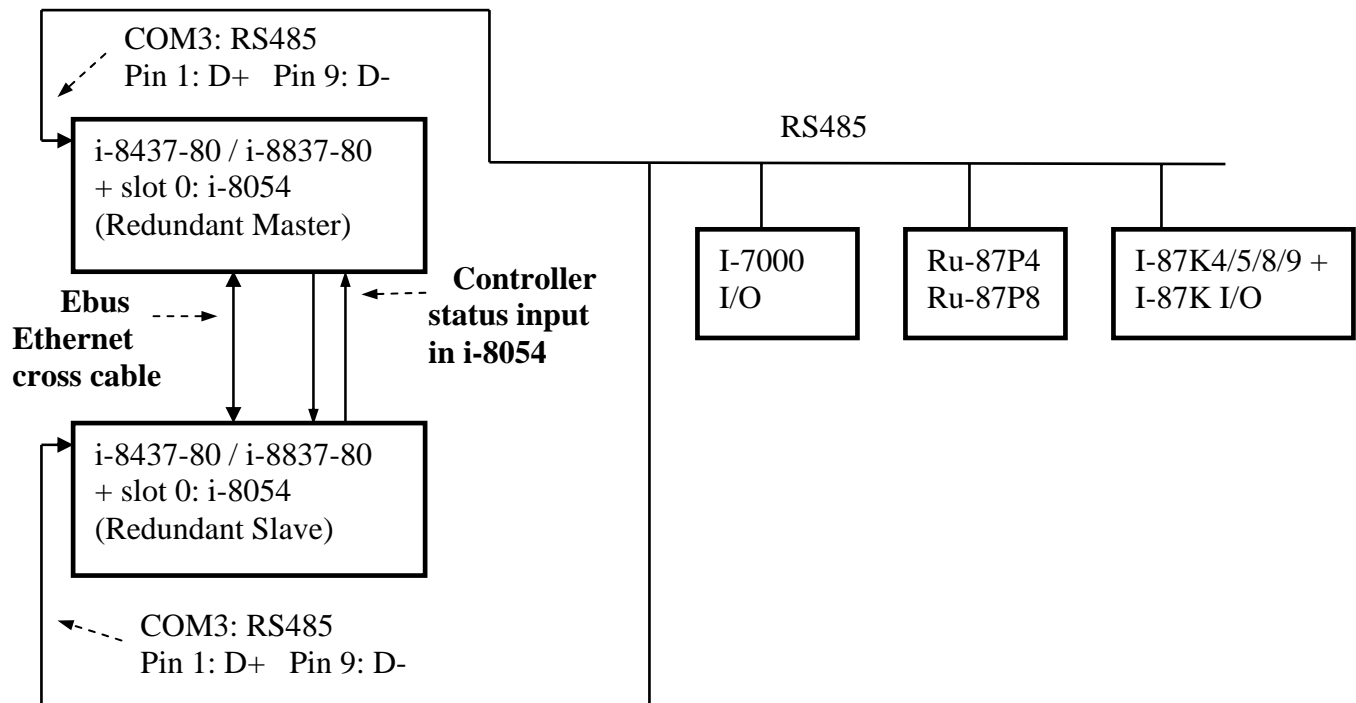


The order of the first 4 programs from top to down should be “M\_GET\_P”, “M\_B7000”, “M\_Contrl”, “M\_Ebus”. DO NOT place them with wrong order.

## 6.4.2: Hot-Swap Redundancy System by i-8437-80 plus RU-87P4 / RU-87P8

I-8437-80 / i-8837-80 (Driver rev.3.23 or later version) supports Redundant Bus7000b. The configuration is listed as the following. It supports hot-swap I/O if using RS-485 expansion unit of RU-87P4 and RU-87P8 (Please refer to section 6.1). The Ebus are for exchanging data between the “Redundant Master” & “Redundant Slave”. The controller status inputs in i-8054’s D/I channel are to indicate the other controller – “I am still alive”. Please wire Ch. 1 output of the redundant master’s i-8054 to Ch.1 input of the redundant slave’s i-8054. And also wire Ch. 1 output of the redundant slave’s i-8054 to Ch.1 input of the redundant master’s i-8054.

**I-8437-80 : Bus7000 redundancy system**

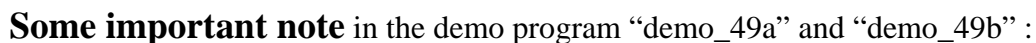


### Operations Principle:

1. When the system is powered up, the control of Bus7000b belong to “Redundant Master”.
2. If “Redundant Master” is damaged( or Power off), “Redundant Slave” takes the control of Bus7000b.
3. If “Redundant Master” is alive from damaged (or power up again), it takes the control of Bus7000b again. The change over time of (2) and (3) is about 5 seconds.
4. Control data is exchanging via Ebus (if using a cross cable,no need any ethernet switch) .
5. All I/O should be RS-485 I/O except the status I/O in the slot 0: i-8054

<ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/8000/demo/>

PC / HMI / SCADA can connect to this bus7000 redundancy system with only one IP of the i-7188EX-MTCP



6-20

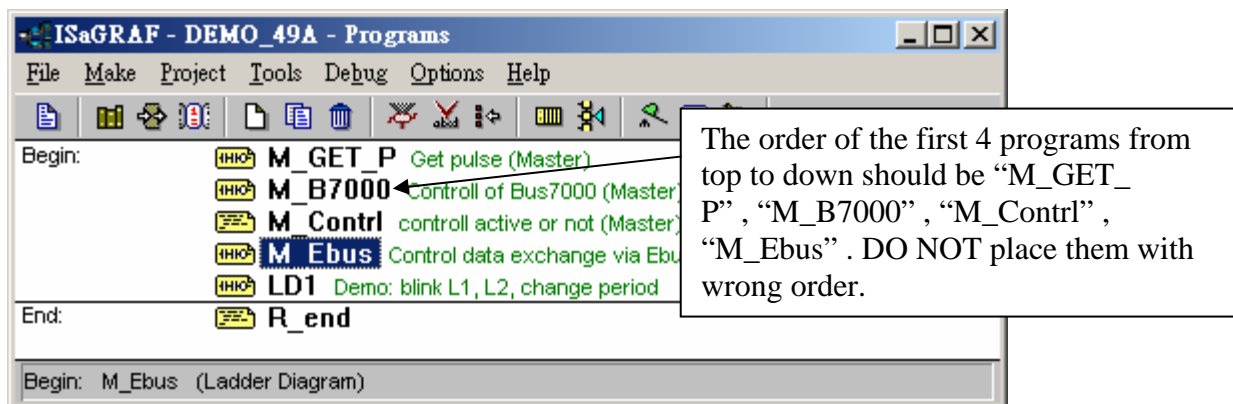
2. There are three useful variables which show “who is the current active controller” and the “controller (or Ebus communication or DI / DO) status” . Your HMI and SCADA can request these three values to know the current working status if assigning a proper Modbus network number to them. They are listed as following.

| Name      | Type    | Description  | NetWork number   |
|-----------|---------|--|------------------|
| i_am      | Integer | 1 : means I am Master controller<br>2 : means I am Slave controller  | Not assigned yet |
| Master_Ok | Boolean | True : Master controller is working OK<br>Flase: Master controller is not working OK<br>(for ex, power OFF, Ebus cable broken, DI/DO broken) | Not assigned yet |
| Slave_Ok  | Boolean | True : Slave controller is working OK<br>Flase: Slave controller is not working OK<br>(for ex, power OFF, Ebus cable broken, DI/DO broken)   | Not assigned yet |

3. There are three programs need to be modified in “demo\_49a” and “demo\_49b” to fit your real application.

- (1) In the “M\_B7000” and “S\_B7000” programs:  
search the “Add YOUR RS-485 I/O CONTROL of BUS7000 Here”
- (2) In the “M\_Contrl” and “S\_Contrl” programs:  
search the “Add YOUR CONTROL data here”
- (3) In the “M\_Ebus” and “S\_Ebus” programs:  
search the “ADD YOUR EBUS CONTROL in the Following”

4. All the other programs below the “M\_Ebus” and “S\_Ebus” in the “demo\_49a” and “demo\_49b” should be identical programs. Only the first 4 - “M\_GET\_P”, “M\_B7000”, “M\_Contrl” & “M\_Ebus” in the “demo\_49a” are different as the first 4 - “S\_GET\_P”, “S\_B7000”, “S\_Contrl” & “S\_Ebus” in the “demo\_49b” .



---

## Chapter 7. Controller To Controller Data Exchange

---

The I-8xx7, iPAC-8447/8847, I-7188EG/XG, uPAC-7186EG & W-8xx7 controller system provides the capability of exchanging data with other I-8xx7, iPAC-8447/8847, I-7188EG/XG, uPAC-7186EG & W-8xx7 controller systems.

There is one another way to deliver data to other system. Please refer to section 19.2 and 19.3 for "Deliver message via UDP and TCP/IP" .

### Important Note:

**The max. boolean & integer package No. of Fbus & Ebus reduce from 256 to 128 since driver version of I-8417/8817/8437/8837:2.42 , I-7188EG:1.32 & I-7188XG:1.29**

### 7.1: Basic Fbus Rules

---

Any I-8xx7 , I-7188EG/XG & uPAC-7186EG controller system can access data from another I-8xx7, I-7188EG/XG & uPAC-7186EG through the Fbus data exchange system. **While Wincon-8xx7 doesn't support Fbus, it supports Ebus only. Please refer to section 7.5 for programming Ebus in I-8437/8837, iPAC-8447/8847, I-7188EG, uPAC-7186EG & W-8037/8337/8737.** There are 2 types of data that can be exchanged through the Fbus protocol; they are "Boolean" and "integer".

The Fbus driver first creates a packet of eight Boolean values to form a "Boolean package", and then creates a packet of eight 32-bit integers to form an "integer package". Both of the "Boolean packages" and "integer packages" can be distributed on the Fbus to allow the data to be exchanged from one I-8xx7 & I-7188EG/XG controller system to another I-8xx7 & I-7188EG/XG controller system.

The Following Fbus Rules MUST Be Observed:

**RULE #1:** Each "Boolean package" must have an attached identification number ranging from 1 to 128. This means that there is a maximum of 128 "Boolean packages" that can be exchanged across an Fbus connection.

Each "Boolean package" contains 8 Boolean values, and these Boolean values can only have the value of either "True" or "False". The Boolean values in the "Boolean package" can be assigned and exchanged with either "Internal", "Input", or "Output" Boolean variables or Boolean constants.

**RULE #2:** Each "integer package" must have an attached identification number ranging from 1 to 128. This means that there is a maximum of 128 "integer packages" that can be exchanged across an Fbus connection.

Each "integer package" contains eight 32-bit integer values. The integer values can range from -2147483648 to 2147483647. The integer values in the "integer package" can be assigned and exchanged with either "Internal", "Input", or "Output" integer variables or integer constants.

**Rule #3:** Each number assigned to a "Boolean package" or an "integer package" can only be written to by one I-8xx7 & I-7188EG/XG controller system across the Fbus.

Each I-8xx7 & I-7188EG/XG controller system CANNOT **write** the same identification number for either a "Boolean package" or an "integer package" across the Fbus. WRITTING A PACKAGE IS NOT SHARED with the other I-8xx7 & I-7188EG/XG controller systems across the Fbus network.

In this example, there are five I-8xx7 controller systems communicating through an Fbus network, and the controller systems are named S1, S2, S3, S4, and S5 respectively. If the S1 controller system attempts to write a "Boolean package" with an ID of "1" and an "integer package" with an ID of "1" across the Fbus, the other four controllers CANNOT write either a "Boolean package" or an "integer package" with the same number. However, the other controller systems could write a "Boolean package" with an ID of "3" and an "integer package with an ID of "2".

There is no limitation on how many I-8xx7 & I-7188EG/XG controllers can read the same number package across the Fbus network. Any of the S2, S3, S4, S5 controller systems can read the "Boolean package" with an ID of "1" and the "integer package" with an ID of "1" if desired.

**Rule #4:** ONLY ONE I-8xx7 or I-7188EG/XG controller system can be configured as a Fbus "Master", all the others I-8xx7 & I-7188EG/XG controller systems MUST be configured as a Fbus "Slave".

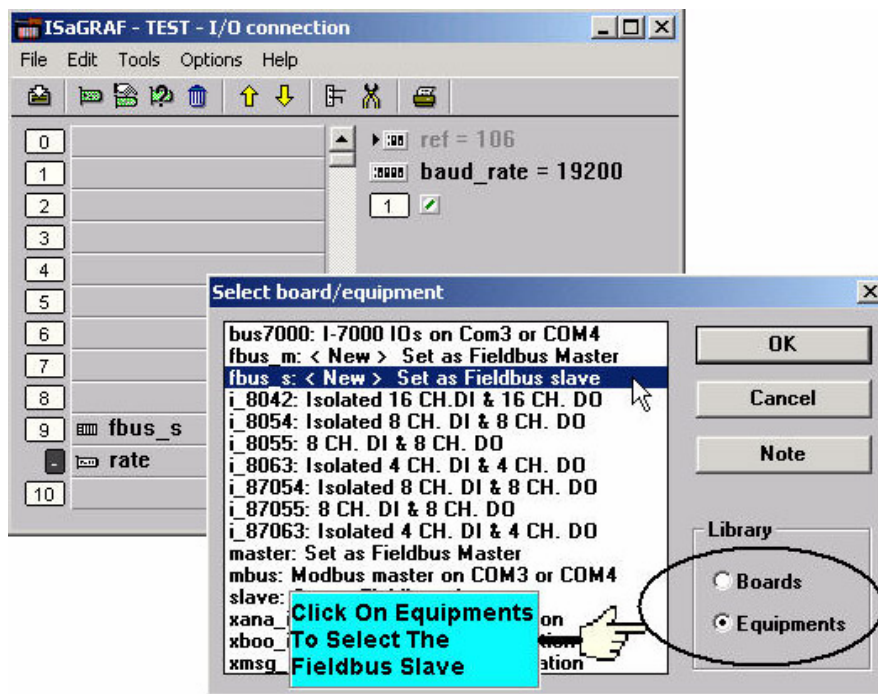
The "master" controller sends commands for how data is to be exchanged across the Fbus network. If you configure more than one I-8xx7 or I-7188EG/XG controller system as a "master", or configure none of the I-8xx7 & I-7188EG/XG controller systems as a "master" on the Fbus, NO DATA CAN BE EXCHANGED across the Fbus network.

**Important Note:**

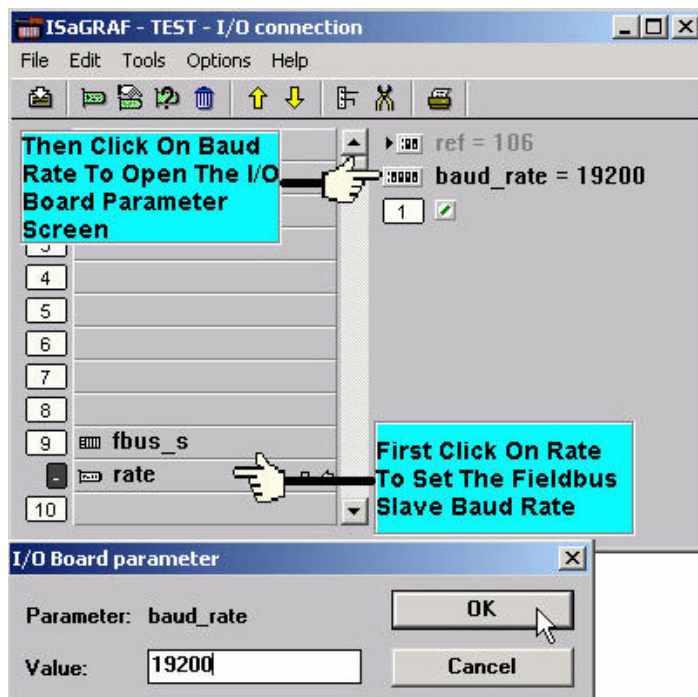
**The max. boolean & integer package No. of Fbus & Ebus reduce from 256 to 128 since driver version of I-8417/8817/8437/8837:2.42 , I-7188EG:1.32 & I-7188XG:1.29**

## **7.2: Configuring An I-8xx7 To Be A Fbus "Master" Or "Slave"**

To begin configuring an I-8xx7 & I-7188EG/XG controller system as either a Fbus master or slave, first open up the "ISaGRAF I/O Connections" window and double click on a slot number higher than 7. The "Select Board/Equipments" window will now open, click on "Equipments", and then double click on the "fbus\_s" selection to configure an Fbus slave, or double click on "fbus\_m" to configure an Fbus master. Remember, **ONLY ONE** controller can be the Fbus master, and you **CANNOT** configure an I-8xx7 & I-7188EG/XG controller system to be both a Fbus master and a Fbus slave.



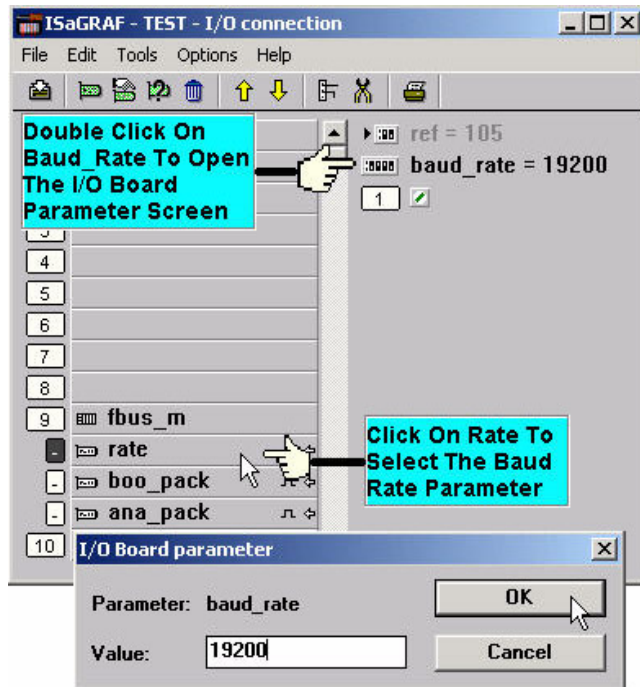
If you configure an I-8xx7 & I-1788EG/XG controller system as an Fbus slave, only one parameter needs to be set, and that is the "baud\_rate" parameter. The baud rate parameter can be set to 2400, 4800, 9600, 19200, 38400, 57600 or 115200 baud rate. The default baud rate value is 19200 for the I-8xx7 & I-1788EG/XG controller system. All controllers on the same Fbus network MUST be set to the same baud rate.



There is a digital input channel associated with the "fbus\_s: rate" equipment. This function will return the status when opening up an Fbus connection. If the Fbus connection has been established, the digital input channel will return a "TRUE" value. If the Fbus connection failed to establish, the digital input channel will return a "FALSE" value.



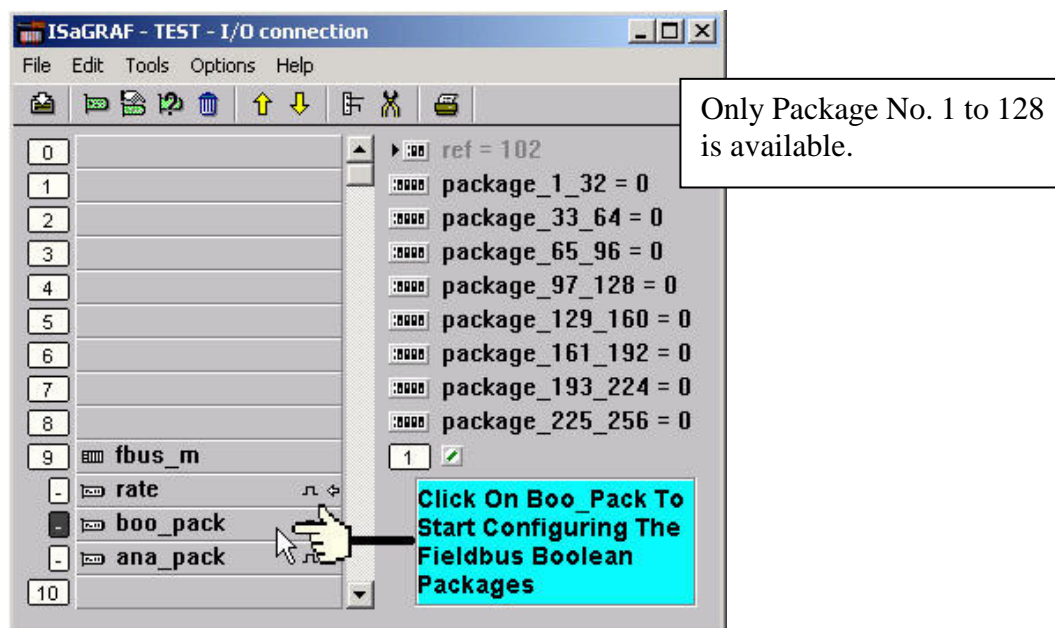
**If you configure an controller as Fbus master**, the parameter "baud\_rate" and "fbus\_m: rate" can be set to 2400, 4800, 9600, 19200, 38400, 57600 or 115200. The default value is 19200 for the controller. All controllers on the same Fbus **MUST** be set to the same baud rate.



There is a digital input channel associated with the "fbus\_m: rate" equipment. This function will return the status when opening up an Fbus connection. If the Fbus connection has been established, the digital input channel will return "TRUE" value, if the Fbus connection failed to establish, the digital input channel would return a value of "FALSE".

### 7.2.1: Configuring The Fbus Master Boolean Packages

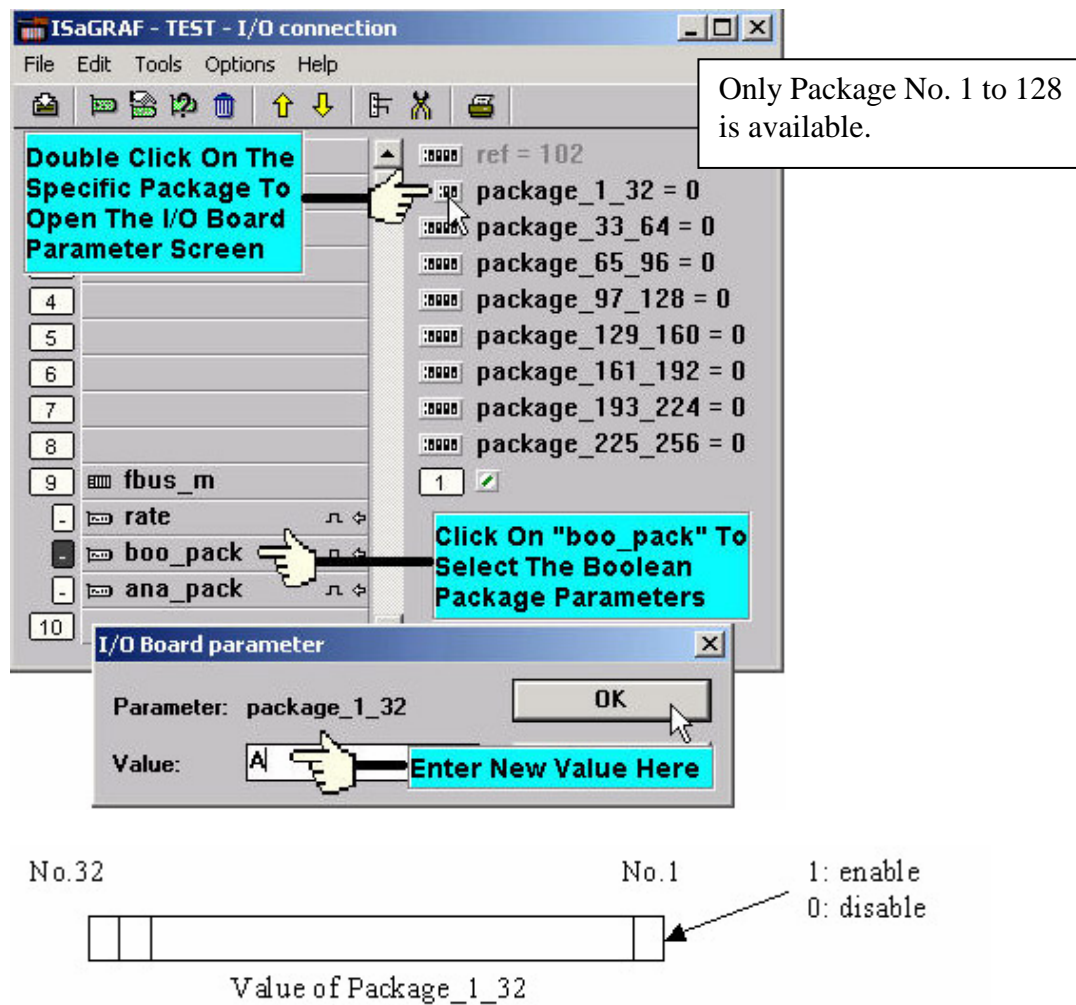
To begin configuring the Fbus Master Boolean Packages, click on the "boo\_pack" selection from the "fbus\_m" I/O connection.



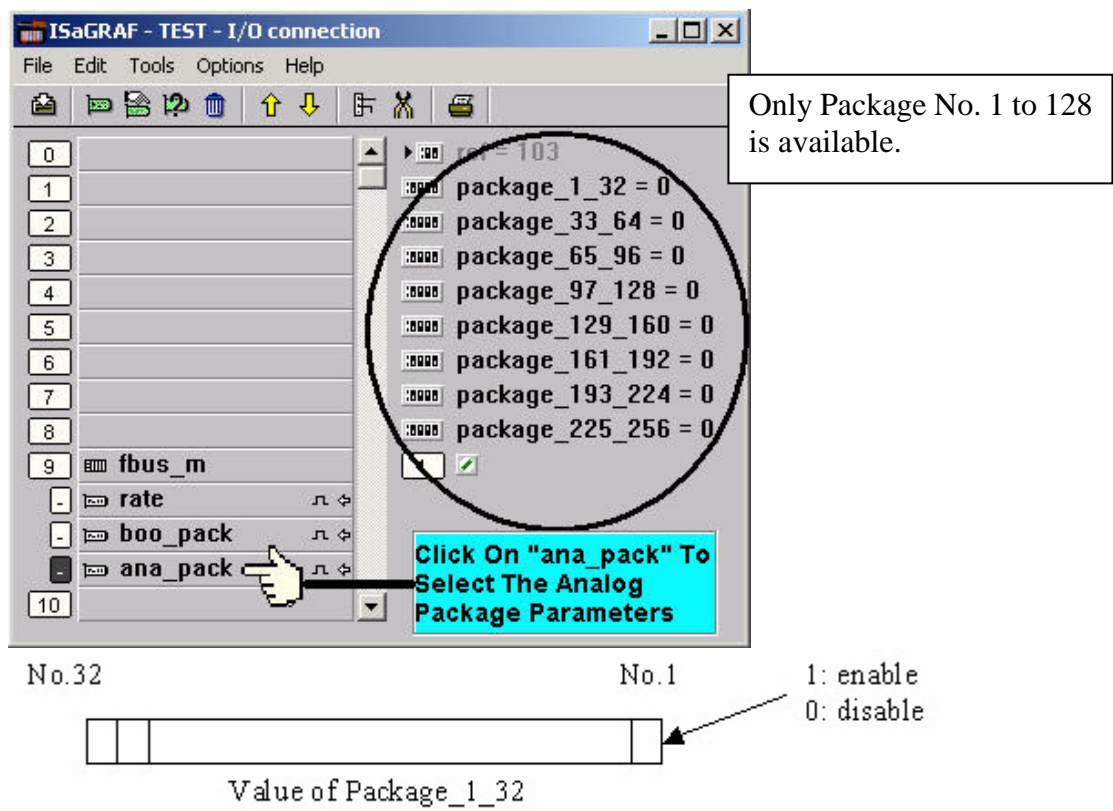


The parameter "package\_xxx\_xxx" at "fbus\_m: boo\_pack" indicates the "Boolean package" number which is allowed to be written to or read from across the Fbus network. The parameter value is given as a 32-bit integer in **hexadecimal**.

As an example, if the "package\_1\_32" is set to "FFFFFFFF" this will enable all the packages from number 1 to number 32 to be written to or read from across the Fbus network. If the "package\_1\_32" is set to a value of "A", this will only enable the number 2 and number 4 Boolean packages to be written to or read from across the Fbus network. The more packages that are enabled on a Fbus network the slower the communication efficiency will be. **With this in mind, always remember to enable only the required number of packages that you need for your application so you will have greater communication efficiency across the Fbus network.**



The parameter "package\_xxx\_xxx" at "fbus\_m: ana\_pack" indicates the "integer package" number which will be written to and read from on the Fbus network. The "fbus\_m: ana\_pack" is used to read and write 32-bit integer values across the Fbus network. Each of the parameter values is expressed as 32-bit integer values in **hexadecimal**, and the same configuration rules apply as those for the "Boolean package".



## 7.3: Programming Fbus Packages

Before you can exchange any data across a Fbus network, you must make sure that each I-8xx7 & I-7188EG/XG is either configured as either a Fbus master "fbus\_m" (and remember, only ONE controller can be the master) or Fbus slave "fbus\_s". Refer to Section 7.2 for details on how to implement these configurations.

The following Fbus function blocks can be used in a LD program to exchange data across an Fbus network.

|          |                            |
|----------|----------------------------|
| Fbus_b_r | read one boolean package.  |
| Fbus_b_w | write one boolean package. |
| Fbus_n_r | read one integer package.  |
| Fbus_n_w | write one integer package. |
| Fbus_f_r | read one REAL package      |
| Fbus_f_w | write one REAL package     |

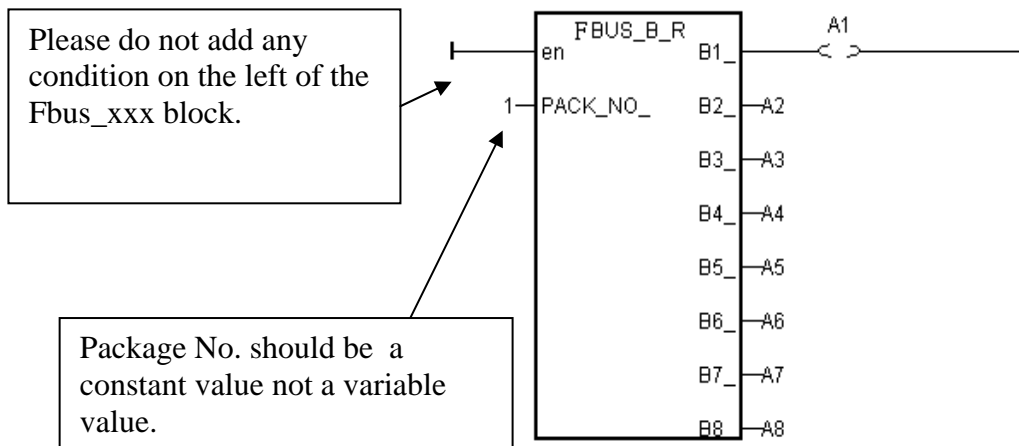
**(The Integer package and REAL package use the same memory. Please DO NOT use the same package No. as Integer package and also as REAL package at the same time. Or the local fault No. 116 may happen. Please refer to Chapter 10.6)**

The below block is to get the communication status of each Boolean & Integer Package.

|          |                             |
|----------|-----------------------------|
| Fbus_sts | Get status of each Package. |
|----------|-----------------------------|

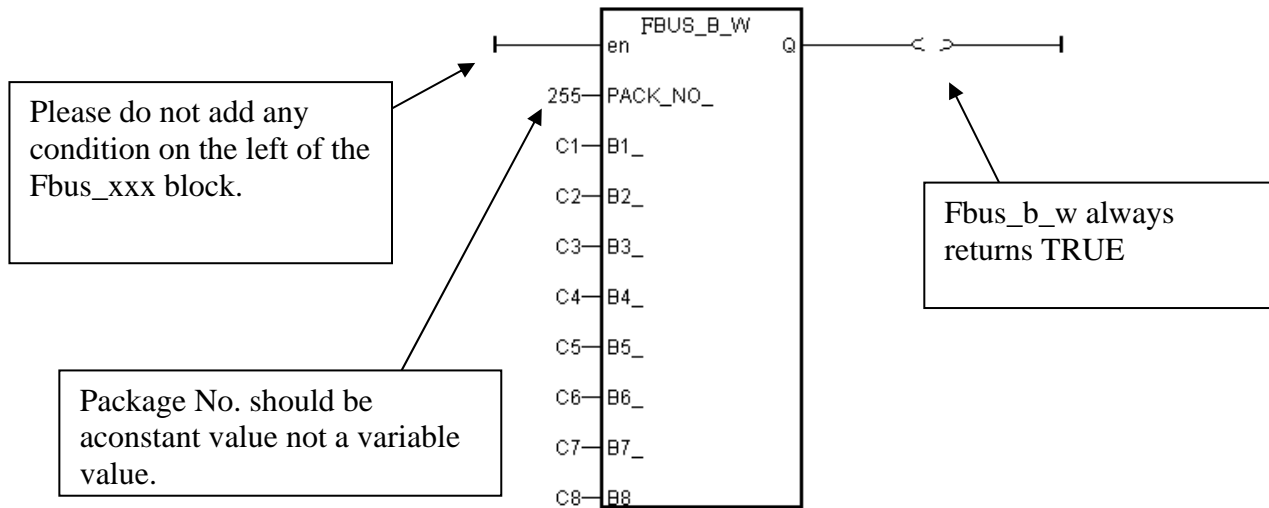
### Fbus Function #1: "Fbus\_b\_r"

The "Fbus\_b\_r" function reads one Boolean package from the Fbus network. In the example below the "Fbus\_b\_r" function has a Boolean package ID address of "1". The "A1" output contains the value of the first Boolean of the package No. of 1, the "A2" output contains the value of the second Boolean of the package No. of 1, and the "A3" output contains the value of the third Boolean of the package No. of 1. The other outputs follow the same format to where the "A8" output contains the value of the eighth Boolean of the package No. of 1.



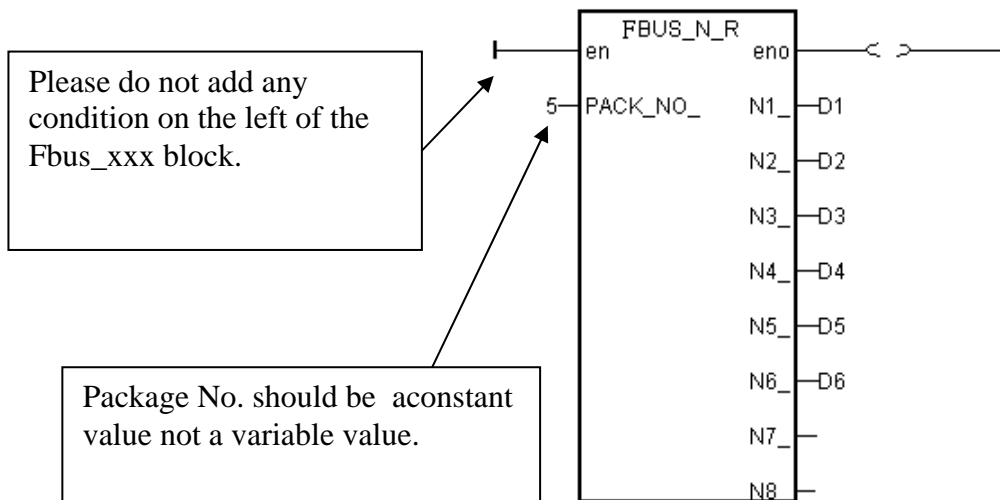
### Fbus Function #2: "Fbus\_b\_w"

The "Fbus\_b\_w" function writes one Boolean package on the Fbus network. In the example below the "Fbus\_b\_w" function has a Boolean package ID address of "255", the "C1" input writes a value to the first Boolean of the package No. of 255, the "C2" input writes a value of the second Boolean of the package No. of 255, and the "C3" input writes a value of the third Boolean of the package No. of 255. The other inputs follow the same format to where the "C8" input writes a value of the eighth Boolean of the package No. of 255.



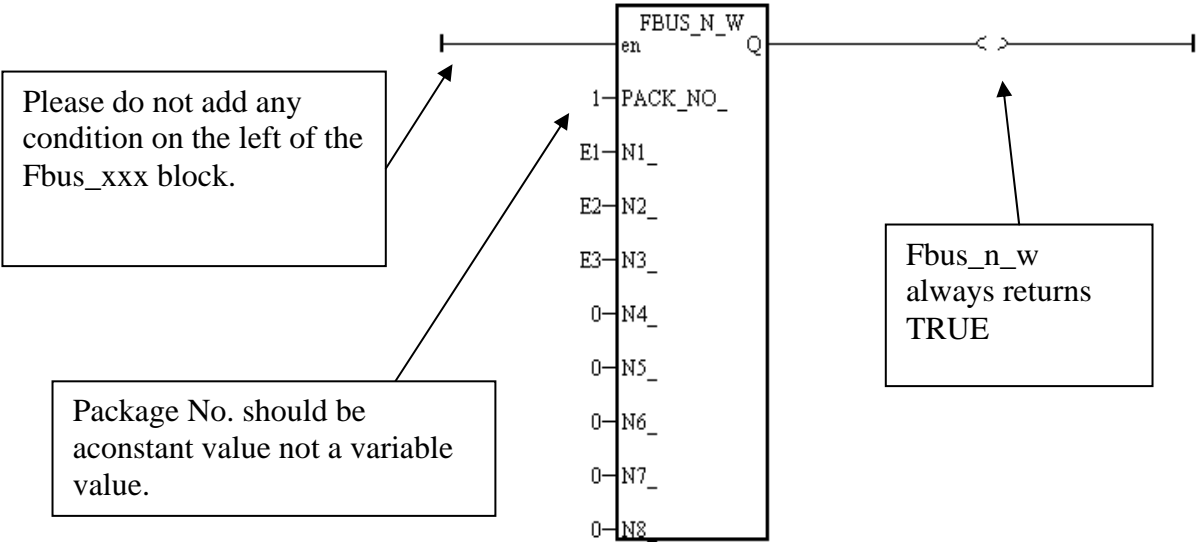
### Fbus Function #3: "Fbus\_n\_r"

The "Fbus\_n\_r" function reads one integer package from the Fbus network. In the example below the "Fbus\_n\_r" function has an Integer package ID address of "5". The "D1" output contains the value of the first integer of the package No. of 5, the "D2" output contains the value of the second integer of the package No. of 5, and the "D3" output contains the value of the third integer of the package No. of 5. The other outputs follow the same format to where the "D6" output contains the value of the sixth integer of the package No. of 5.



**Fbus Function #4: "Fbus\_n\_w"**

The "Fbus\_n\_w" function writes one integer package to the Fbus network. In the below example the "Fbus\_n\_w" function write variables “E1” to the first integer of the package of No. 1. “E2” to the second integer of the package of No. 1. “E3” to the third integer .



## 7.4: An Fbus Data Exchange Example

### Example Description:

In this Fbus data exchange example there are three I-8xx7 controller systems linked together in an Fbus network. The I-8xx7 controller systems are named "SA (master I-8xx7 controller system #1)", "SB (slave I-8xx7 controller system #2)", and "SC (slave I-8xx7 controller system #3)".

One of the digital input values from the SA controller (master I-8xx7 system) needs to be shared with the SB and SC (the slave I-8xx7 systems) controllers across the Fbus network, and the name for this digital input value will be called "ZZ".

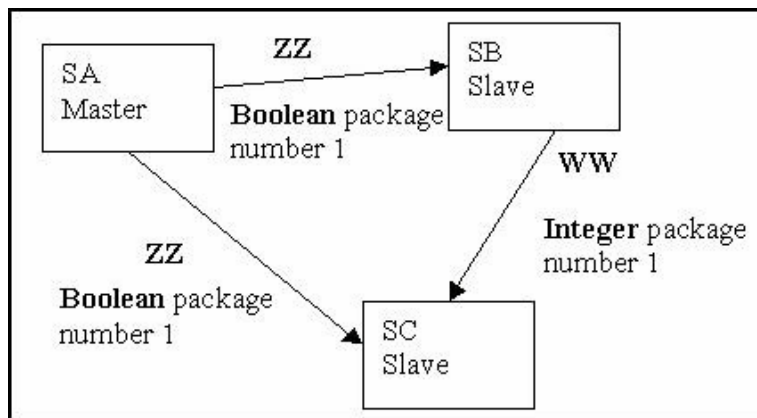
The first task of this example is to create an **Input** variable named ZZ on the SA controller system. Use the "ISaGRAF Project" window to declare ZZ as an "input" variable, and then link the ZZ input variable using the "ISaGRAF I/O Connections" window for the SA controller system.

Next, you will need to declare a Boolean **Internal** variable named ZZ for both the SB and SC controllers (so they can exchange the ZZ value with the SA controller system). You must declare the ZZ variable as an internal variable for the SB and SC controllers because there is only one real input variable (from the SA controller) that is being exchanged, and neither the SB or SC has a real input variable named ZZ.

An additional requirement for this example is that an internal integer value named "WW" that comes from the SB controller system needs to be shared with the SC controller system. To accomplish this declare an **Internal** integer variable named WW on both the SB and SC controller systems.

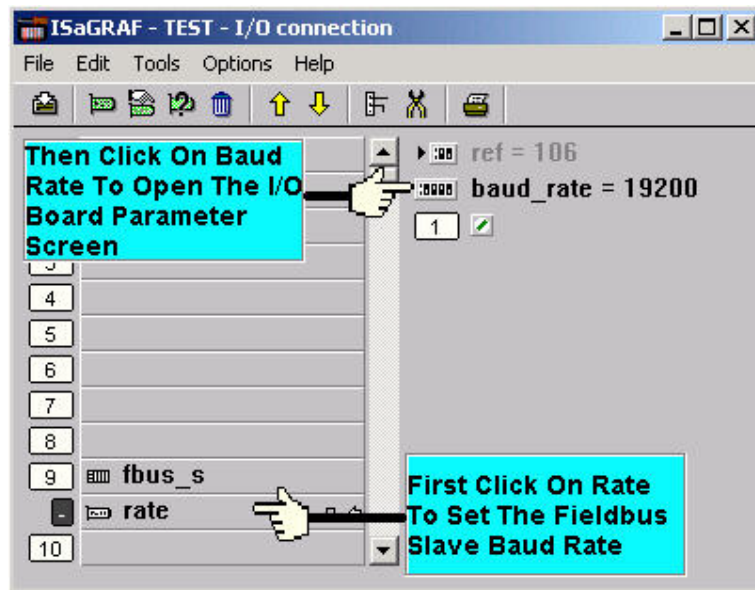
### Example Prerequisites:

The SA controller system is the Fbus master controller and the SB and SC controllers are Fbus slave controllers. Each of the controllers has their baud rates set to 19200.

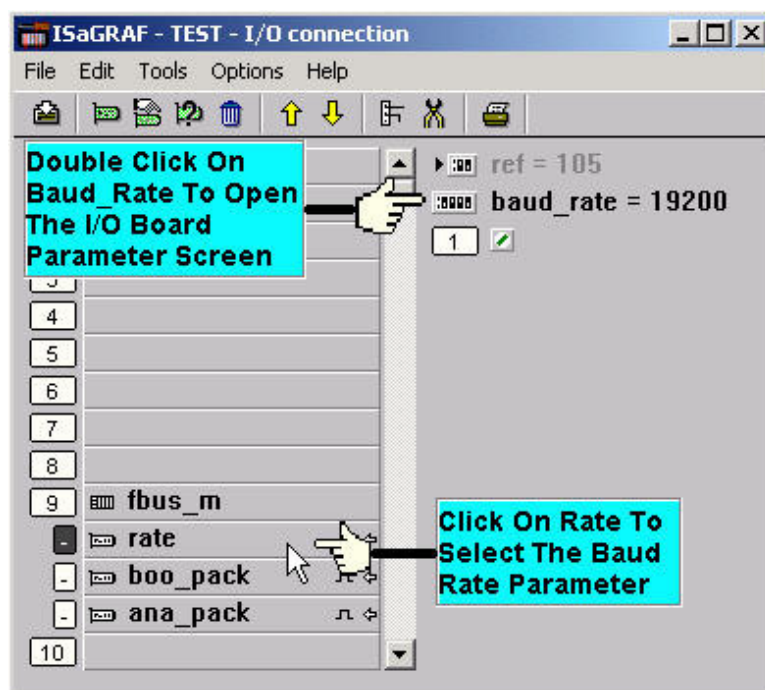


Setting The SB and SC Controllers As Fbus Slaves:

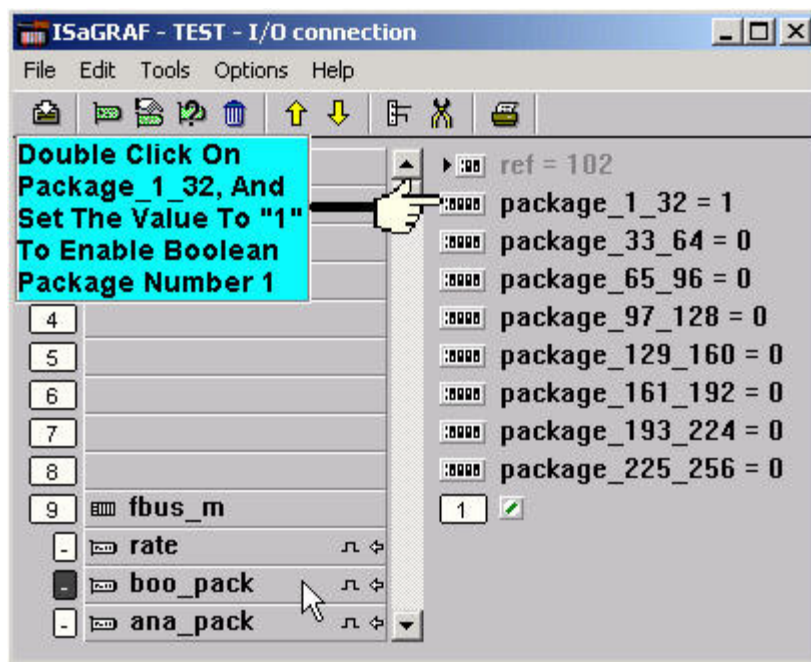
You should use the "ISaGRAF I/O Connections" window to declare the SB and SC controller systems as Fbus slaves.



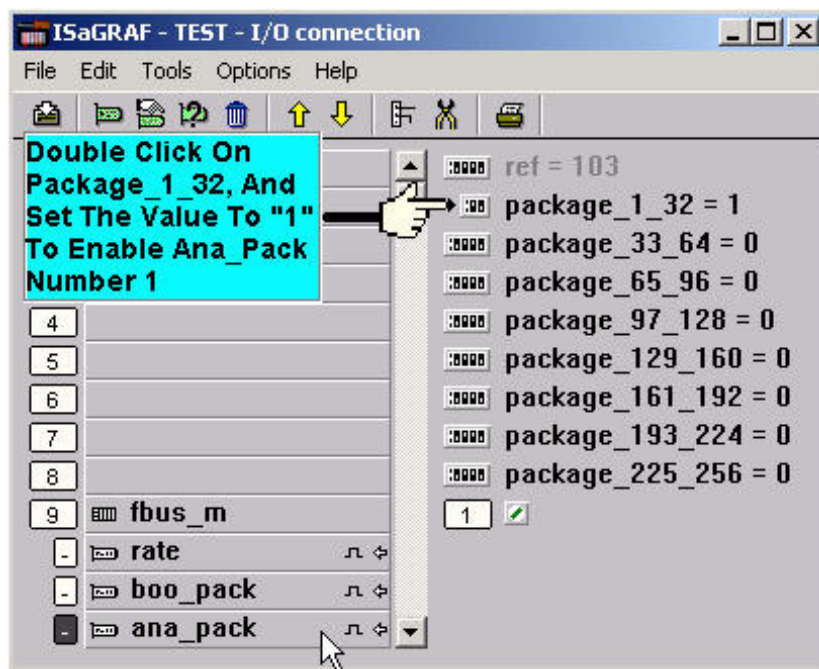
Use the "ISaGRAF I/O Connections" window to declare the SA controller system as the Fbus master controller.



Additionally, enable the Boolean package for the SA controller:

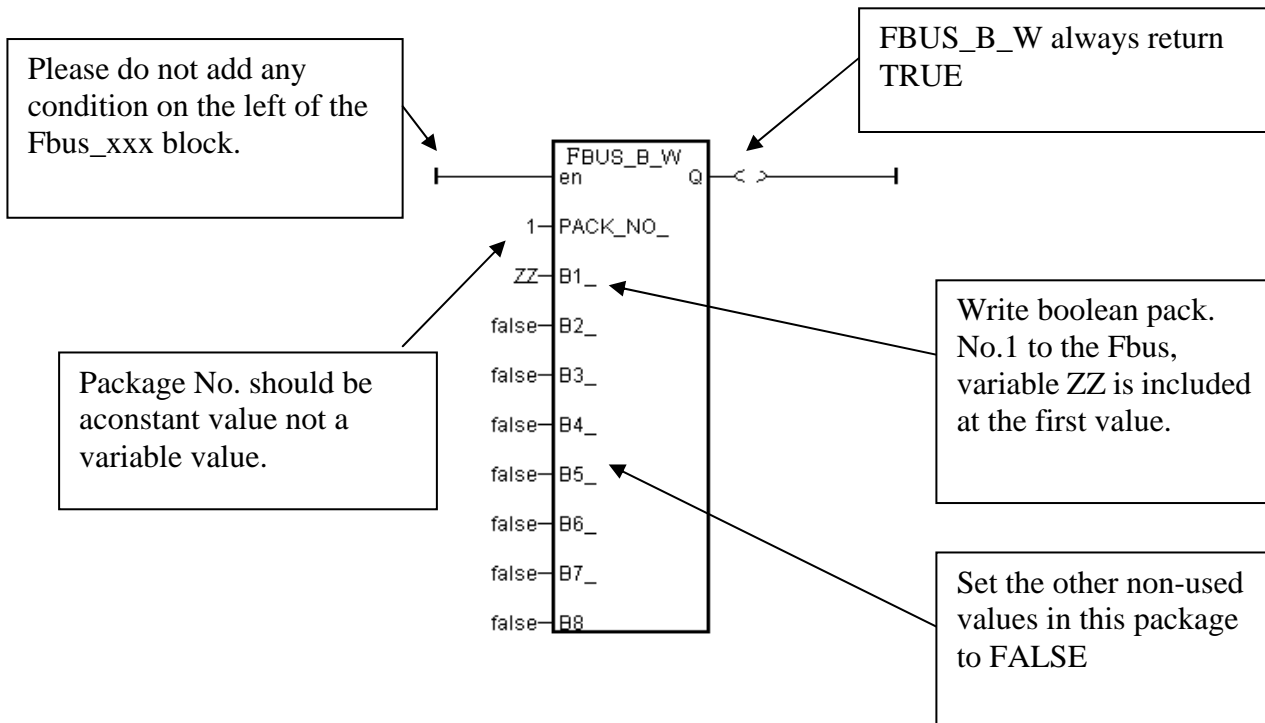


Also enable the integer package for the SA controller system:

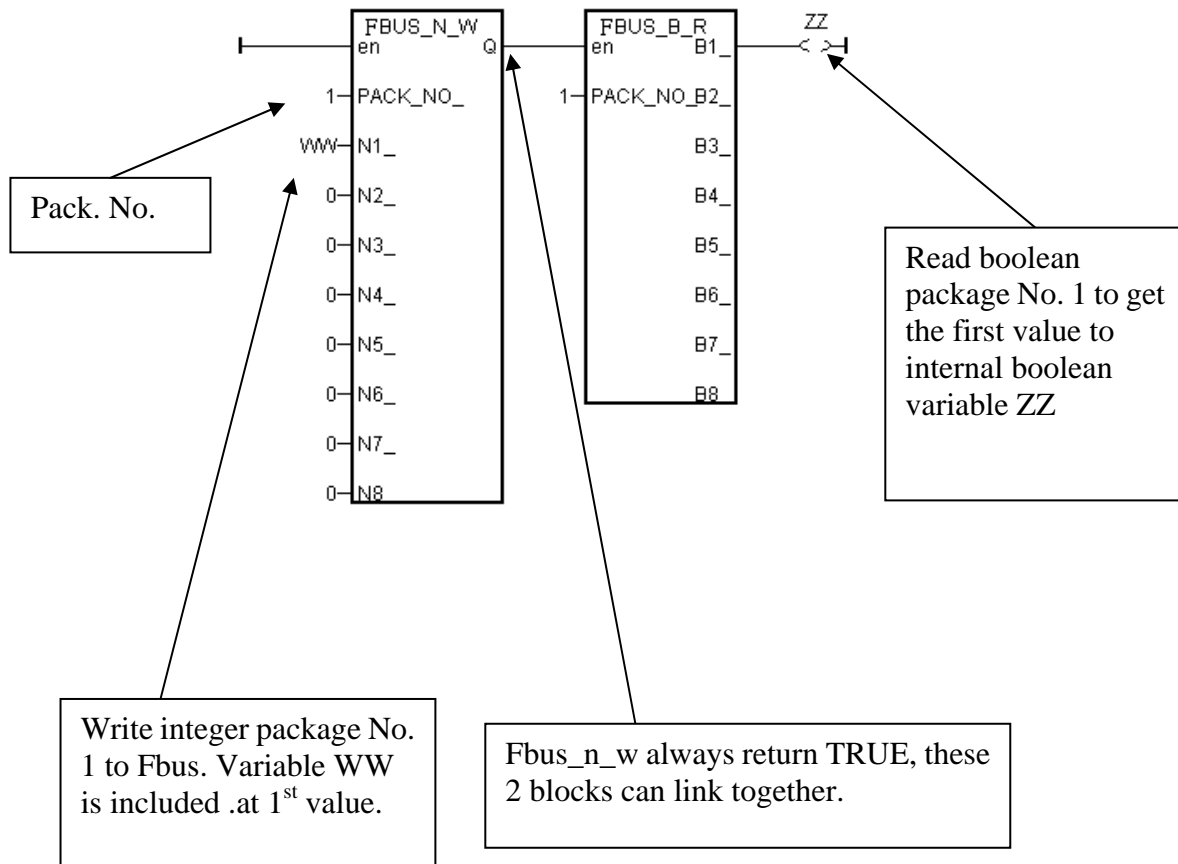




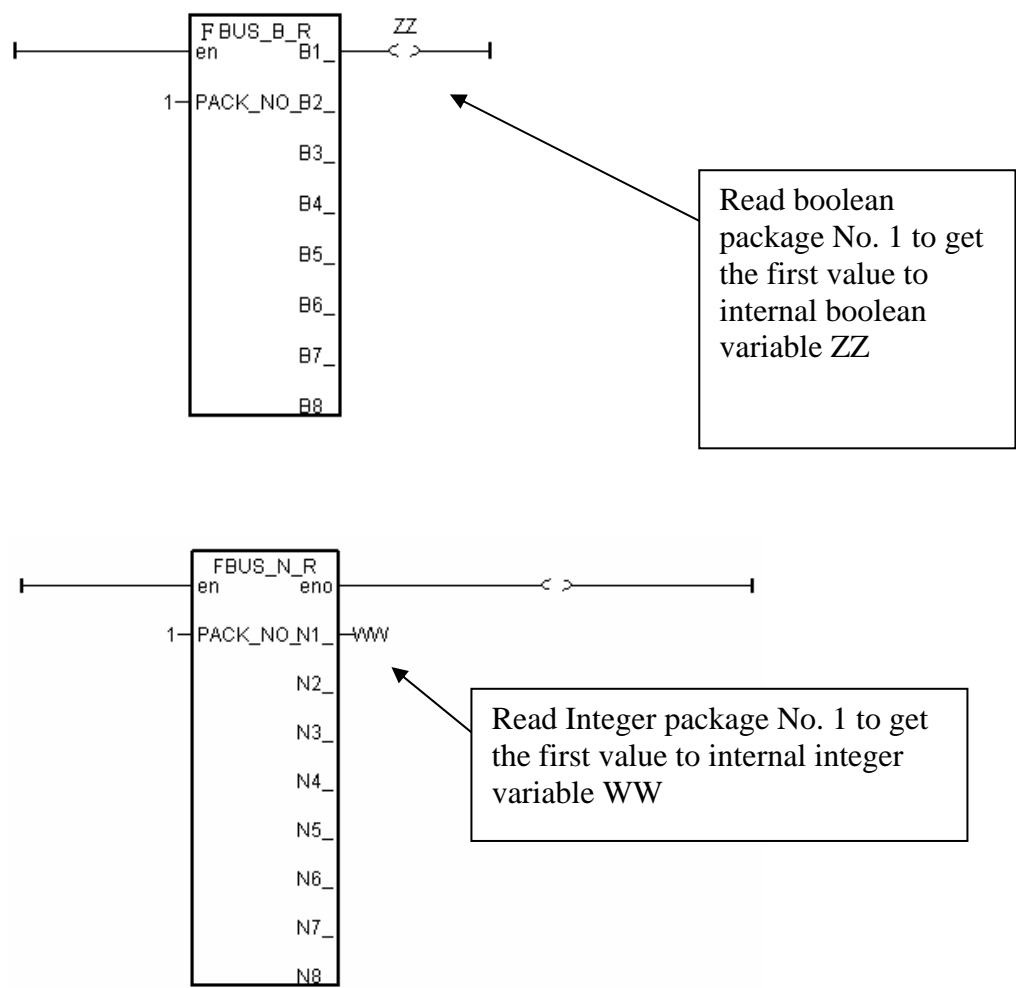
## The ISaGRAF LD Project For The SA Controller:



## The ISaGRAF LD Project For The SB Controller:



The ISaGRAF LD Project For The SC Controller:

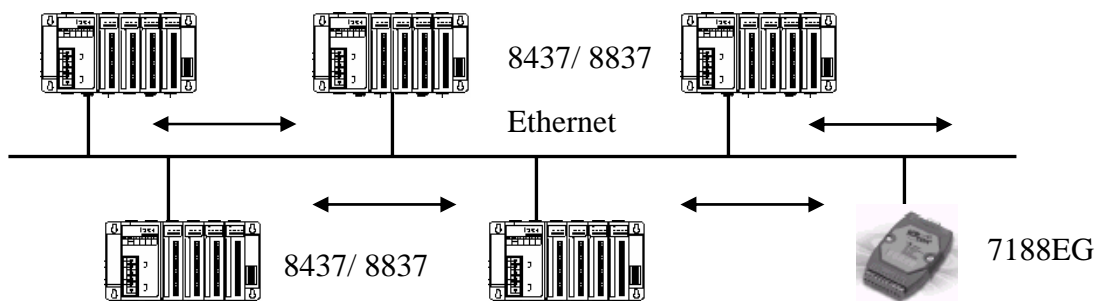


## 7.5: Programming The Ebus

Ebus is a software mechanism which allows I-8437/8837, iPAC-8447/8847, I-7188EG, uPAC-7186EG & W-8x37 / W-8x47 controllers to access data to each other through the ethernet port. Ebus is only working on the local area. That means exchanging data through a gateway is no possible.

### Important Note:

1. If the controller is W-8347 / 8747, please connect Ebus at their “LAN2” port, and please use “NS-205” or “NS-208” Ethernet switch. (refer to Appendix F to Enable LAN2)
2. The max. boolean & integer package No. of Fbus & Ebus reduce from 256 to 128 since driver version of I-8417/8817/8437/8837:2.42 , I-7188EG:1.32 & I-7188XG:1.29
3. All Ebus Controllers’s IP should be set in the same domain and their mask IP should be 255.255.255.0 . For ex, (192.168.1.3), (192.168.1.5), (192.168.1.20).



The I-8437, I-8837 controllers support Ebus since its driver version of 2.15 and the I-7188EG support Ebus since its driver version of 1.08. And W-8x37 / W-8x47 support Ebus since its driver version of 3.10. Please refer to Appendix C to make sure your I-8xx7’s controller driver version is the same or higher. You can obtain the new released driver from:

<http://www.icpdas.com/products/PAC/i-8000/isagraf.htm>

### 7.5.1: Basic Ebus Rules

The I-8437/ 8837, iPAC-8447/8847, I-7188EG, uPAC-7186EG & W-8x37 / W-8x47 Ebus driver first creates a packet of eight Boolean values to form a "Boolean package", and then creates a packet of eight 32-bit integers to form an "integer package". Both of the "Boolean packages" and "integer packages" can be distributed on the Ebus to allow the data to be exchanged from one controller to another controller.

The basic Ebus rules are similar as Fbus (refer to 7.1) as below.

**RULE #1:** Each Ebus network is identified with a “Group\_No” ranging from 1 to 10. Data is only exchangeable with controllers that are assigned with the same “Group No”.

For example, there are 5 controllers located at the same local ethernet area, named A1, A2, A3, A4, A5 respectively. A1, A2 & A3 are assigned with Ebus: Group\_No = 1 while A4 & A5 are assigned with Ebus: Group\_No = 2. Therefore, A1 can access data from A2 & A3 however can not access data from A4 & A5.

**RULE #2:** Each "Boolean package" in the same Ebus:Group\_No must have an attached identification number ranging from 1 to 128. This means that there is a maximum of 128 "Boolean packages" that can be exchanged across an Ebus:Group\_No connection.

Each "Boolean package" contains 8 Boolean values, and these Boolean values can only have the value of either "True" or "False". The Boolean values in the "Boolean package" can be assigned and exchanged with either "Internal", "Input", or "Output" Boolean variables or Boolean constants.

**RULE #3:** Each "integer package" in the same Ebus:Group\_No must have an attached identification number ranging from 1 to 128. This means that there is a maximum of 128 "integer packages" that can be exchanged across an Ebus:Group\_No connection.

Each "integer package" contains eight 32-bit integer values. The integer values can range from -2147483648 to 2147483647. The integer values in the "integer package" can be assigned and exchanged with either "Internal", "Input", or "Output" integer variables or integer constants.

**Rule #4:** Each number assigned to a "Boolean package" or an "integer package" can only be written to by one I-8437/ 8837 (or I-7188EG or W-8x37 / W-8x47) controller system across the same Ebus:Group\_No network.

Each I-8437/ 8837, I-7188EG or W-8x37 / W-8x47 controller CANNOT **write** the same identification number for either a "Boolean package" or an "integer package" across the same Ebus:Group\_No. WRITTING A PACKAGE IS NOT SHARED with the other controller across the same Ebus:Group\_No network.

In this example, there are five controllers communicating through an Ebus:Group\_No network, and the controllers are named S1, S2, S3, S4, and S5 respectively. If the S1 controller attempts to write a "Boolean package" with an ID of "1" and an "integer package" with an ID of "1" across the Ebus:Group\_No, the other four controllers CANNOT write either a "Boolean package" or an "integer package" with the same number. However, the other controllers could write a "Boolean package" with an ID of "3" and an "integer package" with an ID of "2".

There is no limitation on how many controllers can read the same number package across the same Ebus:Group\_No network. Any of the S2, S3, S4, S5 controllers can read the "Boolean package" with an ID of "1" and the "integer package" with an ID of "1" if desired.

**Rule #5:** ONLY ONE I-8437/ 8837, I-7188EG or W-8x37 / W-8x47 controller in the same Group\_No can be configured as a Ebus "Master", all the others controller in the same Group\_No MUST be configured as a Ebus "Slave".

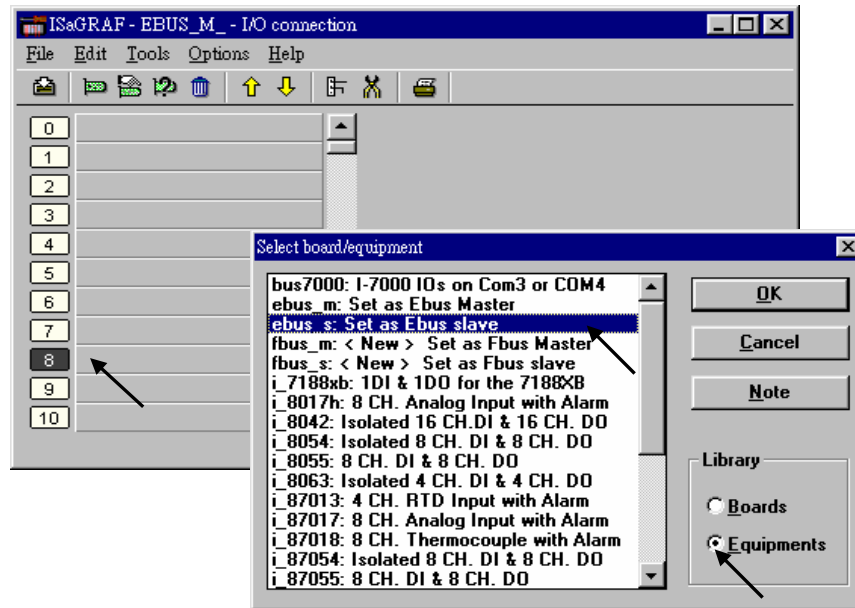
The "master" controller sends commands for how data is to be exchanged across the same Ebus:Group\_No network. If you configure more than one controller as a "master", or configure none of the controllers as a "master", NO DATA CAN BE EXCHANGED across the Ebus:Group\_No network.

**Important Note:**

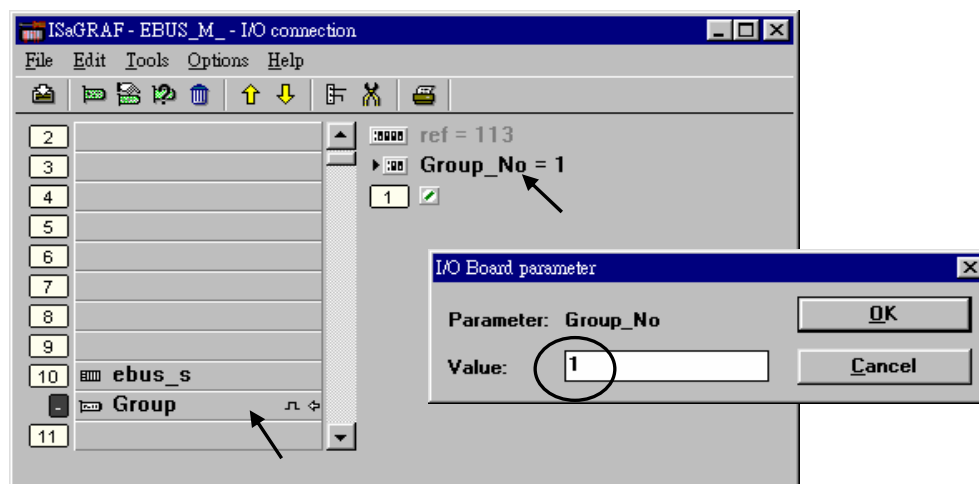
**The max. boolean & integer package No. of Fbus & Ebus reduce from 256 to 128 since driver version of I-8417/8817/8437/8837:2.42 , I-7188EG:1.31 & I-7188XG:1.28**

## 7.5.2: Configuring the Controller To Be A Ebus "Master" Or "Slave"

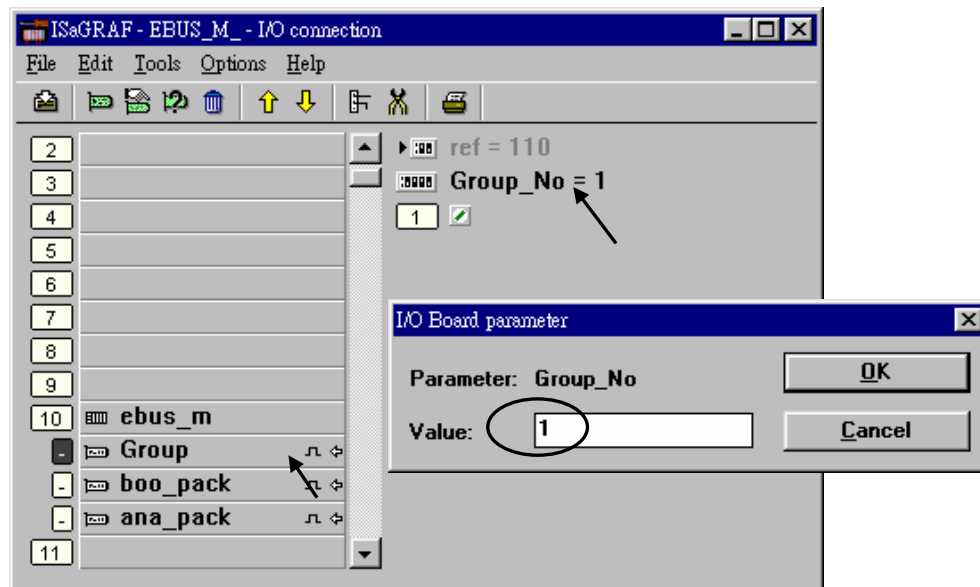
To begin configuring an I-8437/ 8837, I-7188EG or W-8x37 / W-8x47 controller system as either a Ebus master or a slave, first open up the "ISaGRAF I/O Connections" window and double click on a slot number higher than 7. The "Select Board/Equipments" window will now open, click on "Equipments", and then double click on the "Ebus\_s" selection to configure an Ebus slave, or double click on "Ebus\_m" to configure an Ebus master. Remember, **ONLY ONE** I-8437/ 8837, I-7188EG or W-8x37 / W-8x47 controller system can be the Ebus master, and you **CANNOT** configure an controller to be both a master and a slave.



If you config a controller as an Ebus slave, only one parameter needs to be set, the "Group\_No". The valid value is ranging from 1 to 10. Set to other value will become a default value , 1.

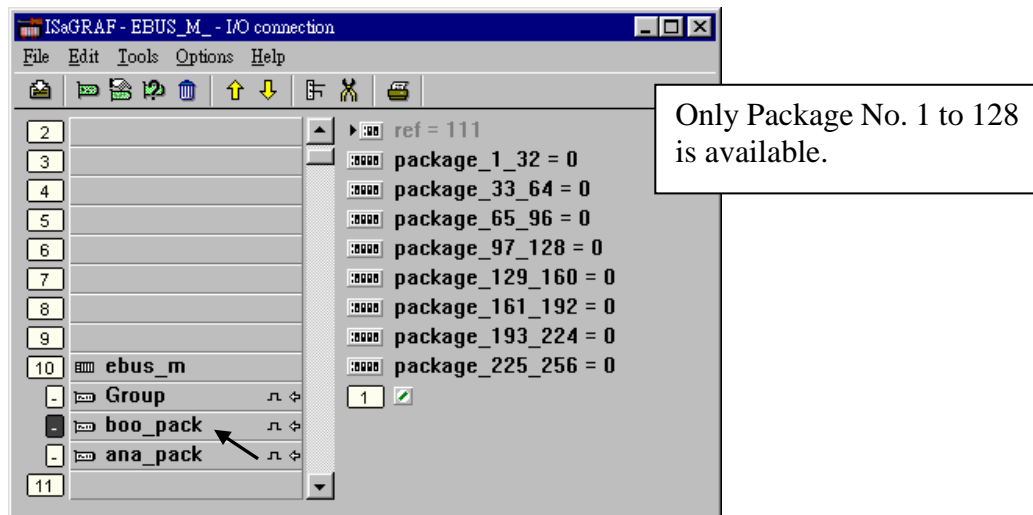


If you config a controller as an Ebus master, the parameter "Group\_No" should be set to the same as the slave. The valid value is ranging from 1 to 10. Set to other value will become a default value , 1.



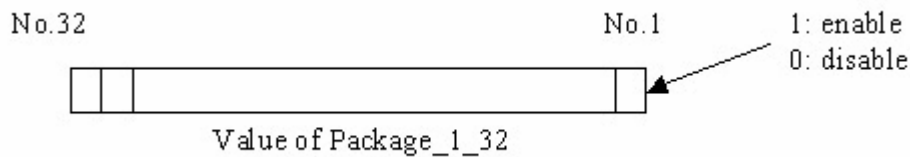
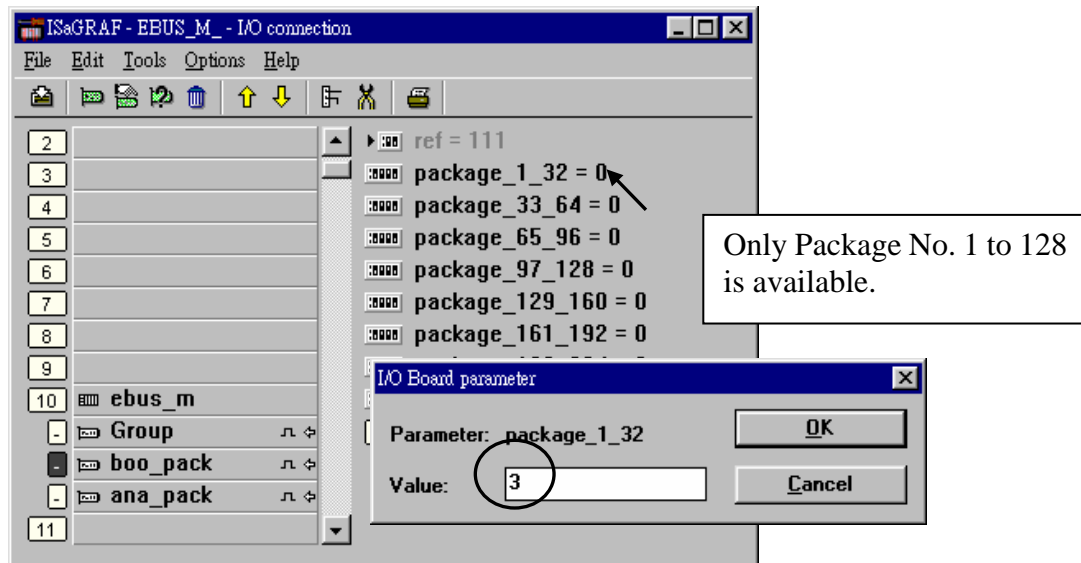
### Configuring The Ebus Master Boolean Packages:

To begin configuring the Ebus Master Boolean Packages, click on the "boo\_pack" selection from the "Ebus\_m" I/O connection.

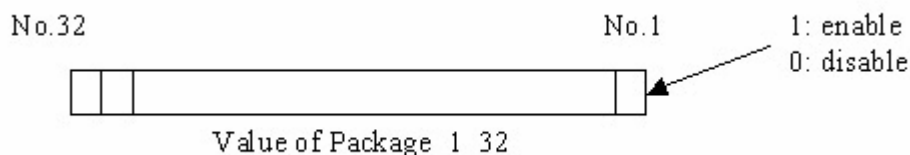
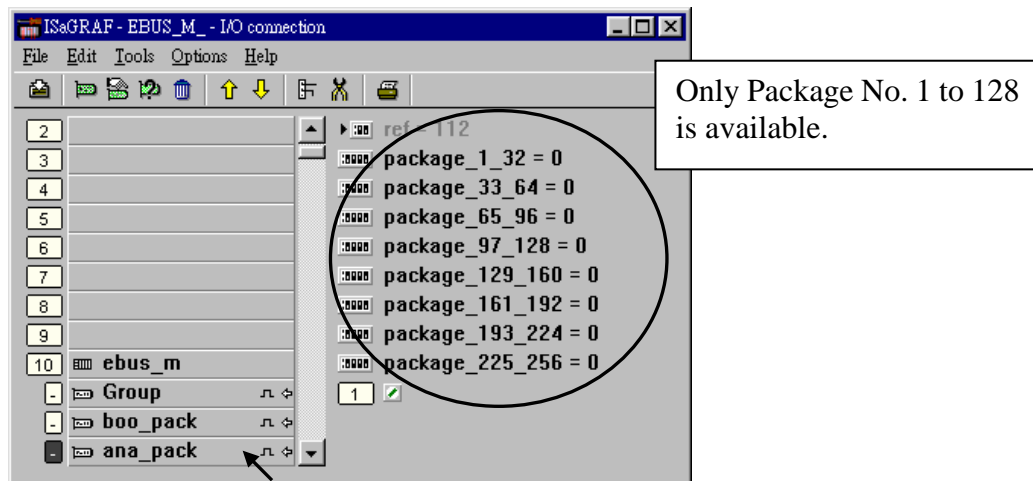


The parameter "package\_xxx\_xxx" at "Ebus\_m: boo\_pack" indicates the "Boolean package" number which is allowed to be written to or read from across the Ebus network. The parameter value is given as a 32-bit integer in **hexadecimal**.

As an example, if the "package\_1\_32" is set to "FFFFFFFF" this will enable all the packages from number 1 to number 32 to be written to or read from across the Ebus network. If the "package\_1\_32" is set to a value of "A", this will only enable the number 2 and number 4 Boolean packages to be written to or read from across the Ebus network. The more packages that are enabled on a Ebus network the slower the communication efficiency will be. **With this in mind, always remember to enable only the required number of packages that you need for your application so you will have greater communication efficiency across the Ebus network.**



The parameter "package\_xxx\_xxx" at "Ebus\_m: ana\_pack" indicates the "integer package" number which will be written to and read from on the Ebus network. The "Ebus\_m: ana\_pack" is used to read and write 32-bit integer values across the Ebus network. Each of the parameter values is expressed as 32-bit integer values in **hexadecimal**, and the same configuration rules apply as those for the "Boolean package".



### 7.5.3: Programming Ebus Packages

Before you can exchange any data across a Ebus network, you must make sure that each I-8437/ 8837, I-7188EG & W-8x37 / W-8x47 is configured as either a Ebus master "ebus\_m" (and remember, only ONE controller can be the master in the same Ebus "Group\_No") or Ebus slave "ebus\_s". Refer to Section 7.5.2 for details on how to implement these configurations.

The following Ebus function blocks can be used in a LD program to exchange data across an Ebus network.

|          |                            |
|----------|----------------------------|
| Ebus_b_r | read one boolean package.  |
| Ebus_b_w | write one boolean package. |

|          |                            |
|----------|----------------------------|
| Ebus_n_r | read one integer package.  |
| Ebus_n_w | write one integer package. |

|          |                        |
|----------|------------------------|
| Ebus_f_r | read one REAL package  |
| Ebus_f_w | write one REAL package |

**(The Integer package and REAL package use the same memory. Please DO NOT use the same package No. as Integer package and also as REAL package at the same time. Or the local fault No. 116 may happen. Please refer to Chapter 10.6)**

The below block is to get the communication status of each Boolean & Integer Package.

|          |                             |
|----------|-----------------------------|
| Ebus_sts | Get status of each Package. |
|----------|-----------------------------|

To program the Ebus\_x\_x blocks is similar to the Fbus, please refer to section 7.3 & 7.4 for detail.



---

## Chapter 8. Linking The Controller To Modbus RTU & Modbus ASCII Devices

---

Note: ICP DAS ISaGRAF controllers support Modbus Master ports in different port No. as below table.

(To use COM5 to COM14 of the Wincon-8xx7, please refer to Appendix E of the W-8xx7 “Getting Started Manual” or W-8xx7 CD-ROM:\napdos\isagraf\wincon\english\_manu\ , or visit below web site to download it.

[http://www.icpdas.com/products/PAC/i-8000/getting\\_started\\_manual.htm](http://www.icpdas.com/products/PAC/i-8000/getting_started_manual.htm) )

| (Table 1)  | I-7188EG/XG<br>uPAC-7186EG                         | I-8xx7<br>I-8437/8837-80  | iPAC-8447<br>iPAC-8847   | Wincon-8xx7<br>Wincon-8xx6  |
|--|--|---|--|---|
| Max. Modbus Master port                                    | Max. 2 ports                                       | Max. 2 ports  | Max. 4 ports   | Max.10 ports  |
| Possible port No.  | COM1<br>COM2<br><br>COM3 (in X5xx expansion board) | COM1 ,<br>COM3 , COM4 ,<br><br>COM5 ( in I-8112/8114 or i-8142/8144 ) | COM1 ,<br>COM3 , COM4 ,<br><br>COM5 to 8 ( in I-8112/8114 or i-8142/8144 ) | COM2 or COM3,<br><br>COM5 to COM14 (in I-8112/8114 or i-8142/8144 ) |
| Max. “Mbus_xxx” function blocks can be used in the program | Total 64   | Total 64  | Total 128  | Each port can use up to 256   |

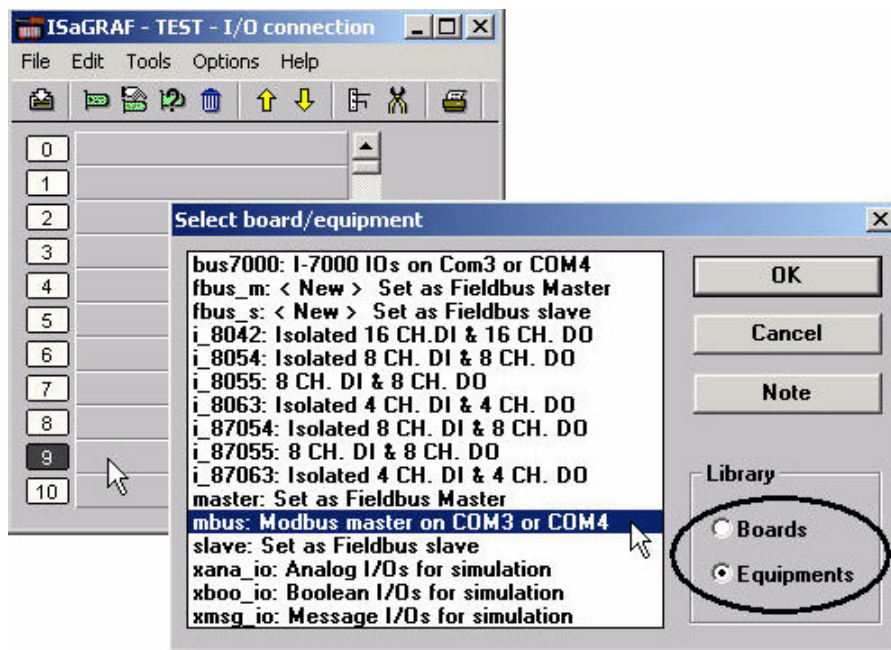
### 8.1: Configuring The Controller To Be A Modbus Master

To begin configuring an I-8xx7, I-7188EG/XG, uPAC-7186EG, iPAC-8447/8847 & W-8xx7 controller system to interface with a Modbus device, you must first configure the ISaGRAF program by linking the "Mbus" function to the ISaGRAF project. Open the "ISaGRAF I/O Connections" window and double click on a slot number higher than 7 and the "Select Board/Equipments" window will open. From the "Library", click on the "Equipments" choice, and then click on the "Mbus: Modbus Master On ..." selection, and then click on the "OK" to complete the installation.

For using multi-ports of Modbus RTU & ASCII Master, please refer to section 8.4  
Please refer to section 8.3 for Modbus ASCII Master function

#### IMPORTANT NOTE:

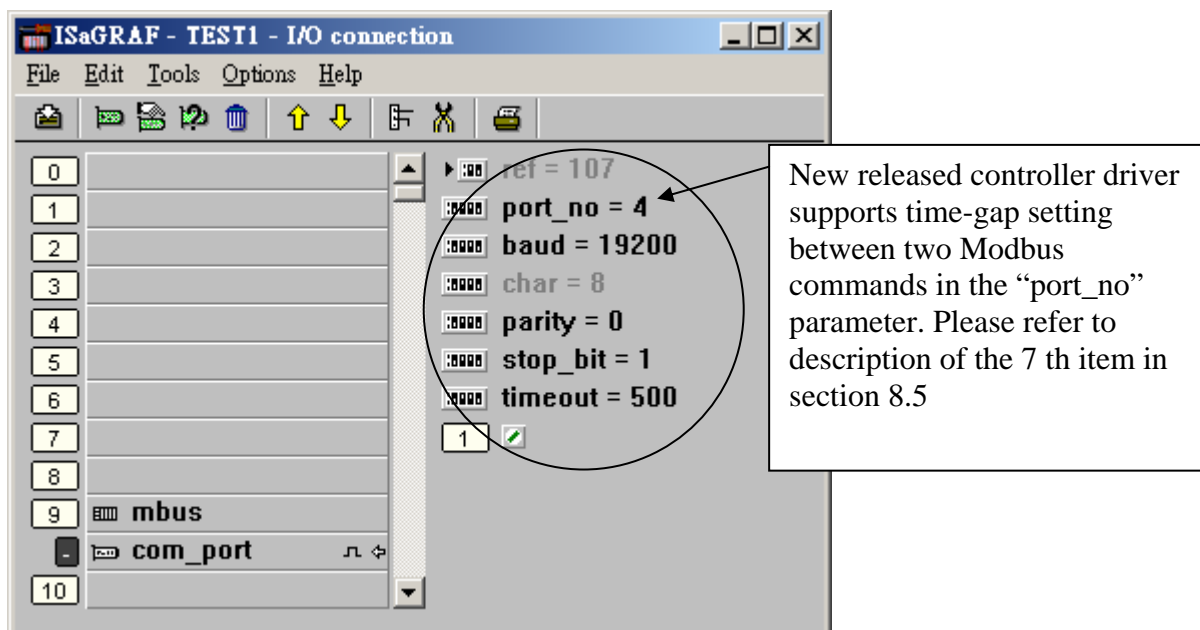
Only Max. **TWO** "Mbus" + “Mbus\_asc” complex equipment function can be linked to **ONE** I-8xx7, I-7188EG/XG controller system, however max. **TEN** for W-8xx7 controller system.



"Mbus: com\_port" Parameter

The "**port\_no**" parameter defines which COM port the Modbus devices will communicate with the controller. The "port\_no" parameter can be set as 1 to 14 (please refer to "Table 1" in the former page).

The "**baud**" parameter defines what the communications baud rate setting will be. The " baud" can be set to 2400, 4800, 9600, 19200, 38400, 57600 or 115200 baud rate. The default baud rate value is 19200 for the I-8xx7, I-7188EG/XG & W-8xx7 controller system. All controllers on the same Modbus MUST be set to the same baud rate.



**Note:**

**When setting COM1 of the I-8417/8817/8437/8837 & the I-7188EG to be a Modbus master port, please refer to Appendix C.1 – “Setting COM1 As None-Modbus Port” to disable COM1:Modbus RTU port.**

**If W-8xx7’s COM2 is set as Modbus RTU Slave port, please disable it if using it as a Modbus master port. Please refer to W-8xx7’s “Getting Started” Manual.**

The "**parity**" parameter defines what the communications parity setting will be. Setting the " parity" parameter to a value of "0" sets the parity to "none", a value of "1" sets the parity to even, and a value of "2" sets the parity to odd.

The "**stop\_bit**" parameter defines the number of stop bits will be used in the Modbus communications. If the "stop\_bit" parameter is set to "1", this equals 1 stop bit, and a value of "2" equals 2 stop bits.

The "**timeout**" parameter defines the allowed time to wait for response from remote device, unit is ms, min value is 50, max value is 5000. setting out of range ( > 5000, < 50 ) will set to value of 250 ms. For ex, 200 is 200ms, 1000 is 1 sec, setting 30 will become 250 ms, setting 7000 will become 250ms.

**Important:**

**The “timeout” setting is very important. If the response time of the connected Modbus RTU device is larger than this value, it will become Timeout – communication will fail. To solve this problem with such a slowly response device, please set a larger “timeout” value.**

## 8.2: Programming A Modbus RTU Master

---

The following function blocks can be used to pass data through the Modbus protocol in an LD program.

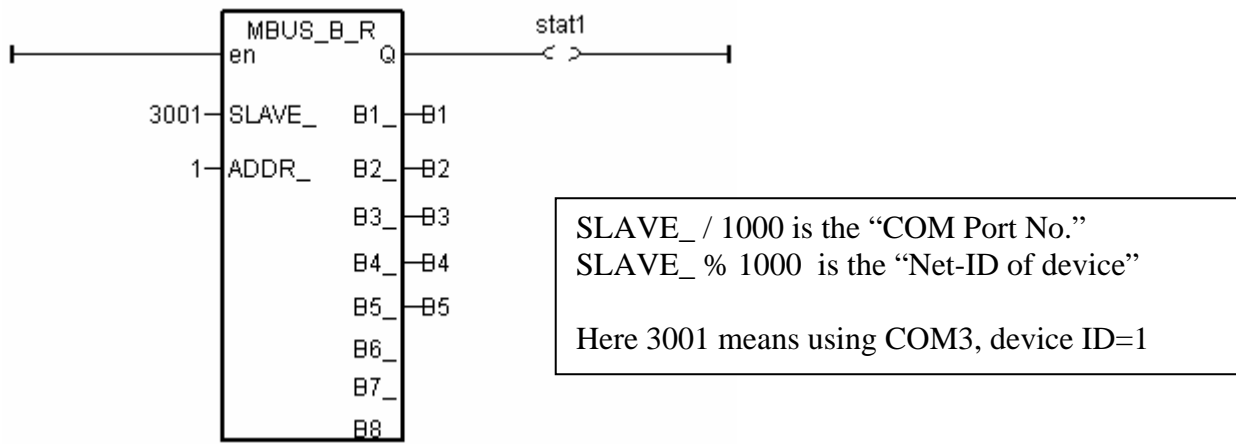
|          |   |
|----------|---|
| Mbus_R   | Read max. 12 word-value (-32768 ~ +32767) using Modbus function code 3 or 4<br>Read max.192 bit-value using Modbus function code 1 or 2 |
| Mbus_R1  | Same as Mbus_R but with one extra setting – Period.<br>Read words or bits with a specified period time (unit is second)                 |
| Mbus_N_R | Read 8 word-value (-32768 ~ +32767) using Modbus function code 3  |
| Mbus_NR1 | Same as Mbus_N_R but with one extra setting – Period.<br>Read 8 words with a specified period time (unit is second)                     |
| MBUS_B_R | Read 8 bit-value using Modbus function code 1   |
| MBUS_BR1 | Same as Mbus_B_R but with one extra setting – Period.<br>Read 8 bits with a specified period time (unit is second)                      |
| MBUS_N_W | Write max. 4 word-value (-32768 ~ +32767) using Modbus function code 6 or 16  |
| MBUS_B_W | Write max. 4 bit-value using Modbus function code 5 or 15   |
| MBUS_WB  | Write max. 16 bit-value using Modbus function code 15   |

**NOTE:**

The maximum number of each "Mbus\_x\_x" function block that can be used with one I-8xx7 & I-7188EG/XG controller system is 64, while 256 for one port of W-8xx7 (W-8xx7 can have 10 ports supporting Modbus Master).

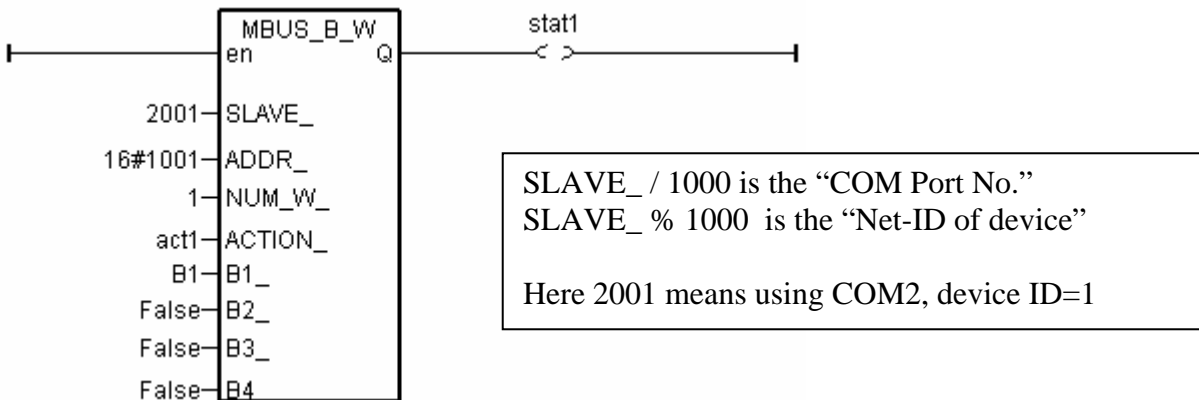
## Modbus Example Function: "Mbus\_b\_r"

The following example the "Mbus\_b\_r" function block is reading five (5) bits from a slave Modbus device with a NET ID address of 1, with the Modbus address starting from 1 and using controller's COM3 port. In this example the results of "B1" contains the value of the Modbus address 1, "B2" equals the value of Modbus address 2, etc. "B5" equals the value of the Modbus address 5. If device is connected Ok, "stat1" will be TRUE.



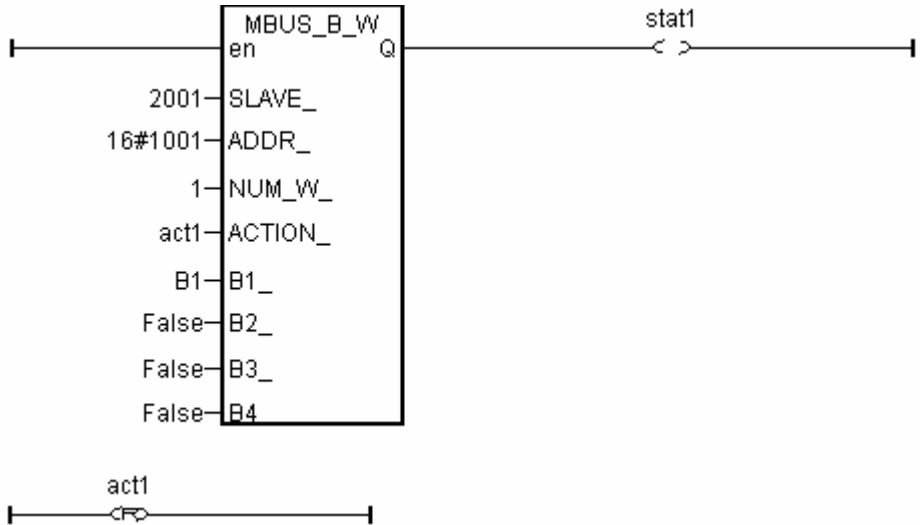
## Modbus Example Function : "Mbus\_b\_w"

The following example of the "Mbus\_b\_w" function block is writing one (1) bit to a slave Modbus device with a NET ID address of 1. The "Mbus\_b\_w" function will only write this one bit when the "ACTION\_" line is true. In the example below the resulting value of "B1" is written to the Modbus address 16#1001 (or 4097) of that Modbus device when the "ACTION\_" line is true.

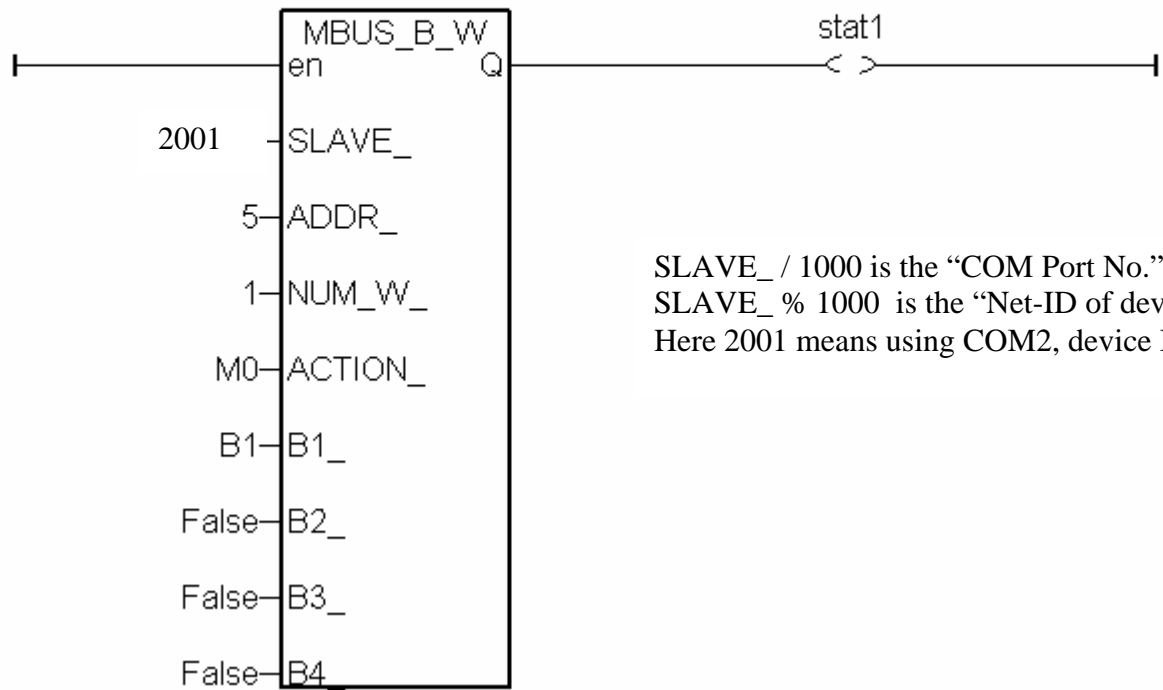
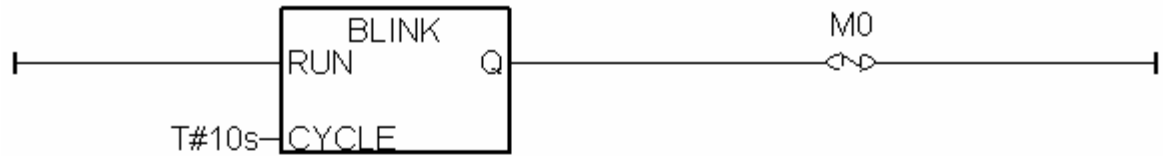


The value of "Stat1" is connected to the output coil and if the operation is successful "Stat1" will be true, otherwise the value of "Stat1" will be false.

If the “ACTION\_” input keeps at the status of TRUE, it will continue to write this "B1" many times to that Modbus device until it is reset to FALSE. If you just want to write one time, you can write a LD program similar as the following. The “act1” is declared as an internal Boolean variable. If set “act1” to TRUE, the below “MBUS\_B\_W” will write once and immediately reset “act1” to become False.



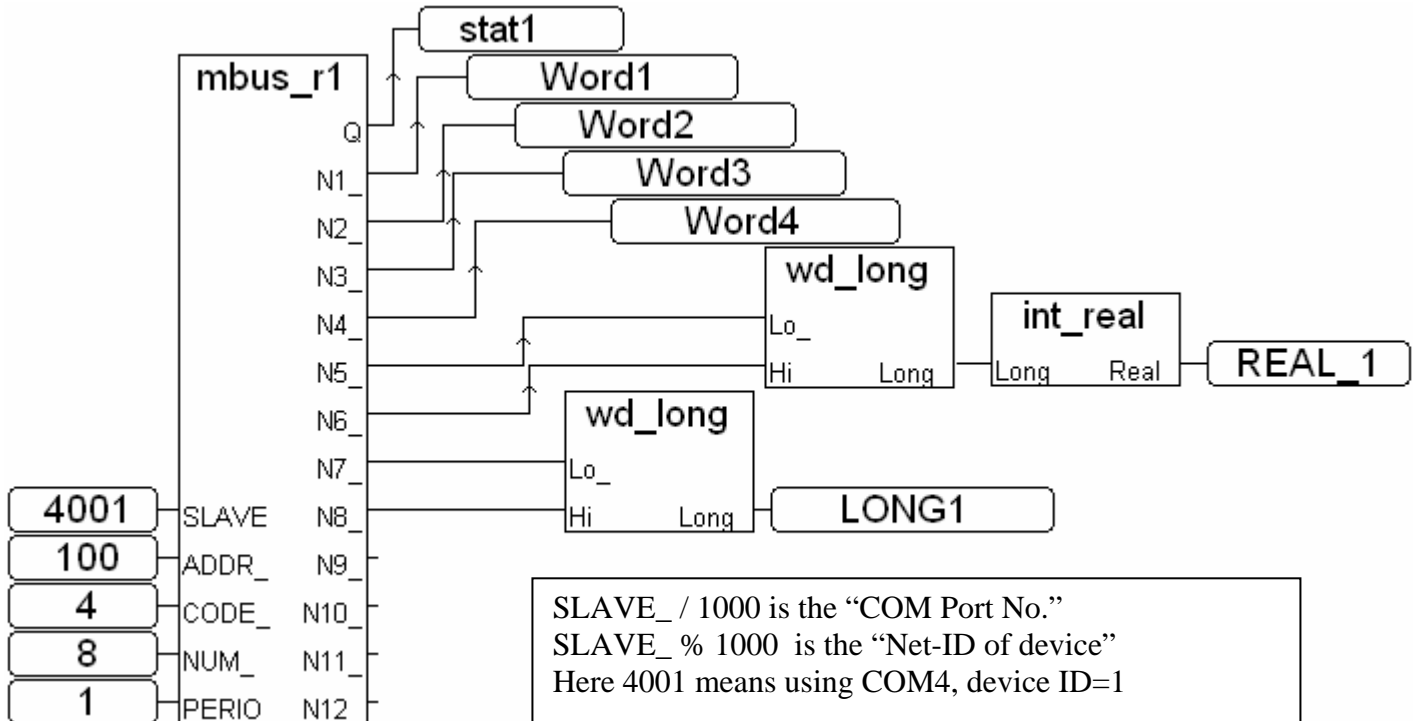
To write Modbus command periodic, user can use below similar code. (Here write once every 10 second)



SLAVE\_ / 1000 is the “COM Port No.”  
 SLAVE\_ % 1000 is the “Net-ID of device”  
 Here 2001 means using COM2, device ID=1

### Modbus Example Function: "Mbus\_r1"

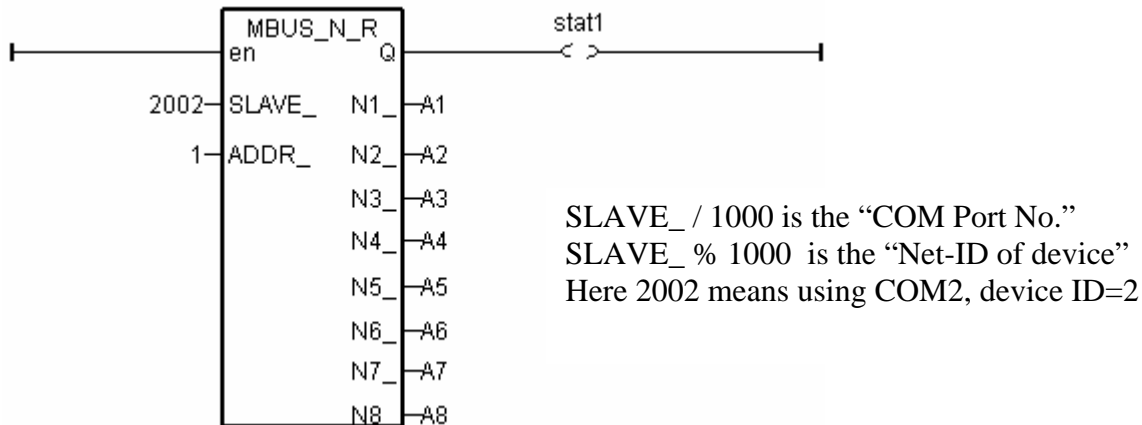
Below example use Modbus function code=4, device NET-ID=1 to read 8 word values starting at device's Modbus address No.=100 every second via controller's COM4. The first four Word values are stored in "Word1" to "Word4" variables. The fifth and sixth word value are converted to become a REAL value stored in "REAL\_1" variable. The 7th and 8th word values are converted to become a long integer value (32-bit signed integer) stored in "LONG1" variable.



### Modbus Example Function : "Mbus\_n\_r"

The following example the "Mbus\_n\_r" function block is reading eight (8) words from a slave Modbus device with a NET ID address of 2 (the Modbus address starts from 1). In this example the results of "A1" contains the value of the Modbus address 1, "A2" equals the value of Modbus address 2, etc., through "A8" which equals the value of the Modbus address 8.

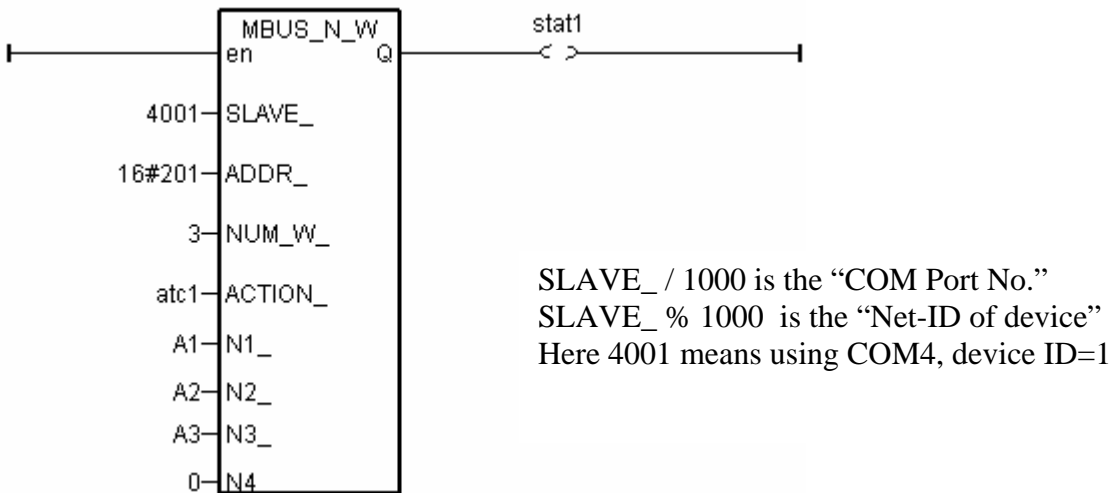
The value of "Stat1" is connected to the output coil and if the operation is successful "Stat1" will be true, otherwise the value of "Stat1" will be false.



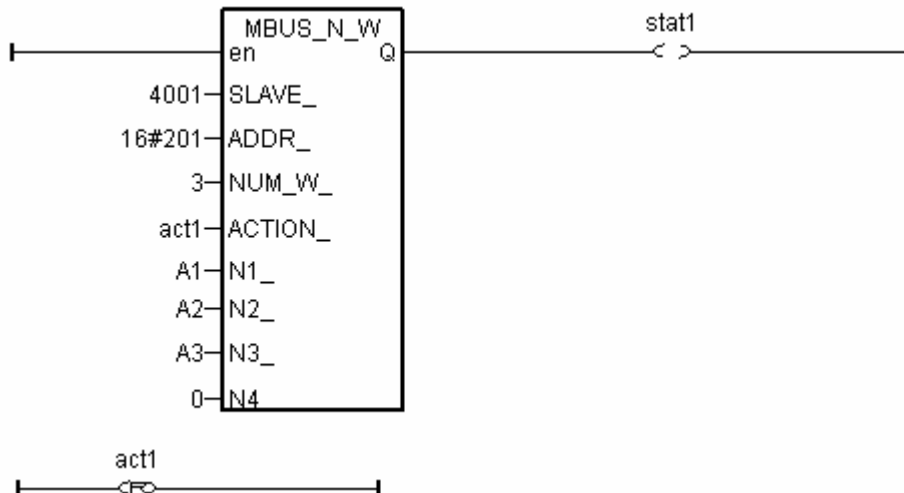
### Modbus Example Function : "Mbus\_n\_w"

Below example of the "Mbus\_n\_w" function block is writing three (3) words to a slave Modbus device with a NET ID address of 1, and the Modbus address is starting from 16#201 (513). The "Mbus\_n\_w" function will only write when the "ACTION\_" line is true. In this example when the "ACT1" line is True, the value of A1 will be written to the value of Modbus address 16#201 of that Modbus device, the value of A2 will be written to the value of Modbus address 16#202, and A3 will be written to the value of Modbus address 16#203.

The value of "Stat1" is connected to the output coil and if the operation is successful "Stat1" will be true, otherwise the value of "Stat1" will be false.



If the "ACTION\_" input keeps at the status of TRUE, it will continue to write these "A1" through "A3" many times to that Modbus device until it is reset to FALSE. If you just want to write one time, you can write a LD program similar as the following. The "act1" is declared as an internal Boolean variable. If setting "act1" to TRUE, it write only once.



More information about Modbus Master is at

[www.icpdas.com](http://www.icpdas.com) – FAQ – Software – ISaGRAF – FAQ 027 , 028 , 045 , 046 and 047 .

## 8.3: Modbus ASCII Master

I-7188EG : driver ver. 2.07 or later (better to be ver. 2.17 or later)  
I-7188XG : driver ver. 2.05 or later (better to be ver. 2.15 or later)  
I-8xx7 : driver ver. 3.08 or later (better to be ver. 3.19 or later)  
W-8xx7 : driver ver. 3.20 or later (better to be ver. 3.36 or later)

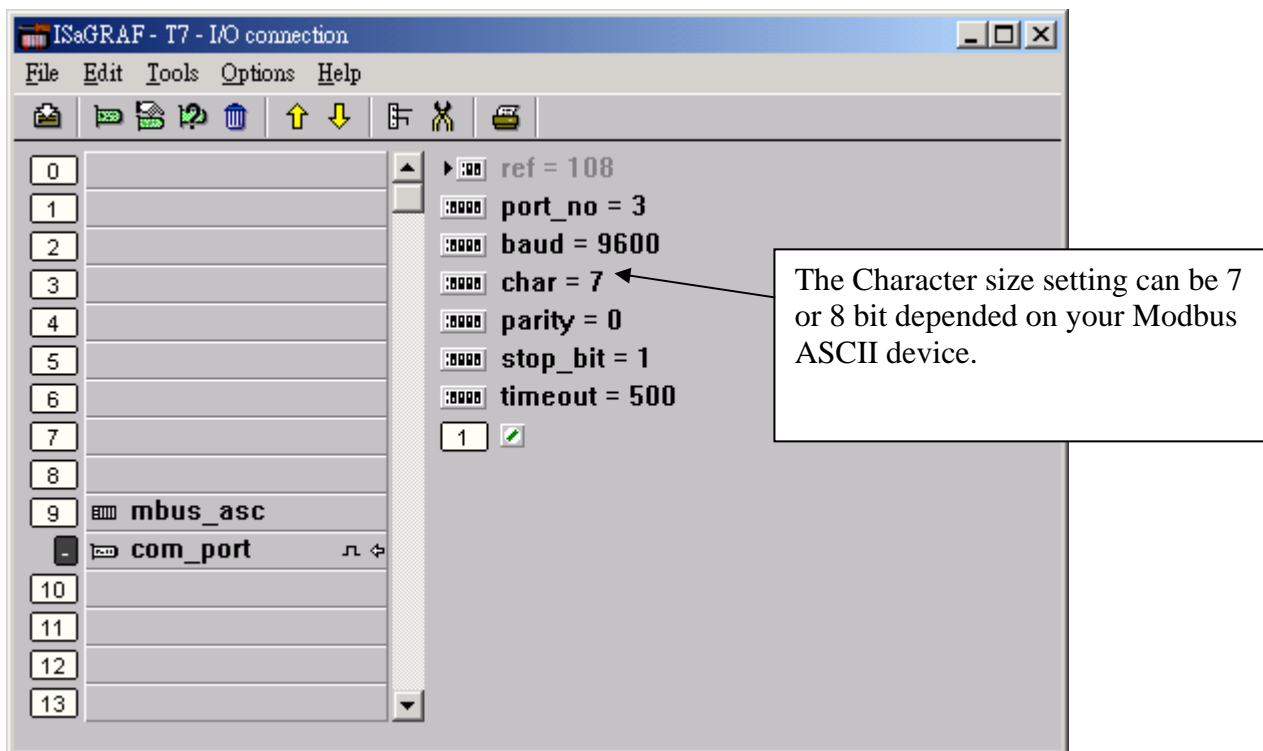
I-7188EG/XG & I-8417/8817/8437/8837 can support two Modbus Master port. (can be Modbus RTU Master or Modbus ASCII Master). Please refer to Table 1 listed in 1<sup>st</sup> page of Chapter 8.

Wincon-8xx7 support max. 10 ports of Modbus Master. (can be Modbus RTU Master or Modbus ASCII Master), please refer to ISaGRAF section 8.4

To use Modbus ASCII Master, please connect “mbus\_asc” in the IsaGRAF I/O connection windows as below.

If you can not find “mbus\_asc” in your ISaGRAF, please visit <http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> to download “ICP DAS Utilities For ISaGRAF.zip”.

For new driver please click “New Driver for I-8xx7, 7188EG/XG & W-8x37”

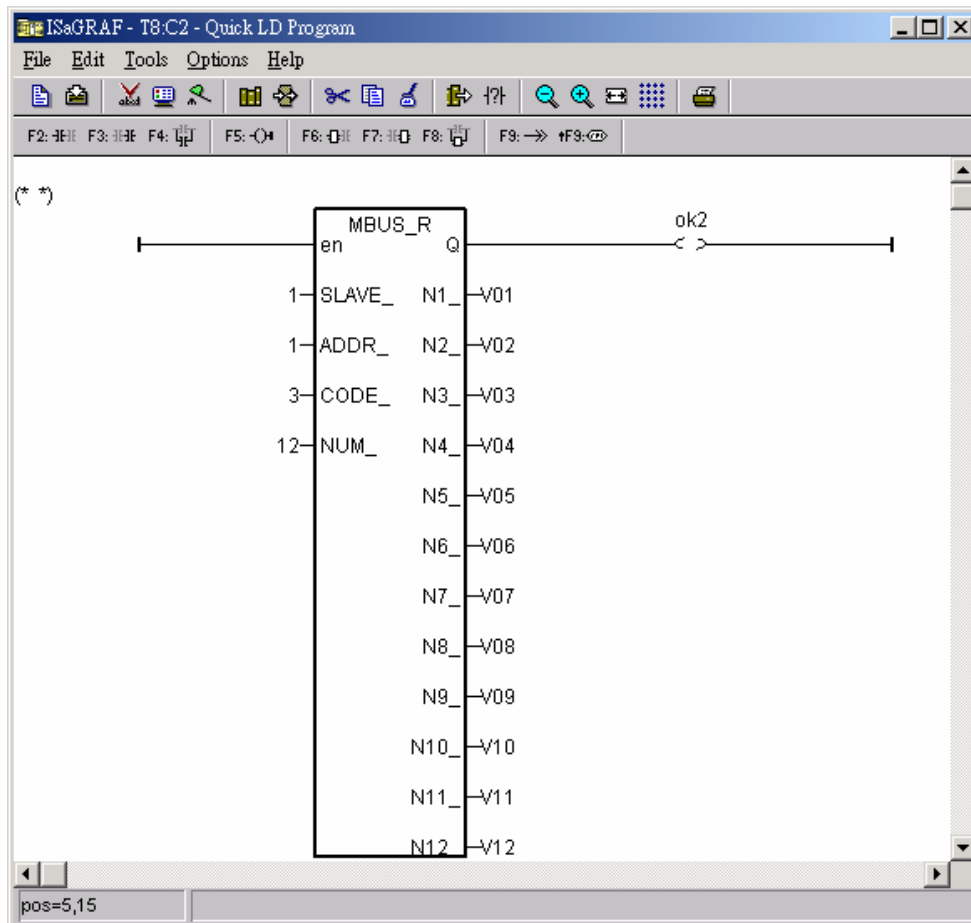




Then using below function blocks in your ISaGRAF Ladder or Function block program.

|          |   |
|----------|---|
| Mbus_R   | Read max. 12 word-value (-32768 ~ +32767) using Modbus function code 3 or 4<br>Read max.192 bit-value using Modbus function code 1 or 2 |
| Mbus_R1  | Same as Mbus_R but with one extra setting – Period.<br>Read words or bits with a specified period time (unit is second)                 |
| Mbus_N_R | Read 8 word-value (-32768 ~ +32767) using Modbus function code 3  |
| Mbus_NR1 | Same as Mbus_N_R but with one extra setting – Period.<br>Read 8 words with a specified period time (unit is second)                     |
| MBUS_B_R | Read 8 bit-value using Modbus function code 1   |
| MBUS_BR1 | Same as Mbus_B_R but with one extra setting – Period.<br>Read 8 bits with a specified period time (unit is second)                      |
| MBUS_N_W | Write max. 4 word-value (-32768 ~ +32767) using Modbus function code 6 or 16  |
| MBUS_B_W | Write max. 4 bit-value using Modbus function code 5 or 15   |
| MBUS_WB  | Write max. 16 bit-value using Modbus function code 15   |

For example:



## 8.4: Multi-Ports Modbus RTU/ASCII Master

I-7188EG : driver ver. 2.07 or later (better to be ver. 2.17 or later)  
I-7188XG : driver ver. 2.05 or later (better to be ver. 2.15 or later)  
I-8xx7 : driver ver. 3.08 or later (better to be ver. 3.19 or later)  
W-8xx7 : driver ver. 3.20 or later (better to be ver. 3.42 or later)

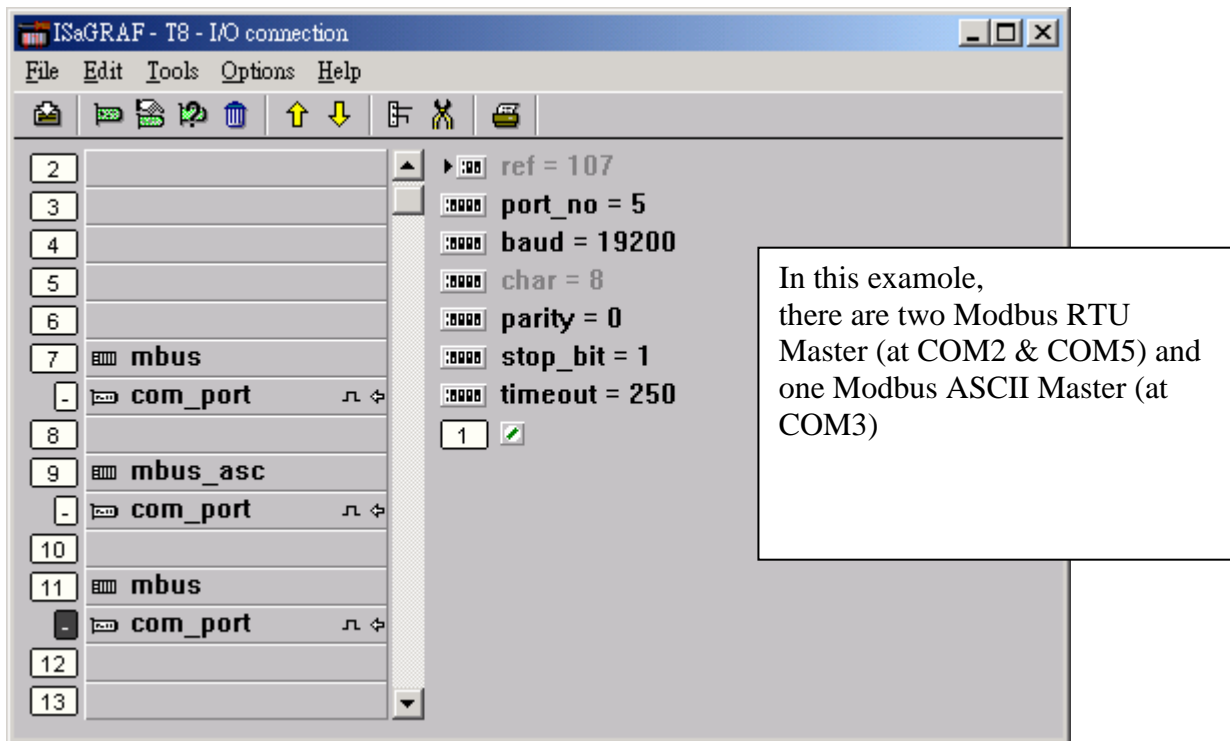
I-7188EG/XG & I-8417/8817/8437/8837 can support two Modbus Master ports. (can be Modbus RTU Master or Modbus ASCII Master). Please refer to Table 1 listed in 1<sup>st</sup> page of Chapter 8

Wincon-8xx7 support max. 10 ports of Modbus Master. (can be Modbus RTU Master or Modbus ASCII Master) . Please refer to Table 1 listed in 1<sup>st</sup> page of Chapter 8.

To use multi-ports of Modbus RTU or ASCII Master in Wincon, please connect “mbus” or “mbus\_asc” in the ISaGRAF I/O connection windows as below.

If you can not find “mbus\_asc” in your ISaGRAF, please visit <http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> to download “ICP DAS Utilities For ISaGRAF.zip”.

For new driver please click “New Driver & ...”



Then using below function blocks in your ISaGRAF Ladder or Function block program.

**Note:**

1. The “SLAVE\_” setting in the below function blocks means Port No. & slave No.

Port No. = SLAVE / 1000

slave No. = SLAVE mod 1000

For example,

if SLAVE = 2001, Port No. = COM2 & slave No. = 1

if SLAVE = 9002, Port No. = COM9 & slave No. = 2

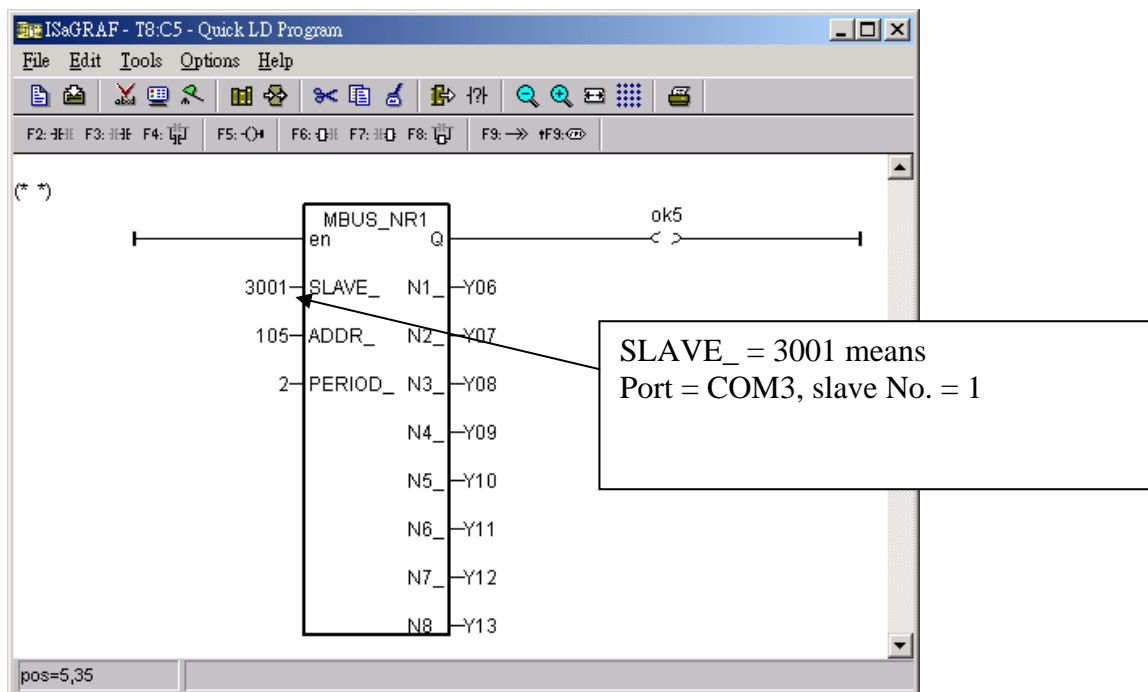
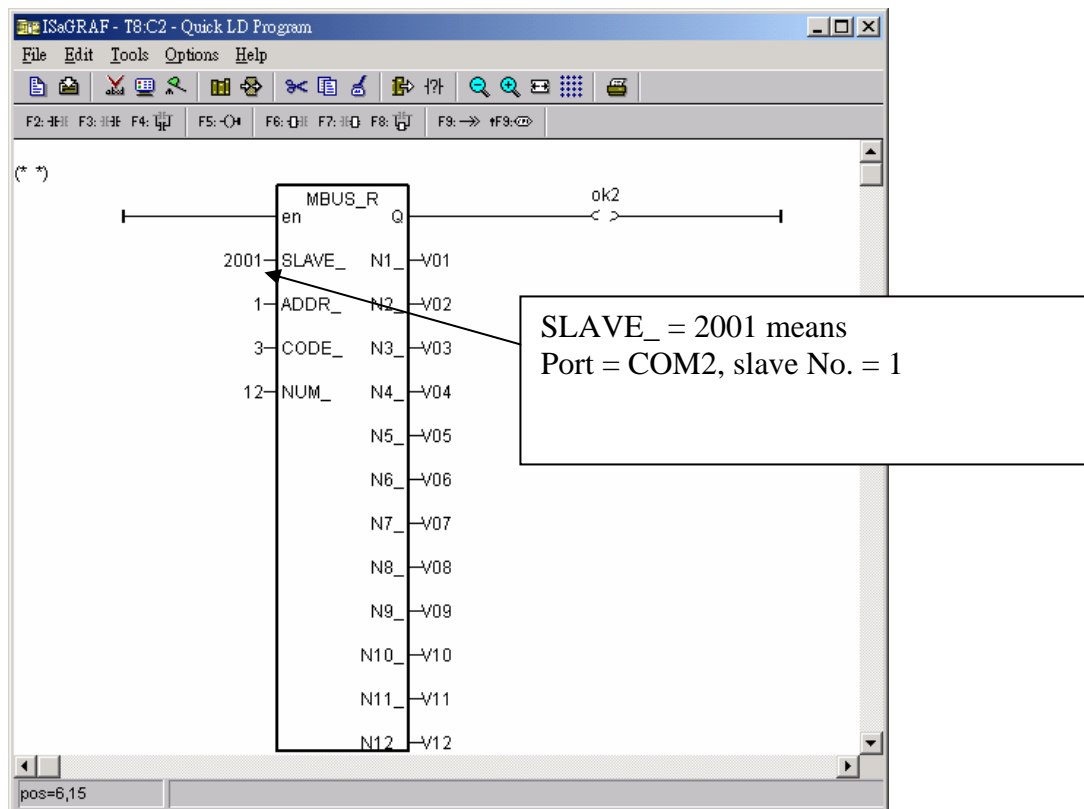
if SLAVE = 10002, Port No. = COM10 (MSP1:) & slave No. = 2;

2. If you are using I-8112/8114/8142/8144 in Wincon, please plug these boards in Wincon first and then run “Start” – “Programs” - “Wincon utility” - “Com” - “New Card Wizard” - “Slot Scan” , then click on “Save New Module” to properly set each extra COM port.

|          |   |
|----------|---|
| Mbus_R   | Read max. 12 word-value (-32768 ~ +32767) using Modbus function code 3 or 4<br>Read max.192 bit-value using Modbus function code 1 or 2 |
| Mbus_R1  | Same as Mbus_R but with one extra setting – Period.<br>Read words or bits with a specified period time (unit is second)                 |
| Mbus_N_R | Read 8 word-value (-32768 ~ +32767) using Modbus function code 3  |
| Mbus_NR1 | Same as Mbus_N_R but with one extra setting – Period.<br>Read 8 words with a specified period time (unit is second)                     |
| MBUS_B_R | Read 8 bit-value using Modbus function code 1   |
| MBUS_BR1 | Same as Mbus_B_R but with one extra setting – Period.<br>Read 8 bits with a specified period time (unit is second)                      |
| MBUS_N_W | Write max. 4 word-value (-32768 ~ +32767) using Modbus function code 6 or 16  |
| MBUS_B_W | Write max. 4 bit-value using Modbus function code 5 or 15   |
| MBUS_WB  | Write max. 16 bit-value using Modbus function code 15   |

Then follow below Ladder or Function block program.

For example:



## 8.5: Why can not connect Modbus slave device correctly ?

The i-8417/8817/8437/8837 , I-7188EG/XG , Wincon-8xx7 supports the “Delay\_time” (time gap) setting between two Modbus commands since below driver version.

|                         |                               |
|-------------------------|-------------------------------|
| I-7188EG + X607 / X608: | version 2.20 or later version |
| I-7188XG + X607 / X608: | version 2.18 or later version |
| I-8xx7+ S256 / S512 :   | version 3.22 or later version |
| Wincom-8xx7:            | version 3.42 or later version |

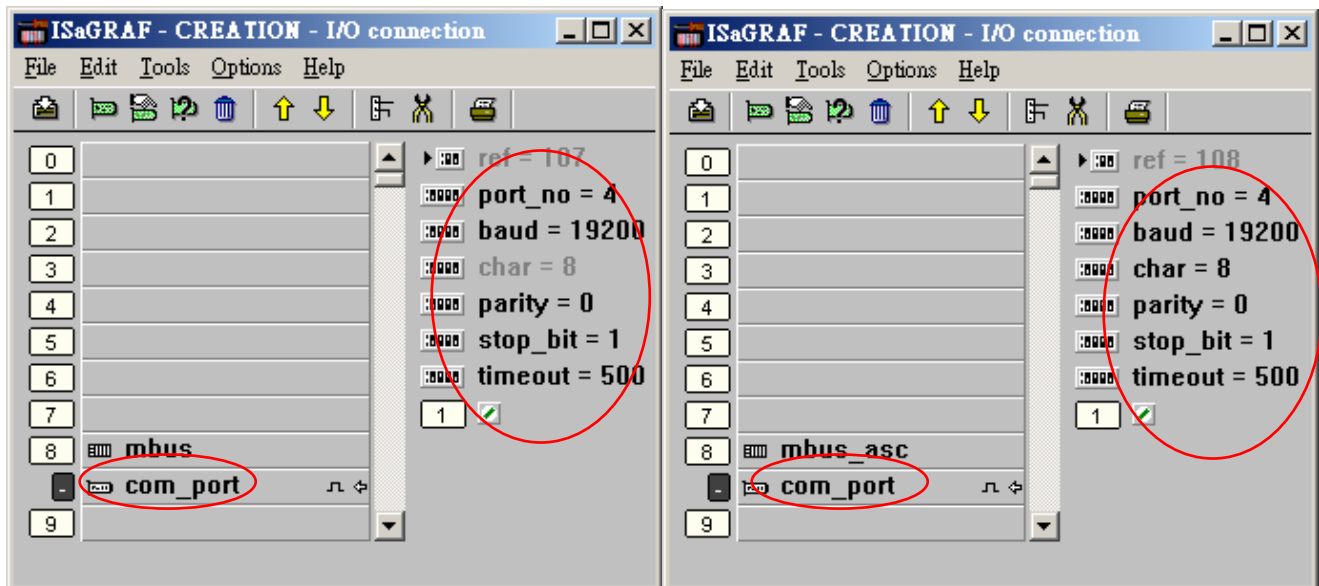
New released driver can be download at <http://www.icpdas.com/products/PAC/i-8000/isagraf-link.htm>

Please refer to “ISaGRAF User’s Manual” – Chapter 8 for more information about “Modbus Master” function. It can be found in the below position.

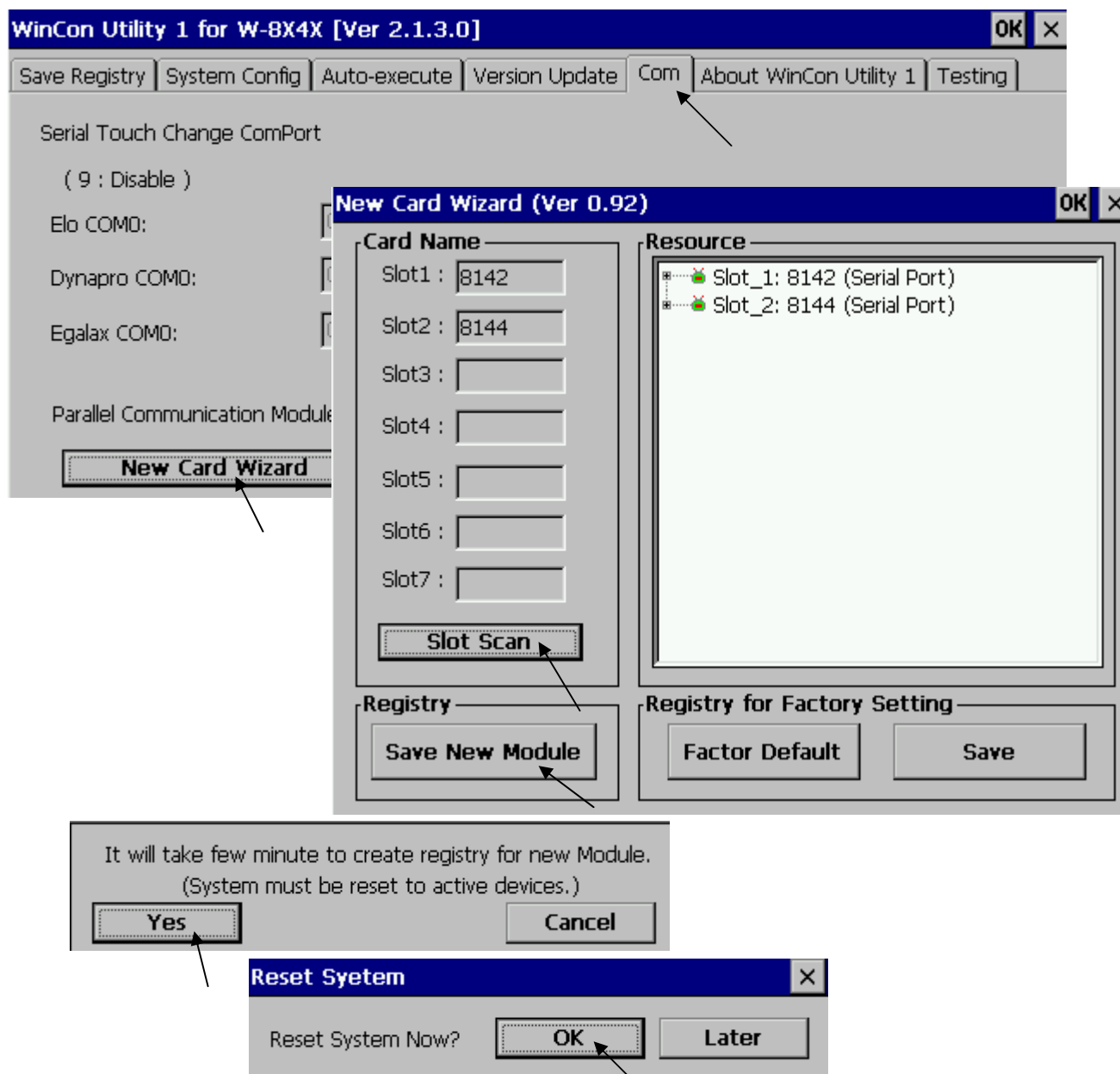
i-8000 CD-ROM:\napdos\isagraf\8000\English\_manu\ “User\_Manual\_I\_8xx7.pdf” or  
W-8xx7 CD-ROM:\napdos\isagraf\wincon\ English\_manu\ “User\_Manual\_I\_8xx7.pdf” or  
[http://www.icpdas.com/products/PAC/i-8000/getting\\_started\\_manual.htm](http://www.icpdas.com/products/PAC/i-8000/getting_started_manual.htm)

The normal reasons which make the ISaGRAF Controller can not connect the Modbus Slave devices well are listed as below.

1. The connected device doesn’t support the Modbus RTU Slave or Modbus ASCII Slave protocols.
2. The ISaGRAF Controller doesn’t run the ISaGRAF program which setup the “mbus” or “mbus\_asc” in the IO connection windows (Please refer to the “ISaGRAF User’s manual” – Chapter 8).
3. The “baud” , “parity” , “stop\_bit” setting in the “mbus” and “mbus\_asc” doesn’t fit the same setting with the Modbus Slave devices. Or the com port wiring is not correct.



4. The “timeout” setting in the “mbus” and “mbus\_asc” is too small for those connected Modbus slave devices. Its unit is ms (0.001 second) . For example, if the slave device will reply in 200 ms later after it received the requested command, then please set “timeout” as twice of this 200 ms. That is 400 ms. If it will reply in 350 ms later, then please set “timeout” as 700. And so on.
5. The “port\_no” setting is not correct in the “mbus” and “mbus\_asc”. For example, if using W-8xx7 and i-8xx7 ‘s COM3 as Modbus Master port, please set “port\_no” as 3. If using W-8xx7 ‘s COM5 to COM14 as Modbus Mastr port, please power off Wincon first. Then plug the i-8112 / 8114 / 8142 / 8144 expansion card in the W-8xx7 ‘s slot 1 to 4, then power On Wincon, run Wincon-utility > Com > New-Card-Wizard > Slot-Scan > Save-New-Module once. It will setup the serial expansion ports and reboot Wincon once, then COM5 to COM14 should work.

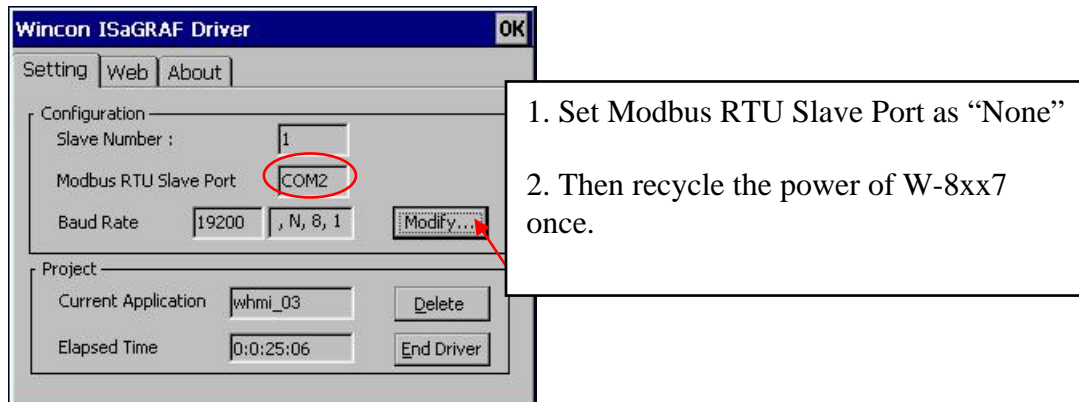


If the controller is i-8xx7, only two of its COM1 , COM3, COM4 and COM5 ports can be used (i-8xx7 ‘s COM5 resides at the i-8112/8114/8142/8144 expansion card) . While only two ports of the i-7188EG ‘s

COM1 , COM2 and COM3 can be used. And only two ports of the i-7188XG 's COM2 and COM3 can be used. (i-7188EG/XG's COM3 resides at the X-5xx expansion card)

6. The set "port\_no" is occupied by other functions. For ex, if using "bus7000b" at the same port No. or using i-8xx7 or i-7188EG 's COM1 as Modbus RTU slave port. Or using Wincon-8xx7 's COM2 or COM3 as Modbus RTU slave port.

**Please follow below steps to disable the Wincon 's COM2 / COM3 Modbus RTU slave setting.**



**Please follow below steps to disable the i-8xx7 's COM1 Modbus RTU Slave setting.**

- Create a "8000" folder in your PC 's hard drive (for ex, "c:\8000".)
- Copy CD:\Napedos\ISaGRAF\8000\Driver\...\7188xw.exe, 7188xw.inito this "8000" folder.
- Run this \8000\7188xw.exe. a "7188 xw" window will show up.
- Connect PC's COM1 or COM2 to i-8xx7 's COM1 by a RS-232 cable.  
If your PC is using other COM port Number, please modify the "7188xw.ini" and then run "7188xw.exe" again.

|                        |         |    |    |    |
|------------------------|---------|----|----|----|
| C1                     | B115200 | P0 | D8 | S1 |
| F                      |         |    |    |    |
| Xautoexec.bat Xisa.exe |         |    |    |    |
| w25                    |         |    |    |    |

➡

|                        |         |    |    |    |
|------------------------|---------|----|----|----|
| C5                     | B115200 | P0 | D8 | S1 |
| F                      |         |    |    |    |
| Xautoexec.bat Xisa.exe |         |    |    |    |
| w25                    |         |    |    |    |

- Power off the I-8437 / 8837 , short its "INIT" pin to " INIT COM". (shorting INIT to GND for 7188EG). Then power it On again.
- If the connection is fine, the 7188xw.exe will display something like "i-8000>".
- Key-in "isa \*f=1" to free this COM1 (Free COM1 as non-Modbus-Slave port)

```
i-8000>
ICP_DAS MiniOS7 for I-8000 Ver. 2.00 build 002, Apr 08 2005 17:06:02
SRAM:512K, FLASH MEMORY:512K
[CPU=Am188ES]
Serial number= 09 63 4A 60 03 00 00 76

i-8000>isa *f=1

Com1 is free now
```

- Press at ALT\_X to exit the "7188xw" window, or the PC 's COM1 or COM2 will be blocked always.
- Remove the connection between "INIT" and "INIT COM" .Recycle the I-8437 /8837 once.

**Please follow below steps to disable the i-7188EG 's COM1 Modbus RTU Slave setting.**

The steps are the similar as the steps in the former page except step “G” .

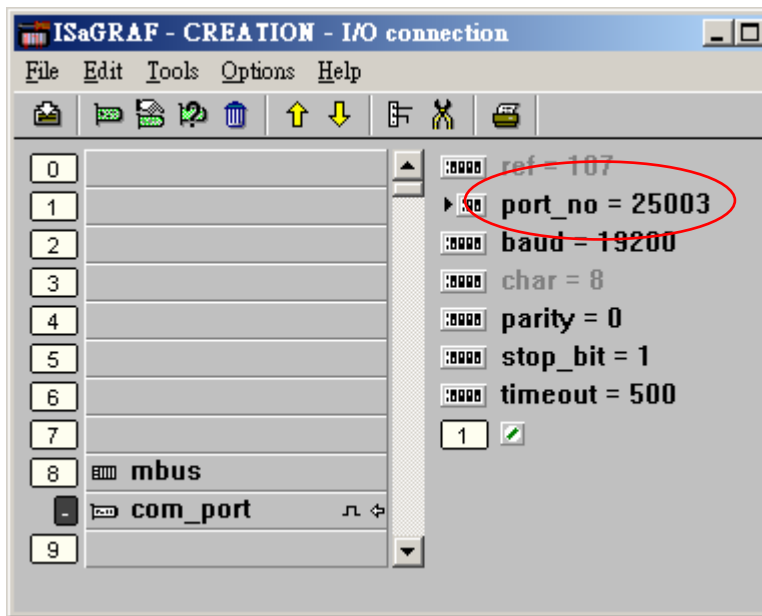
G. Key-in "isa7188e \*f=1" to free COM1 (Free COM1 as non-Modbus-Slave port)

i7188E> isa7188e \*f=1 (for i-7188EG)

i7188E> isa7186e \*f=1 (for μPAC-7186EG)

7. Some Modbus Slave devices need longer time gap between two modbus commands.

The normal time gap between two modbus commands is 3.5 Character time. The “mbus” and “mbus\_asc” default uses this time gap setting. They will delay about 3.5 Character time, then sending next Modbus command. If your Modbus Slave device need longer delay time , please set a special “port\_no” value in the IO connection “mbus” and “mbus\_asc”. (Max. Delay Time is 1000 ms = 1 second)



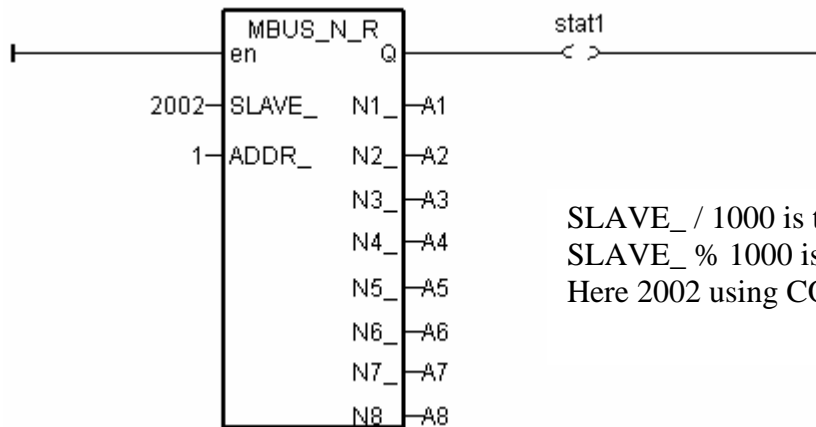
Delay\_time between two commands =  
 $\text{port\_no} / 100 + 3.5 \text{ character Time}$ ,  
(Here is 250 ms + 3.5 character Time)  
Port used =  $\text{port\_no} \bmod 100$   
(Here is COM3)

If setting port\_no as 50002, it means  
Delay time is 500 ms + 3.5 character time.  
Used Port is COM2

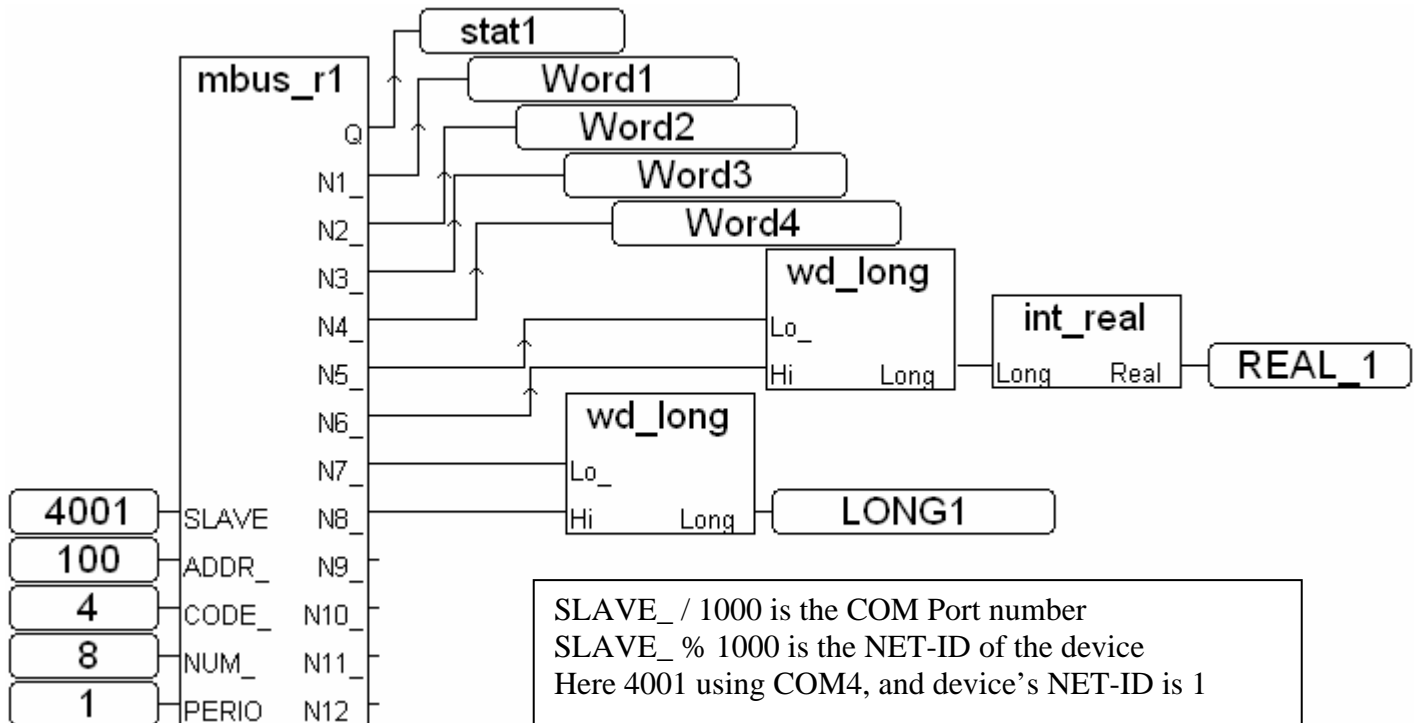
If setting port\_no as 2, it means  
Delay\_time is 0 ms + 3.5 character time .  
used Port is COM2



8. The input parameter of the “mbus\_xxx” block is not correct.



SLAVE\_ / 1000 is the COM Port number  
 SLAVE\_ % 1000 is the NET-ID of the device.  
 Here 2002 using COM2, and device's NET-ID is 2.



SLAVE\_ / 1000 is the COM Port number  
 SLAVE\_ % 1000 is the NET-ID of the device  
 Here 4001 using COM4, and device's NET-ID is 1

Some Modbus Slave devices can be read only one or two words in one modbus command. So the “mbus\_N\_R” block will not work with these devices . Alternately, please use “mbus\_r” or “mbus\_r1” in such a case. Their “NUM” parameter set the word amount can be read in one command.

Some Modbus Slave devices support only Modbus function call = 4 to read words, while function call = 2 to read bits. So the “mbus\_N\_R” and the “mbus\_B\_R” will not work with these devices . Alternately, please use “mbus\_r” or “mbus\_r1” in such a case. Their “CODE” parameter set the “modbus function call” number will be used.

---

## Chapter 9. Commonly Used ISaGRAF Utilities

---

**NOTE:**

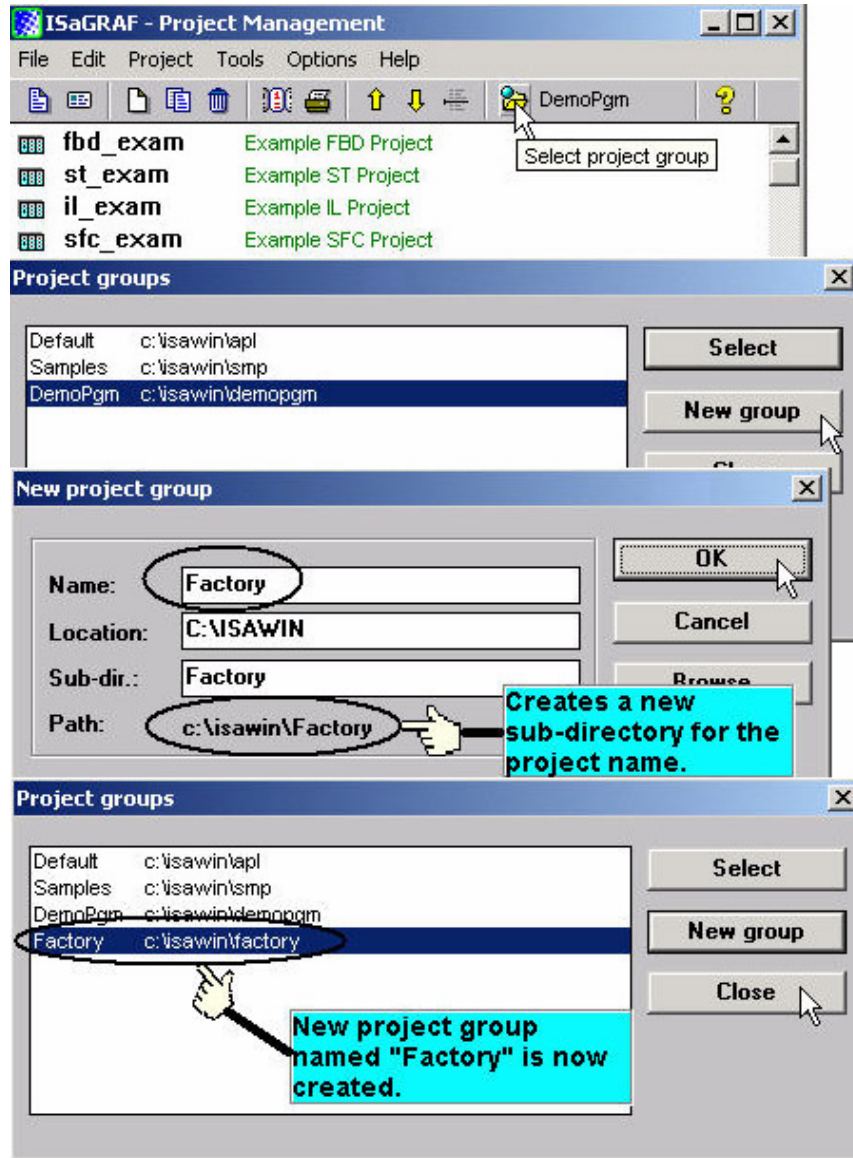
The I-8xx7 abbreviation is for the I-8417, I-8437, I-8817, I-8837, I-8437-80 and I-8837-80 controllers, while W-8xx7 is the abbreviation for the Wincon-8037/8337/8737 and Wincon-8047/8347/8747 controller.

The following chapter describes many useful features and utilities of the ISaGRAF Workbench programming environment. These features and utilities make programming an ISaGRAF project quick and easy.

This chapter in no way contains all of the features and utilities available with the ISaGRAF Workbench program. For more details and information about all the features the ISaGRAF Workbench program has to offer consult the "ISaGRAF USER's GUIDE" manual which can be found from the CD ROM of the ISaGRAF workbench. Its file name is either "ISaGRAF.pdf" or "ISaGRAF.doc".

## 9.1: Creating An ISaGRAF Project Groups

A very useful feature of the ISaGRAF program is the ability to organize numerous programs into "projects". The "Creating Projects" feature assists an ISaGRAF programmer who must create and maintain many different ISaGRAF programs for different application projects.

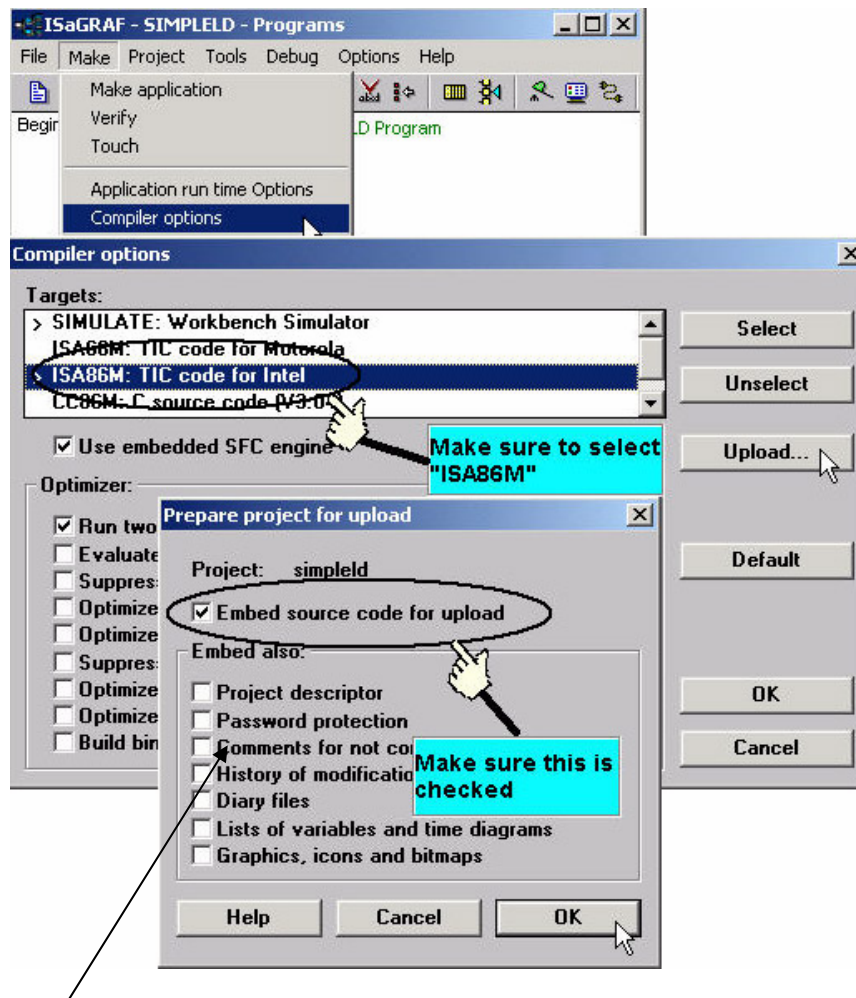


If you want to delete an existing project group, simply use the Windows Explorer to locate the ISaGRAF sub-directory you want to delete. An example of this is that if you wanted to delete the project just created, use the Windows Explorer and go to the C:\isawin\factory directory, and then just delete the "factory" sub-directory.

## 9.2: Uploading An ISaGRAF Project

There may be occasions when you will want to "Upload" an ISaGRAF project from an I-8xx7, I-7188EG/XG, uPAC-7186EG & W-8xx7 controller system to your development PC. This is easily accomplished **IF** the "Upload" function from the "Compiler Option" is turned on.

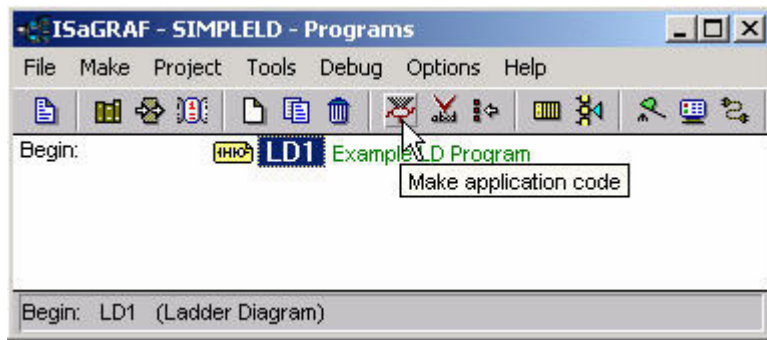
To turn the upload function on from the "Compiler Option", open the "ISaGRAF Programs" window, select "Make" from the menu bar, and then click on "Compiler Options". The "Compiler Options" window will open, make sure the "ISA86M: TIC Code For Intel" is selected, and then click on the "Upload" button. The "Prepare Project For Upload" window will open, click on the "Embed Source Code For Upload" checkbox and then click on the "OK" button.



### VERY IMPORTANT NOTE:

Option "Comments for not connected I/O channels" must be chosen if "Directly represented variables" is used in this project (refer to section 3.4).

After you have checked the "Embed Source Code For Upload" checkbox and clicked on the "OK" button, you will need to recompile the project and download the project to the I-8xx7, I-7188EG/XG & W-8xx7 controller system.



### IMPORTANT NOTE:

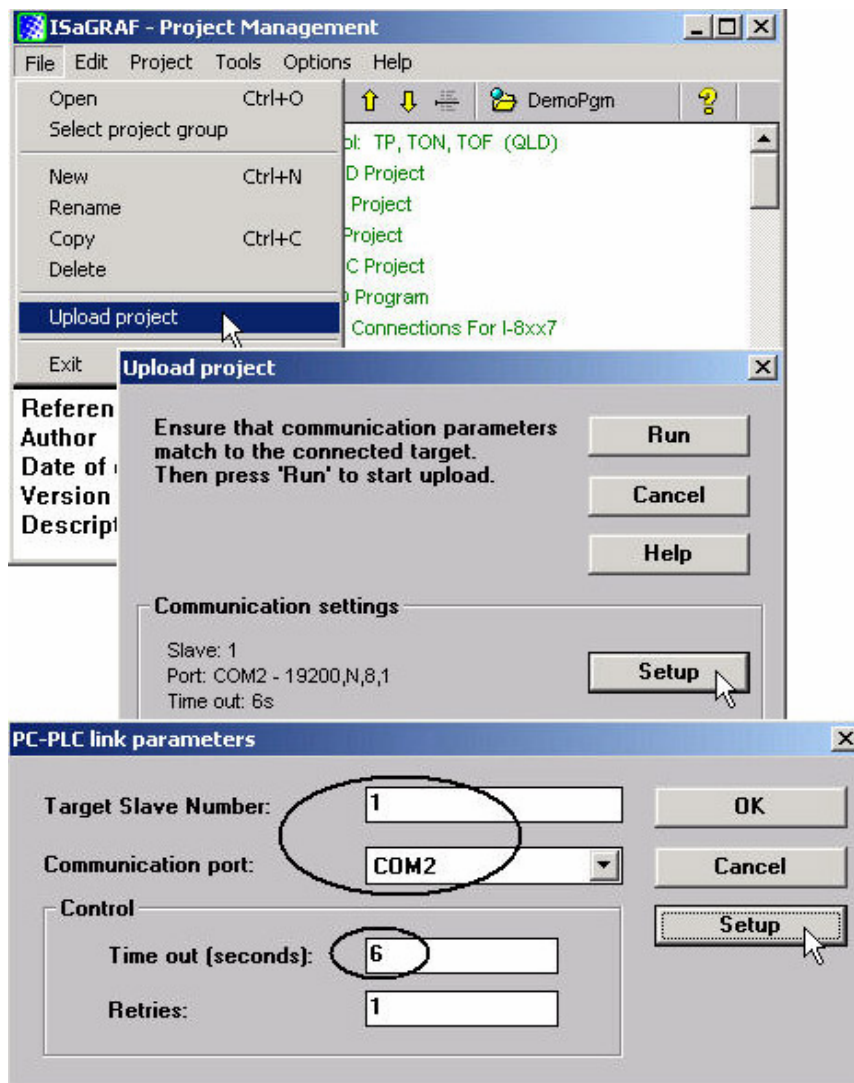
Once you have enabled the "Upload" option, the code generated by the compiler will increase the size of the original program from **ONE & A HALF TO THREE TIMES** the original program size. If the uploaded code size is larger than **64K** bytes, you will not be able to download the program to the I-8xx7, I-7188EG/XG & uPAC-7186EG controller system. The code size limitation is 1M bytes for W-8xx7 controller system.

Before trying to download the program it is advisable that you check the size of the uploaded program. To check the uploaded program size, use the Windows Explorer program and go to the appropriate sub-directory that the application program resides in. As an example, the "SIMPLELD" program that was create resides in the C:\ISAWIN\DEMOPGM\SIMPLELD program sub-directory.

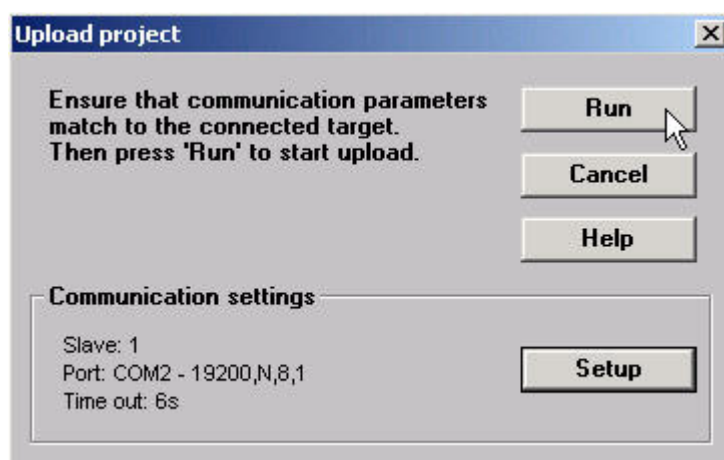
Remember, the "DEMOPGM" sub-directory is the **Project** group that the SIMPLELD program resides in, and the "SIMPLELD" sub-directory is where the actual application code files reside in. Look for the file named "**APPLI.X8M**" and check the size of this file. The "APPLIC.X8M" file is the file that contains the actual code that will be uploaded or downloaded to the I-8xx7 controller system. Make sure the sizes of this file DOES NOT exceed 64K byte for I-8xx7 , iPAC-8447/8847, I-7188EG/XG & uPAC-7186EG. And Do not exceed 1M byte for W-8xx7.

### UPLOADING AN ISaGRAF PROJECT

To upload an ISaGRAF project from an I-8xx7, I-7188EG/XG, uPAC-7186EG & W-8xx7 controller system open the "ISaGRAF Project Management" window, select "File", and then click on "Upload Project". The "Upload Project" window will now open, and check that the communication settings between your development PC and the I-8xx7, I-7188EG/XG, uPAC-7186EG & W-8xx7 controller system match each other. If the communication settings DO NOT match between the development PC and the controller, click on the "Setup" button to configure the proper communication settings.



Once you have made sure that the communication settings are properly configured, click on the "RUN" button in the "Upload Project" windows.

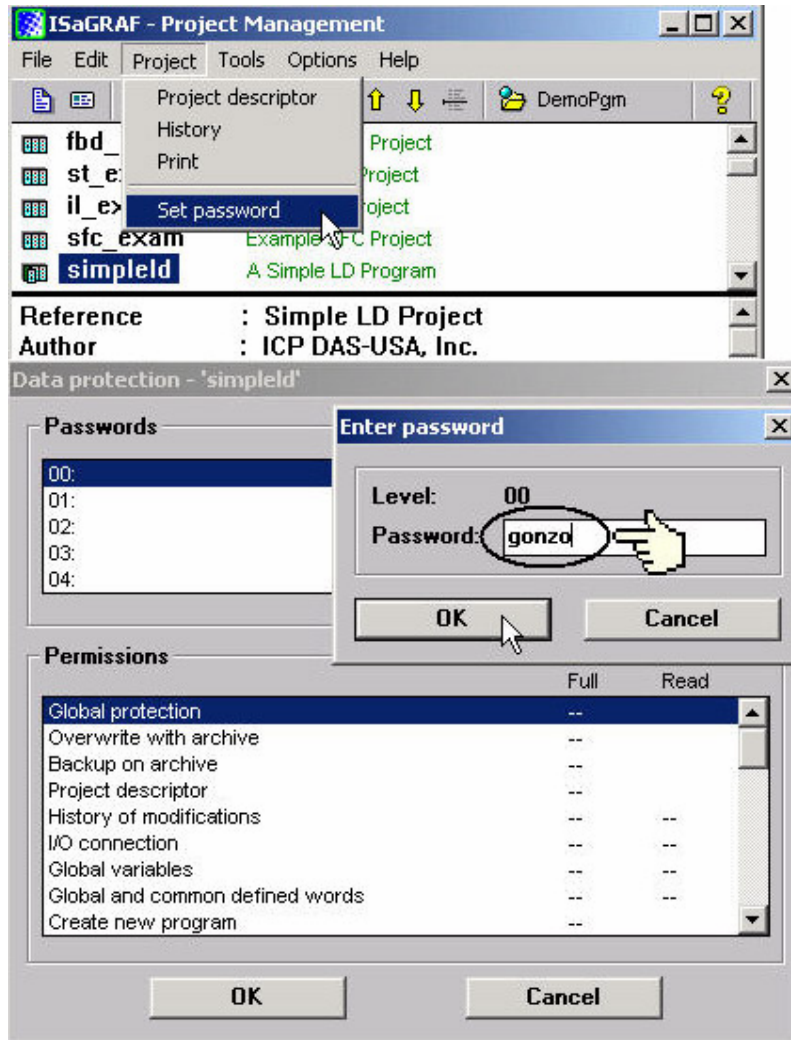




## 9.3: Setting An ISaGRAF Password

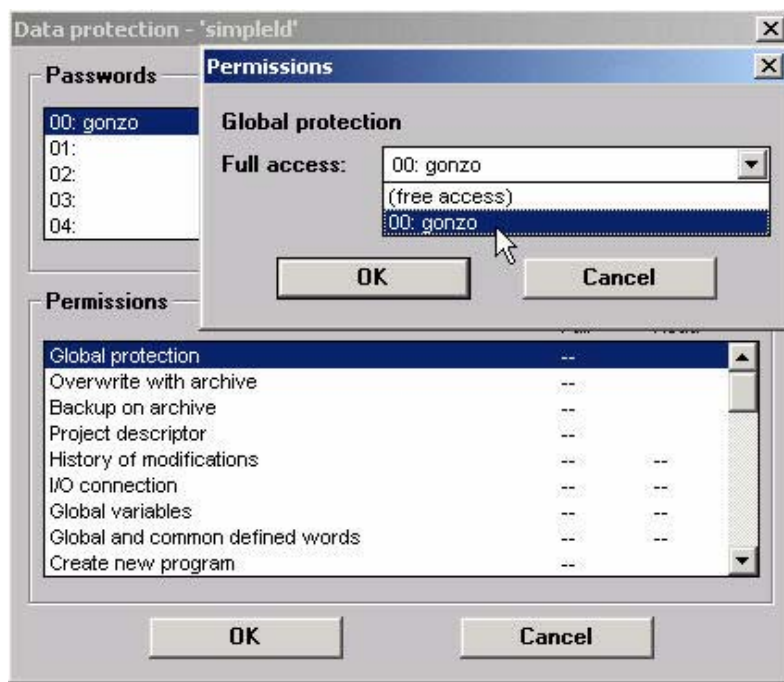
An ISaGRAF Workbench project can be password protected by configuring a user-defined password. To configure an ISaGRAF password, open the "ISaGRAF Project Window", select "Project" from the menu bar, and then click on "Set Password". The "Data Protection" window will open and then select one of the passwords from "00 to 15" to configure a password (this means that up to 16 passwords can be assigned with the ISaGRAF Workbench program).

You will also need to select the type of data protection you are creating for your ISaGRAF project. In the example below we are defining the "Global Protection" for this ISaGRAF project.

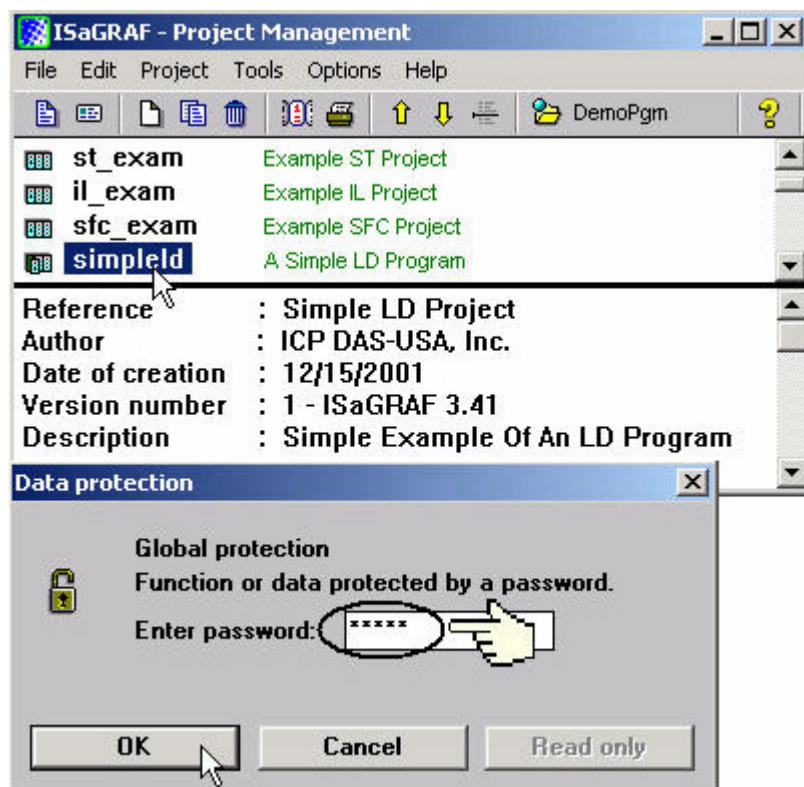


When you click on the "OK" button from the "Enter Password" window your new password will now be associated with the ISaGRAF project.

The next item you need to define is the type of data protection "Permissions" that will define for your ISaGRAF project. Double click on new password you have created and the "Data Protection Permissions" window will open. To allow full access WITH password protection, click on the "Full Access" scroll bar and click on the new password name you have created.



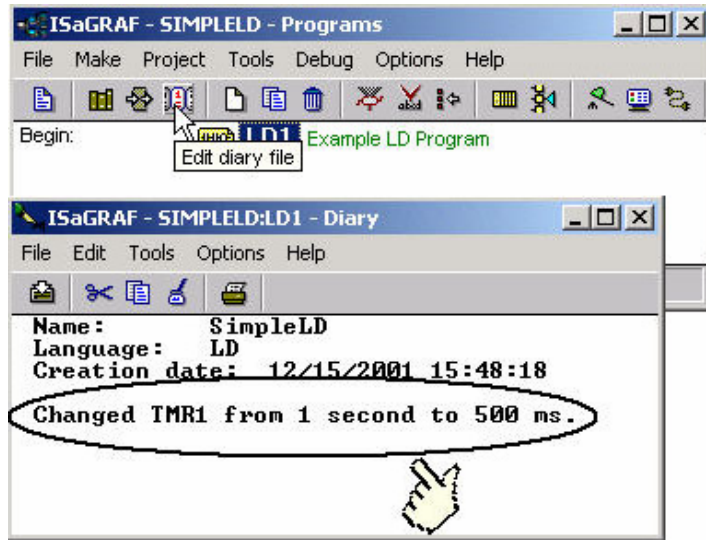
To verify that your password protection is now set for your ISaGRAF program, close all of ISaGRAF windows and then open the "ISaGRAF Project Management" window. Double click on the ISaGRAF program that you have created the password protection for. A "Data Protection" window will now open requiring you to enter the password for the ISaGRAF program you are attempting to open.





## 9.4: Creating An ISaGRAF Program Diary

When you modify an ISaGRAF program you can keep track of these revisions by entering a comment into the "Edit Diary" window. This affords the programmer the opportunity to add comments about program modifications and then save a record of these changes using the "Edit Diary" facility for enhanced program management capability.



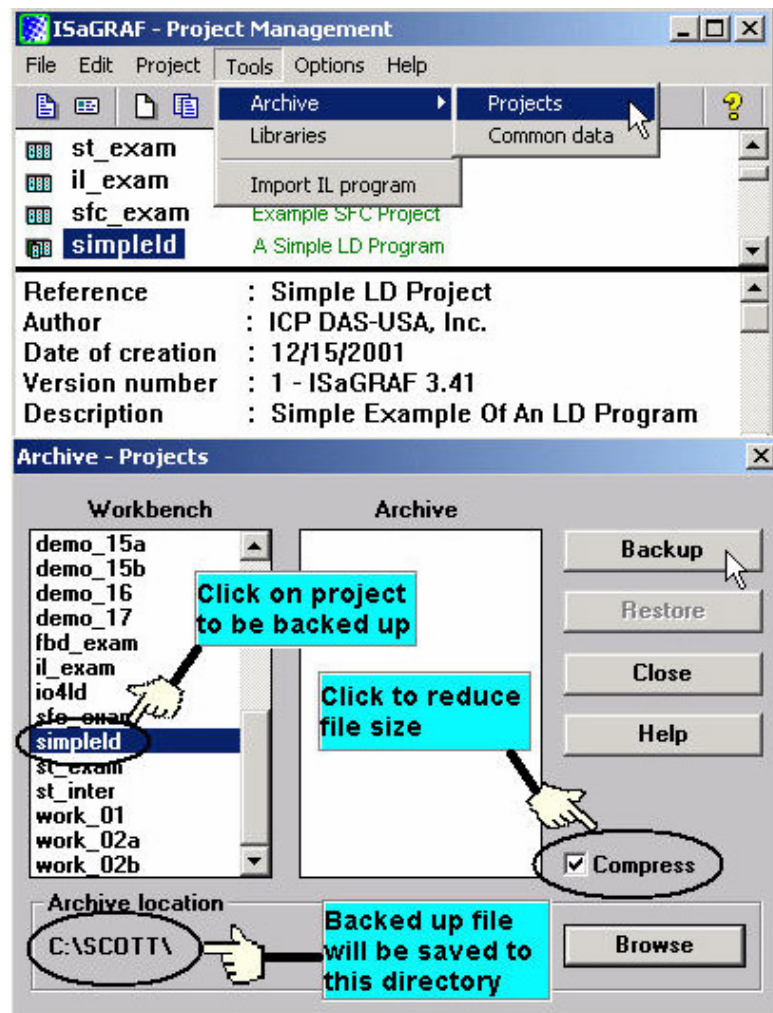
When you have completed entering information in the "ISaGRAF Diary" file, just click on the "Save" icon for your revision notes to be saved.

## 9.5: 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](mailto:service@icpdas.com) for ICP DAS's ISaGRAF technical service.

### Backing Up An ISaGRAF Project

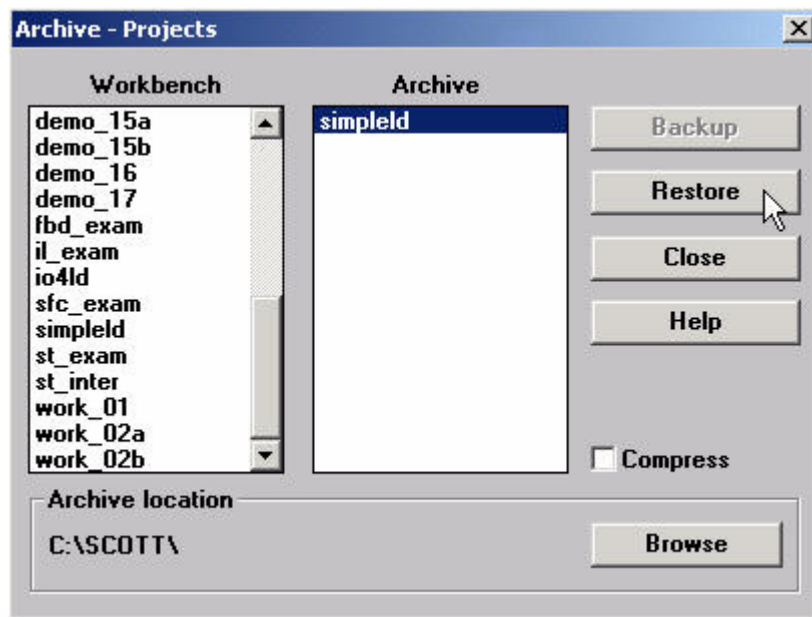
Open the "ISaGRAF Project Management" window, select "Tools" from the menu bar, click on "Archive", and then click on "Projects". An "Archive Projects" window will open which allows you to designate where you want to save the ISaGRAF project to. Click on the name of the ISaGRAF project you want to backup, 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.



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 "simpleld.pia".

### Restoring An ISaGRAF Project

To restore an ISaGRAF project from a backed up file, 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.



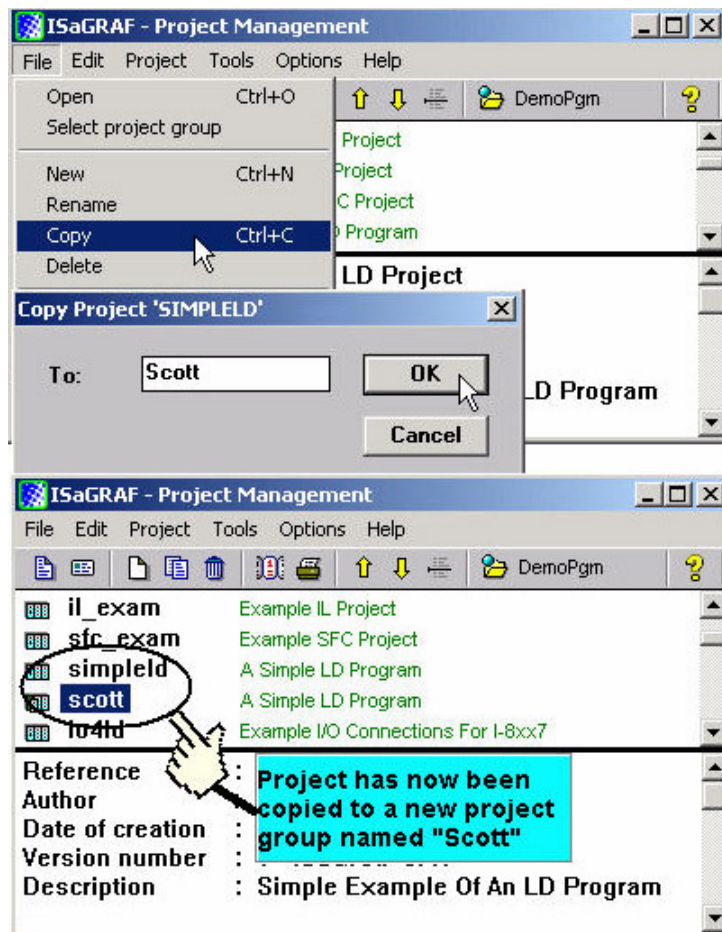
You can now open, edit and download the restored ISaGRAF project file.

## 9.6: Copying & Renaming An ISaGRAF Project

The ISaGRAF Workbench program has the capability of copying and renaming an ISaGRAF project or program. This is useful if you want to maintain a copy of an ISaGRAF project or program in a secondary directory.

### Copying An ISaGRAF Program

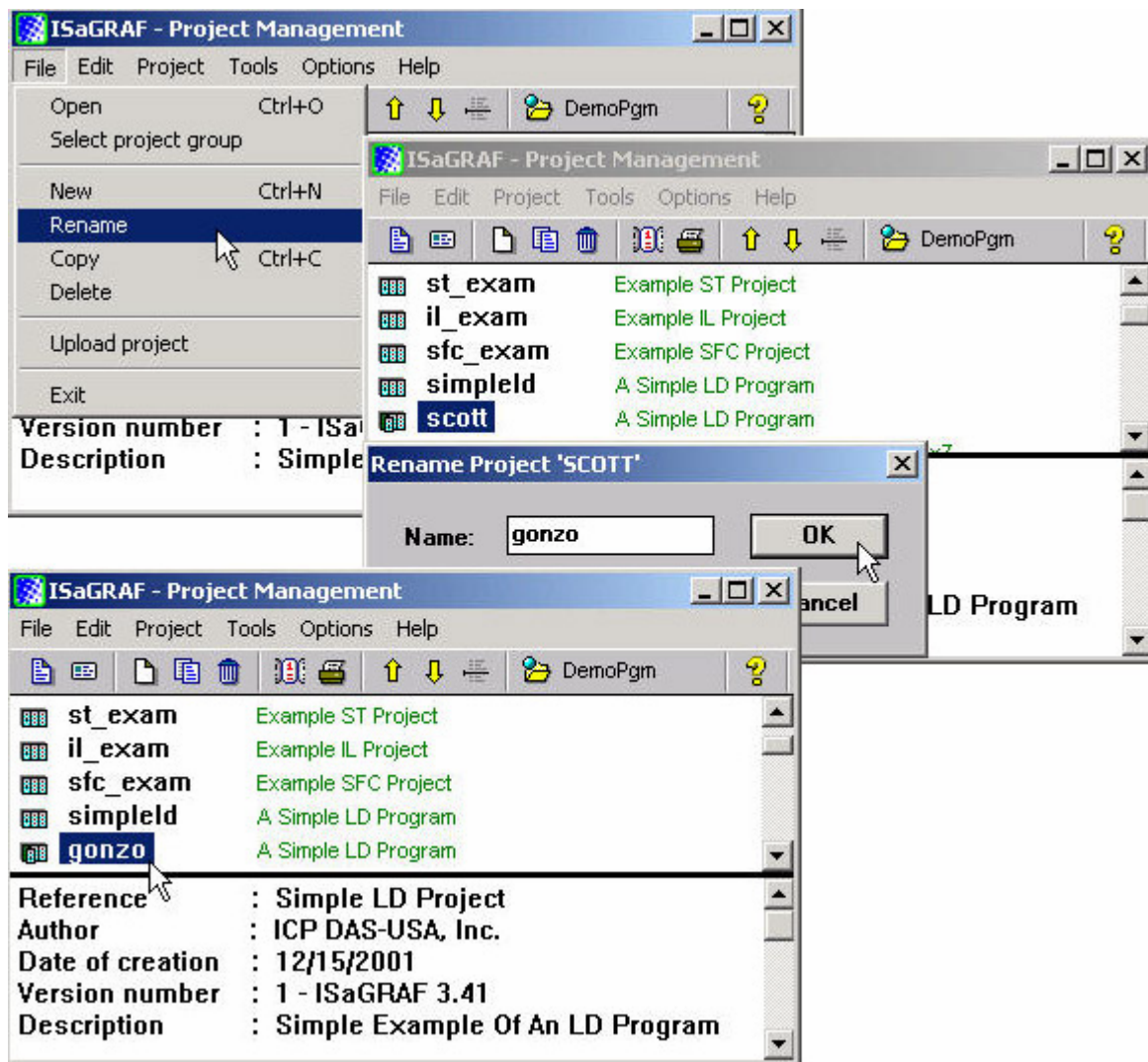
To copy an ISaGRAF program open the "ISaGRAF Project Management" window, first click on the name of the ISaGRAF program you want to copy, then select "File" from the menu bar, and then click on "Copy". When you click on "Copy" the "Copy Project" window will open, and now you can enter the name of the program you have selected to where you want to copy the program. If the new program name does not already exist, ISaGRAF will create the project name for you.



Note in the bottom screen that ISaGRAF has created a new program named "Scott" and placed a copy of all the files from "simpleld" into the "Scott" program group.

### Renaming An ISaGRAF Program

To rename an ISaGRAF program open the "ISaGRAF Project Management" window, click on the name of the ISaGRAF program you want to rename, then select "File" from the menu bar, and then click on "Rename". When you click on "Rename" the "Rename Project" window will open, and now you can enter the new name for the ISaGRAF program.



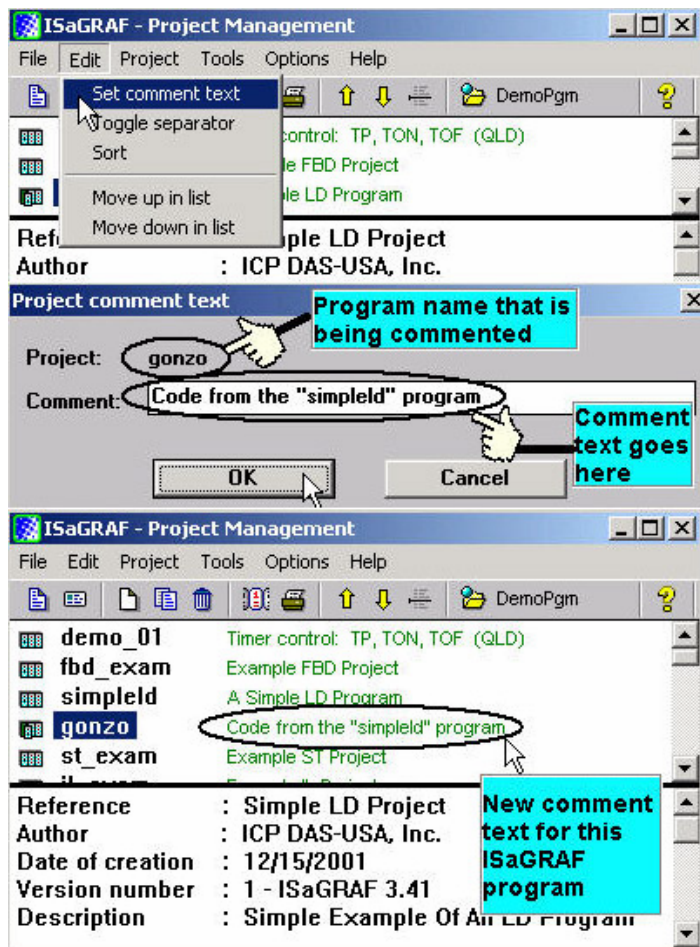
The former program named "scott" has now been changed to "gonzo", but it still has all the files from the "simpleld" program.



## 9.7: Setting Comment Text For An ISaGRAF Project

A useful feature of the ISaGRAF Workbench program is the ability to create "Comment Text" that will be placed next to an ISaGRAF program name in the "ISaGRAF Project Management" window. This way you can provide additional information about the purpose and any other additional comments regarding a particular ISaGRAF program.

To create "Comment Text" for an ISaGRAF program first open the "ISaGRAF Project Management" window, click on the name of the ISaGRAF program you want to create the comment text for, then select "Edit" from the menu bar, and then click on "Set Comment Text". When you click on "Set Comment Text" the "Project Comment Text" window will open, and now you can enter any comments and information you desire for the ISaGRAF program you have selected.

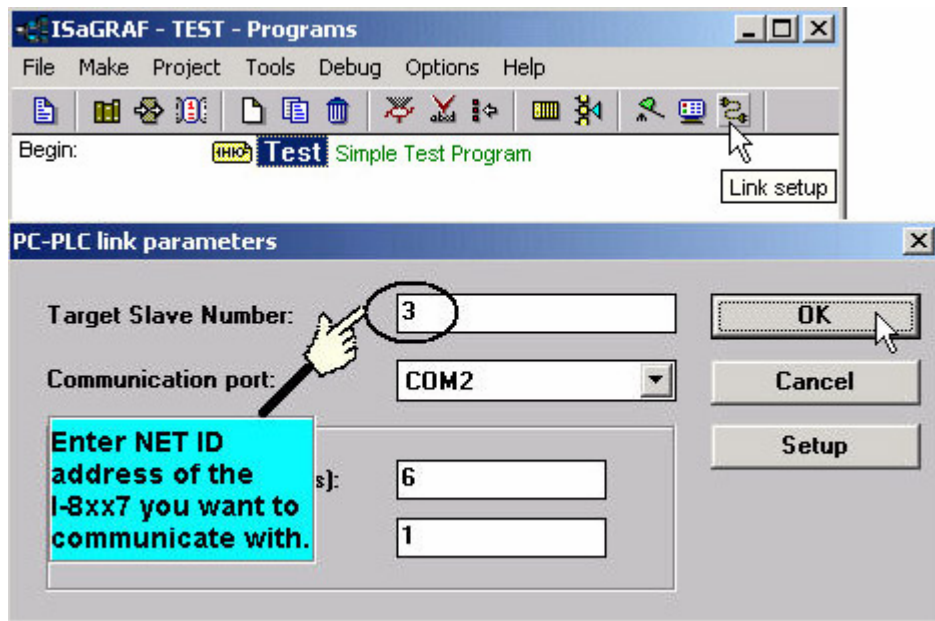


## 9.8: Setting The Slave ID For An ISaGRAF Controller

Each I-8xx7, iPAC-8447/8847, I-7188EG/XG, uPAC-7186EG & W-8xx7 controller system has a "NET ID" address that must be set to identify the controller to the ISaGRAF Workbench program. By default the NET ID address is "1" when it is shipped out.

If you need to communicate with multiple I-8xx7, iPAC-8447/8847, I-7188EG/XG, uPAC-7186EG & W-8xx7 controller systems via RS485 network, you must set the NET ID address in the ISaGRAF program for the specific I-8xx7, I-7188EG/XG & W-8xx7 controller system you want to communicate with. To communicate with different controller systems from one development PC open the "ISaGRAF Programs" window and click on the "Link Setup" icon.

When you click on the "Link Setup" icon, the "PC-PLC Link Parameters" window will open. Enter the "Target Slave Number" of the I-8xx7, iPAC-8447/8847, I-7188EG/XG, uPAC-7186EG & W-8xx7 controller system you want to communicate with.

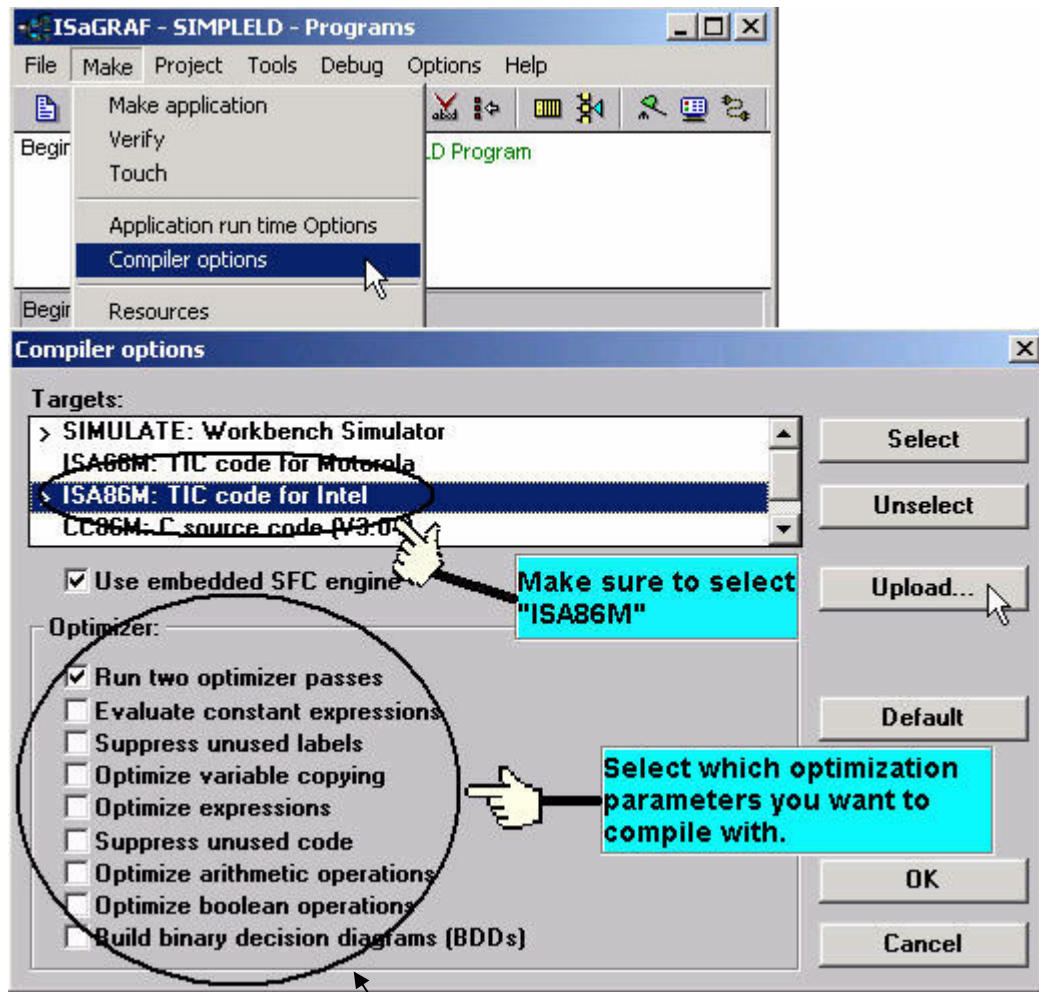


### IMPORTANT NOTE

Remember that the NET ID address of the I-8xx7 controller system is determined by the DIP switch settings on the bottom right hand side of the controller. Refer to Section 1.3.1 for the DIP switch settings to determine the NET ID address for the I-8xx7 controller system you want to communicate with. To set Net-ID for the I-7188EG/XG, uPAC-7186EG & W-8xx7, please refer to their respective "Getting Started Manual" delivered with the product.

## 9.9: Optimizing The ISaGRAF Code Compiler

The ISaGRAF Workbench program allows you to modify the settings for the "Compiler Options" to optimize the ISaGRAF program when you compile your project. To access the "Compiler Options" open the "ISaGRAF Programs" window and select "Make" on the menu bar, and then click on "Compiler Options". The "Compiler Options" window will open, and now you can select which optimization parameters you want for when you compile your ISaGRAF program.



If using "Variable Array" in the program, please DO NOT check the 2nd , 7th , 8th and 9 th Optimizer options, or the value of the Variable array will be incorrect. Recommend to check only the 1st – "Run two optimizer passes" option. (Please refer to Chapter 2.6)

Selecting the "Run Two Optimizer Passes" will insure that the code is compiled into the smallest possible program code.



## 9.10: Using The ISaGRAF Conversion Table

### Note:

The conversion table is only for Input & Output attribution variables, not for internal variables. You may refer to Appendix A.4 for “A4\_20\_to”, “To\_A4\_20” to convert the analog value of 4 to 20 mA to application engineering value. Or “V0\_10\_to”, “To\_V0\_10” for converting analog value of 0 to 10 Volt to application engineering value.

### Conversion Table Example

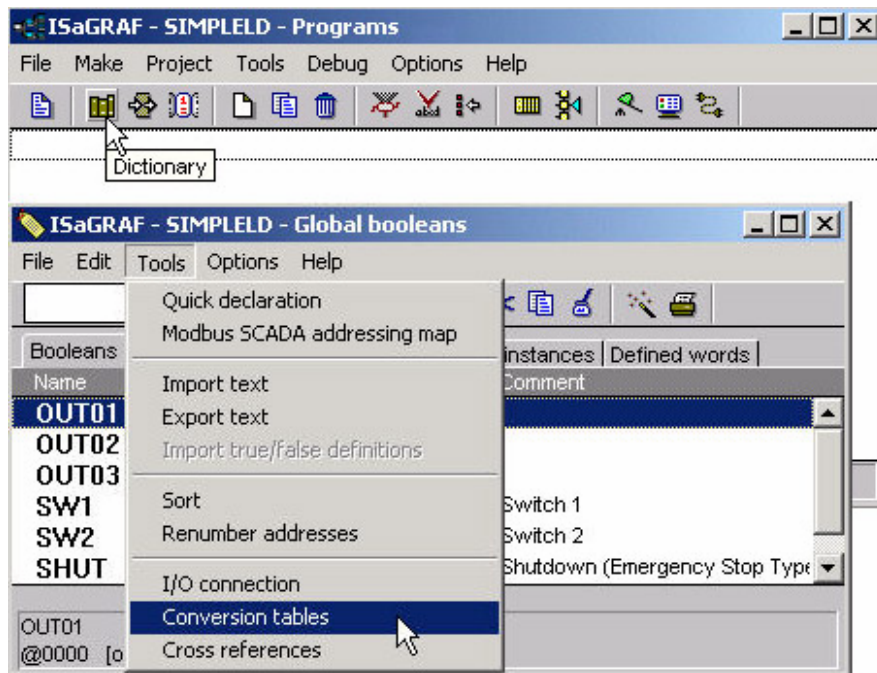
In this "Conversion Table" example the value from an I-87017 (an eight channel analog input module) board needs to be converted. The I-87017 is configured to receive a –10v to +10v signal, where –10v equals a value of "-32768", and a +10v signal equals a value of "+32767". You may refer to Appendix D to see the translation table of each analog board.

In this example we will use the "Conversion Table" to reconfigure the I-87017 so that a –10v signal will equal a value of "-10000" and a +10v signal will equal a value of "10000". In this example a value of +2.573v signal will equal a value of "2573".

### Note:

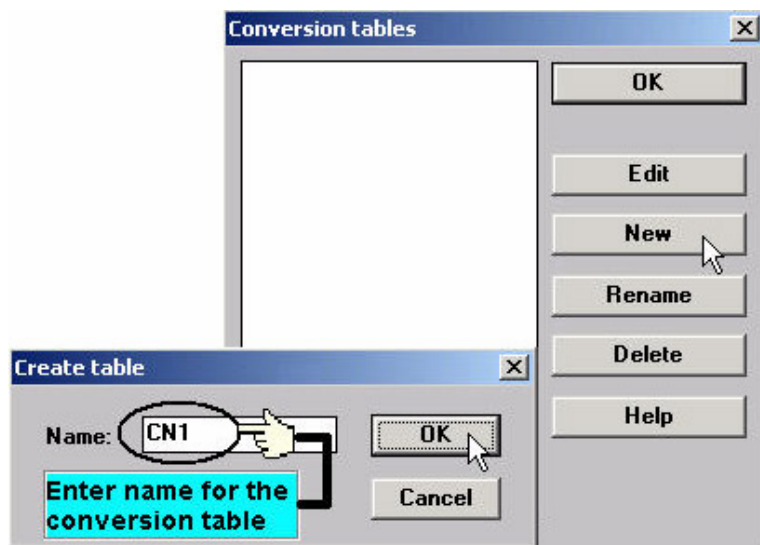
The I-8xx7, I-7188EG/XG & W-8xx7 controller only supports the value before conversion within –32768 to +32767, and the value after conversion within **–10000 to +10000**. Setting conversion table out of these range may cause errors.

To configure a "Conversion Table" open the "ISaGRAF Programs" window and click on the "Dictionary" icon. This will open the "ISaGRAF Global Variables" window, select "Tools" from the menu bar, and then click on "Conversion Tables".

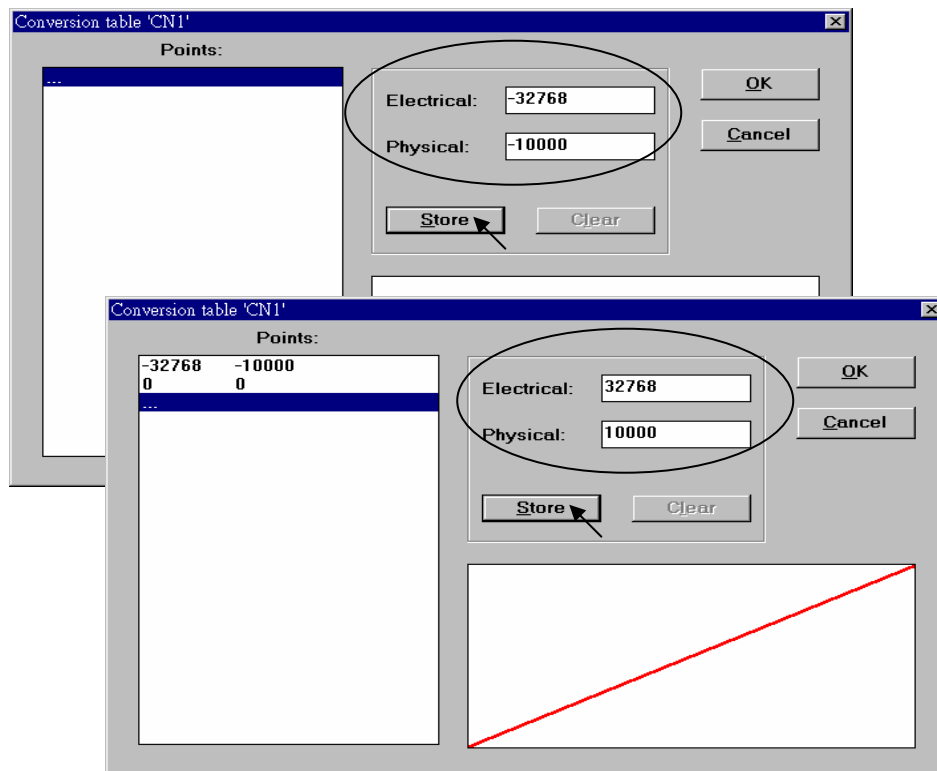


When you click on the "Conversion Tables" selection the "Conversion Tables" window will open.

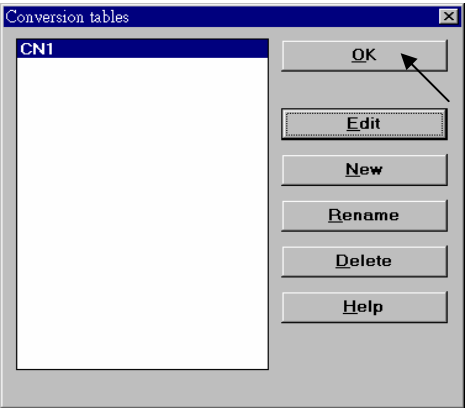
Next, click on the "New" button and then the "Create Table" window will now open. In the "Create Table" window enter the name for the conversion table you are creating.



To properly create our example "Conversion Table" at least two values must be defined. The "Electrical" field means the original value BEFORE conversion and the "Physical" field is for the value AFTER conversion. The two points defined in this example are (-32768, -10000 "lower limit") and (+32767, 10000 "upper limit"). Click on the "STORE" button to save each entry.

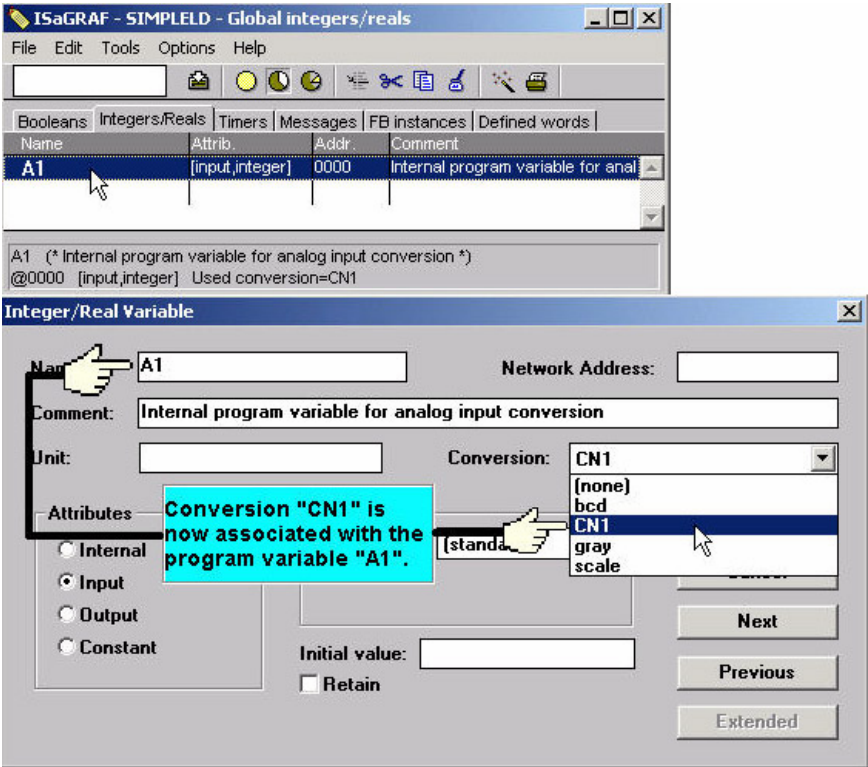


When you have completed entering in the two value points, click on the "OK" button to save the entered values.



The last step is to assign the conversion table "CN1" to a program variable that will be used in an ISaGRAF program.

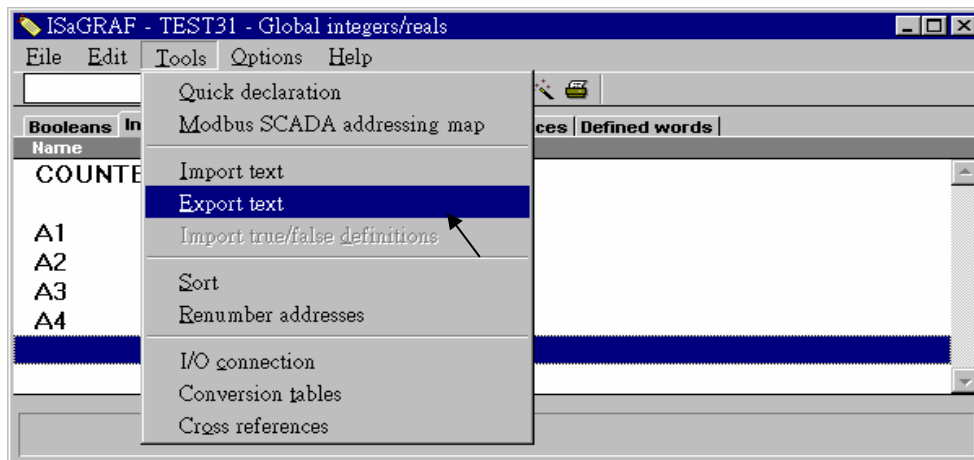
**Note:**  
Only integer variable declared as input or output attribution can be assigned a conversion table.



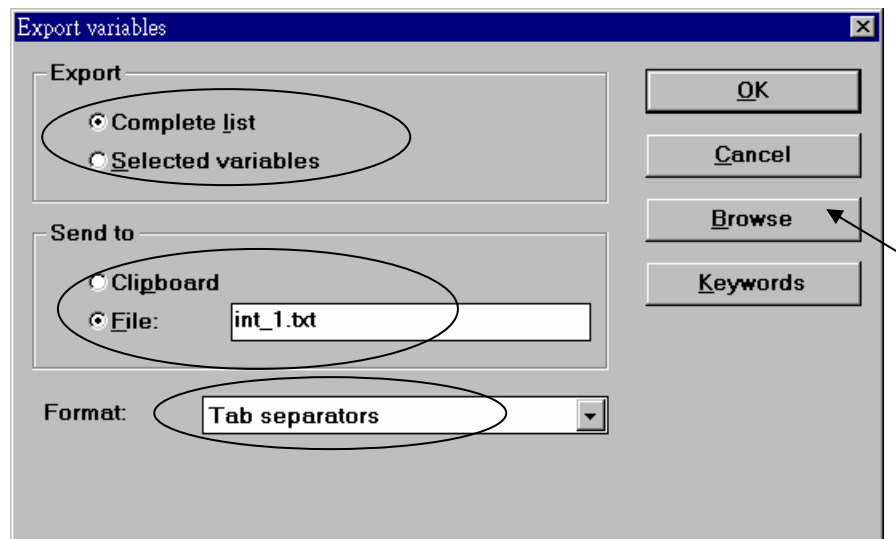
## 9.11: Export / Import Variable Declarations Via Microsoft Excel

Variables can be defined in Microsoft Excel and then be imported to ISaGRAF workbench. And also they can be exported from ISaGRAF to Excel.

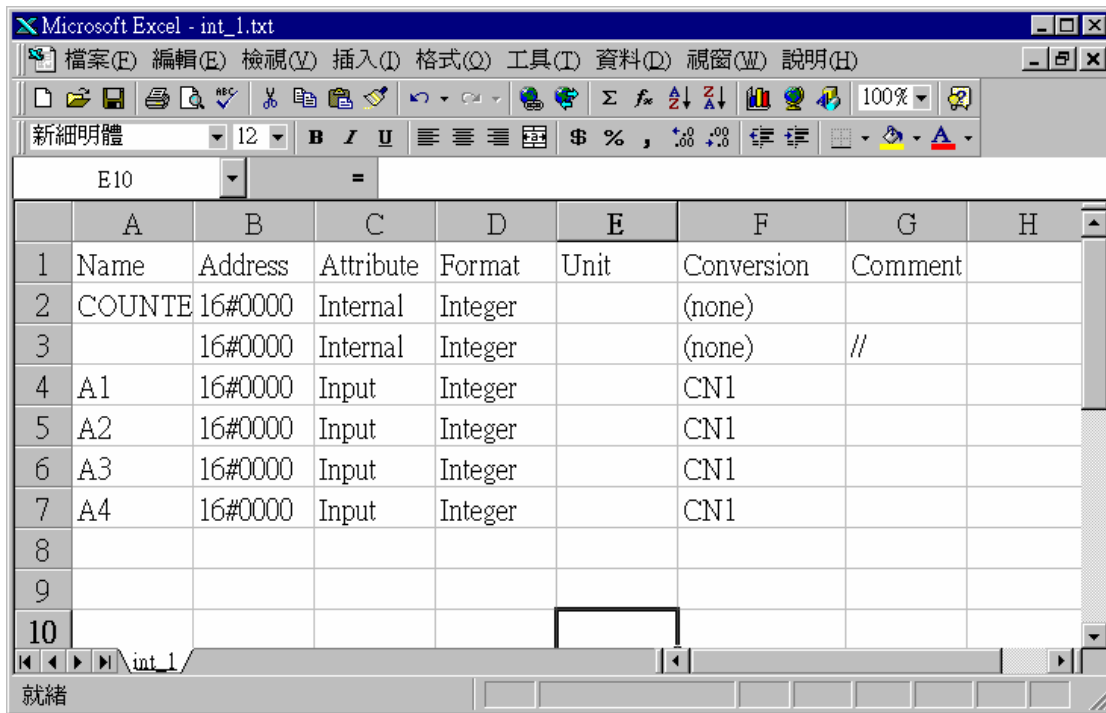
To export to a text file, with an extension name **“.txt”**, run “Tools” - “Export text” from the “dictionary” window.



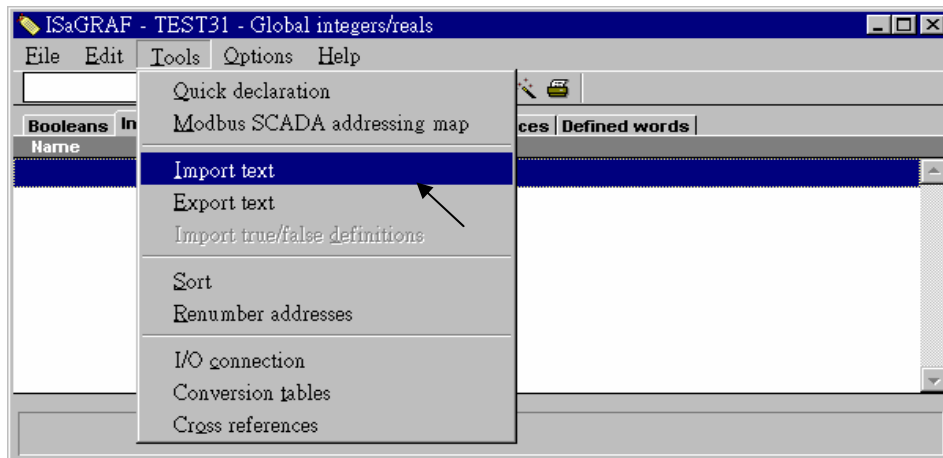
Select “File” and given a name to it, “int\_1.txt” in this sample. Then click on “Browse” to select the directory where this txt file will be saved.



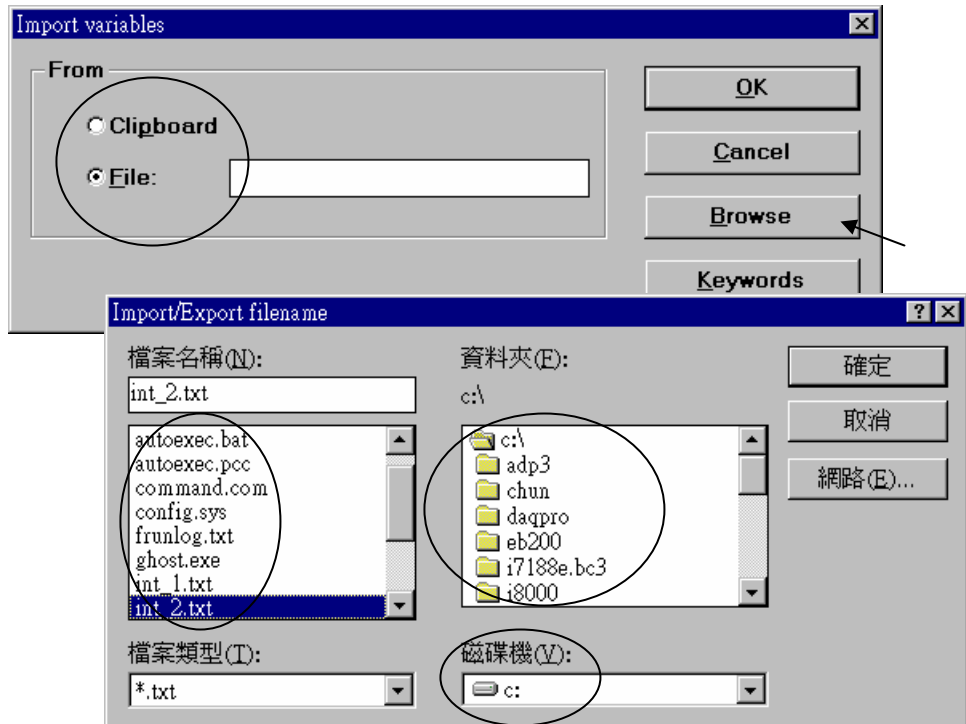
You may open and edit the file from the Excel. Please make sure to save this file with an extension “.txt”.



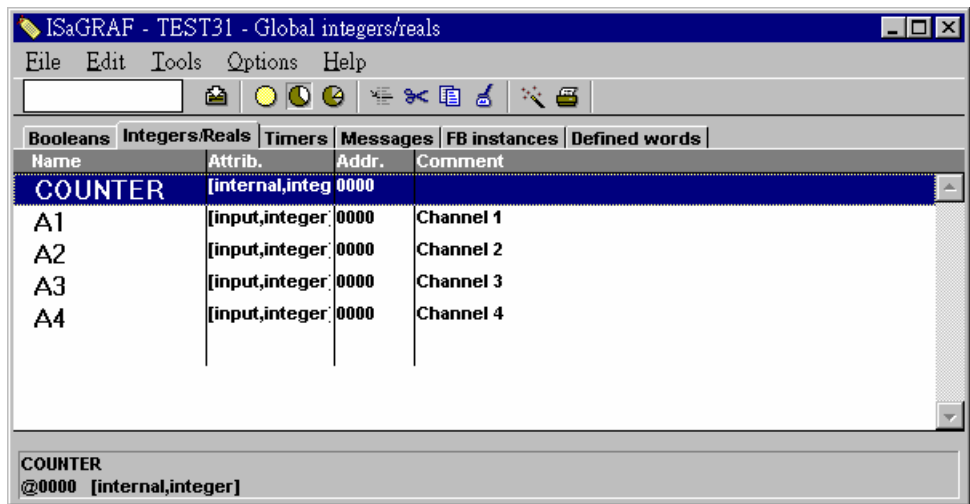
To **import** a text file to ISaGRAF, with an extension name “.txt”, run “Tools” - “Import text” from the dictionary window.



Then click on “Browse” to select the associated text file.



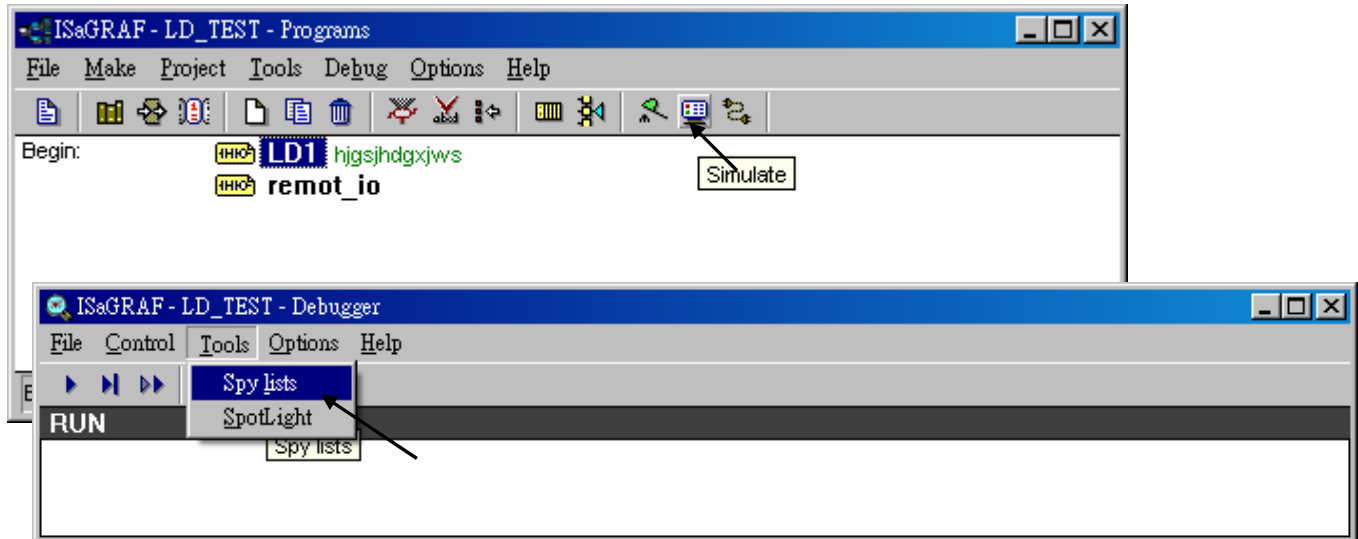
And then it is done as below.



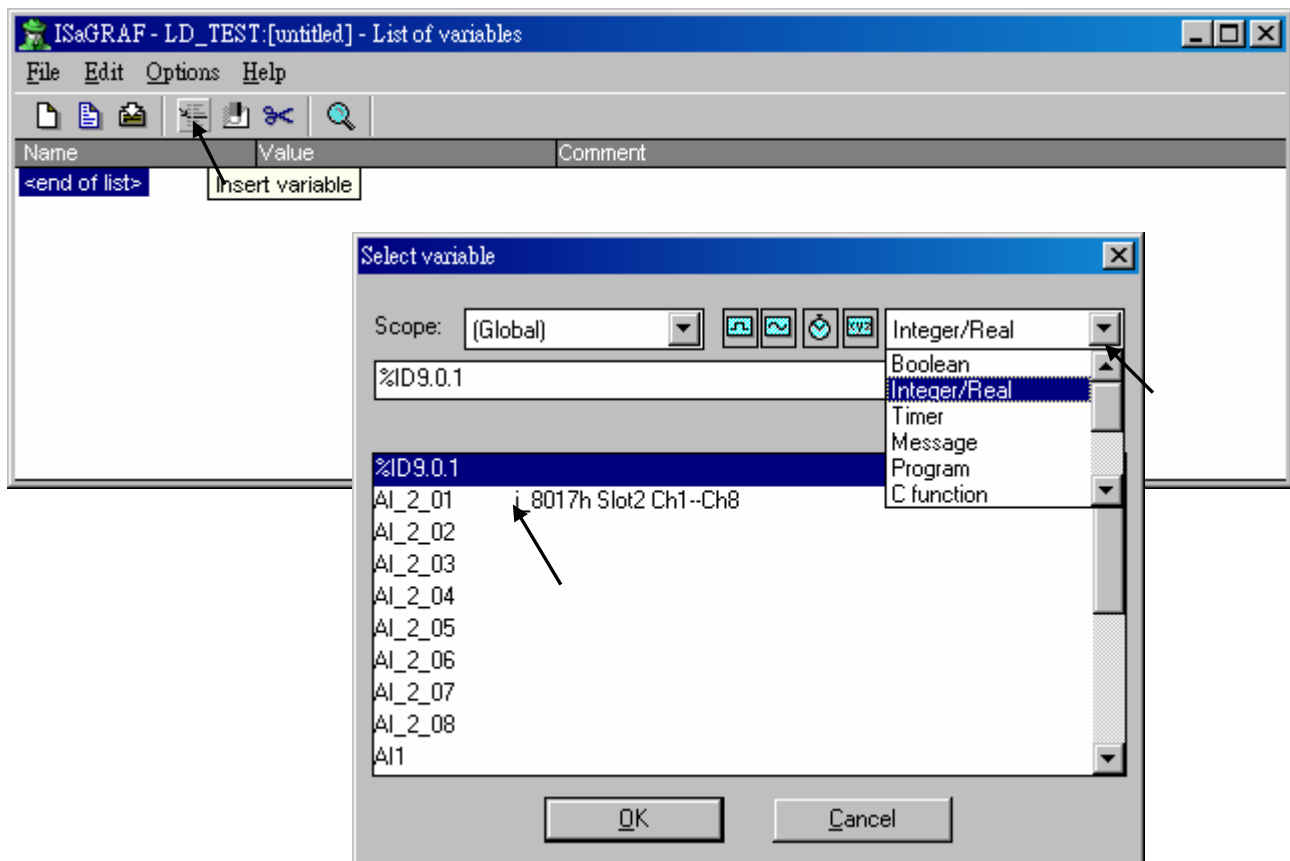
## 9.12: Spy list

ISaGRAF supports “Spy list” to spy some specific variables when linking to the controller. Please follow below steps to create a “spy list”.

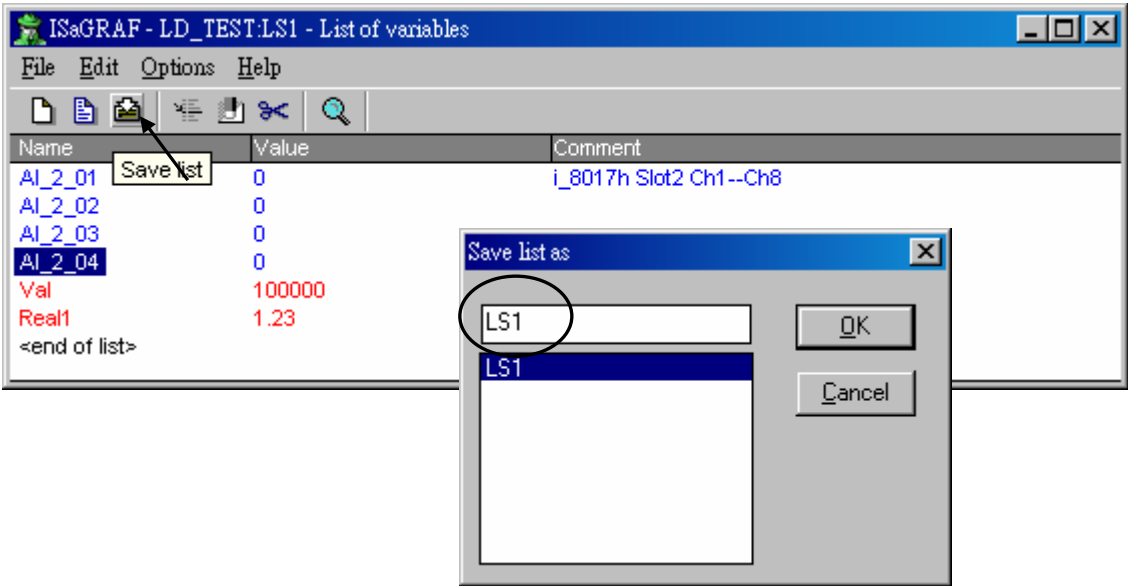
First click on “Simulate”, then click on “Tools – Spy list”.



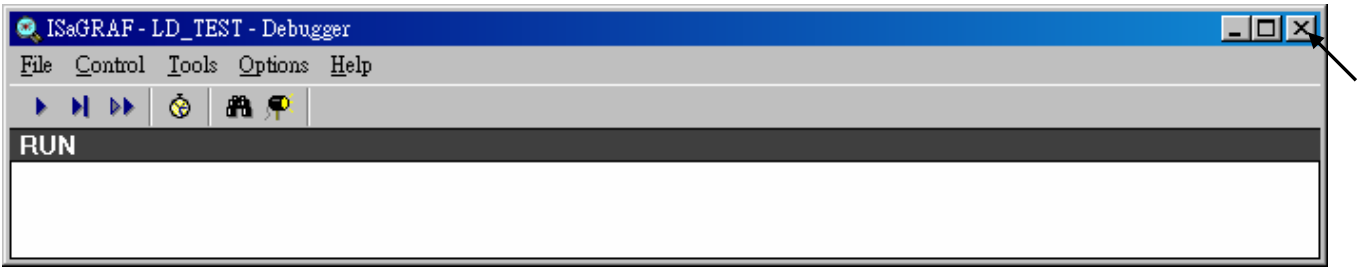
Next click on “Insert variable” to insert the variable to be spied.



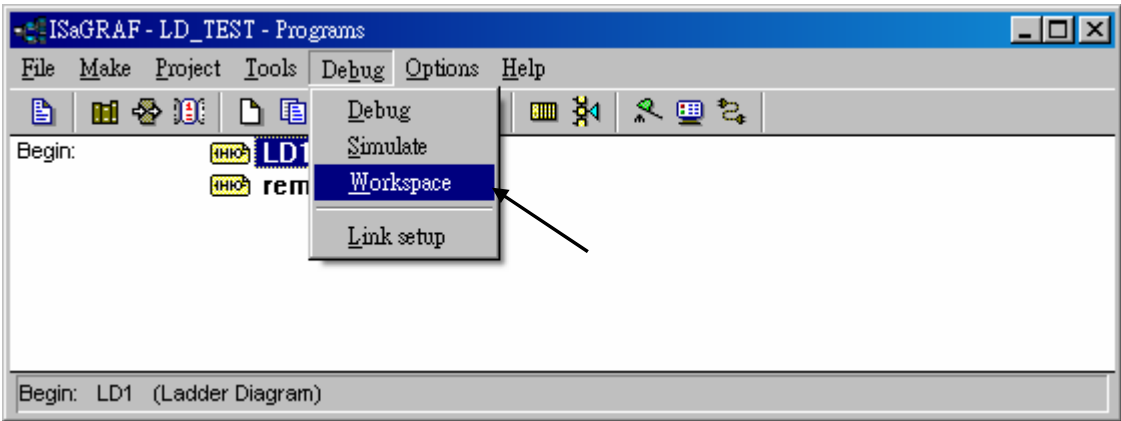
When all spied variables are inserted, remember to click on “Save list”.



Then close the ”Debugger” window.

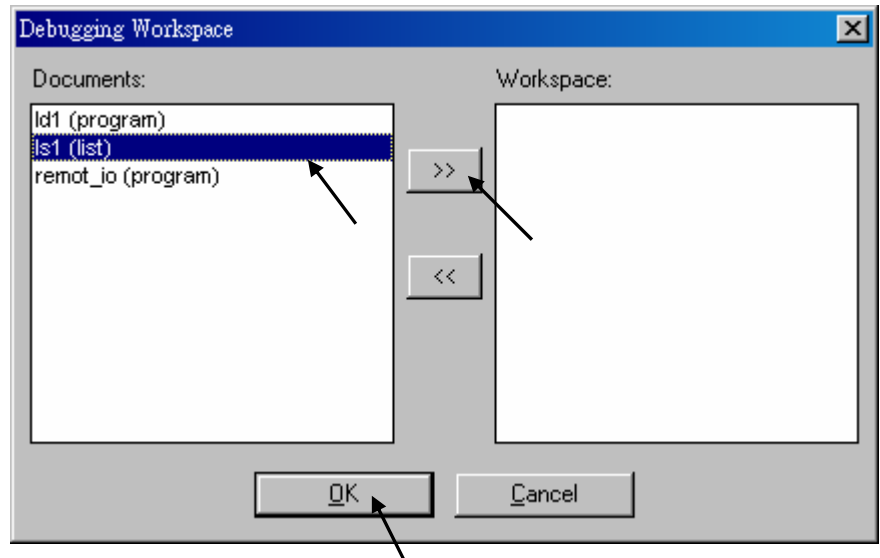


Click on “Debug – Workspace”

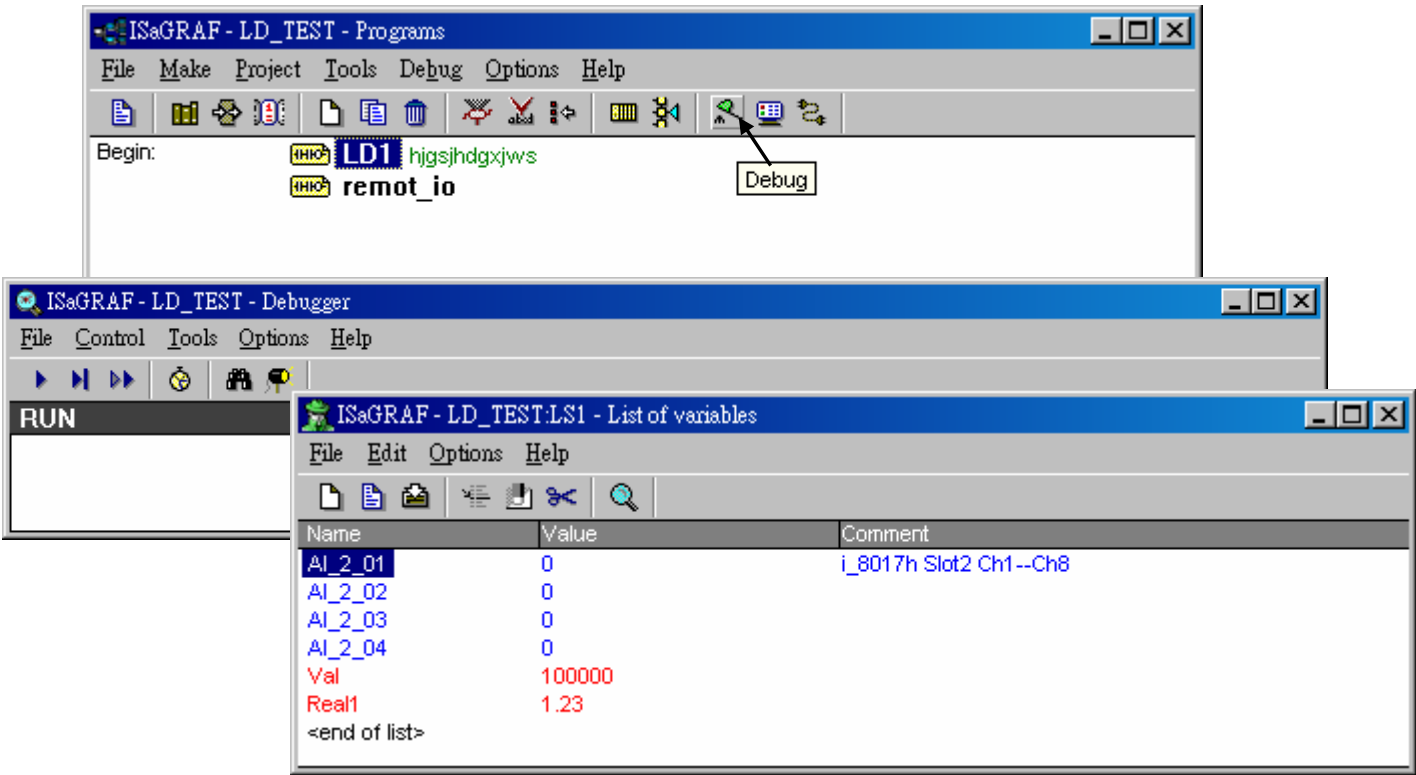




Move all “List” to the right hand side.

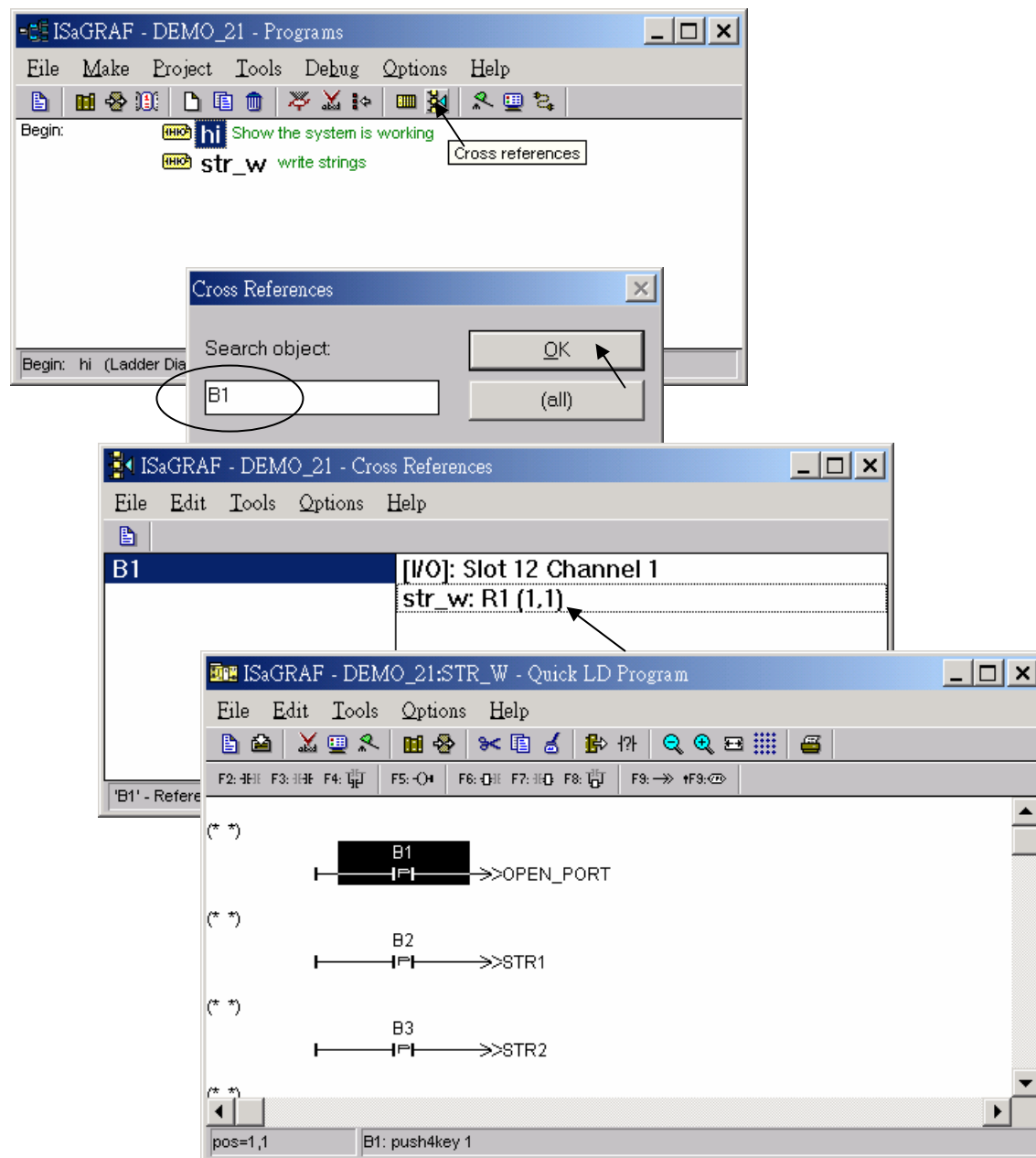


Then, you will see the “spy list” will automatically display when ISaGRAF linking to the controller.



## 9.13: How to search a variable name in an ISaGRAF project ?

Please click on “Cross references” and then entering the name you would like to search. The location will appear on the right hand side. Just click on it to get into it.



---

# Chapter 10. The Retained Variable And Data Backup

---

## 10.1: The Retained Variable

---

**Note:** Read floating point value from S-256/512 & X607/608 may cause controller fault if there is no floating point value saved inside. Please refer to Section 10.6 – “Controller Fault Detection” (**Note:** 0-slot Wincon doesn’t support S-256/S-512)

### New Retain Function:

The I-8417/8817/8437/8837, iPAC-8447/8847, I-7188EG/XG, uPAC-7186EG, Wincon-8x37/8x36 & Wincon-8x47/8x46 supports new retain function since below driver version.

I-7188EG + X607 or X608: driver ver. 2.05 or later (better to be 2.17 or later)

I-7188XG + X607 or X608: driver ver. 2.04 or later (better to be 2.15 or later)

uPAC-7186EG + X607 or X608: driver ver. since they are released around Jan.2008

I-8xx7+ S256 or S512 : driver ver. 3.07 or later (better to be 3.19 or later)

W-83x7/83x6+ S256 or S512 : driver ver. 3.18 or later (better to be 3.36 or later) with new back-plane of WB-831 (For 3-slot ): Rev 2.6

W-87x7/87x6+ S256 or S512 : driver ver. 3.18 or later (better to be 3.36 or later) with new back-plane of WB-871 (For 7-slot ): Rev 2.8

If battery backup SRAM is found in the back-plane of the controller (I-8xx7: S256/S512, I-7188EG/XG: X607/X608, Wincon-83x7/87x7/83x6/87x6: S256/S512 ), the maximum number of retained variables for new retain function are listed as below. New retain variable is supported by below ISaGRAF “C-function”

Target 1 : I-7188EG/XG/uPAC-7186EG+X607/608, I-8417/8817/8437/8837+S256/512

Target 2 : Wincon+S256/512 with new Wincon back-plane

Retain\_B : retain Boolean variable. Target 1: max. 256 variables, Target 2: max. 1024.

Retain\_N : retain Integer variable. Target 1: max. 1024 variables, Target 2: max. 4096.

Retain\_F : retain Real variable. Target 1: max. 1024 variables, Target 2: max. 4096.

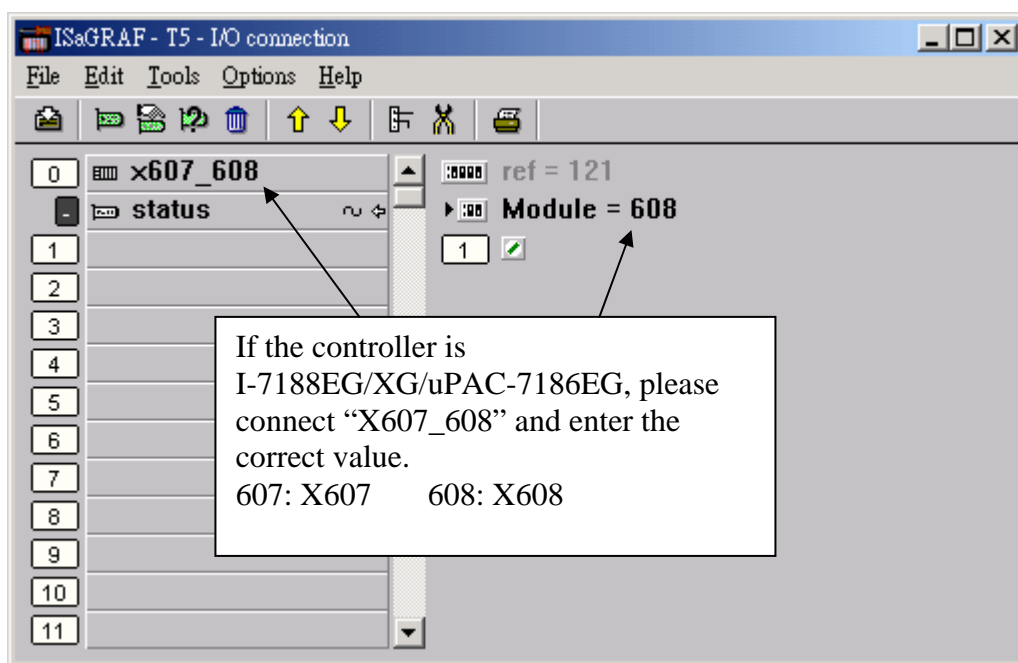
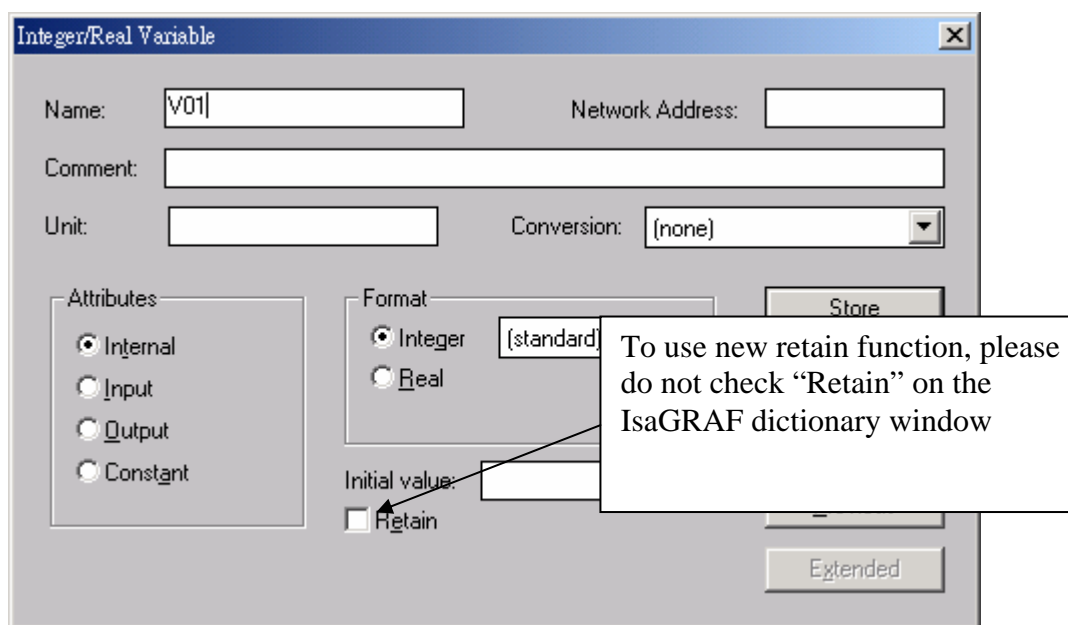
Retain\_T : retain Timer variable. Target 1: max. 256 variables, Target 2: max. 1024.

Retain\_X : retain variable by using its Network address

The retain value by new retain function will keep alive always whatever controller’s power is off, or modifying , re-compiling & download a new ISaGRAF project. Data will be lost when rinning out of the battery power. Please refer to below two ST examples to use new retain function.

### Important:

To use new retain function, please **do not** check “Retain” on the IsaGRAF dictionary window. And if your controller is **I-7188EG/XG or uPAC-7186EG**, please connect IO complex equipment “**X607\_608**” in the IO connection windows.



**Example1: (\* Set by variable name \*)**

```
(* To_Retain is declared as an internal boolean variable with initial value as TRUE
Tmp is declared as an internal boolean variable
B1 , B2 is declared as internal Boolean variable, Do not check "Retain"
N1 , N2 is declared as internal Integer variable, Do not check "Retain"
F1 , F2 is declared as internal Real variable, Do not check "Retain"
T1 , T2 is declared as internal Timer variable, Do not check "Retain"  *)

(* To set retained variables when controller is start running *)

if To_Retain then
  To_Retain := False ;  (* Only do it once *)
  Tmp := Retain_B( B1 , 1 ) ;      Tmp := Retain_B( B2 , 2 ) ;
  Tmp := Retain_N( N1 , 1 ) ;      Tmp := Retain_N( N2 , 2 ) ;
  Tmp := Retain_F( F1 , 1 ) ;      Tmp := Retain_F( F2 , 2 ) ;
  Tmp := Retain_T( T1 , 1 ) ;      Tmp := Retain_T( T2 , 2 ) ;
end_if ;

(* After then B1, B2, N1, N2, F1, F2, T1, T2 will be automatically retained in the program *)
```

**Example2: (\* Set by variable's network address No. \*)**

```
(* To_Retain is declared as an internal boolean variable with initial value as TRUE
Tmp is declared as internal boolean variable
ii is declared as an internal integer
N01 ~ N10 is declared as internal Integer variable with network address No. = 1, 3, 5, 7, 9, 11, 13,
15, 17, 19, Do not check "Retain"
F01 ~ F10 is declared as internal Real variable with network address No. = 21, 23, 25, 27, 29, 31,
33, 35, 37, 39, Do not check "Retain" *)

(* To set retained variables when controller is start running *)

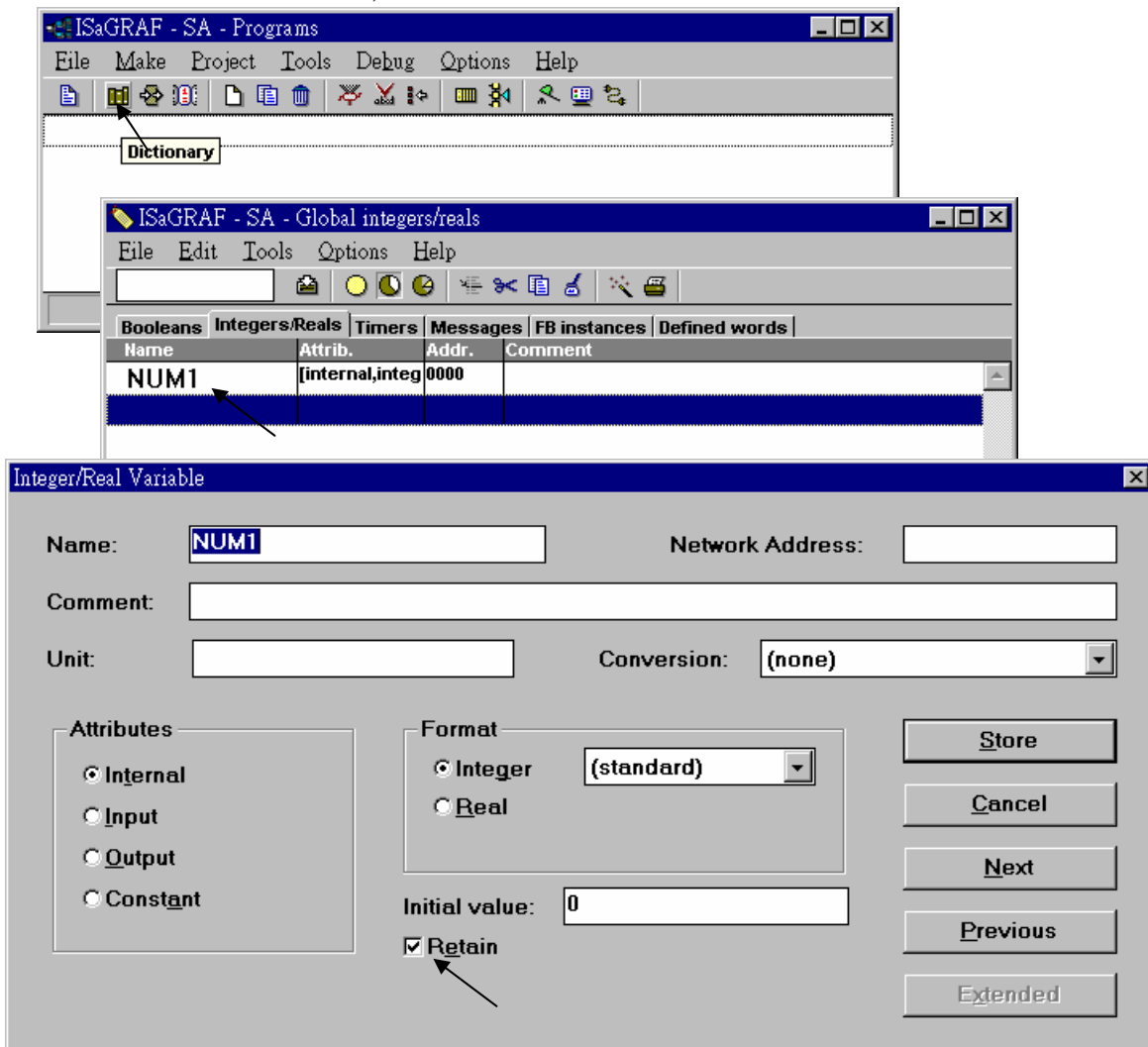
if To_Retain then
  To_Retain := False ;    (* Only do it once *)
  for ii := 1 to 10 do
    Tmp := Retain_X( 'N' , 2*ii-1 , ii ) ;  (* retained N01 to N10 *)
    Tmp := Retain_X( 'F' , 2*ii+19 , ii ) ;  (* retained F01 to F10 *)
  end_for ;
end_if ;

(* After then N01 to N10 & F01 to F10 will be automatically retained in the program *)
```

## Old Retain Method:

If the controller doesn't find the battery backup SRAM in the back-plane of the controller (I-8xx7: S256/S512, I-7188EG/XG/uPAC-7186EG: X607/X608, Wincon: S256/S512 ). The I-8xx7 and I-7188EG/XG/uPAC-7186EG supports old retain variable, while Wincon supports no retain variable. There is a 31-byte "NVSRAM" in the I-8xx7 & I-7188EG/XG/uPAC-7186EG 's CPU board . Up to **six Integers/Reals** (signed 32-bit) and **sixteen Booleans** can be retained with this 31-byte NVSRAM.

To enable the old retained function, click on "Retain" for each associated variable.



## Note:

If battery backup SRAM is found in the controller (I-8xx7: S256/S512, I-7188EG/XG/uPAC-7186EG: X607/X608, Wincon: S256/S512), Please use **new retain function** listed in the former section. **The old retain method has two disadvantage:** (1) The data will lost when download a modified ISaGRAF project. (2) Its retain variable amount is less than new method.

## 10.2: Data Backup To The EEPROM

Data can be stored into the EEPROM. The value will be always hold even the power is dead unless the value is updated. The EEPROM of I-8xx7, I-7188EG/XG/uPAC-7186EG & W-8xx7 controller can be read freely however can be written only about to 100,000 times. To read a value from the EEPROM, the following functions can be used.

|          |   |
|----------|---|
| EEP_B_R  | Read one boolean                                  |
| EEP_BY_R | Read one byte                                     |
| EEP_WD_R | Read one word (2 bytes, signed, -32768 to +32767) |
| EEP_N_R  | Read one integer (4 bytes, signed)                |
| EEP_F_R  | Read one REAL (4 bytes, float)                    |

( If the related data saved in the EEPROM is not REAL format, using “EEP\_F\_R” to read it may generate a controller local fault No. = 114. Please refer to Chapter 10.6 )

To write a value to the EEPROM, should remove the protection of the EEPROM first and then write operation is possible. The following functions can be used.

|          |  |
|----------|--|
| EEP_EN   | Remove the protection of EEPROM, then write operation is allowed.  |
| EEP_PR   | Set the protection of EEPROM, then write operation is not allowed. |
| EEP_B_W  | Write one boolean.   |
| EEP_BY_W | Write one byte (Byte: 0 to 255).                                   |
| EEP_WD_W | Write one word (2 bytes, signed, -32768 to +32767).                |
| EEP_N_W  | Write one integer (4 bytes, signed).                               |
| EEP_F_W  | Write one REAL (4 bytes, float).                                   |

Note:

1. “EEP\_F\_R”, “EEP\_N\_R”, “EEP\_N\_W” and “EEP\_F\_W” all use the same EEPROM memory, please DO NOT operate the same EEPROM address as Integer and also as REAL at the same time.
2. If the related data saved in the EEPROM is not REAL format, using “EEP\_F\_R” to read it may generate a controller local fault No. = 114. Please refer to Chapter 10.6

Bytes, words and integers will be stored to the same memory area in the EEPROM. Be careful to arrange their address before using the above write functions. There are total 1,512 bytes in the EEPROM memory area of the I-8xx7 & I-7188EG/XG/uPAC-7186EG, while much more in the W-8xx7.

**For I-8xx7 & I-7188EG/XG/uPAC-7186EG**, the addressing No. of bytes is range from 1 to 1,512, while words is 1 to 756, and integers is 1 to 378. The following No. will use the same memory address in the EEPROM.

|         |                      |                        |
|---------|----------------------|------------------------|
| Byte    | 4n-3, 4n-2, 4n-1, 4n | (* n = 1, 2, ...378 *) |
| Word    | 2n-1, 2n             |                        |
| Integer | n                    |                        |

**For W-8xx7**, the addressing No. of bytes is range from 1 to 14272, while words is 1 to 7136, and integers is 1 to 3568. The following No. will use the same memory address in the EEPROM.

|         |                      |                         |
|---------|----------------------|-------------------------|
| Byte    | 4n-3, 4n-2, 4n-1, 4n | (* n = 1, 2, ...3568 *) |
| Word    | 2n-1, 2n             |                         |
| Integer | n                    |                         |

**When using the write functions, the EEPROM will be damaged if the write operation is more than 100,000 times.** For example, the following program is dangerous since the EEPROM will be written once every PLC scan cycle (normally, the cycle time is about 3 to 60 ms depends on the application and controller model No.) .

|  |
|--|
| (* ST program, Val is declared as an integer, TEMP is declared as a boolean *)<br><b>TEMP := eep_n_w(1, Val) ; (* Dangerous *)</b> |
|--|

However the following program is safe if Val is not changed frequently.

|   |
|---|
| (* ST program, Val, Old_Val declared as integers, TEMP declared as a boolean *)<br><b>IF Val &lt;&gt; Old_Val THEN</b><br><b>TEMP := eep_n_w( 1 , Val ) ;</b><br><b>Old_Val := Val ;</b><br><b>END_IF ;</b> |
|---|

Each read / write operation in the EEPROM will consume a lot of CPU time of I-8xx7, I-7188EG/XG/uPAC-7186EG & W-8xx7 controller system. The following approximate time is for each function being called.

|                 |           |                 |           |
|-----------------|-----------|-----------------|-----------|
| <b>EEP_EN</b>   | ~ 0.08 ms | <b>EEP_PR</b>   | ~ 0.08 ms |
| <b>EEP_B_R</b>  | ~ 0.8 ms  | <b>EEP_B_W</b>  | ~ 6 ms    |
| <b>EEP_BY_R</b> | ~ 0.8 ms  | <b>EEP_BY_W</b> | ~ 6 ms    |
| <b>EEP_WD_R</b> | ~ 1.5 ms  | <b>EEP_WD_W</b> | ~ 12 ms   |
| <b>EEP_N_R</b>  | ~ 2.9 ms  | <b>EEP_N_W</b>  | ~ 23 ms   |

Recommend to read values from the EEPROM at one time when the I-8xx7, I-7188EG/XG/uPAC-7186EG & W-8xx7 is powered up, and then updated the associated address in the EEPROM when the value is changed. Please refer to a sample program in Chapter 11 – “**demo\_17**” & “**Wdemo\_10**”. For those data which are frequently changed are not suitable to be stored in the EEPROM.



## 10.3: Battery Backup SRAM

**Note:** Read floating point value from S-256/512 & X607/608 may cause controller fault if no floating point value saved inside. Please refer to Section 10.6 – “Controller Fault Detection”

The I-8xx7 and W-8xx7 controllers can integrate with a S256 or S512 battery backup SRAM to store data, alarm, and information, while X607 & X608 for the I-7188EG/XG/uPAC-7186EG controller. The data stored in these SRAM is always retained unless their battery running out of energy. Their memory size is as below, however the upper 12K is reserved by I-8417/8817/8437/8837 and I-7188EG/XG/uPAC-7186EG, while 64K is reserved by W-8337/8737/8336/8736.

| I-8417/8817/8437/8837, i-8437-80 / i-8837-80 |                         | I-7188EG/XG/uPAC-7186EG |                         |
|--|-------------------------|-------------------------|-------------------------|
| S256   | 244K bytes (256-12=244) | X607                    | 116K bytes (128-12=116) |
| S512   | 500K bytes (512-12=500) | X608                    | 500K bytes (512-12=500) |
| W-8337/8737/8336/8736                        |                         |                         |                         |
| S256   | 192K bytes (256-64=192) |                         |                         |
| S512   | 448K bytes (512-64=448) |                         |                         |

If battery backup SRAM is found in the controller, the maximum number of retained variables for new retain function “Retain\_X”, “Retain\_A”, “Retain\_B”, “Retain\_N”, “Retain\_F” & “Retain\_T” can be extend to as below (please refer to Section 10.1).

| I-7188EG/XG/uPAC-7186EG+X607/608 and I-8xx7+S256/512      |                     |                          |
|---|---------------------|--------------------------|
|   | New Retain function | old retain method        |
| Boolean   | 256                 | 256                      |
| Integer   | 1024                | 256<br>(Integer + Real)  |
| Real  | 1024                |                          |
| Timer   | 256                 | 32                       |
| W-8xx7+S256/512 with new Wincon back-plane (section 10.1) |                     |                          |
|   | New Retain function | old retain method        |
| Boolean   | 1024                | 1024                     |
| Integer   | 4096                | 4096<br>(Integer + Real) |
| Real  | 4096                |                          |
| Timer   | 1024                | 1024                     |

ICP DAS provides an utility “**ICPDAS UDloader**” that can be installed in the PC to upload and download data from/to the ISaGRAF controller. Please copy “**UDloader.exe**” from the ICP DAS’s CD-ROM:\napdos\isagraf\some\_utility\ to your PC / windows.

The I-8417/8817/8437/8837 supports S256/S512 since its driver version of 2.25 (better to be 3.19 or later), while I-7188EG supports X607/608 since its driver version of 1.18 (better to be 2.17 or later), and version 1.16 for I-7188XG (better to be 2.15 or later). W-8337/8737/8336/8736 supports S256/S512 since its driver version 3.18 (better to be 3.36 or later) (Please refer to section 10.1). If your driver is older one, please upgrade the hardware driver to the associate version or a higher version. The driver can be found from the ICP DAS’s web site: <http://www.icpdas.com/products/PAC/i-8000/isagraf.htm>

### 10.3.1: Access to the SRAM

The SRAM can store boolean, byte, word, integer, real & message. Their format is as below.  
(Please refer to Chapter 11.3.7 for a demo program using S-256/512 by UDloader.exe)

|          |                   |           |
|----------|-------------------|-----------|
| Boolean: | True=1, False=0   | 1 byte    |
| Byte:    | 0 ~ 255           | 1 byte    |
| Word:    | -32768 ~ 32767    | 2 bytes   |
| Integer: | signed 32-bit     | 4 bytes   |
| Real:    | float             | 4 bytes   |
| Message: | string (len<=255) | len bytes |

To access to the SRAM, the below functions can be used (Please refer to Appendix A).

S\_B\_R,      S\_B\_W,      S\_BY\_R ,      S\_BY\_W,      S\_M\_R,      S\_M\_W  
S\_WD\_R,    S\_WD\_W,    S\_N\_R ,      S\_N\_W,      S\_R\_R,      S\_R\_W  
S\_MV

### 10.3.2: Upload data stored in the SRAM

For PC to upload data stored in the volatile SRAM of the ISaGRAF controllers, the SRAM should be divided into 1 or up to 8 files. Each file has a ID No. of 1 to 8 and a name of up to 12 characters. The below functions are for handling file format inside the SRAM.

S\_FL\_INI,    S\_FL\_AVL,    S\_FL\_RST,    S\_FL\_STS

Please use functions of S\_FL\_INI & S\_FL\_AVL to arrange the file resident location & current available location (Please refer to Appendix A & demo\_40, 41 or 42).

The volatile SRAM is consisted of bytes. The total number of bytes available depends on which module is used as below. The upper 12K is reserved.

| Module name                   | Byte No.                                      |
|-------------------------------|---|
| I-8xx7: S256                  | 1 ~ 249,856 (244K), (256-244=12K is reserved) |
| I-8xx7: S512                  | 1 ~ 512,000 (500K), (512-500=12K is reserved) |
| I-7188XG/EG/uPAC-7186EG: X607 | 1 ~ 118,784 (116K), (128-116=12K is reserved) |
| I-7188XG/EG/uPAC-7186EG: X608 | 1 ~ 512,000 (500K), (512-500=12K is reserved) |

A file can be located at any place inside these bytes. Each file's location can be described as (**Begin, End**). Begin is the lower limit byte No. of the associated file, while End is the upper limit byte No., and Begin is always less than End.

A file inside the SRAM has a current available area (**Head, Tail**). Head is the starting position of the file, Tail is the ending position. Head can be larger, less than or equal to Tail.

For ex, a file resides at (Begin, End) = (1, 20000)

1. If (Head, Tail) = (1001,5100), it means the available data of the file is starting from byte No. of 1001 to 5100. The available file contains 4100 bytes.
2. If (Head, Tail) = (10001,5000), it means the available data of the file is starting from byte No. of 10001 to 20000 and then continued with 1 to 5000. The available file contains 15000 bytes.
3. If (Head, Tail) = (5001,5000), it means the available data of the file is starting from byte No. of 5001 to 20000 and then continued with 1 to 5000. The available file contains 20000 bytes.

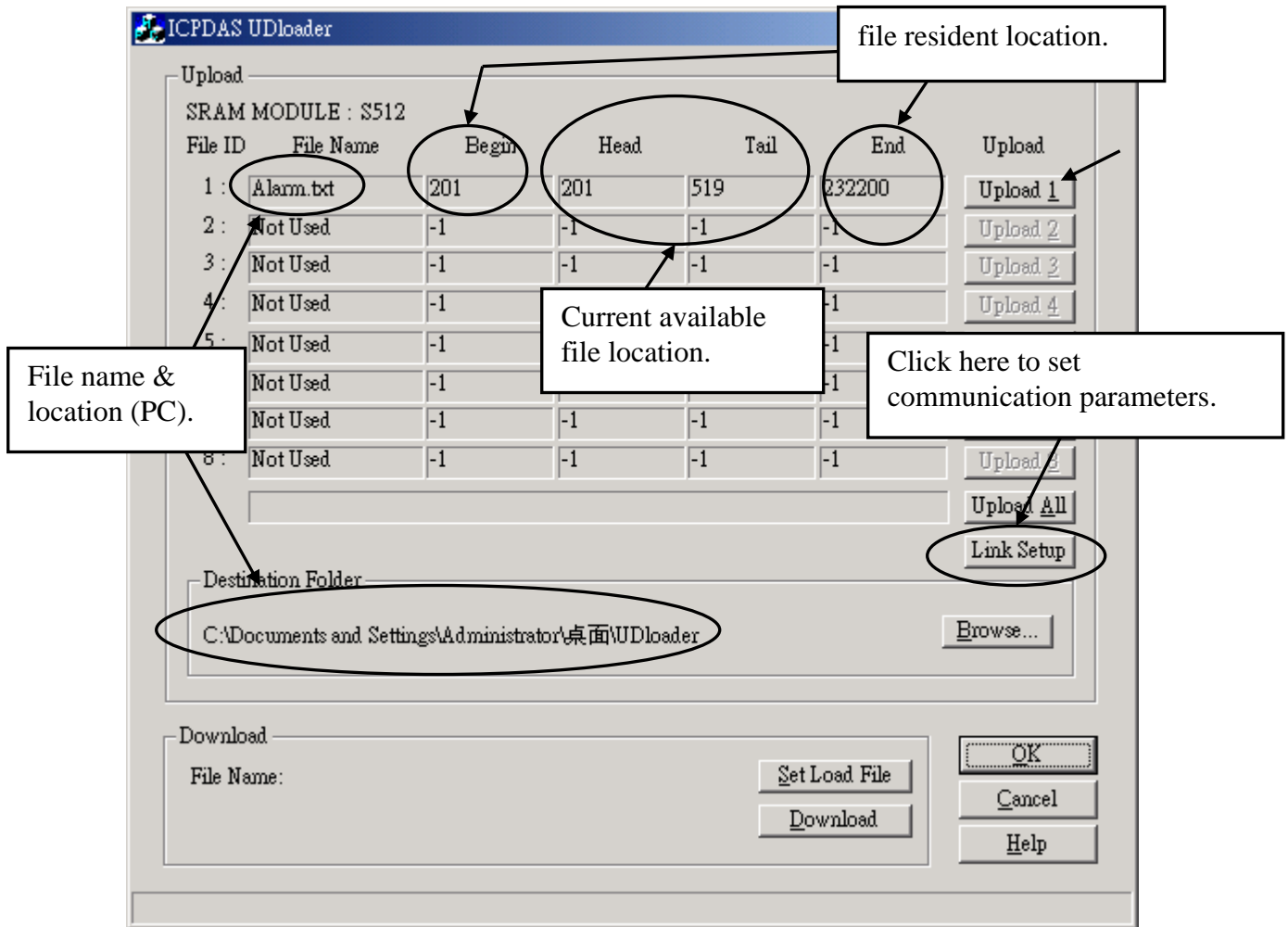
4. If (Head, Tail) = (5000,5000), it means the available data of the file is empty, 0 byte.
5. If (Head, Tail) = (-1,-1), it means the available data of the file is empty, 0 byte.

To upload the data stored in the SRAM, please make sure you have installed the “ICPDAS UDloader” on your PC.

To upload data stored in the SRAM of the ISaGRAF controller to PC, please run “UDloader.exe”, then click on “Link Setup” to set proper communication parameters, then click on “Upload 1” to upload it.

Example:

Please download demo\_41 to one I-8417/8817/8437/8837. Then push button 1 or 2 or 3 or 4 several times. Then upload the file stored in the SRAM.



### 10.3.3: Download data to the SRAM

For PC to download data to the volatile SRAM of the ISaGRAF controllers. The below functions can be used. Please refer to Appendix A & demo\_44.

S\_DL\_EN, S\_DL\_DIS, S\_DL\_RST, S\_DL\_STS

Please call “S\_DL\_EN” to enable it.

The Controller accepts only the binary format for String, Byte, Word, Int & Real.

|         |                        |                 |                                |
|---------|------------------------|-----------------|--------------------------------|
| Byte:   | 0 ~ 255                | 1 byte          |                                |
| Word:   | -32768 ~ +32767        | 2 byte          | [low bye] [high byte]          |
| Int:    | 32-bit, signed integer | 4 byte          | [lowest] [2nd] [3rd] [highest] |
| Real:   | 32-bit float           | 4 byte          | [lowest] [2nd] [3rd] [highest] |
| String: |                        | up to 255 bytes |                                |

If using the “UDloader.exe” to download data to the volatile SRAM, the data to be downloaded should be edited as a text file. Its format should follow the below rules.

The first raw should be a No. indicate that to download to which starting Byte No. of the SRAM. Valid starting byte No is as below.

|       |             |       |            |
|-------|-------------|-------|------------|
| S256: | 1 ~ 249,856 | S512: | 1 ~ 512000 |
| X607: | 1 ~ 118,784 | X608: | 1 ~ 512000 |

The other raws are the data.

#### A. String

String should start and end with the character of ‘ ’, for ex. ‘Abcd123’ (7 byte). The \$NN (NN in hexadecimal and should not equal to 0), could be used to indicate the ASCII character. For ex, ‘ABC\$0D’ contains 4 bytes, the 4th byte is <CR>.

#### B. Byte

Byte should start with ( and end with ) , for ex. (0) , (123), (255). Valid byte range is from (0) to (255).

#### C. Word

Word should be start with [ and end with ] , for ex. [-100] , [20000], [32767]. Valid word range is from [-32768] to [32767].

#### D. Integer

Integer should be start with { and end with } , for ex. {-1234567} , {200000}. Valid integer range is from {-2147483648} to {2147483647}.

#### E. Real

Real value should be start with < and end with > , for ex. <123> , <1.56E-2>, <-123.456>.

3. The character between each Byte, Word, Integer, Real, String at the same line should be at least one space character <SP> or , <Comma> or, <Tab>

For ex.

|   |
|---|
| 201 ← to download to the SRAM which staring from byte No. 201<br>‘Hello’ (10) (20) (30) (40) [-10000] {70000} ‘End’ ← data (total 18 bytes) |
|---|

1      ← to download to the SRAM which starting from byte No. 1  
 (23) ← data (total 57 bytes)  
 {-1},{2},{-3},{4},{-5},{6} {-7} {8} {-9} {10} ← comma, <SP> & <Tab> are all acceptable  
 <0.123> <456.789> <100> , <2.3E3>

Example:

Please download demo\_44 to one I-8417/8817/8437/8837. Then edit a text file as below.

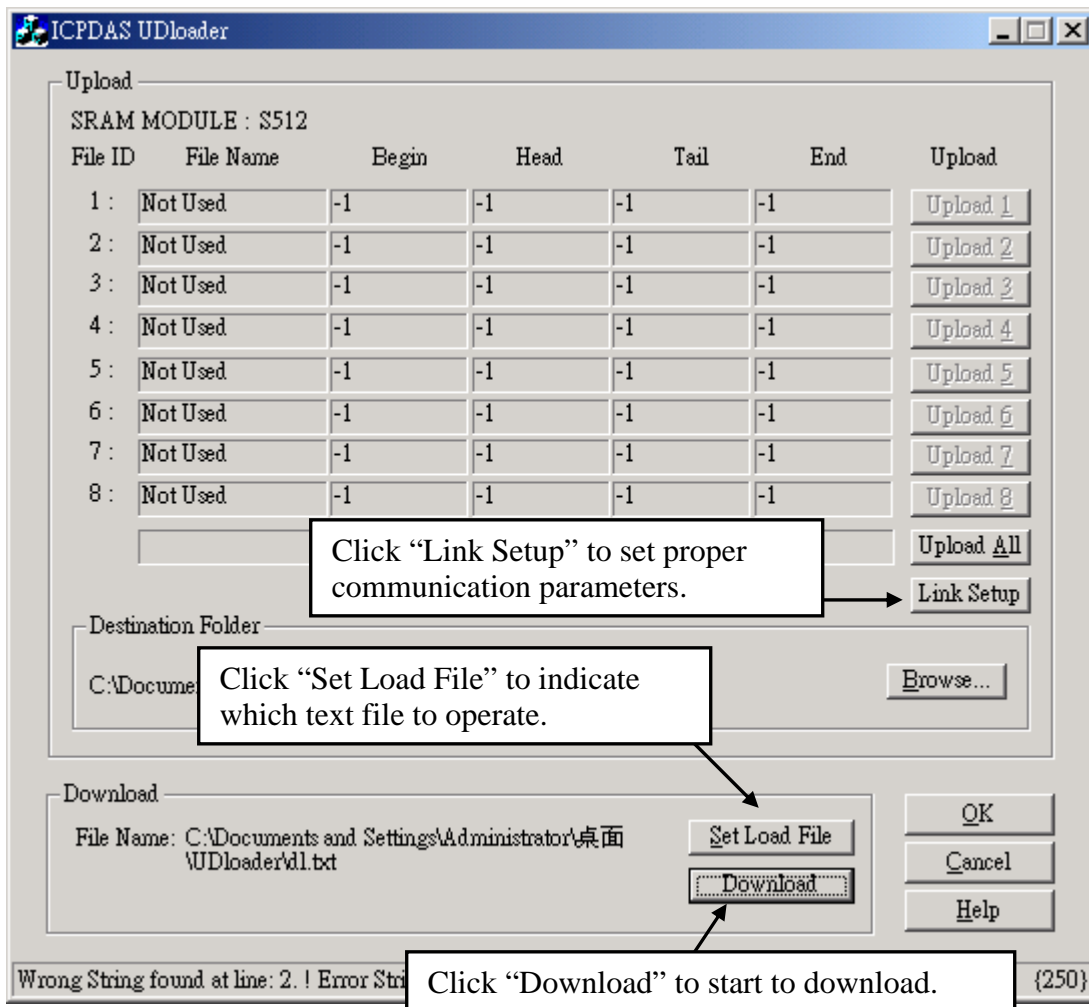
```
1
{1000} {250} {100} 'sTART'
```

The {1000} means the blinking period of L1 is 1000 ms.

The {250} means the blinking period of L2 is 250 ms.

The {100} means the blinking period of L3 is 100 ms. .

Then run “UDloader.exe”. You will see something change on the led of the controller.



### 10.3.4: Operation Functions for the battery backup SRAM

The below functions are for the ISaGRAF controller to access to the volatile SRAM. More information listed at Appendix A.4

S\_FL\_INI     Init one file's name & location for the volatile SRAM  
S\_FL\_AVL     Set one file's current available byte No. for the volatile SRAM  
S\_FL\_STS     Get file's Status, end byte No. that has been load by PC for the volatile SRAM  
S\_FL\_RST     Reset file's Status to "Not been load by PC yet" for the volatile SRAM

S\_B\_R:       Read one Boolean (TRUE, FALSE)  
S\_BY\_R:       Read one Byte (0 ~ 255)  
S\_WD\_R:       Read one Word (-32768 ~ +32767)  
S\_N\_R:       Read one Integer (32 bit, signed)

S\_R\_R:       Read one Real (32 bit, float)  
(If the data in related address of the battery backup SRAM is not a REAL value, using "S\_R\_R" to read it may generate a controller local fault No. = 102. please refer to Chapter 10.6)

S\_M\_R:       Read one String

S\_B\_W:       Write one Boolean (TRUE, FALSE)  
S\_BY\_W:       Write one Byte (0 ~ 255)  
S\_WD\_W:       Write one Word (-32768 ~ +32767)  
S\_N\_W:       Write one Integer (32 bit, signed)  
S\_R\_W:       Write one Real value (32 bit, float)  
S\_M\_W:       Write one String

S\_DL\_EN     Enable the download permission for PC to download data to the volatile SRAM  
S\_DL\_DIS     Disable the download permission for PC to download data to the volatile SRAM  
S\_DL\_STS     Get PC's Download Status for the volatile SRAM  
S\_DL\_RST     Reset the Download Status to "-1:No action" for the volatile SRAM

S\_MV        copy data in the volatile SRAM

## 10.4: Using I-8073 - MultiMediaCard to store data

---

The I-8072 / 8073 is not support by I-8xx7, iPAC-8447/8847, I-7188EG/XG/uPAC-7186EG & W-8xx7.

## 10.5: Reading & Writing File in Wincon

### Note:

1. If the data type in the related file position is not REAL type (32-bit float format), using “F\_READ\_F” function to read this data may generate a local controller fault No = 117 (please refer to Chapter 10.6).
2. Only Wincon-8xx7 / 8xx6 support File operating functions. Not for I-8xx7 and I-7188EG/XG.
3. If the file path is inside the ‘\CompactFlash\’ folder, for example- ‘\CompactFlash\data1.txt’, the file will be stored in the Compact Flash card in Wincon and this file will exist even the Wincon’s power is switched Off. However it is very slow to Read / Write file in the Compact Flash card.
4. If the file path is not in the ‘\CompactFlash\’ folder, for example – ‘\Temp\data2.txt’, it will be stored in the RAM memory of the Wincon. The file saved in Wincon’s RAM will be lost when power is switched OFF. The advantage of Reading / Writing file in the RAM memory is it is much faster than operating in the Compact Flash card.

W-8xx7 supports below ISaGRAF standard functions. (more information listed at Appendix A.4)

|          |   |
|----------|---|
| F_OPEN   | Open file in Binary format for read operation (file should exist already) . |
| F_WOPEN  | Open file in Binary format for read and write operation (file should exist) |
| F_CLOSE  | Close a file.   |
| F_EOF    | Test if reach the End-Of-File position.                                     |
| FA_READ  | Read one binary long integer (4-bytes, signed) from file.                   |
| FA_WRITE | Write one binary long integer (4-bytes, signed) to file.                    |
| FM_READ  | Read one message (string) from file.  |
| FM_WRITE | Write one message (string) with <CR> <LF> char. at end of message to file.  |

W-8xx7 supports below ICP DAS c-functions.

|          |  |
|----------|--|
| F_APPEND | Append one file to the end of the other file.                                  |
| F_COPY   | Copy one file to another file.   |
| F_CREAT  | Create a new file.   |
| F_DELETE | Delete a file.   |
| F_DIR    | Create a new directory (folder).   |
| F_END    | move current file position to the End-Of-File position.                        |
| F_SEEK   | Move current file position to a specified position.                            |
| F_READ_B | Read one binary byte (0 - 255) (1 byte, unsigned) from file.                   |
| F_WRIT_B | Write one binary byte (0 - 255) (1 byte, unsigned) to file.                    |
| F_READ_W | Read one binary word (-32768 to +32767) (2 byte, signed) from file .           |
| F_WRIT_W | Write one binary word (-32768 to +32767) (2 byte, signed) to file.             |
| F_READ_F | Read one binary REAL (4-bytes, Float) from file. Like 123.45, -2.15E-03, ...   |
| F_WRIT_F | Write one binary REAL (4-bytes, Float) to file .                               |
| F_WRIT_S | Write one message <b>without</b> <CR> <LF> char. At the end of string to file. |

Please refer to below for demo program.

Wincon CD-ROM: \napdos\isagraf\wincon\demo\ “wdemo\_54.pia” , 55, 56, 51, 50, 1 or 2 or  
[ftp://ftp.icpdas.com/pub/cd/wincon\\_isagraf/napdos/isagraf/wincon/demo/](http://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/) or  
[www.icpdas.com](http://www.icpdas.com) – FAQ – Software –ISaGRAF - 057 & 060

### 10.5.1: Wdemo\_51: Read 10 REAL values from a file. Total 10 rows, each contains one REAL value

The “Wdemo\_51.pia” can be found at

W-8xx7 CD-ROM:\napdos\isagraf\wincon\demo\ or

[ftp://ftp.icpdas.com/pub/cd/wincon\\_isagraf/napdos/isagraf/wincon/demo/](ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/)

If functions of Msg\_F , Msg\_N , ARY\_F\_R, AFY\_F\_W are not found in your PC / ISaGRAF, please download “ICP DAS utilities For ISaGRAF” at

<http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> . Then run “setup.exe” to restore them to your ISaGRAF workbench

The “Wdemo\_51” program will read 10 REAL values from “\CompactFlash\data51.txt” when the W-8xx7 is just power up or user set the “RE\_LOAD” value to become “TRUE” at any time .

(To read / write file in the CompactFlash card take lots of CPU time, please do not read / write it frequently. And please always close the file after the operation. If user read / write file in every PLC scan cycle, the PLC scan time will become a very big time. It will be a bad performance !)

To test this sample program, please edit a text file “data51.txt” in your PC by , for example – “Notepad”. Please enter 10 rows, each contains one Real value. Then download this “data51.txt” to Wincon’s “\CompactFlash\” folder by “ftp” utility. The content of the “data51.txt” looks like below.

```
2.345
999.03
-1.01
456.789
2
456.77
5.9E-12
32.3
45.1
33.3
```

Variables:

| <b>Name</b> | <b>Type</b> | <b>Attribute</b> | <b>Description</b>                            |
|-------------|-------------|------------------|---|
| RE_LOAD     | Bool        | Internal         | Set as True to read File once, init as TRUE   |
| TMP         | Bool        | Internal         | Internal use                                  |
| File_name1  | Message     | Internal         | Len is 64, init as \CompactFlash\data51.txt   |
| Msg1        | Message     | Internal         | Len is 128, File processing state             |
| str1        | Message     | Internal         | Len is 255, internal use                      |
| F_VAL[0..9] | REAL        | Internal         | Variable array, Dim is 10. The 10 REAL value. |
| TMP_F       | REAL        | Internal         | Internal use                                  |
| File1       | Integer     | Internal         | File ID                                       |
| ii          | Integer     | Internal         | Index of “for” loops                          |



ST program:

```
if RE_LOAD then      (* Read file once if “RE_LOAD” is TRUE *)

RE_LOAD := FALSE ;

File1 := f_wopen( File_name1 ) ;      (* Open file in Read & Write mode *)

if File1 = 0 then      (* 0: open file fail *)
  Msg1 := 'Can not Open file ' + File_name1 ;
  return ;      (* Cannot open file, just exit this ST program *)
end_if ;

for ii := 0 to 9 do      (* Total 10 rows *)

  if f_eof(File1) = TRUE then      (* test if reaches the End-Of-File *)
    Msg1 := 'Data number is not enough in ' + File_name1 ;
    Exit ;      (* Exit this “for” loops *)
  end_if ;

  str1 := fm_read(File1) ;      (* read one string in the File *)
  TMP_F := str_real(str1) ;      (* convert string to a REAL value *)
  if TMP_F = 1.23E-20 then      (* if returns 1.23E-20, it means format error *)
    Msg1 := 'The ' + Msg(ii+1) + 'th Data format is not correct !' ;
    exit ;      (* Exit this “for” loops *)
  end_if ;

  F_VAL[ii] := TMP_F ;      (* Read & Convert Ok. Store value to F_VAL[0..9] *)

end_for ;

TMP := f_close(File1) ;      (* always close File after its operation *)
If ii=10 then      (* All data is successfully read and converted, 10 rows *)
  Msg1 := 'Read ' + File_name1 + ' Ok ' ;
end_if ;

end_if ;
```

### 10.5.2: Wdemo\_54: Read 20 REAL values from a file. Total 4 rows, each contains 5 REAL values

The “Wdemo\_54.pia” can be found at

W-8xx7 CD-ROM:\napdos\isagraf\wincon\demo\ or

[ftp://ftp.icpdas.com/pub/cd/wincon\\_isagraf/napdos/isagraf/wincon/demo/](ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/)

If functions of Msg\_F , Msg\_N , ARY\_F\_R, AFY\_F\_W are not found in your PC / ISaGRAF, please download “ICP DAS utilities For ISaGRAF” at

<http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> . Then run “setup.exe” to restore them to your ISaGRAF workbench

The “Wdemo\_54” program will read 20 REAL values from “\CompactFlash\data54.txt” when the W-8xx7 is just power up or user set the “RE\_LOAD” value to become “TRUE” at any time .

(To read / write file in the CompactFlash card take lots of CPU time, please do not read / write it frequently. And please always close the file after the operation. If user read / write file in every PLC scan cycle, the PLC scan time will become a very big time. It will be a bad performance !)

To test this sample program, please edit a text file “data54.txt” in your PC by , for example – “Notepad”. Please enter 4 rows, each contains 5 Real values. Then download this “data54.txt” to Wincon’s “\CompactFlash\” folder by “ftp” utility. The content of the “data54.txt” looks like below.

```
23 , 65.9 , 0.012 , 5.87 , 88.2
0.34 , 8.0005 , -2.0E8 , 4.08 , 5.32E-6
2 , -7 , 6666.8 , 456.07 , 1.01
5 , 6 , 7 , 8 , 9
```

Variables:

| <b>Name</b>  | <b>Type</b> | <b>Attribute</b> | <b>Description</b>                            |
|--------------|-------------|------------------|---|
| RE_LOAD      | Bool        | Internal         | Set as True to read File once, init as TRUE   |
| TMP          | Bool        | Internal         | Internal use                                  |
| File_name1   | Message     | Internal         | Len is 64, init as \CompactFlash\data54.txt   |
| Msg1         | Message     | Internal         | Len is 128, File processing state             |
| str1         | Message     | Internal         | Len is 255, internal use                      |
| F_VAL[0..19] | REAL        | Internal         | Variable array, Dim is 20. The 20 REAL value  |
| NUM1         | Integer     | Internal         | Get return of Msg_F( ), -1 means format error |
| File1        | Integer     | Internal         | File ID                                       |
| ii           | Integer     | Internal         | Index of “for” loops                          |
| jj           | Integer     | Internal         | Index of another “for” loops                  |

ST program:

```
if RE_LOAD then      (* Read file once if "RE_LOAD" is TRUE *)

RE_LOAD := FALSE ;

File1 := f_wopen( File_name1 ) ;    (* Open file in Read & Write mode *)

if File1 = 0 then    (* 0: open file fail *)
  Msg1 := 'Can not Open file ' + File_name1 ;
  return ;          (* Cannot open file, just exit this ST program *)
end_if ;

for ii := 0 to 3 do    (* total 4 rows *)

  if f_eof(File1) = TRUE then    (* test if reaches the End-Of-File *)
    Msg1 := 'There should be at least 4 rows in ' + File_name1 + ' !!!' ;
    Exit ;          (* exit this "for" loops *)
  end_if ;

  str1 := fm_read(File1) ;    (* read one one row as string from file *)

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

  (* If the amount of the converted REAL values is not 5 , it lacks of data. -1 means format error *)
  if NUM1 <> 5 then
    Msg1 := 'The ' + Msg(ii+1) + 'th row data format is not correct or data number is not 5 !' ;
    Exit ;          (* exit this "for" loops *)
  end_if ;

  for jj := 0 to 4 do
    (* Get 5 REAL values from No.1 Float array's addr=1 to 5 , and store them to F_VAL[0..19] *)
    F_VAL[ 5 * ii + jj] := ARY_F_R( 1 , jj + 1 ) ;
  end_for ;

end_for ;

TMP := f_close(File1) ;    (* always close File after its operation *)

If ii = 4 then    (* All data is successfully read and converted, 4 rows *)
  Msg1 := 'Read ' + File_name1 + ' Ok ' ;
end_if ;

end_if ;
```

### 10.5.3: Wdemo\_55: Read 20 Integer values from a file. Total 2 rows, each contains 10 Integer values

The “Wdemo\_55.pia” can be found at

W-8xx7 CD-ROM:\napdos\isagraf\wincon\demo\ or

[ftp://ftp.icpdas.com/pub/cd/wincon\\_isagraf/napdos/isagraf/wincon/demo/](ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/)

If functions of Msg\_F , Msg\_N , ARY\_F\_R, AFY\_F\_W are not found in your PC / ISaGRAF, please download “ICP DAS utilities For ISaGRAF” at

<http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> . Then run “setup.exe” to restore them to your ISaGRAF workbench

The “Wdemo\_55” program will read 20 Integer values from “\CompactFlash\data55.txt” when the W-8xx7 is just power up or user set the “RE\_LOAD” value to become “TRUE” at any time .

(To read / write file in the CompactFlash card take lots of CPU time, please do not read / write it frequently. And please always close the file after the operation. If user read / write file in every PLC scan cycle, the PLC scan time will become a very big time. It will be a bad performance !)

To test this sample program, please edit a text file “data55.txt” in your PC by , for example – “Notepad”. Please enter 2 rows, each contains 10 Integer values. Then download this “data55.txt” to Wincon’s “\CompactFlash\” folder by “ftp” utility. The content of the “data55.txt” looks like below.

|   |
|---|
| -1 , 1 , 2 , 3 , 4 , 5 , -6 , 7 , 8 , 9<br>100001 , 200002 , +300003 , 404 , -505 , 606 , 7007 , 8008 , 9009 , 10 |
|---|

Variables:

| <i><b>Name</b></i> | <i><b>Type</b></i> | <i><b>Attribute</b></i> | <i><b>Description</b></i>                        |
|--------------------|--------------------|-------------------------|--|
| RE_LOAD            | Bool               | Internal                | Set as True to read File once, init as TRUE      |
| TMP                | Bool               | Internal                | Internal use                                     |
| File_name1         | Message            | Internal                | Len is 64, init as \CompactFlash\data55.txt      |
| Msg1               | Message            | Internal                | Len is 128, File processing state                |
| str1               | Message            | Internal                | Len is 255, internal use                         |
| N_VAL[0..19]       | Integer            | Internal                | Variable array, Dim is 20. The 20 Integer values |
| NUM1               | Integer            | Internal                | Get return of Msg_N( ), -1 means format error    |
| File1              | Integer            | Internal                | File ID  |
| ii                 | Integer            | Internal                | Index of “for” loops                             |
| jj                 | Integer            | Internal                | Index of another “for” loops                     |

ST program:

```
if RE_LOAD then    (*Read file once if "RE_LOAD" is TRUE *)

RE_LOAD := FALSE ;

File1 := f_wopen( File_name1 ) ;    (* Open file in Read & Write mode *)

if File1 = 0 then    (* 0: open file fail *)
  Msg1 := 'Can not Open file ' + File_name1 ;
  return ;    (* Cannot open file, just exit this ST program *)
end_if ;

for ii := 0 to 1 do    (* total 2 rows *)

  if f_eof(File1) = TRUE then    (*test if reaches the End-Of-File *)
    Msg1 := 'There should be at least 2 rows in ' + File_name1 + ' !!!' ;
    Exit ;    (* exit this "for" loops *)
  end_if ;

  str1 := fm_read(File1) ;    (* read one one row as string from file *)

  (* Convert string to become serval Integer values and store them into No. 2 Integer array *)
  NUM1 := Msg_N(str1 , 2) ;

  (* If the amount of the converted Integer values is not 10 , it lacks of data. -1 means format error *)
  if NUM1 <> 10 then
    Msg1 := 'The ' + Msg(ii+1) + 'th row data format is not correct or data number is not 10 !' ;
    Exit ;    (* exit this "for" loops *)
  end_if ;

  for jj := 0 to 9 do

    (* Get 10 Integer values from No.2 Integer array's addr=1 to 10 , and store them to N_VAL[0..19] *)
    N_VAL[ 10 * ii + jj] := ARY_N_R( 2 , jj + 1 ) ;
  end_for ;

end_for ;

TMP := f_close(File1) ;    (*always close File after its operation *)

If ii = 2 then    (*All data is succefully read and converted, 2 rows *)
  Msg1 := 'Read ' + File_name1 + ' Ok ' ;
end_if ;

end_if ;
```

## 10.5.4: Wdemo\_56: Retain values of 1 to 255 Real variable in CompactFlash card

The “Wdemo\_56.pia” can be found at

W-8xx7 CD-ROM:\napdos\isagraf\wincon\demo\ or

[http://ftp.icpdas.com/pub/cd/wincon\\_isagraf/napdos/isagraf/wincon/demo/](http://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/)

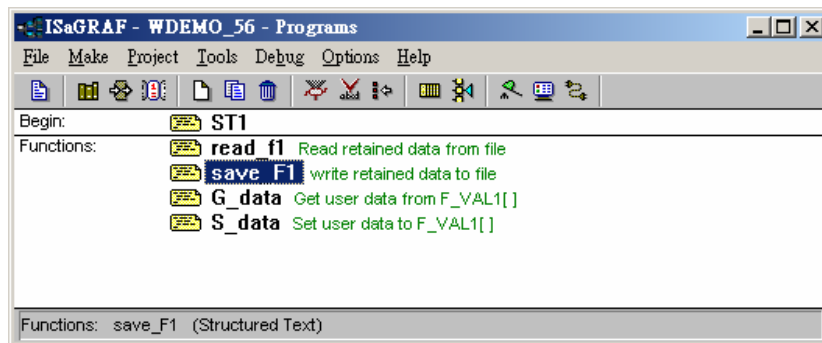
If functions of Msg\_F , Msg\_N , ARY\_F\_R, AFY\_F\_W are not found in your PC / ISaGRAF, please download “ICP DAS utilities For ISaGRAF” at

<http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> . Then run “setup.exe” to restore them to your ISaGRAF workbench

The “Wdemo\_56” program will read 1 to 255 REAL values from “\CompactFlash\data56.txt” to the related variable when the W-8xx7 is just power up . If this “data56.txt” doesn’t exist, all 1 to 255 values will be init as 0.0 . At run time, if any value of these variable is modified, all 1 to 255 values will be written once to the “data56.txt” to make sure these variable’s value are well retained in file. If the file doesn’t exist, this program will create it.

To read / write file in the CompactFlash card take lots of CPU time, please do not read / write it frequently. And please always close the file after the operation. If user read / write file in every PLC scan cycle, the PLC scan time will become a very big time. It will be a bad performance ! If user need fast retain function, please refer to Chapter 10.1 for retaining data in the S256/S512.

Project Architecture:

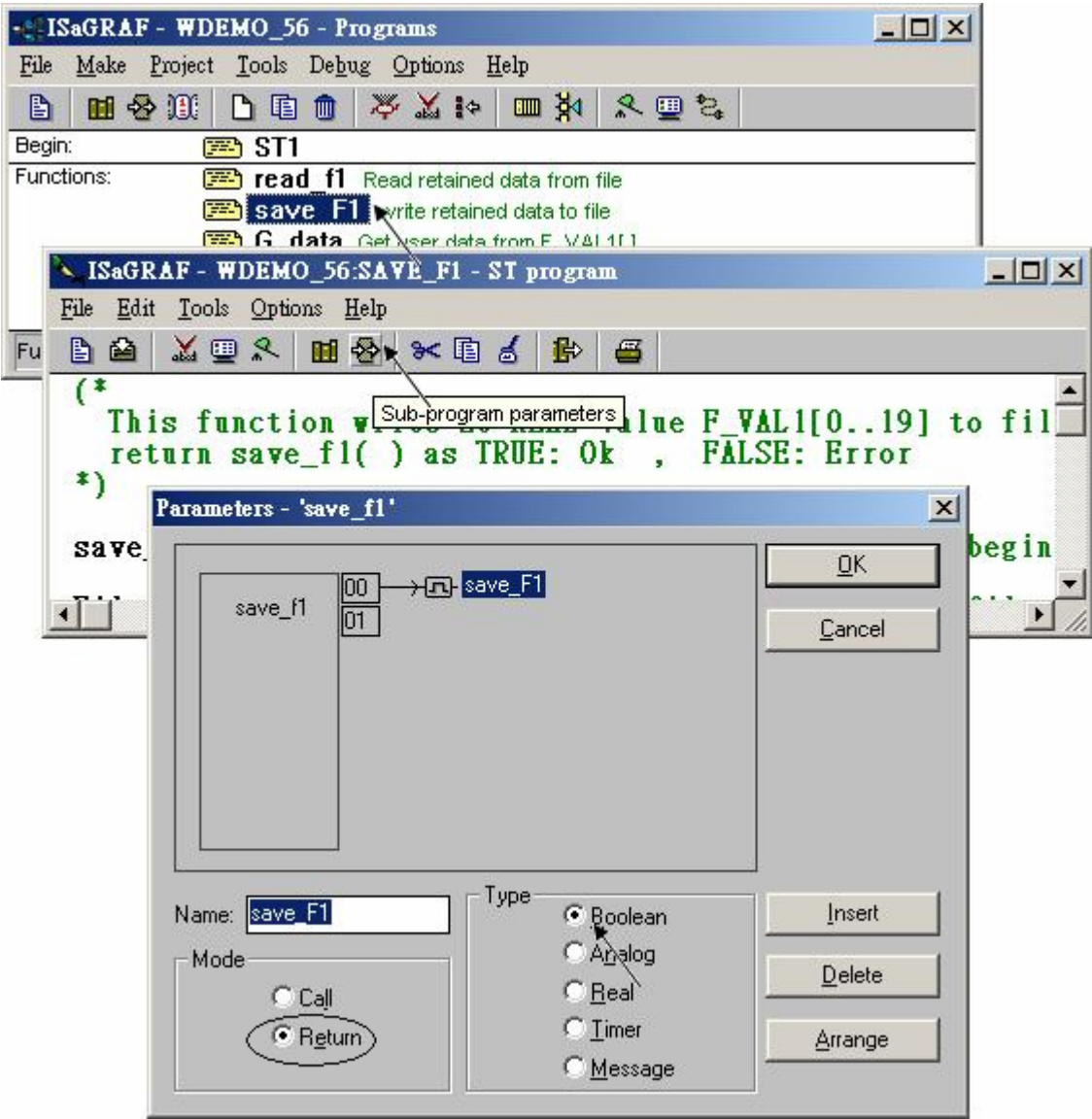


There are 5 ST programs in this “wdemo\_56” project. Four of them are ISaGRAF user-defined functions – “read\_f1”, “save\_f1”, “G\_data” and “S\_data” .

### Important note:

1. User may modify the constant value of “SIZE1” in the ISaGRAF “dictionary” window to a value between 1 to 255 according his own application. And then remember to compile it .
2. Please also modify the “Dim” value of the “F\_VAL1[ ]” and “Old\_F\_VAL1[ ]” variable array in the ISaGRAF “dictionary” window to become the same value as the “SIZE1” . And also please modify the “G\_data” and “S\_data” program.
3. There is one advantage of retaining vale in the CompactFlash card. The data file can be edited in advance in PC. Then using “ftp” utility to download it to Wincon. The file path name of this example is “\CompactFlash\data56.txt” . Then set “RE\_LOAD” value to TRUE once, all related variable will update to the new value.

The following ST programs are all declared as functions. They are “read\_f1”, “save\_f1”, “G\_data” and “S\_data”. They all return a Boolean value. Please refer to below figure to declare function’s return-value type (more description in the Chapter 15)



The “read\_f1” and “save\_f1” program use “local variables” as below .

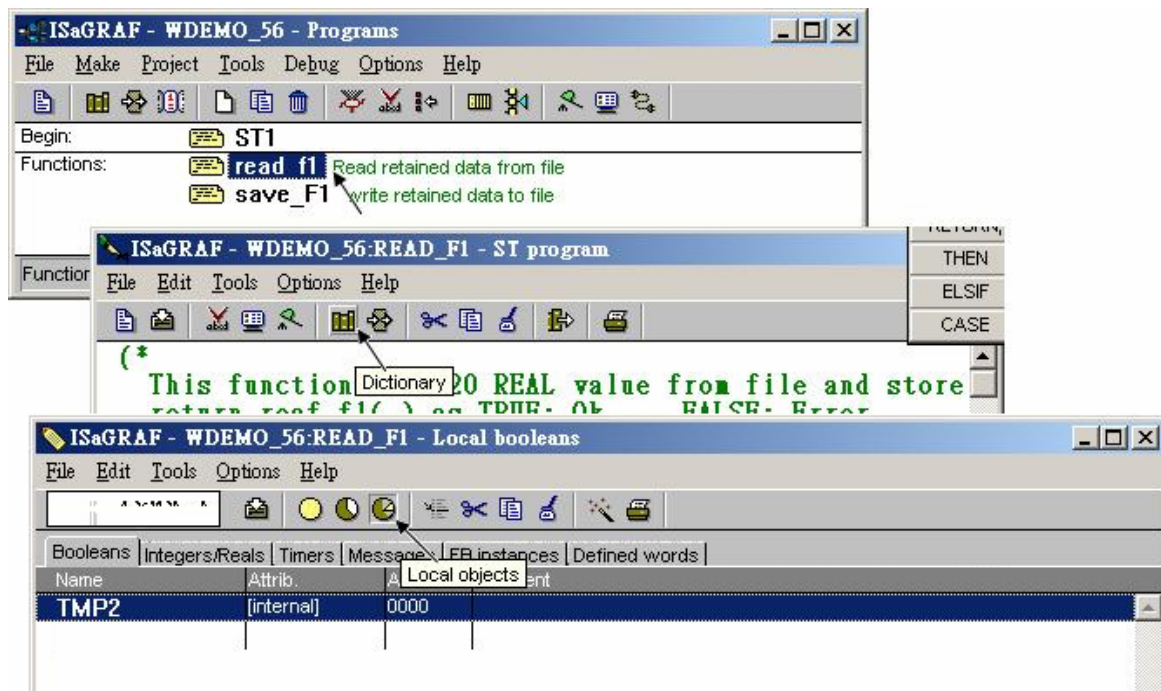
read\_f1 :

| <b>Name</b> | <b>Type</b> | <b>Attribute</b> | <b>Description</b>   |
|-------------|-------------|------------------|----------------------|
| TMP2        | Bool        | Internal         | Internal use         |
| ii2         | Integer     | Internal         | Index of “for” loops |
| jj2         | Integer     | Internal         | Index of “for” loops |
| num2        | Integer     | Internal         | Internal use         |

save\_f1 :

| <b>Name</b> | <b>Type</b> | <b>Attribute</b> | <b>Description</b>   |
|-------------|-------------|------------------|----------------------|
| TMP2        | Bool        | Internal         | Internal use         |
| ii2         | Integer     | Internal         | Index of “for” loops |
| jj2         | Integer     | Internal         | Index of “for” loops |
| num2        | Integer     | Internal         | Internal use         |

To declare “local variable”, please double click “read\_f1” to get into this program. Then get into the “Dictionary” window. Then click on “Local objects” to declare them.





Variables (Global variable) :

| <b>Name</b>                             | <b>Type</b> | <b>Attribute</b> | <b>Description</b>   |
|---|-------------|------------------|--|
| SIZE1                                   | Integer     | Constant         | The number of retained variables. Can be 1 to 255 . Please also modify the “Dim” value of the “F_VAL1[ ]” and “Old_F_VAL1[ ]” to the same value as “SIZE1”. Here we use “SIZE1” as 17                              |
| num_row1                                | Integer     | Internal         | How many rows in the file ? This value is automatically calculated by “SIZE1” . Each row should have 10 REAL values, except the last row.  |
| Last_num1                               | Integer     | Internal         | How many data in the last row ? This value is automatically calculated by “SIZE1”.   |
|   |             |                  |  |
| RE_LOAD                                 | Bool        | Internal         | Set as True to read File once, init as TRUE  |
| TMP                                     | Bool        | Internal         | Internal use   |
| Data_Ok1                                | Bool        | Internal         | TRUE means File Ok   |
| Flag_to_save                            | Bool        | Internal         | If program want to save data, it will set this value to TRUE.  |
|   |             |                  |  |
| File_name1                              | Message     | Internal         | Len is 64, init as \CompactFlash\data56.txt  |
| Msg1                                    | Message     | Internal         | Len is 128, File processing state  |
| str1                                    | Message     | Internal         | Len is 255, Internal use   |
|   |             |                  |  |
| F_VAL1[0..16]                           | REAL        | Internal         | Variable array, “Dim” should be init as the same value as “SIZE1”  |
| Old_F_VAL1<br>[0..16]                   | REAL        | Internal         | Old value of “F_VAL1[ ]”<br>Variable array, “Dim” should be init as the same value as “SIZE1” .  |
|   |             |                  |  |
| NUM1                                    | Integer     | Internal         | Get return of Msg_F( ), -1 means format error  |
| File1                                   | Integer     | Internal         | File ID  |
| ii                                      | Integer     | Internal         | Index of “for” loops   |
| jj                                      | Integer     | Internal         | Index of “for” loops   |
|   |             |                  |  |
| Data1 ~ Data5<br>And<br>Data06 ~ Data17 | REAL        | Internal         | The User Data variable. Here we have 17 variables in the demo program.<br>User can declare them to different variable name.<br>If name is modified, the “G_data” and the “S_data” program should be modified also. |

ST program - ST1:

---

```
if RE_LOAD then      (* if RE_LOAD is TRUE, get retained data from file *)

    RE_LOAD := FALSE ; (* Set RE_LOAD as FALSE *)

    (* caculate number of rows and data number of the last row *)
    num_row1 := SIZE1 / 10 ;
    last_num1 := SIZE1 - 10 * num_row1 ;
    if last_num1 <> 0 then
        num_row1 := num_row1 + 1 ; (* if last_row has data, num_row1 must plus 1 *)
    else
        last_num1 := 10 ;
    end_if ;

    (* Get retained value from file when controller is powered up *)
    TMP := read_F1( ) ;

    if TMP = FALSE then (* Read file error or file not exist *)

        for ii := 0 to SIZE1 - 1 do
            F_VAL1[ ii ] := 0.0 ; (* set all F_VAL1[ ] 's value as 0.0 *)
        end_for ;

        Data_Ok1 := FALSE ; (* set data is not Ok *)
        Msg1 := 'File : ' + File_name1 + ' not exist or data error ! or File is open now' ;

    else (* Read data Ok *)

        Data_Ok1 := TRUE ; (* set data is Ok *)
        Msg1 := 'Get Retained data from file Ok ' ;

    end_if ;

    (* Update Old_F_VAL1[ ] *)
    for ii := 0 to SIZE1 - 1 do
        Old_F_VAL1[ ii ] := F_VAL1[ ii ] ;
    end_for ;

    (* Get user data from F_VAL1[ ] when controller is just powered up *)
    TMP := G_DATA( ) ;

end_if ;

(* At run time, Set user data to F_VAL1[ ] *)
TMP := S_DATA( ) ;
```

```

(* At run time, test any value of F_VAL1[ ] is modified *)
for ii := 0 to SIZE1 - 1 do

  if Old_F_VAL1[ii] <> F_VAL1[ii] then  (* if any value is modified *)
    Flag_to_save := TRUE ;  (* now save command is given *)
    Old_F_VAL1[ii] := F_VAL1[ii] ;  (* Update Old_F_VAL1[ ] if it is modified *)
  end_if ;

end_for ;

(* if save command is given, it means value is modified *)
if Flag_to_save then

  TMP := save_f1( ) ;  (* save data to file *)

  (* if save file failed, keep this save command *)
  if TMP = FALSE then
    Msg1 := 'Can not save data to file. May be file is open now by Wincon screen ! ' ;

    (* Save Ok, cancel this save command *)
  else
    Flag_to_save := FALSE ;  (* Set as "No save" at the beginning *)

  end_if ;

end_if ;

```

---

ST functions – G\_data :

-----  
(\* If any name of Data1 to Data17 is modified or value of “SIZE1” is modified, User must modify the below code \*)

```
Data1 := F_VAL1[0] ;    (* get variable value from F_VAL1[0..16] *)  
Data2 := F_VAL1[1] ;  
Data3 := F_VAL1[2] ;  
Data4 := F_VAL1[3] ;  
Data5 := F_VAL1[4] ;  
Data06 := F_VAL1[5] ;  
Data07 := F_VAL1[6] ;  
Data08 := F_VAL1[7] ;  
Data09 := F_VAL1[8] ;  
Data10 := F_VAL1[9] ;  
Data11 := F_VAL1[10] ;  
Data12 := F_VAL1[11] ;  
Data13 := F_VAL1[12] ;  
Data14 := F_VAL1[13] ;  
Data15 := F_VAL1[14] ;  
Data16 := F_VAL1[15] ;  
Data17 := F_VAL1[16] ;  
G_data := TRUE ;    (* function returns TRUE *)
```

-----

ST functions – S\_data :

-----  
(\*If any name of Data1 to Data17 is modified or value of “SIZE1” is modified, User must modify the below code \*)

```
F_VAL1[0] := Data1 ;    (* store variable value to F_VAL1[0..16] *)  
F_VAL1[1] := Data2 ;  
F_VAL1[2] := Data3 ;  
F_VAL1[3] := Data4 ;  
F_VAL1[4] := Data5 ;  
F_VAL1[5] := Data06 ;  
F_VAL1[6] := Data07 ;  
F_VAL1[7] := Data08 ;  
F_VAL1[8] := Data09 ;  
F_VAL1[9] := Data10 ;  
F_VAL1[10] := Data11 ;  
F_VAL1[11] := Data12 ;  
F_VAL1[12] := Data13 ;  
F_VAL1[13] := Data14 ;  
F_VAL1[14] := Data15 ;  
F_VAL1[15] := Data16 ;  
F_VAL1[16] := Data17 ;  
S_data := TRUE ;    (* function returns TRUE *)
```

-----

ST functions - read\_f1 :

---

```
(* This function read "SIZE1" number of REAL value from file and store them to F_VAL1[ ]
return read_f1() as TRUE: Ok , FALSE: Error *)
read_f1 := FALSE ; (* set as FALSE: Error at the beginning *)
File1 := f_wopen( File_name1 ) ; (* Try to open file in Read & Write mode *)

if File1 = 0 then (* File doesn't exists *)
  return ; (* exit this function *)
end_if ;

(* max "num_row1" rows to read these "SIZE1" number of REAL values, Each row in the file contains
10 REAL values *)
for ii2 := 0 to num_row1 - 1 do

  if f_eof( File1 ) = TRUE then (* test if End_Of_File reached *)
    exit ; (* Reach End Of File, exit "for" loop *)
  end_if ;

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

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

  (* if data number of last row is not correct *)
  if ( ( ii2 = num_row1 - 1 ) and ( NUM1 <> last_num1 ) ) or

    (* non-last row must have 10 REAL values *)
    ( ( ii2 <> num_row1 - 1 ) and ( NUM1 <> 10 ) ) then

    (* error, it means the format is not correct REAL values or data number is not enough *)
    exit ; (* exit for loop *)

  end_if;

  (* conversion Ok, store these REAL values to F_VAL1[ ] *)
  if ii2 = num_row1 - 1 then (* last row *)
    num2 := last_num1 ; (* last row has only "last_num1" number of data *)
  else
    num2 := 10 ; (* non-last row has 10 data *)
  end_if ;

  (* Get these converted REAL values from No.1 Float array 's addr. 1 to 10 (or 1 to last_num1 for last
row) *)
  for jj2 := 0 to num2 - 1 do
    F_VAL1[ 10*ii2 + jj2 ] := ARY_F_R( 1 , jj2 + 1 ) ;
  end_for ;

end_for ;
```

(\* Any file been open should be closed by f\_close( ) \*)

**TMP2 := f\_close( File1 ) ;**

(\* All rows are read Ok \*)

**if ii2 = num\_row1 then**

**read\_F1 := TRUE ;** (\* return value as TRUE:Ok \*)

**end\_if ;**

-----  
ST functions 程式 save\_f1 :

-----  
(\* This function write 20 REAL value F\_VAL1[0..19] to file

    return save\_f1( ) as TRUE: Ok , FALSE: Error \*)

**save\_f1 := FALSE ;** (\* set as FALSE: Error at the beginning \*)

**File1 := f\_creat( File\_name1 ) ;** (\* Creat a new file to write \*)

**if File1 = 0 then**

**return ;** (\*creat failed , exit this function \*)

**end\_if ;**

(\* max "num\_row1" rows to save these REAL values, Each row in the file contains 10 REAL values \*)

**for ii2 := 0 to num\_row1 - 1 do**

**str1 := ' ' ;** (\* set initial value of str1 \*)

**if ii2 = num\_row1 - 1 then** (\* last row \*)

**num2 := last\_num1 ;** (\* last row has only "last\_num1" number of data \*)

**else** (\* non-last row \*)

**num2 := 10 ;** (\* non-last row has 10 data \*)

**end\_if ;**

**for jj2 := 0 to num2 - 2 do**

**str1 := str1 + REAL\_STR( F\_VAL1[ 10 \* ii2 + jj2 ] ) + ',' ;**

**end\_for ;**

(\* the last data in each row should end with <CR> <LF> character \*)

**str1 := str1 + REAL\_STR( F\_VAL1[ 10 \* ii2 + num2 - 1 ] ) + '\$0D\$0A' ;**

**TMP2 := f\_writ\_s( File1 , str1 ) ;** (\* write one row to file \*)

**end\_for ;**

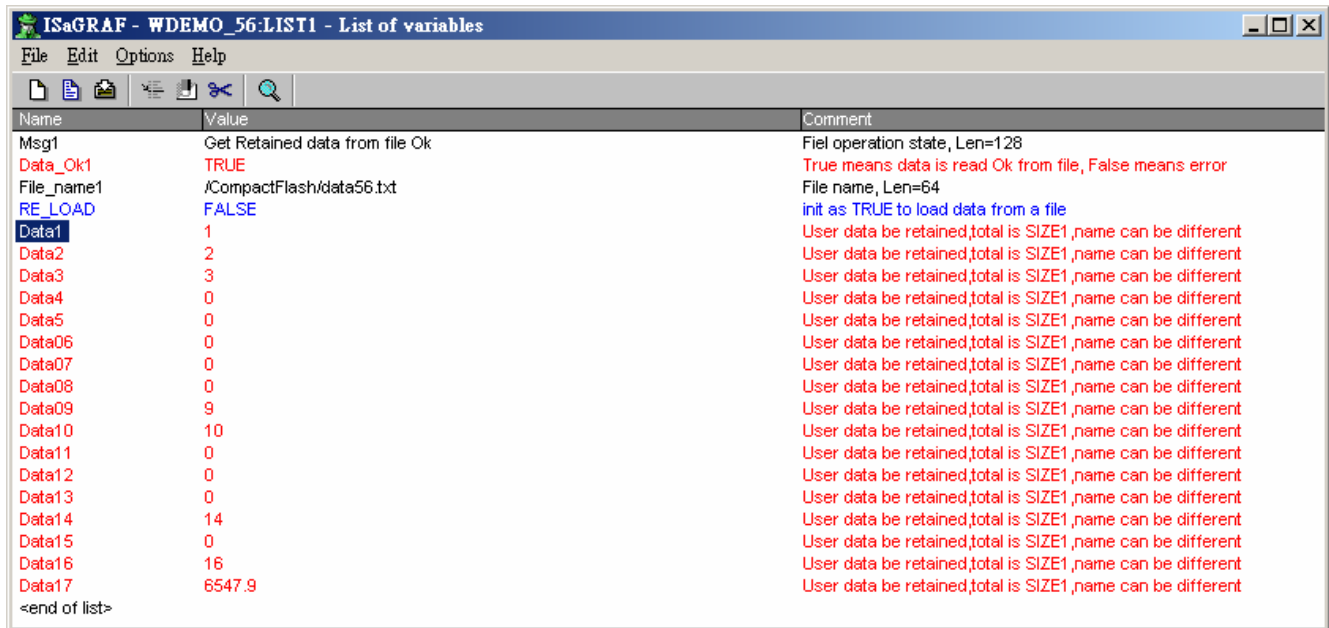
(\* Any file been open should be closed by f\_close( ) \*)

**TMP2 := f\_close( File1 ) ;**

**save\_f1 := TRUE ;** (\* return value as TRUE:Ok \*)

How to test this “wdemo\_56” project ?

1. Please download “wdemo\_56” to W-8xx7, then the “Spy list” window will pop-up as below.



| Name       | Value                          | Comment  |
|------------|--------------------------------|--|
| Msg1       | Get Retained data from file Ok | Fiel operation state, Len=128                              |
| Data_Ok1   | TRUE                           | True means data is read Ok from file, False means error    |
| File_name1 | /CompactFlash/data56.txt       | File name, Len=64  |
| RE_LOAD    | FALSE                          | init as TRUE to load data from a file                      |
| Data1      | 1                              | User data be retained,total is SIZE1,name can be different |
| Data2      | 2                              | User data be retained,total is SIZE1,name can be different |
| Data3      | 3                              | User data be retained,total is SIZE1,name can be different |
| Data4      | 0                              | User data be retained,total is SIZE1,name can be different |
| Data5      | 0                              | User data be retained,total is SIZE1,name can be different |
| Data06     | 0                              | User data be retained,total is SIZE1,name can be different |
| Data07     | 0                              | User data be retained,total is SIZE1,name can be different |
| Data08     | 0                              | User data be retained,total is SIZE1,name can be different |
| Data09     | 9                              | User data be retained,total is SIZE1,name can be different |
| Data10     | 10                             | User data be retained,total is SIZE1,name can be different |
| Data11     | 0                              | User data be retained,total is SIZE1,name can be different |
| Data12     | 0                              | User data be retained,total is SIZE1,name can be different |
| Data13     | 0                              | User data be retained,total is SIZE1,name can be different |
| Data14     | 14                             | User data be retained,total is SIZE1,name can be different |
| Data15     | 0                              | User data be retained,total is SIZE1,name can be different |
| Data16     | 16                             | User data be retained,total is SIZE1,name can be different |
| Data17     | 6547.9                         | User data be retained,total is SIZE1,name can be different |

<end of list>

You may modify any value of USER Data - Data1 to Data17. Then the new value will be saved once into file of “\CompactFlash\data56.txt” .

Then you can open this file on the Wincon’s monitor screen by double click on the file name. You will see the related value is modified. (Please do not always keep this file open. Please close it later, or the new modified data will not be saved . That is because the file is open, write operation is not allowed)

2. Recycle the power of Wincon. You will see the value keep at the last modified value when Wincon is boot up well.

3. Edit a “data56.txt” file on PC as below by “NotePad” utility. (total 17 data)

1.1 , 2.2 , 3.3 , 4.4 , 5.5 , 6.66 , 7.77 , 8.88 , 9.99 , 10.01  
0.01 , 0.02 , 0.03 , 0.04 , 0.05 , 0.06 , 0.07

Then please download this “data56.txt” file to Wincon’s \CompactFlash\ path by “ftp” utility. Then set “RE\_LOAD” to become TRUE on ISaGRAF “Spy list” window. You will see the related variable value is updated.

**10.5.5: Record i-8017H 's Ch.1 to Ch.4 voltage input in a user allocated RAM memory in the Wincon-8xx7 ? The sampling time is one record every 0.01 second. The record period is 1 to 10 minutes. Then PC can download this record and display it as a trend curve diagram by M.S. Excel.**

Please refer to Chapter 11.3.10 or visit [www.icpdas.com](http://www.icpdas.com) – FAQ – Software – ISaGRAF – 057 .



## 10.6: Controller Fault Detection

There is some event may cause “controller fault” happens. For example, value divided by zero or reading a floating point value from EEPROM or S256 or file which has no floating point value saved inside.

ICP DAS ISaGRAF controllers support Controller Fault detection since below driver version.

|  |  |
|--|--|
| I-7188EG                                     | ver. 2.05 (better to be 2.17 or later) |
| uPAC-7186EG                                  | Since it is released around Jan.2008   |
| I-8417/8817/8437/8837<br>i-8437-80 / 8837-80 | ver. 3.07 (better to be 3.19 or later) |
| iPAC-8447/8847                               | Since it is released around Q1/Q2,2008 |
| I-7188XG                                     | ver. 2.04 (better to be 2.15 or later) |
| W-8x37/8x47                                  | ver. 3.18 (better to be 3.36 or later) |

There is two type of controller fault. One is called “Global” fault. The other is “Local” fault. When Global fault happens, the ISaGRAF project will stop running. Waiting the new modified project to be download. When Local fault happens, the ISaGRAF project still runs.

**PC/HMI/OPC Server can request the controller fault state by using Modbus protocol.**

Word address of 9999 is the controller fault state. 0: Ok , 1: Controller fault.

**R\_MB\_ADR( 1, 9999)** to get controller\_state

Word address of 9998 is the controller fault type.

**R\_MB\_ADR(1, 9998)** to get fault\_type.

101 : **Global fault**

**(other value is Local fault)**

- 102 : S\_R\_R Float error
- 103 : R\_MB\_REL Float error
- 104 : INT\_REAL Float error
- 105 : RETAIN\_F Float error
- 106 : RETAIN\_X Float error
- 107 : Real value divided by 0
- 108 : Integer value divided by 0
- 109 : RETAIN\_A Float error
- 110 : Real value multiplication is overflow (exceeds valid range of 32-bit float)
- 111 : Real value division is overflow (exceeds valid range of 32-bit float)
- 112 : Real value addition is overflow (exceeds valid range of 32-bit float)
- 113 : Real value subtraction is overflow (exceeds valid range of 32-bit float)
- 114 : EEP\_F\_R Float error
- 115 : EBUS\_F\_R Float error
- 116 : FBUS\_F\_R Float error
- 117 : F\_REAF\_F Float error (Only in Wincon-8xx7)
- 118 : Can not find I-87K I/O board in slot 0 to 7
- 119 : ARY\_F\_R Float error

- 121 : ANA( ) operation error. For ex, ANA(1.23E20) , ANA(-2.0e25)
- 122 : TMR( ) operation error. For ex, TMR(1.23E20) , TMR(-100)
- 123 : Floatng point calculation error. For ex, pow(1.23E20 , 3.0) , expt(5.0, 10000000)
- 124 : PID\_AL( ) floating point calculation error. (exceeds valid range of 32-bit float)
- 125 : REAL “Variable array” float error. It may be the array index out of the declared range

When Local fault happens, the project is still running, the ISaGRAF project can use

**R\_MB\_ADR( 1, 9999)** to get controller\_state  
**R\_MB\_ADR(1, 9998)** to get fault\_type.

To clear the value in Network address 9999 & 9998, please use **W\_MB\_ADR(1, 9999, 0)** and **W\_MB\_ADR(1, 9998, 0)**. Please refer to below example.

Example:

(\* When controller "Local Fault" happens, the ISaGRAF program can detect it and then program can do the right action \*)

(\* is\_fault & fault\_type are declared as internal integer \*)

(\* tmp is declared as internal boolean \*)

(\* PC / HMI can request controller fault state & type by Modbus protocol at No.=9999 & 9998 \*)

(\* to get controller state \*)

**is\_fault := R\_MB\_ADR(1, 9999) ;** (\* 0: Ok , 1: controller fault happens \*)

(\* To get controller fault type \*)

**fault\_type := R\_MB\_ADR(1, 9998) ;**

**if is\_fault=1 then**

(\* Do action here when "Local Fault" happens \*)

(\* ... \*)

(\* Only for Wincon-8x37: Stop program running & reset all output in slot 1 to 7 \*)

(\* tmp := Stop\_APL( ) ; \*)

(\* To clear the value in Network address 9999 & 9998 when Local fault happens \*)

**tmp := W\_MB\_ADR(1, 9999, 0) ;**

**tmp := W\_MB\_ADR(1, 9998, 0) ;**

**end\_if ;**

---

# Chapter 11. ISaGRAF Programming Examples

---

When you receive the I-8xx7, iPAC-8447/8847, I-7188EG/XG, uPAC-7186EG & W-8xx7 controller system, ICP DAS has created a number of ISaGRAF programming examples for them. These example programs are useful for understanding how to program the controller system with the ISaGRAF Workbench software program.

Users may refer to section 11.3 for the description of some demo examples.

## 11.1: Installing The ISaGRAF Programming Examples

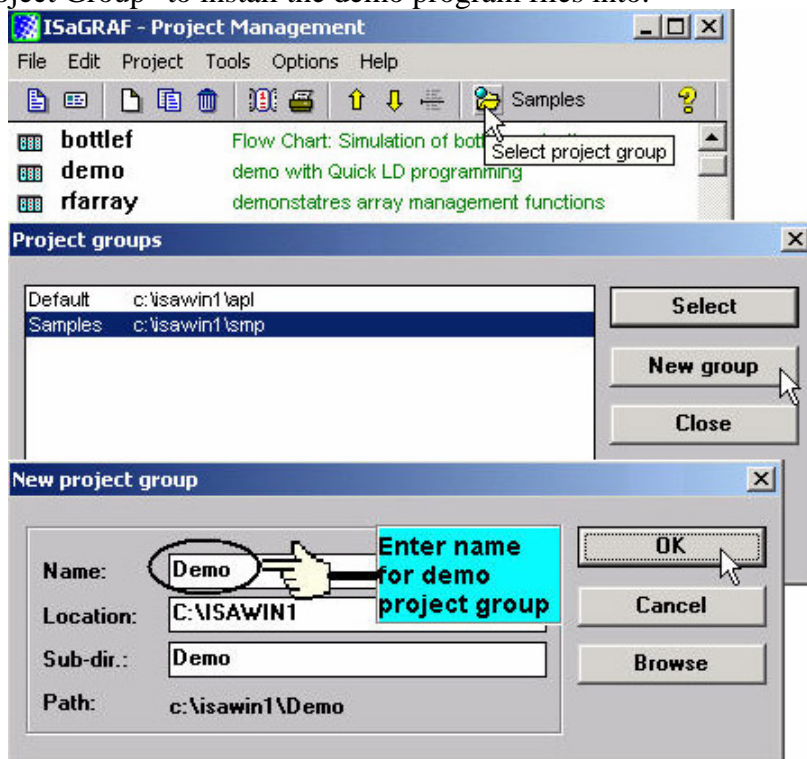
The ISaGRAF programming examples are installed on the same CD-ROM which the "ICP DAS Utilities For ISaGRAF" resides. The CD-ROM is delivered with the product. You will find the programming example files in the below sub-directory in the CD-ROM.

|                        |   |
|------------------------|---|
| I-8xx7/iPAC-8447/8847: | I-8000 CD-ROM: \napdos\isagraf\8000\demo\   |
| I-7188EG/uPAC-7186EG:  | I-8000 CD-ROM: \napdos\isagraf\7188eg\demo\ |
| I-7188XG:              | I-8000 CD-ROM: \napdos\isagraf\7188xg\demo\ |
| W-8xx7:                | Wincon CD-ROM: \napdos\isagraf\wincon\demo\ |

Or you may download them from below web site:

I-8xx7/iPAC-8447/8847 : <ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/8000/demo/>  
I-7188EG/XG/uPAC-7186EG: <ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/7188eg/demo/>  
W-8xx7: [ftp://ftp.icpdas.com/pub/cd/wincon\\_isagraf/napdos/isagraf/wincon/demo/](ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/)

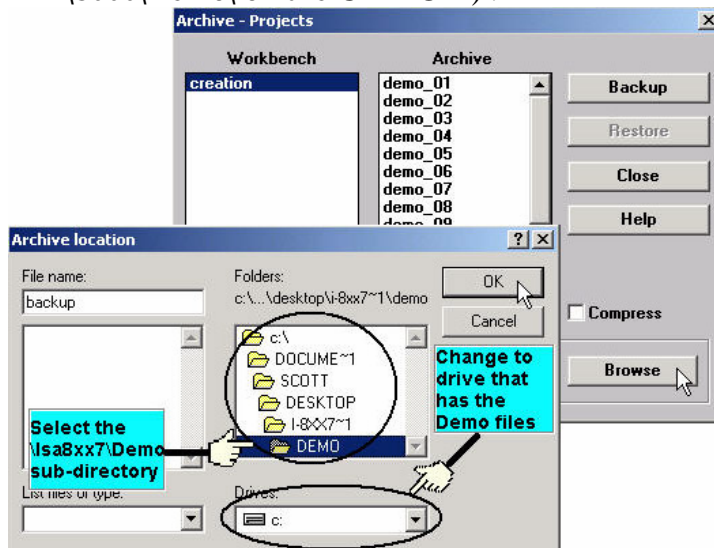
When you install the ISaGRAF example for the controller system it is recommended that you create an "ISaGRAF Project Group" to install the demo program files into.



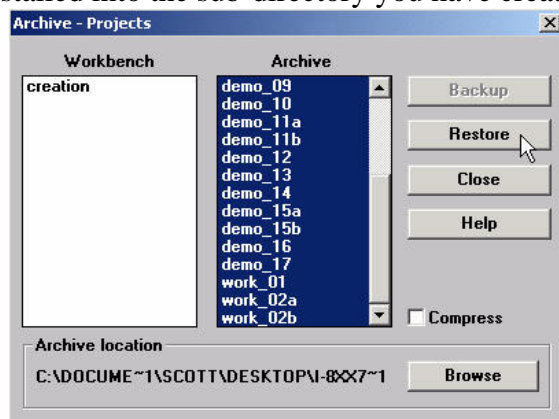
To install the demo programs into the project you have created, open the "ISaGRAF Project Management" window to select "Tools" from the menu bar, then select the "Archive" option and then click on "Projects".



When you click on the "Projects" selection the "Archive Projects" window will open. Click on the "Browse" button to select the drive and the sub-directory where the demo files are located (**For example: Napdos\ISaGRAF\8000\Demo\ on the CD-ROM**).



To install all of the Demo files, click on the "demo\_01" file, then press and hold down the "Shift" key, continue to hold down the "Shift" key and use your mouse to scroll down to last file in the "Archive" window. Click on the last file name from the demo file location and that will select the entire group of demo files. Lastly, click on the "Restore" button in the "Archive Projects" window and all of the demo files will be installed into the sub-directory you have created.



## 11.2: ISaGRAF Demo Example Files

**Visual Basic example program:** (Please refer to their respective “Read.txt” file before using them)

I-8000 CD-ROM:\napdos\isagraf\vb\_demo\  
[ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/vb\\_demo](ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/vb_demo)

|        |   |   |
|--------|---|---|
| Demo_1 | PC Read / Write data in the I-8437/8837 by Modbus TCP/IP  | I-8437/8837<br>I-8054                         |
| Demo_2 | PC use Modem + phone line to link to remote I-8437/8837 (please refer Chapter 13)   | I-84x7/88x7<br>I-87064<br>Modem<br>Phone line |
| Demo_3 | PC run “VB.net 2005” or “VB 6.0” program to Read / Write data in I-8437/8837, I-7188EG or W-8xx7 by Modbus TCP/IP protocol. Please refer to <a href="http://www.icpdas.com/faq/isagraf.htm">http://www.icpdas.com/faq/isagraf.htm</a> 051 & 052 |   |
| Demo_4 | PC run “VB 6.0” program to Read / Write data in “I-8437/8837, I-7188EG or W-8xx7” + I-7018Z by Modbus TCP/IP protocol to display temperature information. Please refer Chapter 11.3.9   | i-7018z                                       |
| Demo_6 | PC run “VB 6.0” program to link to (I-8437-80 / 8837-80) + I-8024 & I-8017H by Modbus TCP/IP. Please refer Chapter 11.3.7   | Slot2: I-8024<br>Slot3: I-8017H               |
| Demo_7 | This demo shows how to use one PC/Server to receive string coming from i-7188EG or i-8x37 or Wincon-8xx7 via UDP protocol. Please refer to <a href="http://www.icpdas.com">www.icpdas.com</a> > FAQ > Software > ISaGRAF > 065                  |   |
| Demo_8 | This demo shows how to use one PC/Server to synchronize the date and time of many i-7188EG or i-8x37 or Wincon-8xx7 via UDP protocol. Please refer to <a href="http://www.icpdas.com">www.icpdas.com</a> > FAQ > Software > ISaGRAF > 070       |   |

W-8xx7 CD-ROM:\napdos\isagraf\wincon\vb6\_demo\_pc\  
[ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/wincon/vb6\\_demo\\_pc](ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/wincon/vb6_demo_pc)  
W-8xx7 CD-ROM:\napdos\isagraf\wincon\vb.net\_2005\_demo\_pc\  
[ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/wincon/vb.net\\_2005\\_demo\\_pc](ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/wincon/vb.net_2005_demo_pc)

|        |   |                                 |
|--------|---|---------------------------------|
| Demo_3 | PC run “VB.net 2005” or “VB 6.0” program to Read / Write data in I-8437/8837, I-7188EG or W-8xx7 by Modbus TCP/IP protocol. Please refer to <a href="http://www.icpdas.com/faq/isagraf.htm">http://www.icpdas.com/faq/isagraf.htm</a> FAQ-051 and 052   |                                 |
| Demo_5 | PC run “VB 6.0” program to link to W-8xx7 + I-8024 & I-8017H by Modbus TCP/IP. Please refer Chapter 11.3.10   | Slot2: I-8024<br>Slot3: I-8017H |
| Demo_9 | This demo shows how to use one PC/Server to synchronize the date and time of many Wincon-8xx7 via UDP protocol and monitoring their communication state. The W-8xx7 will send an email with one attached data file at specified time. Please refer to <a href="http://www.icpdas.com">www.icpdas.com</a> > FAQ > Software > ISaGRAF > 071 |                                 |

**I-7188EG/XG example program:****I-8000 CD-ROM:** \napdos\isagraf\7188eg\demo 或<ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/7188eg/demo/>

| <b>Project name</b>      | <b>Description</b>   | <b>I/O board used</b>                                |
|--------------------------|--|--|
| <a href="#">Demo_01</a>  | Receive data and send to Com2 and Com3   | X503 / 4 / 5 / 6                                     |
| <a href="#">Demo_02</a>  | Send one string to COM5 and COM6 in X503 board   | X503   |
| <a href="#">Demo_03</a>  | Receive message and then send to Com6 or Com7 (using “Comary_r” and “Comary_w”)  | X503   |
| <a href="#">Demo_04</a>  | Linking remote I-7000 and using X107 board   | Bus7000b<br>X107                                     |
| <a href="#">Demo_05</a>  | Timer control, TP, TON, TOF  | X304   |
| <a href="#">Demo_06</a>  | Display a value to S-MMI by “VAL10LED”   | X304   |
| <a href="#">Demo_07</a>  | Using X107 and remote I-7060D Relay I/O  | Bus7000b<br>X107                                     |
| <a href="#">Demo_08</a>  | Receive message and then send to COM3 in X507/8/9 and control Diital Output .  | X507_8_9   |
| <a href="#">Demo_09</a>  | Using S-MMI and Timer control command “tStart” , “tStop” and Reset to 0  |  |
| <a href="#">Demo_10</a>  | Using S-MMI  | X107   |
| <a href="#">Demo_11</a>  | Linking other Modbus RTU device  | mbus   |
| <a href="#">Demo_12</a>  | Training box demo 1  | Bus7000b   |
| <a href="#">Demo_13</a>  | Trainning box demo 2   | Bus7000b   |
| <a href="#">Demo_18</a>  | PID control by “PID_AL”. This program can not simulate in PC, please download to controller.   |  |
| <a href="#">Demo_21</a>  | Write one string to Com3 and Com4  | Xbi8 (set as virtual)<br>X50x                        |
| <a href="#">Demo_22</a>  | Receive message and send to Com3 and Com4  | X50x   |
| <a href="#">Demo_23</a>  | Receive command from PC and return a Integer value.<br>Comary_R , Comary_W   | X50x   |
| <a href="#">Demo_35a</a> | Time synchronization by using Fbus between two or more controllers. “Demo_35A” should be used with “Demo_35B” demo<br>If the time is modified in 35A, the time in controller running 35B will be automatically modified<br>(User can modify the program to use Ebus) | Fbus_m   |
| <a href="#">Demo_35b</a> | Time synchronization by using Fbus between two or more controllers.  | Fbus_s   |
| <a href="#">Demo_36</a>  | Get driver version of the I-8xx7, 7188EG/XG  |  |
| <a href="#">Demo_41</a>  | Record alarm (text) in X607/X608, then PC can download this record by "ICPDAS UDloader"  | X607_608<br>Xbi8 (virtual D/I)<br>Xbo8 (virtual D/O) |
| <a href="#">Demo_43</a>  | SMS demo, Please modify to your own phone number in the  | SMS  |

| Project name             | Description  | I/O board used                 |
|--------------------------|--|--------------------------------|
|                          | ISaGRAF dictionary window  |                                |
| <a href="#">Demo_43a</a> | Same as Demo_43 but sending SMS to two cell. phone   | SMS                            |
| <a href="#">Demo_44</a>  | PC download data to X607/X608  | X607_608<br>Xbo8 (virtual D/O) |
| <a href="#">Demo_50</a>  | PWM I/O demo. (Pulse Width Modulation)   | X107                           |
| <a href="#">Demo_51a</a> | Redundancy: I-7188EG redundant Master (refer to Section 6.4)   | Bus7000b<br>Ebus_m             |
| <a href="#">Demo_51b</a> | Redundancy: I-7188EG redundant slave (refer to Section 6.4)  | Bus7000b<br>Ebus_s             |
| <a href="#">Demo_61</a>  | D/I counters using DI_CNT, I-7188EG/XG + X107,<br>Do something when D/I signal happens   | X107                           |
| <a href="#">Demo_70</a>  | Send message to COM2 or COM3 when Alarm 1 to 8 happens   |                                |
| <a href="#">Demo_72</a>  | Controller link one RS-485 remote i-7018z, and also PC can run “VB 6.0” program to become an HMI screen. (please refer to Chapter 11.3.9)      | i-7018z                        |
| <a href="#">Demo_74</a>  | 7188EG: Auto-report data to PC via UDP.<br>Please refer to <a href="http://www.icpdas.com">www.icpdas.com</a> > FAQ > Software > ISaGRAF > 065 | i-7018z                        |

**NOTE:**

Demo\_18 uses PID\_AL which is provided by CJ International for evaluation. Please refer to “CD\Napdos\isagraf\8000\english\_manu\ PID\_AL.ComplexPIDalgorithm implementation.htm”.



## I-8417/8817/8437/8837 example program:

**I-8000 CD-ROM:** \napdos\isagraf\8000\demo or  
<ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/8000/demo/>

| Project name             | Description   | I/O board used             |
|--------------------------|---|----------------------------|
| <a href="#">Demo_01</a>  | Timer control, TP, TON, TOF   | Push4Key, Show3Led         |
| <a href="#">Demo_01a</a> | To do something at some second later when an event happens  | Push4Key, Show3Led         |
| <a href="#">Demo_02</a>  | Start, Stop and Reset a Timer, tStart, tStop  | Push4Key, Show3Led         |
| <a href="#">Demo_03</a>  | Read / Write Date & Time<br>SYSDAT_R, SYSDAT_W, SYSTIM_R, SYSTIME_W<br>To output at a time interval. Like, Monday, 09:00 ~ 18:00, Sunday, 10:00 ~ ... | NONE                       |
| <a href="#">Demo_04</a>  | Calculate empty cycle time  | NONE                       |
| <a href="#">Demo_05</a>  | Blinking output, TP, Blink  | Push4Key, Show3Led         |
| <a href="#">Demo_06</a>  | Change output mode  | Push4Key, Show3Led         |
| <a href="#">Demo_07</a>  | Display a value to S-MMI, VAL10LED, tStart, tStop   | Push4Key, Show3Led         |
| <a href="#">Demo_08</a>  | Input a value from S-MMI, INP10LED  | Push4Key, Show3Led         |
| <a href="#">Demo_09</a>  | +, -, *, /  | NONE                       |
| <a href="#">Demo_10</a>  | Display analog input value to S-MMI   | I-87017, I-87024, Push4Key |
| <a href="#">Demo_11a</a> | Fbus Master, NET_ID = 1   | Fbus_m, Push4Key, Show3Led |
| <a href="#">Demo_11b</a> | Fbus Slave, NET_ID = 2  | Fbus_s, Push4Key           |
| <a href="#">Demo_12</a>  | Using COM3 to receive data from PC  | Show3Led                   |
| <a href="#">Demo_13</a>  | Send data to Com3 every 3 seconds   | I-87017                    |
| <a href="#">Demo_14</a>  | Convert I-7K & I-87K protocol to Modbus protocol  | Bus7000b                   |
| <a href="#">Demo_15a</a> | Link other Modbus device  | Mbus                       |
| <a href="#">Demo_15b</a> | Simulate I-8417 as a modbus device for Demo_15a to link to this project   | None                       |
| <a href="#">Demo_16</a>  | Send Modbus command to device once every second   | Push4Key, Mbus             |
| <a href="#">Demo_17</a>  | Read / Write EEPROM   | None                       |
| <a href="#">Demo_18</a>  | PID control by "PID_AL". This program can not simulate in PC, please download to controller.  | None                       |
| <a href="#">Demo_21</a>  | Send string to Com5 and Com6  | Push4Key,                  |



|                          |   |                              |
|--------------------------|---|------------------------------|
|                          |   | Show3Led                     |
| <a href="#">Demo_22</a>  | Receive data from Com5 or Com6 (RS-232) and echo back   | Show3Led                     |
| <a href="#">Demo_23</a>  | Receive user defined protocol from PC   | Show3Led                     |
| <a href="#">Demo_27</a>  | Motion control: x axis, slot 0: i-8091,<br>Slot1:i-8090, Napdos\ISaGRAF\8000\Driver\motion.pdf  | I-8091<br>I-8090<br>Show3Led |
| <a href="#">Demo_27a</a> | To move some pulse at x-axis of i-8091 of slot 1  | I-8091                       |
| <a href="#">Demo_28</a>  | Motion control: x , y axes, slot0: i-8091, slot1:i-8090,<br>Napdos\ISaGRAF\8000\Driver\motion.pdf   | I-8091<br>I-8090<br>Show3Led |
| <a href="#">Demo_29</a>  | Store 1200 short integer values every 75 seconds and then send to PC<br>via COM3  | I-87017                      |
| <a href="#">Demo_30</a>  | Store 2880 short integer values every 18 seconds and then send to PC<br>via COM3  | I-8017h                      |
| <a href="#">Demo_33</a>  | Read / Write user defined protocol via COM3   | Show3Led                     |
| <a href="#">Demo_35a</a> | Time synchronization by using Fbus between two or more<br>controllers. “Demo_35A” should be used with “Demo_35B” demo<br>If the time is modified in 35A, the time in controller running 35B will<br>be automatically modified<br><b>(User can modify the program to use Ebus)</b> | Fbus_m                       |
| <a href="#">Demo_35b</a> | Time synchronization by using Fbus between two or more<br>controllers.  | Fbus_s                       |
| <a href="#">Demo_37</a>  | Spotlight demo (Simple HMI) . please refer to Chapter 14  | Push4Key<br>Show3Led         |
| <a href="#">Demo_38</a>  | I-8xx7 link MMICON , demo 1, please refer to Chapter 16   |                              |
| <a href="#">Demo_39</a>  | I-8xx7 link MMICON , demo 2, please refer to Chapter 16   |                              |
| <a href="#">Demo_40</a>  | Store 8 A/I (binary) to S256 per minute, then PC can load it by<br>"ICPDAS UDownloader"   |                              |
| <a href="#">Demo_41</a>  | Record Alarm (text) to S256/512 & PC can load it by "ICPDAS<br>UDloader"  |                              |
| <a href="#">Demo_42</a>  | Store 8 A/I (text) to S256 per min, then PC can load it by "ICPDAS<br>UDloader"   |                              |
| <a href="#">Demo_43</a>  | SMS demo, Please declare your own phone No. in the dictionary,<br>message type  | SMS                          |
| <a href="#">Demo_43a</a> | Same as demo_43, but send to many cell. phones.   | SMS                          |
| <a href="#">Demo_44</a>  | Demo of PC to download data to the S256/512   |                              |
| <a href="#">Demo_46</a>  | Motion control:<br>Pulse move at a specified speed  | I-8091<br>I-8090             |

**NOTE:**

Demo\_18 uses PID\_AL which is provided by CJ International for evaluation. Please refer to  
“CD\Napdos\isagraf\8000\english\_manu\ PID\_AL.ComplexPIDalgorithm implementation.htm”.

| <b>Project name</b>      | <b>Description</b>  | <b>I/O board used</b>        |
|--------------------------|---|------------------------------|
| <a href="#">Demo_49a</a> | I-8437/8837 redundant Master (refer to Section 6.4)<br>(Wincon-8347/8747 is much better, Wdemo_18, Chapter 20)  | Bus7000b<br>Ebus_m           |
| <a href="#">Demo_49b</a> | I-8437/8837 redundant slave (refer to Section 6.4)<br>(Wincon-8347/8747 is much better, Wdemo_18, Chapter 20)   | Bus7000b<br>Ebus_s           |
| <a href="#">Demo_50</a>  | PWM I/O , please refer to Chapter 3.7   | I-8055                       |
| <a href="#">Demo_52</a>  | Get counter value in I-8K D/I board. Refer to Chapter 3.8<br>(Counter value is retained)  | I-8051<br>Push4Key           |
| <a href="#">Demo_53</a>  | Get counter value in I-8K D/I board. Refer to Chapter 3.8<br>(Counter value is not retained)  | I-8051<br>I-8056<br>Push4key |
| <a href="#">Demo_54a</a> | Modbus Master demo (Master)   |                              |
| <a href="#">Demo_54b</a> | For Demo_54a to link to this project (Slave)  |                              |
| <a href="#">Demo_55</a>  | PWM I/O demo 2. please refer to Chapter 3.7   | I-8055                       |
| <a href="#">Demo_61</a>  | I-8K DI Counter demo, 8xx7 + 8051, Refer to Chapter 3.8<br>Do something when D/I event happens  | I-8051                       |
| <a href="#">Demo_70</a>  | Send string to COM2 when alarm 1 to 8 happens, slot 1: i8077  |                              |
| <a href="#">Demo_71</a>  | record i-8017H 's Ch.1 to Ch.4 voltage input in S-256 / 512 in<br>i-8437-80 or i-8837-80 ? The sampling time is one record every 0.05<br>second. The record period is 1 to 10 minutes. Then PC can download<br>this record and display it as a trend curve diagram by M.S. Excel.<br>Please refer to Chapter 11.3.7 | i-8024<br>i-8017h            |
| <a href="#">Demo_72</a>  | Controller link one RS-485 remote i-7018z, and also PC can run "VB<br>6.0" program to become an HMI screen. (please refer to Chapter<br>11.3.9)   | i-7018z                      |
| <a href="#">Demo_73</a>  | COM3 demo to receive an integer value from remote RS-232 device.<br>Please refer to Chapter 24 and <a href="http://www.icpdas.com">www.icpdas.com</a> > FAQ > Software ><br>ISaGRAF > 066   |                              |

**W-8047/8347/8747 example program:**

**Wincon-8xx7 CD-ROM:** \napdos\isagraf\wincon\demo\ or  
[ftp://ftp.icpdas.com/pub/cd/wincon\\_isagraf/napdos/isagraf/wincon/demo/](ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/)

| Project name              | Description   | I/O used              |
|---------------------------|---|-----------------------|
| <a href="#">Wdemo_01</a>  | Read / Write REAL value (32-bit float format) from file<br>Please refer to Wdemo_50, 51, 54, 55, 56 about file operation.   |                       |
| <a href="#">Wdemo_02</a>  | Read / Write Integer value (32-bit signed integer) from file<br>Please refer to Wdemo_50, 51, 54, 55, 56 about file operation   |                       |
| <a href="#">Wdemo_03</a>  | Read / Write Date & Time<br>SYSDAT_R, SYSDAT_W, SYSTIM_R, SYSTIME_W<br>To output at a time interval. Like, Moday, 09:00 ~ 18:00, Sunday, 10:00 ~ ...  |                       |
| <a href="#">Wdemo_04</a>  | User defined Modbus protocol (No using "Mbus")  |                       |
| <a href="#">Wdemo_05</a>  | To do something at some sec later when an event happens   | 8055                  |
| <a href="#">Wdemo_06</a>  | Using Message Array, MsgAry_r , MsgAry_w  |                       |
| <a href="#">Wdemo_07</a>  | Convert float value to string, using real_str & rea_str2  |                       |
| <a href="#">Wdemo_08</a>  | PID control by "PID_AL". This program can not simulate in PC, please download to controller.  |                       |
| <a href="#">Wdemo_09</a>  | Read / Write file<br>Please refer to Wdemo_50, 51, 54, 55, 56 about file operation.   |                       |
| <a href="#">Wdemo_10</a>  | Read / Write EEPROM   |                       |
| <a href="#">Wdemo_12</a>  | File path is in \CompactFlash\ . Record 3 REAL values in 3 different files every 10 minutes. Save to another 3 file names every month.  |                       |
| <a href="#">Wdemo_13</a>  | record i-8081 ODM-2 's frequency into RAM Disk files every 5 seconds. When finished, copy this RAM disk file to \CompactFlash\  | i-8081 ODM2<br>i-8055 |
| <a href="#">Wdemo_14</a>  | Using new retained variables, Retain_b, Retain_N, Retain_f, Retain_T (refer to Chapter 10.1)  |                       |
| <a href="#">Wdemo_17</a>  | File path is \CompactFlash\ . Record 3 REAL values in a file every minute. Save to another file name every day.   |                       |
| <a href="#">Wdemo_18</a>  | Redundant, Wincon+I-87K4+I-87055,<br>Master IP 10.0.0.103, Slave IP 10.0.0.105<br>Please refer to <a href="http://www.icpdas.com/faq/isagraf.htm">http://www.icpdas.com/faq/isagraf.htm</a> FAQ-036 , 041 & Chapter 20  |                       |
| <a href="#">Wdemo_19</a>  | Send message to remote PC via UDP, Time_gap is 1 second<br>Refer to <a href="http://www.icpdas.com/faq/isagraf.htm">http://www.icpdas.com/faq/isagraf.htm</a> FAQ-029   | i-8077                |
| <a href="#">Wdemo_19a</a> | When Alarm happens and last for longer than 3 seconds, send message to remote PC via UDP. Time_gap is 0.25 秒 ("wdemo19a" is better than "wdemo19b")<br>Refer to <a href="http://www.icpdas.com/faq/isagraf.htm">http://www.icpdas.com/faq/isagraf.htm</a> FAQ-029 | i-8077                |
| <a href="#">Wdemo_20</a>  | Receive message from remote PC or remote device via UDP<br>Refer to <a href="http://www.icpdas.com/faq/isagraf.htm">http://www.icpdas.com/faq/isagraf.htm</a> FAQ-029   |                       |
| <a href="#">Wdemo_21</a>  | using "com_MRTU" to disable/enable Modbus RTU slave port,   |                       |
| <a href="#">Wdemo_22</a>  | PWM I/O   | i-8055                |

|                          |   |                  |
|--------------------------|---|------------------|
| <a href="#">Wdemo_23</a> | Send Time String to COM2:RS232 every second by using COMOPEN, COMSTR_W  |                  |
| <a href="#">Wdemo_24</a> | Send message to COM2 if Alarm 1 to 8 happens and last for longer than 3 seconds   | 8077             |
| <a href="#">Wdemo_25</a> | Redundant, Wincon+I-87K4+I-87055 Plus Modbus devices<br>Please refer to <a href="http://www.icpdas.com/faq/isagraf.htm">http://www.icpdas.com/faq/isagraf.htm</a> FAQ-036, 041 & Chapter 20                         |                  |
| <a href="#">Wdemo_26</a> | Motion control:<br>To move some pulse at x-axis of i-8091 of slot 1 in W-8337/8737  | i-8091           |
| <a href="#">Wdemo_27</a> | Motion control: x axis, slot 1: i-8091, slot 2: i-8090,<br>Napdos\ISaGRAF\8000\Driver\motion.pdf  | i-8091<br>i-8090 |
| <a href="#">Wdemo_28</a> | Motion control: x, y axes, slot1: i-8091, slot2: i-8090,<br>Napdos\ISaGRAF\8000\Driver\motion.pdf   | i-8091<br>i-8090 |
| <a href="#">Wdemo_29</a> | Motion control: move to absolute position,<br>slot 1 : i-8091, slot 2: i-8090   | i-8091<br>i-8090 |
| <a href="#">Wdemo_30</a> | Wincon link two I-8KE4/8-MTCP Ethernet I/O<br>Refer to <a href="http://www.icpdas.com/faq/isagraf.htm">http://www.icpdas.com/faq/isagraf.htm</a> FAQ-042  |                  |
| <a href="#">Wdemo_31</a> | Wincon link one I-8KE4/8-MTCP Ethernet I/O.<br>Refer to <a href="http://www.icpdas.com/faq/isagraf.htm">http://www.icpdas.com/faq/isagraf.htm</a> FAQ-042   |                  |
| <a href="#">Wdemo_32</a> | Setup Wincon as a TCP/IP Client, then link to another TCP/IP Server to send message.<br>Please refer to <a href="http://www.icpdas.com/faq/isagraf.htm">http://www.icpdas.com/faq/isagraf.htm</a> FAQ043            | i-8077           |
| <a href="#">Wdemo_33</a> | Same as “wdemo_32”, but send message only when event happens and last for longer than 3 seconds Please refer to<br><a href="http://www.icpdas.com/faq/isagraf.htm">http://www.icpdas.com/faq/isagraf.htm</a> FAQ043 | i-8077           |
| <a href="#">Wdemo_35</a> | Wincon link EKAN Modview LED . Refer to Chapter 8.6 or<br><a href="http://www.icpdas.com/faq/isagraf.htm">http://www.icpdas.com/faq/isagraf.htm</a> FAQ-045   |                  |
| <a href="#">Wdemo_36</a> | Read REAL data from Modbus RTU device (refer to Chapter 8)  |                  |
| <a href="#">Wdemo_37</a> | Write REAL data to Modbus RTU device (refer to Chapter 8)   |                  |
| <a href="#">Wdemo_38</a> | Write 16 bits to device by using Modbus Master function call 6.<br>(refer to Chapter 8)   |                  |
| <a href="#">Wdemo_39</a> | Using FRNET Fast D/I/O  | i-8172           |
| <a href="#">Wdemo_41</a> | Link M-7000 I/O. M7053D + M7045D (D/I & D/O)<br>Refer to <a href="http://www.icpdas.com/faq/isagraf.htm">http://www.icpdas.com/faq/isagraf.htm</a> FAQ-050  |                  |
| <a href="#">Wdemo_42</a> | Link M-7000 I/O. M7053D to get D/I Counter value<br>Refer to <a href="http://www.icpdas.com/faq/isagraf.htm">http://www.icpdas.com/faq/isagraf.htm</a> FAQ-050  |                  |
| <a href="#">Wdemo_43</a> | Link M-7000 I/O. M7017R + M7024 (A/I & A/O)<br>Refer to <a href="http://www.icpdas.com/faq/isagraf.htm">http://www.icpdas.com/faq/isagraf.htm</a> FAQ-050   |                  |
| <a href="#">Wdemo_44</a> | Link M-7000 I/O. M7017RC (curent input +/- 20 mA , 4 to 20mA)<br>Refer to <a href="http://www.icpdas.com/faq/isagraf.htm">http://www.icpdas.com/faq/isagraf.htm</a> FAQ-050   |                  |
| <a href="#">Wdemo_45</a> | Link M-7000 I/O. M7019R (Thermo-Couple)<br>Refer to <a href="http://www.icpdas.com/faq/isagraf.htm">http://www.icpdas.com/faq/isagraf.htm</a> FAQ-050   |                  |
| <a href="#">Wdemo_46</a> | Link M-7000 I/O. M7080 (High speed Counter)<br>Refer to <a href="http://www.icpdas.com/faq/isagraf.htm">http://www.icpdas.com/faq/isagraf.htm</a> FAQ-050   |                  |
| <a href="#">Wdemo_48</a> | PC run VB.net 2005 or VB 6.0 program to link to controllers by Modbus TCP/IP protocol<br>Refer to <a href="http://www.icpdas.com/faq/isagraf.htm">http://www.icpdas.com/faq/isagraf.htm</a> FAQ-051 & 052           |                  |
| <a href="#">Wdemo_50</a> | Non-linear conversion. like give P to find V (P , V relation listed in a  |                  |

|                          |   |                                |
|--------------------------|---|--------------------------------|
|                          | file). Please refer to Chapter 10.5   |                                |
| <a href="#">Wdemo_51</a> | Read 10 REAL value from a file,10 rows,each row has 1 REAL value, use str_real  |                                |
| <a href="#">Wdemo_52</a> | Using “Msg_F” to extract some REAL values from a string<br>Controllers support “Msg_F” & “Msg_N” since below driver<br>i-8xx7: 3.19 , i-7188EG: 2.17 , i-7188XG: 2.15 W-8xx7 : 3.36   |                                |
| <a href="#">Wdemo_53</a> | Using “Msg_N” to extract some Integer values from a string<br>Controllers support “Msg_F” & “Msg_N” since below driver<br>i-8xx7: 3.19 , i-7188EG: 2.17 , i-7188XG: 2.15 W-8xx7 : 3.36  |                                |
| <a href="#">Wdemo_54</a> | Read 20 REAL values from a file,4 rows,each row has 5 REAL values,uses “MSG_F”<br>Controllers support “Msg_F” & “Msg_N” since below driver<br>i-8xx7: 3.19 , i-7188EG: 2.17 , i-7188XG: 2.15 W-8xx7 : 3.36  |                                |
| <a href="#">Wdemo_55</a> | Read 20 Integers from a file,2 rows, each row has 10 Integers,uses “MSG_N”<br>Controllers support “Msg_F” & “Msg_N” since below driver<br>i-8xx7: 3.19 , i-7188EG: 2.17 , i-7188XG: 2.15 W-8xx7 : 3.36  |                                |
| <a href="#">Wdemo_56</a> | Retain 17 REAL values in a file, 2 rows, Each row has 10 REAL value. Wincon will read these retained values when it is Power Up. Then if any value is modified, all value will be retained once in a file. (using new retained functions in the S-256/512 is much better, please refer to section 10.1) |                                |
| <a href="#">Wdmo_56e</a> | Retain more than 255 REAL, 255 Boolean, 255 Integers in two files in the CompactFlash cards. (using new retained functions in the S-256/512 is much better, please refer to section 10.1)   |                                |
| <a href="#">Wdemo_57</a> | Redundancy system of two Wincon-8347 plus Ethernet I/O  |                                |
| <a href="#">Wdemo_60</a> | Wincon-8347/8747 TCP/IP client demo   | i-87018z                       |
| <a href="#">Wdemo_61</a> | i8xx7, W8xx7: Auto-report data to PC via UDP. Please refer to <a href="http://www.icpdas.com">www.icpdas.com</a> > FAQ > Software > ISaGRAF > 065   | i-7018z                        |
| <a href="#">Wdemo_63</a> | For W-8xx7 only. Send email to one receiver with one attached file. (Please refer to Chapter 12 & <a href="http://www.icpdas.com">www.icpdas.com</a> > FAQ > Software > ISaGRAF > 067)  |                                |
| <a href="#">Wdmo_64a</a> | station 1001 , Time synchronization of many controllers via Ethernet. Please refer to <a href="http://www.icpdas.com">www.icpdas.com</a> > FAQ > Software > ISaGRAF > 070   |                                |
| <a href="#">Wdmo_64b</a> | station 1002 , Time synchronization of many controllers via Ethernet. Please refer to <a href="http://www.icpdas.com">www.icpdas.com</a> > FAQ > Software > ISaGRAF > 070   |                                |
| <a href="#">Wdmo_65a</a> | W-8xx7: Record temperature per minute to a file. Then send it by email per day. Please refer to <a href="http://www.icpdas.com">www.icpdas.com</a> > FAQ > Software > ISaGRAF > 071   | i-87018z                       |
| <a href="#">Wdmo_65b</a> | W-8xx7: Same as wdmo_65a but add time synchronization and state report to PC  | i-87018z                       |
| <a href="#">Wdemo_66</a> | Record 1 to 4-Ch. i8017H voltage pe 20ms, then send this record file by Email. Please refer to <a href="http://www.icpdas.com">www.icpdas.com</a> > FAQ > Software > ISaGRAF > 072  | i-8024<br>i-8017H              |
| <a href="#">Wdemo_70</a> | FRNET in W8xx7 or iPAC-8447. slot1:i-8172, port 0, FR-2057(Adr=4) , FR-2053(Adr=8) . Please refer to Chapter 23   | i-8172<br>FR-2053T<br>FR-2057T |

## **VB.NET 2003 example program: Running with ISaGRAF program in the same Wincon**

Wincon-8xx7 CD-ROM: \napdos\ISaGRAF\Wincon\VB.NET\_Demo\” or  
(Please copy related “Demo\_xx” folder to the hard drive in your PC and remember to  
clear the “read-only” attribution)

[ftp://ftp.icpdas.com/pub/cd/wincon\\_isagraf/napdos/isagraf/wincon/vb.net\\_demo/](ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/vb.net_demo/)

| Project name | Description  | I/O board used     |
|--------------|--|--------------------|
| vbdmo_01     | VB.net demo 01: Digital output module (slot 1: 8077)   | i-8077             |
| vbdmo_02     | VB.net demo 02: Change output mode (slot 1: 8077)      | i-8077             |
| vbdmo_03     | VB.net demo 03: DIO demo (slot 1: 8077)                | i-8077             |
| vbdmo_05     | VB.net demo 05: Analog output (slot 2: 87024)          | i-87024            |
| vbdmo_06     | VB.net demo 06: Analog I/O (slot 2: 87024 , 3: 8017H)  | i-87024<br>i-8017H |
| vbdmo_07     | VB.net demo 07: R/W long , float & Timer (No IO board) |                    |

## Wincon Web HMI example program :

Web page source file :

Wincon-8xx7 CD-ROM: \napdos\isagraf\wincon\WebHMI\_Demo\ or  
[ftp://ftp.icpdas.com/pub/cd/wincon\\_isagraf/napdos/isagraf/wincon/webhmi\\_demo/](ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/webhmi_demo/)

Related ISaGRAF program:

Wincon-8xx7 CD-ROM: \napdos\isagraf\wincon\demo\ or  
[ftp://ftp.icpdas.com/pub/cd/wincon\\_isagraf/napdos/isagraf/wincon/demo/](ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/)

| Project name | Description   | I/O board used   |
|--------------|---|--|
| sample       | An empty Web HMI demo example   |  |
| example1     | Example used in Chapter 4 of W-8xx7 “Getting Started Manual”  | slot 1: I-8077   |
| whmi_01      | Display controller's date & time (No I/O board)   |  |
| whmi_02      | DI & DO demo (slot 1: I-8077)   | slot 1: I-8077   |
| whmi_03      | R/W Long, float & Timer value (No I/O board)  |  |
| whmi_04      | R/W controller's String (No I/O board)  |  |
| whmi_05      | Multi-Pages demo (slot 1: I-8077), Menu is on the Left  | slot 1: I-8077   |
| whmi_05a     | Multi-Pages demo (slot 1: I-8077), Menu is on the Top   | slot 1: I-8077   |
| whmi_06      | AIO demo (slot 2: I-87024, 3:8017H), scaling is in ISaGRAF  | slot 2: I-87024<br>slot 3: I-8017H   |
| whmi_07      | AIO demo (slot 2: I-87024, 3:8017H), , scaling is in PC   | slot 2: I-87024<br>slot 3: I-8017H   |
| whmi_08      | Download controller's file to PC (slot 1: I-8077)   | slot 1: I-8077   |
| whmi_09      | pop up an alarm window on PC (slot 1: I-8077)   | slot 1: I-8077   |
| whmi_11      | trend curve demo  | I-87024 , I-8017H  |
| whmi_12      | Wincon record voltage input of i-8017H in file of “\Web_Data1\curve1.js” in the RAM Disk (Max. recording speed is <b>20Hz</b> ), Then PC can download this record file and display as a Trend Curve diagram by M.S. Excel | Slot2:I-8024<br>Slot3:I-8017H<br>i-8024's voltage output Ch.1 to 4 wired to i-8017H's Ch.1 to 4. |
| whmi_13      | Wincon record voltage input of i-8017H in a user-allocated RAM memory (Max. recording speed is <b>100Hz</b> ), Then PC can download this record file and display as a Trend Curve diagram by M.S. Excel                   | Slot2:I-8024<br>Slot3:I-8017H<br>i-8024's voltage output Ch.1 to 4 wired to i-8017H's Ch.1 to 4. |

## 11.3: Description Of Some Demo Examples

### 11.3.0 Demo\_01A & Demo\_03: Do something at specific time

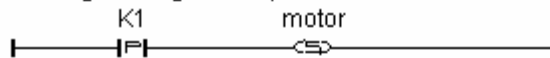
**Demo\_01A: Do something at some seconds later when an event happens.**

Location: I-8000 CD-ROM: \napdos\isagraf\8000\demo\ “demo\_01a.pia”

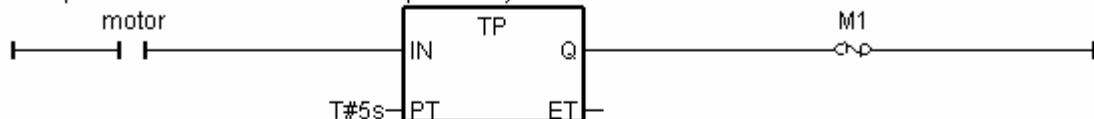
Variables :

| Name  | Type    | Attribute       | Description  |
|-------|---------|-----------------|--|
| K1    | Boolean | <b>Input</b>    | push K1 to start running motor<br>(pushbutton 1 on the I-8xx7) |
| Motor | Boolean | <b>Output</b>   | True means to run motor, False means to stop motor             |
| Gate  | Boolean | <b>Output</b>   | True means to open gate, False means to close gate             |
| M1    | Boolean | <b>Internal</b> | event generated at 5 sec later when K1 is pushed               |
| M2    | Boolean | <b>Internal</b> | event generated at 15 sec later when K1 is pushed              |
| M3    | Boolean | <b>Internal</b> | event generated at 18 sec later when K1 is pushed              |
| T1    | Timer   | <b>Internal</b> | Time past  |

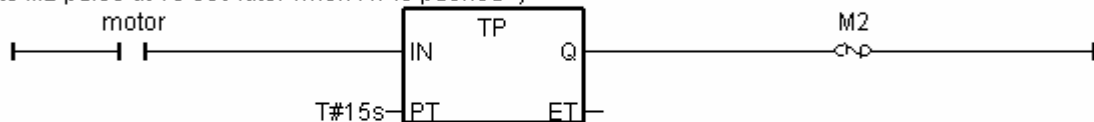
(\* Push K1 to starting running motor \*)



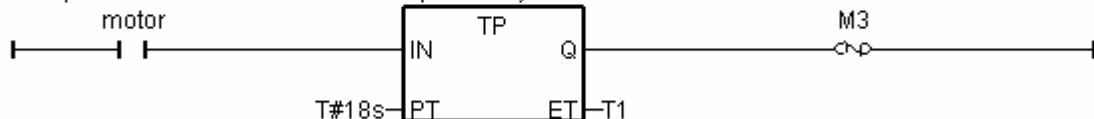
(\* To generate M1 pulse at 5 sec later when K1 is pushed \*)



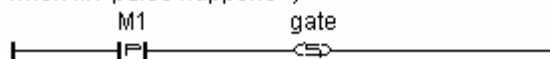
(\* To generate M2 pulse at 15 sec later when K1 is pushed \*)



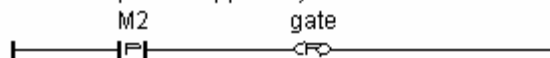
(\* To generate M3 pulse at 18 sec later when K1 is pushed \*)



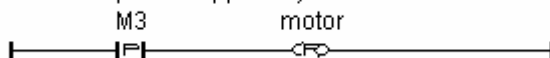
(\* Open gate when M1 pulse happens \*)



(\* Close gate when M2 pulse happens \*)



(\* Stop motor when M3 pulse happens \*)





### Demo\_03: Do something at specific weekday & some time interval

Location: I-8000 CD-ROM: \napdos\isagraf\8000\demo\ “demo\_03.pia”

Variables :

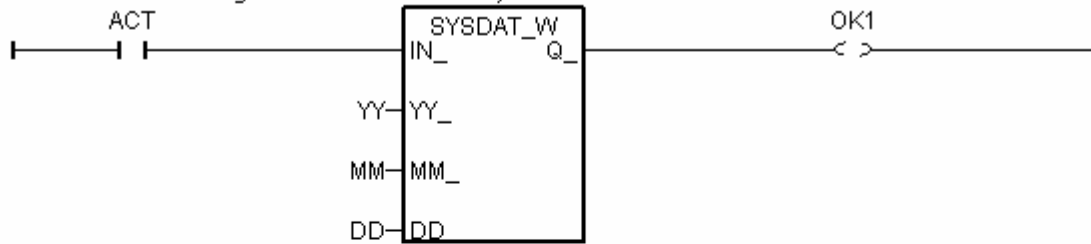
| Name     | Type    | Attribute | Description   |
|----------|---------|-----------|---|
| Year     | Integer | Internal  | System year, 2001 ~   |
| Month    | Integer | Internal  | System Month, 1 ~ 12  |
| Day      | Integer | Internal  | System date, 1 ~ 31   |
| Wday     | Integer | Internal  | System Wday, 1:Monday ~ 6:Saturday, 7:Sunday                        |
| Hour     | Integer | Internal  | System hour, 0 ~ 23   |
| Minute   | Integer | Internal  | System minute, 0 ~ 59   |
| Second   | Integer | Internal  | System second, 0 ~ 59   |
| YY       | Integer | Internal  | New system year to set  |
| MM       | Integer | Internal  | New system month to set   |
| DD       | Integer | Internal  | New system date to set  |
| HH       | Integer | Internal  | New system hour to set  |
| Mn       | Integer | Internal  | New system minute to set  |
| Sec      | Integer | Internal  | New system second to set  |
| Act      | Boolean | Internal  | Trigger to set new date   |
| Act1     | Boolean | Internal  | Trigger to set new time   |
| OK1      | Boolean | Internal  | Read back of “SYSDAT_W”   |
| OK2      | Boolean | Internal  | Read back of “SYSTIM_W”   |
| L1 ~ L3  | Boolean | Internal  | Simulate Boolean Output 1 to 3                                      |
| Time_val | Integer | Internal  | unit is sec, = 3600 x hour + 60 x minute + sec, every day = 0~86399 |

Operation action:

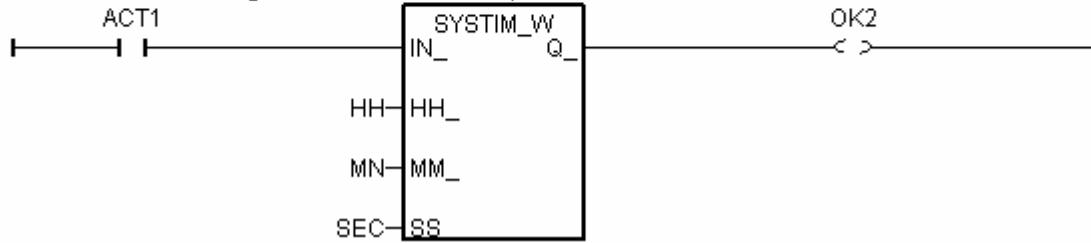
1. Monday ~ Saturday, L1 ~ L3, 09:00:00 ~ 18:00:00 ON
2. Sunday, L1, 13:00:00 ~ 20:00:00 ON
3. Other time, L1 ~ L3 are all OFF

## Ladder program : get\_time

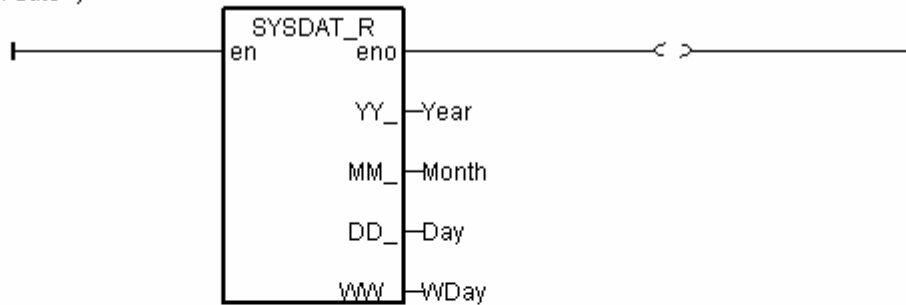
(\* set system date when ACT rising from FALSE to TRUE \*)



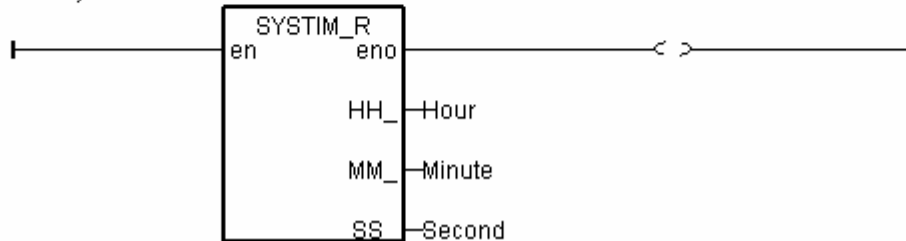
(\* set system time when ACT1 rising from FALSE to TRUE \*)



(\* get system date \*)



(\* get system time \*)



ST program : control

```
time_val := 3600 * hour + 60 * minute + second ; (* calculate time in sec. *)
```

```
(* set as False at the beginning of this ST program*)
```

```
L1 := False ;
```

```
L2 := False ;
```

```
L3 := False ;
```

```
(* Monday ~ Saturday, L1 ~ L3, 09:00:00 ~ 18:00:00 ON *)
```

```
IF ( Wday >= 1 ) AND ( Wday <= 6 ) THEN
```

```
  IF (time_val >= 32400 ) AND ( time_val <= 64800 ) THEN
```

```
    L1 := True ;
```

```
    L2 := True ;
```

```
    L3 := True ;
```

```
  END_IF ;
```

```
END_IF ;
```

```
(* Sunday, L1, 13:00:00 ~ 20:00:00 ON *)
```

```
IF ( Wday = 7 ) THEN
```

```
  IF ( time_val >= 46800 ) AND ( time_val <= 72000 ) THEN
```

```
    L1 := True ;
```

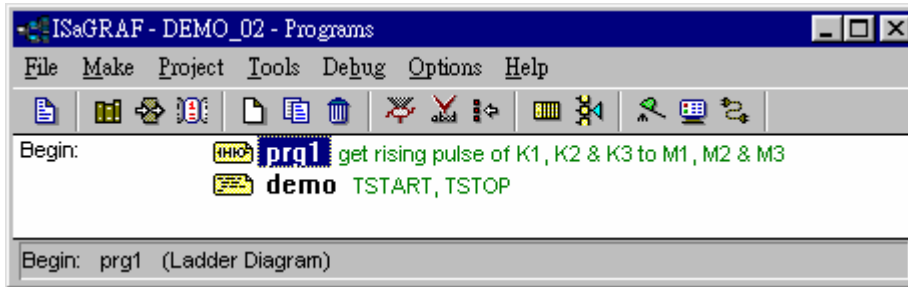
```
  END_IF ;
```

```
END_IF ;
```

### 11.3.1 Demo\_02 : Start, Stop And Reset Timer

Location: I-8000 CD-ROM: \napdos\isagraf\8000\demo\ “demo\_02.pia”

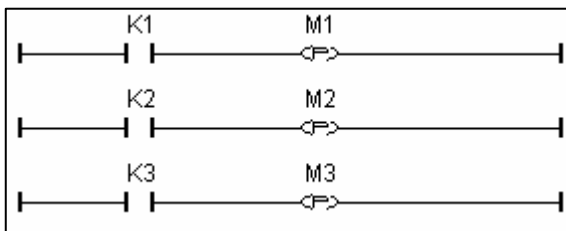
Project architecture:



Variables :

| Name | Type    | Attribute       | Description  |
|------|---------|-----------------|--|
| M1   | Boolean | <b>Internal</b> | Indicate a rising pulse of K1                          |
| M2   | Boolean | <b>Internal</b> | Indicate a rising pulse of K2                          |
| M3   | Boolean | <b>Internal</b> | Indicate a rising pulse of K3                          |
| K1   | Boolean | <b>Input</b>    | Pushbutton 1   |
| K2   | Boolean | <b>Input</b>    | Pushbutton 2   |
| K3   | Boolean | <b>Input</b>    | Pushbutton 3   |
| L1   | Boolean | <b>Output</b>   | Output 1   |
| L2   | Boolean | <b>Output</b>   | Output 2   |
| L3   | Boolean | <b>Output</b>   | Output 3   |
| T1   | Timer   | <b>Internal</b> | Operation timer, <b>initial value is set at "T#0s"</b> |

LD program “prg1” :



Get rising pulse of K1, K2, K3 and save to M1, M2, & M3

ST program “demo” :

(\* Start timer \*)

**IF M1 THEN**  
    **TSTART(T1) ;**  
**END\_IF ;**

“TSTART” will start ticking the “T1” timer

(\* Stop timer \*)

**IF M2 THEN**  
    **TSTOP(T1) ;**  
**END\_IF ;**

“TSOP” will stop ticking “T1” timer

(\* Reset timer \*)

**IF M3 THEN**  
    **T1 := T#0s ;**  
**END\_IF ;**

Reset “T1” timer to 0 sec.

(\* Output L1 ~ L3 \*)

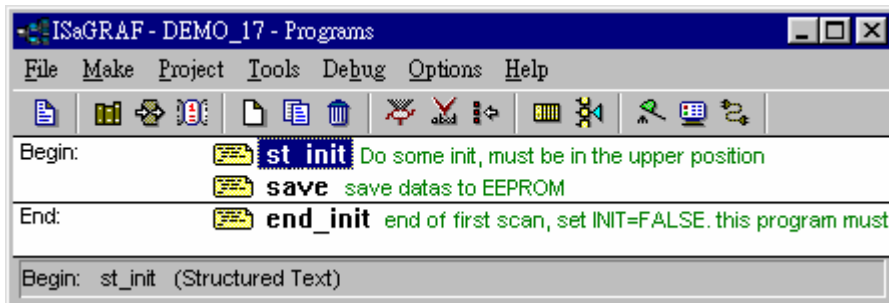
**L1 := ( T1 > T#2s ) AND ( T1 < T#15s ) ;**  
**L2 := L1 ;**  
**L3 := L1 ;**

“L1” will be TRUE between 2 and 15 sec of the value of “T1”

### 11.3.2 Demo\_17 : R/W Integer Value From/To The EEPROM

Location: I-8000 CD-ROM: \napdos\isagraf\8000\demo\ “demo\_17.pia”

Project architecture:



Variables: (Please refer to Chapter 2.6 for more information about Variable Array)

| Name        | Type    | Attribute | Description   |
|-------------|---------|-----------|---|
| V[0..7]     | Integer | Internal  | Variable Array, Dim is 8<br>If modifying the value of V[0..7], the new value will be stored to the EEPROM |
| Old_V[0..7] | Integer | Internal  | Variable Array, Dim is 8<br>Old value of V[0..7]  |
| TEMP        | Boolean | Internal  | Internal use  |
| ii          | Integer | Internal  | Index of “for” loops  |
| INIT        | Boolean | Internal  | If controller is just powered up, <b>initial value is TRUE</b>  |

ST program “st\_init” :

```
if INIT = TRUE then (* First scan cycle *)  
  
  (* Read 8 integers from EEPROM *)  
  (* save them to Old_V[0..7] , V[0..7] *)  
  for ii := 0 to 7 do  
    V[ii] := eep_n_r( ii+1 ) ;  
    Old_V[ii] := V[ii] ;  
  End_for ;  
  
  (* remove protection of EEPROM *)  
  TEMP := eep_en( ) ;  
  
end_if ;
```

Read long integers stored at the position from 1 to 8 of the eeprom at the first scan cycle.

Remove the protection of EEPROM, so that it can be written later.

ST program “save” :

```
(* save V[0..7] to EEPROM if it is modified *)  
(* You will find write to EEPROM take lots of time, about 23ms for each eep_n_w *)
```

```
for ii := 0 to 7 do
```

```
  IF V[ii] <> Old_V[ii] THEN  
    TEMP := eep_n_w( ii+1 , V[ii] ) ;  
    Old_V[ii] := V[ii] ;  
  END_IF ;
```

The value will be saved to eeprom only when the current value is changed.

Then update Old value to the new value.

```
End_for ;
```

ST program “end\_init” :

```
INIT := FALSE ;
```

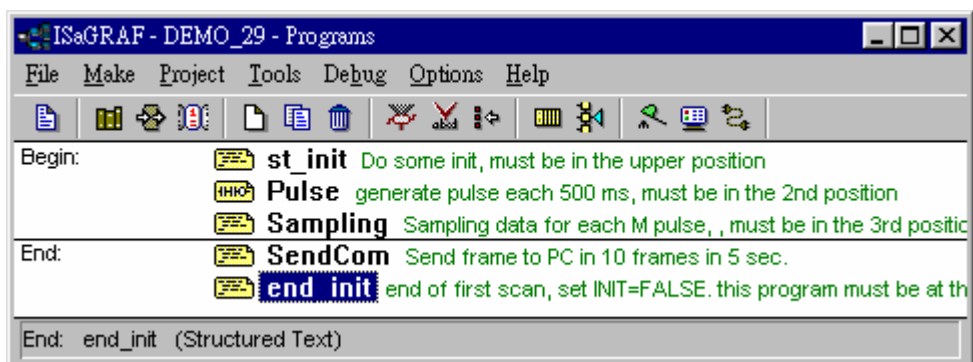
Set “INIT” to False, so that “INIT” is only TRUE at the first scan cycle since it is declared with the initial value - TRUE.

### 11.3.3 Demo\_29: Store 1200 Short Int Every 75 sec & Send To PC Via Com3

This demo program is to save the 8 analog input value (8 samples) of the I-87017 to the short-integer array every 500ms. Then when the number of samples reach 1200, these samples will be divided in 10 frames, each frame contain 120 samples, and sent to one PC via COM3 (RS232/RS485).

Location: I-8000 CD-ROM: \napdos\isagraf\8000\demo\ “demo\_29.pia”

Project architecture:

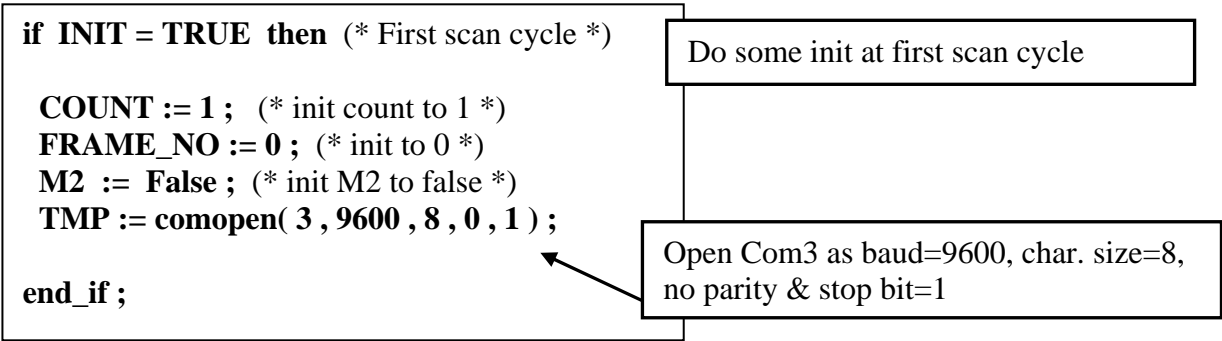


Variables :

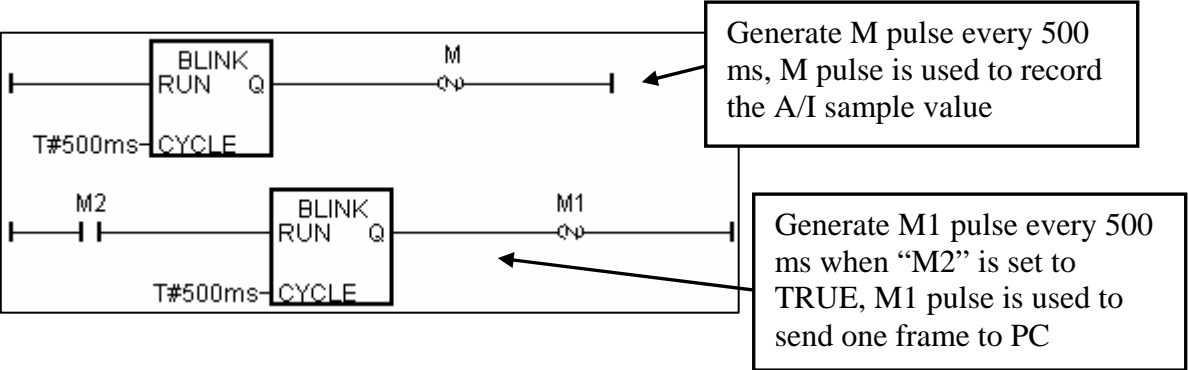
| Name     | Type    | Attribute | Description  |
|----------|---------|-----------|--|
| M        | Boolean | Internal  | pulse to store a sample  |
| M1       | Boolean | Internal  | pulse to send frame  |
| M2       | Boolean | Internal  | To generate M1 pulse   |
| INIT     | Boolean | Internal  | If controller is just powered up, <b>initial value is TRUE</b> |
| TMP      | Boolean | Internal  | For temporal use   |
| A1       | Integer | Input     | Connect to Ch. 1 of I-87017                                    |
| A2       | Integer | Input     | Connect to Ch. 2 of I-87017                                    |
| A3       | Integer | Input     | Connect to Ch. 3 of I-87017                                    |
| A4       | Integer | Input     | Connect to Ch. 4 of I-87017                                    |
| A5       | Integer | Input     | Connect to Ch. 5 of I-87017                                    |
| A6       | Integer | Input     | Connect to Ch. 6 of I-87017                                    |
| A7       | Integer | Input     | Connect to Ch. 7 of I-87017                                    |
| A8       | Integer | Input     | Connect to Ch. 8 of I-87017                                    |
| count    | Integer | Internal  | No. of sample(1~1200) that is processing, <b>init value=1</b>  |
| position | Integer | Internal  | position in current short integer array, 1 ~ 256               |
| No       | Integer | Internal  | current short integer array No. which is processing            |
| Frame_No | Integer | Internal  | only = 0 ~ 10  |
| TMP_VAL  | Integer | Internal  | For temporal use   |



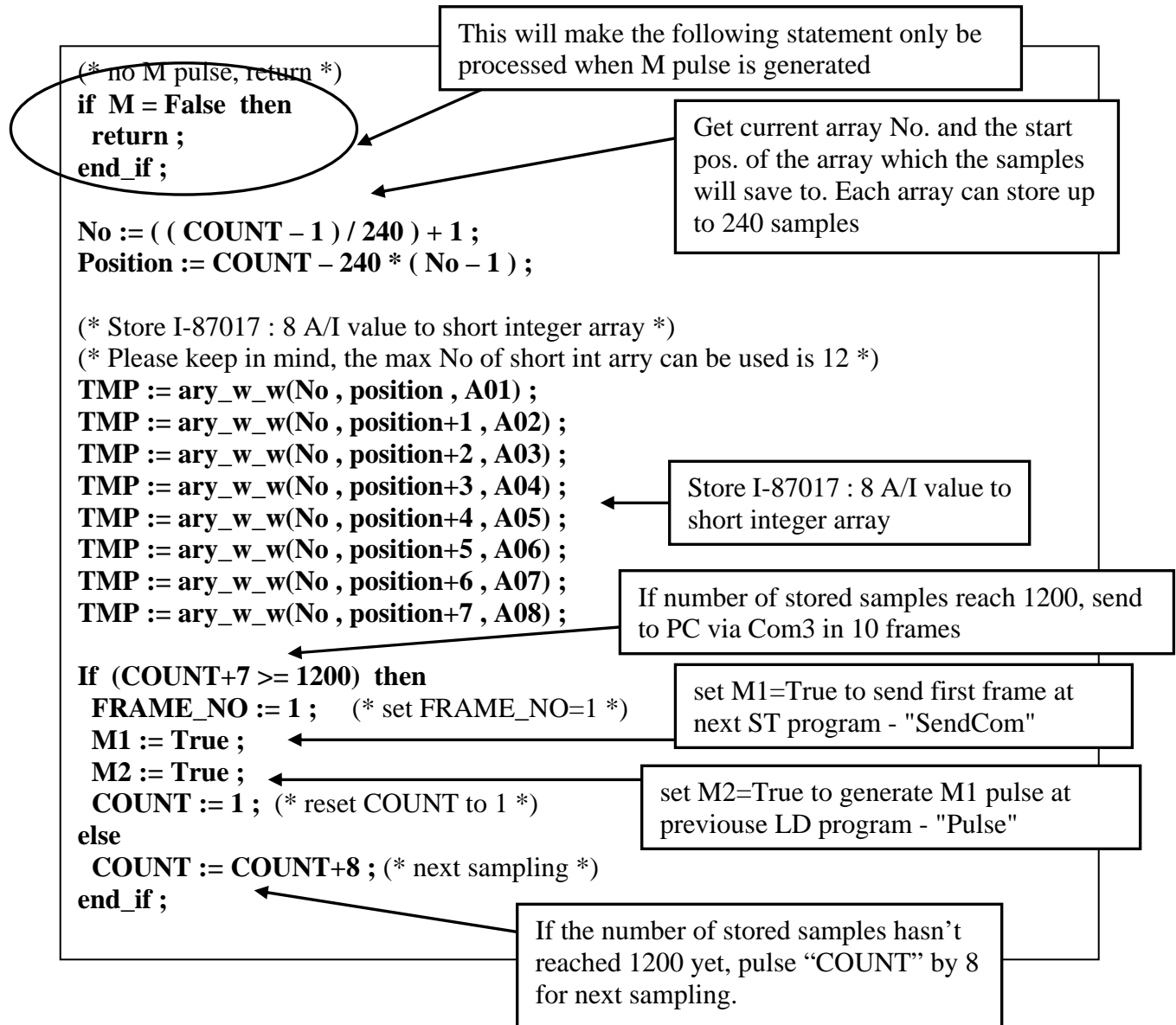
ST program “st\_init” :



LD program “Pulse” :



## ST program "Sampling" :



ST program "SendCom" :

**If M1 = False then  
Return ;  
end\_if ;**

This will make the following statement only be processed when M1 pulse is generated

User defined frame format : Each contains 120 short integers

|                 | STX | DLE  | FRAME_NO | DATA  | ETX  |
|-----------------|-----|------|----------|-------|------|
| number of bytes | 1   | 1    | 1        | 120x2 | 1    |
| value           | 0x2 | 0x10 | 1~10     | ?     | 0x03 |

When "FRAME\_NO" is between 1 and 10

**If ( FRAME\_NO >= 1 ) and ( FRAME\_NO <= 10 ) then**

**No := ( FRAME\_NO - 1 ) / 2 + 1 ;**

Get the short integer array No to process. Keep in mind, each array store up to 240 samples. (in other word -- 2 frames)

**case ( FRAME\_NO - 2 \* ( No - 1 ) ) of**

**1: position := 1 ;**

**2: position := 121 ;**

**end\_case ;**

Get starting position inside the array

Send one frame via Com3

**TMP := comwrite( 3 , 16#2 ) ;** (\* write one byte = STX to Com3 \*)

**TMP := comwrite( 3 , 16#10 ) ;** (\* write one byte = DLE to Com3 \*)

**TMP := comwrite( 3 , FRAME\_NO ) ;** (\* write frame No = 1 ~ 10 to Com3 \*)

(\* write 120 short integers inside the array to Com3 \*)

**TMP := comay\_ww( 3 , No , 120 , position ) ;**

**TMP := comwrite( 3 , 16#3 ) ;** (\* write one byte = ETX to Com3 \*)

**M1 := False;**

500 ms later, send next frame. "M1" will be turned ON after 500 ms later at LD program - "Pulse"

**If ( FRAME\_NO = 10 ) then**

**FRAME\_NO := 0 ;**

**M2 := False ;**

**else**

**FRAME\_NO := FRAME\_NO + 1 ;** (\* for next cycle \*)

**end\_if ;**

If all frames are sent, reset "FRAME\_NO" to 0 , and set "M2" to FALSE to stop to generate "M1"

**end\_if ;**

If some frames have not been sent yet, plus "FRAME\_NO" by 1 to send next frame when next "M1" is generated

ST program “end\_init” :

**INIT := FALSE ;**

Set “INIT” to False, so that “INIT” is only TRUE at the first scan cycle since it is declared with the initial value - TRUE.

How to test ?

Plug one I-87017 in the slot 0 of the I-8xx7 controller.

Download Demo\_29 to the controller.

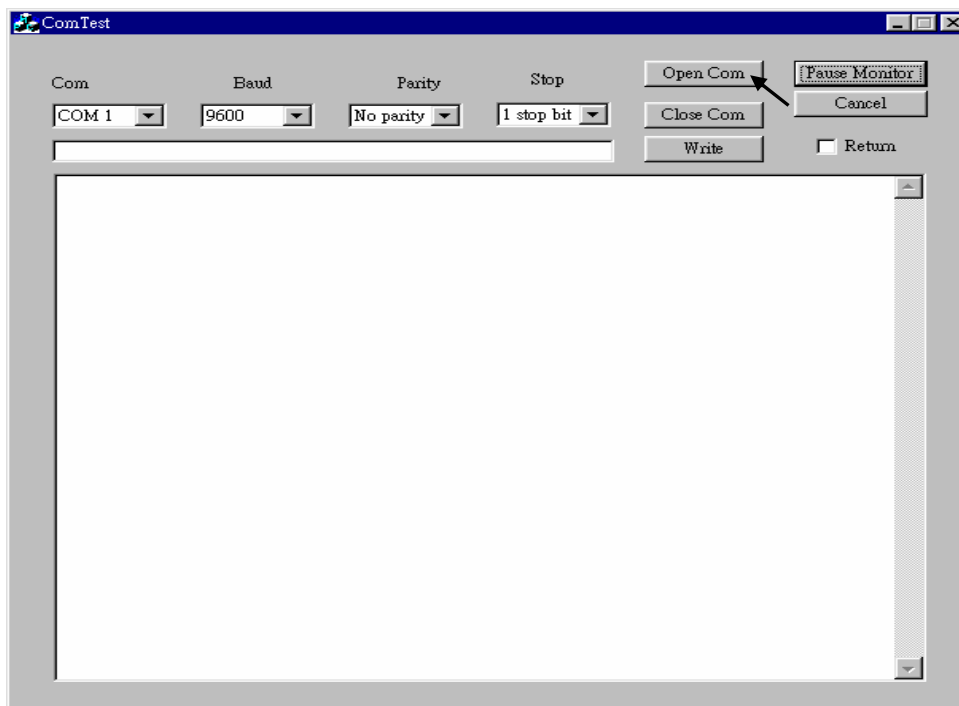
Prepare a RS232 cable to connect Com3 of the controller to Com1 of your PC.

There is one utility named “ComTest.exe” located in the ICP DAS’s CD-ROM. Copy it to your PC.

“\Napedos\ISaGRAF\some\_utility\Comtest.exe” or you may obtain it from below site.

[ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/some\\_utility/](ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/some_utility/)

Execute “ComTest” and select the parameter to “COM1” , “9600” , “No parity” , “1 stop bit” and then click on “Open Com”.



You will receive 10 frames coming from the target controller every 75 seconds.

### 11.3.4 Demo\_33 : R/W User Defined protocol Via Com3:RS232/RS485

This demo program can let I8417/ 8817/ 8437/ 8837 accept commands coming from PC via a RS232 cable. The command protocol format can be defined by the user. We use the below protocol format in this example.

Command is case insensitive, that means M1 & m1 are same

Protocol Format:

PC req.

**M1<CR>** : Change to Mode 1

**M2<CR>** : Change to Mode 2

**M3<CR>** : Change to Mode 3

**Txxxx<CR>** : Change Period time to xxxx ms

for ex. T250<CR> will change period time to 250ms

Controller Ans.

**OK<CR>**

PC req.

**M?<CR>** : Request the current Mode

Controller Ans.

**Mx<CR>** : for ex. M1 means Mode 1

PC req.

**T?<CR>** : Request the current Period time

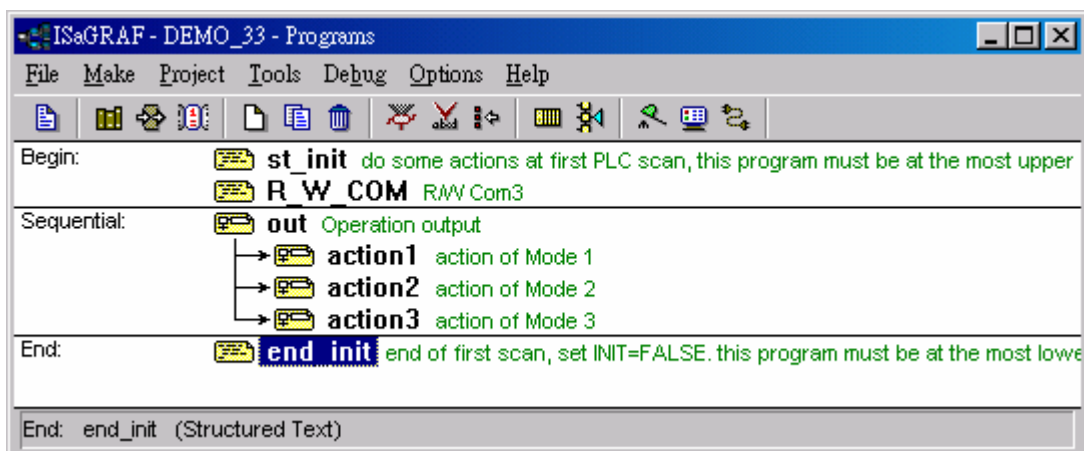
Controller Ans.

**Txxxx<CR>** : for ex. T1500 means Period time is 1500ms

Timeout:

a valid command should be completely sent in 5 sec.

Project architecture:



Variables :

| Name     | Type    | Attribute       | Description  |
|----------|---------|-----------------|--|
| L1       | Boolean | <b>Output</b>   | Output 1   |
| L2       | Boolean | <b>Output</b>   | Output 2   |
| L3       | Boolean | <b>Output</b>   | Output 3   |
| INIT     | Boolean | <b>Internal</b> | If controller is just powered up, <b>initial value is TRUE</b> |
| TMP      | Boolean | <b>Internal</b> | For temporal use   |
| Mode     | Integer | <b>Internal</b> | Operation Mode, range from 1 to 3                              |
| Step     | Integer | <b>Internal</b> | Processing step  |
| NUM      | Integer | <b>Internal</b> | Received valid byte number                                     |
| Num_com3 | Integer | <b>Internal</b> | return value of Comary_R                                       |
| byt      | Integer | <b>Internal</b> | Current operating byte   |
| index    | Integer | <b>Internal</b> | Index of byte array  |
| CMD      | Integer | <b>Internal</b> | command type, M, m, T, or t                                    |
| TMP_val  | Integer | <b>Internal</b> | for temporal use   |
| ii       | Integer | <b>Internal</b> | for temporal use   |
| T1       | Timer   | <b>Internal</b> | Period time, valid range is 50 ~ 9999 ms                       |
| tout     | Timer   | <b>Internal</b> | timer to measure timeout, tick when first valid byte recved    |

ST program “st\_init” :

```

If INIT = TRUE then

  (* Init *)
  Mode := 1 ;
  STEP := 0 ;
  T1 := T#500ms ;
  NUM := 0 ;
  tout := T#0s ;

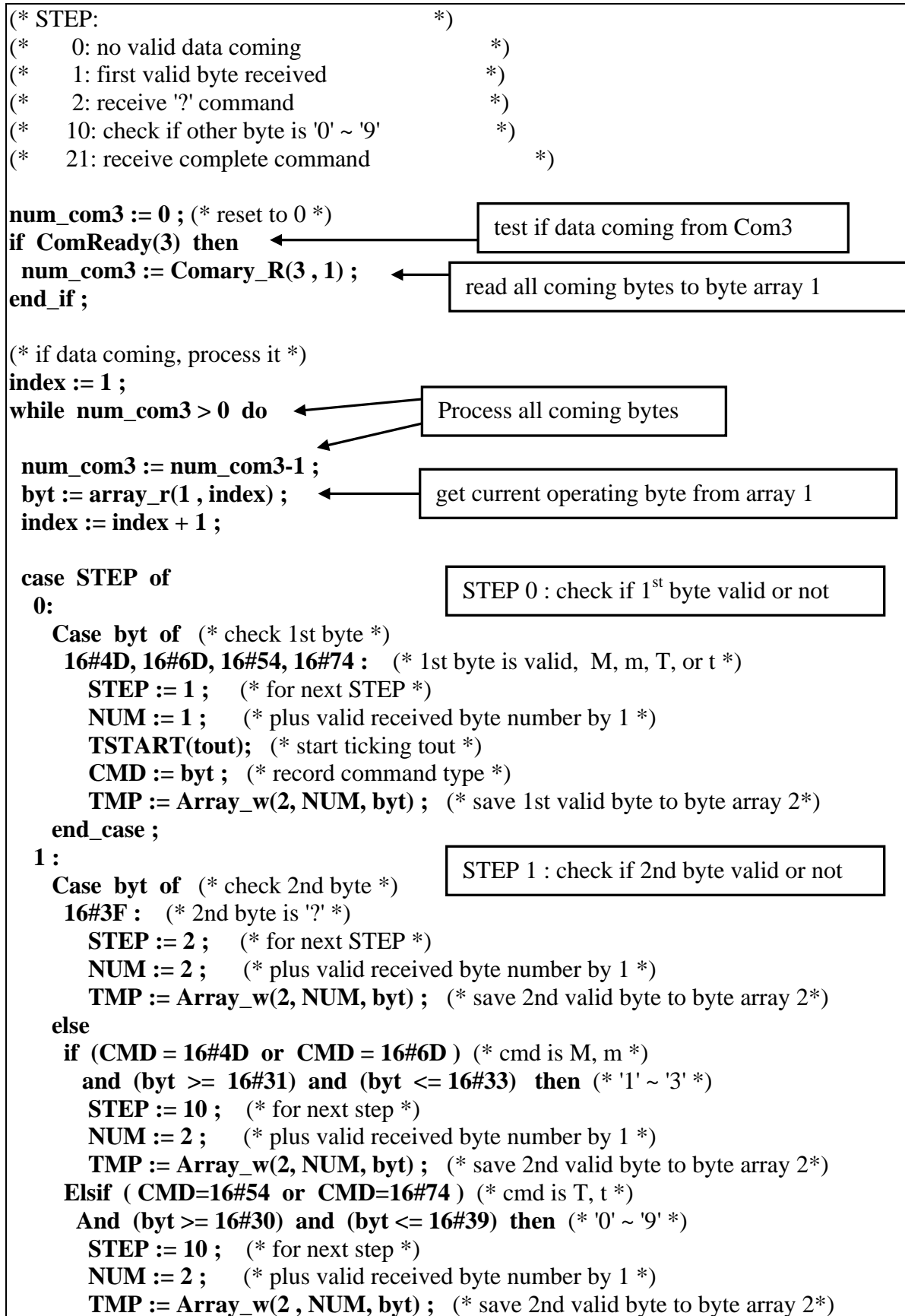
  (* Open Com3 as baud=9600, char. size=8, no parity & stop bit=1 *)
  TMP := comopen( 3 , 9600 , 8 , 0 , 1 ) ;

end_if ;

```

Do some init at the first scan cycle

ST program "R\_W\_COM" :



```

else
  STEP := 0 ; (* not valid data, reset STEP to 0 *)
  TSTOP(tout) ; (* stop ticking "tout" *)
  tout := T#0s ; (* reset "tout" *)
  NUM := 0 ; (* reset NUM *)
end_if ;
end_case ;

```

STEP 2 : after receive 2<sup>nd</sup> byte = '?',  
check if 3<sup>rd</sup> byte is <CR>

2 :

```

if byt = 16#0D then (* check 3rd byte is <CR> or not *)
  STEP := 21 ; (* complete command is received *)
  (* send answer to Com3 *)

```

```

case CMD of

```

```

  16#4D, 16#6D : (* M or m *)

```

```

  TMP := ComWrite(3, 16#4D) ; (* M *)

```

```

  TMP := ComWrite(3, Mode+16#30) ; (* Mode *)

```

```

  TMP := ComWrite(3, 16#0D) ; (* <CR> *)

```

```

  16#54, 16#74 : (* T or t *)

```

```

  TMP := ComWrite(3, 16#54) ; (* T *)

```

```

  TMP := ComStr_w(3, MSG(ANA(T1))) ; (* Timer value *)

```

```

  TMP := ComWrite(3, 16#0D) ; (* <CR> *)

```

```

end_case ;

```

```

else

```

```

  STEP := 0 ; (* not valid data, reset STEP to 0 *)

```

```

  TSTOP(tout) ; (* stop ticking "tout" *)

```

```

  tout := T#0s ; (* reset "tout" *)

```

```

  NUM := 0 ; (* reset NUM *)

```

```

end_if ;

```

10 :

```

If (byt = 16#0D) then (* received <CR> *)

```

```

  STEP := 21 ; (* complete command is received *)

```

```

case CMD of

```

```

  16#4D, 16#6D : (* M or m *)

```

```

  Mode := Array_r(2, 2) - 16#30 ; (* Change Mode *)

```

```

  (* send answer to Com3 *)

```

```

  TMP := ComStr_w(3, 'OK') ;

```

```

  TMP := ComWrite(3, 16#0D) ; (* <CR> *)

```

```

  16#54, 16#74 : (* T or t *)

```

```

  (* get Period *)

```

```

  TMP_val := 0 ;

```

```

  for ii := 1 to NUM-1 do

```

```

    TMP_val := 10*TMP_val + (Array_r(2, 1+ii) - 16#30) ;

```

```

  end_for ;

```

```

  if (TMP_val >= 50) and (TMP_val < 10000) then (* must be in 50 ~ 9999 ms *)

```

```

    T1 := TMR(TMP_val) ; (* Change T1 *)

```

```

    (* send answer to Com3 *)

```

```

    TMP := ComStr_w(3, 'OK') ;

```

```

    TMP := ComWrite(3, 16#0D) ; (* <CR> *)

```

Receive valid "M?" command,  
reply "Mx", x = '1' ~ '3'

Receive valid "T?" command,  
reply "Txxxx", x = '0' ~ '9'

STEP 10 : check 3<sup>rd</sup> and other byte  
when command is "Mx" or "Txxxx"

Receive valid "Mx" command,  
chang Mode value and reply  
"OK" to PC

Receive valid "Txxx" command,  
change T1 value and reply "OK" to  
PC



```

    end_if ;
    end_case ;

    elsif (byt >= 16#30) and (byt <= 16#39) then (* '0' ~ '9' *)

        STEP := 10 ;    (* for next step *)
        NUM := NUM+1 ;    (* plus valid received byte number by 1 *)
        TMP := Array_w(2, NUM, byt) ; (* save other valid byte to byte array 2*)

        if NUM > 5 then    (* command is too long, drop it *)
            STEP := 0 ;    (* reset STEP *)
            TSTOP(tout) ;    (* stop ticking "tout" *)
            tout := T#0s ;    (* reset "tout" *)
            NUM := 0 ;    (* reset NUM *)
            EXIT ;    (* exit while loop *)
        end_if ;

    else
        STEP := 0 ;    (* not valid data, reset STEP to 0 *)
        TSTOP(tout) ;    (* stop ticking "tout" *)
        tout := T#0s ;    (* reset "tout" *)
        NUM := 0 ;    (* reset NUM *)

    end_if ;

end_case ;

end_while ;

(* Check timeout *)
If tout > T#5s then (* if timeout *)
    STEP := 0 ;    (* reset STEP *)
    TSTOP(tout) ;    (* stop ticking "tout" *)
    tout := T#0s ;    (* reset "tout" *)
    NUM := 0 ;    (* reset NUM *)
end_if ;

(* reset STEP to 0 *)
if STEP = 21 then
    TSTOP(tout) ;    (* stop ticking "tout" *)
    tout := T#0s ;    (* reset "tout" *)
    NUM := 0 ;    (* reset NUM *)
    STEP := 0 ;
end_if ;

```

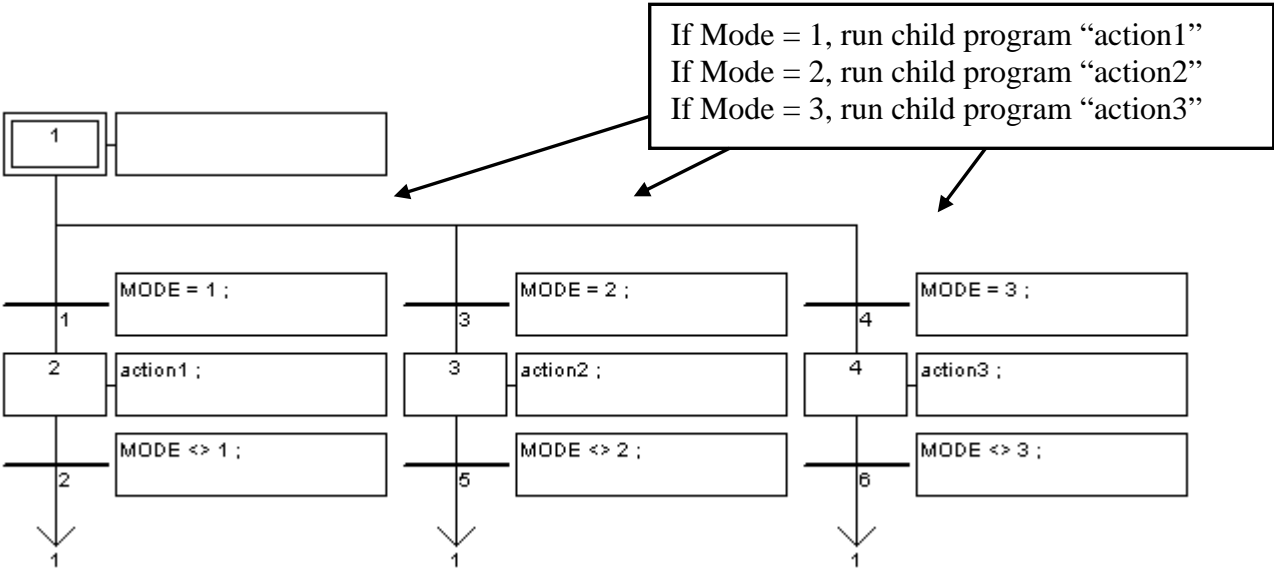
Receive '0' ~ '9', command is not completely received yet, process next byte

Check timeout, a valid complete command should be received in 5 seconds

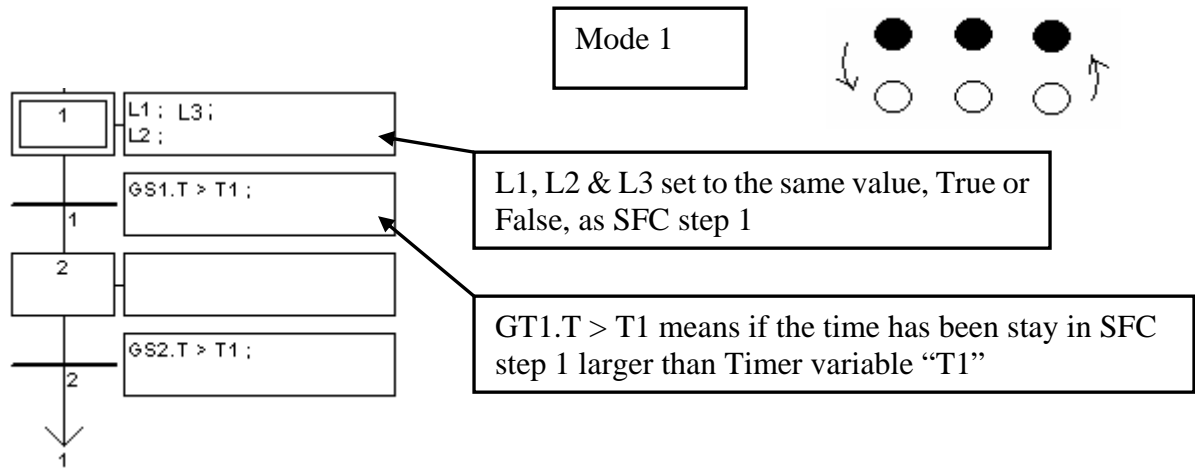
Valid command has been processed, reset to STEP 0

SFC program “Out” :

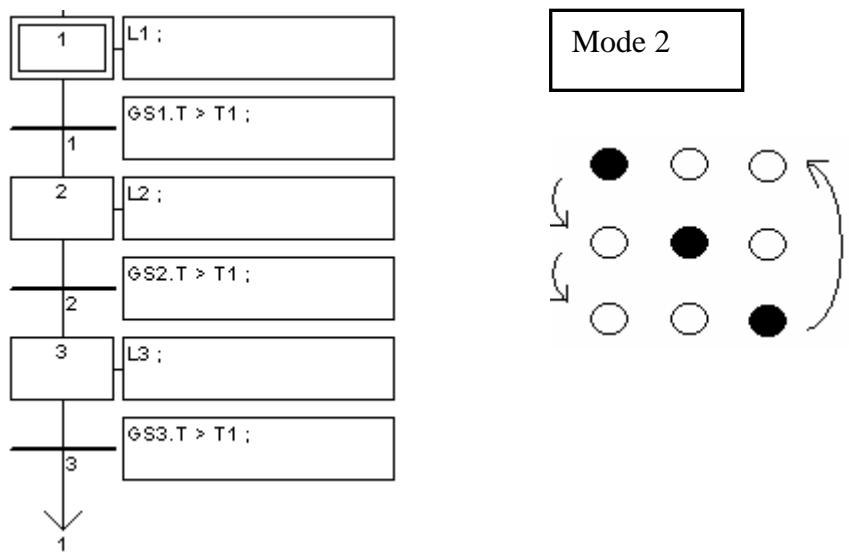
Each statement should end with a colon “,”



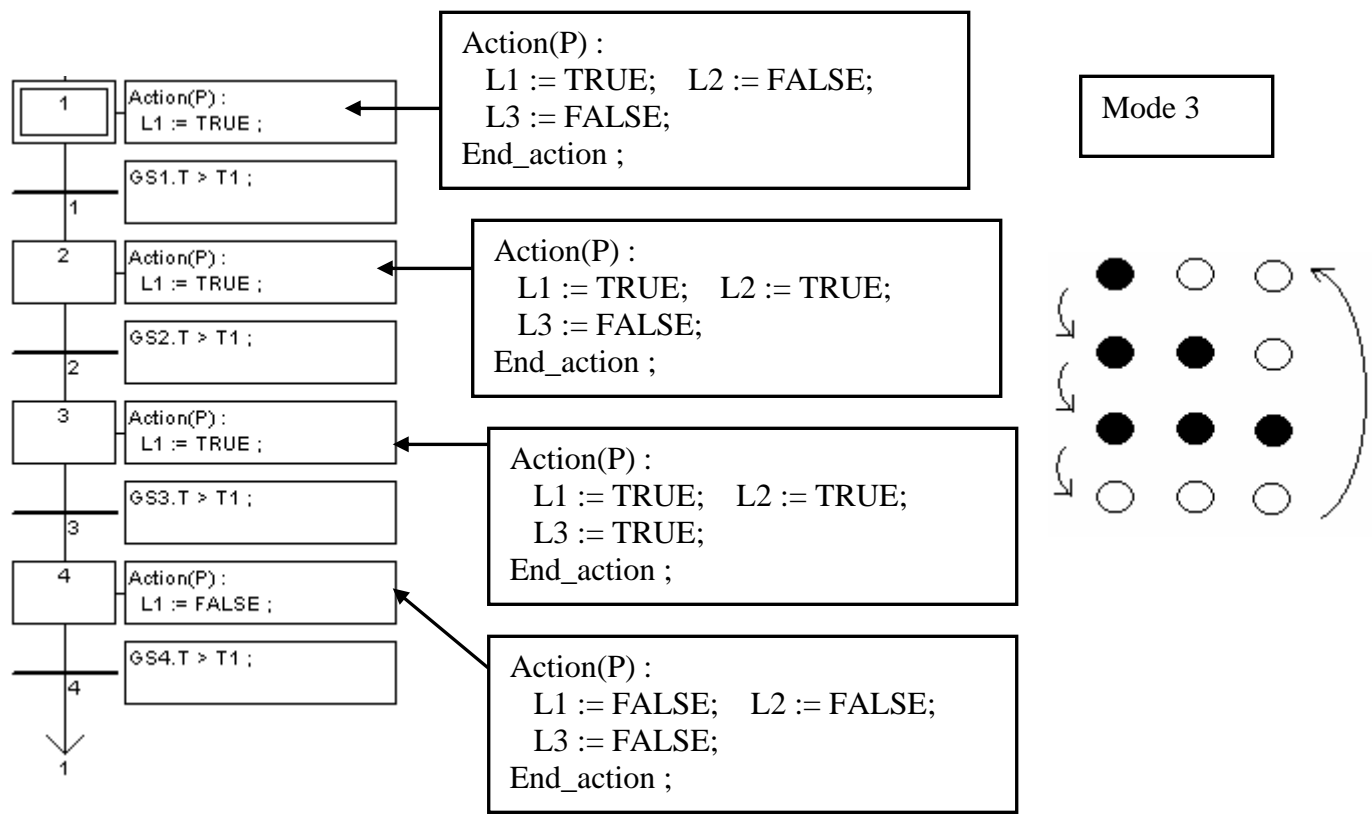
SFC child program “action1” :



SFC child program “action2” :



SFC child program “action3” :



ST program “end\_init” :

```
INIT := FALSE ;
```

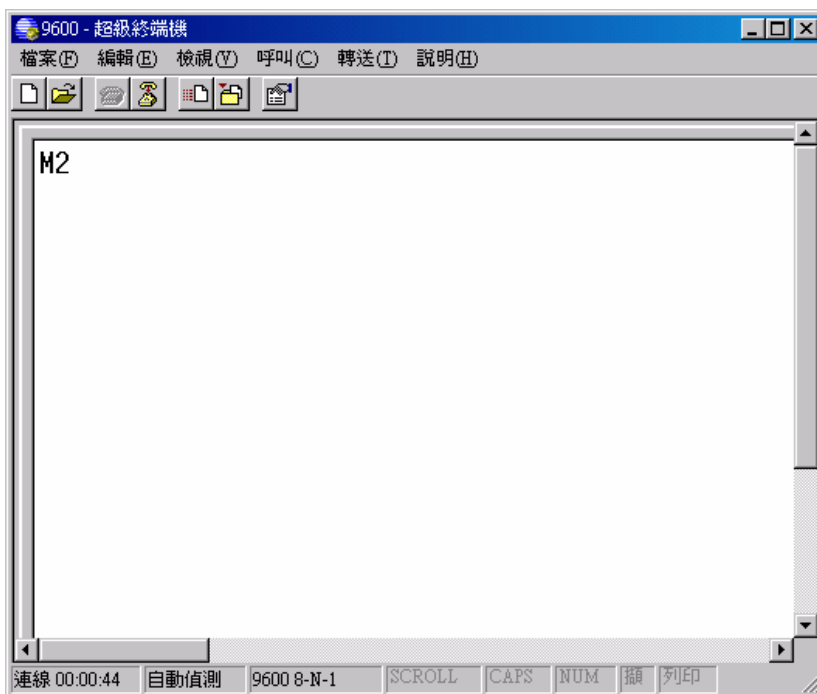
Set “INIT” to False, so that “INIT” is only TRUE at the first scan cycle since it is declared with the initial value - TRUE.

How to test ?

- 1 . Download Demo\_33 to the controller.
2. Prepare a RS232 cable to connect Com3 of the controller to Com1 of your PC.
3. There is one utility named “ComTest.exe” located in the ICP DAS’s CD-ROM. Copy it to your PC.  
“\Napdos\ISaGRAF\some\_utility\Comtest.exe” or you may obtain it from below site.  
[ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/some\\_utility/](ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/some_utility/)
4. You may open a “Hyper Terminal” with Com1, 9600, N, 8, 1 and “No flow control” to type the following command to test

```
M2<CR>    : change to mode 2
T?<CR>    : request current period time
T200<CR>   : change to 200ms
T1500<CR>  : change to 1500ms
M?<CR>    : request current mode
```

<CR> is the return char.



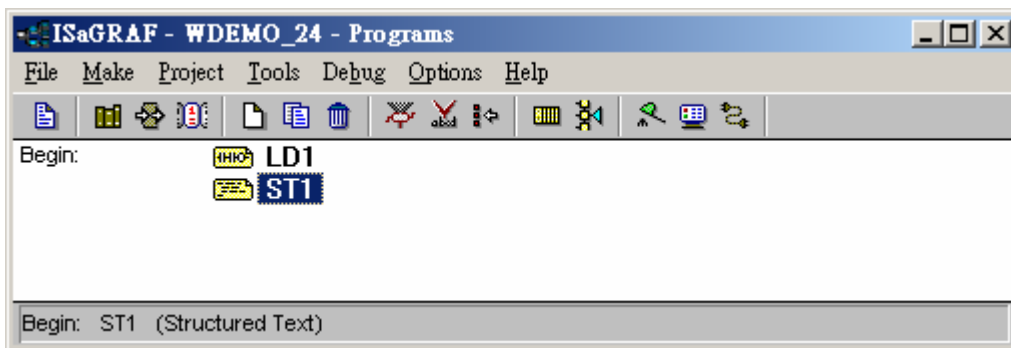
### 11.3.5 Send string to COM2 or COM3 when alarm 1 to 8 happens. (Access to variables as array)

This demo program can be running in Wincon-8xx7/8xx6 or in I-8xx7. Please init “PORT” as 2 if your target is Wincon, while 3 for I-8xx7.

Variables : (Please refer to Chapter 2.6 for more informayion about Variable Array)

| Name         | Type    | Attribute       | Description  |
|--------------|---------|-----------------|--|
| INIT         | Boolean | <b>Internal</b> | Init as TRUE, True indicates first PLC scan cycle        |
| TMP          | Boolean | <b>Internal</b> | Internal use   |
| Tick1        | Boolean | <b>Internal</b> | pulse generated every 1 sec to counting time             |
| IN[0..7]     | Boolean | <b>Input</b>    | Variable Array, Dim is 8<br>The input signal             |
| OLD_IN[0..7] | Boolean | <b>Internal</b> | Variable Array, Dim is 8<br>The old value of IN[0..7]    |
| ii           | Integer | <b>Internal</b> | Index of “For” loops                                     |
| Port         | Integer | <b>Internal</b> | A COM PORT Number to open, init as 2 for Wincon          |
| CNT[0..7]    | Integer | <b>Internal</b> | Variable Array, Dim is 8<br>To count the elapsed seconds |
| Msg1         | Message | <b>Internal</b> | Message to send to COM2, init length as 128              |

Project architecture:



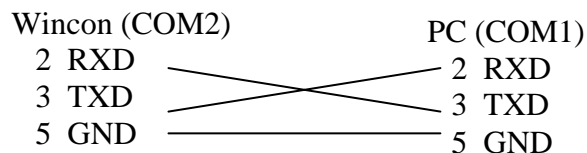
Operations:

1. If IN[0..7] rising from False to True and hold in True for at least 3 seconds, send one message = ‘Alarm N’ + <LF> <CR> to COM2. N= 1,2, ... 8 depends on which Input is triggered. For ex, if IN[2] is rising and hold in True longer than 3 seconds, send ‘Alarm 3’ + <LF> <CR> to COM2
2. If after IN[0..7] ‘s first alarm is sent and then continusly hold in True for 30 seconds, then send one more messge after every 30 second past to COM2 until the state of IN[0..7] is falling to FALSE. The string is for ex, ‘Alarm 3 , 30 sec past !’

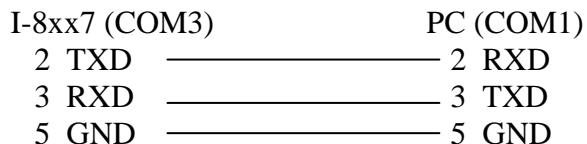
This demo project is in W-8xx7's CD-ROM:\napdos\isagraf\wincon\demo\ “wdemo\_24” or i-8xx7's CD-ROM:\napdos\isagraf\8000\demo\ “demo\_70”

### How to test ?

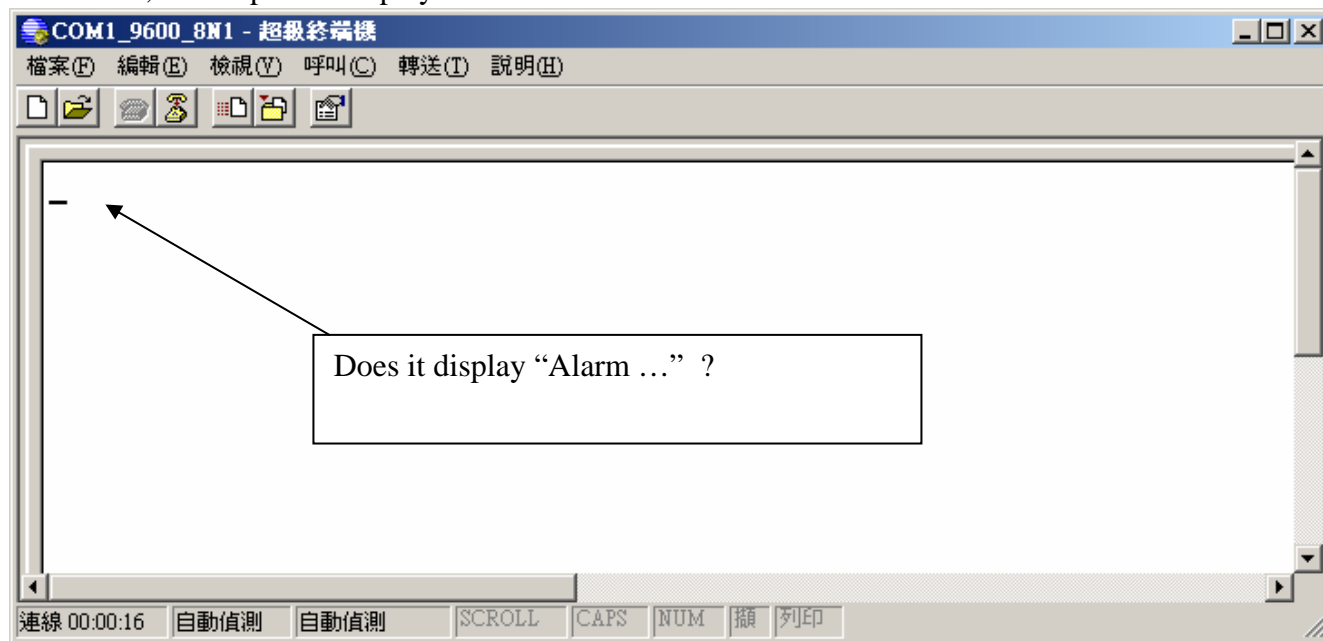
1. Please download wdemo\_24 to W-8xx7+ slot 1: I-8077 (or demo\_70 for I-8xx7+slot 0: I-8077)
2. Connect a RS232 cable between W-8xx7's COM2 to your PC's COM1



Or if you are using I-8xx7's COM3 to your PC's COM1

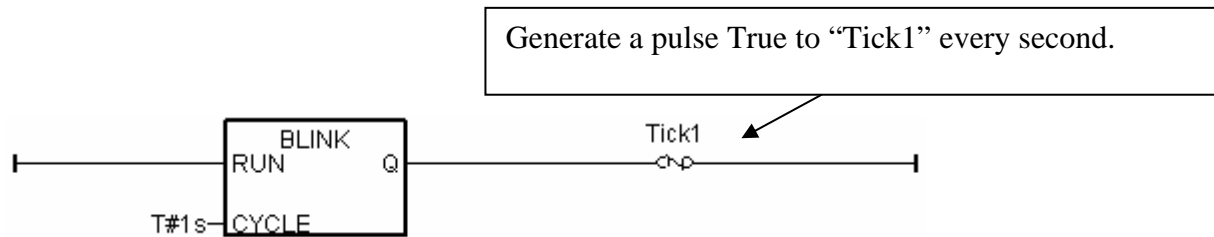


3. Open PC's Hyper terminal at COM1 with 9600, 8 char. size, no parity, 1 stop bit and No flow control. And then please switch I-8077's Input1 or 2 or ... from FALSE to TRUE and wait about three seconds. If it works, there should be a message “Alarm ...” displayed. And then please hold this input TRUE more than 30 seconds, there should be one another message “Alarm ..., 30 sec past !” displayed.



Program description:

LD1 program:



ST1 program:

(\* only do it in 1st PLC scan \*)

**if INIT then**

**INIT := FALSE ;** (\* No more 1st PLC scan cycle \*)

**TMP := COMOPEN( PORT , 9600 , 8 , 0 , 1 ) ;** (\* open COM2, 9600,8,N,1 \*)

(\* init value of CNT[0..7] to -7 \*)

**For ii := 0 to 7 do**

**CNT[ii] := -7 ;**

**end\_for ;**

**end\_if ;**

(\* test all IN[0..7] if rising from False to True \*)

**for ii := 0 to 7 do**

(\* test if IN[0..7] signal rising \*)

**If ( IN[ii] = True ) and ( OLD\_IN[ii] = False ) then**

(\* set related CNT[ ] value to -3 when Input event is triggered \*)

(\* if CNT[ ] value is not -7, it means "INPUT been triggered" \*)

(\* the CNT[ ] value will plus 1 every 1 sec past later, except the related INPUT become False \*)

**CNT[ii] := -3 ;**

**end\_if ;**

(\* if INPUT is cleared or "if related INPUT become FALSE", the related CNT[ ] value will reset to -7: "No input event happens at that INPUT channel" \*)

**if IN[ii] = False then** (\* signal is becoming FALSE \*)

(\* set related CNT[ ] value to -7: "No input event happens at that INPUT channel" \*)

**CNT[ii] := -7 ;**

**end\_if ;**

**if Tick1 then** (\* Tick1 is generated as pulse "True" every second in "LD1" program \*)

```

(* if CNT[ ] value is not -7, means the related input is triggered *)
if CNT[ii] > -7 then

    CNT[ii] := CNT[ii] + 1 ; (* plus 1, Tick1 = True means 1 sec has passed *)

    (* ----- *)
    (* INPUT event happens and 3 sec past, send 1st alarm message to COM2 *)
    if ( CNT[ii] = 0 ) then (* send 1st alarm when CNT[ ] is from -3, -2, -1 ---> 0 *)
        CNT[ii] := 0 ; (* re-start from 0 and then count to 30 second to send alarm *)
        (* send one message to COM2 *)
        msg1 := 'Alarm ' + MSG(ii+1) + ' $0A$0D' ;
        TMP := comstr_w( PORT , msg1 ) ;
    end_if ;
    (* ----- *)

    (* ----- *)
    (* INPUT event happens and every 30 second past, send one alarm message *)
    If ( CNT[ii] = 30 ) then (* send one alarm when CNT[ ] is from 0, 1, 2, ..., 30 *)
        CNT[ii] := 0 ; (* re-start from 0 and then count to 30 second to send alarm *)
        (* send one message to COM2 *)
        msg1 := 'Alarm ' + MSG(ii+1) + ', 30 sec past ! $0A$0D' ;
        TMP := comstr_w(PORT , msg1) ;
    end_if ;
    (* ----- *)

end_if ; (* "if CNT[ ] > -7 then" *)

end_if ; (* "if Tick1 then" *)

(* Update OLD_IN[ ] *)
OLD_IN[ii] := IN[ii] ;

end_for ;

```



### 11.3.7 Recording i-8017H 's Ch.1 to Ch.4 voltage input in S-256 / 512 in i-8437-80 or i-8837-80 . The sampling time is one record every 0.05 second. The record period is 1 to 10 minutes. Then PC can download this record and display it as a trend curve diagram by M.S. Excel

This demo is the “Demo\_71” can run in i-8437-80 or in i-8837-80 (80MHz) . The controller driver should be version of 3.19 or later version.

New drive: <http://www.icpdas.com/products/PAC/i-8000/isagraf-link1.htm>  
“demo\_71.pia” resides at I-8000 CD-ROM:\napdos\isagraf\8000\demo\ or  
<ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/8000/demo/> or  
[www.icpdas.com](http://www.icpdas.com) – FAQ – Software – ISaGRAF – FAQ058

i-8437-80 and i-8837-80 controller 's CPU is running at 80MHz . The speed is about 2 to 4 time faster than i-8437 and i-8837 (40MHz CPU). So it can record minimum to 25 milli-second sampling data. While using i-8437 and i-8837 (40MHz) , it can record only 100 ms or above sampling data. This minimum sampling time depends on your ISaGRAF program 's PLC scan time. If the PLC scan time is large, like 200ms, then you can do sampling only larger than 200 ms.

#### How to test this demo ?

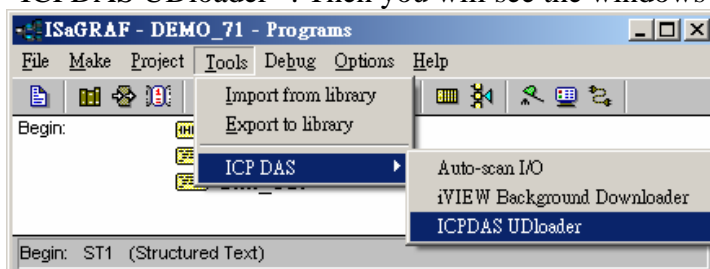
You may run VB 6.0 - “Demo\_6” in your PC to on-line control this i-8837-80 via ethernet . Or just push the pushbutton on the front panel of the i-8437-80 / 8837-80 to start / stop it.

1. Please plug one i-8024 in i-8437-80 's Slot 2, one i-8017H in Slot 3. (Note: the left-most I/O slot No. of the i-8xx7 is 0 , not 1) .Then connect Ch1. to Ch.4 voltage output of i-8024 to Ch1. to Ch.4 of i-8017H. Then power up this i-8437-80.

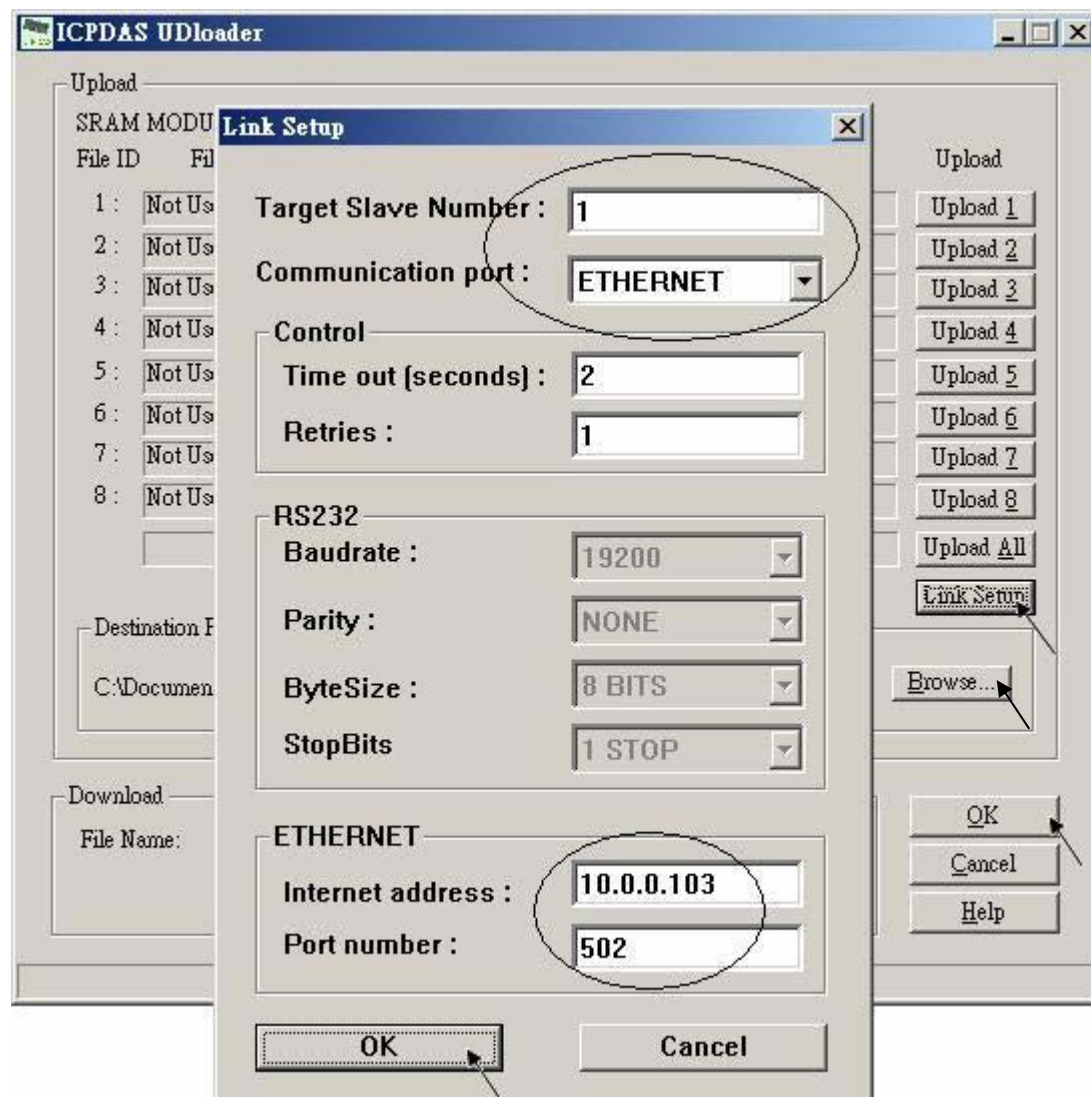
2. Download the ISaGRAF project - “Demo\_71” to the i-8437-80 .

3. At run time, you may press pushbutton 1 to start recording. Then it will record data during 2 minutes. You can see the displayed number on the front panel decreasing to 0 and blinking. If recording is finished, the 3 Leds on the front panel will blink and the displayed number will be 0. To stop at anytime, just press pushbutton 4 once.

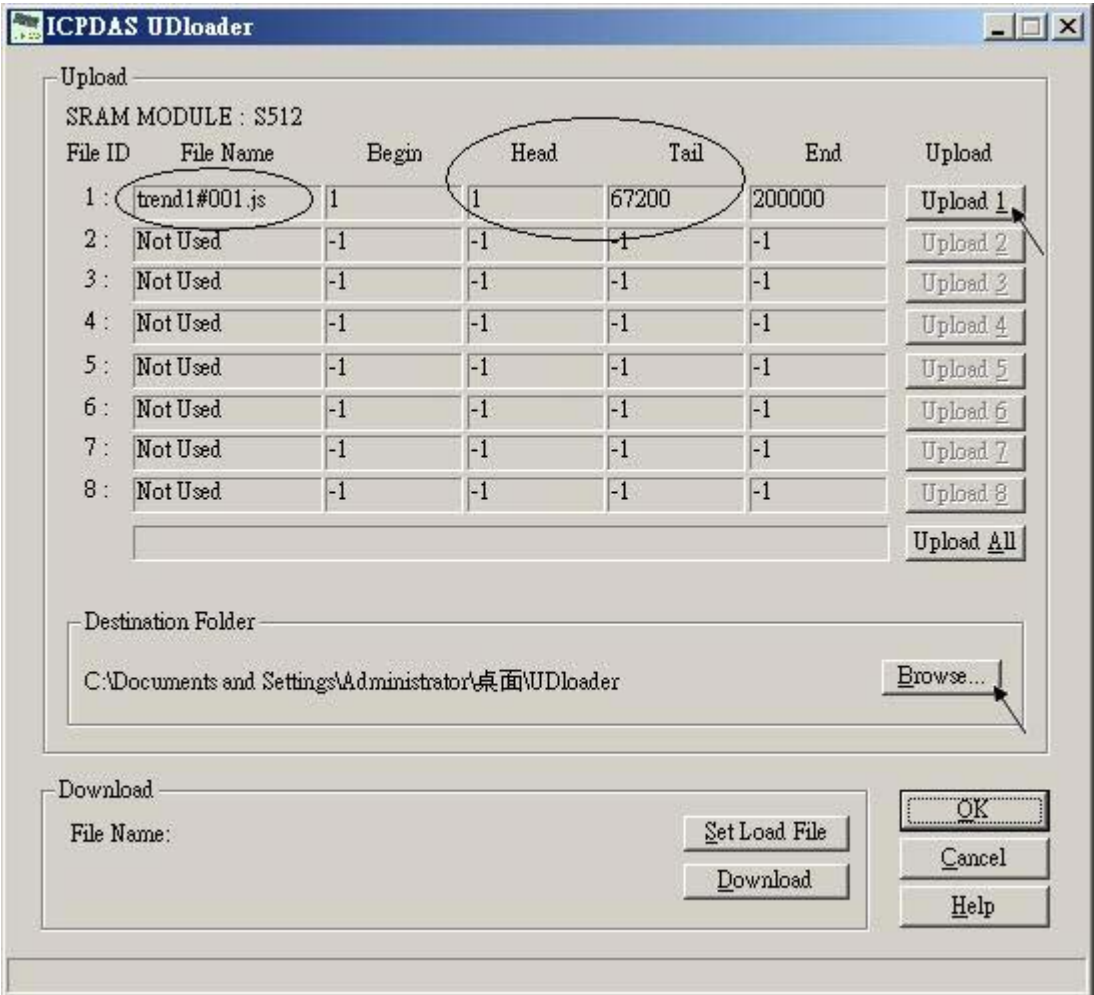
4. When recording is finished, please run UDLloader in your PC to upload the record file in the S-256 / 512 to PC. If your PC is currently running ISaGRAF workbench, please run “Tools” – “ICP DAS” – “ICPDAS UDLloader” . Then you will see the windows listed in step (5) .



If your PC is not running ISaGRAF workbench, please copy i-8000-CD:\napdos\isagraf\some\_utility\“udloader.exe” to for example PC windows ‘s desktop. Then please run it. Set proper “Link Setup” ( If click on “Browse” , you may modify the file upload location path). Then click on “Ok” and “Ok” to save this setting, then run it again.

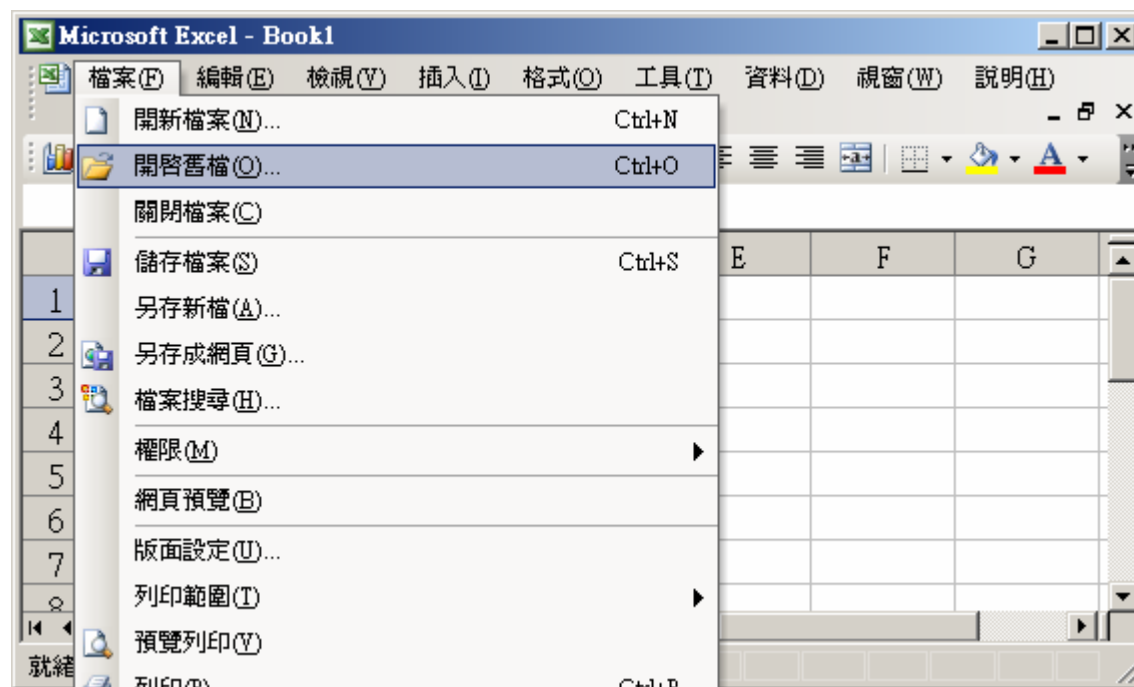


5. If the controller is well connected, you will see a File Name displayed on the below window. Value of “Head” and “Tail” is the current size of the record file in the S-256 / S-512. The below example shows 67,200 bytes. (Note: S-256 / S-512 has size limitation, please refer to Chapter 10.3). you may modify the file upload location path by click on “Browse ...” . Then please click on “Upload 1” to upload this record file to your PC.

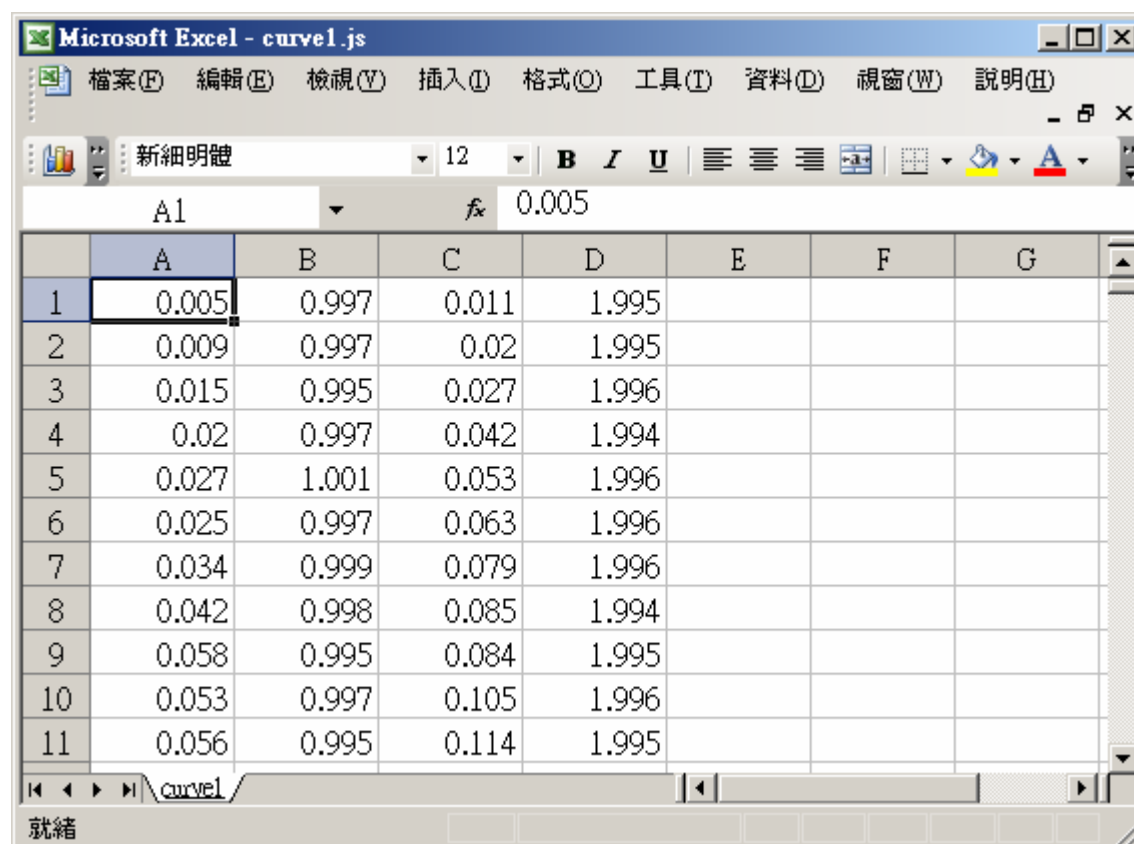


Then you may check if the record file is upload to your PC at the same path.

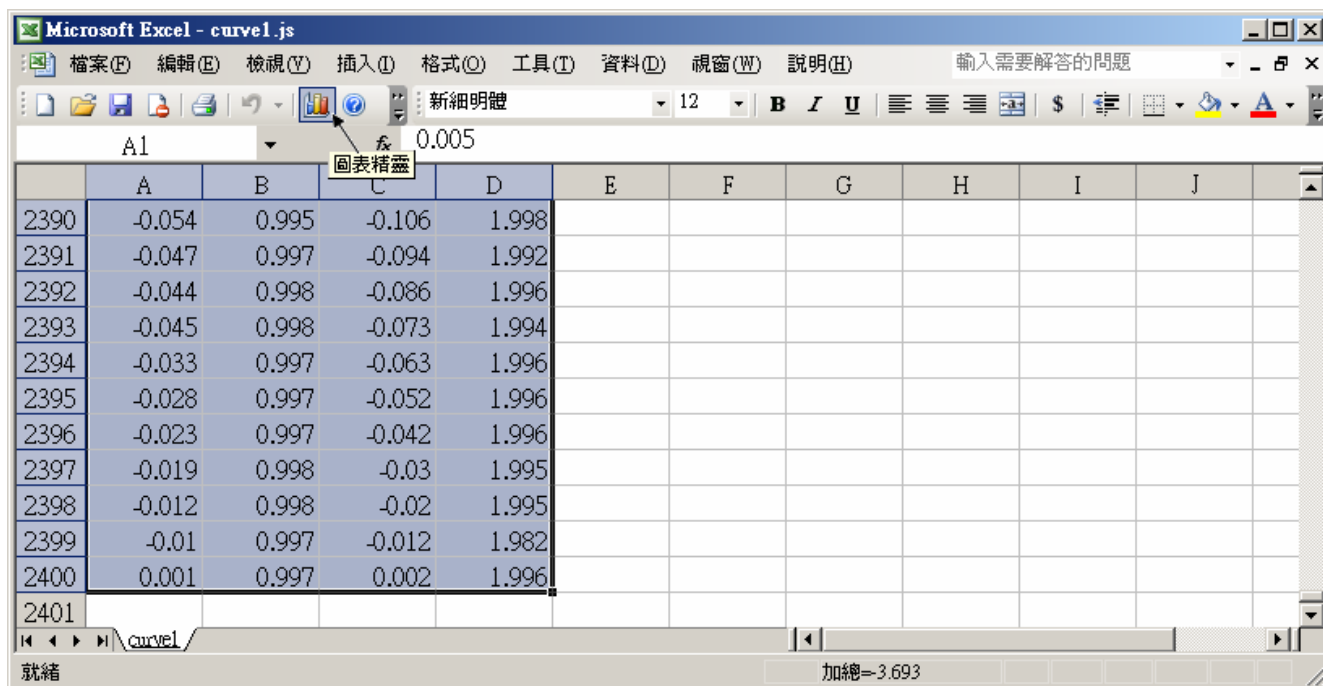
6. Then please open this record file - “trend1.js” on M.S. Excel.



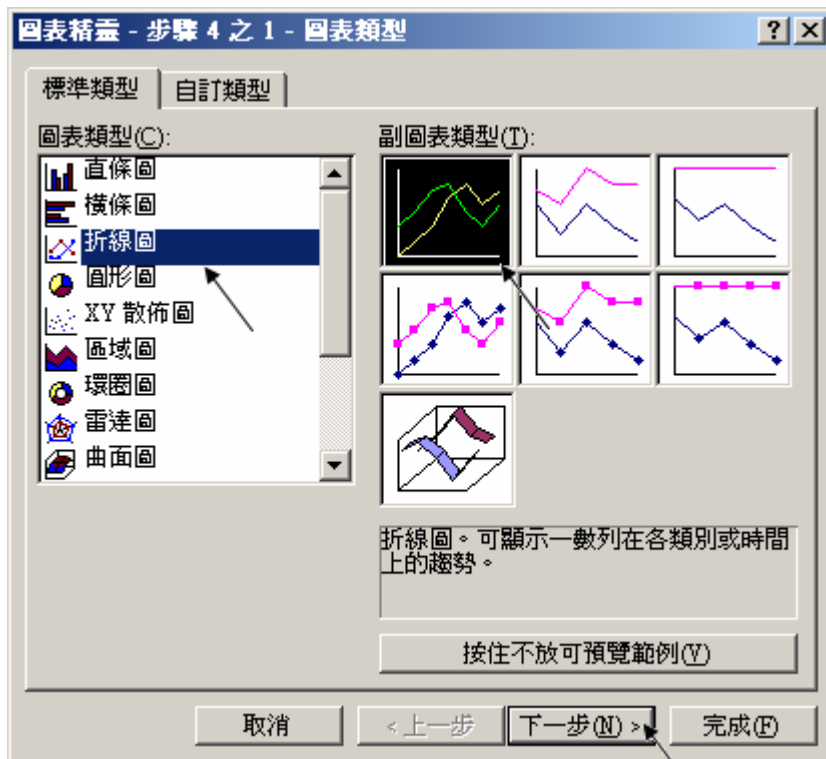
Please click on the first data at the left-top position. Then press and hold in “Shift”, and at the same time press “Ctrl” – “End”. You will see all data been selected.

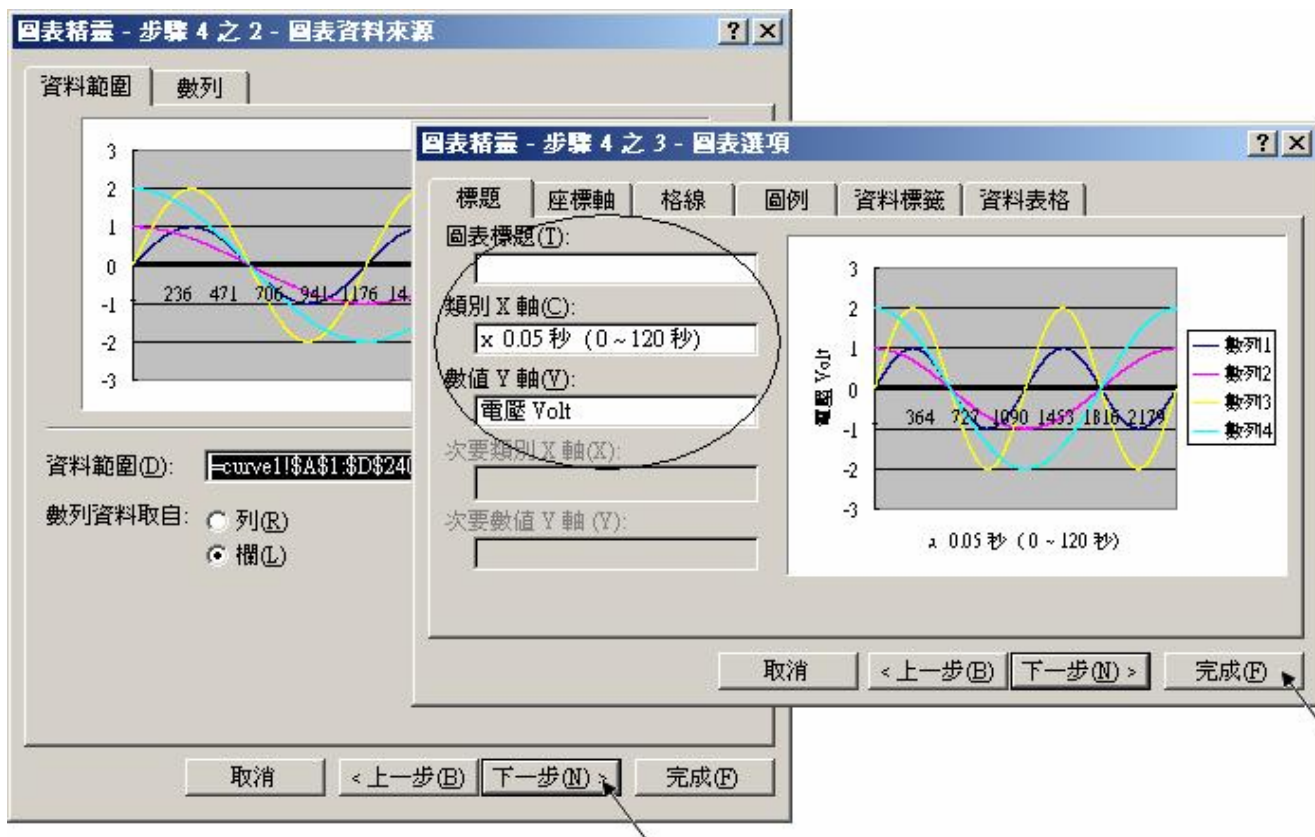


Then click on 

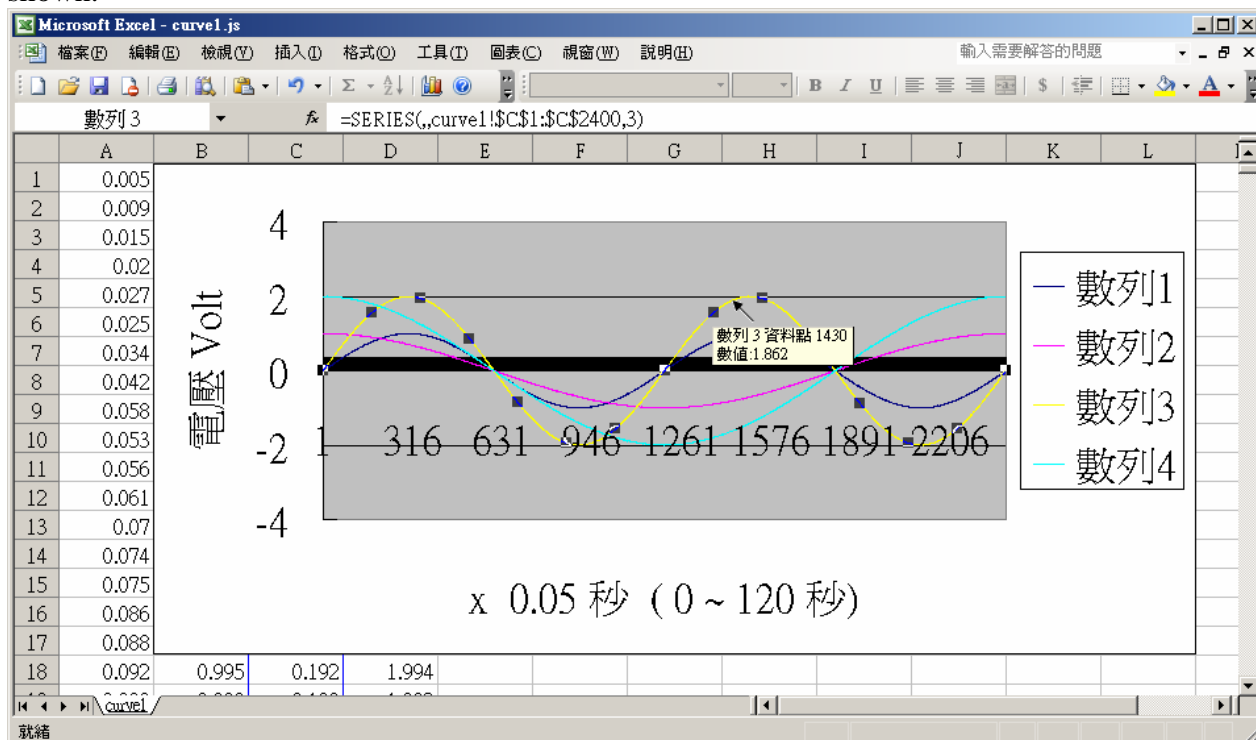


Please select the correct diagram on the left-hand side. And check the left-top type on the right-hand side. Then go Next .

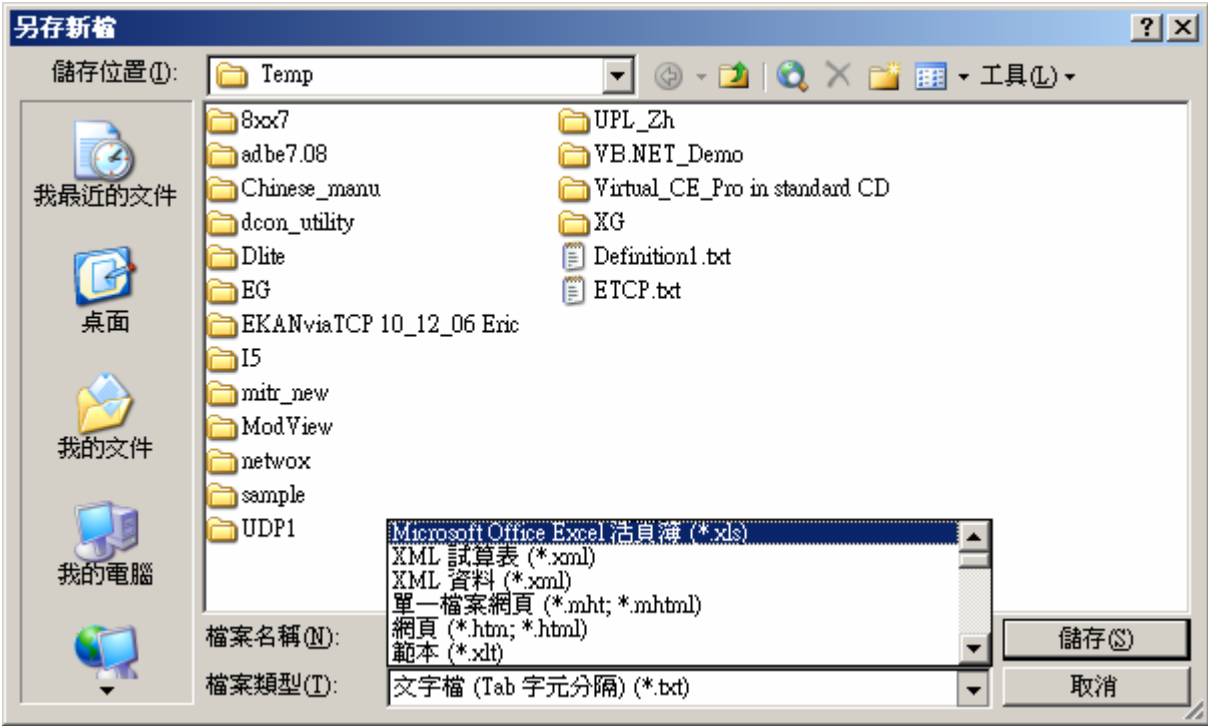
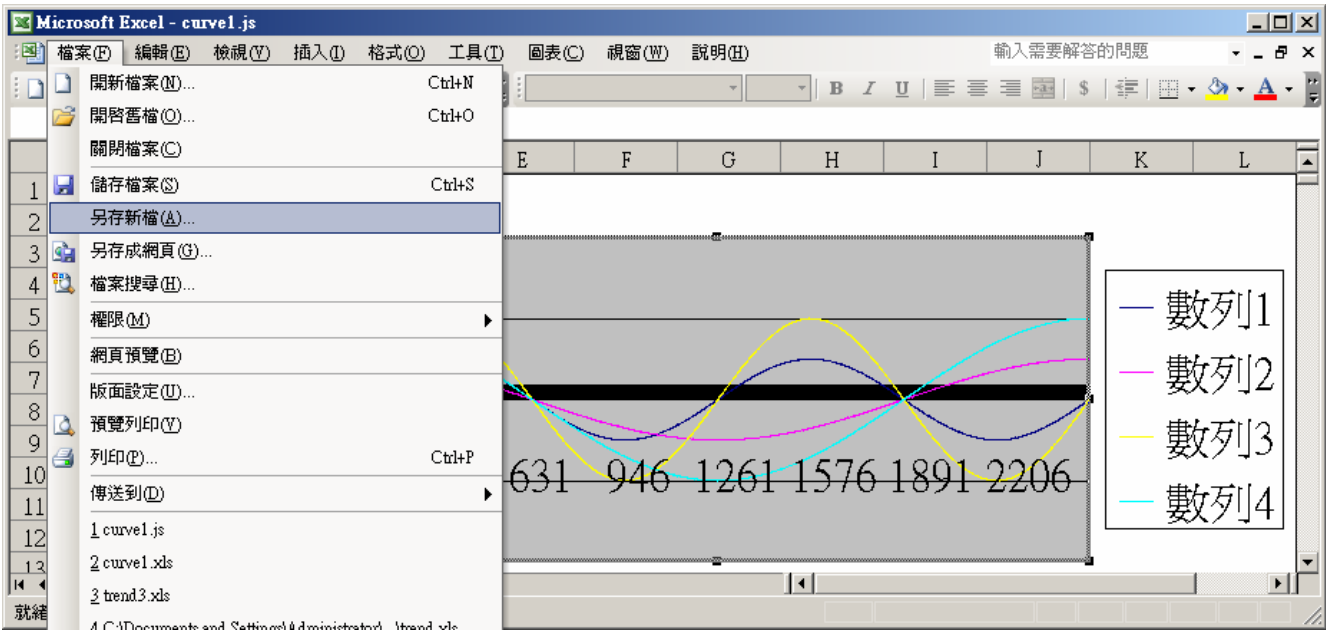




By the procedure, you will get the trend curve as the below window. You can modify its size, or check at any trend line. If you move your mouse to point at some position at the trend line, the related data is shown.

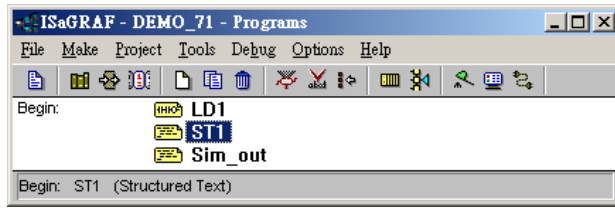


Please save this trend curve diagram as a “Microsoft Office Excel (\*.xls)” format. Then at any later time, you can open it to display the trend curve directly.





ISaGRAF project architecture:

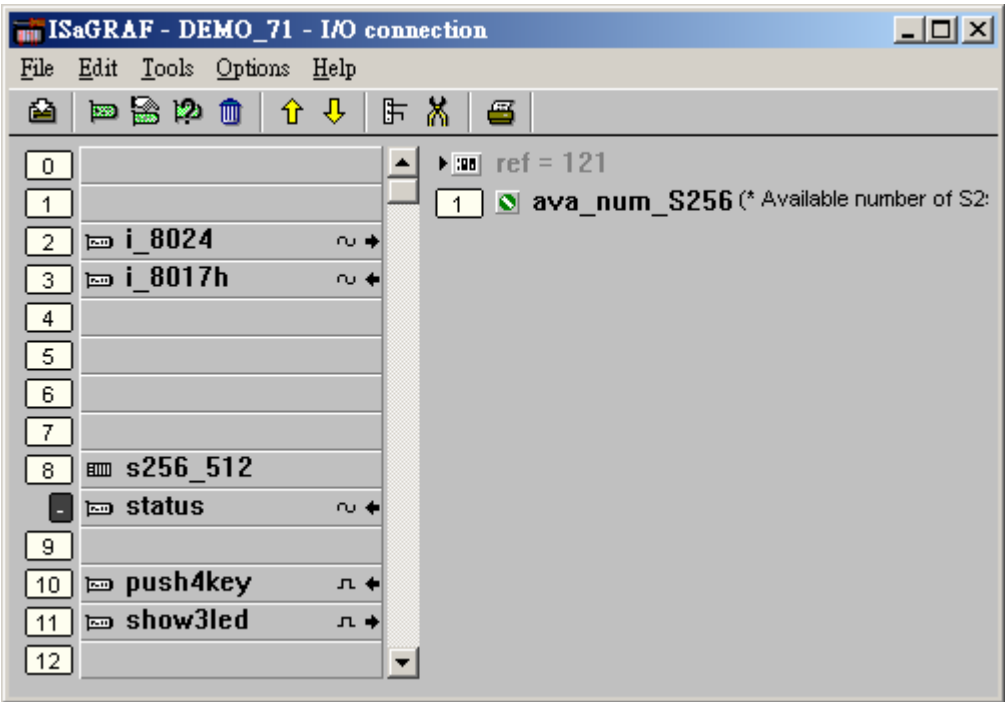


Variable : Please refer to Chapter 2.6 or FAQ-039 for description of “Variable array”

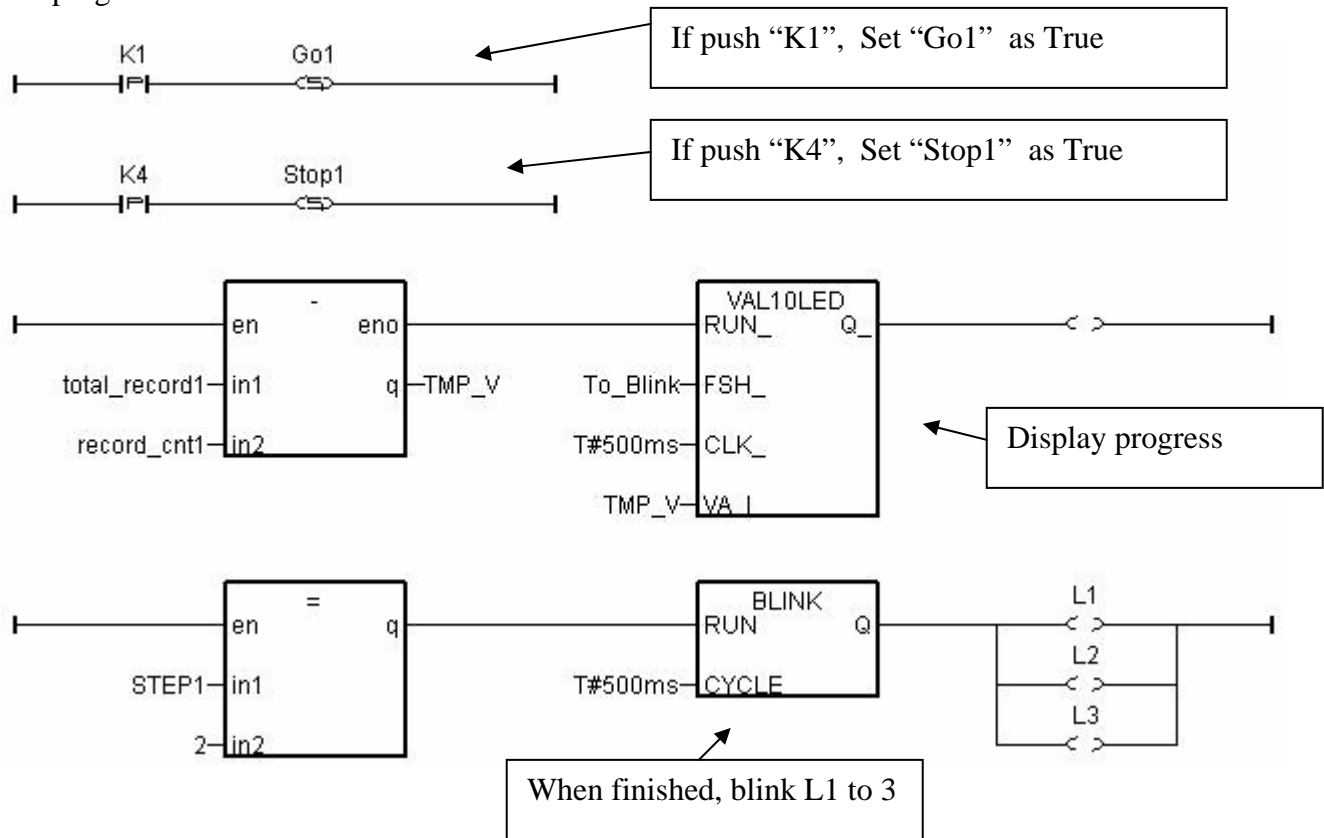
| Name                     | Type    | Attribute       | Description  |
|--------------------------|---------|-----------------|--|
| Go1                      | Boolean | <b>Internal</b> | Set as True to start, <b>addr defined as 21</b> (Hex. is 15)   |
| Stop1                    | Boolean | <b>Internal</b> | Set as True to stop, <b>addr defined as 22</b> (Hex. is 16)  |
| TMP                      | Boolean | <b>Internal</b> | Internal use   |
| INIT                     | Boolean | <b>Internal</b> | Init as True   |
| L1 , L2 , L3             | Boolean | <b>Output</b>   | connect to show3Led 's Ch.1 to Ch.3  |
| K1 , K4                  | Boolean | <b>Input</b>    | Connect to push4key 's Ch.1 to Ch.4<br>Push K1 to start recording. K4 to stop  |
| To_Blink                 | Boolean | <b>Internal</b> | To control blinking of the number on the front pannel  |
| MUM_CH                   | Integer | <b>Constant</b> | How many chanel in i-8017H to record ?<br>We use 4 channels in this demo (Ch.1 to 4)   |
| Ava_num_s256             | Integer | <b>Input</b>    | Connect to “S256_512” 's Ch.1 . if value is 0, it means can not find S-256 / 512   |
| Current_pos1             | Integer | <b>Internal</b> | Current operating byte address in S-256 / 512  |
| Len1 , TMP_V             | Integer | <b>Internal</b> | Internal use   |
| File_begin1<br>File_end1 | Integer | <b>Constant</b> | The Begin & End byte address in the S-256 / S-512 allocated for the record file  |
| STEP1                    | Integer | <b>Internal</b> | Recording state. 0:No action , 1:recording , 2:finished  |
| Period1                  | Integer | <b>Internal</b> | How long to record ? unit is minute, <b>addr as 3</b>  |
| Interval1                | Integer | <b>Internal</b> | How long to save a record ? unit is ms, <b>addr as 1</b>   |
| Total_record1            | Integer | <b>Internal</b> | How many records in this recording action ? This value is calculated by the IsaGRAF program automatically. <b>addr declared as 5</b> |
| Record_cnt1              | Integer | <b>Internal</b> | Current finished record count. <b>addr declared as 7</b>   |
| ii                       | Integer | <b>Internal</b> | To use in “for” loops  |
| i8017H[0..7]             | Integer | <b>Input</b>    | Variable array, Dim as 8. link to i-8017H 's Ch1 to Ch. 8  |
| Volt1[0..7]              | REAL    | <b>Internal</b> | Variable array, Dim declared as 8.<br>The voltage value converted from “i8017H[0..7]”  |
| i8024[0..3]              | Integer | <b>Output</b>   | Variable array, Dim declared as 4. link to i-8024 's Ch1 to Ch. 4  |
| T1                       | Timer   | <b>Internal</b> | For counting time  |
| T1_next                  | Timer   | <b>Internal</b> | The time to get and save next record   |
| T1_Interval              | Timer   | <b>Internal</b> | The interval time between two record   |
| Msg1                     | Message | <b>Internal</b> | Operation state message, Len is 255, init as “No Action now”, <b>addr as 41</b> (Hex. is 29)   |
| Str1                     | Message | <b>Internal</b> | Len is 255, internal use   |



IO connection:



LD program – LD1



## ST program – Sim\_out

---

```
(* Output i-8024 's Ch1 to Ch4 as different voltage curve *)
(* 2 * Pi * T1 / 60000 = T1 * 1.047197E-4 *)
(* 2 * Pi * T1 / 120000 = T1 * 5.235985E-5 *)
i8024[0] := ANA( sin( REAL(T1) * 1.047197E-4 ) * 3276.8 ) ;
i8024[1] := ANA( cos( REAL(T1) * 5.235985E-5 ) * 3276.8 ) ;
i8024[2] := ANA( sin( REAL(T1) * 1.047197E-4 ) * 6553.6 ) ;
i8024[3] := ANA( cos( REAL(T1) * 5.235985E-5 ) * 6553.6 ) ;
```

---

## ST program – ST1

---

**if INIT then**

**INIT := FALSE ;** (\* set as False to only do it once at 1st PLC scan \*)

**if ava\_num\_S256 = 0 then** (\* S256 / S512 is not installed in i-8xx7, return \*)

**Msg1 := 'S256 / S512 is not installed in i-8xx7 controller !' ;**

**Return ;**

**end\_if ;**

(\* Allocate S256/512 memory of byte No.1 to 200,000 for file ID = 1 , name='trend1.js' \*)

**TMP := S\_FL\_INI( 1 , 'trend1.js' , File\_begin1 , File\_end1 ) ;**

**TMP := S\_FL\_AVL( 1 , -1 , -1 ) ;** (\* Init file content as No data at the beginning \*)

**end\_if ;**

**if ava\_num\_S256 = 0 then** (\* S256 / S512 is not installed in i-8xx7, return \*)

**return ;**

**end\_if ;**

(\* If stop command is gived \*)

**if Stop1 then**

**Stop1 := False ;**

**STEP1 := 0 ;** (\* 0: no action \*)

**TStop(T1) ;** (\* stop T1 \*)

**T1 := T#0s ;**

**Msg1 := 'User stop recording !' ;**

**To\_Blink := FALSE ;** (\* Set as FALSE not to blink the display value \*)

**end\_if ;**

(\* Get file status in S256 or S512 : -1: PC hasn't load the file yet \*)

(\* others: the end byte No. that PC has load the file \*)

**TMP\_V := S\_FL\_STS(1) ;**

**if TMP\_V <> -1 then** (\* PC has load the file \*)

**TMP := S\_FL\_RST(1) ;** (\* reset status to -1 ( PC hasn't load the file yet) \*)

**end\_if ;**

```

(* If start command is gived *)
if Go1 then
  Go1 := False ;

  (* STEP1 : 0: no action , 1: recording , 2: recond finished *)
  if STEP1=1 then
    Msg1 := 'It is still recording now ...' ;
  else

    (* Check interval valid or not *)
    (* we assume 25 to 10000 ms is valid in this example *)
    (* If your average PLC scan time is larger, for example, near 20 ms,
       Please use Interval larger than 25 ms. Or the record time won't be correct *)
    if (Interval1 < 25) or (Interval1 > 10000) then
      Msg1 := 'Wrong Interval value, it should be in 25 to 10000 milli-second !' ;

    (* Check period valid or not *)
    (* we assume 1 to 10 minute is valid in this example *)
    Elsif ( Period1 < 1) or (Period1 > 10) then
      Msg1 := 'Wrong Period value, it should be in 1 to 10 minute !' ;

    else

      (* parameter is correct, start recording *)
      total_record1 := (Period1 * 60000) / Interval1 ; (* calculate total record number *)
      record_cnt1 := 0 ; (* reset current record count as 0 *)
      STEP1 := 1 ; (* set step as 1:recording *)
      Msg1 := 'Recording now ...' ;

      (* start ticking T1 from 0 second *)
      T1 := T#0s ;
      T1_Interval := TMR(Interval1) ;
      T1_next := T1 + T1_Interval ;
      TStart(T1) ; (* ticking now *)
      Current_pos1 := 1 ; (* reset current data position in S256/S512 as 1 *)
      To_Blink := TRUE ; (* Set as TRUE to blink the display value *)

    end_if ;
  end_if ;
end_if ;

  (* in reconrding state *)
  if STEP1=1 then
    (* store one record *)
    if T1 >= T1_next then

      (* Re-calculate next T1 *)
      T1_next := T1_next + T1_Interval ;
    end_if ;
  end_if ;

```

```

(* T1 will be overflow after T#23h59m59s999ms, so reset it at T#20h *)
if T1 >= T#20h then
    T1 := T#0s ;
    T1_next := T1 + T1_Interval ;
end_if ;

str1 := '' ; (* init str1 as empty string *)
for ii := 0 to NUM_CH-1 do

    (* convert i8017H analog input value to Volt value *)
    Volt1[ii] := Real(i8017H[ii]) * 0.000305176 ; (* 10.0 / 32768 = 0.000305176 *)
    str1 := str1 + Rea_Str2(Volt1[ii], 3) + '$09' ; (* delimiter is <TAB> character *)

end_for ;

str1 := str1 + '$0D$0A' ; (* add <CR> <LF> at the end of each row *)
Len1 := MLEN(str1) ; (* get string length *)

(* data number larger than file's max. allocated memory *)
if (Current_pos1 + Len1 - 1) > File_end1 then
    STEP1 := 0 ; (* 0: no action *)
    Msg1 := 'File allocated memory is not enough to hold the data !' ;
    Tstop(T1) ;
    T1 := T#0s ;
    To_Blink := FALSE ; (* Set as FALSE not to blink the display value *)
    Return ;
end_if ;

TMP := S_M_W( Current_pos1 , Len1 , str1 ) ; (* write all bytes in str1 to S256/S512 *)
Current_pos1 := Current_pos1 + Len1 ; (* Current position move on *)
TMP := S_FL_AVL( 1 , 1 , Current_pos1-1 ) ; (* Re-caculate File 's Head & Tail *)

(* Check if record number reach the end *)
record_cnt1 := record_cnt1 + 1 ; (* current record count plus 1 *)
if (record_cnt1 >= total_record1) then
    STEP1 := 2 ; (* 2: record finished *)
    Msg1 := 'Record is finished ! You may download the record file to your PC now !' ;
    Tstop(T1) ;
    T1 := T#0s ;
    To_Blink := FALSE ; (* Set as FALSE not to blink the display value *)
end_if ;

end_if ;

end_if ;

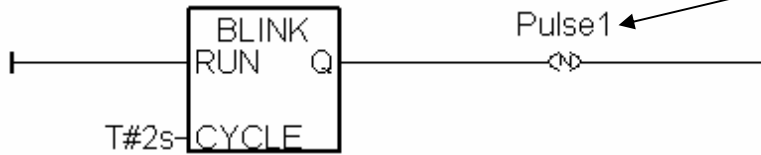
```

---

### 11.3.8: How to do periodic operation in ISaGRAF controllers ?

The “BLINK” function block can apply to generate a Pulse True periodically. So it can apply in some periodic operations like as below.

(\* LD program \*)



“Pulse1” is Boolean Internal  
Periodic operation every 2 sec

(\* ST program \*)

**IF Pulse1 THEN** (\* above LD program will generate a pulse TRUE in “pulse1” variable \*)

    (\* do operation \*)

    (\* ..... \*)

**END\_IF ;**

The above program has a disadvantage. When the periodic interval time is short, for example – 200ms or smaller, or the controller ‘s PLC scan time is bigger , the operation time will not be precise. For example to do a periodic operation every 50 milli-second. Because 50ms is a shorter interval , it is much closer to the PLC scan time compared with interval time of 250 ms or 2 seconds, the result time will not be precise. To improve this, following codes can be applied.

ST program:

```
IF INIT THEN
    INIT := False ;
    T1 := T#0s ;
    T1_next := T1 + T#50ms ;
    Tstart (T1) ;
END_IF ;
```

“INIT” is declared as Boolean Internal  
And init as TRUE  
“T1” and “T1\_next” are Timer Internal

```
IF T1 >= T1_next THEN
```

```
    IF T1 > T#22h THEN
        T1 := T#0s ;
        T1_next := T#0s ;
    END_IF ;
```

Timer will be overflow if it is ticking to  
T#23h59m59s999ms. So we can reset it to 0  
second when it just reach the “22h” or “16h”  
whatever a bigger time you like.

```
    T1_next := T1_next + T#50ms ; (* calculate next operation time *)
```

    (\* do operation \*)

    (\* ..... \*)

**END\_IF ;**

### 11.3.9: Connecting i-7018z to get 6 channels of 4 to 20 mA input and 4 channels of Thermo-couple temperature input. And then also display the value on PC by VB 6.0 program .

The ISaGRAF demo project name is “Demo\_72” . It can run in the i-7188EG / 7186EG. If user want to run in i-8xx7 or Wincon-8xx7, please set the “com\_port” parameter of ”Bus7000b” in the IO connection window to COM3 and then re-compile the project.

“demo\_72.pia” resides at I-8000 CD-ROM:\napdos\isagraf\8000\demo\ or  
<ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/8000/demo/> or  
[www.icpdas.com](http://www.icpdas.com) – FAQ – Software – ISaGRAF – FAQ055

VB 6.0 project - “Demo\_4” resides at  
i-8000 CD: \napdos\isagraf\vb\_demo\demo\_4\ or  
<http://www.icpdas.com/faq/isagraf.htm> FAQ-055

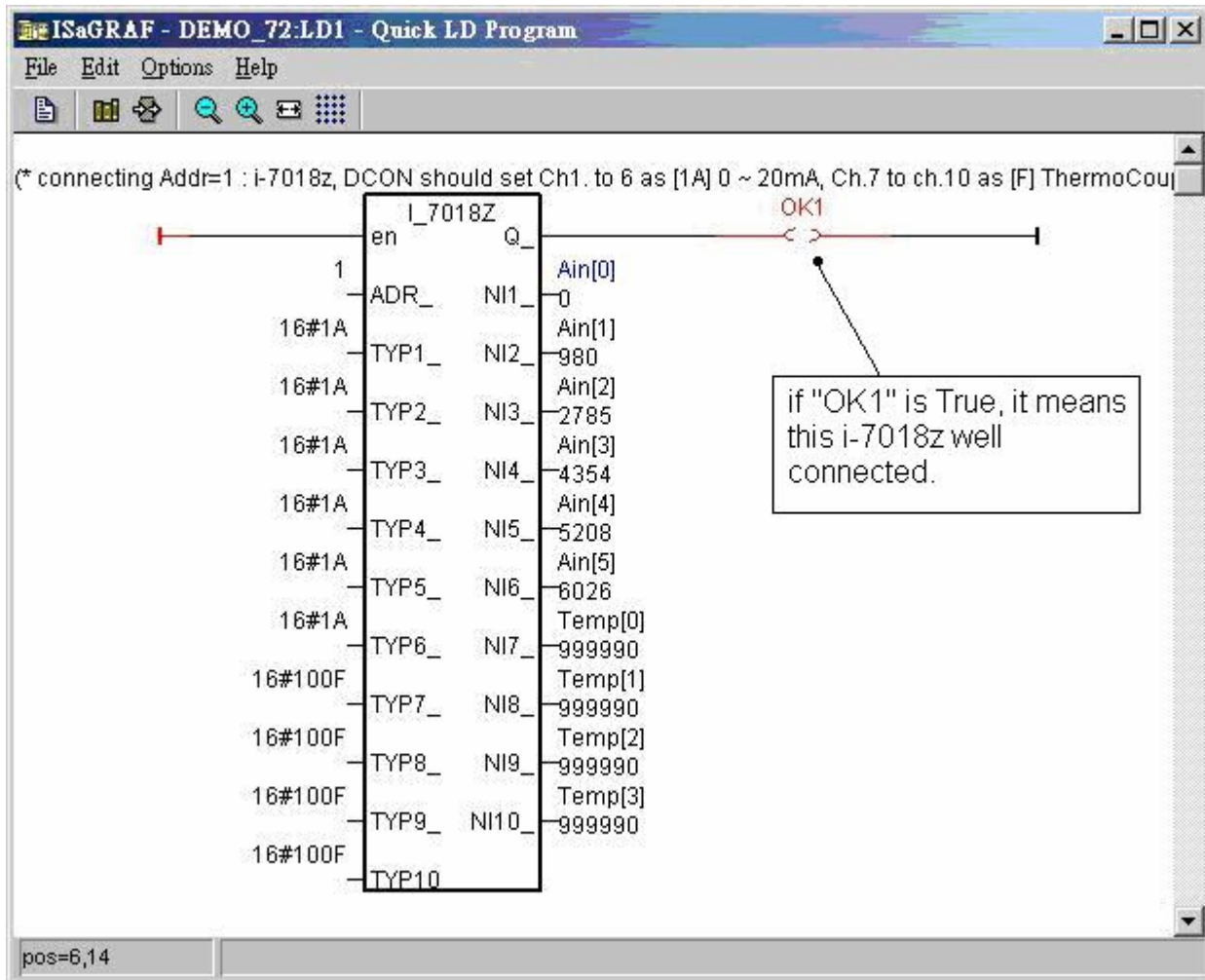
i-7188EG ‘s COM2:RS-485 can connect i-7000 or i-87K/4/5/8/9 expansion base plus i-87xxx I/O boards in it. One i-7188EG can connect max. 64 pcs. of i-7000 modules (or i-87xxx I/O boards, the total amount of “i-7000 + i-87xxx” is up to 64 pcs.). To use i-8xx7’s COM3:RS-485 to connect i-7000 + i-87xxx is the same as I-7188EG, the total amount is also 64 pcs. While max. 255 pcs. for using W-8xx7 ‘s COM3:RS-485 to connect i-7000 + i-87xxx .

The more RS-485 I/O modules connected, the more I/O scan time will be . For example, if setting baud-rate as 9600 bps (Bit Per Second), one RS-485 D/I & D/O module will consume about 20 to 40 milli-second to scan its I/O channels. If connecting RS-485 A/I & A/O module, one will consume about 40 to 60 ms (The I/O scan time of the remote RS-485 I/O module depends on the module’s type and function. If there are more than one I/O type in the module, the time consumed will be longer than the above value. For example, the i-7050D is a 7-Ch digital Input plus 8-ch digital output module, it will consume more than 20 to 40 ms). If connecting 20 pcs. of D/I/O modules, the approximate I/O scan time of all channels in these I/O modules will be about 0.4 to 0.8 second. If connecting 20 pcs. of A/I/O modules, the I/O scan time is about 0.8 to 1.2 second. To have better (shorter) remote I/O scan time, here recommend not to connect more than 24 pcs. of I/O modules in the i-7188EG/XG and i-8xx7, while 64 pcs. in the Wincon-8xx7.

#### How to test this demo ?

1. To configure i-7018z and i-87018z, please install **DCON utility (Version should be 4.4.3 or later version)** in your PC. The new released DCON utility can be found in the i-8000 CD-ROM or at [ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon\\_utility/](ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon_utility/) “setup” folder .
2. Please do initial configuration in i-7018z, (please refer to step (1) to (4) in chapter 6.1). Set i-7018z ‘s Address as 1, baud rate as 9600, Format as “2’s compliment” , Checksum disable. And also set Ch.1 to Ch.6 type as “[1A] : 0 ~ 20 mA” , while Ch.7 to Ch.10 type as “[0F] : T/C K-Type” . If initial setting is finished, please switch the “Dip Switch” on the back of i-7018z to “Normal” and recycle its power.
3. Please set the i-7188EG ‘s IP as 192.168.1.3 (refer to Appendix B), NET-ID as 1. Then power OFF the i-7188EG, connecting its COM2 to the i-7018z. Then power up i-7188EG and i-7018z. (To connect this i-7188EG well in the local network, PC ‘s IP should be in the same domain as 192.168.1.x. For example, setting PC ‘s IP as 192.168.1.2 , Mask=255.255.255.0)

4. PC run ISaGRAF to download “demo\_72” project to the i-7188EG via ethernet. Then open the Ladder program window in the ISaGRAF to check if i-7018z is well connected.



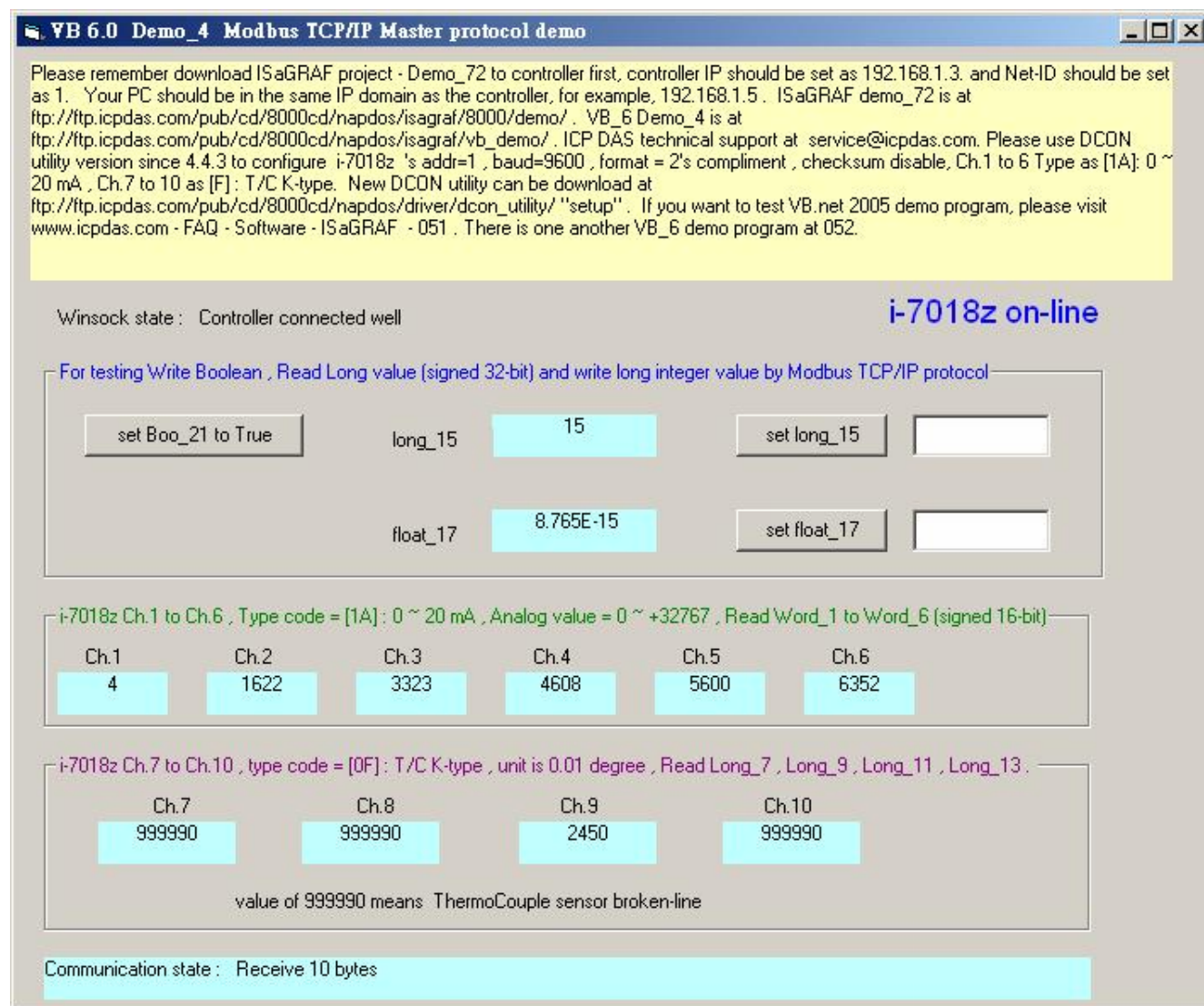
5. Then please run VB 6.0 – “Demo\_4.exe” in your PC. It resides at  
i-8000 CD: \napdos\isagraf\vb\_demo\demo\_4\demo\_4.exe or  
<http://www.icpdas.com/faq/isagraf.htm> FAQ-055

(As the figure in the nex page)

There is one another VB.net 2005 demo project can be study. Please visit  
<http://www.icpdas.com/faq/isagraf.htm> FAQ-051 or  
[www.icpdas.com](http://www.icpdas.com) – FAQ – Software – ISaGRAF – 051

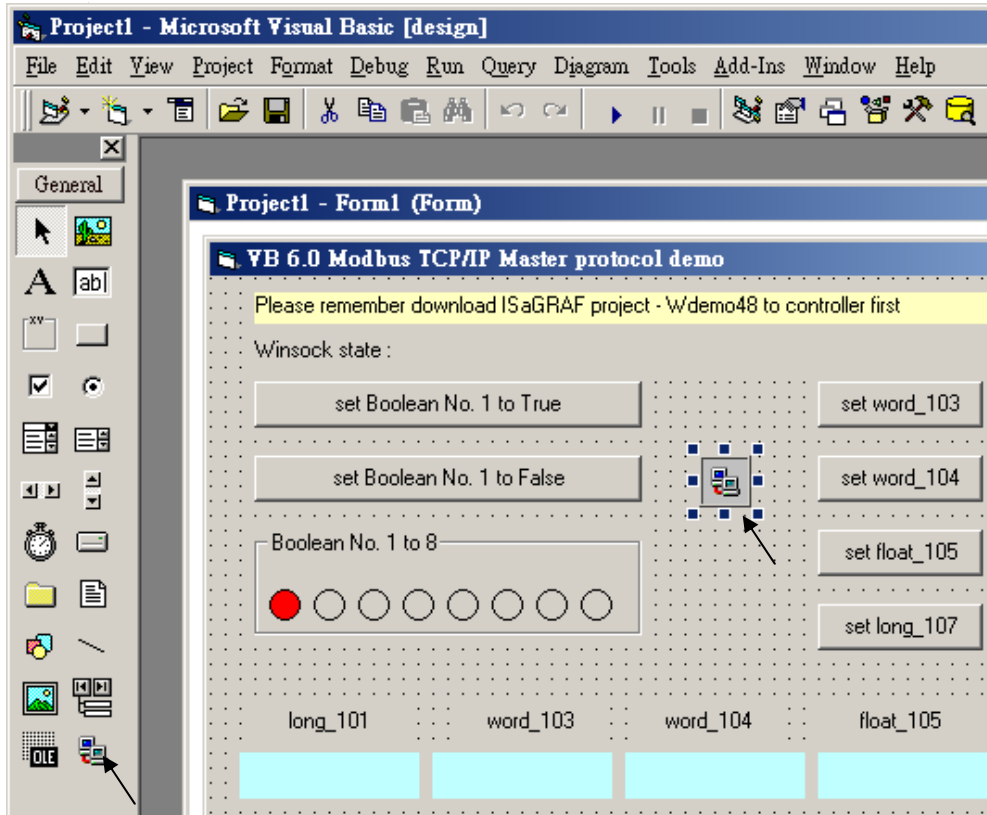
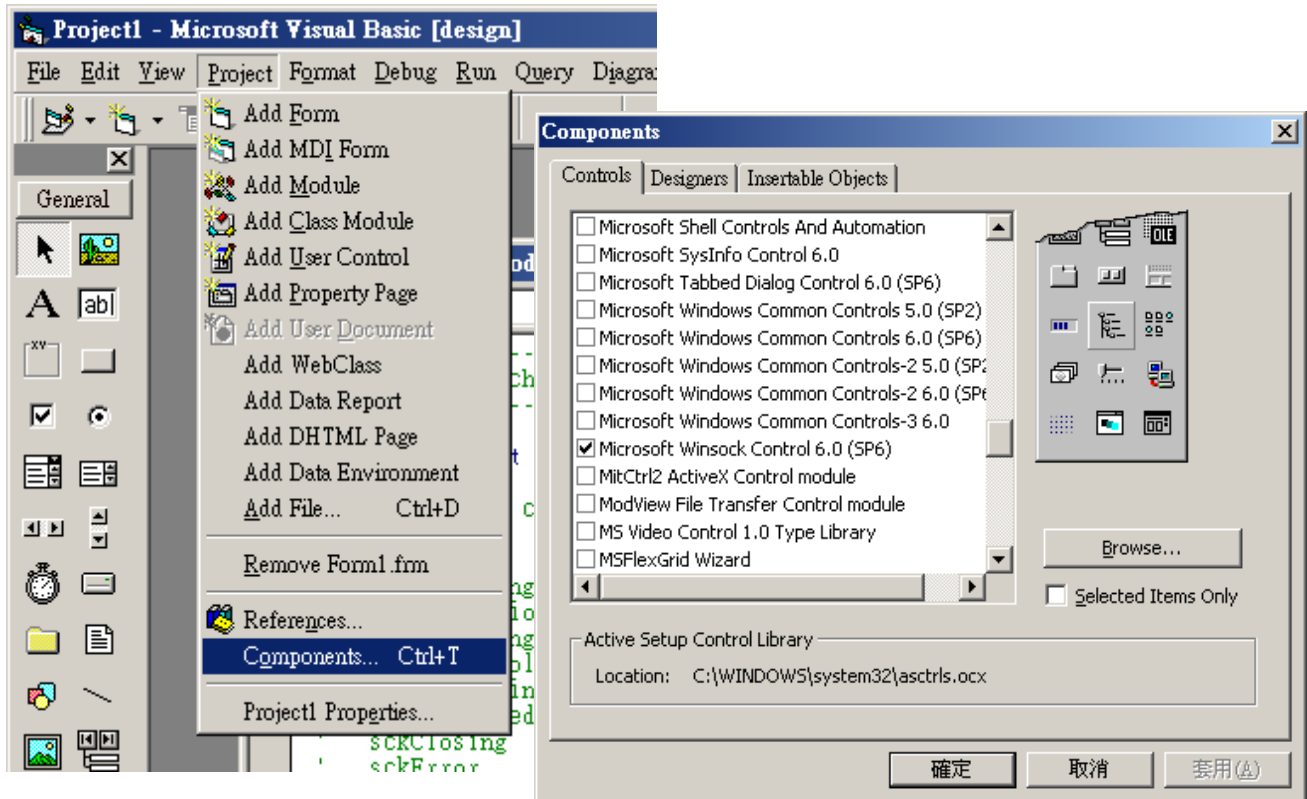
If PC can not link the i-7188EG well, the “Communication state” at the bottom will display the related error message. If the i-7188EG can not connect i-7018z well, there will be a “i-7018z not on-line” message displayed in red color.

You may click on “set Boo\_21 to True” button. One click will increase the “long\_15” value by 1. You may also enter a value to “set long\_15” column, then click on “set long\_15”

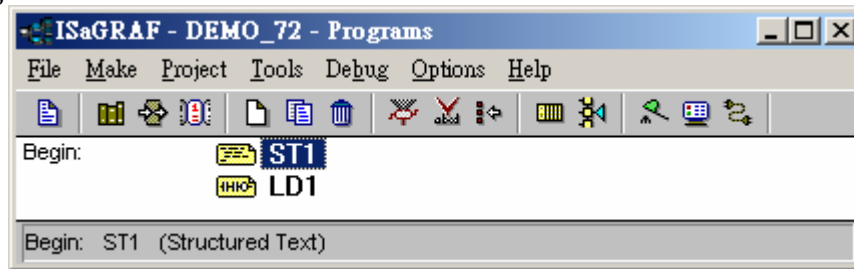




At designing time of the VB 6.0 program, please add "Winsock control" to your VB 6.0 project as below. Then ethernet operation will be possible in the project.



ISaGRAF Project architecture:



We use “Variable array” in this demo project. Please refer to Chapter 2.6 or FAQ-039 for more information about “Variable array” .

Variables :

| Name       | Type    | Attribute       | Description  |
|------------|---------|-----------------|--|
| INIT       | Boolean | <b>Internal</b> | Set initial value as True  |
| OK1        | Boolean | <b>Internal</b> | Communication state of i-7018z, <b>addr as 31</b> (Hex. is 1F)   |
| M1         | Boolean | <b>Internal</b> | For testing by VB 6.0 , <b>addr as 21</b> (Hex. is 15)   |
| TMP        | Boolean | <b>Internal</b> | Internal using   |
| Ain[0..5]  | Integer | <b>Internal</b> | Variable array, Dim as 6, <b>addr as 1</b><br>To get the input value of i-7018z ‘s Ch.1 to Ch.6        |
| Temp[0..3] | Integer | <b>Internal</b> | Variable array, Dim as 4, <b>addr as 7</b><br>To get the temperature input of i-7018z ‘s Ch.7 to Ch.10 |
| CNT1       | Integer | <b>Internal</b> | For testing by VB 6.0, <b>addr as 15</b> (Hex. is F)   |
| Float_17   | Integer | <b>REAL</b>     | For testing by VB 6.0, <b>addr as 17</b> (Hex. is 11)<br>Set initial value as 1.02345                  |

STprogram – ST1

-----  
**if INIT then**

**INIT := False ;**

(\* Configure Ain[0..5] ‘s network addr as 1, 2, 3, 4, 5, 6, the initial addr. 1 should be assigned when doing variable declaration in the ISaGRAF dictionary window \*)

**TMP := S\_MB\_ADR( 1 , 6 , 0 ) ;** (\* the 3rd parameter 0 means setting as continuous addr. \*)

(\*Configure Temp[0..3] ‘s network addr as 7, 9, 11, 13, the initial addr. 7 should be assigned when doing variable declaration in the ISaGRAF dictionary window \*)

**TMP := S\_MB\_ADR( 7 , 4 , 1 ) ;** (\*the 3rd parameter 1 means setting as jummping addr. \*)

**end\_if ;**

**if M1 then**

**M1 := False ;**

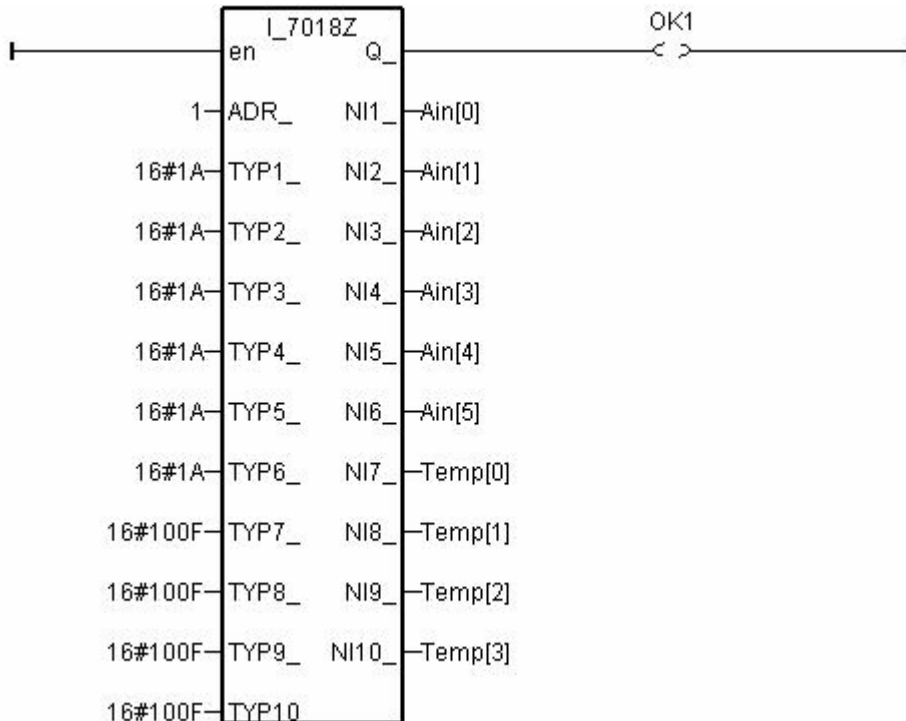
**CNT1 := CNT1 + 1 ;** (\* if M1 is set as TRUE by VB 6.0 program, increase CNT1 by 1 \*)

**end\_if ;**  
-----

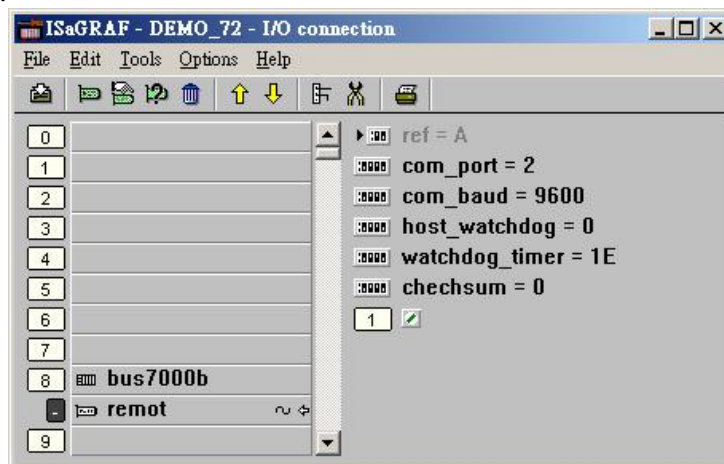
## LD program – LD1

The “TYP1\_” to “TYP6\_” parameter of the i-7018z block should be set as the same type code value in the DCON utility (Here we use [1A] 0 ~ 20 mA in this demo). And “TYP7\_” to “TYP10\_” set as 16#100F (This demo set [0F] T/C K-Type in the DCON utility) . Because we want to convert the temperature value to Celsius degree, so we use 16#100F here (unit is 0.01 degree). (If applying as Degree Fahrenheit, please set as 16#200F). If any converted value of the Temp[0] to Temp[3] returns 999990, it means the related channel’s temperature input sensor is break.

If the i-7018z is connected well, OK1 will be True.



IO connection:



**11.3.10: Recording i-8017H 's Ch.1 to Ch.4 voltage input in a user allocated RAM memory in the Wincon-8xx7 . The sampling time is one record every 0.01 second. The record period is 1 to 10 minutes. Then PC can download this record and display it as a trend curve diagram by M.S. Excel.**

The “Whmi\_13.pia” can run in Wincon-8xx7 / 8xx6 with driver version of 3.36 or later version.

New drive: <http://www.icpdas.com/products/PAC/i-8000/isagraf-link1.htm>  
“whim\_13.pia” resides at W-8xx7 CD-ROM:\napdos\isagraf\wincon\demo\ or  
[ftp://ftp.icpdas.com/pub/cd/wincon\\_isagraf/napdos/isagraf/wincon/demo/](ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/) or  
[www.icpdas.com](http://www.icpdas.com) – FAQ - Software – ISaGRAF – FAQ057

VB6 - “Demo\_5” code at

W-8xx7 CD-ROM:\napdos\isagraf\wincon\vb6\_demo\_pc\ or  
[ftp://ftp.icpdas.com/pub/cd/wincon\\_isagraf/napdos/isagraf/wincon/vb6\\_demo\\_pc/](ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/vb6_demo_pc/) or  
[www.icpdas.com](http://www.icpdas.com) – FAQ - Software – ISaGRAF – FAQ057

If using Web HMI in this demo, the Web HMI codes resides at below location. (Please refer to Chapter 3, 4 and 5 in the “Wincon ISaGRAF Getting Started manual” or

W-8xx7 CD-ROM:\napdos\isagraf\wincon\english\_manu\ “getting\_started\_w8337.pdf”)

W-8xx7 CD-ROM:\napdos\isagraf\wincon\WebHMI\_Demo\ or  
[ftp://ftp.icpdas.com/pub/cd/wincon\\_isagraf/napdos/isagraf/wincon/webhmi\\_demo/](ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/webhmi_demo/) or  
[www.icpdas.com](http://www.icpdas.com) – FAQ - Software – ISaGRAF – FAQ057

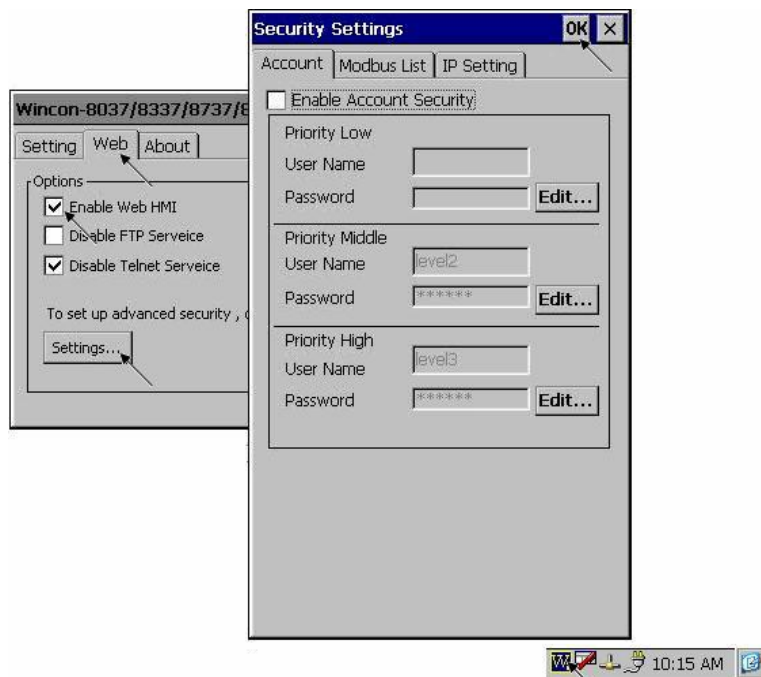
If new c-function of Msg\_F , Msg\_N , ARY\_F\_R and AFY\_F\_W doesn't exist in ISaGRAF in your PC, please visit <http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> to download the “ICP DAS utilities For ISaGRAF” . Then run “setup.exe” inside it to re-install all new ISaGRAF c-function & I/O boards definition to your ISaGRAF workbench.

We use “Variable array” in this demo program, please refer to Chapter 2.6 or FAQ-039 for more information about “Variable array” .

## How to test this demo ?

The following steps is only for using Web HMI as Human-Machine-Interface. If you are using VB 6.0 – “Demo\_5” as HMI , please run it in your PC and only do procedure listed in step 1 (not necessary to enable “Web HMI”) , step 3 and step 6 .

1. Please plug one i-8024 in W-8xx7 ’s Slot 2, one i-8017H in Slot 3. Then connect Ch1. to Ch.4 voltage output of i-8024 to Ch1. to Ch.4 of i-8017H. Then power up Wincon, Check “Enable Web HMI” option as below. For demo purpose, please don’t check “Enable Account Security”



2. Copy all files of Web HMI ‘s Demo\_13 to Wincon ‘s \CompactFlash\Temp\HTTP\WebHMI\ folder by ftp utility (For example, run <ftp://10.0.0.103> in Internet Explorer)

Web HMI codes resides at

W-8xx7 CD-ROM:\napdos\isagraf\wincon\WebHMI\_Demo\ “whim\_13” folder or  
[ftp://ftp.icpdas.com/pub/cd/wincon\\_isagraf/napdos/isagraf/wincon/webhmi\\_demo/](ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/webhmi_demo/)

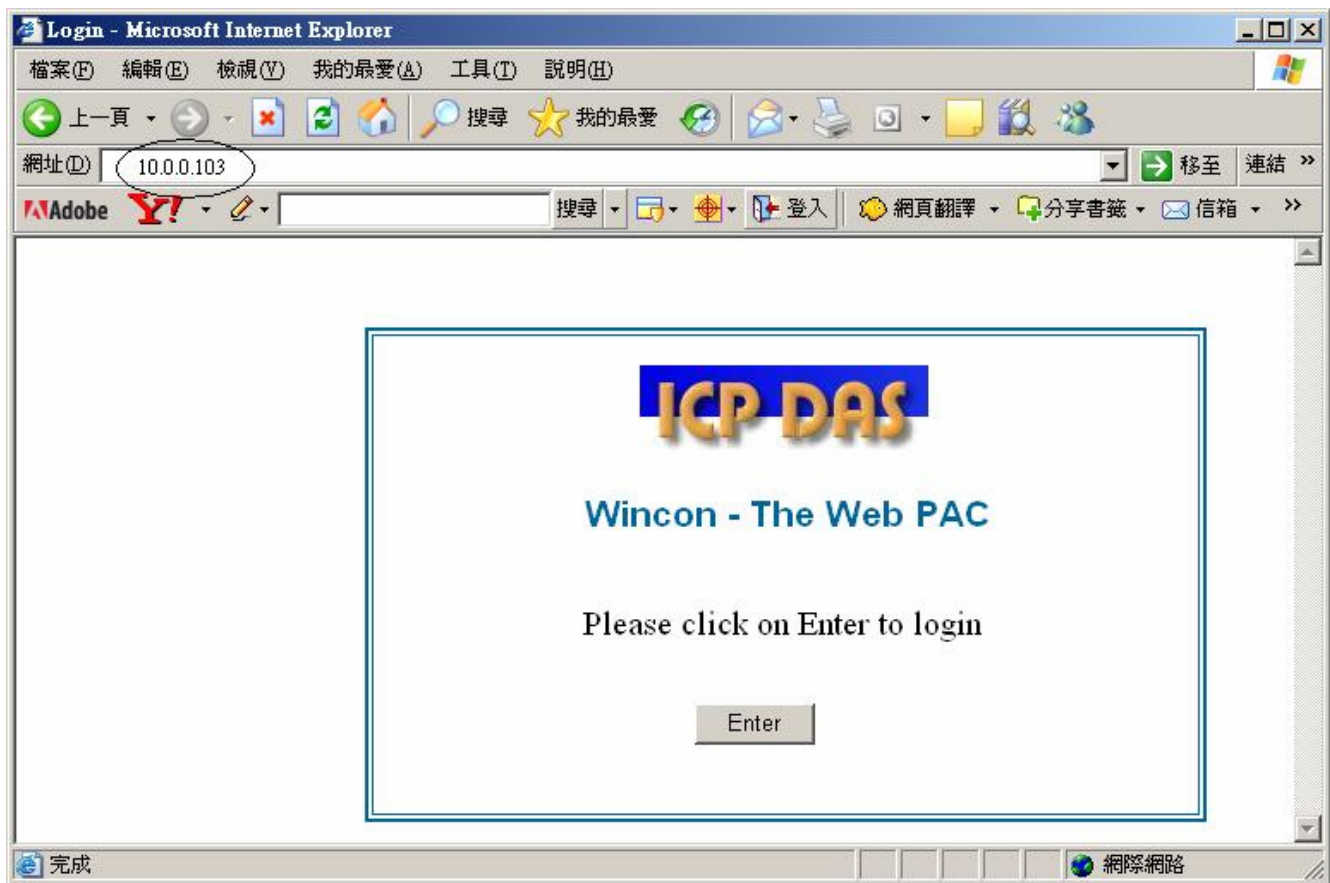
There are 7 Files plus 2 folder should be copied to Wincon ‘s \CompactFlash\Temp\HTTP\WebHMI\

Main.htm , menu.htm , index.htm , login.htm , main.dll , login.dll , whim\_filter.dll  
“img” & “msg” folder

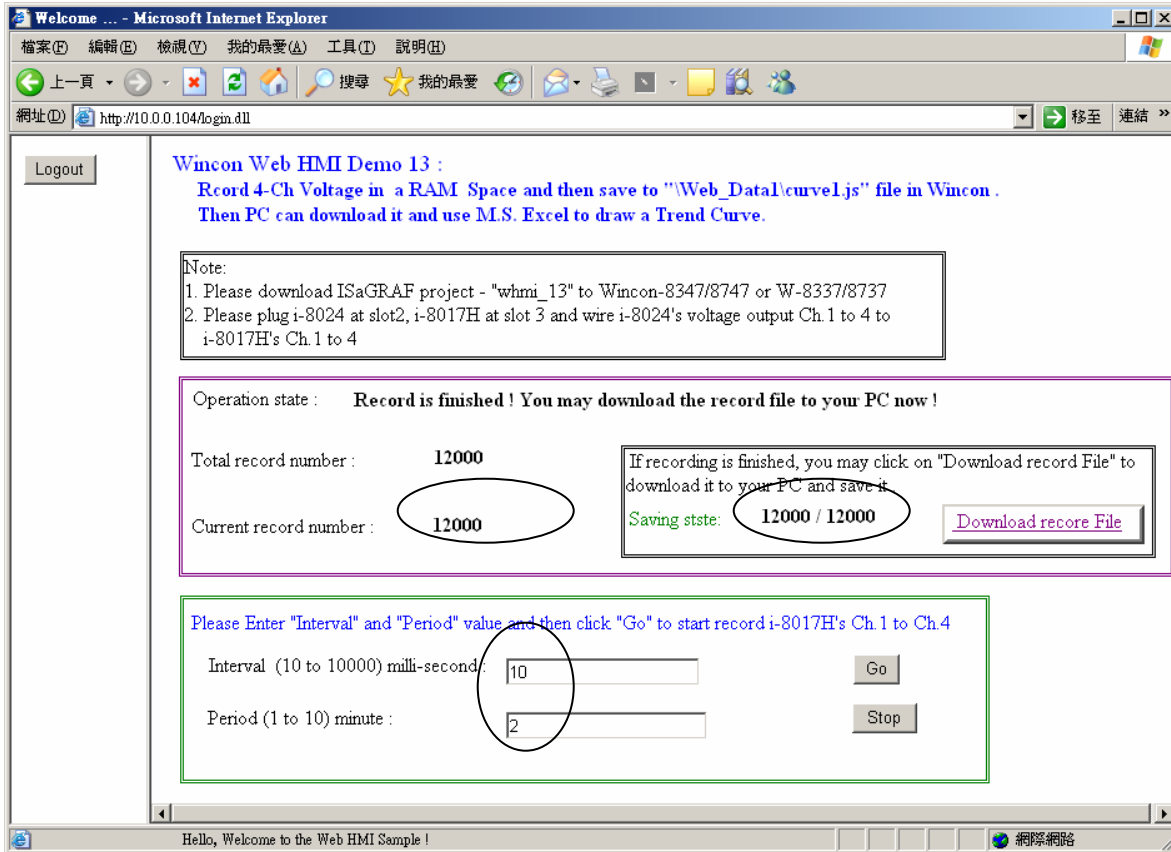
3. Download ISaGRAF project “whim\_13” to W-8xx7. (If using Web HMI as HMI, please finish procedure listed in step 2 first, then do step 3 )

The “Whmi\_13.pia” can only run in Wincon-8xx7/8xx6 (not in I-8xx7). It resides at  
W-8xx7 CD-ROM:\napdos\isagraf\wincon\demo\ or  
[ftp://ftp.icpdas.com/pub/cd/wincon\\_isagraf/napdos/isagraf/wincon/demo/](ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/)

4. PC run Internet Explorer (I.E. version should be 5.0 or later version). Enter W-8xx7 IP. If connecting well, click on “Enter”



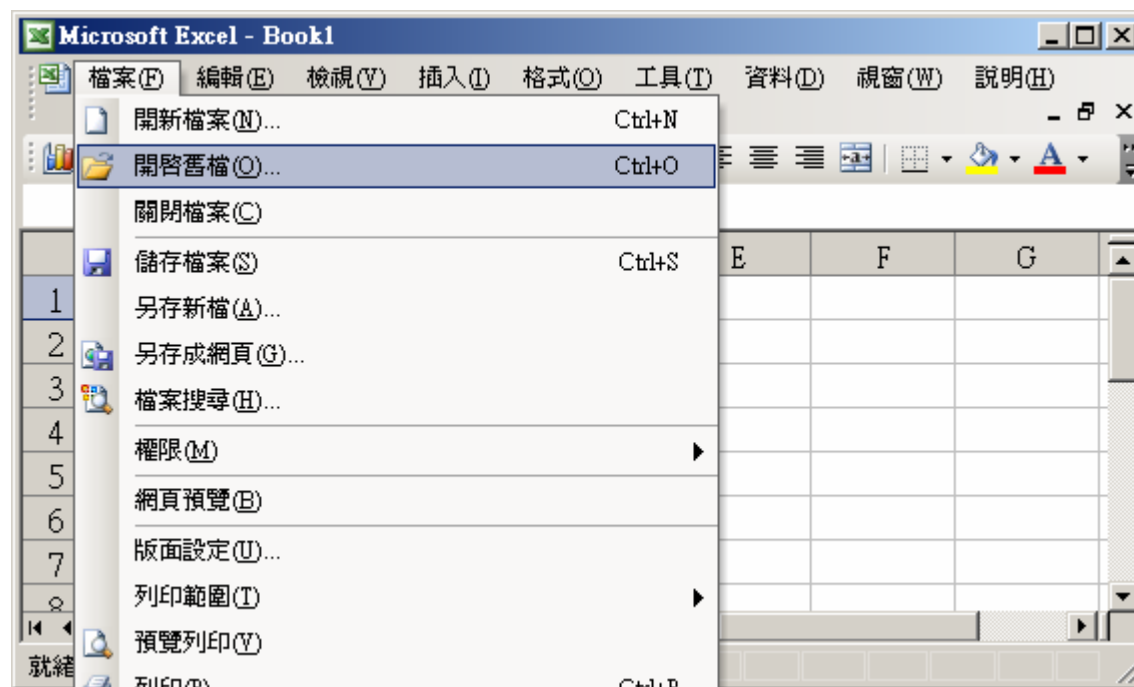
5. Then please enter proper “Interval” value. Unit is 0.001 second (1 milli-second). For example, if enter 10, it means to store one record every 10 ms. The ”Period” is the time period to record. Unit is minute. Then please click on “Go” to start recording. W-8xx7 will then output different voltage in i-8024 Ch.1 to Ch.4 . If user has finished procedure listed in step 1 – “connect Ch1. to Ch.4 voltage output to Ch1. to Ch.4 of i-8017H” , the i-8017H Ch1. to Ch.4 ‘s voltage input will also change during this period. And they will be recorded.



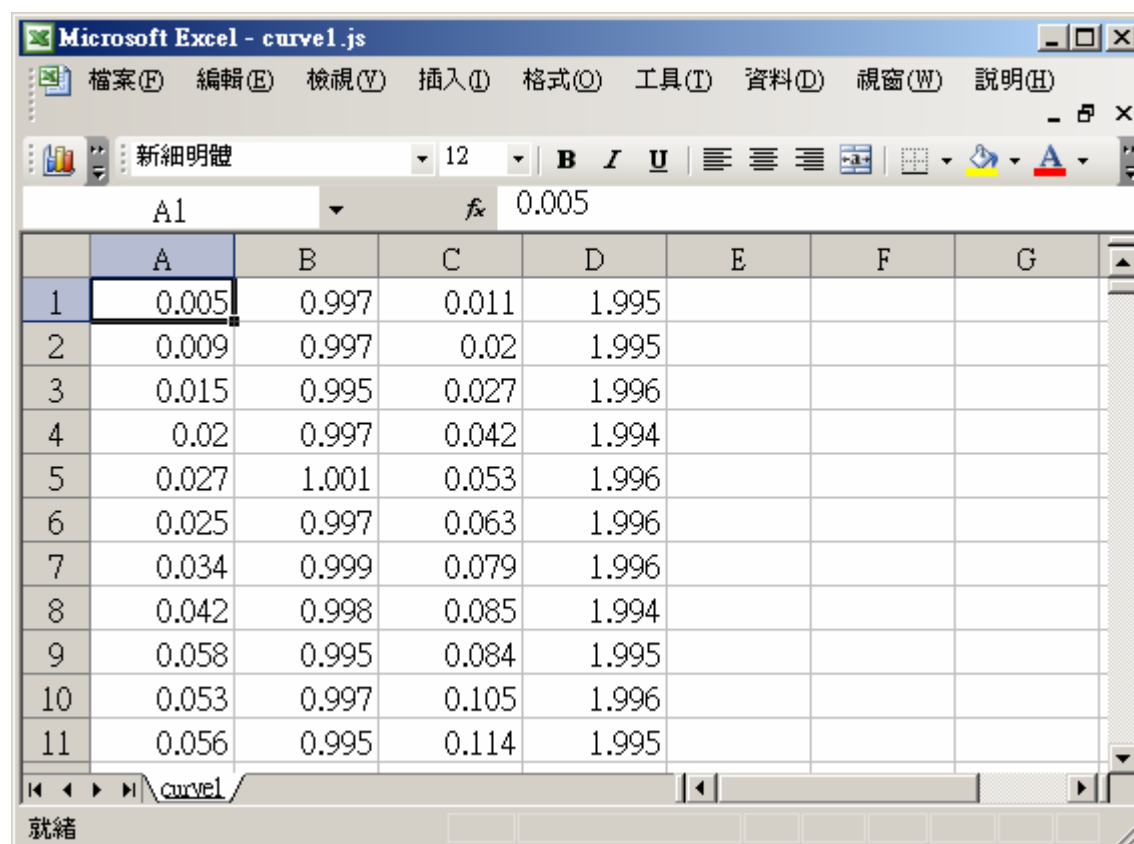
During the recording period, the “Current record number” value will count up. If it reaches the value of “Total record number” , it means recording is finished. Then the ISaGRAF program will store these records to a RAM file automatically. You can see the progress in “Saving state” . If all done, please click on “Download record File” to download this record file to your PC.



6. Then please open this record file - “curver1.js” on M.S. Excel.

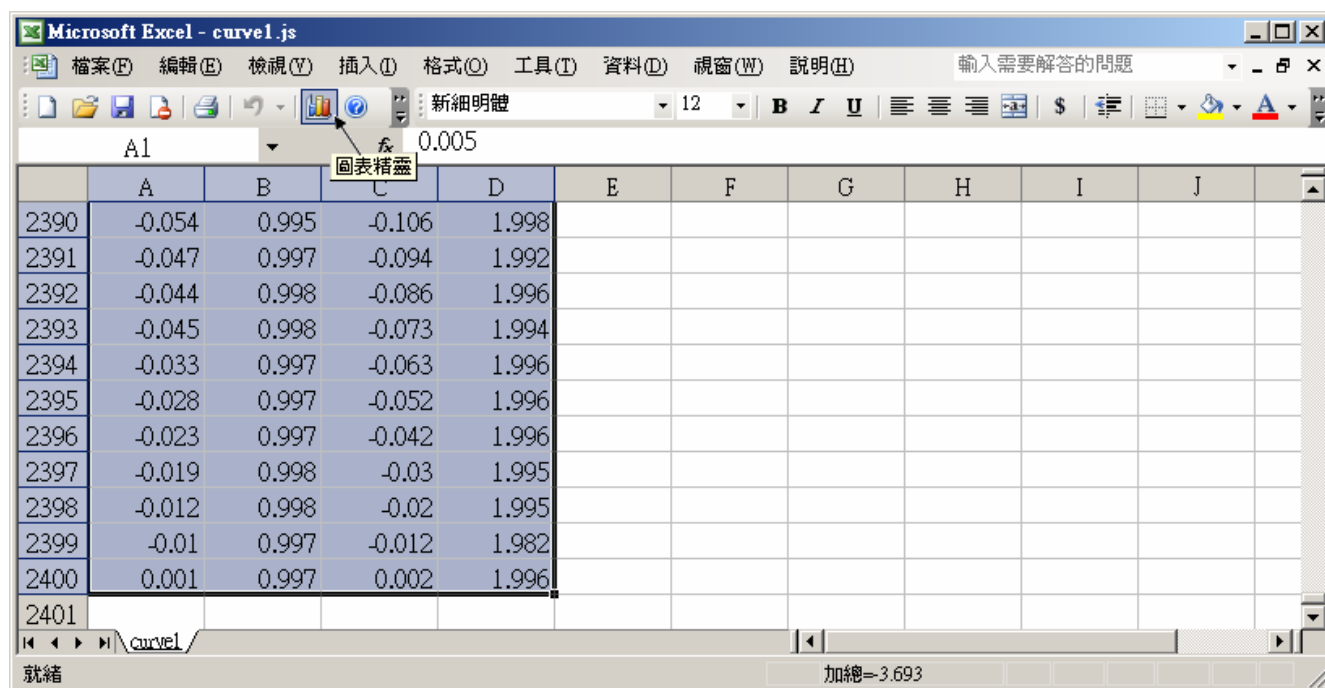


Please click on the first data at the left-top position. Then press and hold in “Shift”, and at the same time press “Ctrl” – “End”. You will see all data been selected.

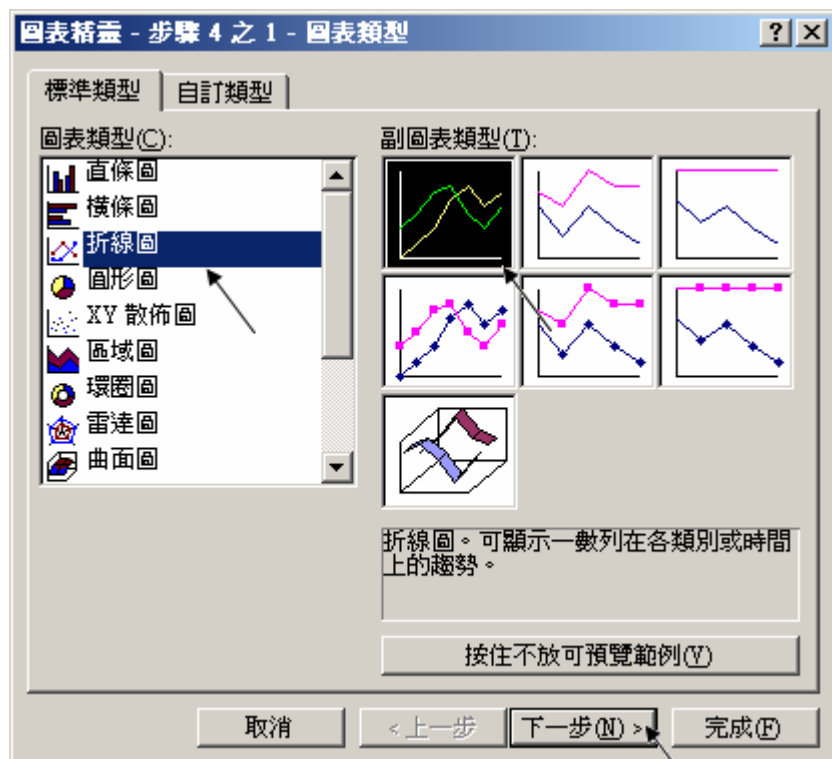


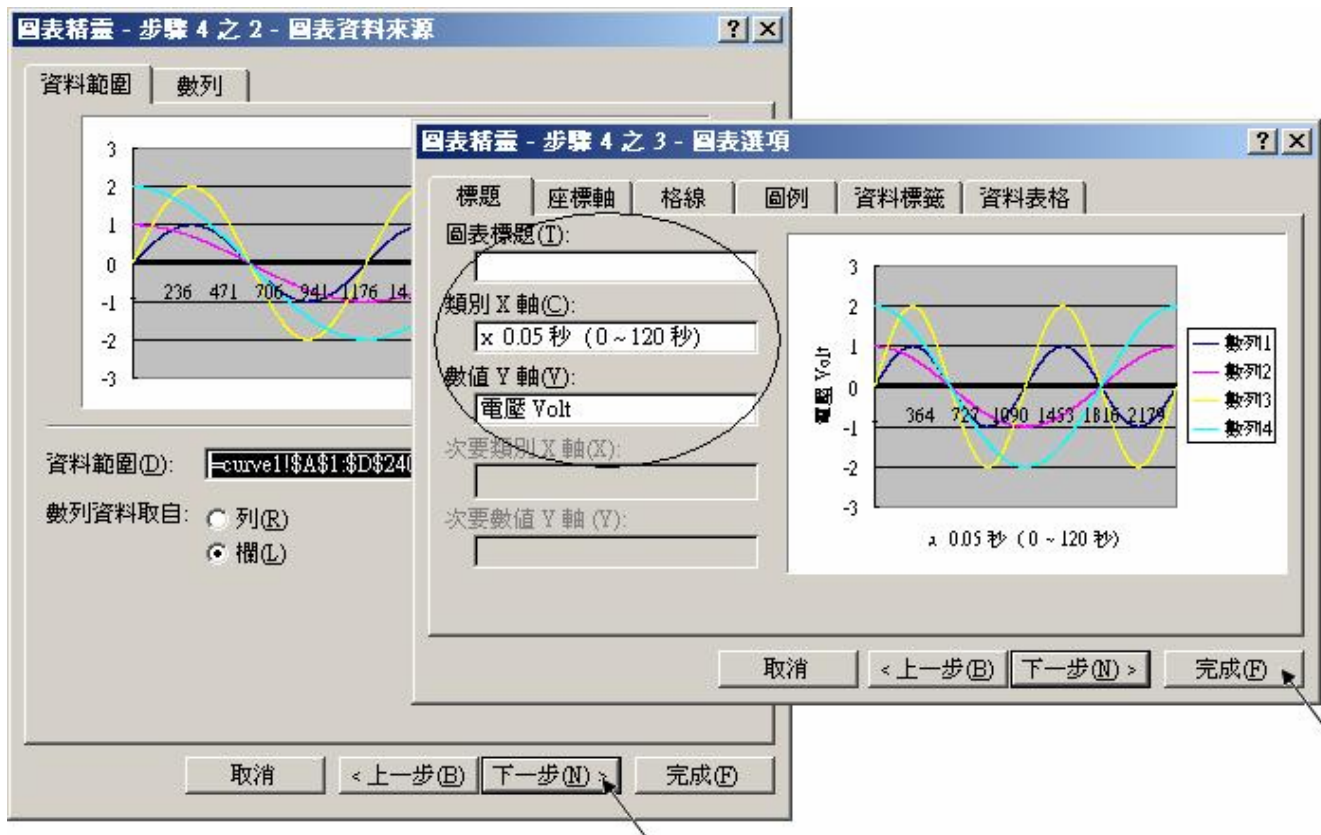


Then click on 

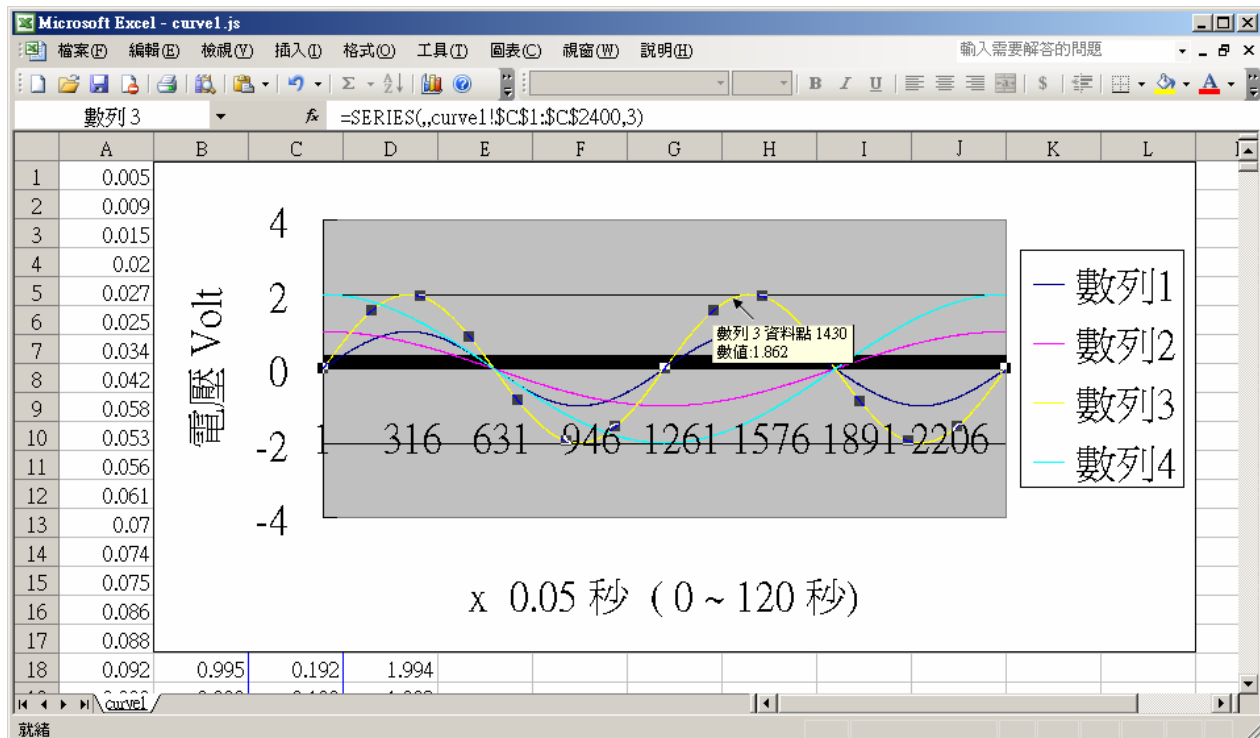


Please select the correct diagram on the left-hand side. And check the left-top type on the right-hand side. Then go Next .

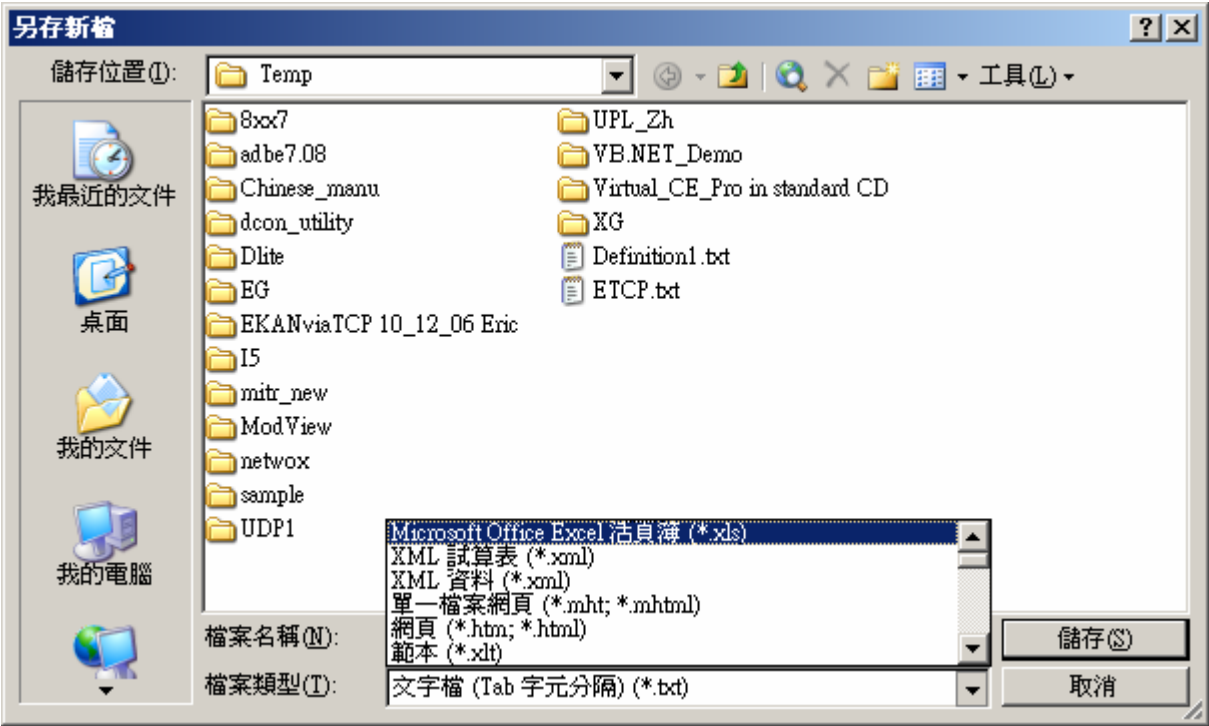
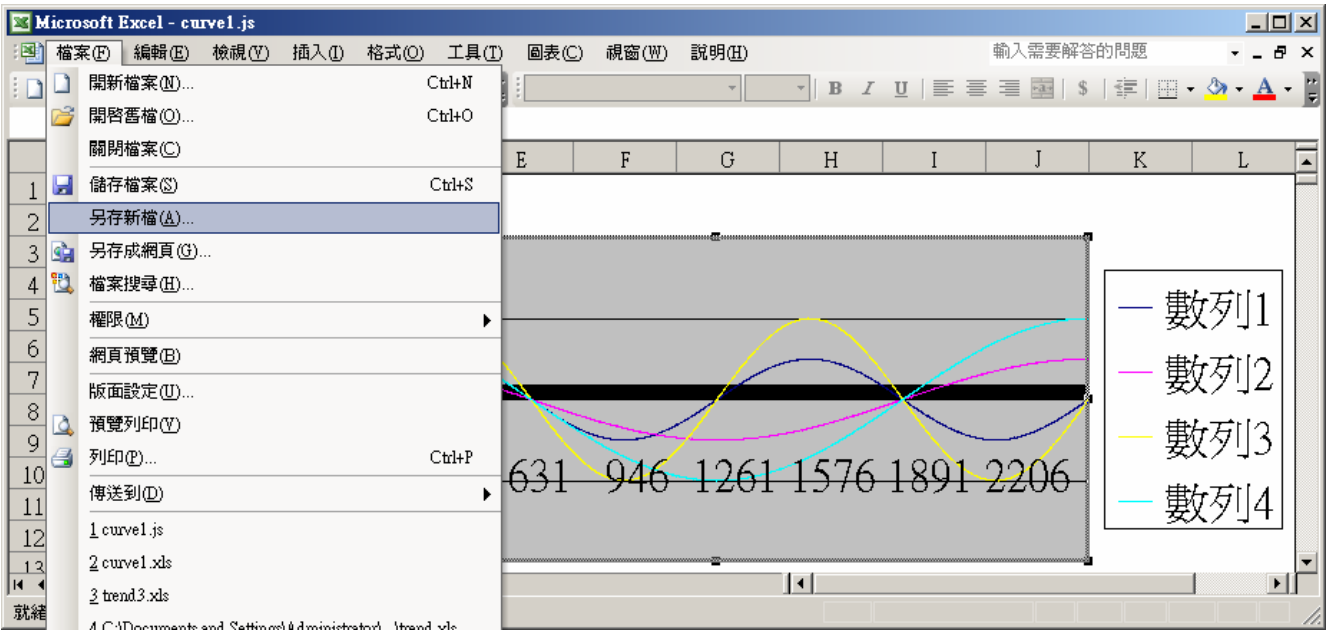




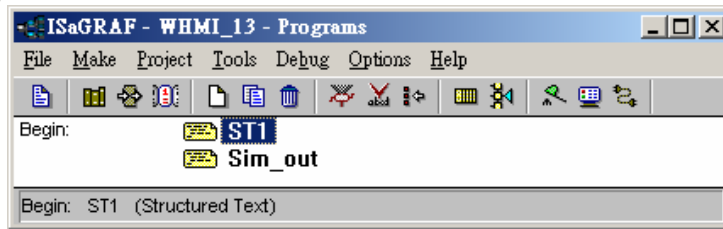
By the procedure, you will get the trend curve as the below window. You can modify its size, or check at any trend line. If you move your mouse to point at some position at the trend line, the related data is shown.



Please save this trend curve diagram as a “Microsoft Office Excel (\*.xls)” format. Then at any later time, you can open it to display the trend curve directly.



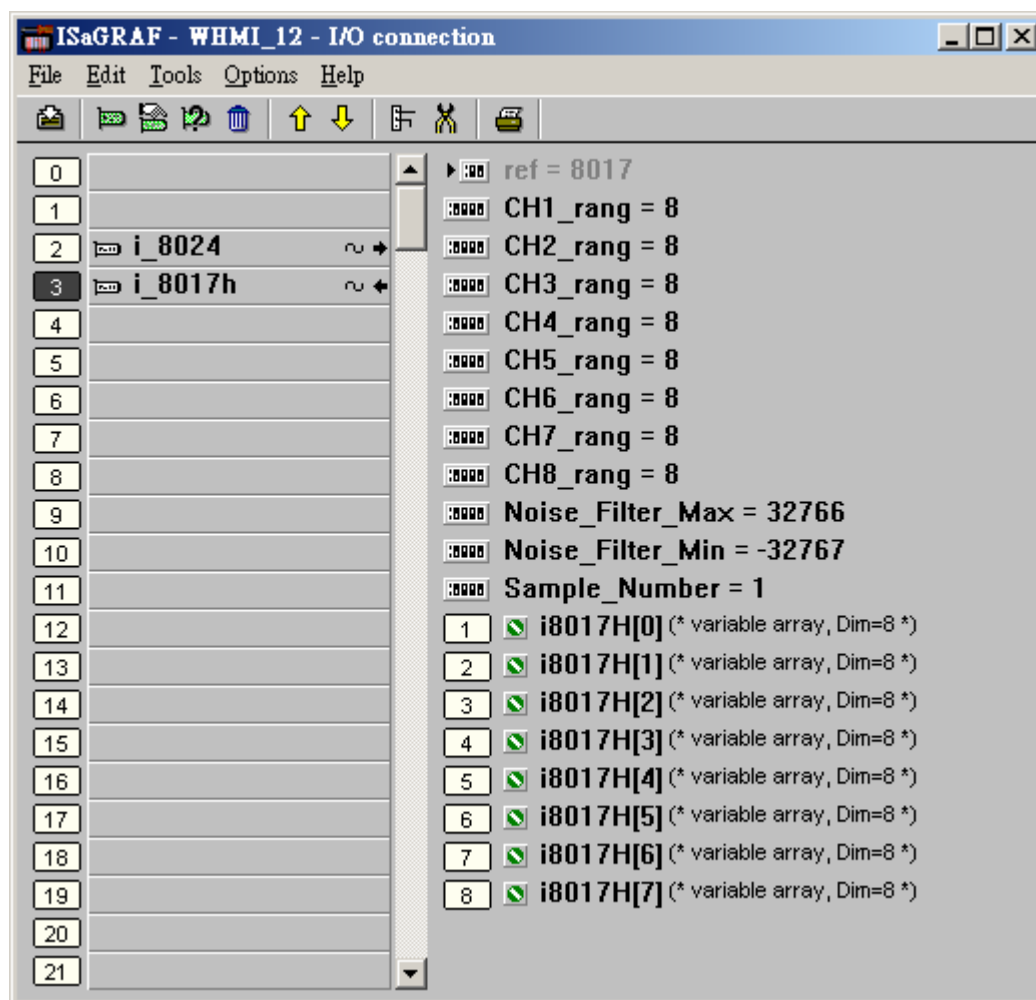
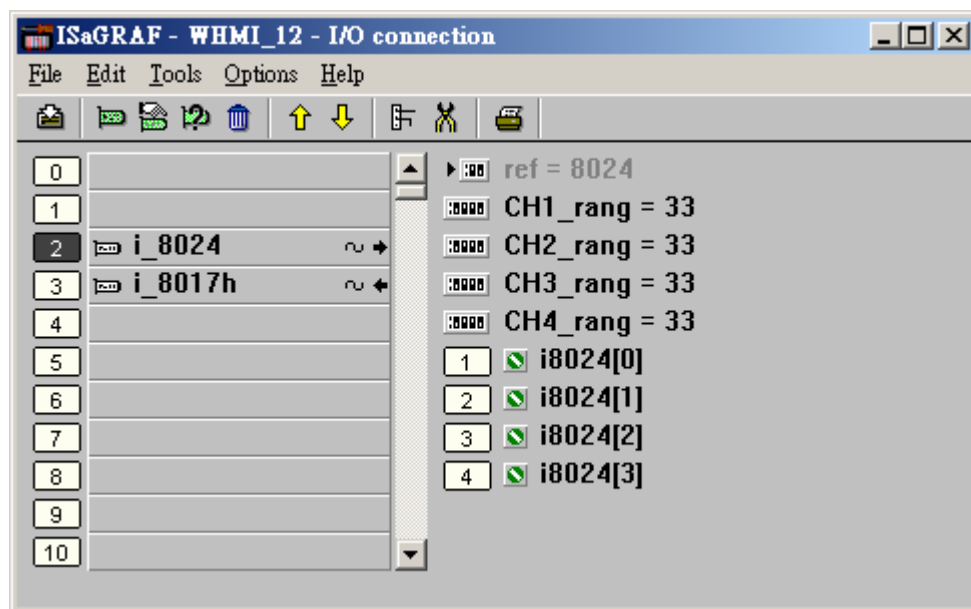
ISaGRAF project architecture:



Variables :

| Name          | Type    | Attribute       | Description  |
|---------------|---------|-----------------|--|
| Go1           | Boolean | <b>Internal</b> | Set as True to start, <b>addr defined as 21</b> (Hex. is 15)   |
| Stop1         | Boolean | <b>Internal</b> | Set as True to stop, <b>addr defined as 22</b> (Hex. is 16)  |
| TMP           | Boolean | <b>Internal</b> | Internal use   |
| INIT          | Boolean | <b>Internal</b> | Init as True   |
| Save_file1    | Boolean | <b>Internal</b> | The IsaGRAF program will set this value to True to store records to a RAM Disk File  |
| MUM_CH        | Integer | <b>Constant</b> | How many chanel in i-8017H to record ?<br>We use 4 chanel in this demo (Ch.1 to 4)   |
| File1         | Integer | <b>Internal</b> | File ID  |
| STEP1         | Integer | <b>Internal</b> | Recording state. 0:No action , 1:recording , 2:finished  |
| Period1       | Integer | <b>Internal</b> | How long to record ? unit is minute, <b>addr as 3</b>  |
| Interval1     | Integer | <b>Internal</b> | How long to save a record ? unit is ms, <b>addr as 1</b>   |
| Total_record1 | Integer | <b>Internal</b> | How many records in this recording action ? This value is calculated by the IsaGRAF program automatically. <b>addr declared as 5</b>   |
| Record_cnt1   | Integer | <b>Internal</b> | Current finished record count. <b>addr declared as 7</b>   |
| ii & ii2      | Integer | <b>Internal</b> | To use in “for” loops  |
| i8017H[0..7]  | Integer | <b>Input</b>    | Variable array, Dim as 8. link to i-8017H ‘s Ch1 to Ch. 8  |
| Volt1[0..7]   | REAL    | <b>Internal</b> | Variable array, Dim declared as 8.<br>The voltage value converted from “i8017H[0..7]”  |
| i8024[0..3]   | Integer | <b>Output</b>   | Variable array, Dim declared as 4. link to i-8024 ‘s Ch1 to Ch. 4  |
| Save_cnt1     | Integer | <b>Internal</b> | Current saving record amount in the RAM disk File, <b>addr declared as 9</b>   |
| TMP_v         | Integer | <b>Internal</b> | Internal use   |
| T1            | Timer   | <b>Internal</b> | For counting time  |
| T1_next       | Timer   | <b>Internal</b> | The time to get and save next record   |
| T1_Interval   | Timer   | <b>Internal</b> | The interval time between two record   |
| File_name1    | Message | <b>Internal</b> | File name, Len is 64, init as \Web_Data1\curve1.js<br>Web HMI support only RAM Disk File in \Web_Data1\ , If the file is in CompactFlash File, Web HMI support only in \CompactFlash\Temp\HTTP\Data\ (Please refer to Chapter 11.2 - Whmi_08 demo) |
| Msg1          | Message | <b>Internal</b> | Operation state message, Len is 255, init as “No Action now”, <b>addr as 41</b> (Hex. is 29)   |
| Str1          | Message | <b>Internal</b> | Len is 255, internal use   |

IO connection:



## ST program – Sim\_out

---

```
(* Output i-8024 's Ch1 to Ch4 as different voltage curve *)

(* 2 * Pi * T1 / 60000 = T1 * 1.047197E-4 *)
(* 2 * Pi * T1 / 120000 = T1 * 5.235985E-5 *)

i8024[0] := ANA( sin( REAL(T1) * 1.047197E-4 ) * 3276.8 ) ;
i8024[1] := ANA( cos( REAL(T1) * 5.235985E-5 ) * 3276.8 ) ;
i8024[2] := ANA( sin( REAL(T1) * 1.047197E-4 ) * 6553.6 ) ;
i8024[3] := ANA( cos( REAL(T1) * 5.235985E-5 ) * 6553.6 ) ;
```

---

## ST program – ST1

---

```
(* W-8xx7 can have max. speed of 100Hz to record data ( minimum sample interval is 10 ms *)
(* This example assume max. 8-Ch. , so 1 second will record 100 x 8 REAL value *)
(* 1 minute will record 100 x 8 x 60 = 48,000 REAL value *)
(* If period is set as 10 minute, we need 48,000 x 10 = 480,000 REAL value memory = 480,000 x 4 =
1,920,000 bytes *)
```

```
if INIT then
  INIT := False ;
```

```
  (* Allocate 500,000 integer (or 32-bit REAL) space to store records up to 10 minutes. total 500,000 x 4
= 2,000,000 bytes , W-8xx7 support only No.1 Arccreate( ) up to 3,000,000 integer space, that is
12,000,000 bytes . The first parameter in ARcreate( ) should be 1, it doesn't support 1st parameter as 0, 2,
3, ... , 15 *)
```

```
  (* Arccreate( ) can be called only once in the ISaGRAF program *)
  TMP_v := ARcreate(1, 500000) ;
  if TMP_v <> 1 then
    Msg1 := 'Parameter error or can not allocate memory by ARcreate( ) function!' ;
  end_if ;
```

```
  TMP := PLC_mode(-1) ;  (* Set W-8xx7 ISaGRAF driver running at fastest mode *)
end_if ;
```

```
(* If stop command is gived *)
```

```
if Stop1 then
  Stop1 := False ;
  STEP1 := 0 ; (* 0: no action *)
  TStop(T1) ; (* stop T1 *)
  T1 := T#0s ;
  Msg1 := 'User stop recording !' ;
  save_cnt1 := 0 ;
end_if ;
```

(\* If start command is given \*)

**if Go1 then**

**Go1 := False ;**

(\* STEP1 : 0: no action , 1: recording , 2: record finished \*)

**if STEP1=1 then**

(\* It is still recording now \*)

**Msg1 := 'It is still recording now ... Please wait' ;**

**else**

(\* Check interval valid or not \*)

(\* we assume 10 to 10000 ms is valid in this example \*)

(\* If your average PLC scan time is larger, for example, near 10 ms,

Please use Interval larger than 10 ms. Or the record time won't be correct \*)

**if (Interval1 < 10) or (Interval1 > 10000) then**

**Msg1 := 'Wrong Interval value, it should be in 10 to 10000 milli-second !' ;**

(\* Check period valid or not \*)

(\* we assume 1 to 10 minute is valid in this example \*)

**elsif (Period1 < 1) or (Period1 > 10) then**

**Msg1 := 'Wrong Period value, it should be in 1 to 10 minute !' ;**

**else**

(\* parameter is correct, start recording \*)

**total\_record1 := (Period1 \* 60000) / Interval1 ;** (\* calculate total record number \*)

**record\_cnt1 := 0 ;** (\* reset current record count as 0 \*)

**STEP1 := 1 ;** (\* set step as 1:recording \*)

**Msg1 := 'Recording now ... Please wait' ;**

(\* start ticking T1 from 0 second \*)

**T1 := T#0s ;**

**T1\_Interval := TMR(Interval1) ;**

**T1\_next := T1 + T1\_Interval ;**

**TStart(T1) ;** (\* ticking now \*)

**save\_cnt1 := 0 ;**

**end\_if ;**

**end\_if ;**

**end\_if ;**

```

(* in reconrding state *)
if STEP1 = 1 then

    (* store one record *)
    if T1 >= T1_next then

        (* Re-calculate next T1 *)
        T1_next := T1_next + T1_Interval ;

        (* T1 will be overflow after T#23h59m59s999ms, so reset it at T#20h *)
        if T1 >= T#20h then
            T1 := T#0s ;
            T1_next := T1 + T1_Interval ;
        end_if ;

        (* record data *)
        for ii := 0 to NUM_CH-1 do
            Volt1[ii] := Real(i8017H[ii]) * 0.000305176 ; (* convert to voltage *)

            (* using Real_int( ) to map REAL value to become integer value & then store it by ARwrite( ) *)
            TMP_v := ARwrite( 1 , NUM_CH * record_cnt1 + ii , Real_int(Volt1[ii]) ) ;

            (* check if ARwrite( ) correct *)
            if TMP_v <> 1 then
                Msg1 := 'Can not operate ARwrite( ) !' ;
                STEP1 := 0 ; (* 0: no action *)
                TStop(T1) ; (* stop T1 *)
                T1 := T#0s ;
            end_if ;

        end_for ;

        (* Check if record number reach the end *)
        record_cnt1 := record_cnt1+1 ; (* current record count plus 1 *)
        if (record_cnt1 >= total_record1) then

            (* record is finished, prepare to save records to a RAM disk file in serval separate PLC scans *)
            STEP1 := 0 ; (* set step as 0 at the beginning of saving *)
            Tstop(T1) ;
            T1 := T#0s ;

            (* Create a new file *)
            File1 := F_creat(File_name1) ;
            if File1 = 0 then
                (* Can not create file *)
                Msg1 := 'Create File ' + 'File_nam1 Error !!!' ;
            else

```



(\* Because saving lots of data to file take lots of PLC scan time , we are not going to save all data in a single PLC scan. We will save it in serval separate PLC scans \*)

**Msg1 := ' Please wait ... Saving data to file : ' + File\_name1 + ' ...' ;**

**save\_file1 := True ;** (\* set as True to start saving RAM disk file \*)

**save\_cnt1 := 0 ;** (\* from 0 to total\_record1-1 \*)

**end\_if ;**

**end\_if ;**

**end\_if ;**

**end\_if ;**

(\* Because saving lots of data to file take lots of PLC scan time , we are not going to save all data in a single PLC scan. We will save it in serval separate PLC scans \*)

(\* save records to a RAM disk file in serval separate PLC scans \*)

**if save\_file1 then**

**for ii2 := 0 to 50 do** (\* we limit one PLC scan can save max. 50 records \*)

**if save\_cnt1 < total\_record1 then**

**str1 := '' ;** (\* init str1 as empty string \*)

**for ii := 0 to NUM\_CH - 1 do**

(\* delimiter is <TAB> character \*)

**str1 := str1 + Rea\_Str2( Int\_real(ARread(1, NUM\_CH \* save\_cnt1 + ii)) , 3 ) + '\$09' ;**

**end\_for ;**

**str1 := str1 + '\$0D\$0A' ;** (\* add <CR> <LF> at the end of each row \*)

**TMP := F\_writ\_s(File1 , str1) ;**

**save\_cnt1 := save\_cnt1 + 1 ;**

**else**

(\* saving is finished \*)

**save\_file1 := False ;**

**TMP := F\_close(File1) ;** (\* Close file \*)

**STEP1 := 2 ;** (\* 2: recond finished \*)

**Msg1 := 'Record is finished ! You may download the record file to your PC now !' ;**

**end\_if ;**

**end\_for ;**

**end\_if ;**

---

## Chapter 12. How to send email with one attached file by Wincon-8xx7 or iPAC-8447 / 8847 or uPAC-7186EG ?

---

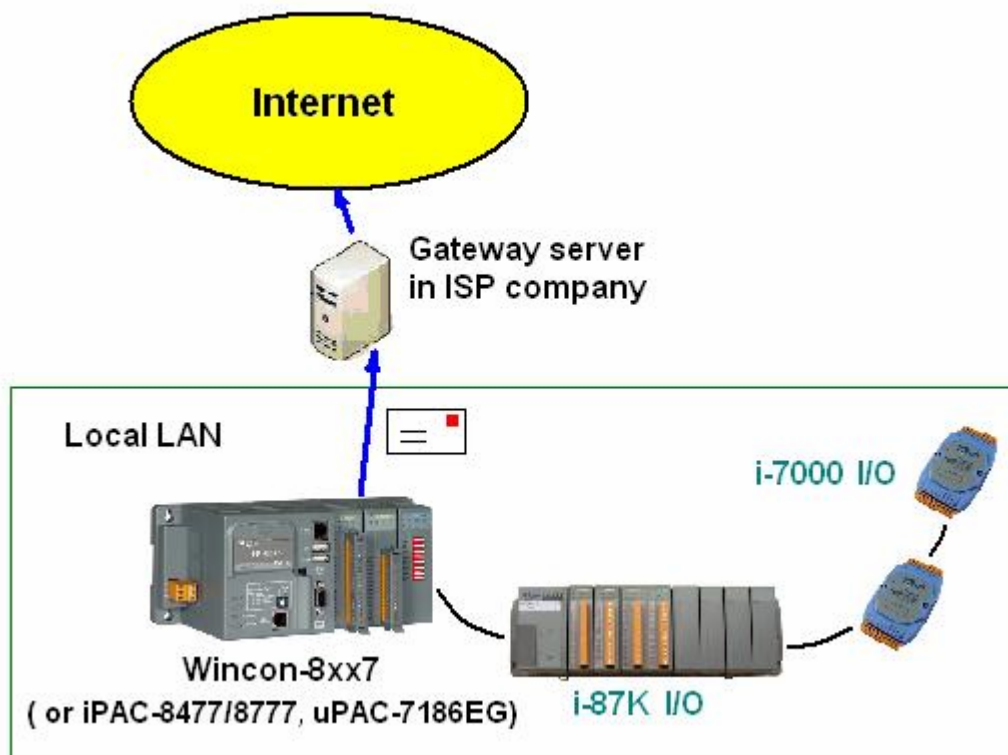
Wincon-8437/8747 , Wincon-8037/8337/8737 , iPAC-8447 / 8847 and uPAC-7186EG can send email via its Ethernet port since its following ISaGRAF driver version .

|                    |   |
|--------------------|---|
| Wincon-8xx7 :      | 3.42                                      |
| iPAC-8447 / 8847 : | since it is released (around Q1/Q2, 2008) |
| uPAC-7186EG:       | since it is released (around Q1, 2008)    |

These controllers must reside at a local network which can connect to the Internet, or sending email is not possible.

New released ISaGRAF driver: <http://www.icpdas.com/products/PAC/i-8000/isagraf-link.htm>  
Demo program: [www.icpdas.com](http://www.icpdas.com) – FAQ – Software – ISaGRAF – 067 , 071, 072, 076 and 077 or Wdemo\_62 , Wdemo\_63 and Wdmo\_63a

### Controller can send email without or with one attach file



## Features:

1. The sending Email can contain one attached file or without any attached file. The attached file format can be text or binary or any file format. The approximate max. file size is listed as the following.

|                     |            |
|---------------------|------------|
| Wincon-8xx7:        | 2M Bytes.  |
| iPAC-8447 / 8847:   | 488K Bytes |
| uPAC-7186EG + X607: | 112K Bytes |
| uPAC-7186EG + X608: | 488K Bytes |

2. Email Title can be max. 128 bytes. Email content can be max. 510 bytes. Local language word can be used (English, Chinese, any language character which computer can use).
3. One email can send to 10 receivers at one sending.
4. Each email can be assigned as High , Low or Normal priority.
5. Please assign at least one Mail server IP in the ISaGRAF program. **Or for safety, assign two Mail servers IP. Then if one Mail server is out of service, the controller will send this email by the other Mail server.**
6. If controller model is Wincon-8347 / 8747 (dual LAN) and both LAN ports are enabled, Wincon-8347 / 8747 will automatically switch to the other Ethernet port to send email if one is broken or damaged.
7. If the sending email has one file attached, this file must be stored or copied to the correct file path before it is sent.

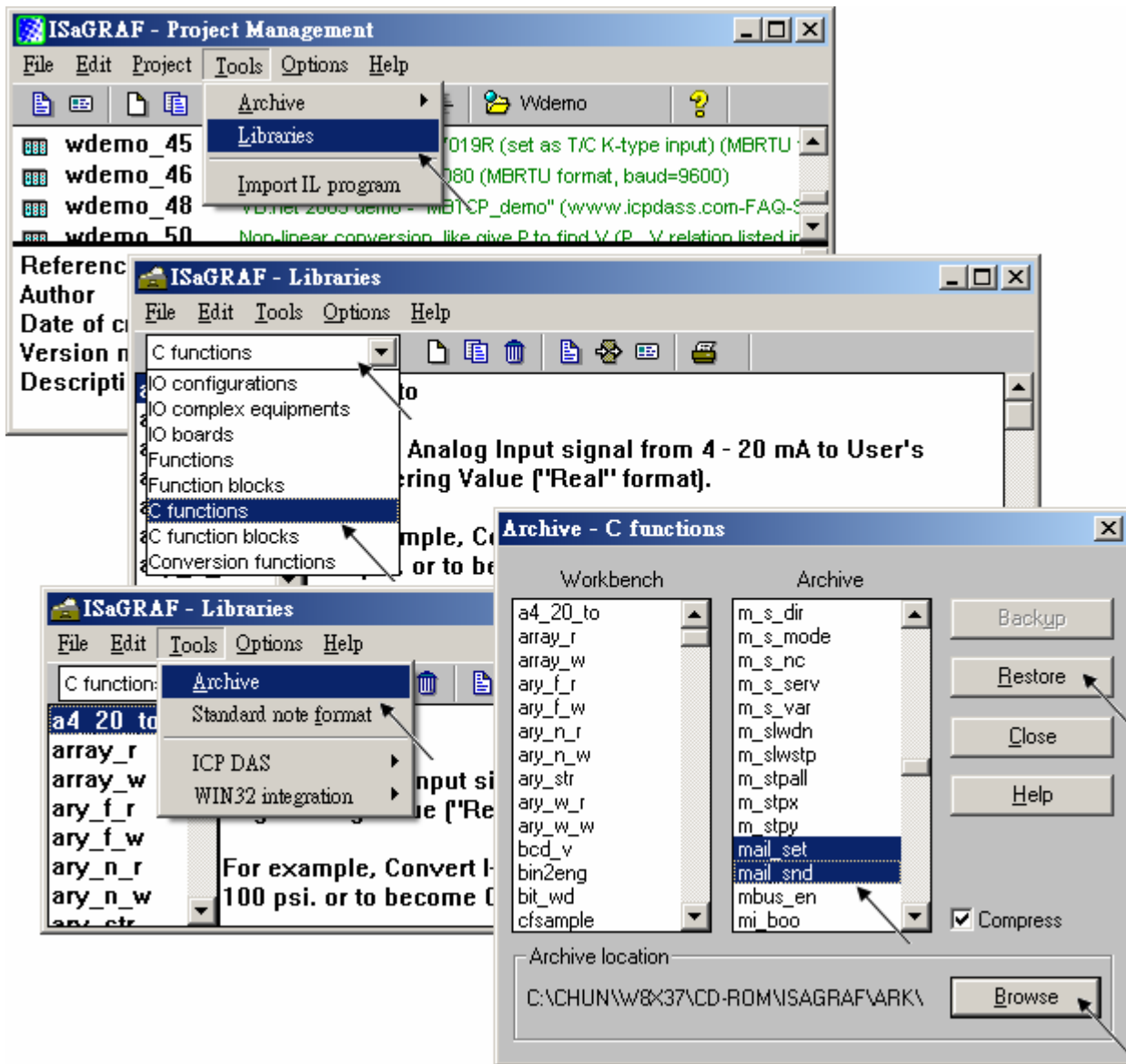
### Wincon-8xx7:

in the path of '\Email\_ETH\' , for ex, the '\Email\_ETH\A1.txt'

### iPAC-8447 / 8847 & uPAC-7186EG:

file should be stored in the battery backup memory by the "S\_xxx" functions, like the "s\_fl\_ini" , "s\_fl\_avl" , "s\_m\_r" , ... (please refer to section 10.3 and appendix A.4)

Please make sure if your ISaGRAF software in PC has installed the ISaGRAF c-function of “Mail\_snd”, “Mail\_set” and “R\_mb\_adr”. If not installed, please visit [www.icpdas.com](http://www.icpdas.com) – FAQ – Software – ISaGRAF – 067, 076 & 077 to download the Demo program. Then restore “Mail\_snd.uia”, “Mail\_set.uia” and “R\_mb\_adr.uia” to your ISaGRAF in PC by below steps.

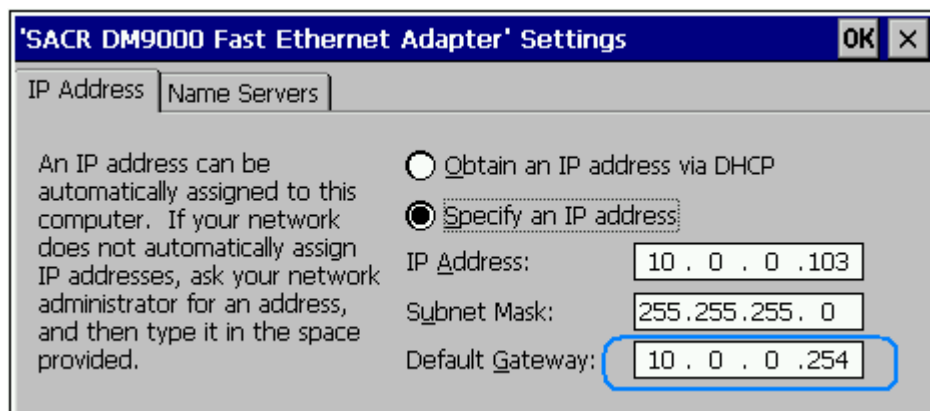


To send email correctly, please set proper Gateway IP in the controller’s Ethernet port setting. Please type command “ipconfig” in a PC ‘s command prompt window at the same local network to get the Gateway IP setting as below. (Here is 10.0.0.254)

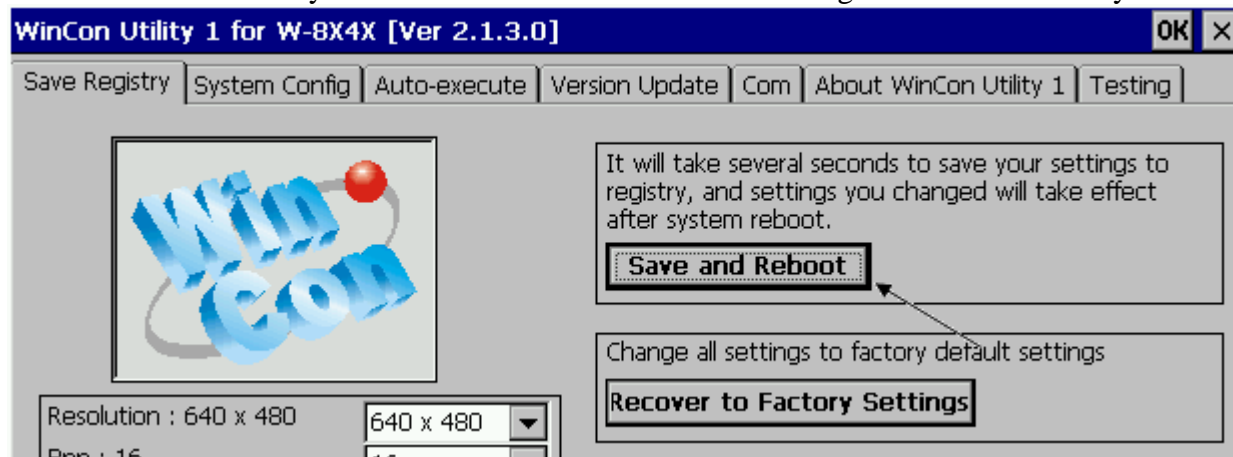


Then please fill-in this Gateway IP address to your controller's Ethernet port setting (If controller model is W-8347/8747, you can enable two Ethernet ports, then you need to fill-in both with the same Gateway IP)

#### Wincon-8xx7:



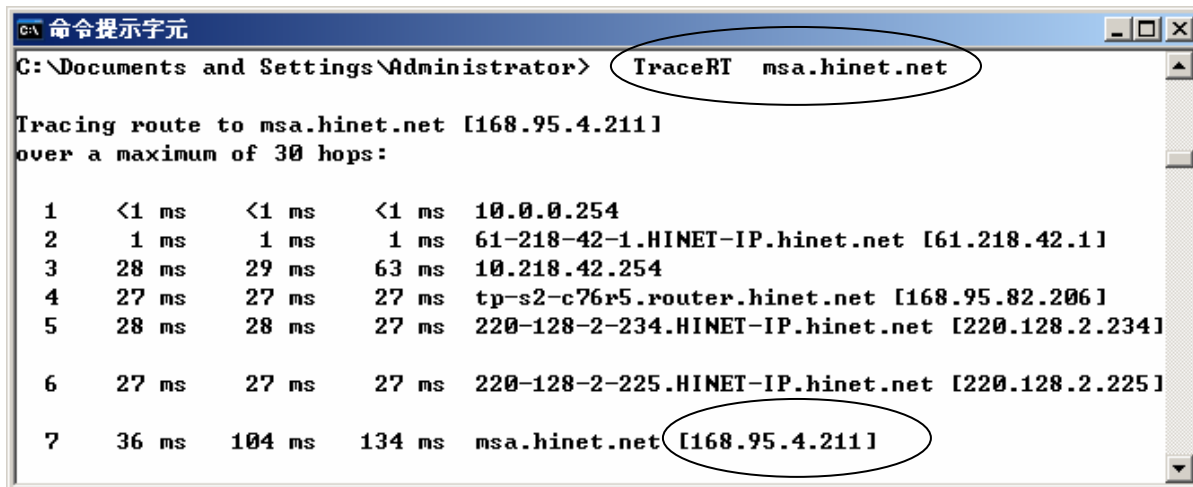
Then run Wincon Utility – Save and Reboot to store the IP setting. It will automatically re-boot once.



## iPAC-8447 / 8847 & uPAC-7186EG :

Please run “7188xw.exe” in the PC and give command for ex, “gateway 10.0.0.254” if the gateway IP is 10.0.0.254. (Please refer to appendix B)

The PC ‘s command prompt windows can also request the Mail server ‘s IP address (We need it in the ISaGRAF program). For example, to request IP of msa.hinet.net , please type command TraceRT msa.hinet.net as below (Here is 168.95.4.211)



```
C:\Documents and Settings\Administrator> TraceRT msa.hinet.net

Tracing route to msa.hinet.net [168.95.4.211]
over a maximum of 30 hops:

  1  <1 ms    <1 ms    <1 ms    10.0.0.254
  2   1 ms     1 ms     1 ms     61-218-42-1.HINET-IP.hinet.net [61.218.42.1]
  3  28 ms    29 ms    63 ms    10.218.42.254
  4  27 ms    27 ms    27 ms    tp-s2-c76r5.router.hinet.net [168.95.82.206]
  5  28 ms    28 ms    27 ms    220-128-2-234.HINET-IP.hinet.net [220.128.2.234]
  6  27 ms    27 ms    27 ms    220-128-2-225.HINET-IP.hinet.net [220.128.2.225]
  7  36 ms   104 ms   134 ms    msa.hinet.net [168.95.4.211]
```

Email demo download from FAQ – 067 , 076 and 077 has three example programs.

“Wdemo\_62.pia” is the demo without attached file.

“Wdemo\_63.pia” is the demo for Wincon-8xx7 with one attached file .

“Wdmo\_63a.pia” is the demo for uPAC-7186EG and iPAC-8447 / 8847 with one attached file .

Please modify at least the below setting in the demo program to your own setting .

```
TMP := MAIL_SET( 1 , 'chun@icpdas.com' ) ;    (* Receiver 1. please modify it *)
TMP := MAIL_SET( 100 , 'go_mao@hotmail.com' ) ; (* Sender. please modify it *)
TMP := MAIL_SET( 101 , '168.95.4.211' ) ;    (* Mail server 1 's IP, please modify it *)
```

Then re-compile it and then download it to the controller to run. The below windows will show up. Please set “to\_send” as TRUE to trigger to send one email. Few seconds later, value of “Email\_state” will be 21 or 22 if succeed. However value of “Email\_state” will be less than 0 if failed. When “Email\_progress” reach value of 100, it means the email data is 100% sent.

| ISaGRAF - WDEMO_62:LIST1 - List of variables |  |   |
|--|--|---|
| File Edit Options Help                       |  |   |
| Name   | Value                                    | Comment   |
| msg1   |  | Remember to assign the Gateway IP to controller             |
| EMAIL_state                                  | 21                                       | 0:Sleep, 1:Busy, 21:server1, 22:server2 succeed, <0:Error   |
| EMAIL_progress                               | 100                                      | progress: 0:No action, 1 - 10:connecting, 11 - 100: percent |
| Year1  | 2007                                     |   |
| Month1                                       | 7  |   |
| Day1   | 4  |   |
| WeekDay1                                     | 3  |   |
| Hour1  | 13                                       |   |
| Minute1                                      | 20                                       |   |
| Second1                                      | 34                                       |   |
| mail_subject                                 | Testing Email No. = 1                    | Email subject. Max. 128 character. (Can be local language)  |
| mail_data1                                   | 2007/7/4 13:20:27\$0D\$0AThis message is | Email data1 Max. 255 character. (Can be local language)     |
| mail_data2                                   | (More message ...)                       | Email data2 Max. 255 character. (Can be local language)     |
| TMP_v  | 1  | return value of Mail_snd( ) . 1: Ok, <0: error              |
| to_send                                      | FALSE                                    | Set as TRUE to trigger to send an email                     |
| Email_Priority                               | 3  | 1: High, 3: Normal, 5: Low                                  |
| <end of list>                                |  |   |

Below is the description of the three ISaGRAF functions for sending email.

### MAIL\_Set( CMD\_ , MSG\_ )

Parameters:

CMD\_ Integer Can be the following value.

- 1 : Set receiver 1 , for ex, TMP := Mail\_set( 1 , 'chun\_tsai@icpdas.com' ) ;  
Max. receiver length can not exceeds 48 characters.
- 2 to 10 : Set receiver 2 to 10 if they exist.
- 100 : Set the sender , for ex, TMP := Mail\_set( 100 , 'sender1@icpdas.com' ) ;  
Max. sender length can not exceeds 48 characters.
- 101 : Set the mail server 1 's IP address , for ex, TMP := Mail\_set( 101 , '168.95.4.211' ) ;
- 102 : Set the mail server 2 's IP address if it exist.
- 103 : a new TCP port No. for sending email. (Default is 25 "SMTP protocol" )

MSG\_ Message the related message setting according to the 1st parameter - CMD\_

Return:

Q\_ Boolean True : Ok .  
False : the related setting is not correct or the "CMD\_" value is not correct.

**MAIL\_snd**( Start\_ , Num\_ , Subject\_ , Prio\_ , Data1\_ , Data2\_ , Attach\_ )

Parameters:

Start\_ Integer Starting receiver No. Can be 1 through 10.  
Num\_ Integer Number of receivers. Can be 1 through 10.  
Subject\_ Message Subject of the email. Max. length is 128 characters.  
For ex, 'Alarm of plant 1'  
Prio\_ Integer Set Email Priority symbol.  
Value can be 1 : High , 3 : Normal , 5 : Low .default setting is 3.  
Data1\_ Message The email data 1 (Max. 255 characters).  
For ex, 'Pressure 1 is too high. Please check it soon ! ... '  
Data2\_ Message The email data 2 (Max. 255 characters).  
More message behind the "Data1\_". For ex, 'More message ...'  
Attach\_ Message The attached file name if it exists. Max. 64 charcters.  
Please give " (empty messge) if no attached file used.

Wincon-8xx7 : file must store in the '\Email\_ETH\' folder . For ex, '\Email\_ETH\A1.txt'

uPAC-7186EG : file must store in the X-607, X-608 memory .

iPAC-8447/8847 : file must store in the built-in battery SRAM in the backplane .  
the valid value is '1' , '2' , ... , '8' . the number is the file ID No. set  
by the "S\_FL\_AVL" function. (refer to section 10.3 or appendix A.4)

The max. attached file size are listed as following.

|                     |              |                     |            |
|---------------------|--------------|---------------------|------------|
| Wincon-8xx7:        | 2M Bytes. ,  | iPAC-8447 / 8847:   | 488K Bytes |
| uPAC-7186EG + X607: | 112K Bytes , | uPAC-7186EG + X608: | 488K Bytes |

Return:

Q\_ Integer

- 1 : Ok, then start sending email ...
- < 0 : error
- 1 : Busy . The earlier email is still sending ...
- 2 : The first Receiver (No. = "Start\_" ) is empty or error.
- 3 : Mail server 1 is empty or error.
- 4 : Sender is empty or error.
- 5 : "Start\_" value less than 1 or larger than 10
- 6 : "Subject\_" exceeds 128 characters.
- 7 : Email system is not active yet, Please use "mail\_set( )" to set at leaset one receiver  
email box address, one mail server IP and the sender email box address
- 8 : "Num\_" value less than 1 or larger than 10
- 9 : The given attached file name doesn't exist or file path name > 64 characters or  
its size exceeds the allowed file size.



**R\_MB\_ADR( 1 , 9995 )** is to get the email sending state sent by "Mail\_snd( )" .

The return value of R\_MB\_ADR( 1 , 9995 ) will remain until next calling "Mail\_snd( )" .

Return :

- 0 : Sleep. No action
- 1 : Busy. one email is still sending now
- 21 : Email is successfully sent through Mail server 1
- 22 : Email is successfully sent through Mail server 2
  
- < 0 : Error happens
- 1 : Can not connect to the Mail server
- 2 : Sender setting is rejected by the Mail server
- 3 : Time out
- 4 : Ethernet socket error
- 5 : receiver setting is reject by the Mail server

**R\_MB\_ADR( 1 , 9994 )** is to get the current email sending progress sent by "Mail\_snd( )" .

Calling "R\_MB\_ADR( 1 , 9994 )" can not get the Error No. when error happens. Please use "R\_MB\_ADR( 1 , 9995 )" to get it .

If error happens while sending email, the return value will stay at its last value until next calling "Mail\_snd( )" .

Return :

- 0 : No action
- 1 : Connecting to Mail server 1
- 2 : Mail server 1 connected . Sending "HELO "
- 3 : Sending "MAIL FROM: ..." to Mail server 1
- 4 : Sending "RCPT TO: ..." to Mail server 1
- 5 : Sending "DATA" to Mail server 1
- 6 : Connecting to Mail server 2
- 7 : Mail server 2 connected . Sending "HELO ..."
- 8 : Sending "MAIL FROM: ..." to Mail server 2
- 9 : Sending "RCPT TO: ..." to Mail server 2
- 10 : Sending "DATA" to Mail server 2

11 ~ 100 : the current progress of sending email data.

For ex,

|     |       |                                     |
|-----|-------|-------------------------------------|
| 25  | means | 25 / 100 = 25 %                     |
| 36  | means | 36 / 100 = 36 %                     |
| 95  | means | 95 / 100 = 95 %                     |
| 100 | means | 100 / 100 = 100 % (sent completely) |

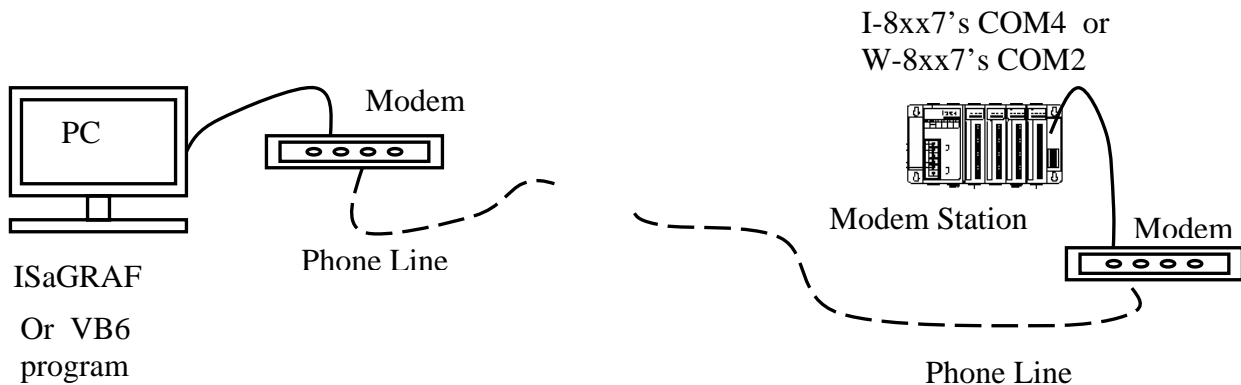
# Chapter 13. Remotely Download Via Modem\_Link

## 13.1: Introduction

COM4 of The I-8417/8817/8437/8837, i-8437-80/8837-80, iPAC-8447/8847 & COM2 of the W-8xx7 supports full modem signals. It has embedded the Modem\_Link protocol for remotely download and monitoring since the I-8xx7 driver version of 2.14 & W-8xx7 driver version of 3.10. Please refer to Appendix C to make sure your I-8xx7 controller driver version is the same or higher. You can obtain the new released driver from:

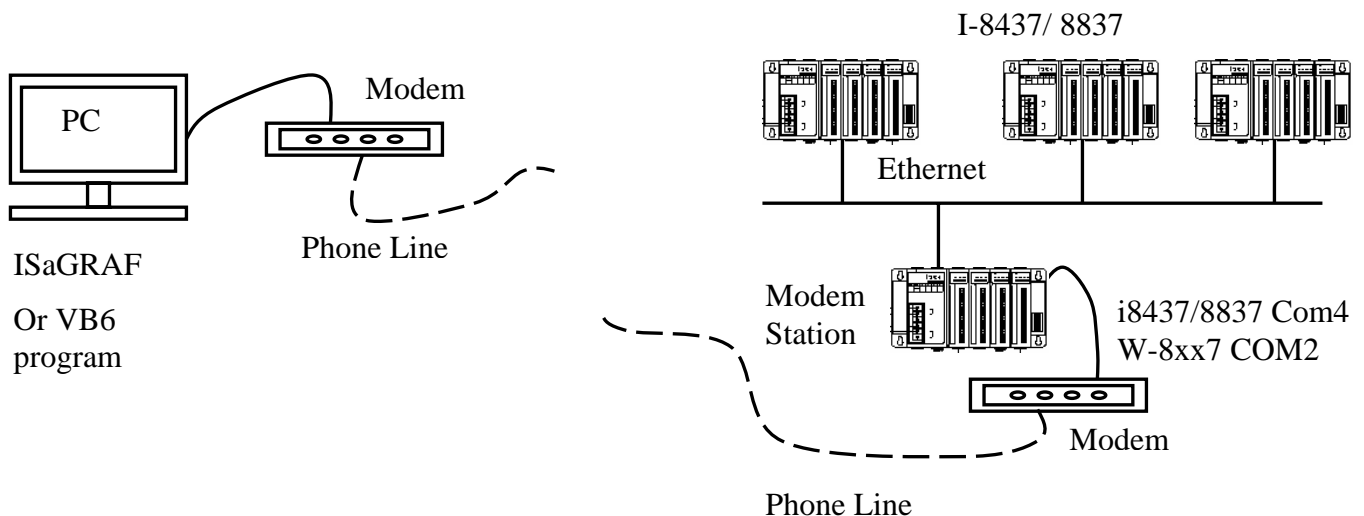
<http://www.icpdas.com/products/PAC/i-8000/isagraf.htm>

To Remotely download and monitor program via the Modem\_Link, I-8xx7's Com4 & W-8xx7's Com2 has to link to a modem. They have exactly the same pin assignments as the Com1 (9-pin Dsub) of the PC. We name the controller as “**Modem Station**” since it will pick up the phone call coming from the remote



PC running ISaGRAF. If the controller is either I-8437 or I-8837 (Ethernet controller), The configuration can be extended to link many controllers together. Therefore, the PC running ISaGRAF can remotely download to anyone of them through the modem and the Modem station.

**Note:** W-8xx7's COM2 can be set as Modbus RTU port, **please disable it if using as “modem\_link”**



**port.** Please refer to W-8xx7's “Getting Started” Manual.

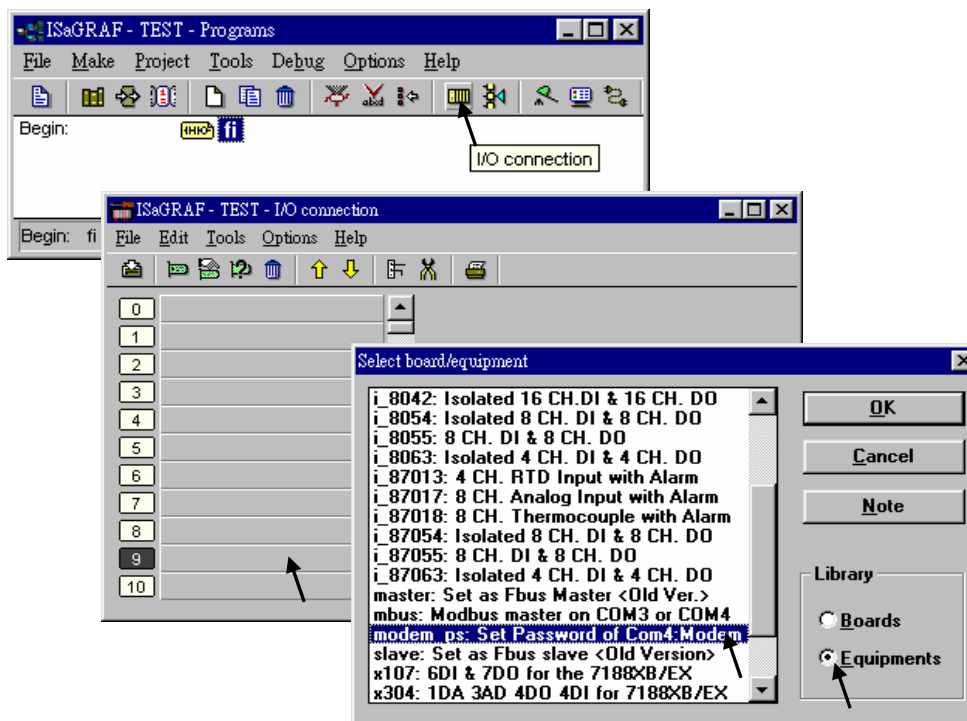
## 13.2: Download Program Via Modem\_Link

### Warning:

Do not download a project which uses I-8xx7's Com4 & W-8xx7's COM2 to do other things to the "Modem station" controller. For ex, do not connect "Bus7000" & "Mbus" with port\_no = 4 (for I-8xx7) & port\_no=2 (for W-8xx7). And do not use "Comopen" to open Com4(for I-8xx7) & Com2(for W-8xx7). It will disable "Modem\_Link" if you use them for other purpose. That means, you can not remotely connect to it.

**Note:** W-8xx7's COM2 can be set as Modbus RTU port, **please disable it if using as "modem\_link" port.** Please refer to W-8xx7's "Getting Started" Manual.

The first thing is to add a "modem password" to your ISaGRAF program of the "Modem station" controller for security. To do it, click on one empty slot No. from the I/O connection window. Then connect "Modem\_PS" on the slot.

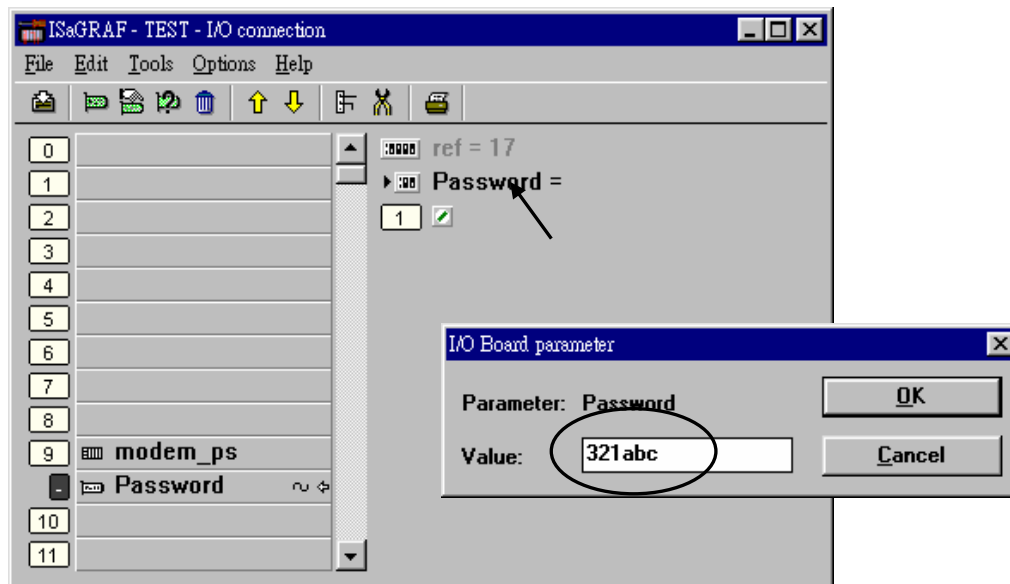


Then you got the window similar as below. Type in your preferred password for the "Modem station" controller. The password can contain up to 12 characters & can't use character " and ' . Then re-compile it and download it to the "Modem station" controller.

### Note:

User can write Visual Basic program to access to the I-8417/8817/8437/8837 & W-8xx7 via Modem. Please download VB6 demo source code at

<http://www.icpdas.com/products/PAC/i-8000/I-8417.htm> or  
[ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/vb\\_demo/](ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/vb_demo/) or  
CD-ROM:\napdos\isagraf\vb\_demo\

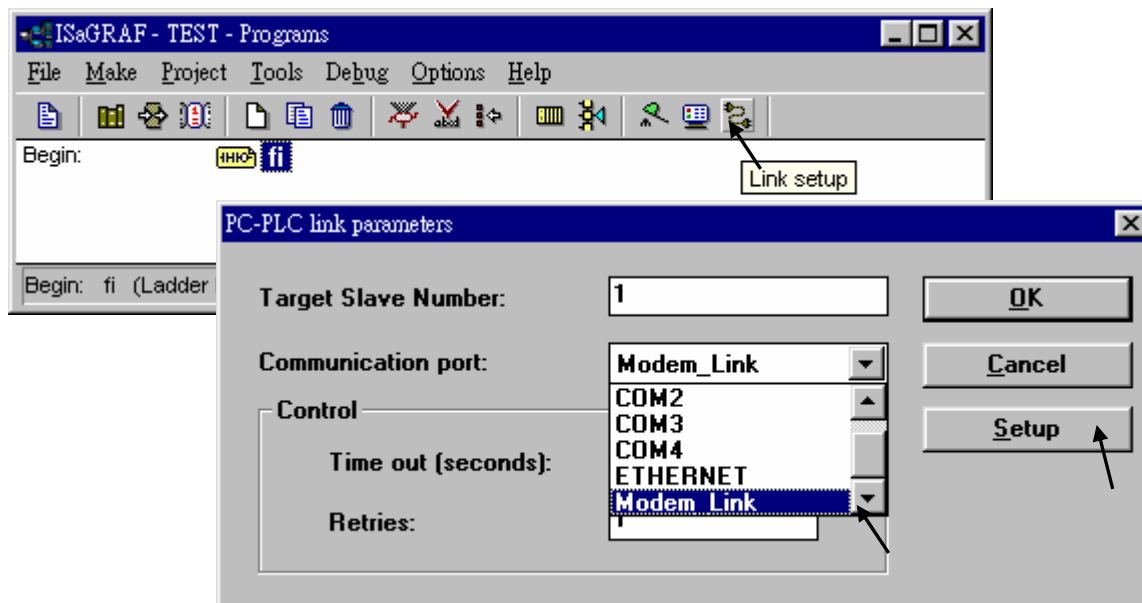


### Very Important:

If you don't assign the Modem password to the "Modem station" controller, anyone who has the phone No. of your "Modem station" controller can link to it to do anything. Be very careful.

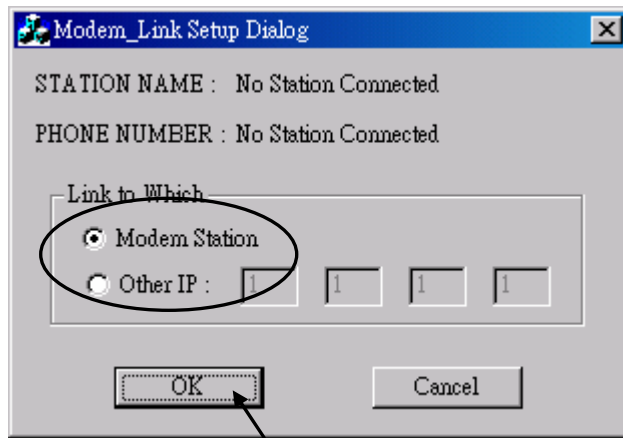
Now we are going to download and monitor the program of faraway controllers.

Click on "Link setup", select "Modem\_Link", and then click on "Setup"

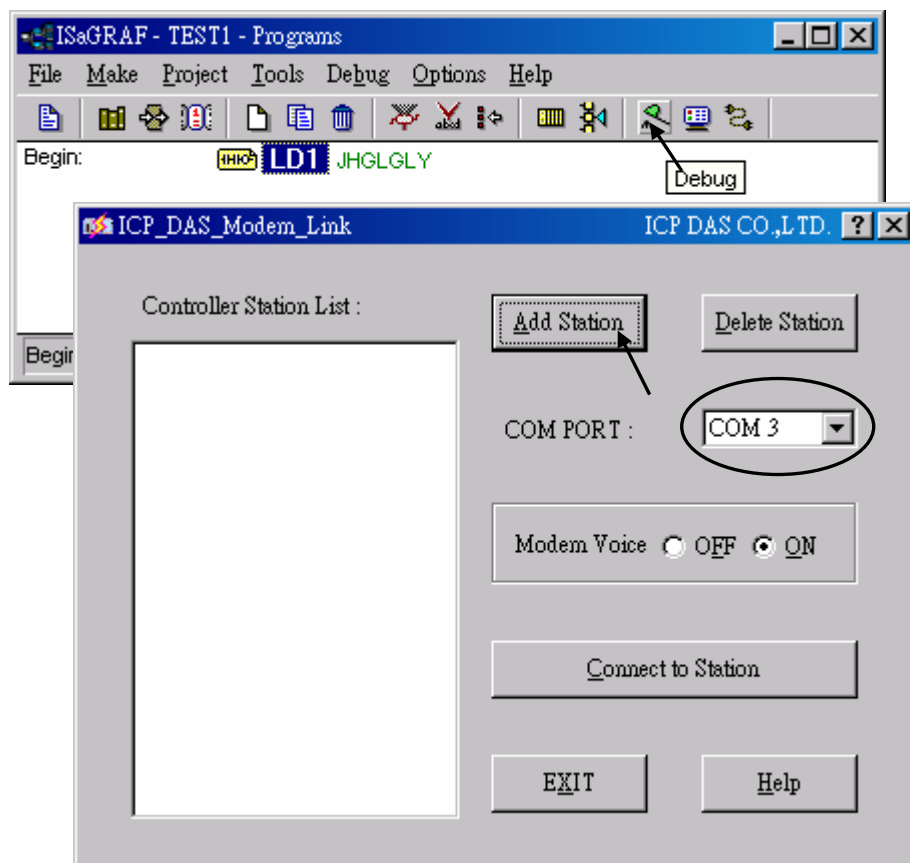


### For windows NT, 2000 & XP users:

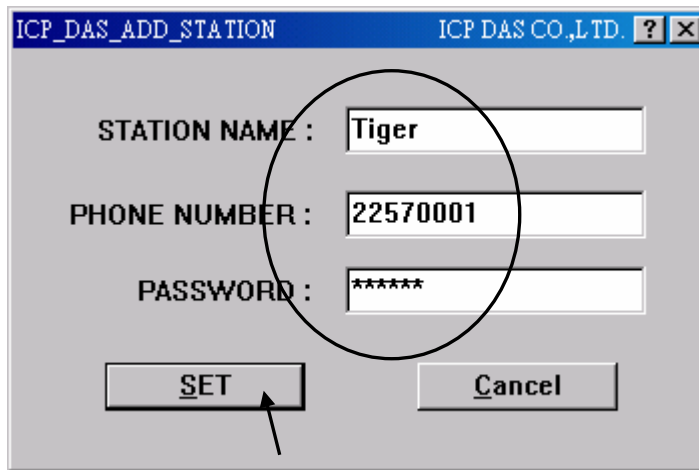
If you are going to connect the “Modem station” controller, check “Modem station”, otherwise check “Other IP”. “Other IP” means the target controller is not connect to a modem however connect to the “Modem station” controller via an ethernet cable, the IP address has to be assigned.



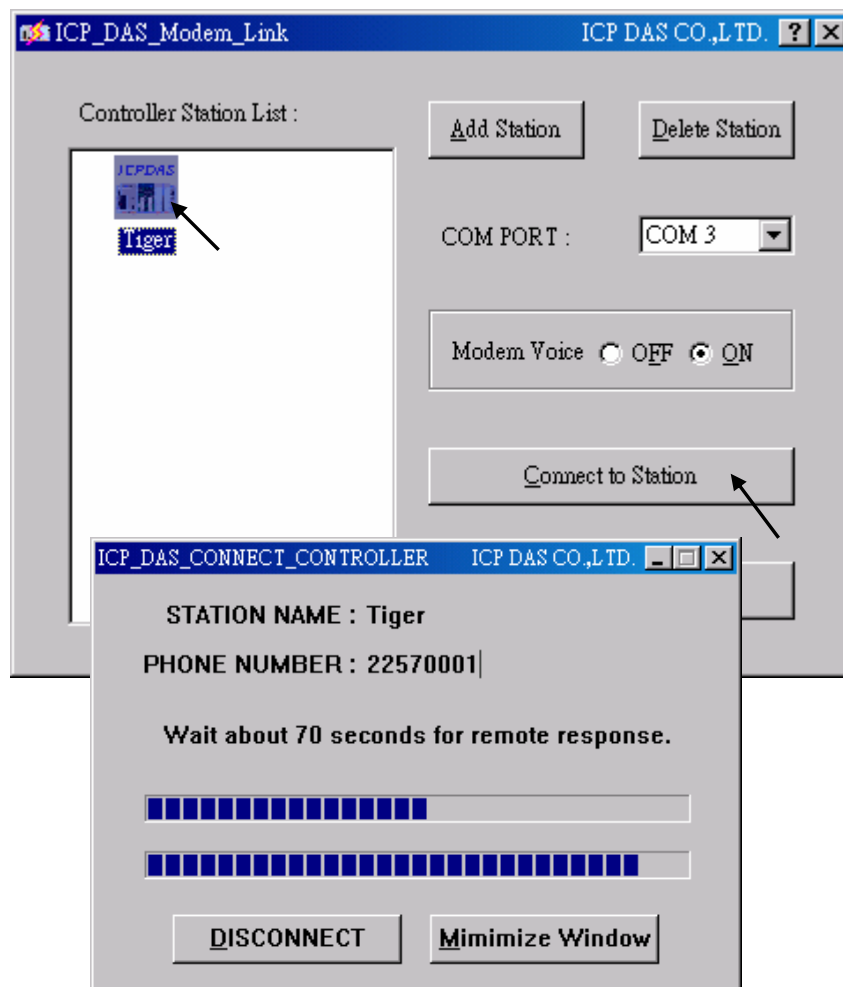
Then click on “debug”. Select the correct Com port of your PC which will dial the modem. And then click on “Add Station” to add a station if you have none.



Then you will see the below window. Given a name for this new station and the target phone No. If you add a “,” character inside the phone No. It will wait one second and then dial the rest No. For ex. Given No. as “9,,22570001” will dial “9” first, then wait 2 seconds and then dial “22570001”. The password must set to the same password of the “modem station” controller.



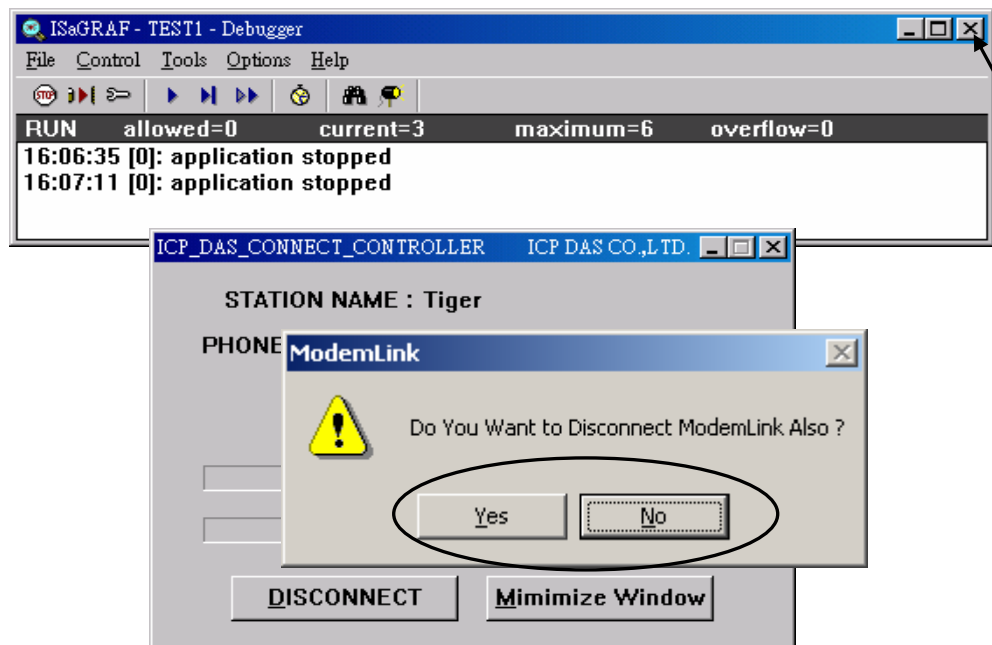
Click on the station you would like to connect first and then click on “Connect to Station” to command the modem dialing to the faraway controller.



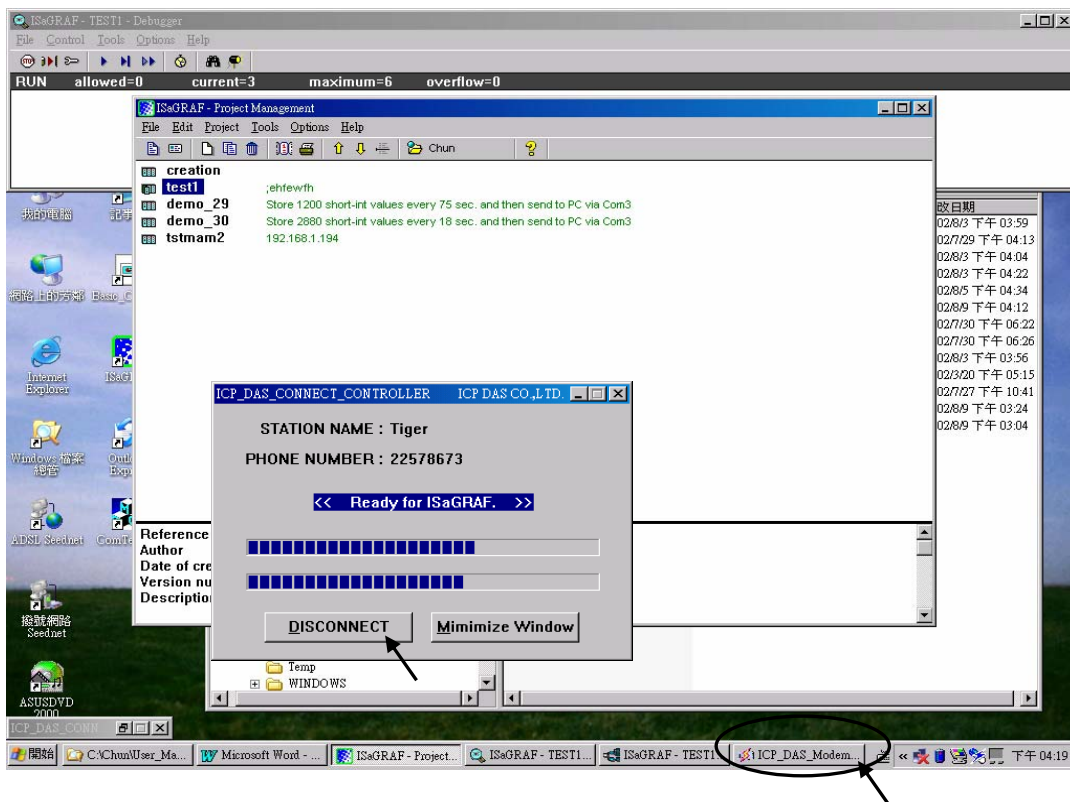
After the connection is Ok. You can download, monitor and change the variable value just like you did when the controller is near beside you.

To disconnect from the target controller, close the “ ... Debugger” window. Then you can choose “No” to keep the phone connected, or “Yes “ to hang off phone.

If you choose to keep the phone connected, you can open another ISaGRAF project to directly connect to another faraway target. The modem won't dial again.

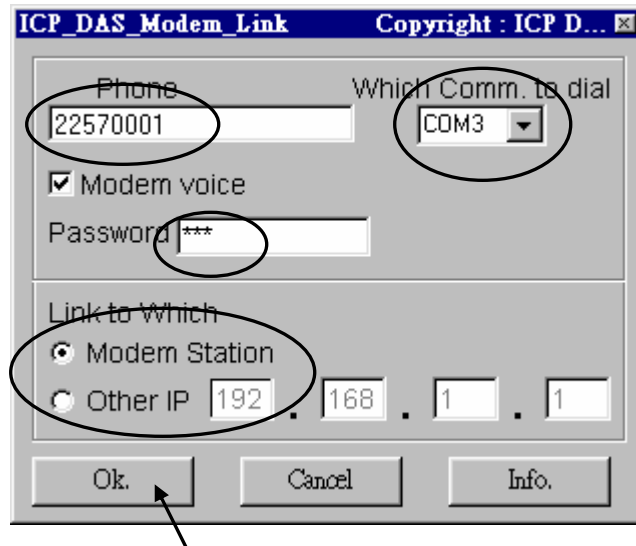


However, keep in mind, remember to disconnect the modem\_link when you finish your work, don't waste the money to the telecom company.

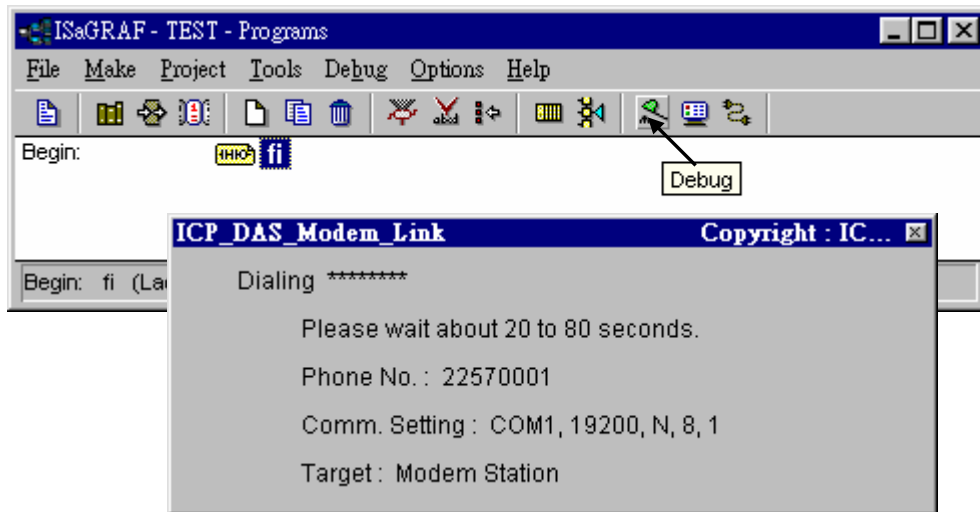


### For windows 95 & 98 users:

Given the correct target phone No. and the correct Com port of your PC which will dial the modem. If you add a “,” character indise the phone No. It will wait one second and then dial the rest No. For ex. Given No. as “9,,22570001” will dial “9” first, then wait 2 seconds and then dial “22570001”. The password must set to the same password of the “modem station” controller. If you are going to connect the “Modem station” controller, check “Modem station”, otherwise check “Other IP”. “Other IP” means the target controller is not connect to a modem however connect to the “Modem station” controller via an ethernet cable, the IP address has to assign.



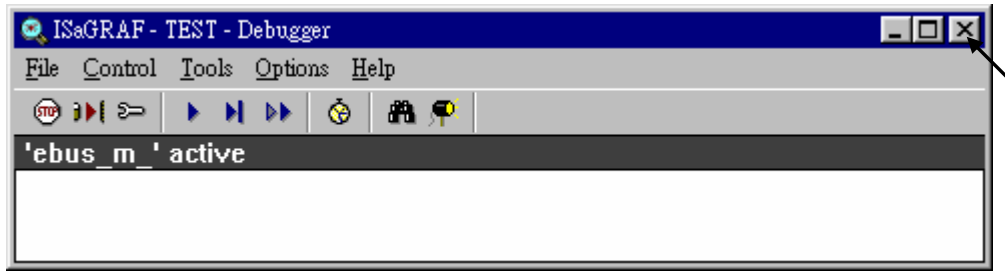
Then click on “debug” to start dialing the modem to connect to the faraway controller.



After the connection is Ok., you can download a new program, monitor the variable status just like you did when the controller is near beside you.

When you close the “ ... Debugger” window, the PC will command the modem to hang off the phone and disconnect with the faraway controller.





**Note:**

The Modem\_Link software installed on windows 95 & 98 doesn't support "keep the phone connected" function. That means each time you close the " ... Debugger" window, the phone will be hanged off too. So next time when click on "debug", you will see the modem dialing again to conect to the faraway controller.

For Windows NT, 2000 and XP users, the modem will not dial if you keep the phone connected.

---

## Chapter 14. Spotlight : Simple HMI

---

Spotlight is a simple HMI coming with ISaGRAF which allows user to build **Boolean Icon, Bar Graph, Trend Curve, Value Text, Bitmap Picture** to make application more friendly.

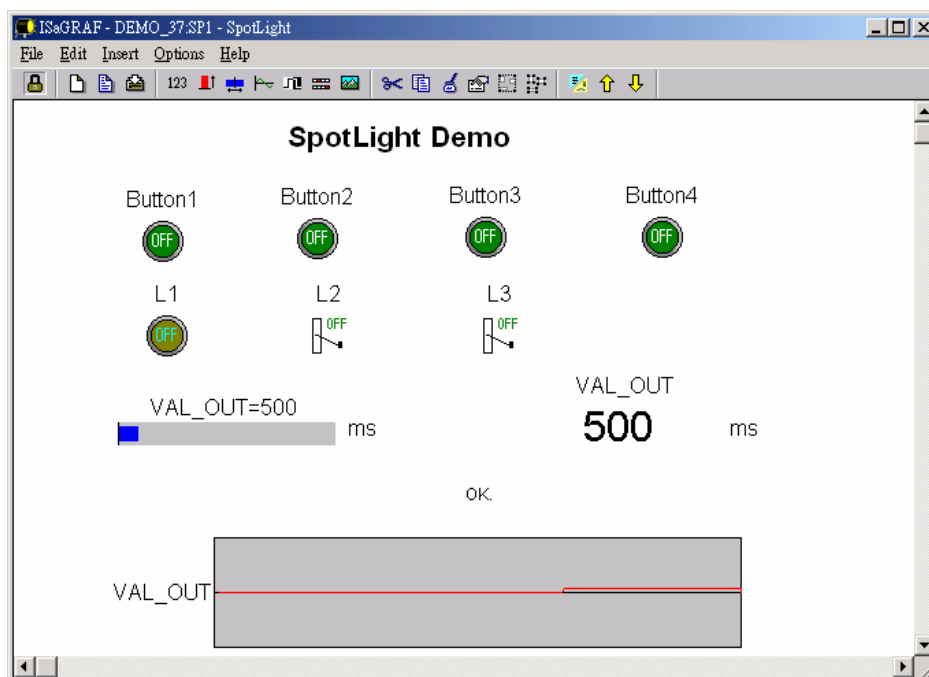
### 14.1 A Spotlight Example:

This Demo example can be restored from the ICP DAS's I-8000 CD-ROM - "demo\_37" (For I-8xx7). Please refer to Chapter 11 to restore it.

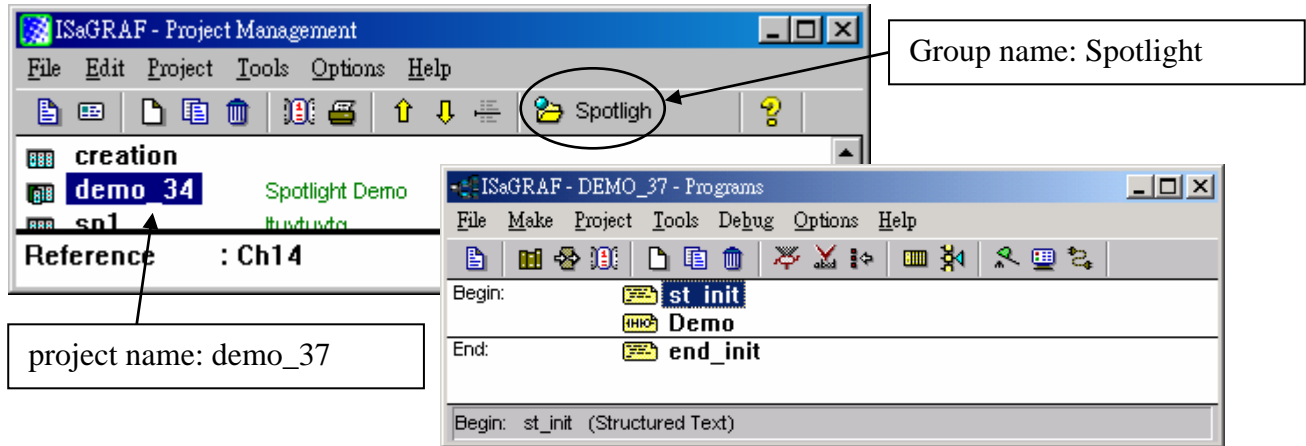
**Variables used In the example:**

| Name        | Type    | Attribute | Description  |
|-------------|---------|-----------|--|
| INIT        | Boolean | Internal  | Only = TRUE at the 1st scan cycle, <b>INIT value is TRUE</b>         |
| L1          | Boolean | Output    | Output 1, connect to Ch1 of "show3led"                               |
| L2          | Boolean | Output    | Output 2, connect to Ch2 of "show3led"                               |
| L3          | Boolean | Output    | Output 3, connect to Ch3 of "show3led"                               |
| Button1     | Boolean | Inpput    | Input 1, connect to Ch1 of "push4key"                                |
| Button2     | Boolean | Inpput    | Input 2, connect to Ch2 of "push4key"                                |
| Button3     | Boolean | Inpput    | Input 3, connect to Ch3 of "push4key"                                |
| Button4     | Boolean | Inpput    | Input 4, connect to Ch4 of "push4key"                                |
| VAL_OUT     | Integer | Internal  | to set blinking period, <b>initial value is set at 500 (unit:ms)</b> |
| OLD_VAL_OUT | Integer | Internal  | Old value of VAL_OUT   |
| T1          | Timer   | Internal  | Time Period of blinking  |
| MSG1        | Message | Internal  | Status report, please set its Maximum Length to 48                   |

**HMI screen outline:**



Project architecture:



ST Program “st\_init” in the “Begin” area :

```
(* Do some init action *)
if INIT=TRUE then
  T1 := TMR(VAL_OUT) ; (* Convert integer:VAL_OUT to Timer:T1 in ms *)
  MSG1:='OK.' ;
  OLD_VAL_OUT := VAL_OUT ; (* init OLD value *)
end_if ;

(* if set a new value to VAL_OUT *)
if VAL_OUT <> OLD_VAL_OUT then

  (* VAL_OUT is acceptable *)
  if (VAL_OUT>=200) & (VAL_OUT<=5000) then
    T1 := TMR(VAL_OUT) ; (* Convert integer:VAL_OUT to Timer:T1 in ms *)
    MSG1 := 'OK.' ;
  else (* VAL_OUT out of range *)
    MSG1 := 'VAL_OUT should be between 200 and 5000 .' ;
  end_if ;

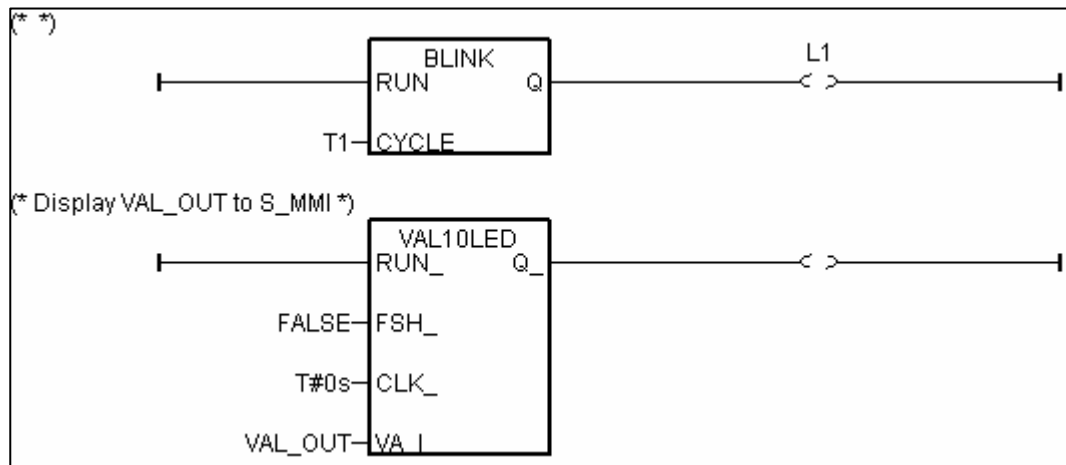
  OLD_VAL_OUT := VAL_OUT ; (* update OLD value *)

end_if ;
```

ST Program “end\_init” in the “End” area :

```
INIT := FALSE ;
```

LD Program “Demo” in the “Begin” area:



#### Operations :

The status of four push buttons will be displayed on the HMI screen

The first output will be blinking with the period defined by “VAL\_OUT” in ms

Value of “VAL\_OUT” can be modified from the HMI screen

The second and third output “L2” & “L3” can be controlled by the HMI screen.

The Value of “VAL\_OUT” will also be displayed on the front panel of the controller.

## Steps to build a Spotlight: HMI screen:

Complete this Demo project as described above.

After you finish it. Compile it to make sure there is no error.

Copy all files inside “ICO” folder to the associate directory of your project.

The “ICO” folder contains some boolean icon files already built by ICP DAS. They can be found from the ICP DAS’s CD-ROM : \napdos\isagraf\ICO\

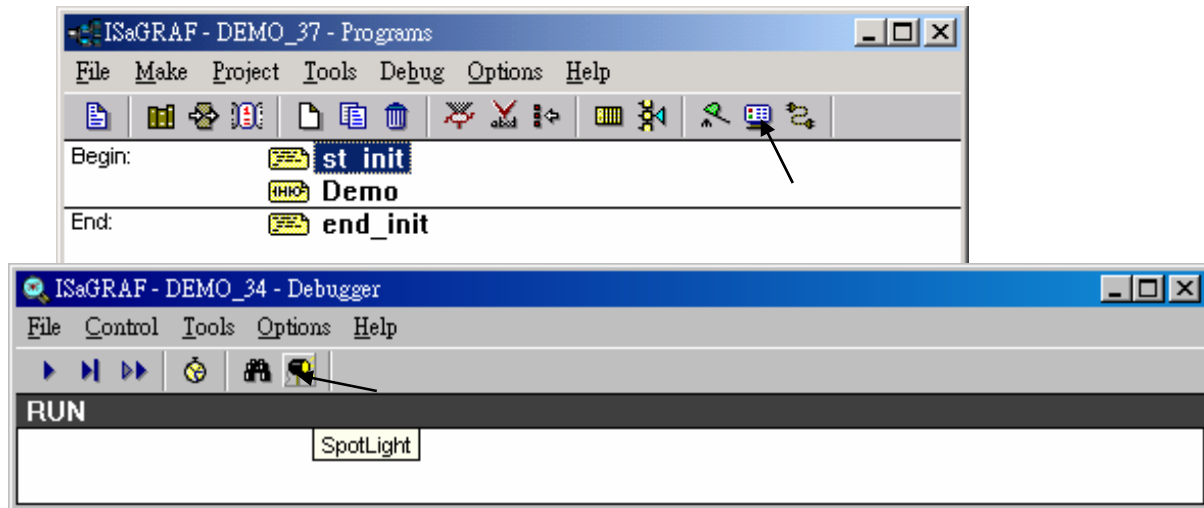
For example, this demo project is inside group “spotligh” and the project name is “demo\_37”, then copy CD-ROM: \napdos\isagraf\ICO\\*. \* to c:\isawin\spotligh\demo\_37\

If the “ICO” folder is not found in your CD-ROM. Please download it from the below site.

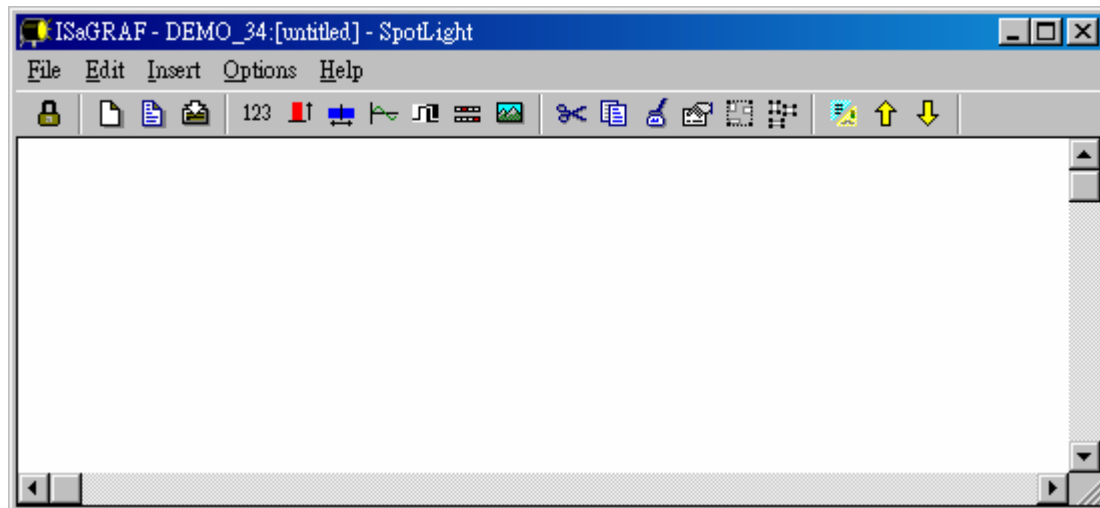
<ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/>

Get into the Spotlight editor.

Click on “Simulate”, then click on “Spotlight” to open spotlight editor.

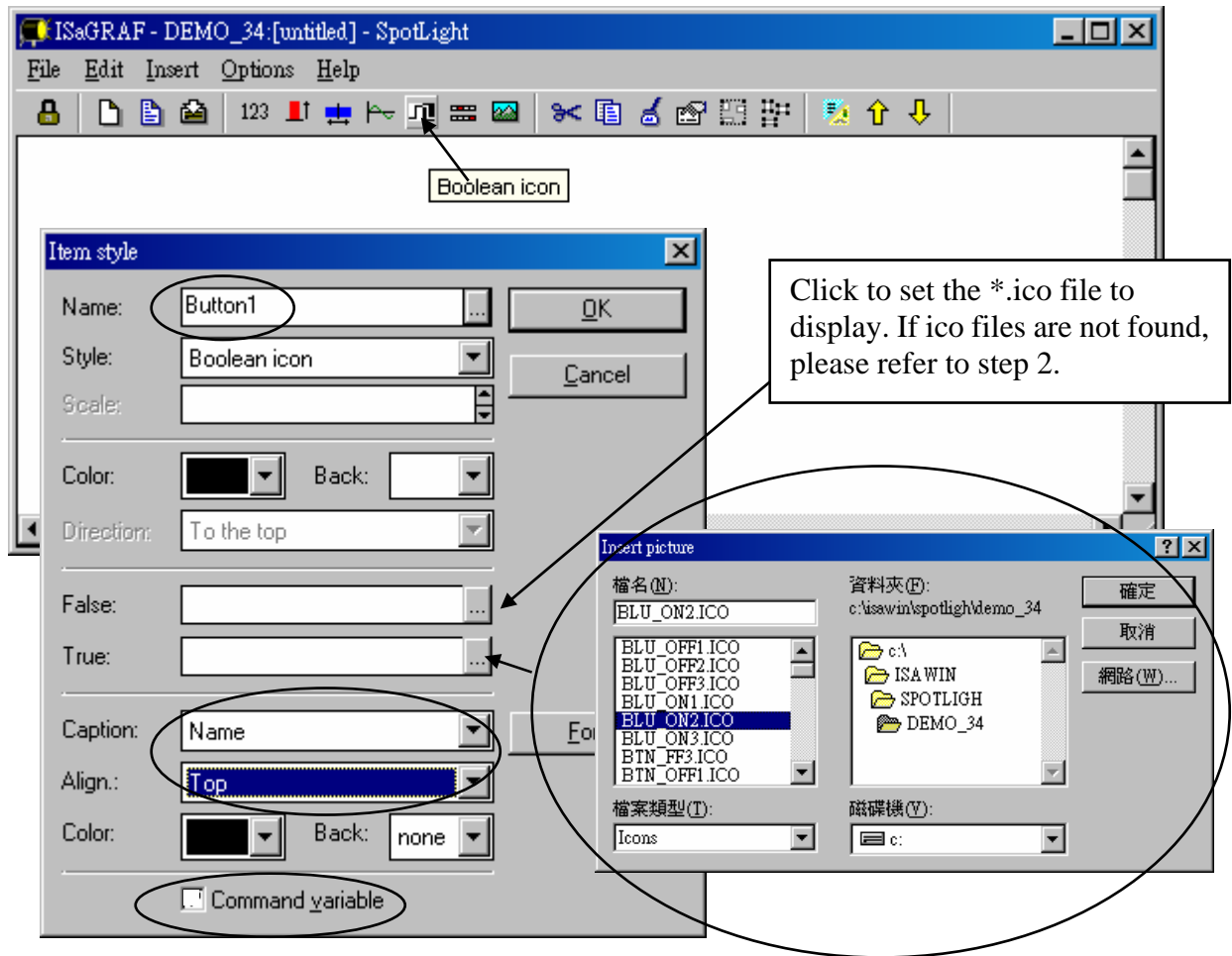


A “SpotLight” window will appear as below.

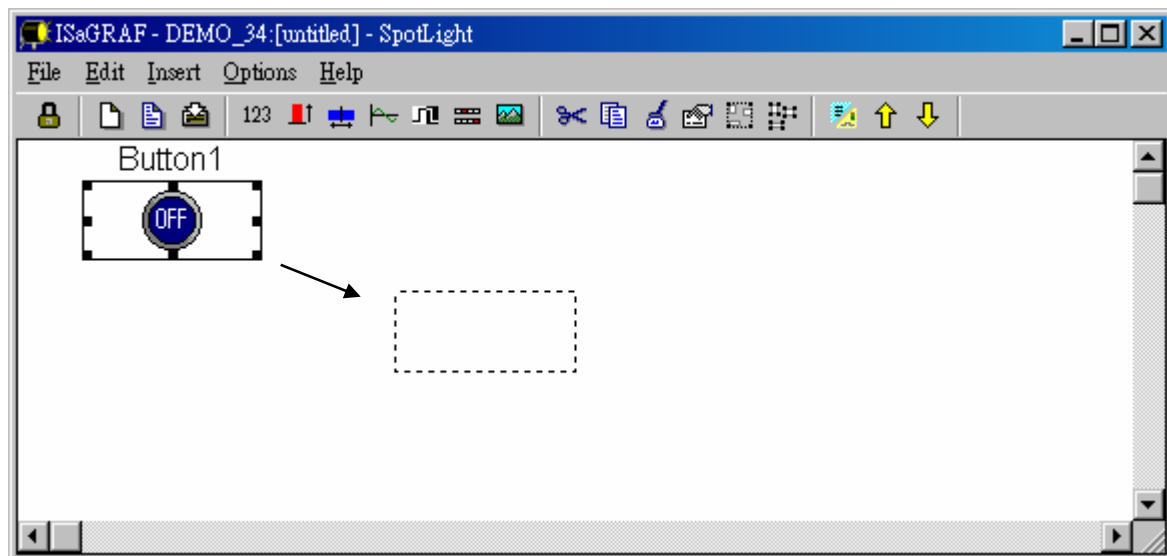


Add “boolean Icons”

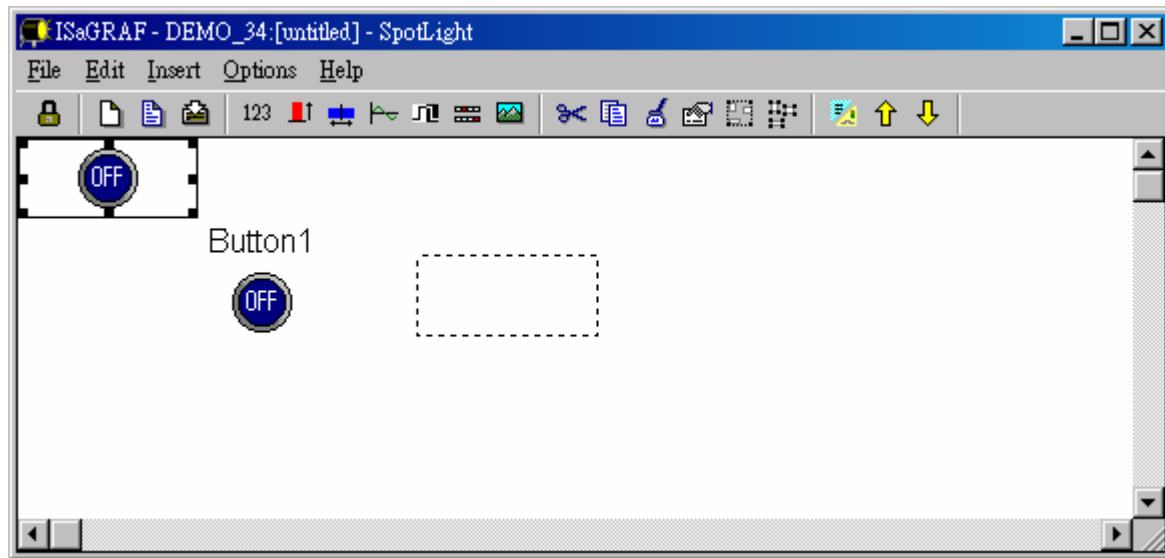
Click on “Boolean icon”, then set the associated Name as “Button1”, Caption as “Name”, Align as “Top” and then set the preferred \*.ico file to display with “FALSE” and “TRUE”, and un-check “Command variable”.



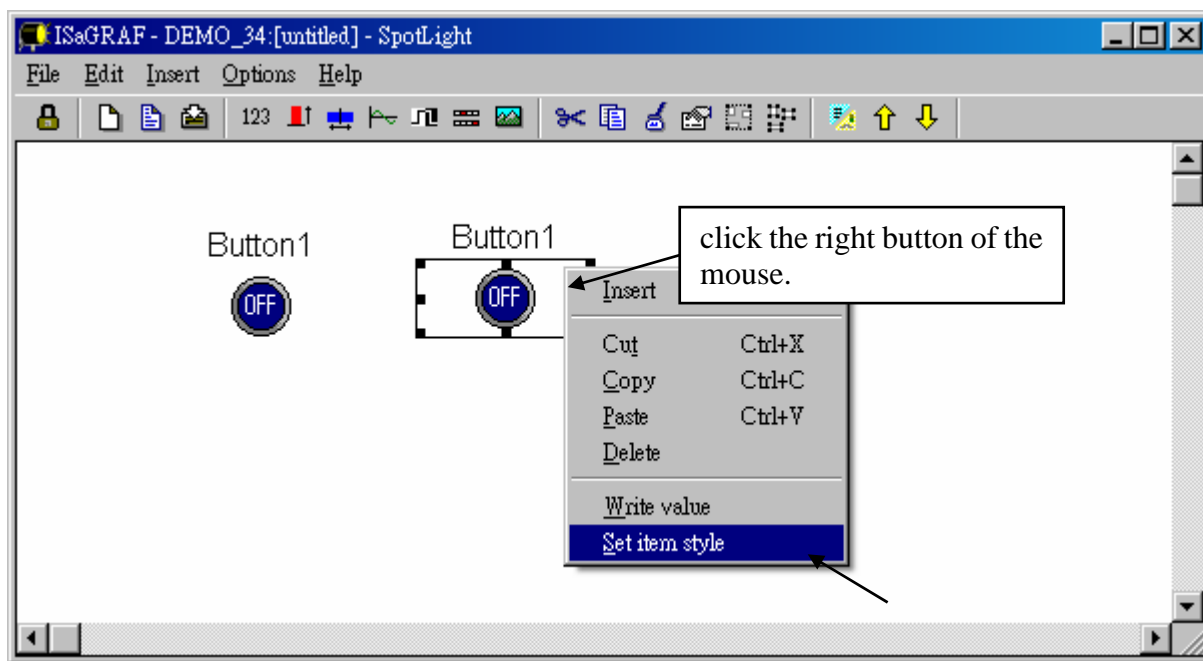
Then drag the boolean icon to appropriate place.



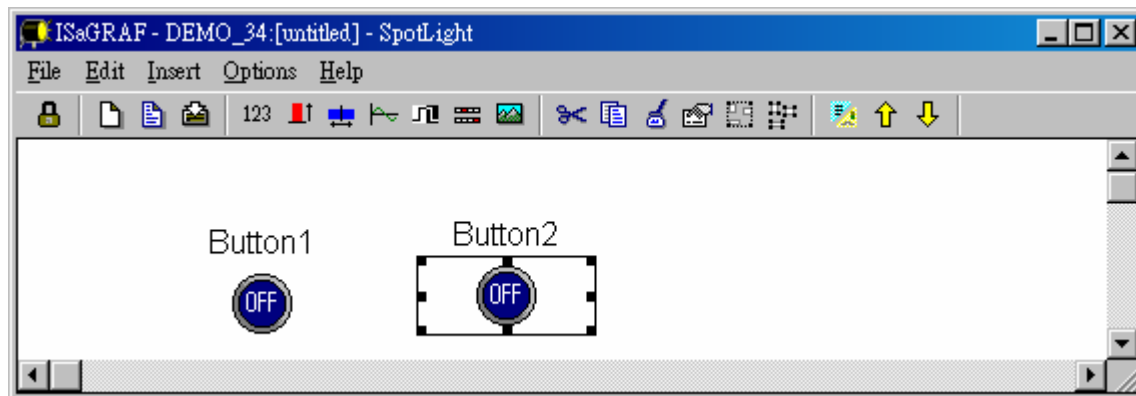
Check on the new created boolean icon, copy it(Ctrl+c) and then paste it (Ctrl+v) to reproduce one another boolean icon. Then drag it to the preferred place.



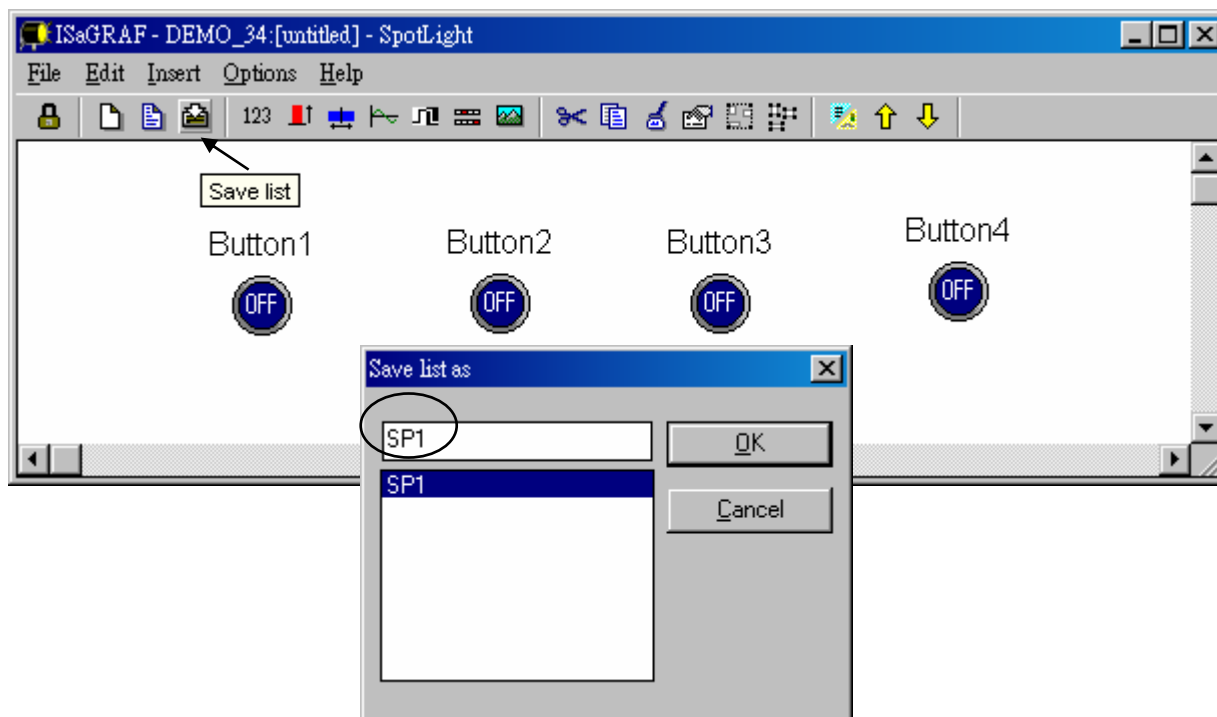
Check on the new created boolean icon, then click the right button of the mouse, select “Set item style” to modify the name to “Button2”.



Then we have ...

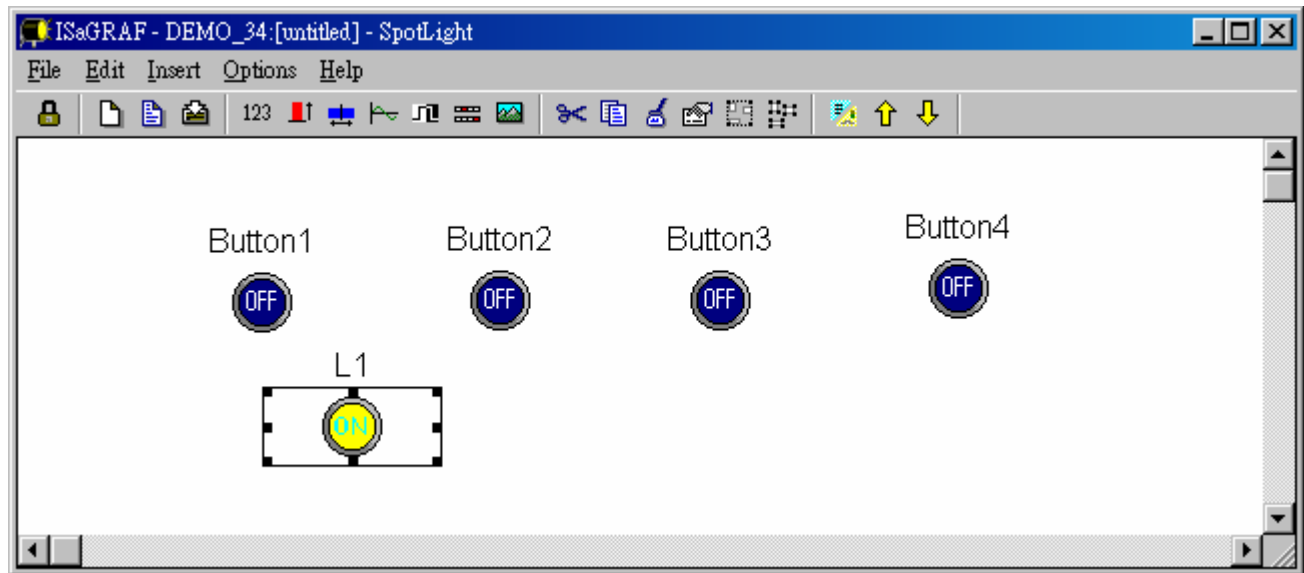


Follow the same method to create 4 boolean icons as below. Recommend to save it anytime for safety. Given a name to this screen.

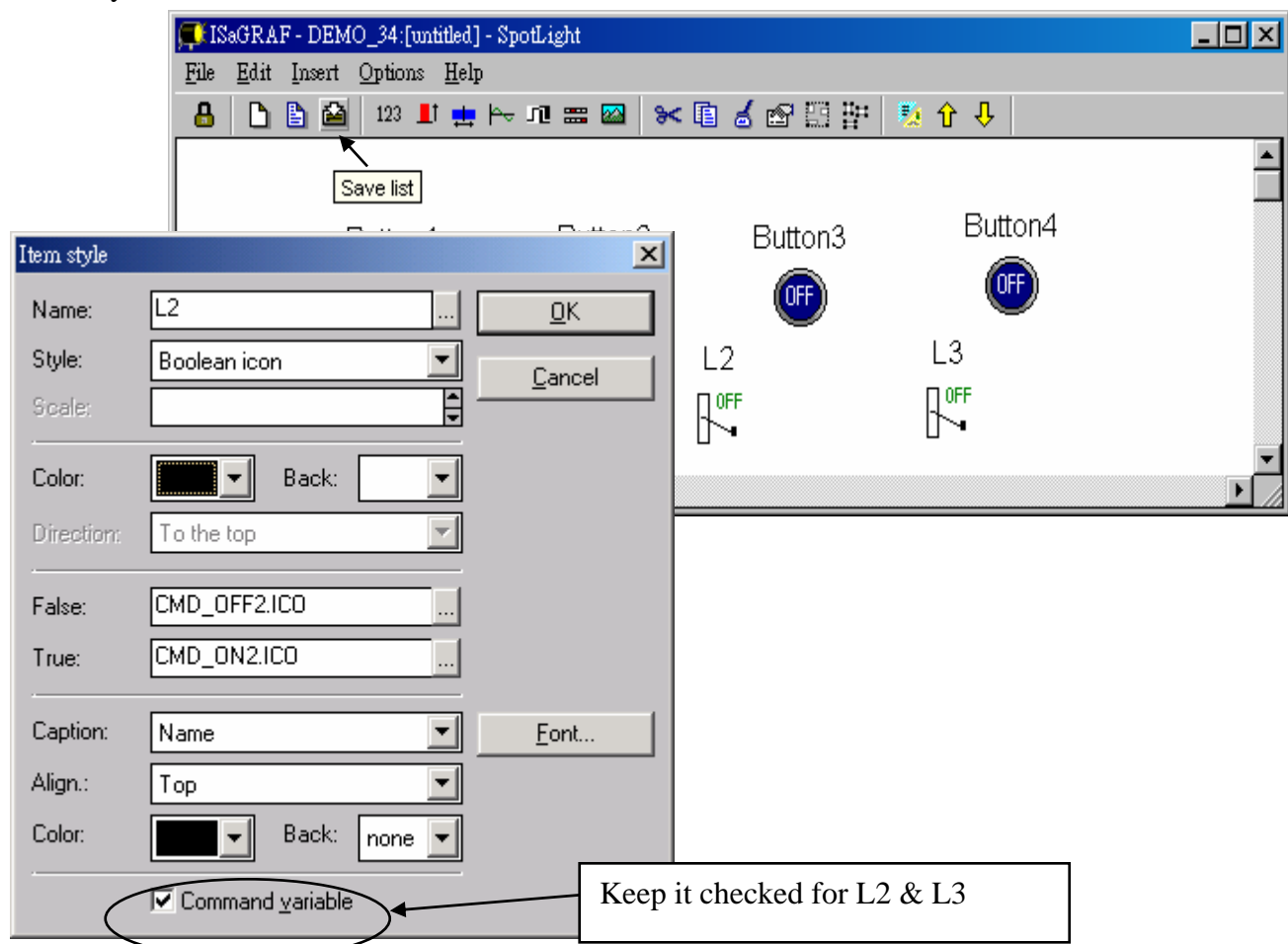




We need one another Boolean icon to display the status of “L1”. Create it with a different color (TRUE : “YEL\_ON2.ico”, FALSE : “YEL\_OFF2.ico” ).

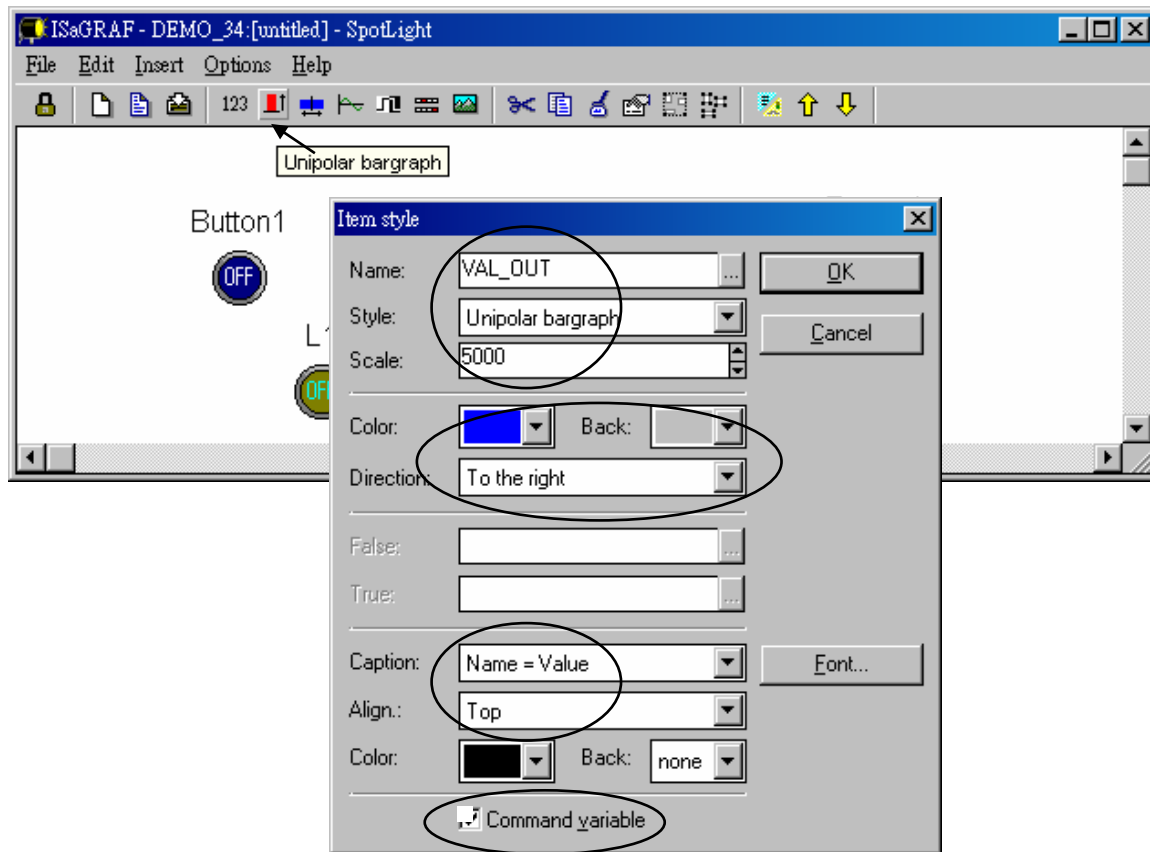


And then create L2 & L3 with TRUE:”CMD\_ON2.ico” and FLASE: “CMD\_OFF2.ico” as below. Save it anytime, **L2 & L3 should not un-check “Command variable”**.

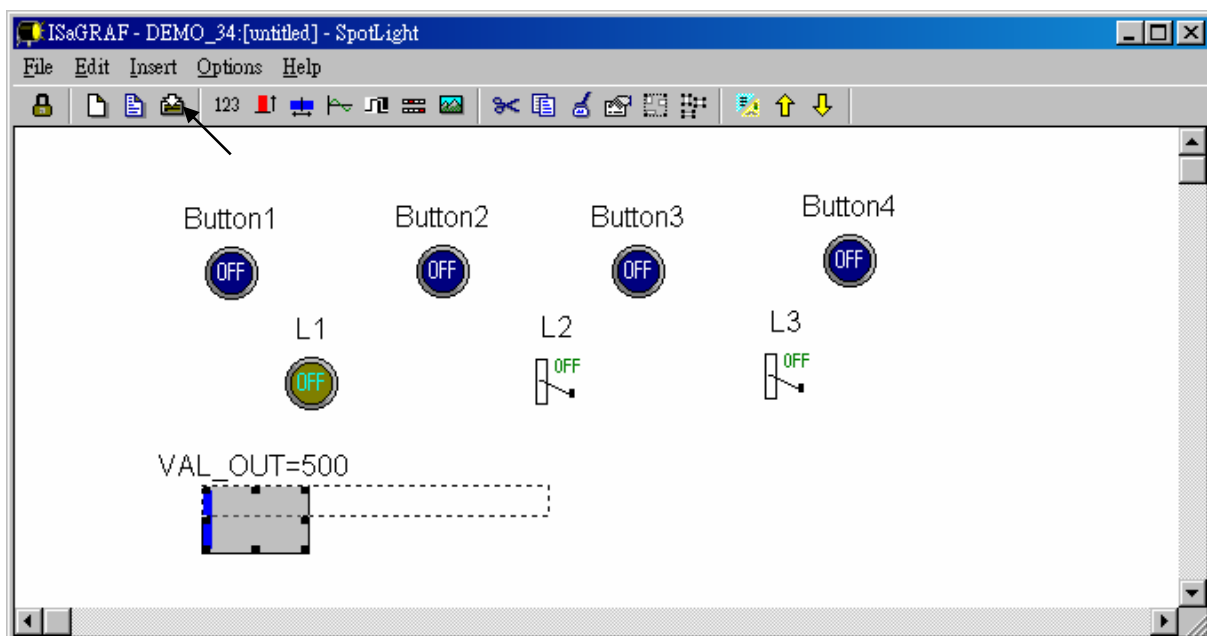


Add “Unipolar bargraph”

Click on “Unipolar bargraph”, set the associated Name as “VAL\_OUT”, Scale as “5000”, Color as blue, Back as gray, Direction as “To the right”, Caption as “Name=Value”, Align as “Top”, and un-check “Command variable”

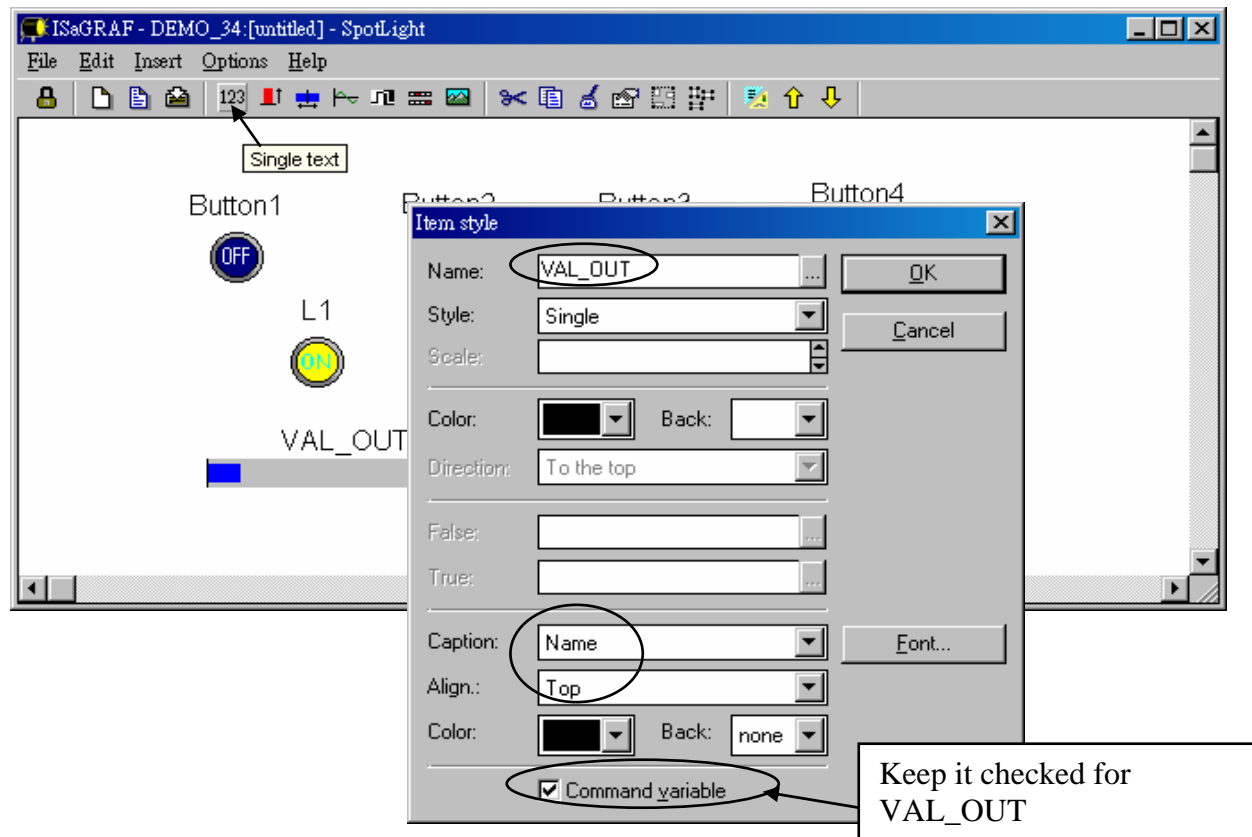


Click and hold on the left button of the mouse to change to the preferred shape as below. Save it anytime.

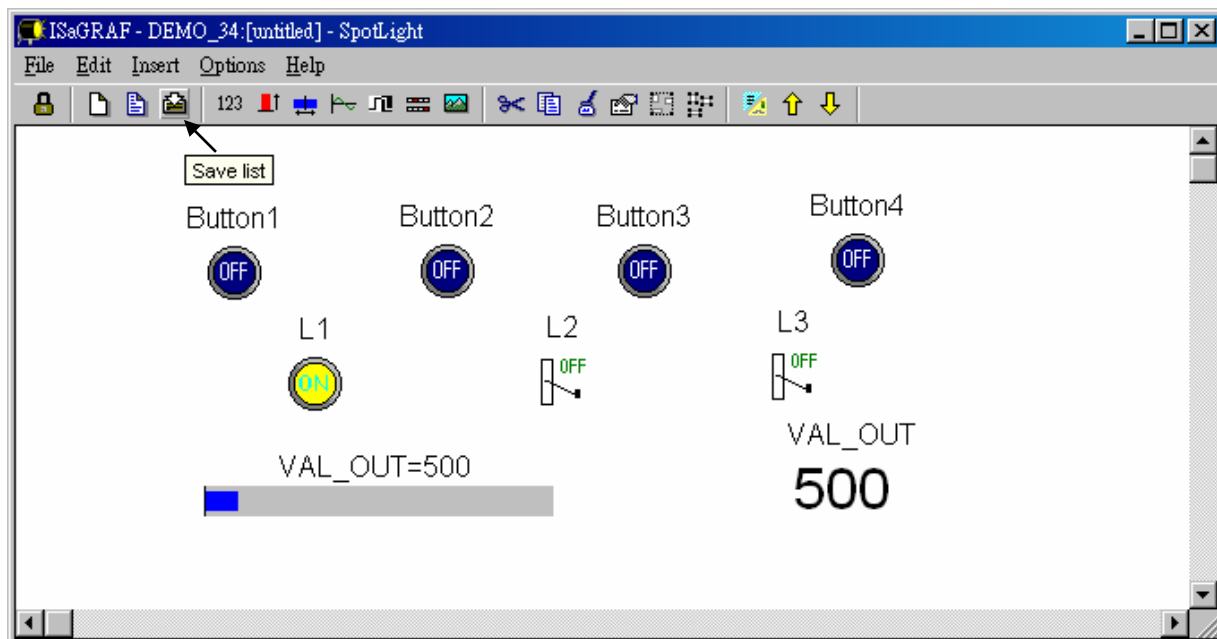


Add “Single text”

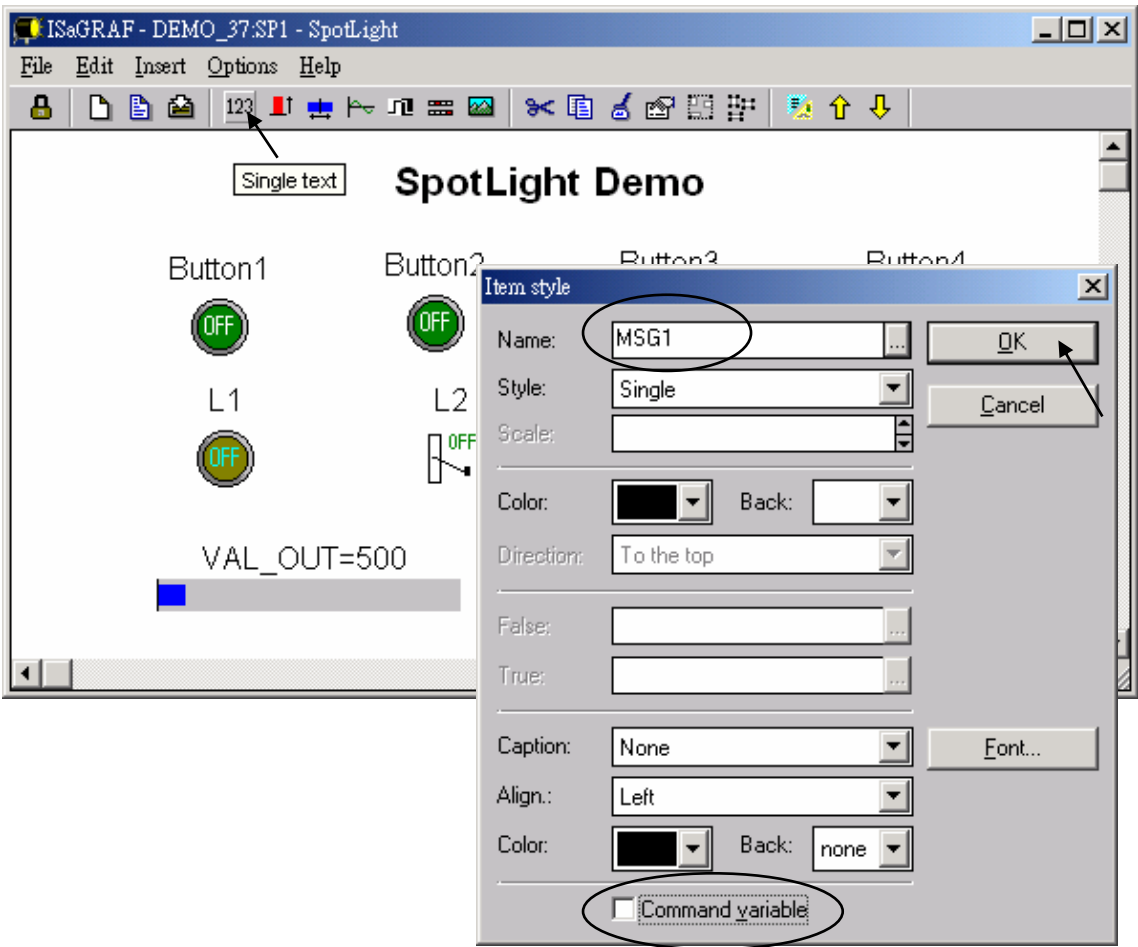
Click on “Single text”, set the associated Name as “VAL\_OUT”, Caption as “Name”, Align as “Top”



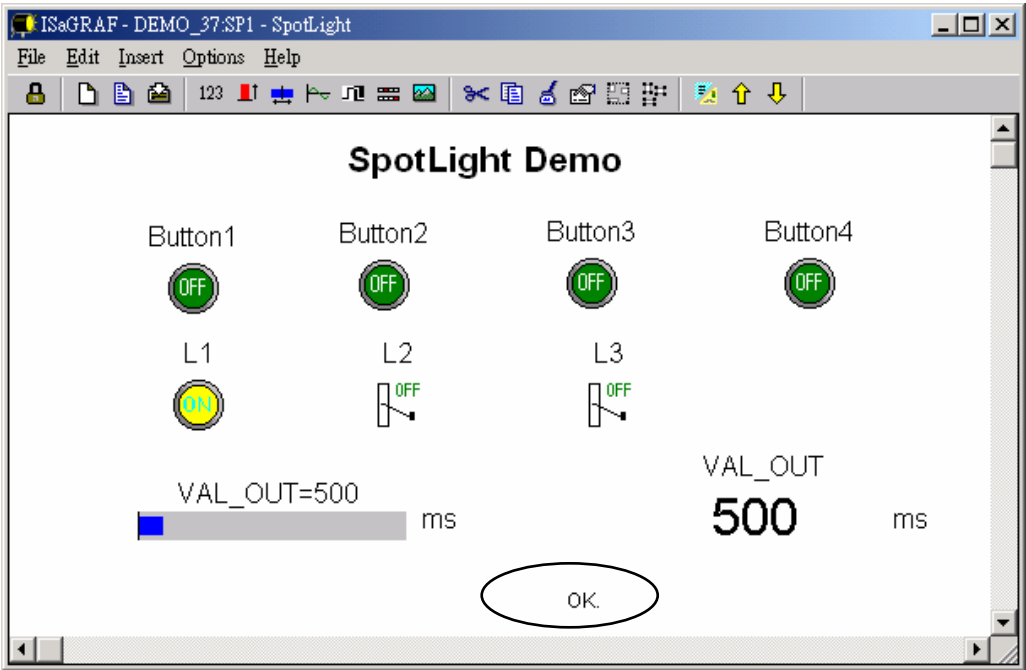
Move it to the preferred place and save it.



Click on “Single text” again, set the associated Name as “MSG1”, Caption as “None”, Align as “Left” and un-check “Command variable”.

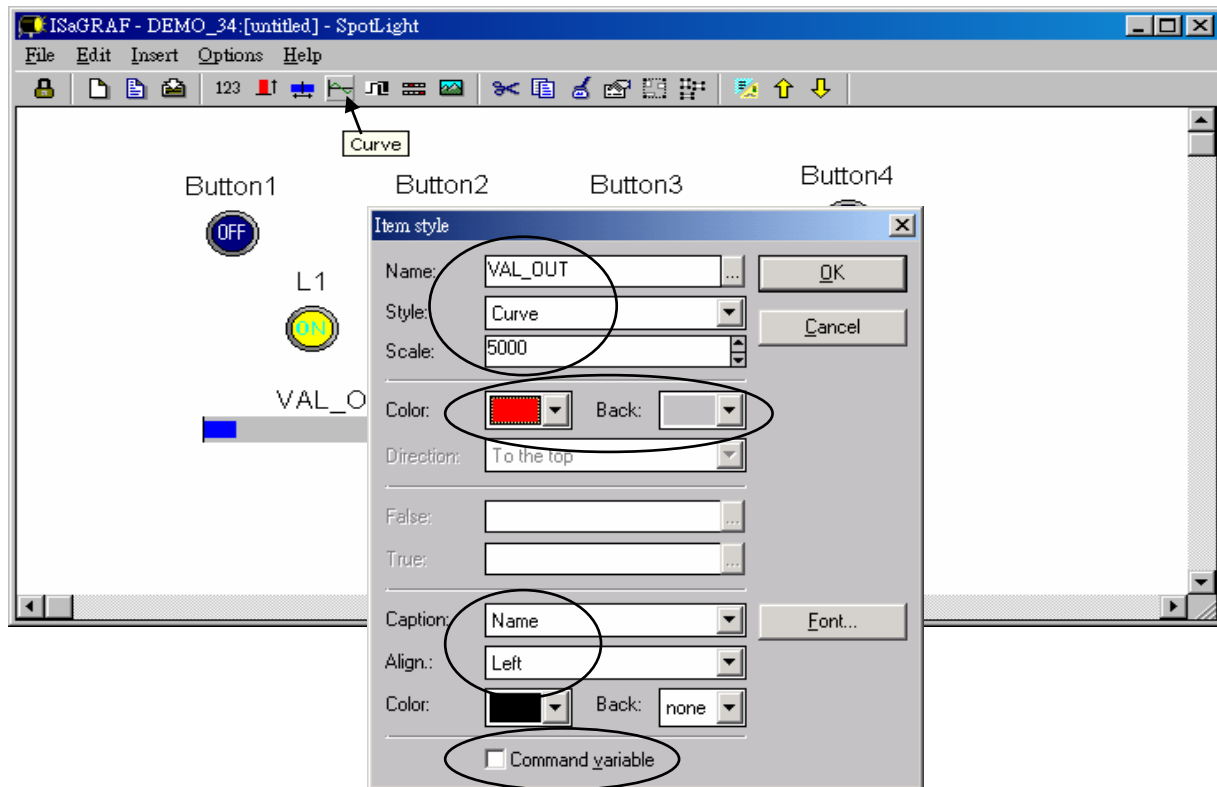


Move it to the preferred place and save it.

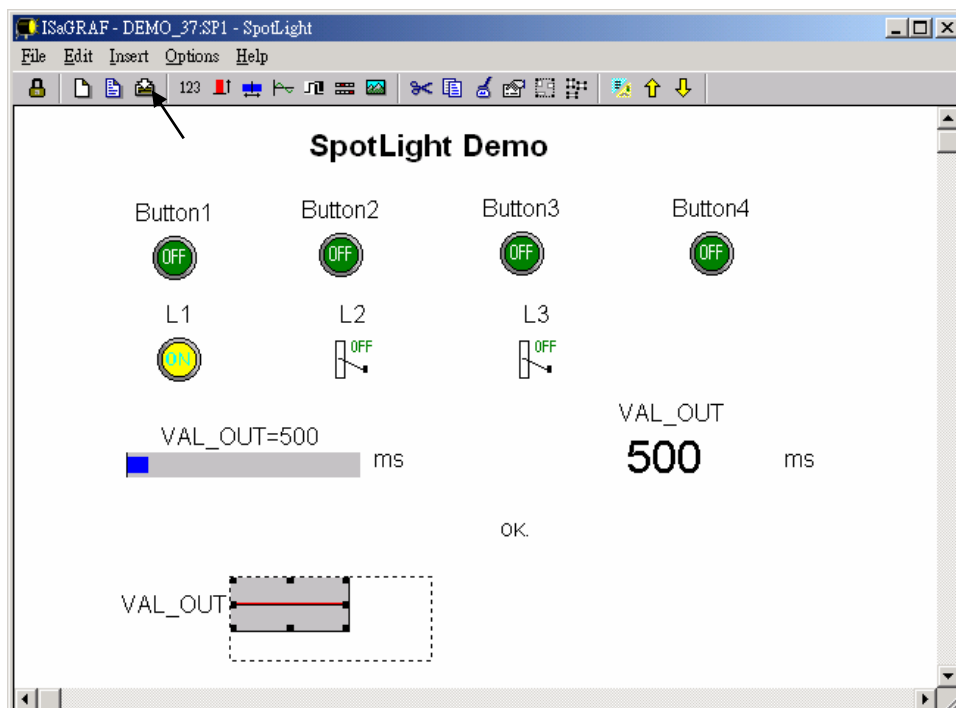


Add “Curve”

Click on “Curve”, set the associated Name as “VAL\_OUT”, Scale as “5000”, Color as red, Back as gray, Caption as “Name”, Align as “Top”, and un-check “Command variable”



Click and hold on the left button of the mouse to change to the preferred shape as below. Save it anytime

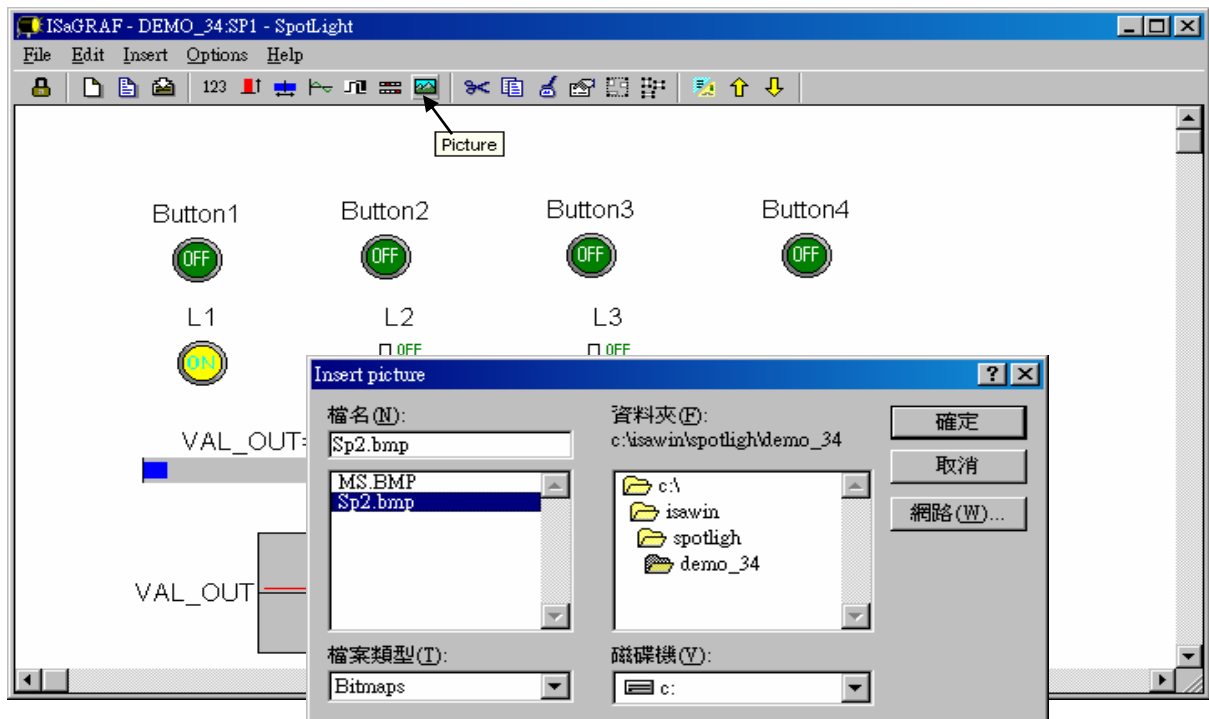


Add “picture”

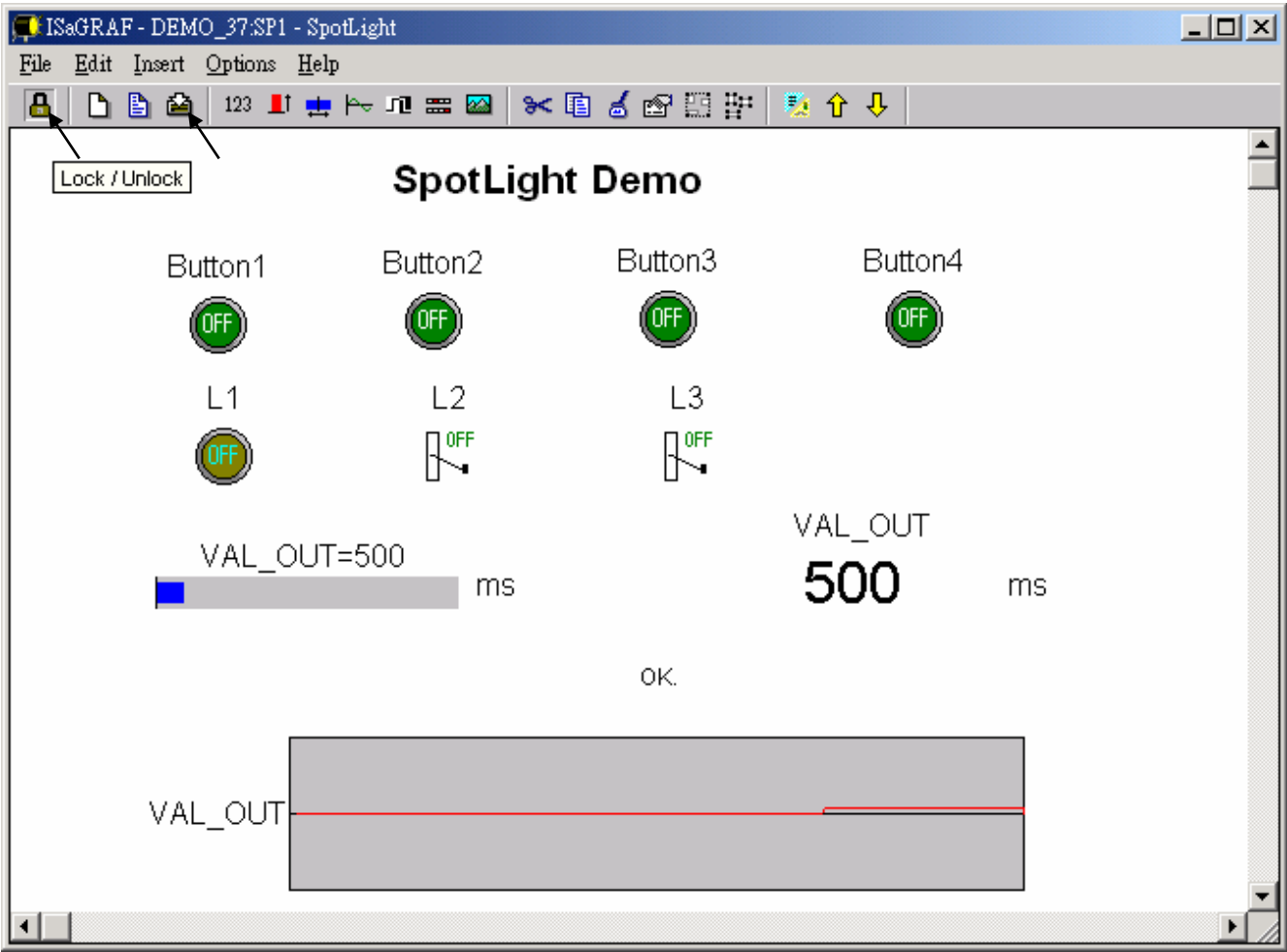
Please build 2 bitmap pictures by MS painter as below. Then save them respectively with file names of “sp2.bmp” & “ms.bmp” to the associate project directory. (For this example “c:\isawin\spotlighdemo\_37\”)



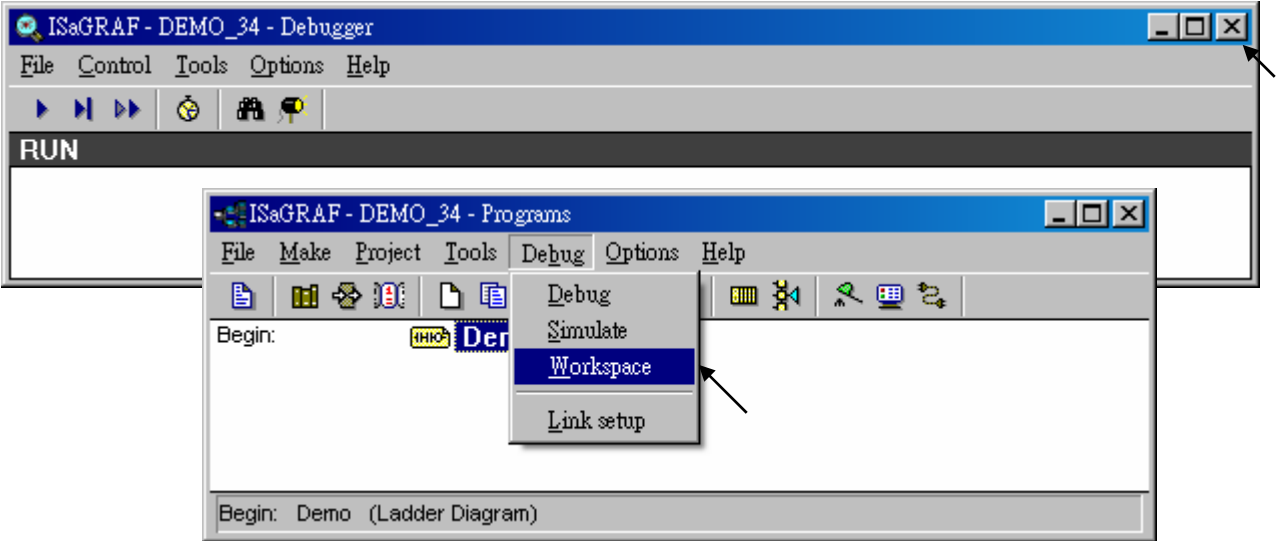
Click on “Picture”, Select the associate bmp file name.



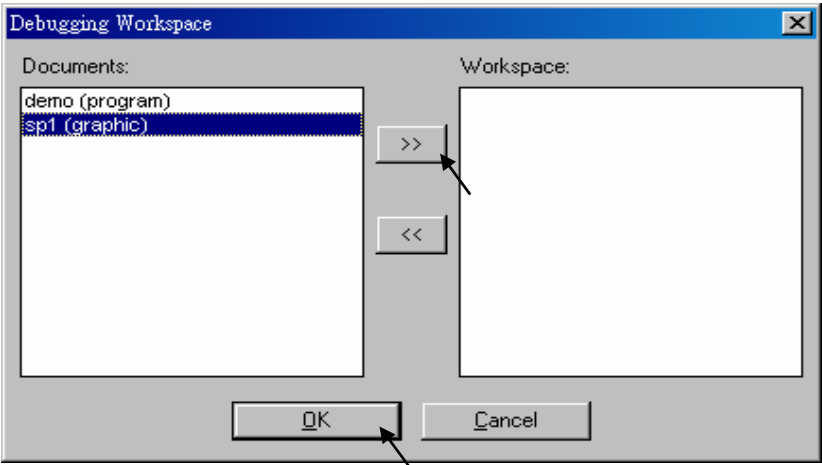
Add 2 pictures “sp2.bmp” and “ms.bmp” to the preferred place, then we got the below window. Click on “Lock” to protect it (No modification allowed). Save it anytime.



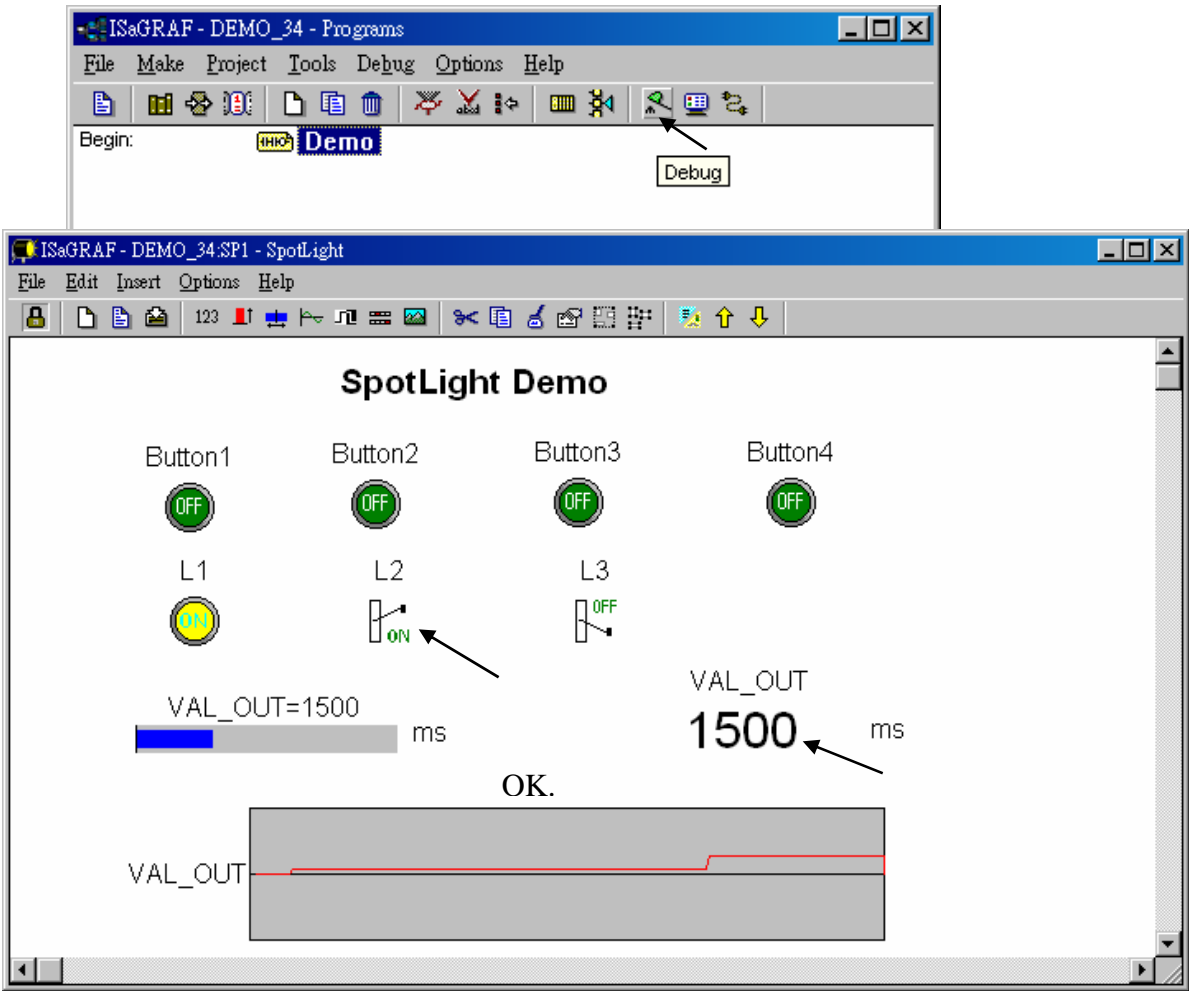
Add the HMI screen to the “Workspace”  
Quit “simulation”, then run “Debug”-“Workspace”.



Move the HMI screen to the right (Workspace).

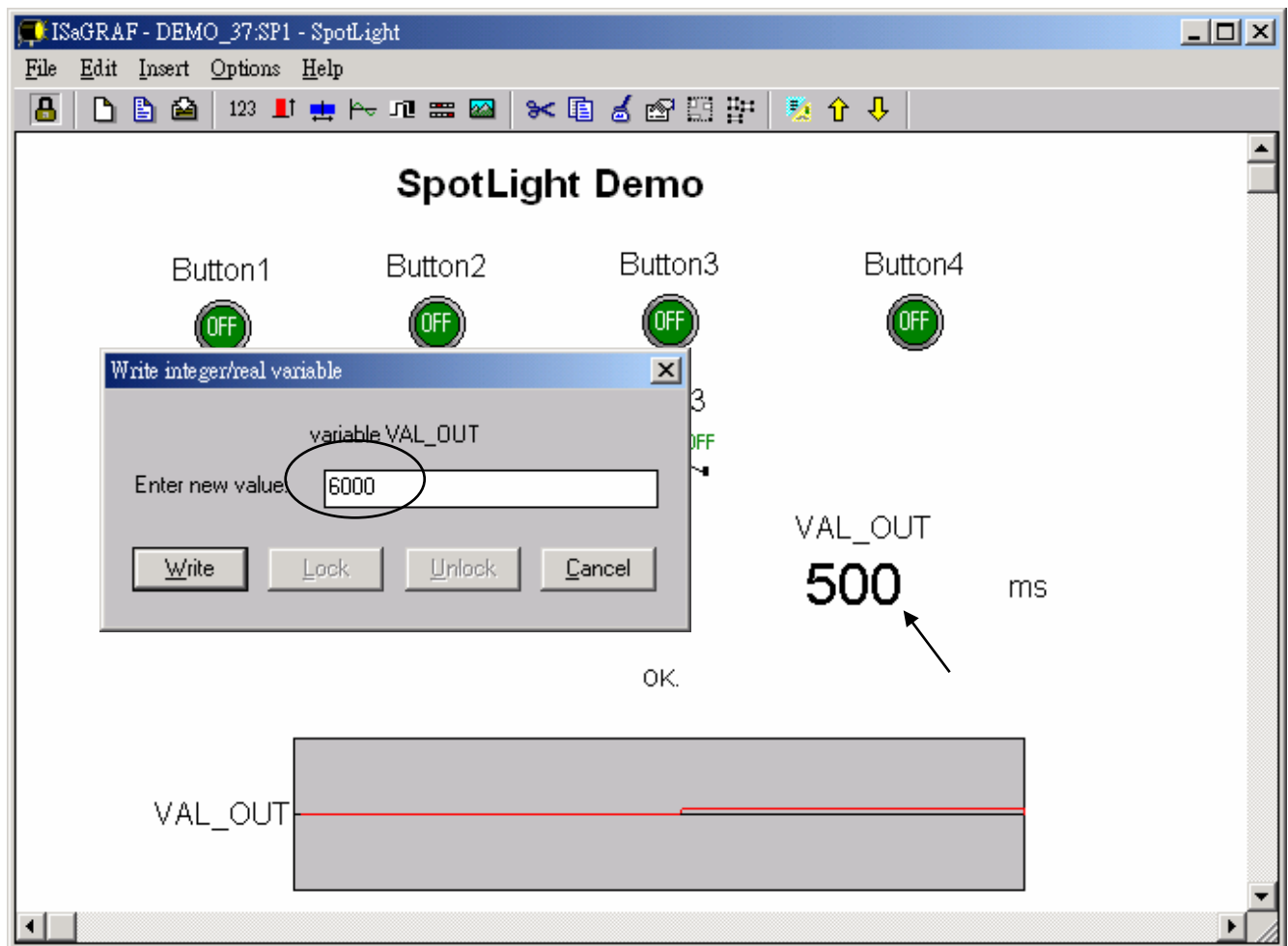


J. Time to download to the controller and test  
Click on “Debug” to download the project to the controller and test it. You may double click on “L2”, “L3” or “VAL\_OUT” to modify the value and see what it happens on the controller. And also you can press the 4 pushbuttons on the controller.

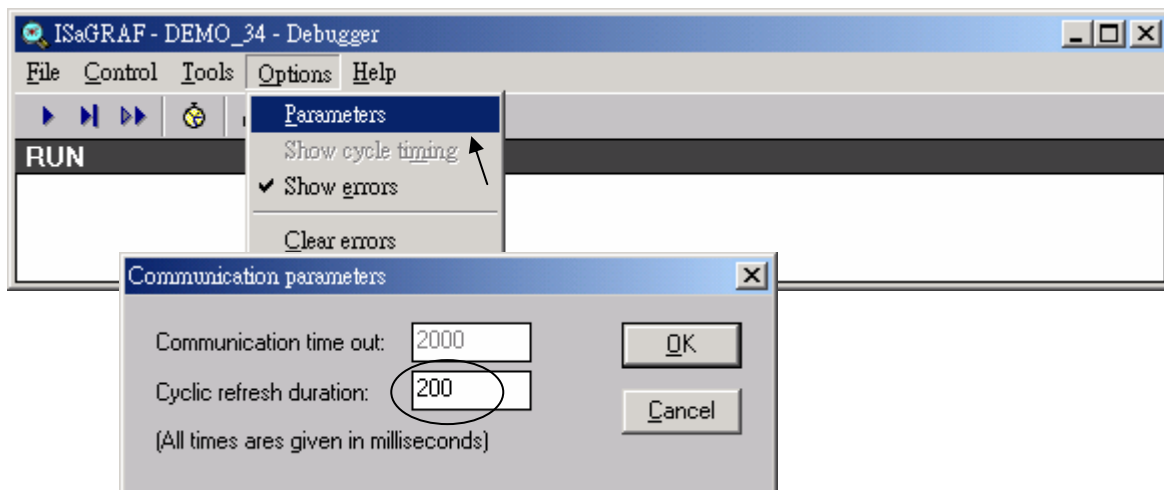




You may double click on “VAL\_OUT” and give a value large than 5000 to see what it happens.



**Note:** For quick response, user may click on “Options” – “Parameters”, and then set the “Cyclic refresh duration” to a smaller value. (Recommend not to set below 200 ms)



---

## Chapter 15. Creating User-Defined Functions

---

ISaGRAF supports functions written in ST, FBD, IL and QLD languages. User-defined functions are normally for some algorithm which has been used again and again.

A function always has a return value (output parameter) and its name should be the same name as the function, and may have up to 31 input parameters. The code written inside functions can not call any **function block**, however can call other ISaGRAF standard **functions** and **c functions** provided by ICP DAS.

We are going to create a function to save an integer value to the EEPROM. Its format is as below.

Function name :       W\_EEP

Description:               Save an integer to the EEPROM when its value changed

Input parameters:

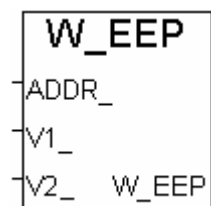
    ADDR\_ (integer) :   the address of the EEPROM to write

    V1\_ (integer) :     New value

    V2\_ (integer) :     Old value

Return parameter:

    W\_EEP (integer):    return the new value

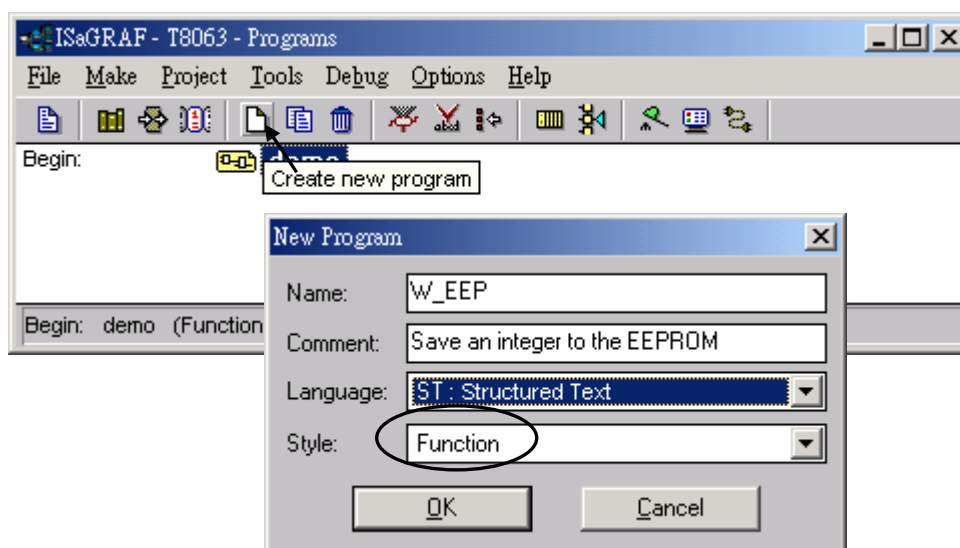


**Note:** The parameter names been used will become reserved names. That's why we use ADDR\_ , V1\_ , V2\_ rather than ADDR , V1 & V2.

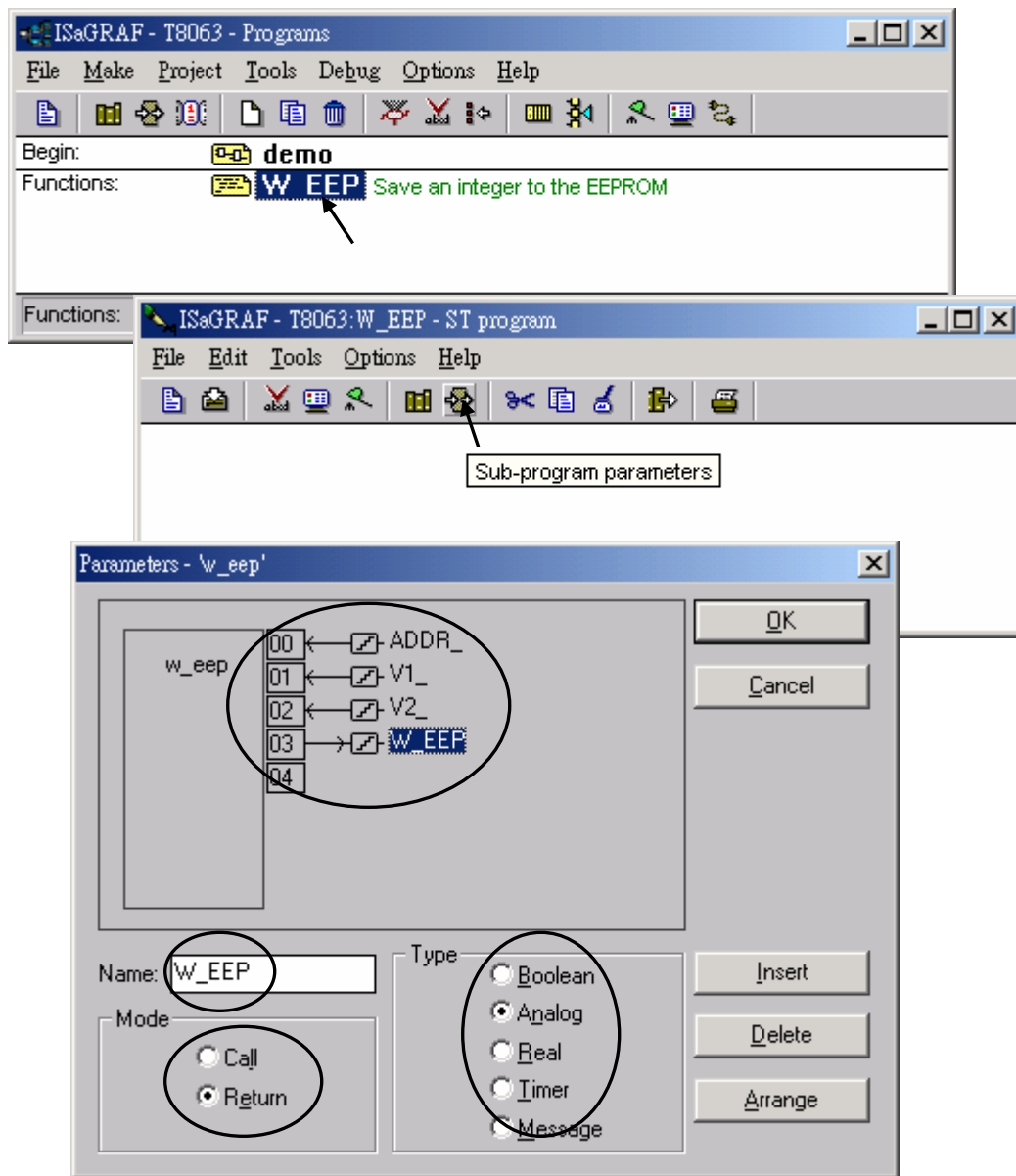
### 15.1: Creating functions inside one project

Functions created inside one project can be only called by other programs written in the same project.

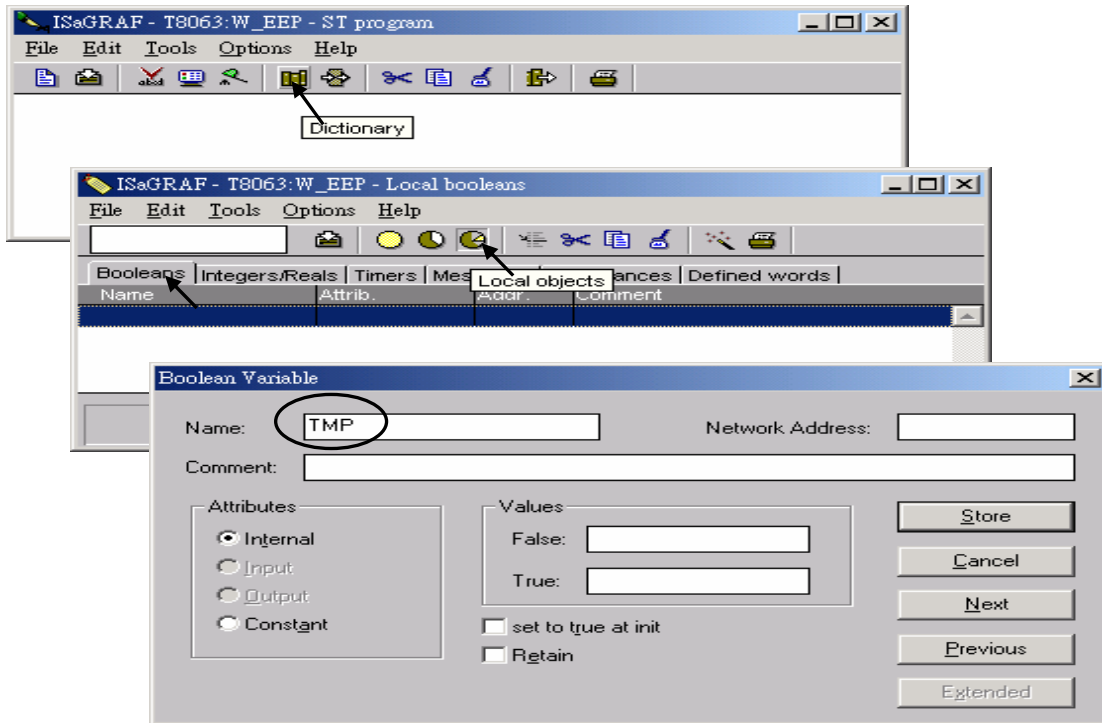
A. Click on “Create new program” inside the project. Given Name as “W\_EEP”, Language as “ST:...”, Style as “Function”.



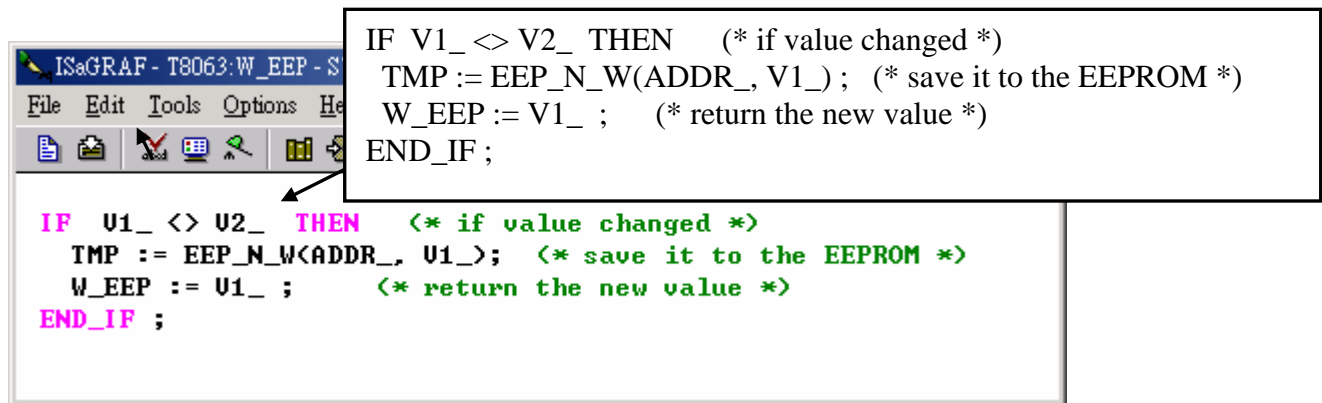
- B.** Double click on the function to get into it. Then click on “Sub-program parameters” to define input and output parameters.



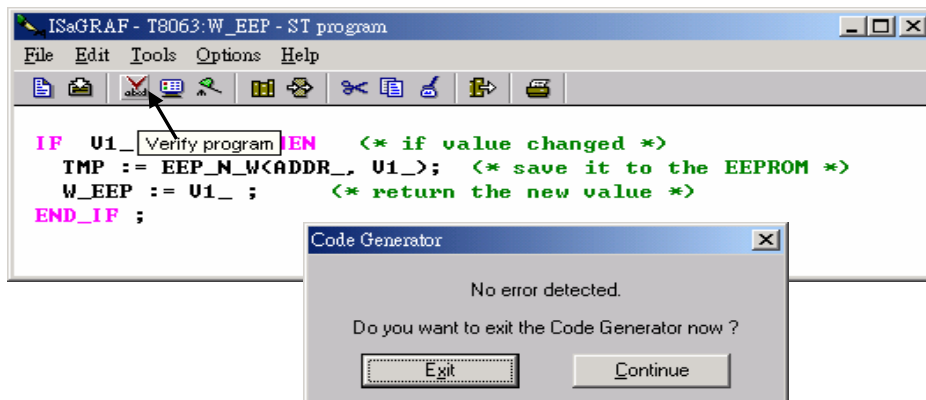
C. Declare local variables. We need a local **boolean internal** variable “TMP” in this example.



D. Enter function codes.



E. Verify the function.

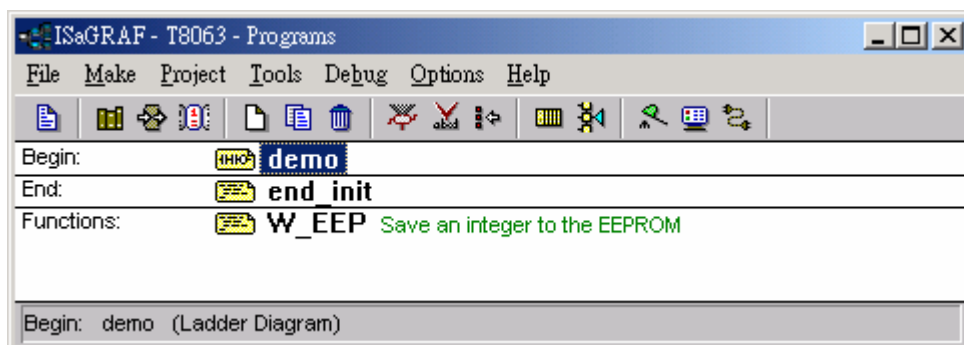


F. Call it in other programs in the same project.

Global variables used in the project:

| Name    | Type    | Attribute | Description   |
|---------|---------|-----------|---|
| INIT    | Boolean | Internal  | initial value at “TRUE”. TRUE means 1 <sup>st</sup> scan cycle      |
| K1      | Boolean | Input     | Connect to 1 <sup>st</sup> ch. Of “push4key”, press it to get “Val” |
| New_Val | Integer | Internal  | New value wish to save to the EEPROM                                |
| Old_Val | Integer | Internal  | Old value   |
| Val     | Integer | Internal  | Read back value of the EEPROM                                       |

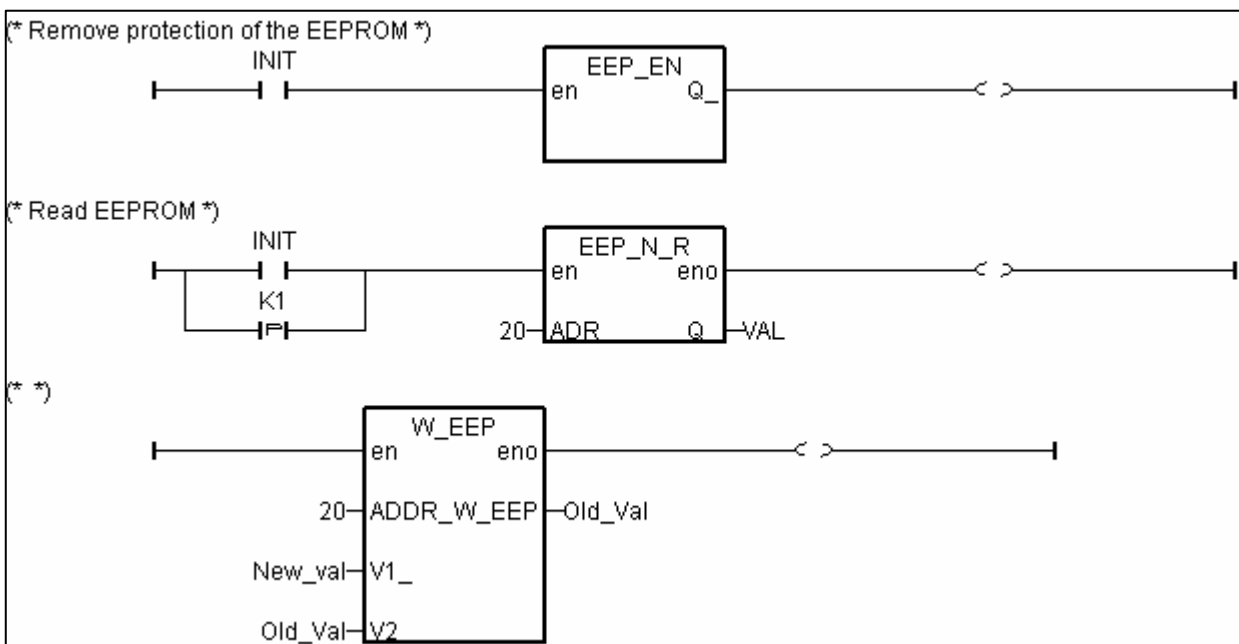
Project architecture:



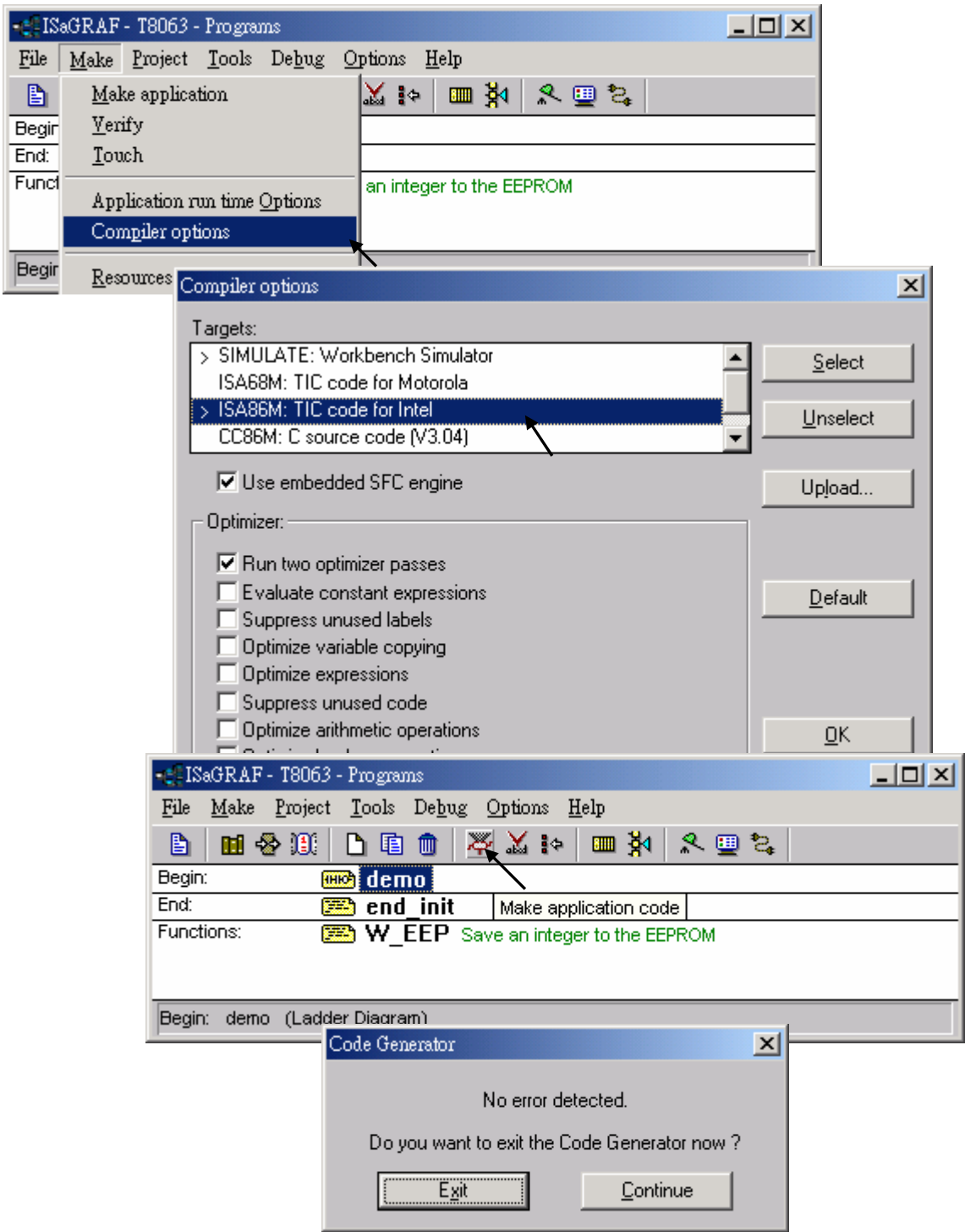
ST program – “end\_init” in the “End” area :

```
IF INIT=TRUE THEN
    INIT := FALSE ;
END_IF ;
```

LD program – “demo” :



G. Set Compiler Options and compile the project.



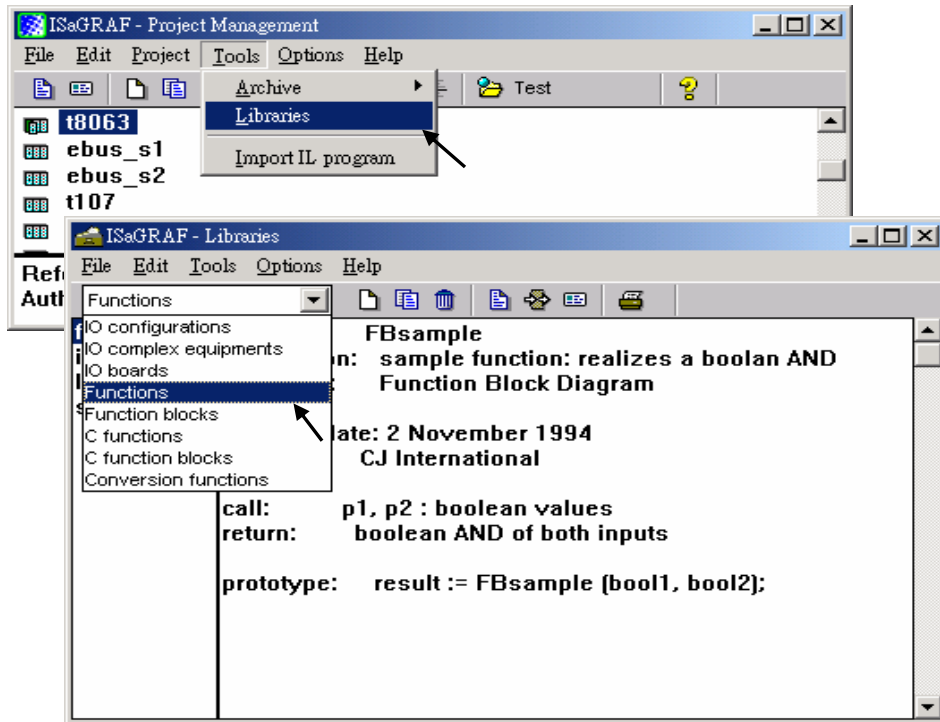
After download to the controller, you may change the “New\_Val”, and then press “K1” to see what it happens.

## 15.2: Creating functions in the library

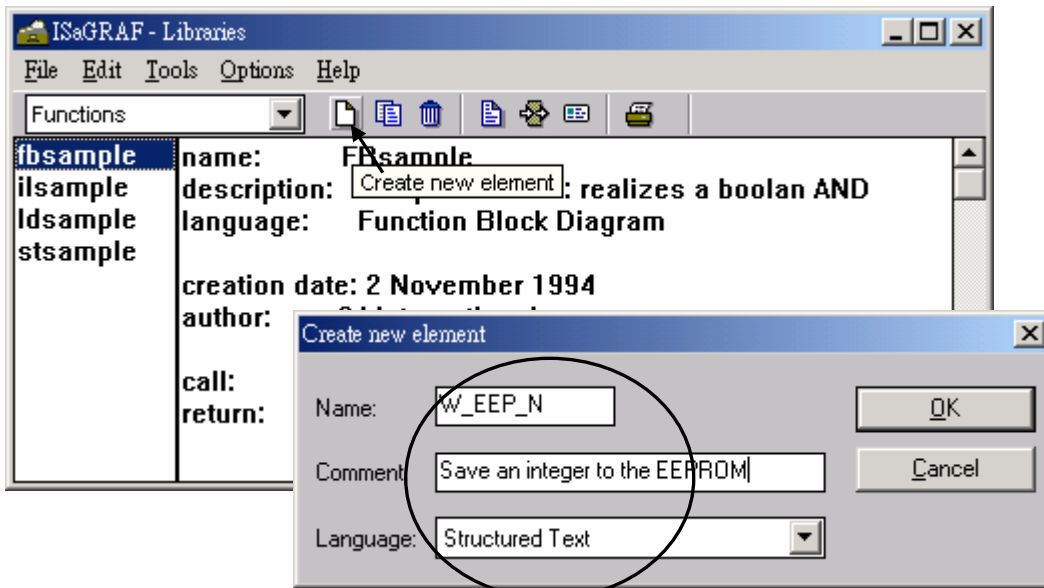
Functions created in the library can be called by programs in any project.

The steps is similar to the former section 15.1. Please refer to it in advance.

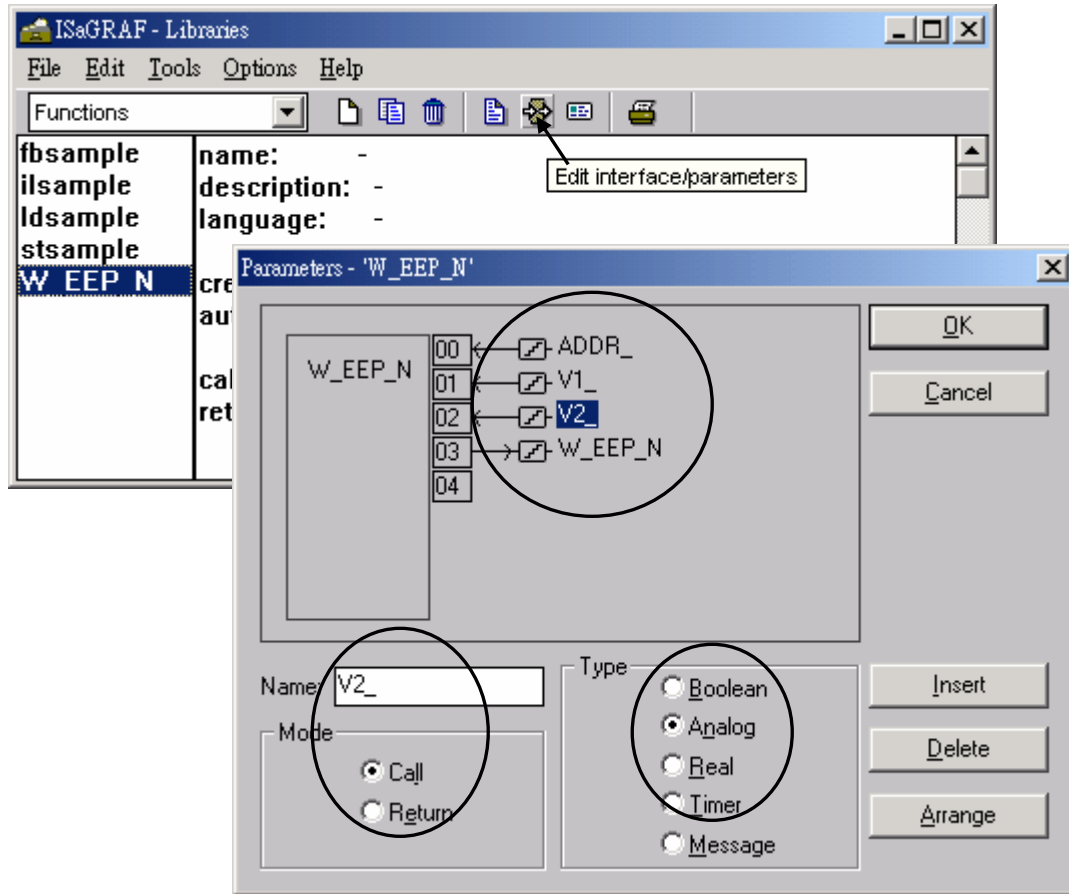
A. Get into the library. Then click on “Functions”



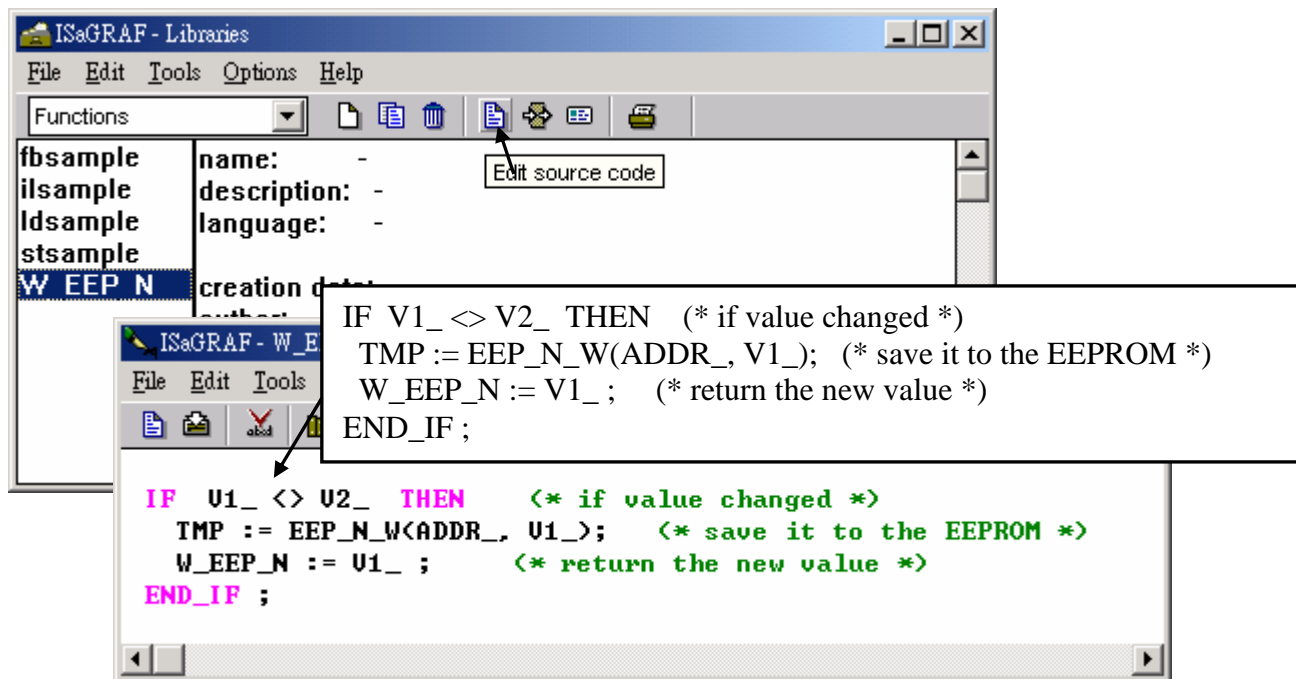
B. Create an new function and given Name as “W\_EEP\_N” , Language as “Structured Text”.



### C. Define input and return parameters

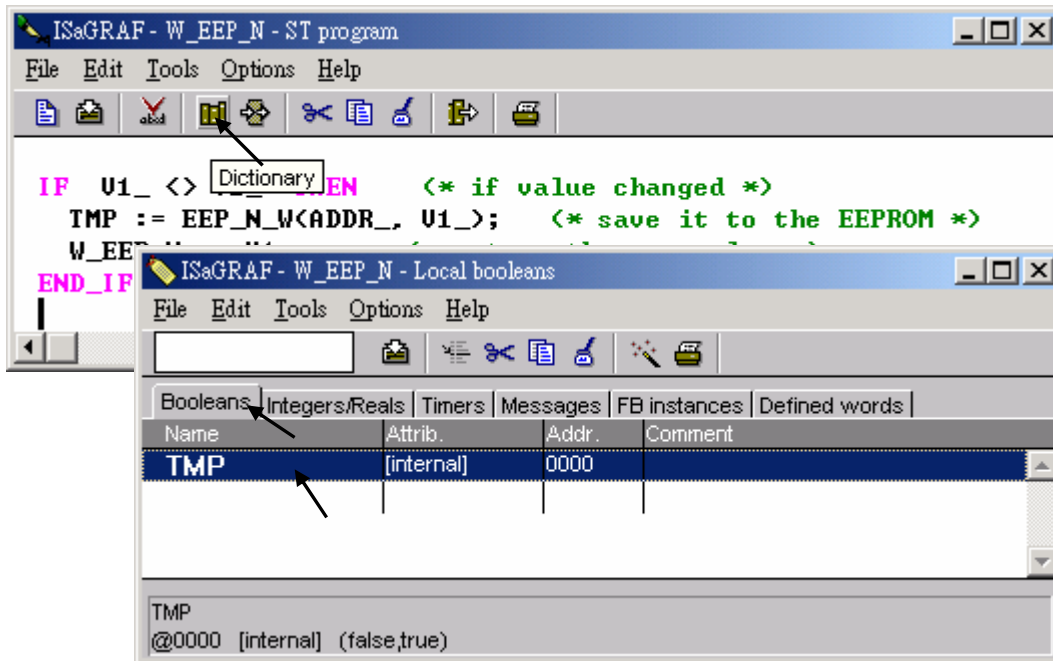


### D. Add codes.

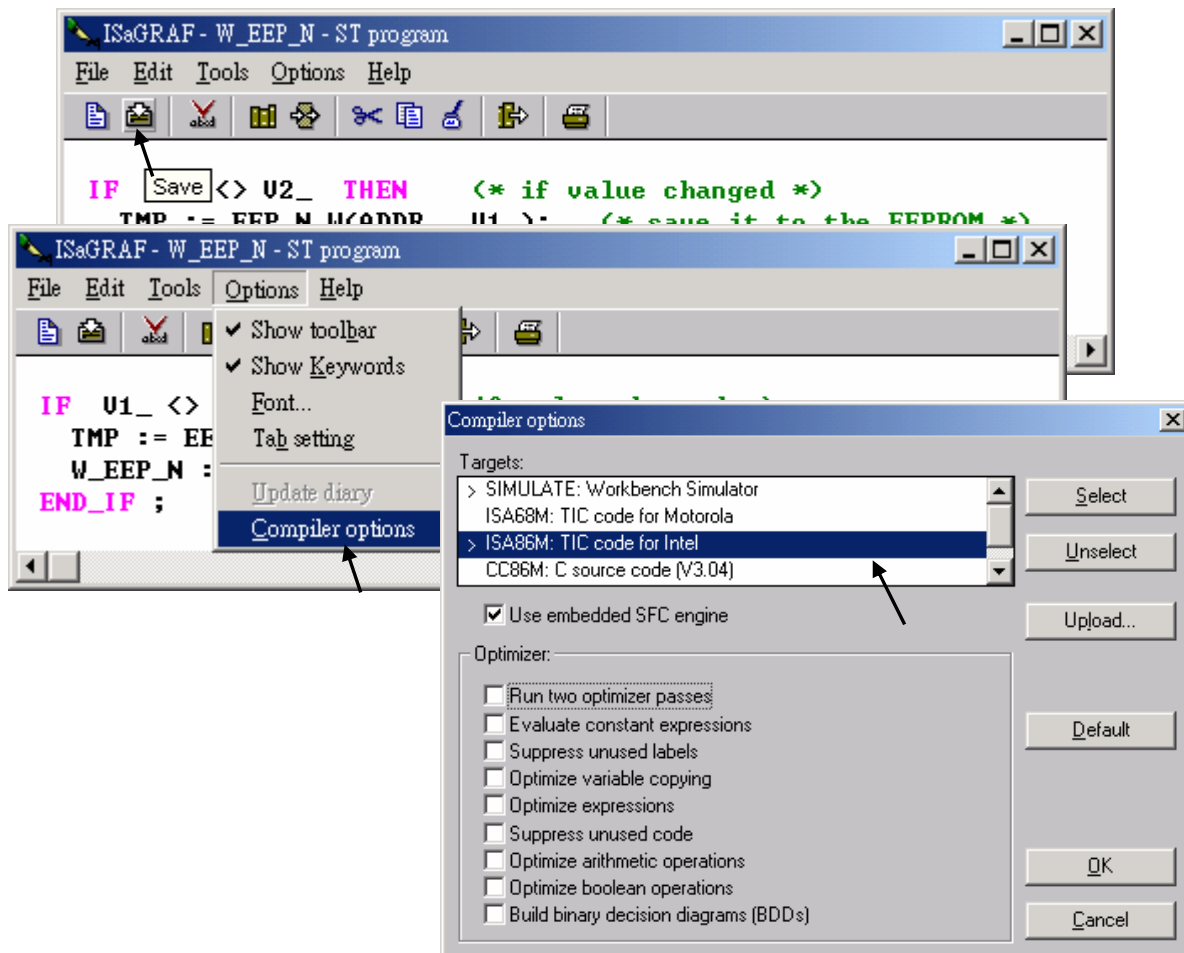




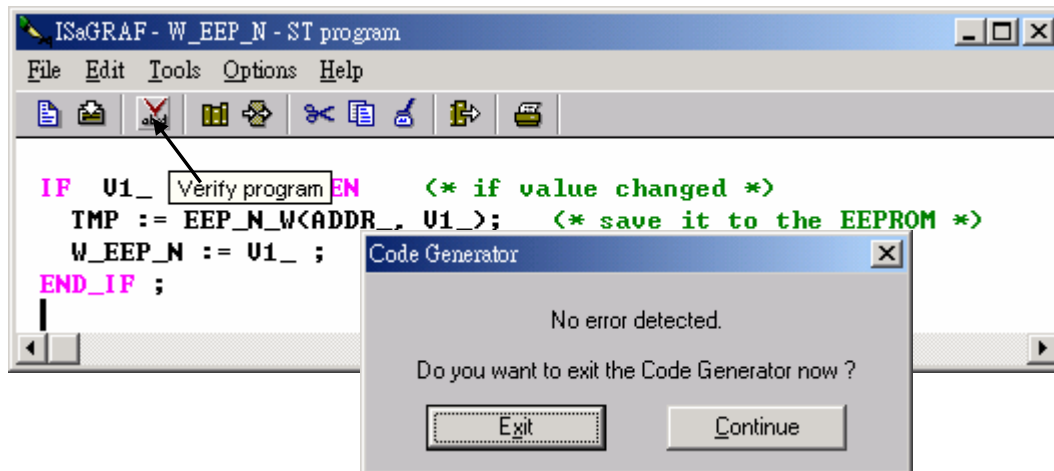
E. Declare local variables. We need a boolean internal variable – “TMP”



F. Save the function and set compiler options.



G. Verify the function.



Then you can call it in any project.

---

## Chapter 16. Linking MMICON

---

The I-8417/8817/8437/8837, iPAC-8447/8847, I-7188EG, I-7188XG, uPAC-7186EG & W-8xx7 controller can integrate the ICP DAS's MMICON to become their Man Machine Interface. The MMICON is featured with a 240 x 64 dot LCD and a 4 x 4 Keyboard. User can use it to display picture, string, integer, float, and input a character, string, integer and float. All control logic is written in ISaGRAF program.

[http://www.icpdas.com/products/HMI/touch\\_lcd/man\\_machine\\_list.htm](http://www.icpdas.com/products/HMI/touch_lcd/man_machine_list.htm)

### 16.1: Hardware Installation

---

Please refer to the “MMICON Hardware Manual” which is delivered with the hardware for more hardware details.

1. The MMICON has a COM port. Please set as a RS232 port. (Please look at the jumper “J7” & “J8” setting on the hardware).

Pin assignment :

**I-8417/8817/8437/8837:** COM3 & COM4 can be used. **W-8xx7:** COM2 or COM5 to COM9

| I-8xx7 (COM4)<br>W-8xx7 (COM2) | MMICON (CN3)<br>RS232 | I-8xx7 (COM3)<br>RS232 | MMICON (CN3)<br>RS232 |
|--------------------------------|-----------------------|------------------------|-----------------------|
| 2 RXD                          | ===== 2 TXD           | 3 RXD                  | ===== 2 TXD           |
| 3 TXD                          | ===== 3 RXD           | 2 TXD                  | ===== 3 RXD           |
| 5 GND                          | ===== 5 GND           | 5 GND                  | ===== 5 GND           |

**I-7188EG/XG:** COM3 can be used. (COM3 is added on X503 ~ X51x board)

| I-7188EG/XG<br>RS232 | MMICON (CN3)<br>RS232 |
|----------------------|-----------------------|
| RXD                  | ===== 2 TXD           |
| TXD                  | ===== 3 RXD           |
| GND                  | ===== 5 GND           |

2. Please set Jumper “J2” of MMICON to position “INIT”. I-8417/8817/8437/8837, I-7188EG/XG & W-8xx7 only support COM parameter “9600, 8, N, 1” and “address = 0” to talk to the MMICON.

## 16.2: Create Background Picture Of the MMICON

---

Please refer to the “MMIDOS Software User Manual” which is delivered with the hardware for more software details.

The number of the background pictures depends on the ROM memory on the MMICON. It can up to 256 pages for EPROM like “27040”, and 128 pages for “27020”, and 64 pages for “27010”.

**Note:** ROM/ EPROM/ EEPROM/ FLASH are all validate.

Please Install the “MMICON” folder from CD-ROM: \Napdos\others\mmicon\ to your hard disk.

**Note:** Please change all these file’s attribute : **removing “Read-only”**

Create all the background pages by Microsoft painter (Please refer to “P0.bmp”).

Edit your “Autox.dat” file (Please refer to “Auto1.dat”). This file must remove its “Read-only” attribute.

Run “MMIDOS.exe” to build the “romx.bin”, For ex. “rom1.bin”

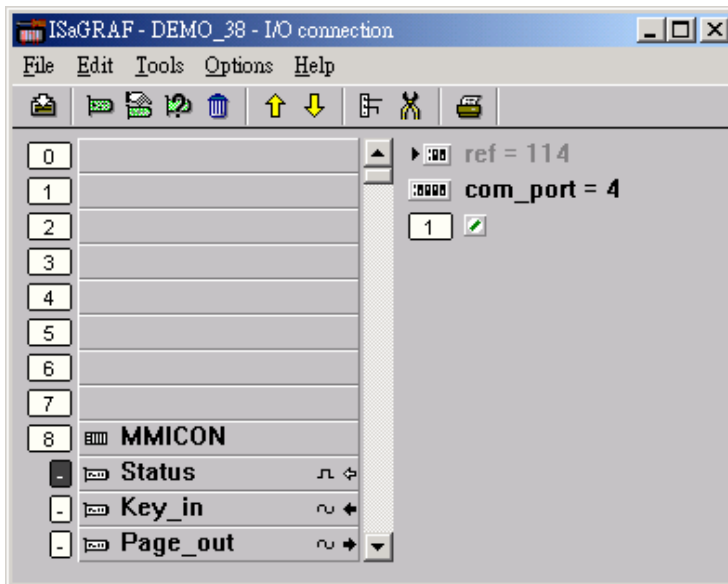
Using your ROM programmer to burn this “romx.bin” image to the ROM memory. Then plug it into the socket on the MMICON.

Please refer to the “MMIDOS Software User Manual” which is delivered with the hardware for more software details.

## 16.3: Writing Control program

---

The I/O complex equipment “mmicon” should be connected to the I/O connection window first. You can find 3 boards under “MMICON”.



**Status:**

Parameter “com\_port” defines the COM No. to link to the MMOCON. 3 or 4 for I-8xx7, while 2 or 3 for I-7188EG/XG , and 2 or 5 to 9 for W-8xx7

1 channel of Digital Input: True means communication between the controller and the MMICON is Ok. FALSE means fail.

**Key\_in:**

1 channel of Integer Input: The value is the key been pressed. And the value will last only for one scan cycle, then go back to 0.

| Key      | Key code value | Key               | Key code value |
|----------|----------------|-------------------|----------------|
| <b>0</b> | 16#30          | <b>Enter</b>      | 16#0D          |
| <b>1</b> | 16#31          | <b>.</b>          | 16#2E          |
| <b>2</b> | 16#32          | <b>Left</b>       | 16#1B          |
| <b>3</b> | 16#33          | <b>Right</b>      | 16#1A          |
| <b>4</b> | 16#34          | <b>Up</b>         | 16#18          |
| <b>5</b> | 16#35          | <b>Down</b>       | 16#19          |
| <b>6</b> | 16#36          | <b>Back space</b> | 16#08          |
| <b>7</b> | 16#37          | <b>F1</b>         | 16#F1          |
| <b>8</b> | 16#38          | <b>F2</b>         | 16#F2          |
| <b>9</b> | 16#39          | <b>F3</b>         | 16#F3          |
| <b>A</b> | 16#41          | <b>F4</b>         | 16#F4          |
| <b>B</b> | 16#42          |                   |                |
| <b>C</b> | 16#43          |                   |                |
| <b>D</b> | 16#44          |                   |                |
| <b>E</b> | 16#45          |                   |                |
| <b>F</b> | 16#46          |                   |                |

**Page\_out:**

1 channel of Integer Output: The value output define the page No. to display.

The I-8417/8817/8437/8837, I-7188EG, I-7188XG & W-8xx7 controller provide below functions to control the action of the MMICON.

|          |  |
|----------|--|
| MI_BOO   | Display a boolean value as “ON” or “OFF” |
| MI_INT   | Display an integer value                 |
| MI_REAL  | Display a real value                     |
| MI_STR   | Display a string                         |
| MI_INP_N | To enter an integer                      |
| MI_INP_S | To enter a string                        |
| REAL_STR | Convert a real value to a string         |
| STR_REAL | Convert a string to a real value         |

Please refer to I-8xx7’s demo\_38, dem\_39 and Appendix A.4

---

## Chapter 17. SMS: Short Message Service

---

The I-8417/8817/8437/8837, iPAC-8447/8847, I-7188EG, I-7188XG, uPAC-7186EG & Wincon-8xx7 controller can integrate with a GSM Modem to support SMS: Short Message Service. This allows user to request information or control something from his own cellular phone to the ISaGRAF controller. Beside, the controller can also send information and alarms to user's cellular phone.

### 17.1: Hardware Installation

---

The I-8417/8817/8437/8837 supports SMS since its driver version of 2.24, while version 1.14 for I-7188EG, and version 1.12 for I-7188XG, and version of 3.10 for W-8xx7 (iPAC-8447/8847 & uPAC-7186EG support it since they are released around Q1/Q2, 2008). If your driver is older one, please upgrade the hardware driver to the associate version or a higher version. The driver can be found from the below ICP DAS's web site:

<http://www.icpdas.com/products/PAC/i-8000/isagraf.htm>

The I/O library should be re-installed if yours is older one. Please refer to section 1.2. Or you can refer to Appendix A.2 to simply install "C functions" with the below items.

SMS\_test, SMS\_get, SMS\_gets, SMS\_send, SMS\_sts  
and "I/O complex equipment" : SMS.

The GSM Modem **M1206** (900/1800) is recommended for the ISaGRAF controller since its driver version of I-8xx7:2.47, I-7188EG:1.38, I-7188XG:1.35 & Wincon-8xx7:3.10. You may purchase them from ICP DAS or from your local agent. ICP DAS is not sure for other GSM modems working or not.

**Note: Please REMOVE the password setting in SIM card , then plug it into GSM modem.**

| I-8xx7(COM4/5)<br>W-8xx7(COM2) | GSM cable of<br>M1206 (or GM29) | 7188EG/XG:COM3/4<br>RS232 | GSM cable of<br>M1206 (or GM29) |
|--------------------------------|---------------------------------|---------------------------|---------------------------------|
| 2 RXD                          | ———— 2 TXD                      | RXD                       | ———— 2 TXD                      |
| 3 TXD                          | ———— 3 RXD                      | TXD                       | ———— 3 RXD                      |
| 5 GND                          | ———— 5 GND                      | GND                       | ———— 5 GND                      |
| 4 DTR                          | ----- 4 DSR                     | DTR (or RTS)              | ----- 4 DSR                     |
| 7 RTS                          | ----- 7 CTS                     | DTR (or RTS)              | ----- 7 CTS                     |

## 17.2: A SMS demo example

The demo project is located at I-8xx7's demo\_43 and demo\_43a, please refer to section 11.1 to install it to your ISaGRAF workbench. Or It can be download at ICP DAS's ftp site.

<ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/8000/demo/>

Variables :

| Name          | Type    | Attribute       | Description   |
|---------------|---------|-----------------|---|
| M1            | Boolean | <b>Internal</b> | Trigger to send an alarm message when K1 is pushed        |
| M2            | Boolean | <b>Internal</b> | Trigger to send a report message when a message is coming |
| K1            | Boolean | <b>Input</b>    | Pushbutton 1, connect to push4key                         |
| L1            | Boolean | <b>Output</b>   | Output 1, connect to show3led                             |
| L2            | Boolean | <b>Output</b>   | Output 2, connect to show3led                             |
| L3            | Boolean | <b>Output</b>   | Output 3, connect to show3led                             |
| Q1            | Boolean | <b>Internal</b> | Test if message is coming                                 |
| TMP           | Boolean | <b>Internal</b> | Temportary usage  |
| SMS_available | Boolean | <b>Input</b>    | is SMS available ? connect to SMS - status                |
| T1            | Timer   | <b>Internal</b> | Blinking time of L1 to L3, init at T#500ms                |
| data          | Message | <b>Internal</b> | The coming Message  |
| phone         | Message | <b>Internal</b> | phone No. of sender                                       |
| Date_time     | Message | <b>Internal</b> | Message coming date & time in string format               |
| To_who        | Message | <b>Internal</b> | phone No of receiver, <b>please use your own No.</b>      |
| Msg_to_send   | Message | <b>Internal</b> | Message to send out                                       |
| Year1         | Integer | <b>Internal</b> | Message coming year                                       |
| Mon1          | Integer | <b>Internal</b> | Message coming month                                      |
| Day1          | Integer | <b>Internal</b> | Message coming date                                       |
| Wday1         | Integer | <b>Internal</b> | Message coming week date                                  |
| Hour1         | Integer | <b>Internal</b> | Message coming hour                                       |
| Min1          | Integer | <b>Internal</b> | Message coming minute                                     |
| Sec1          | Integer | <b>Internal</b> | Message coming second                                     |
| Q1_cnt        | Integer | <b>Internal</b> | Message coming count, declared as retained variable       |
| Msg_status    | Integer | <b>Internal</b> | Message sending status                                    |
| TMP_v         | Integer | <b>Internal</b> | temportary usage  |

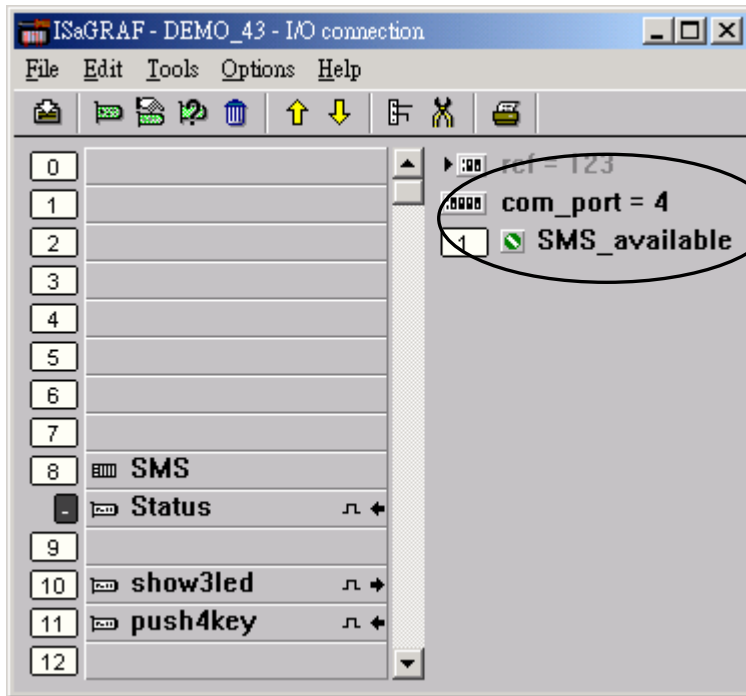
Project architecture :



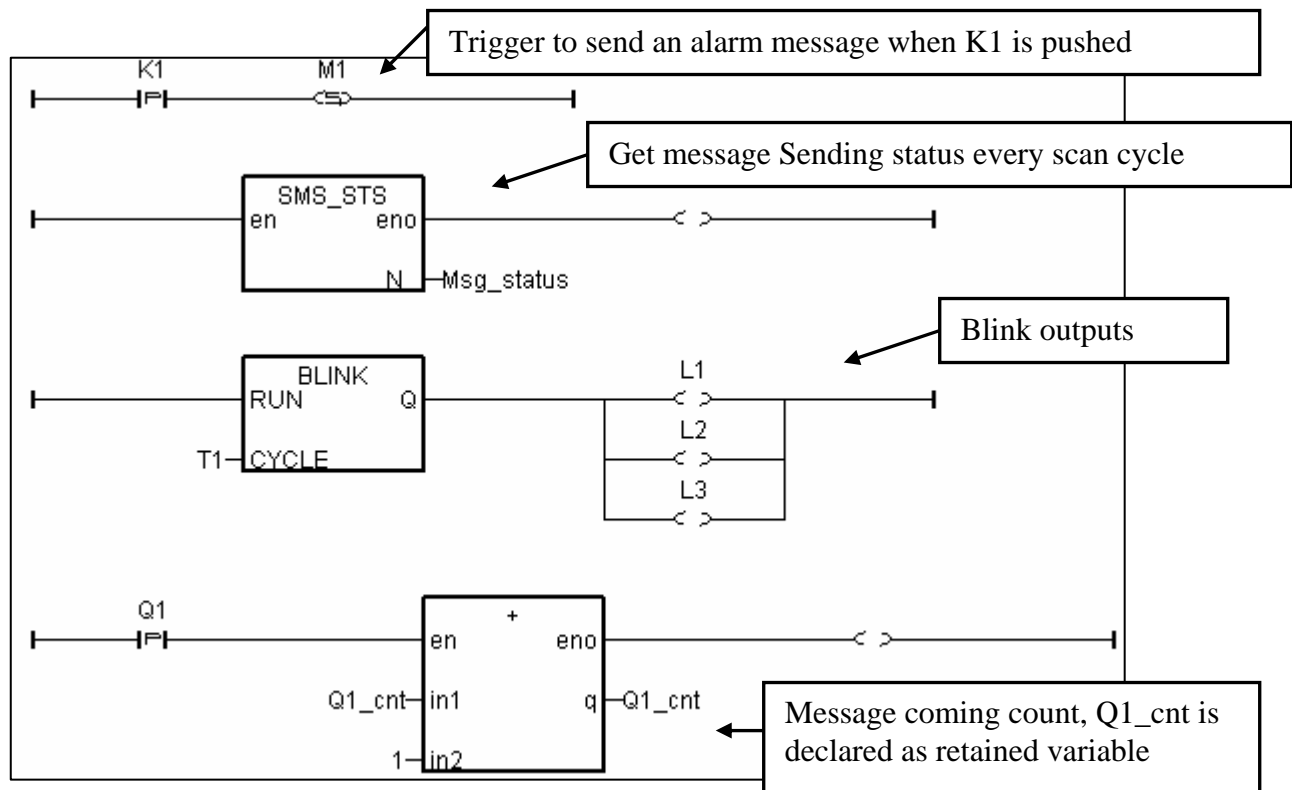
Operation actions:

1. If K1 is pushed, an Alarm message will be sent.
2. If the user send a message in format, for ex. T0200 or T1500 to the controller, the blinking period will change to 200ms and 1500ms. And then the controller will response a report message back to the user.

I/O connection:

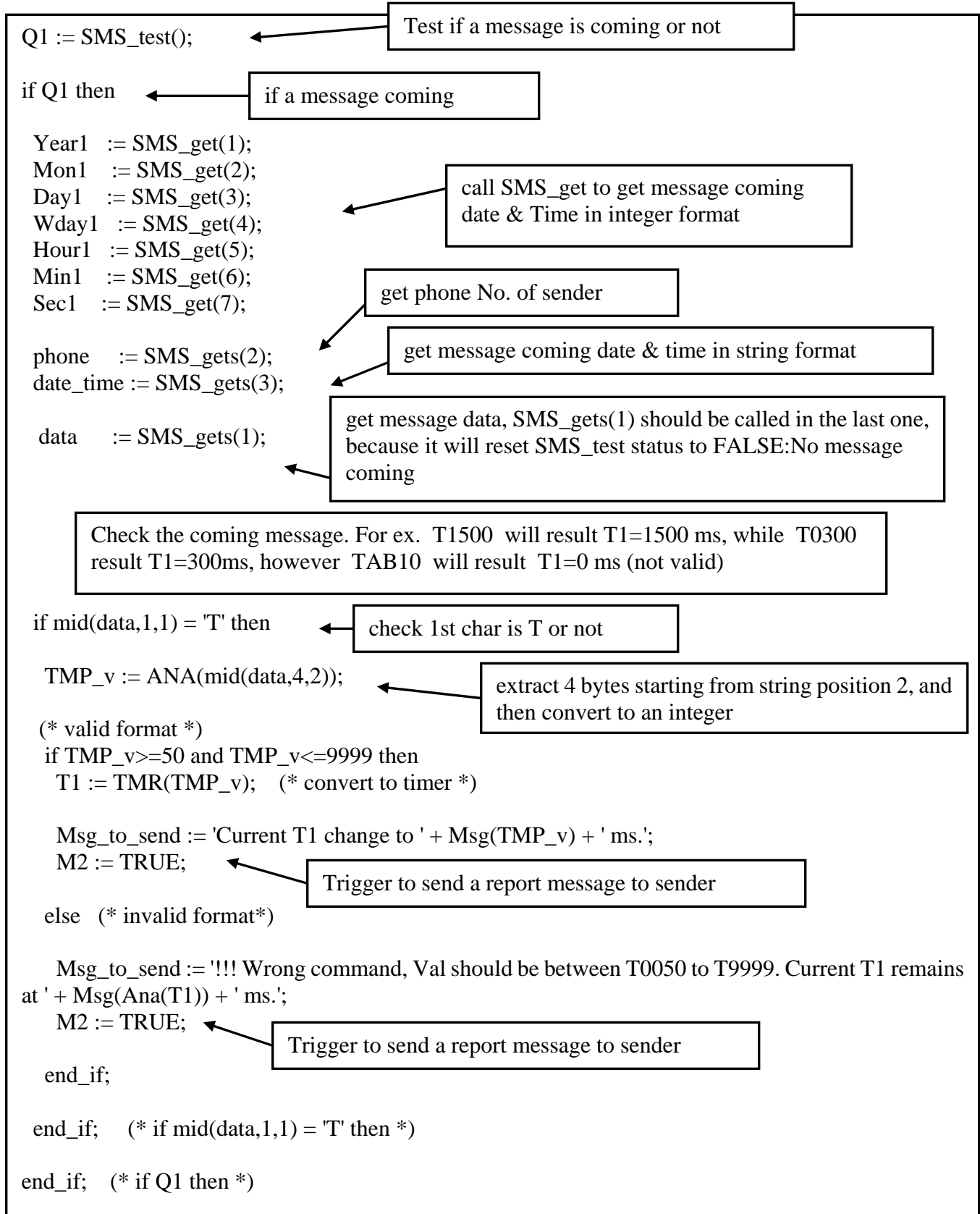


LD program : work

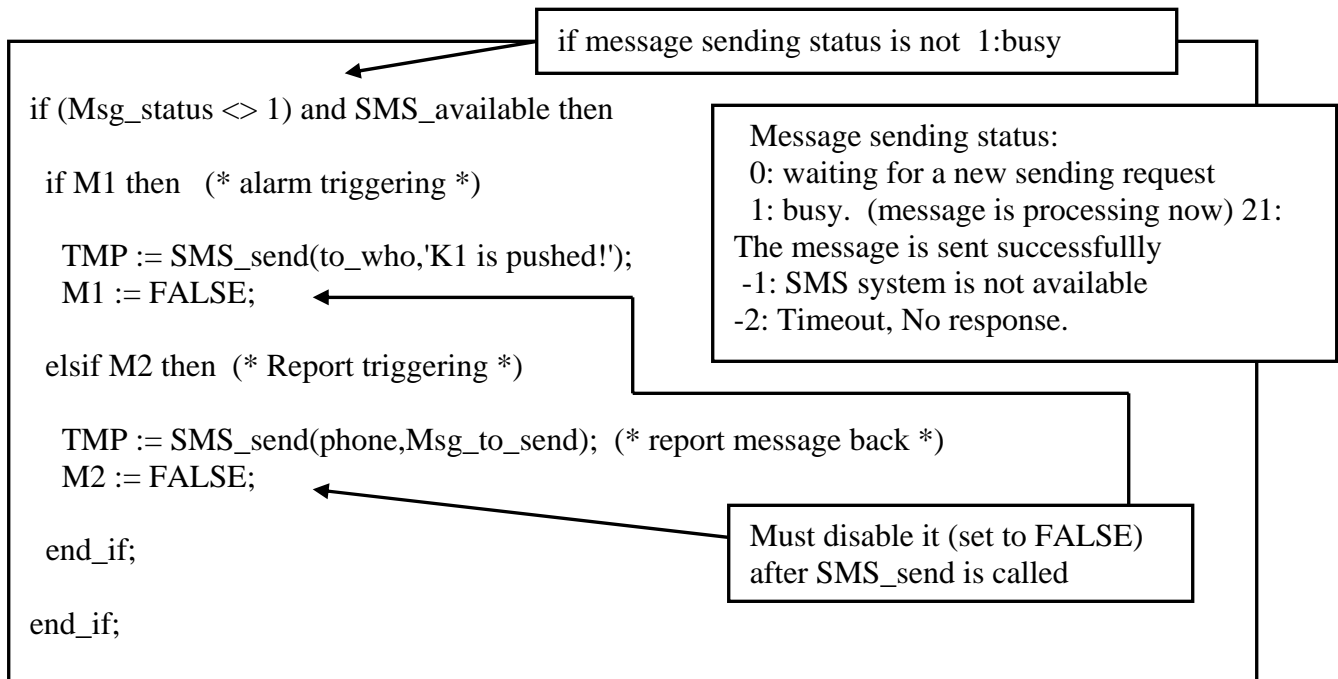




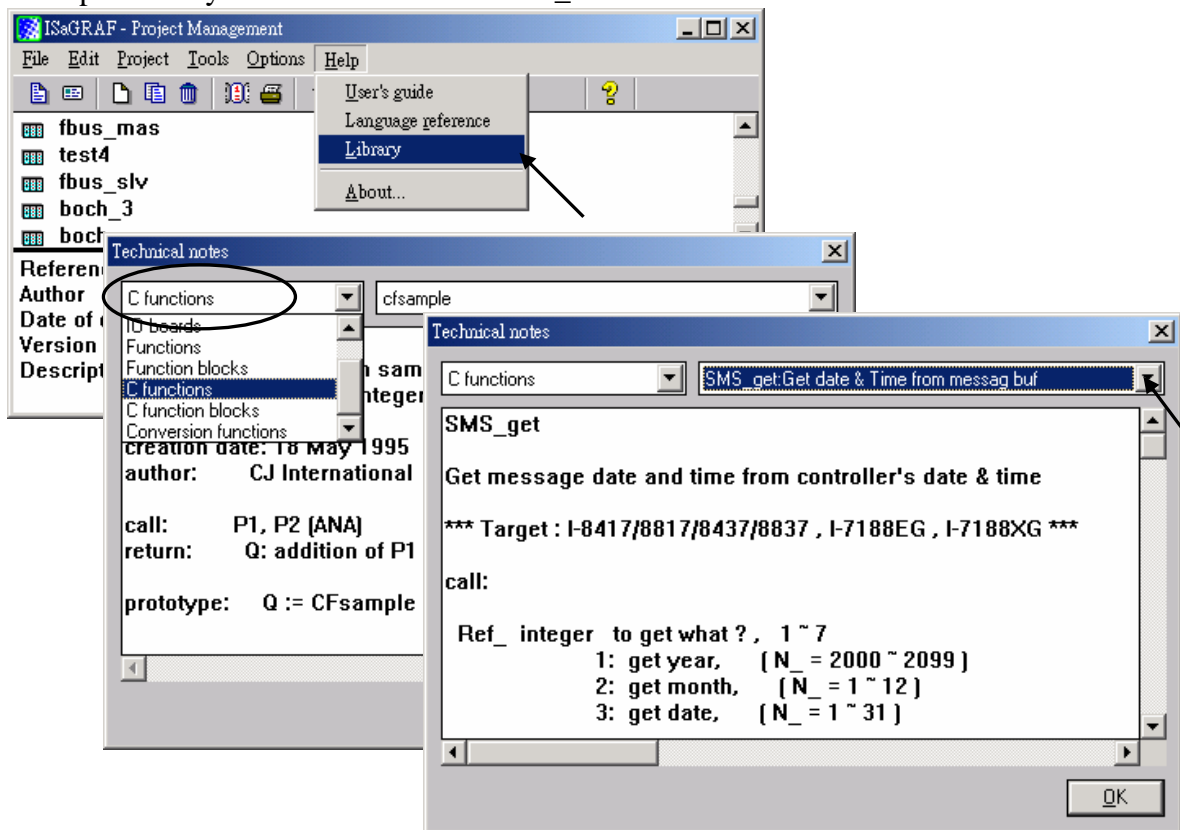
ST program : rcv\_msg



ST program : snd\_msg



More description of SMS\_sts, SMS\_send, SMS\_test, SMS\_get & SMS\_gets, Please refer to ISaGRAF's On-line Help. "Library" – "C functions" – "SMS\_xxxx"



---

# Chapter 18. Motion

---

## 18.1: Install motion driver

---

### Limitation:

1. I-8437/8837 **CAN NOT** do ethernet communication when using I-8091 to do motion control, while W-8xx7 doesn't have this limitation.
2. Only one I-8091 board in I-8417/8817/8437/8837 & W-8xx7 can do X-Y dependent motion, other I-8091s should be moving independent. Or all I-8091s are moving independent.

The I-8417/8817/8437/8837 & Wincon-8xx7 can integrate with the I-8091 to do Motion control. The default ISaGRAF driver burned in the Flash memory of the I-8417/8817/8437/8837 controller is for general usage not for motion control. Please update it to the motion driver by yourself. While user don't need to upgrade the driver of Wincon-8xx7 if its driver version is 3.08 or higher.

The motion driver of I-8417/8817/8437/8837 can be found in the ICP DAS CD-ROM.

napdos\isagraf\8000\driver\motion?.??\

or can be downloaded from

<ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/8000/driver/motion?.??>

Please refer to the "ReadMe.txt" in the folder of "motion?.??" (for ex. "Motion2.45")

### Restriction of the motion driver of I-8417/8817/8437/8837:

The motion driver for I-8417/8817/8437/8837 doesn't support the Ethernet communication, however W-8xx7 doesn't have this limitation.

The ISaGRAF demo projects of motion for I-8417/8817/8437/8837 are "demo\_27", "demo\_28", & "demo\_46". They are located in the 8000 CD-ROM:

napdos\isagraf\8000\demo\

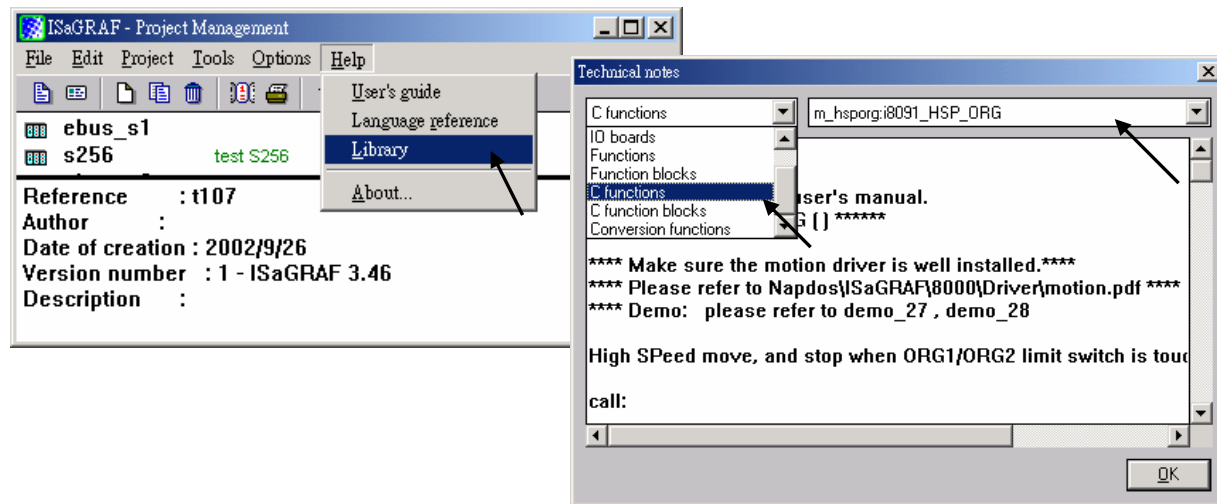
<ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/8000/demo/>

The ISaGRAF demo projects of motion for W-8xx7 are "wdemo\_26", "wdemo\_27", "wdemo\_28" & "wdemo\_29". They are located in the Wincon CD-ROM:

napdos\isagraf\wincon\demo\

<ftp://ftp.icpdas.com/pub/cd/winconcd/napdos/isagraf/wincon/demo/>

All functions that trigger I-8091 & I-8090 are named as "M\_???", Please refer to the On-line help from the ISaGRAF "Help" – "Library" - "C functions" for names starting with "M\_???".



Beside, please refer to "I-8091 & I-8090 User's Manual" .It can be found in the package box of the i-8091, or

CD-ROM: napdos\8000\motion\i8091>manual\

ftp site: <ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/8000/motion/i8091/manual/>

## 18.2: Introduction

### 18.2.1: System Block Diagram

The I-8091 stepping motor control card is a micro-computer controlled, 2-axis pulse generation card. It includes a 2Kbytes-FIFO to receive motion command from host, a micro-computer for profile generation and protection, 2-axis DDA chip to execute DDA function when interpolation command is used, 2500Vrms optical isolation inserted for industrial application.

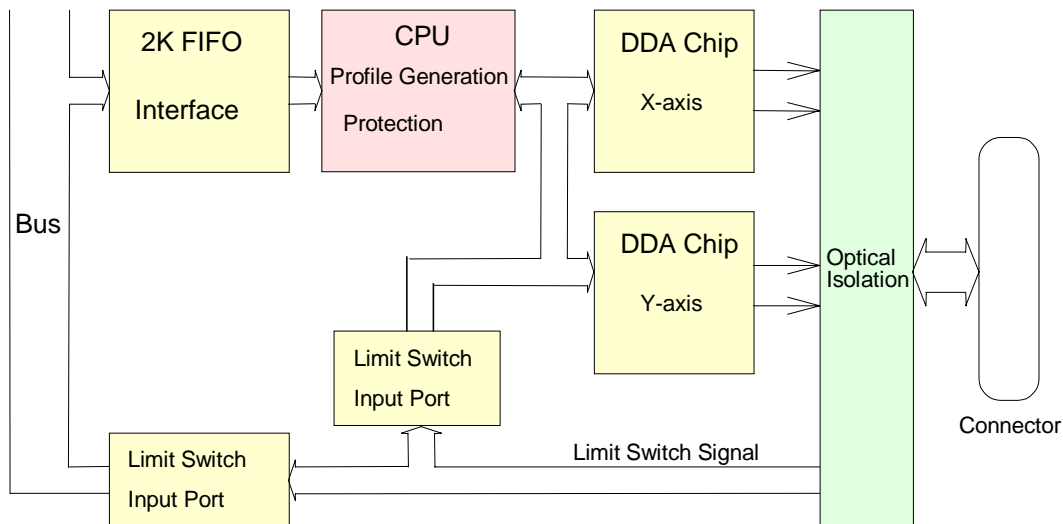


Fig.(1) block diagram of I-8091 card

### 18.2.2: DDA Technology

The DDA chip is the heart of I-8091 card, it will generate equal-space pulse train corresponding to specific pulse number during a DDA period. This mechanism is very useful to execute pulse generation and interpolation function. The DDA period can be determined by DDA cycle. Table(1) shows the relation among DDA cycle, DDA period and output pulse rate. When DDA cycle set to 1, the DDA period is equal to  $(1+1) \times 1.024\text{ms} = 2.048\text{ms}$ . The output pulse number can be set to 0~2047, therefore the maximum output pulse rate will be 1Mpps. The minimum output pulse rate is 3.83pps when set DDA cycle=254 (DDA period =  $(254+1) \times 1.024\text{ms} = 261.12\text{ms}$ ).

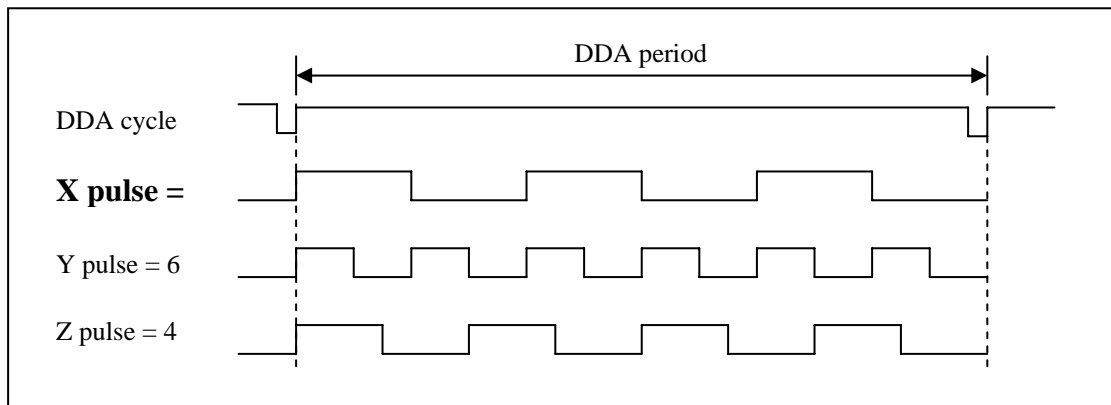


Fig.(2) DDA mechanism

Table(1) The Relation among DDA cycle, DDA period and output pulse rate.

| DDA cycle | DDA period    | Max. pulse rate(n=2047) | Min. pulse rate (n=1) |
|-----------|---------------|-------------------------|-----------------------|
| 1         | 2.048ms       | 999511pps               | 488pps                |
| 2         | 3.072ms       | 666341pps               | 325pps                |
| 3         | 4.096ms       | .                       | .                     |
| .         | .             | .                       | .                     |
| N         | (N+1)*1.024ms | 2047/(DDA period)       | 1/(DDA period)        |
| .         | .             | .                       | .                     |
| 254       | 261.12ms      | 7839pps                 | 3.83pps               |

The DDA cycle can be set by i8091\_SET\_VAR() command which deccribed in charppter 3. The selection criterion of DDA cycle was described as following.

1. The required max. output pulse rate.

$$PR_{max} = \frac{V_{max} * N / 60}{2047}$$

$$PR_{max} = \frac{(DDA_{cycle} + 1) * 1.024ms}{Np}$$

PR<sub>max</sub> : max. output pulse rate.

V<sub>max</sub> : max. speed (rpm).

N : the pulse number of stepping motor per revolution (pulse/rev).

2. The required speed resolution.

The maximum output pulse number is Np(0~2047), therefore the speed resolution is V<sub>max</sub>(max. speed)/Np. The DDA cycle can be obtained by following equation.

$$PR_{max} = \frac{Np}{(DDA_{cycle} + 1) * 1.024ms}$$

3. When choose large DDA cycle (DDA period), it will occur vibration between different pulse input which generally can be observed during acceleration or deceleration. So, the small DDA cycle , the smooth acceleration/deceleration curve as long as the speed resolution is acceptable.

### Example: Stepping Motor

The spec. of stepping motor is 500 pulse/rev, max. speed 500 rpm, speed resolution 2 rpm.

The required max. pulse rate

$$PR_{max} = 500 \text{ rpm} * 500 / 60 = 4166.67 \text{ pps}$$

The maximum output pulse

$$Np = 500 \text{ rpm} / 2 \text{ rpm} = 250 \text{ pulse number}$$

The DDA cycle can be calculated by follow equation

$$PR_{max} = \frac{Np}{(DDA_{cycle} + 1) * 1.024ms}$$

$$4166.67 = \frac{250}{(DDA_{cycle} + 1) * 1.024ms}$$

DDA cycle = 58

High Speed = 247 pulse (4166.67\*58\*0.001024)

The above results means that maximum speed is 500rpm when send command i8091\_SET\_VAR(0, 58, 2, 2, 247) to I-8091 card.

### Example: Pulse type input Servo Motor

The spec. of servo motor is 8000 pulse/rev, max. speed 3000 rpm, speed resolution 2 rpm.

The required max. pulse rate

$$PR_{max} = 3000 \text{ rpm} * 8000 / 60 = 400,000 \text{ pps}$$

The maximum output pulse

$$Np = 3000 \text{ rpm} / 2 \text{ rpm} = 1500 \text{ pulse number}$$

The DDA cycle can be calculated by follow equation

$$PR_{max} = \frac{Np}{(DDA_{cycle} + 1) * 1.024ms}$$

$$400,000 = \frac{1500}{(DDA_{cycle} + 1) * 1.024ms}$$

DDA cycle = 3

High Speed = 1638 pulse (400,000\*4\*0.001024)

The above results means that maximum speed is 3000rpm when send command i8091\_SET\_VAR(0, 3, 2, 2, 1638) to I-8091 card.

## 18.3: Hardware

### 18.3.1: I-8000 hardware address

The hardware address of I-8000 main system is fixed as following table. There are 4 slots I-8000 and 8 slots I-8000.

|                        | Slot 0 | Slot 1 | Slot 2 | Slot 3 | Slot 4 | Slot 5 | Slot 6 | Slot 7 |
|------------------------|--------|--------|--------|--------|--------|--------|--------|--------|
| I-8000, 4 slot address | 0x080  | 0x0A0  | 0x0C0  | 0x0E0  | ---    | ---    | ---    | ---    |
| I-8000, 8 slot address | 0x080  | 0x0A0  | 0x0C0  | 0x0E0  | 0x140  | 0x160  | 0x180  | 0x1A0  |

Fig.(3) I-8000 hardware address

### 18.3.2: LED Indicator

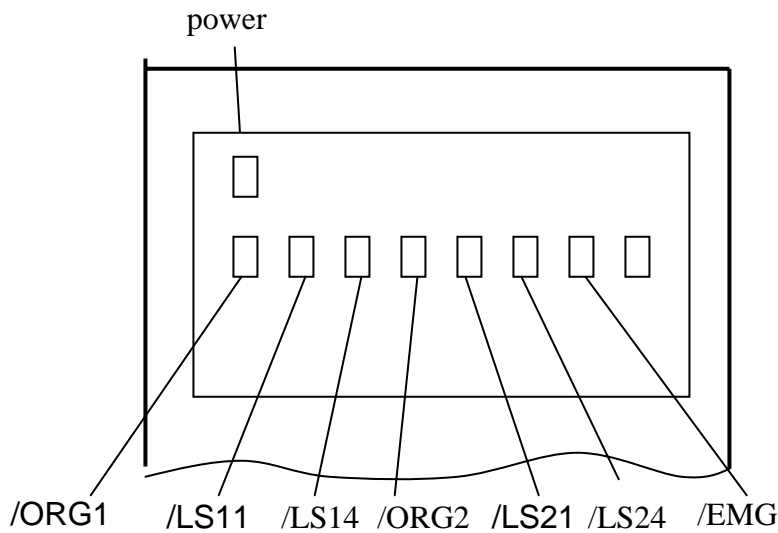


Fig.(4) I-8091 LED indicator

/ORG1: X-axis's original limit switch for machine home position.  
 /LS11, /LS14 : X-axis's negative and positive limit switches.  
 /ORG2: Y-axis's original limit switch for machine home position.  
 /LS21, /LS24 : Y-axis's negative and positive limit switches.  
 /EMG : system's emergency signal input.

### 18.3.3: Hardware Configuration

#### Limit switch configuration

Because the profile generation and protection is executed by the CPU on I-8091 card, the limit switches must configure as following diagram. The motion command just can work properly.

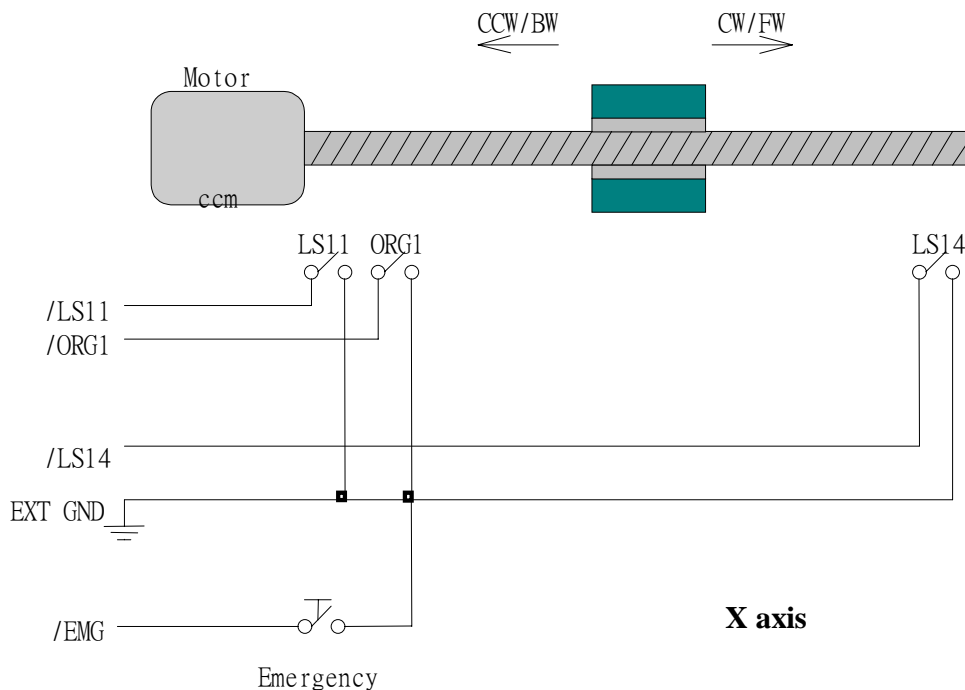


Fig.(5) Limit switch configuration of X axis



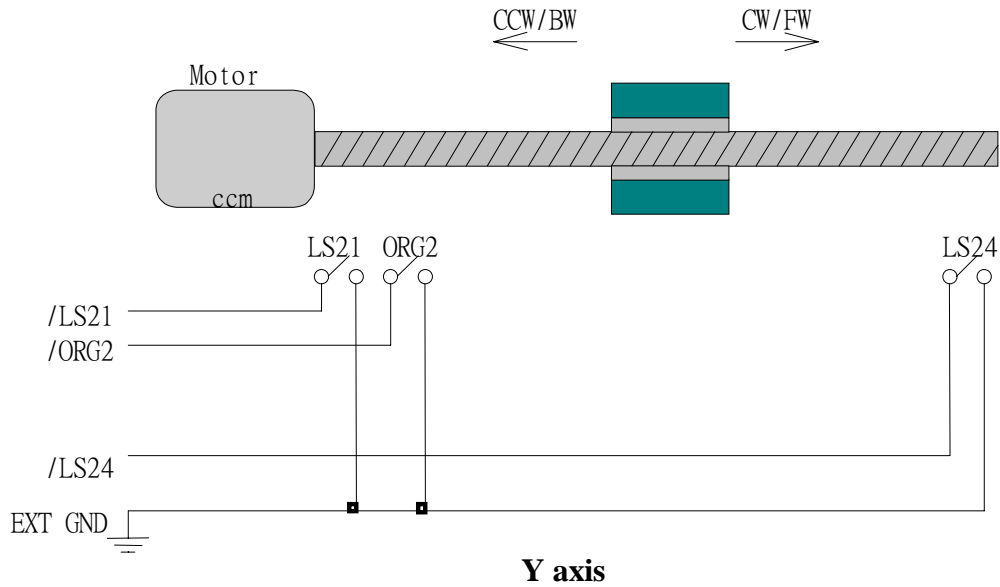


Fig.(6) Limit switch configuration of Y axis

## Output pulse mode configuration

I-8091 card provide two kind output method.

- (a) CW/CCW mode
- (b) Pulse/Direction mode

The command **M\_s\_mode(card\_NO\_, modeX\_, modeY\_)** provide parameters 0: CW\_CCW and 1: PULSE\_DIR to define output pulse mode.

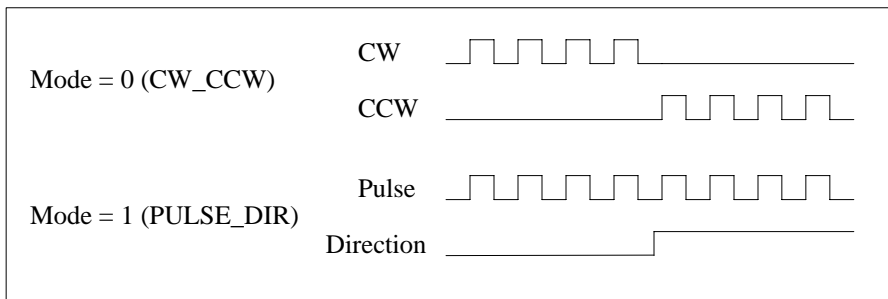


Fig.(7) Output pulse mode

## Direction configuration

Sometimes, the output direction of X-axis, Y-axis is not in the desired direction due to the motor's connection or gear train. It is recommended to unify the output direction as shown in Figure(5)(6). The CW/FW direction is defined as toward outside from motor and the CCW/BW direction is defined as toward inside to motor. The **M\_s\_dir(card\_NO\_, defdirX\_, defdirY\_)** command provides parameters 0: NORMAL\_DIR and 1: REVERSE\_DIR to define the rotating direction of motor.

## Turn Servo ON/OFF (Hold ON/OFF)

To turn servo motor into servo ON(OFF) state, or turn stepping motor into hold ON(OFF) state, the command **M\_s\_serv(card\_NO\_, sonX\_, sonY\_)** provide parameters 1:ON and 0:OFF to turn ON or OFF.

## Automatic protection

The I-8091 card has a automatic protected system.

- (a) If X-axis command is executing and moving toward CW/FW direction, X-axis will immediately stop when LS14 is touched. To release this protection as long as X-axis move toward CCW/BW direction.
- (b) If X-axis command is executing and moving toward CCW/BW direction, X-axis will immediately stop when LS11 is touched. To release this protection as long as X-axis move toward CW/FW direction.
- (c) If Y-axis command is executing and moving toward CW/FW direction, Y-axis will immediately stop when LS24 is touched. To release this protection as long as Y-axis move toward CCW/BW direction.
- (d) If Y-axis command is executing and moving toward CCW/BW direction, Y-axis will immediately stop when LS21 is touched. To release this protection, as long as Y-axis move toward CW/FW direction.
- (e) If the signal of the emergency limit switch /EMG was found in CPU firmware, all motion will be terminated and stop.

## Set limit switch as normal close condition

The limit switches /EMG, /LS11, /LS14, /LS21, /LS24, /ORG1, /ORG2 is initially normal open condition, that is, these signal is active when connect it to ground. In industrial application, it might be recommended normal close condition, that is, these signal is active when open from ground.

The **M\_s\_nc(card\_NO\_, sw\_)** command can be set sw=0 (default), for normal open condition. When set sw=1, for normal close condition.

### 18.3.4: Pin assignment of connector CN2

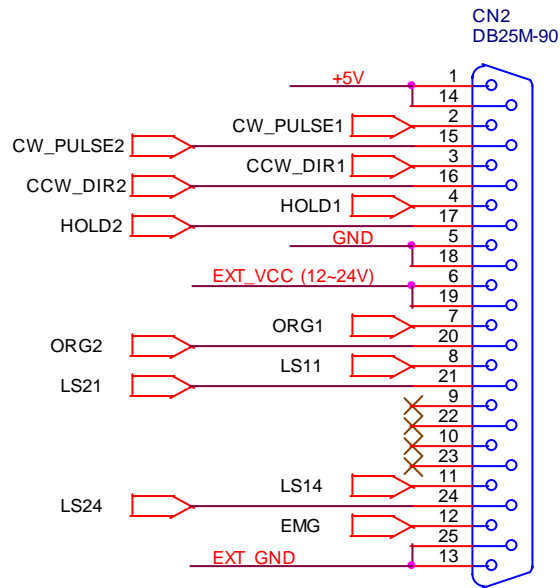


Fig.(8) CN2 connector of I-8091

Table of CN2 connector's pin assignment

| pin name  | pin number | Description                                   |
|-----------|------------|---|
| +5V       | 1          | Internal +5V power, Max. output current: 50mA |
| CW_PULSE1 | 2          | X-axis CW (Pulse) output pin                  |
| CCW_DIR1  | 3          | X-axis CCW (Direction) output pin             |
| HOLD1     | 4          | X-axis HOLD (servo on) output pin             |
| GND       | 5          | Signal ground of pin 2,3,4                    |
| EXT_VCC   | 6          | External power(12~24V) for limit switches     |
| /ORG1     | 7          | X-axis original (home) limit switch           |
| /LS11     | 8          | X-axis limit switch                           |
|           | 9,10       | No used                                       |
| /LS14     | 11         | X-axis limit switch                           |
| /EMG      | 12         | Emergency input                               |
| EXT_GND   | 13         | External ground for limit switch              |
| +5V       | 14         | Internal +5V power, Max. output current: 50mA |
| CW_PULSE2 | 15         | Y-axis CW (Pulse) output pin                  |
| CCW_DIR2  | 16         | Y-axis CCW (Direction) output pin             |
| HOLD2     | 17         | Y-axis HOLD (servo on) output pin             |
| GND       | 18         | Signal ground of pin 15,16,17                 |
| EXT_VCC   | 19         | External power(12~24V) for limit switches     |
| /ORG2     | 20         | Y-axis original (home) limit switch           |
| /LS21     | 21         | Y-axis limit switch                           |
|           | 22,23      | No used                                       |
| /LS24     | 24         | Y-axis limit switch                           |
| EXT_GND   | 25         | External ground for limit switch              |

### The internal circuit of CW\_PULSE, CCW\_DIR, HOLD

When output these signal as 1, it can source 15mA(max.).  
When output these signal as 0, it can sink 50mA(max.)

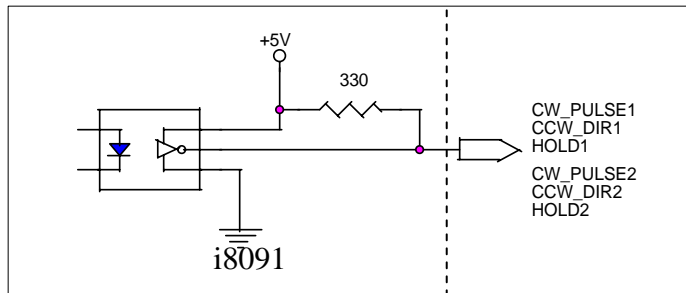


Fig.(9) internal circuit of pulse output pin

### The internal circuit of limit switch input

Initially, the limit switch inputs of I-8091 board are normal open (N.O.), the I-8091 board will automatic protect when limit switch pin connect to EXT\_GND. The user can use the command **M\_s\_nc(card\_NO\_, 1)** to let those limit switch input as normal close condition at the beginning of the user's program.

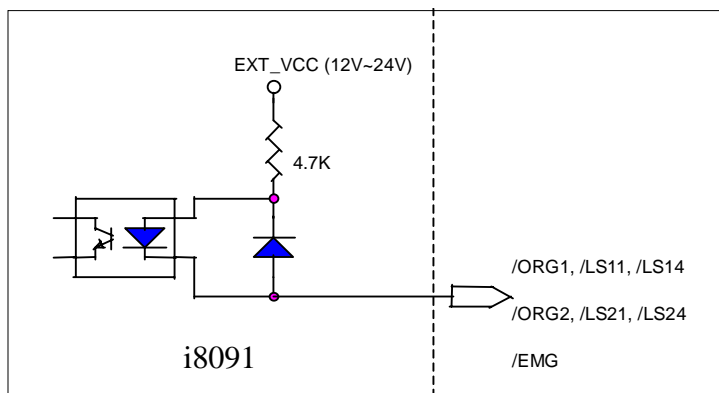


Fig.(10) internal circuit of limit switch input pin

Example of connection

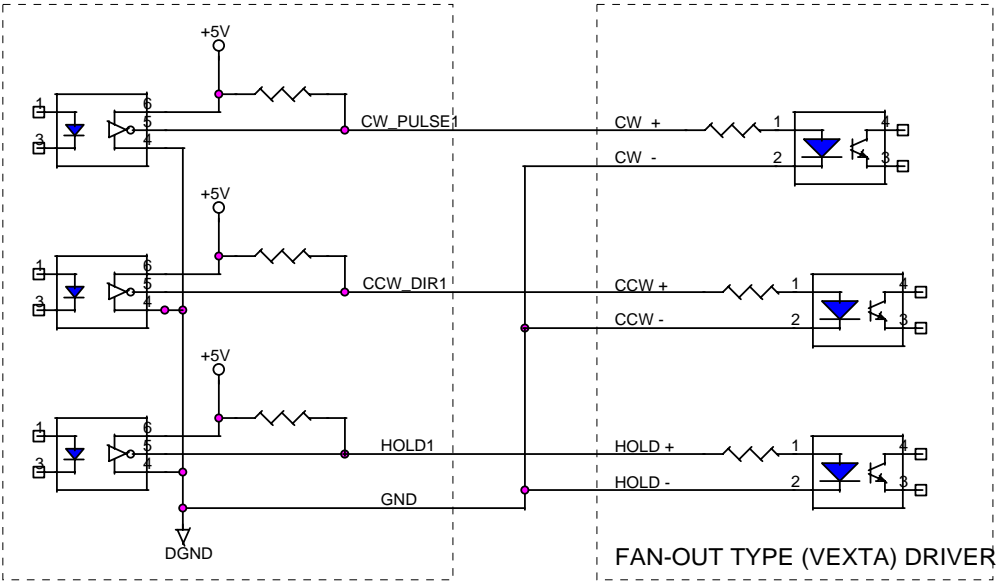


Fig.(11) fan-out type driver (VEXTA's motor driver)

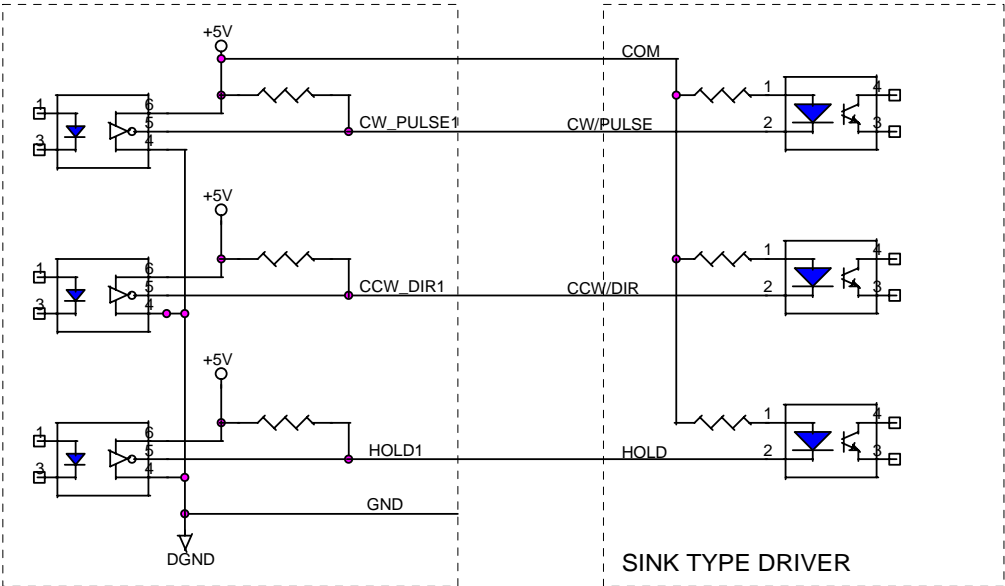


Fig.(12) Sink type driver



## 18.4: Software

### I/O connection:

The “**I-8091A**” connected on the I/O connection window contains 11 digital input channels.

ISaGRAF - WDEMO\_26 - I/O connection

File Edit Tools Options Help

0 1 2 3 4 5 6 7 8 9 10 11 12

ref = 8091A

NO\_OR\_NC

EMG (\* Eng

LS11 (\* Left

LS14 (\* Right

ORG1 (\* Original

XSTOP (\* Stop

LS21 (\* Left

LS24 (\* Right

ORG2 (\* Original

YSTOP (\* Stop

The “NO\_OR\_NC” parameter can be set as  
0: Normal Open  
1: Normal close

Input Channel:  
CH1 : EMG, emergency stop  
CH2 : /FFEF, FIFO is empty or not, TRUE: empty  
CH3 : /FFFF, FIFO is full or not, TRUE: full  
CH4 : LS11, Left limit switch of X-axis  
CH5 : LS14, Right limit switch of X-axis  
CH6 : ORG1, Original position switch of X-axis  
CH7 : XSTOP, Stop or not of X-axis, TRUE: stop  
CH8 : LS21, Left limit switch of Y-axis  
CH9 : LS24, Right limit switch of Y-axis  
CH10 : ORG2, Original position switch of Y-axis  
CH11 : YSTOP, Stop or not of Y-axis, TRUE: stop

I-8090 contains 3 analog input channels.

ISaGRAF - TEST01 - I/O connection

File Edit Tools Options Help

0 1 2 3 4 5 6 7 8 9 10 11 12

ref = 8090

x\_mode = 10

y\_mode = 10

z\_mode = 10

Parameter:  
x\_mode : integer counting mode of X-axis  
y\_mode : integer counting mode of Y-axis  
z\_mode : integer counting mode of Z-axis  
00: quadrant counting mode  
10: CW/CCW counting mode  
20: pulse/direction counting mode

Input Channel:  
CH1 : encorder value of X-axis  
CH2 : encorder value of Y-axis  
CH3 : encorder value of Z-axis

CH1 to CH3 are signed 32-bit integer format

## Setting commands:

### M\_regist                      Register one I-8091

In order to distinguish more than one I-8091 card in I-8417/8817/8437/8837 platform, the I-8091 cards should be registrated before using it. This command will assign a card number = "card\_NO\_" to I-8091 card at that "address\_". If there is no I-8091 at the given address, this command will return FALSE.



**Note: If using "I\_8091A" rather than "I\_8091" on the I/O connection window, user don't need to call "m\_regist" & "m\_s\_nc", they are ignored. The card\_NO of "I-8091A" is equal to its slot No. I-8xx7: 0 ~ 7. W-8xx7: 1 ~ 7.**

Parameters:

|          |         |  |
|----------|---------|--|
| card_NO_ | integer | valid is 0 ~ 19.                           |
| address_ | integer | the plugged slot address of the i8091 card |

slot 0: 16#80  
slot 1: 16#A0  
slot 2: 16#C0  
slot 3: 16#E0  
slot 4: 16#140  
slot 5: 16#160  
slot 6: 16#180  
slot 7: 16#1A0

Return:

|    |         |                        |
|----|---------|------------------------|
| Q_ | boolean | TRUE: Ok , FALSE: Fail |
|----|---------|------------------------|

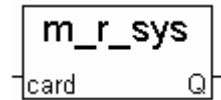
Example: I-8417/8817/8437/8837: demo\_46, demo\_27, demo\_28  
W-8337/8737: wdemo\_26, wdemo\_27, wdemo\_28, wdemo\_29

```
(* declaration: INIT as boolean <internal> and has initial value of TRUE *)
TMP as boolean <internal>
(* cardNO as integer <internal> and has initial value of 1 *)
(* Do some init setting at 1st scan cycle *)
if INIT then
    INIT := FALSE;
    TMP := M_regist(cardNO,16#80); (* plug i8091 in slot 0 *)
    TMP := M_r_sys(cardNO);        (* reset i8091's setting *)
    TMP := M_s_var(cardNO,4,2,5,100);
    TMP := M_s_dir(cardNO,0,0);    (* Normal direction *)
    TMP := M_s_mode(cardNO,1,1);  (* pulse_dir mode *)
    TMP := M_s_serv(cardNO,1,1);  (* X & Y server ON *)
    TMP := M_s_nc(cardNO,0);      (* Normal open *)
end_if;
```



## **M\_r\_sys                  Reset all setting**

To reset I-8091 card, this command will terminate the running command in I-8091 card. User can use this command as software emergency stop. This command also will clear all of setting, so, all I-8091 card's parameter should be set again.



Parameters:

card\_NO\_      integer                  the card No. has been set by **M\_regist**, valid is 0 ~ 19

Return:

Q\_                  boolean                  always return TRUE.

Example: I-8417/8817/8437/8837: demo\_46, demo\_27, demo\_28

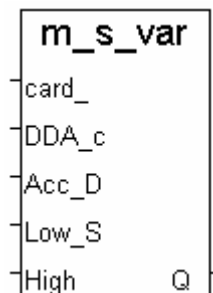
W-8xx7: wdemo\_26, wdemo\_27, wdemo\_28, wdemo\_29

## M\_s\_var Set motion system parameters

To set DDA cycle, accelerating/decelerating speed, low speed and high speed value.

Parameters:

|             |         |  |
|-------------|---------|--|
| card_NO_    | integer | the card No. has been set by <b>M_regist</b> , valid is 0 ~ 19 |
| DDA_cycle_  | integer | DDA cycle , valid is 1 ~ 254                                   |
| Acc_Dec_    | integer | Acc/Dec speed , valid is 1 ~ 200                               |
| Low_Speed_  | integer | low speed , valid is 1 ~ 200 , Low_Speed_ >= Acc_Dec_          |
| High_Speed_ | integer | high speed , Low_Speed_ <= High_Speed <= 2047                  |

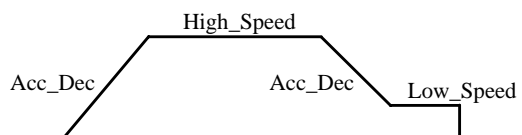


Return:

|    |         |                     |
|----|---------|---------------------|
| Q_ | boolean | always return TRUE. |
|----|---------|---------------------|

Note:

The lower “DDA\_cycle\_” is given, the smaller delay time between /ORG1 ON and /X\_STOP ON (or /ORG2 ON and /Y\_STOP ON) when using M\_hsporg & M\_lsporg command. For ex, DDA\_cycle\_ set to 4, the delay time is about 5 to 13 ms.



Restriction:

$1 \leq DDA\_cycle \leq 254$   
 $1 \leq Acc\_Dec \leq 200$   
 $1 \leq Low\_Speed \leq 200$   
 $Low\_Speed \leq High\_Speed \leq 2047$   
 $Low\_Speed \geq Acc\_Dec$

Default value

DDA\_cycle = 10  
Acc\_Dec = 1  
Low\_Speed = 10  
High\_Speed = 100

Example: I-8417/8817/8437/8837: demo\_46, demo\_27, demo\_28  
W-8xx7: wdemo\_26, wdemo\_27, wdemo\_28, wdemo\_29

```

TMP := M_s_var(1, 5, 2, 10, 150);
(* DDA_cycle = 5    --> DDA period = (5+1)*1.024ms = 6.144ms
  Acc_Dec = 2      --> Acc/Dec speed = 2/(6.144ms)^2 = 52981 p/s^2
  Low_Speed = 10   --> low speed = 10/6.144ms = 1628pps
  High_Speed = 150 --> high speed = 150/6.144ms = 24414pps *)

```

**M\_s\_dir                      Define output direction of axes**

Sometimes, the output direction of X-axis, Y-axis is undesired direction due to the motor’s connection or gear train. In order to unify the output direction as shown in Fig.(5) and Fig.(6). Where CW/FW direction is defined as toward outside from motor, CCW/BW direction is defined as toward inside from motor. This command provide parameters to define the rotating direction of motor.



Parameters:

|          |         |  |
|----------|---------|--|
| card_NO_ | integer | the card No. has been set by <b>M_regist</b> , valid is 0 ~ 19 |
| defdirX_ | integer | X axis direction definition , valid is 0 ~ 1                   |
| defdirY_ | integer | Y axis direction definition , valid is 0 ~ 1                   |

0: normal direction, 1: reverse direction

Return:

|    |         |                     |
|----|---------|---------------------|
| Q_ | boolean | always return TRUE. |
|----|---------|---------------------|

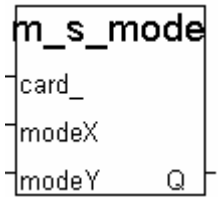
Example: I-8417/8817/8437/8837: demo\_46, demo\_27, demo\_28  
W-8xx7: wdemo\_26, wdemo\_27, wdemo\_28, wdemo\_29

**M\_s\_mode Set output mode**

Parameters:

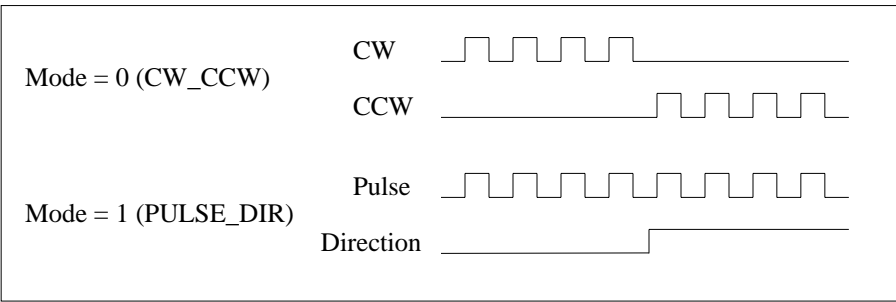
|          |         |  |
|----------|---------|--|
| card_NO_ | integer | the card No. has been set by <b>M_regist</b> , valid is 0 ~ 19 |
| modeX_   | integer | X axis mode, valid is 0 ~ 1                                    |
| modeY_   | integer | Y axis mode, valid is 0 ~ 1                                    |

0: CW\_CCW, 1: PULSE\_DIR



Return:

|    |         |                     |
|----|---------|---------------------|
| Q_ | boolean | always return TRUE. |
|----|---------|---------------------|



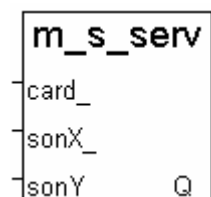
Example: I-8417/8817/8437/8837: demo\_46, demo\_27, demo\_28  
W-8xx7: wdemo\_26, wdemo\_27, wdemo\_28, wdemo\_29

## **M\_s\_serv            Set servo ON/OFF**

Parameters:

|          |         |  |
|----------|---------|--|
| card_NO_ | integer | the card No. has been set by <b>M_regist</b> , valid is 0 ~ 19 |
| sonX_    | integer | X axis servo/hold on switch , valid is 0 ~ 1                   |
| sonY_    | integer | Y axis servo/hold on switch , valid is 0 ~ 1                   |

0: OFF, 1: ON



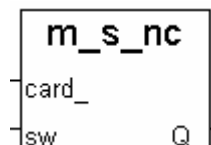
Return:

|    |         |                     |
|----|---------|---------------------|
| Q_ | boolean | always return TRUE. |
|----|---------|---------------------|

Example: I-8417/8817/8437/8837: demo\_46, demo\_27, demo\_28  
W-8xx7: wdemo\_26, wdemo\_27, wdemo\_28, wdemo\_29

## **M\_s\_nc            Set N.O. / N.C.**

To set all of the following limit switches as N.C.(normal close) or N.O.(normal open). If set as N.O., those limit switches are active low. If set as N.C., those limit switches are active high. The auto-protection will automatically change the judgement whatever it is N.O. or N.C..



Limit switches: ORG1, LS11, LS14, ORG2, LS21, LS24, EMG.

**Note: If using “I\_8091A” rather than “I\_8091” on the I/O connection window, user don’t need to call “m\_regist” & “m\_s\_nc”, they are ignored. The card\_NO of “I-8091A” is equal to its slot No. I-8xx7: 0 ~ 7. W-8xx7: 1 ~ 7.**

Parameters:

|          |         |  |
|----------|---------|--|
| card_NO_ | integer | the card No. has been set by <b>M_regist</b> , valid is 0 ~ 19 |
| sw_      | integer | 0: N.O. (default) , 1: N.C.                                    |

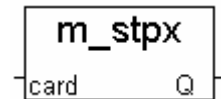
Return:

|    |         |                     |
|----|---------|---------------------|
| Q_ | boolean | always return TRUE. |
|----|---------|---------------------|

Example: I-8417/8817/8437/8837: demo\_46, demo\_27, demo\_28  
W-8xx7: wdemo\_26, wdemo\_27, wdemo\_28, wdemo\_29

## Stop commands:

### **M\_stpx**                      **Stop X axis**



Parameters:

card\_NO\_      integer              the card No. has been set by **M\_regist**, valid is 0 ~ 19

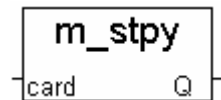
Return:

Q\_              boolean              always return TRUE.

Example: I-8417/8817/8437/8837: demo\_46, demo\_27, demo\_28

W-8xx7: wdemo\_26, wdemo\_27, wdemo\_28, wdemo\_29

### **M\_stpy**                      **Stop Y axis**



Parameters:

card\_NO\_      integer              the card No. has been set by **M\_regist**, valid is 0 ~ 19

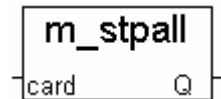
Return:

Q\_              boolean              always return TRUE.

Example: I-8417/8817/8437/8837: demo\_46, demo\_27, demo\_28

W-8xx7: wdemo\_26, wdemo\_27, wdemo\_28, wdemo\_29

### **M\_stpall**                      **Stop X & Y axes**



This command will stop X & Y axes and clear all of commands pending in the FIFO.

Parameters:

card\_NO\_      integer              the card No. has been set by **M\_regist**, valid is 0 ~ 19

Return:

Q\_              boolean              always return TRUE.

Example: I-8417/8817/8437/8837: demo\_46, demo\_27, demo\_28

W-8xx7: wdemo\_26, wdemo\_27, wdemo\_28, wdemo\_29

## Simple motion commands:

### M\_lsporg Low speed move to ORG

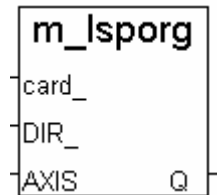
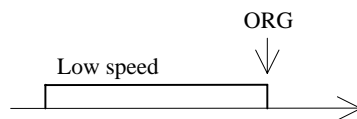
Low speed move , and stop when **ORG1/ORG2** limit switch is touched.

Parameters:

|          |         |  |
|----------|---------|--|
| card_NO_ | integer | the card No. has been set by <b>M_regist</b> , valid is 0 ~ 19 |
| DIR_     | integer | 0: CW , 1: CCW   |
| AXIS_    | integer | 1: X axis , 2: Y axis  |

Return:

|    |         |                     |
|----|---------|---------------------|
| Q_ | boolean | always return TRUE. |
|----|---------|---------------------|



### M\_hsporg High speed move to ORG

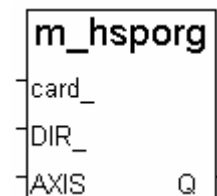
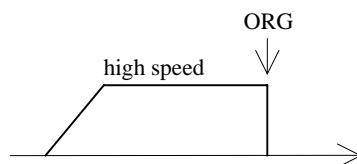
High speed move , and stop when **ORG1/ORG2** limit switch is touched.

Parameters:

|          |         |  |
|----------|---------|--|
| card_NO_ | integer | the card No. has been set by <b>M_regist</b> , valid is 0 ~ 19 |
| DIR_     | integer | 0: CW , 1: CCW   |
| AXIS_    | integer | 1: X axis , 2: Y axis  |

Return:

|    |         |                     |
|----|---------|---------------------|
| Q_ | boolean | always return TRUE. |
|----|---------|---------------------|



Example: I-8417/8817/8437/8837: demo\_46, demo\_27, demo\_28

W-8xx7: wdemo\_26, wdemo\_27, wdemo\_28, wdemo\_29

Note:

The lower "DDA\_cycle\_" is given, the smaller delay time between /ORG1 ON and /X\_STOP ON (or /ORG2 ON and /Y\_STOP ON) when using M\_hsporg & M\_lsporg command. For ex, DDA\_cycle\_ set to 4, the delay time is about 5 to 13 ms.

## M\_lsppmv Low speed pulse move

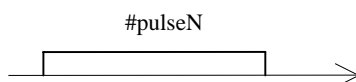
Low speed move a specified “pulse”

Parameters:

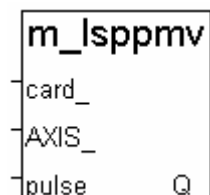
|          |         |  |
|----------|---------|--|
| card_NO_ | integer | the card No. has been set by <b>M_regist</b> , valid is 0 ~ 19                             |
| AXIS_    | integer | 1: X axis , 2: Y axis  |
| Pulse_   | integer | number of pulse to move. if > 0, move toward CW/FW dir.<br>if < 0, move toward CCW/BW dir. |

Return:

|    |         |                     |
|----|---------|---------------------|
| Q_ | boolean | always return TRUE. |
|----|---------|---------------------|



Example: I-8417/8817/8437/8837: demo\_46, demo\_27, demo\_28  
W-8xx7: wdemo\_26, wdemo\_27, wdemo\_28, wdemo\_29



## M\_hsppmv High speed pulse move

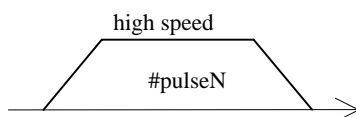
High speed move a specified “pulse”

Parameters:

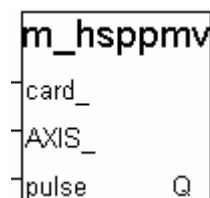
|          |         |  |
|----------|---------|--|
| card_NO_ | integer | the card No. has been set by <b>M_regist</b> , valid is 0 ~ 19                             |
| AXIS_    | integer | 1: X axis , 2: Y axis  |
| Pulse_   | integer | number of pulse to move. if > 0, move toward CW/FW dir.<br>if < 0, move toward CCW/BW dir. |

Return:

|    |         |                     |
|----|---------|---------------------|
| Q_ | boolean | always return TRUE. |
|----|---------|---------------------|



Example: I-8417/8817/8437/8837: demo\_46, demo\_27, demo\_28  
W-8xx7: wdemo\_26, wdemo\_27, wdemo\_28, wdemo\_29

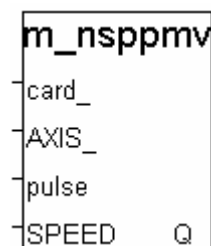


## M\_nsppmv      Normal speed pulse move

Normal speed move a specified “pulse”

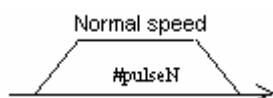
Parameters:

|          |         |  |
|----------|---------|--|
| card_NO_ | integer | the card No. has been set by <b>M_regist</b> , valid is 0 ~ 19                             |
| AXIS_    | integer | 1: X axis , 2: Y axis  |
| Pulse_   | integer | number of pulse to move. if > 0, move toward CW/FW dir.<br>if < 0, move toward CCW/BW dir. |
| SPEED_   | integer | Speed, low speed <= SPEED_ <= high speed   |



Return:

|    |         |                     |
|----|---------|---------------------|
| Q_ | boolean | always return TRUE. |
|----|---------|---------------------|



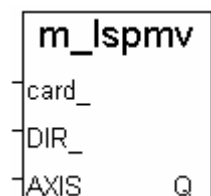
Example: I-8417/8817/8437/8837: demo\_46, demo\_27, demo\_28  
W-8xx7: wdemo\_26, wdemo\_27, wdemo\_28, wdemo\_29

## M\_lspmv      Low speed move

Low speed move toward the direction specified. It can be stop by **M\_stpx** or **M\_stpy** or **M\_stpall** command

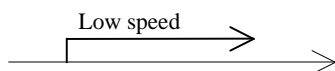
Parameters:

|          |         |  |
|----------|---------|--|
| card_NO_ | integer | the card No. has been set by <b>M_regist</b> , valid is 0 ~ 19 |
| DIR_     | integer | direction. 0: CW , 1: CCW                                      |
| AXIS_    | integer | 1: X axis , 2: Y axis  |



Return:

|    |         |                     |
|----|---------|---------------------|
| Q_ | boolean | always return TRUE. |
|----|---------|---------------------|

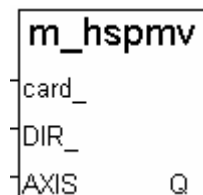


Example: I-8417/8817/8437/8837: demo\_46, demo\_27, demo\_28  
W-8xx7: wdemo\_26, wdemo\_27, wdemo\_28, wdemo\_29



## M\_hspmv High speed move

High speed move toward the direction specified. It can be stop by **M\_stpx** or **M\_stpy** or **M\_stpall** command

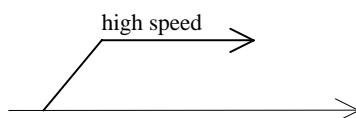


Parameters:

|          |         |  |
|----------|---------|--|
| card_NO_ | integer | the card No. has been set by <b>M_regist</b> , valid is 0 ~ 19 |
| DIR_     | integer | direction. 0: CW , 1: CCW                                      |
| AXIS_    | integer | 1: X axis , 2: Y axis  |

Return:

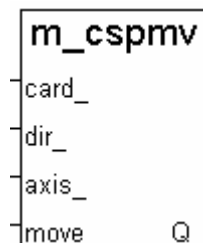
|    |         |                     |
|----|---------|---------------------|
| Q_ | boolean | always return TRUE. |
|----|---------|---------------------|



Example: I-8417/8817/8437/8837: demo\_46, demo\_27, demo\_28  
W-8xx7: wdemo\_26, wdemo\_27, wdemo\_28, wdemo\_29

## M\_cspmv Change speed move

This command will accelerate/decelerate the selected axis's motor to the "move\_speed". This command can be continuously send to I-8091 to dynamicly change speed. The rotating motor can be stop by the command **M\_stpx**, **M\_stpy**, **M\_stpall**, or **M\_slwstp**

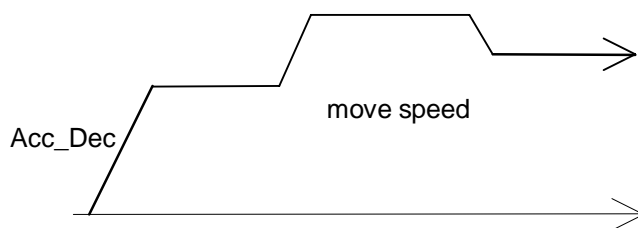


Parameters:

|             |         |  |
|-------------|---------|--|
| card_NO_    | integer | the card No. has been set by <b>M_regist</b> , valid is 0 ~ 19 |
| dir_        | integer | direction. 0: CW , 1: CCW                                      |
| axis_       | integer | 1: X axis , 2: Y axis  |
| move_speed_ | integer | 0 < move_speed_ <= 2040  |

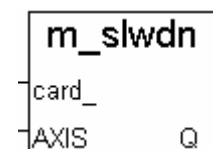
Return:

|    |         |                     |
|----|---------|---------------------|
| Q_ | boolean | always return TRUE. |
|----|---------|---------------------|



Example: I-8417/8817/8437/8837: demo\_46, demo\_27, demo\_28  
W-8xx7: wdemo\_26, wdemo\_27, wdemo\_28, wdemo\_29

## M\_slwdn      Slow down to low speed



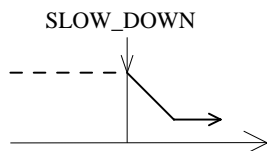
To decelerate to slow speed until **M\_stpx** or **M\_stpy** or **M\_stpall** is executed.

Parameters:

|          |         |  |
|----------|---------|--|
| card_NO_ | integer | the card No. has been set by <b>M_regist</b> , valid is 0 ~ 19 |
| AXIS_    | integer | 1: X axis , 2: Y axis  |

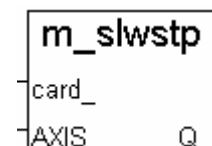
Return:

|    |         |                     |
|----|---------|---------------------|
| Q_ | boolean | always return TRUE. |
|----|---------|---------------------|



Example: I-8417/8817/8437/8837: demo\_46, demo\_27, demo\_28  
W-8xx7: wdemo\_26, wdemo\_27, wdemo\_28, wdemo\_29

## M\_slwstp      Slow down to stop



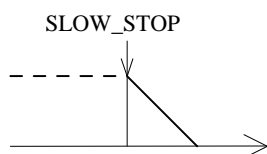
To decelerate to stop.

Parameters:

|          |         |  |
|----------|---------|--|
| card_NO_ | integer | the card No. has been set by <b>M_regist</b> , valid is 0 ~ 19 |
| AXIS_    | integer | 1: X axis , 2: Y axis  |

Return:

|    |         |                     |
|----|---------|---------------------|
| Q_ | boolean | always return TRUE. |
|----|---------|---------------------|

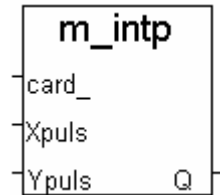


Example: I-8417/8817/8437/8837: demo\_46, demo\_27, demo\_28  
W-8xx7: wdemo\_26, wdemo\_27, wdemo\_28, wdemo\_29

## Interpolation commands:

### M\_intp                      Move a short distance on X-Y plane

This command will move a short distance (interpolation short line) on X-Y plane. This command provided a method for user to generate an arbitrary curve on X-Y plane.

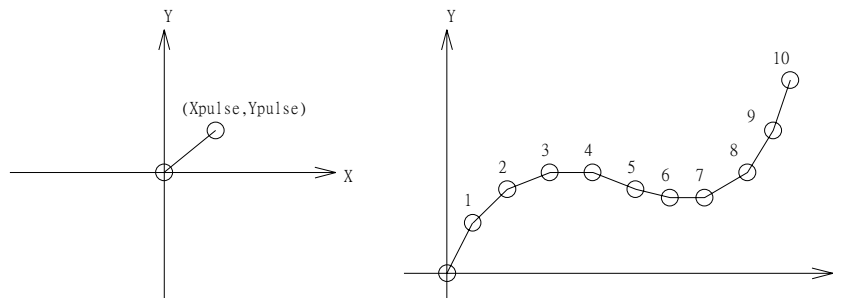


Parameters:

|          |         |  |
|----------|---------|--|
| card_NO_ | integer | the card No. has been set by <b>M_regist</b> , valid is 0 ~ 19 |
| Xpulse_  | integer | -2047 <= Xpulse_ <= 2047                                       |
| Ypulse_  | integer | -2047 <= Ypulse_ <= 2047                                       |

Return:

|    |         |                     |
|----|---------|---------------------|
| Q_ | boolean | always return TRUE. |
|----|---------|---------------------|



Example: I-8417/8817/8437/8837: demo\_46, demo\_27, demo\_28

W-8xx7: wdemo\_26, wdemo\_27, wdemo\_28, wdemo\_29

NOTE:

For a lot of **M\_intp** call set at the same time, please check if the FIFO is not full. Call it if FIFO is not full. FIFO indicator is a Digital Input resides at CH3 of i-8091.

i-8091 D/I channel on ISaGRAF I/O connection window:

CH1 : EMG, emergency stop

CH2 : /FFEF, FIFO is empty or not, TRUE: empty

**CH3 : /FFFF, FIFO is full or not, TRUE: full**

CH4 : LS11, Left limit switch of X-axis

CH5 : LS14, Right limit switch of X-axis

CH6 : ORG1, Original position switch of X-axis

CH7 : XSTOP, Stop or not of X-axis, TRUE: stop

CH8 : LS21, Left limit switch of Y-axis

CH9 : LS24, Right limit switch of Y-axis

CH10 : ORG2, Original position switch of Y-axis

CH11 : YSTOP, Stop or not of Y-axis, TRUE: stop

## M\_intln                      Move a long distance on X-Y plane

This command will move a long distance (interpolation line) on X-Y plane. The CPU on I-8091 card will generate a trapezoidal speed profile of X-axis and Y-axis, and execute interpolation by way of DDA chip.

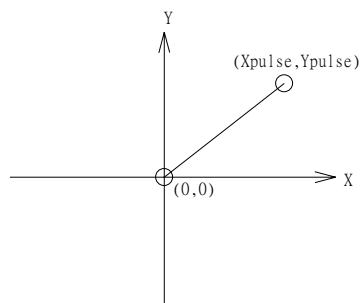


Parameters:

|          |         |  |
|----------|---------|--|
| card_NO_ | integer | the card No. has been set by <b>M_regist</b> , valid is 0 ~ 19 |
| Xpulse_  | integer | -524287 <= Xpulse_ <= 524287                                   |
| Ypulse_  | integer | -524287 <= Xpulse_ <= 524287                                   |

Return:

|    |         |                     |
|----|---------|---------------------|
| Q_ | boolean | always return TRUE. |
|----|---------|---------------------|



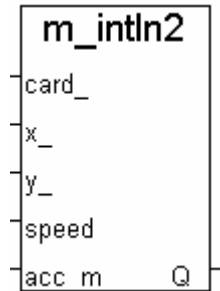
Example: I-8417/8817/8437/8837: demo\_46, demo\_27, demo\_28  
 W-8xx7: wdemo\_26, wdemo\_27, wdemo\_28, wdemo\_29

## M\_intln2                      Move a long distance on X-Y plane

This command will move a long interpolation line on X-Y plane. It will automatically generate a trapezoidal speed profile of X-axis and Y-axis by state-machine-type calculation method.

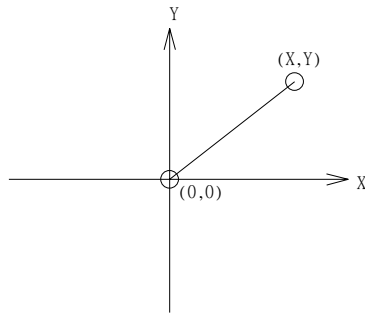
Parameters:

|           |         |   |
|-----------|---------|---|
| card_NO_  | integer | the card No. has been set by <b>M_regist</b> , valid is 0 ~ 19                              |
| x_, y_    | integer | end point relate to present position  |
| speed_    | integer | 0 ~ 2040  |
| acc_mode_ | integer | 0: enable acceleration/deceleration profile<br>1: disable acceleration/deceleration profile |



Return:

|    |         |                     |
|----|---------|---------------------|
| Q_ | boolean | always return TRUE. |
|----|---------|---------------------|



NOTE:

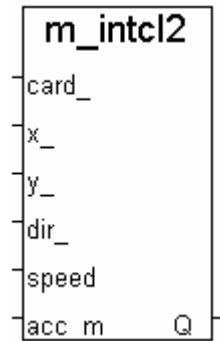
1. Only one of **M\_intln2**, **M\_intcl2** & **M\_intar2** command can be called at one time, the other motion moving commands related to the same I-8091 card should not be called unless it is completed. (Please use **M\_intstp** to test command of **M\_intln2**, **M\_intcl2** & **M\_intar2** completed or not).
2. One controller can only drive one I-8091 to move by **M\_intln2**, **M\_intcl2**, **M\_intar2** command. Two or more I-8091 cards in the same controller to use **M\_intln2**, **M\_intcl2**, **M\_intar2** at the same time is not possible.

## M\_intcl2      Move a circle on X-Y plane

This command will generate an interpolation circle on X-Y plane. It will automatically generate a trapezoidal speed profile of X-axis and Y-axis by state-machine-type calculation method.

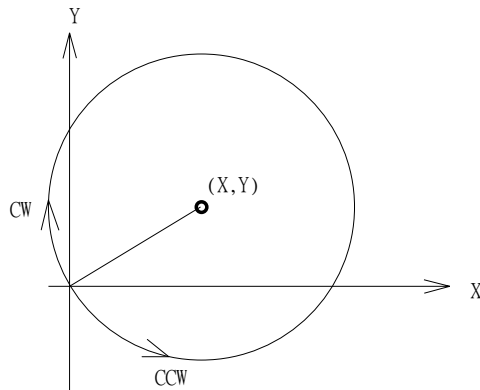
### Parameters:

|           |         |   |
|-----------|---------|---|
| card_NO_  | integer | the card No. has been set by <b>M_regist</b> , valid is 0 ~ 19                              |
| x_, y_    | integer | center point of circle relate to present position   |
| dir_      | integer | moving direction. 0: CW , 1: CCW  |
| speed_    | integer | 0 ~ 2040  |
| acc_mode_ | integer | 0: enable acceleration/deceleration profile<br>1: disable acceleration/deceleration profile |



### Return:

|    |         |                     |
|----|---------|---------------------|
| Q_ | boolean | always return TRUE. |
|----|---------|---------------------|



where radius =  $\sqrt{X^2 + Y^2}$

### NOTE:

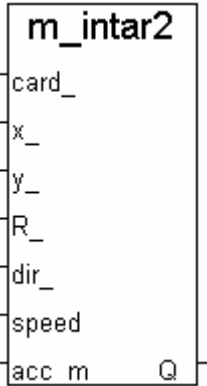
1. Only one of **M\_intln2**, **M\_intcl2** & **M\_intar2** command can be called at one time, the other motion moving commands related to the same I-8091 card should not be called unless it is completed. (Please use **M\_intstp** to test command of **M\_intln2**, **M\_intcl2** & **M\_intar2** completed or not).
2. One controller can only drive one I-8091 to move by **M\_intln2** , **M\_intcl2** , **M\_intar2** command. Two or more I-8091 cards in the same controller to use **M\_intln2** , **M\_intcl2** , **M\_intar2** at the same time is not possible.

## M\_intar2                      Move a arc on X-Y plane

This command will generate an interpolation arc on X-Y plane. It will automatically generate a trapezoidal speed profile of X-axis and Y-axis by state-machine-type calculation method.

### Parameters:

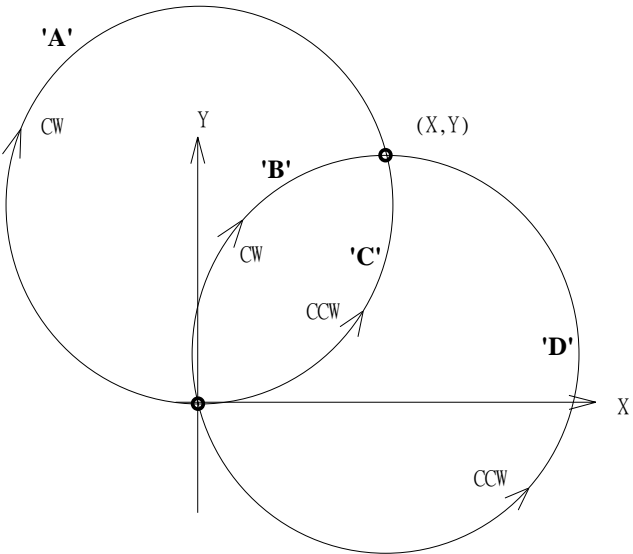
|           |         |   |
|-----------|---------|---|
| card_NO_  | integer | the card No. has been set by <b>M_regist</b> , valid is 0 ~ 19  |
| x_, y_    | integer | end point of arc relate to present position   |
| R_        | integer | radius of arc, if > 0, the arc < 180 degree, if < 0, the arc > 180 degree<br>$R\_ \text{ must } > ( \text{square root of } (X_*X_+Y_*Y_) ) / 2$ |
| dir_      | integer | moving direction. 0: CW , 1: CCW  |
| speed_    | integer | 0 ~ 2040  |
| acc_mode_ | integer | 0: enable acceleration/deceleration profile<br>1: disable acceleration/deceleration profile   |



### Return:

|    |         |                     |
|----|---------|---------------------|
| Q_ | boolean | always return TRUE. |
|----|---------|---------------------|

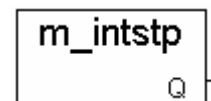
| R   | dir | path of curve |
|-----|-----|---------------|
| R>0 | CW  | 'B'           |
| R>0 | CCW | 'C'           |
| R<0 | CW  | 'A'           |
| R<0 | CCW | 'D'           |



### NOTE:

1. Only one of **M\_intln2**, **M\_intcl2** & **M\_intar2** command can be called at one time, the other motion moving commands related to the same I-8091 card should not be called unless it is completed. (Please use **M\_intstp** to test command of **M\_intln2**, **M\_intcl2** & **M\_intar2** completed or not).
2. One controller can only drive one I-8091 to move by **M\_intln2** , **M\_intcl2** , **M\_intar2** command. Two or more I-8091 cards in the same controller to use **M\_intln2** , **M\_intcl2** , **M\_intar2** at the same time is not possible.

## **M\_intstp                      Test X-Y plane moving command**



To test the below 3 commands completed or not.

**M\_intln2 , M\_intcL2 , M\_intar2**

It will return FALSE for interpolation command completed while return TRUE for busy - not completed yet.

Return:

Q\_      boolean      TRUE: busy , FALSE: completed

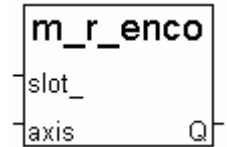
NOTE:

1. Only one of **M\_intln2**, **M\_intcL2** & **M\_intar2** command can be called at one time, the other motion moving commands related to the same I-8091 card should not be called unless it is completed. (Please use **M\_intstp** to test command of **M\_intln2**, **M\_intcL2** & **M\_intar2** completed or not).
2. One controller can only drive one I-8091 to move by **M\_intln2** , **M\_intcL2** , **M\_intar2** command. Two or more I-8091 cards in the same controller to use **M\_intln2** , **M\_intcL2** , **M\_intar2** at the same time is not possible.



## I-8090 encorder commands:

### **M\_r\_enco**            **Reset I-8090's encorder value to 0**



Parameters:

|       |         |  |
|-------|---------|--|
| slot_ | integer | the slot No. where the i8090 is plugged, 0 ~ 7 |
| axis_ | integer | 1: x-axis, 2: y-axis, 3: z-axis                |

Return:

|    |         |                     |
|----|---------|---------------------|
| Q_ | boolean | always return TRUE. |
|----|---------|---------------------|

Example: demo\_27, demo\_28, demo\_46

## Chapter 19. Ethernet Communication and Security

I-8437-80, I-8837-80, I-8437/8837, iPAC-8447/8847, I-7188EG, uPAC-7186EG and W-8xx7 has built in high flow-rate protection in the Ethernet communication. This protect the ISaGRAF program running well when TCP SYN, TCP FIN, ... flood attack happens.

### 19.1: Ethernet Security

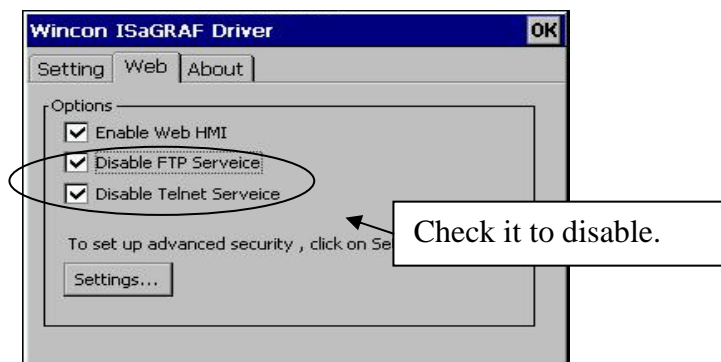
There are some ways user can get access to the Wincon-8xx7 via its ethernet port.

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

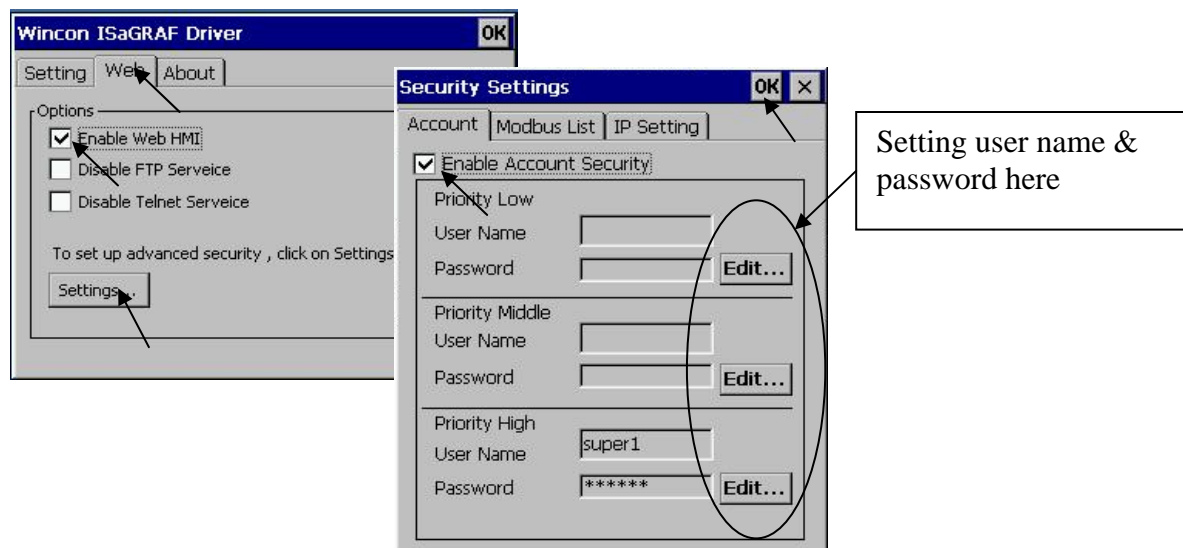
**Note:** 1. While for I-8437-80 / 8837-80, I-8437/8837, iPAC-8447/8847, uPAC-7186EG & I-7188EG, only item 1 is possible.

2. If the controller is W-8x47/8x46, when using “ftp”, “telnet”, “Web HMI” & “Modbus TCP/IP”, please connect your PC/HMI to W-8x47/8x46’s “LAN1” port, and please use “NS-205” or “NS-208” Ethernet switch.

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

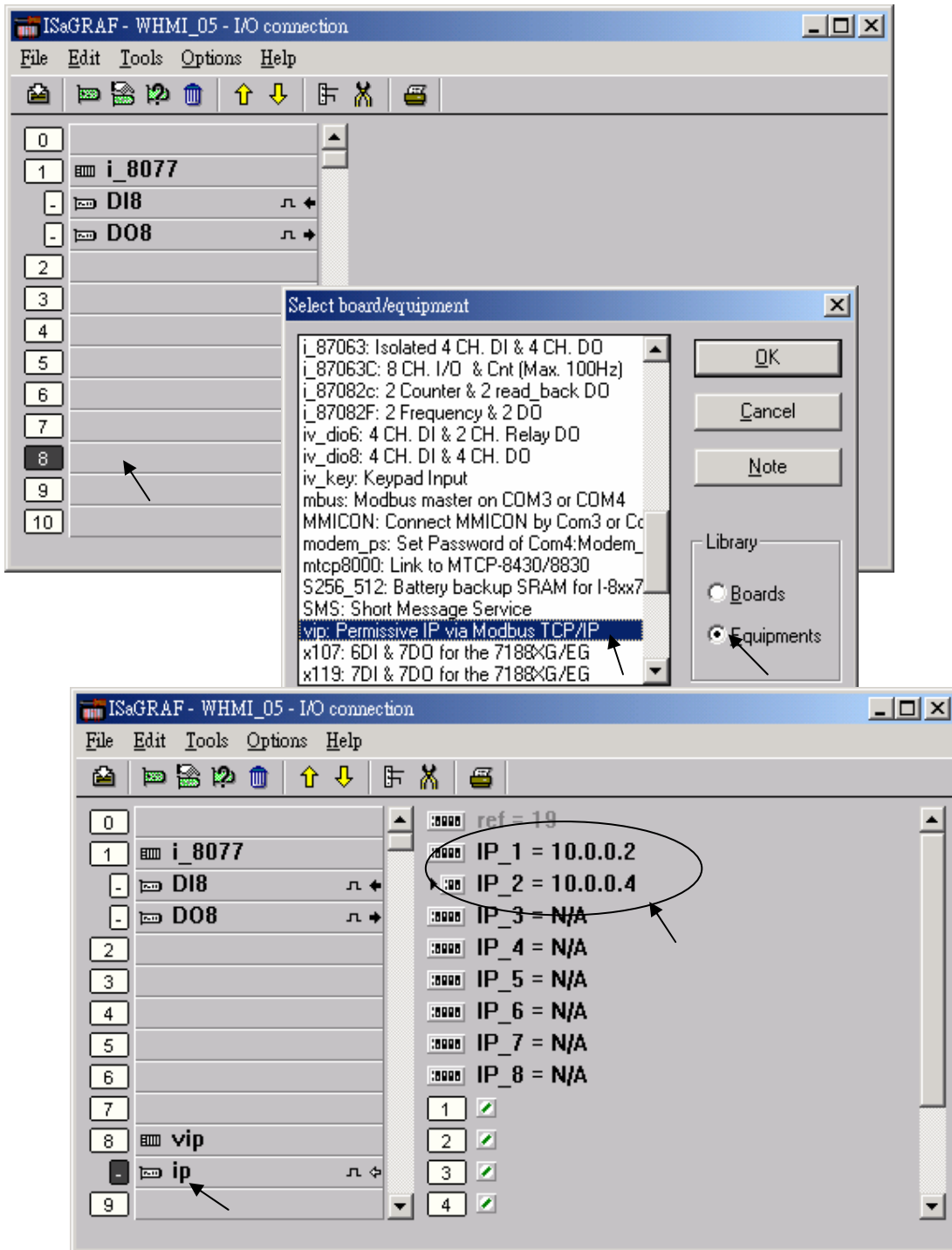


And about item 4, please set proper username & password for the Wincon Web HMI.



About item 1, user may set up to 8 IP address for ISaGRAF or other HMI to get access to the I-8x37, I-7188EG, uPAC-7186EG & W-8xx7 via the Modbus TCP/IP protocol as below.

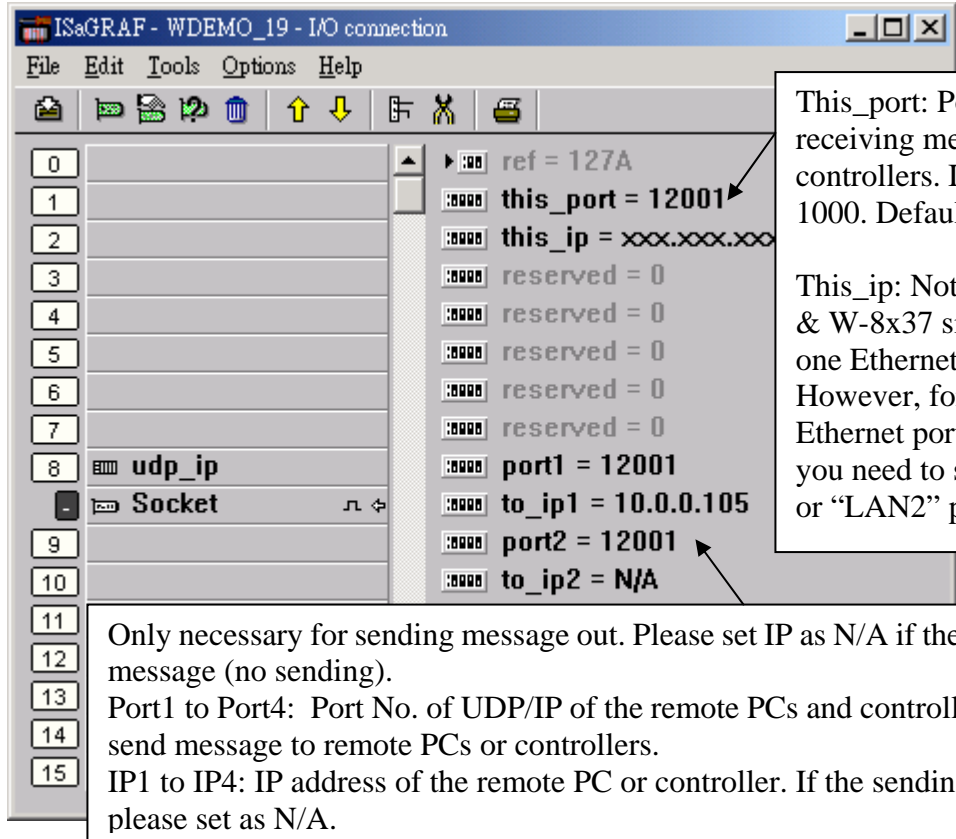
On the IO connection window of ISaGRAF, please connect “vip” and entering the IP which can get access to the controller via Modbus TCP/IP protocol. If “vip” is not connected, any remote IP can get access to your controller via Modbus TCP/IP protocol. If “vip” is connected and No IP is entered (all assigned as “N/A”), No HMI and ISaGRAF can get access to it anymore.



## 19.2: Delivering Message via UDP

When using W-8347/8747, you may connect ethernet cable at “LAN1” or “LAN2” port. Please use NS-205/208 Ethernet switch for them. (refer to Appendix F to Enable LAN2)

Since I-7188EG: driver 2.18, I-8437/8337:driver 3.20 , W-8347/8747 :driver 3.37 or later Ver. Please connect “udp\_ip” before using “udp\_recv” and “udp\_send” functions.



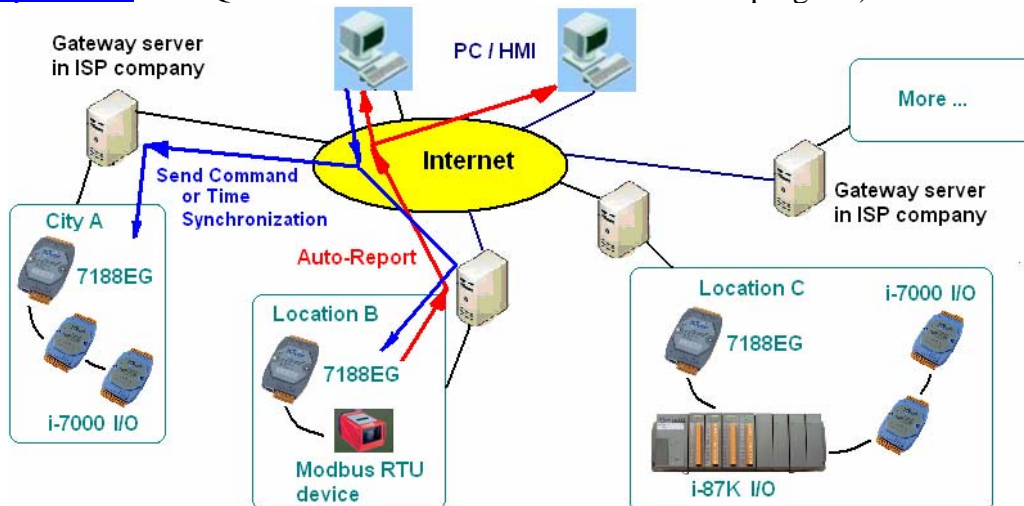
The screenshot shows the ISaGRAF - WDEMO\_19 - I/O connection window. The left pane lists I/O connections, with 'udp\_ip' selected. The right pane shows the configuration for 'udp\_ip':

- ref = 127A
- this\_port = 12001
- this\_ip = xxx.xxx.xxx
- reserved = 0 (multiple instances)
- port1 = 12001
- to\_ip1 = 10.0.0.105
- port2 = 12001
- to\_ip2 = N/A

Annotations in the image:

- Arrow pointing to 'this\_port': This\_port: Port No. of UDP/IP used for receiving message from remote PC or controllers. It is better to use value larger than 1000. Default is 12001
- Arrow pointing to 'this\_ip': This\_ip: Not necessary for I-7188EG, I-8x37 & W-8x37 since these controllers have only one Ethernet port (one IP) However, for Wincon-8x47, there is two Ethernet port (two IP) in the controller. Then you need to specify the correct IP of “LAN1” or “LAN2” port.
- Text box at the bottom: Only necessary for sending message out. Please set IP as N/A if the controller only receiving message (no sending). Port1 to Port4: Port No. of UDP/IP of the remote PCs and controllers. Max. 4 connection to send message to remote PCs or controllers. IP1 to IP4: IP address of the remote PC or controller. If the sending connection is not used, please set as N/A.

i-7188EG/uPAC-7186EG, i-8437/8837, iPAC-8447/8847 and W-8x47 can use its UDP IP to auto-report “acquisition data” or “control data” to local or remote Internet PC / HMI . The PC / HMI can also send “control commands” or “time synchronization command” to the controller. **The advantage is every Controller in different location doesn’t need a fixed “Internet IP”** (Please refer to [www.icpdas.com](http://www.icpdas.com) – FAQ – Software – ISaGRAF – 065 for demo program)



## UDP\_Recv:

To receive message from remote PCs or controllers, please use “udp\_recv” function.

For example:

```
(* test if message is coming from UDP *)
(* Msg1 is declared as Message variable *)
(* if return = " (empty message), that means no message coming *)
Msg1 := udp_recv( ) ;
```

### Note:

The receiving buffer size for Wincon is 8192 bytes - include one extra message end: 1 byte in each message. While for I-7188EG, uPAC-7186EG, iPAC-8447/8847 & I-8437/8837 is 2048 bytes. If the receiving buffer is full, the oldest received message will be overwritten.

## UDP\_Send:

To send message to remote PCs or controllers, please use “udp\_send” function.

For example:

```
(* TMP is declared as Internal / Boolean *)
(* 1st parameter: To which connection - defined in IO connection "udp_ip", can be 1 to 4 *)
(* 2nd parameter: the message to send out *)
(* Return True:Ok, False: sending buffer is full or connection not defined well in “udp_ip” *)
TMP := udp_send(1, 'Alarm1' );
```

### Note:

1. The sending buffer size for Wincon is 2048 bytes - include extra message end: 1 byte. That means max. 2048 bytes in one PLC scan can be send to remote IP.
2. Please do not send lots of bytes in one PLC scan cycle too frequent. The controller driver will actually send only one message out each PLC scan when there is message in the sending buffer. For example, if there is 100 messages in the sending buffer, the controller will send over these 100 message in 100 PLC scan cycles.
3. I-7188EG & I-8437/8837 support only “udp\_ip” & “udp\_recv”

### Example:

Please refer to Wincon CD-ROM:\napdos\isagraf\wincon\demo\wdemo\_19 & Wdemo\_20 or <ftp://ftp.icpdas.com.tw/pub/cd/winconcd/napdos/isagraf/wincon/demo/>

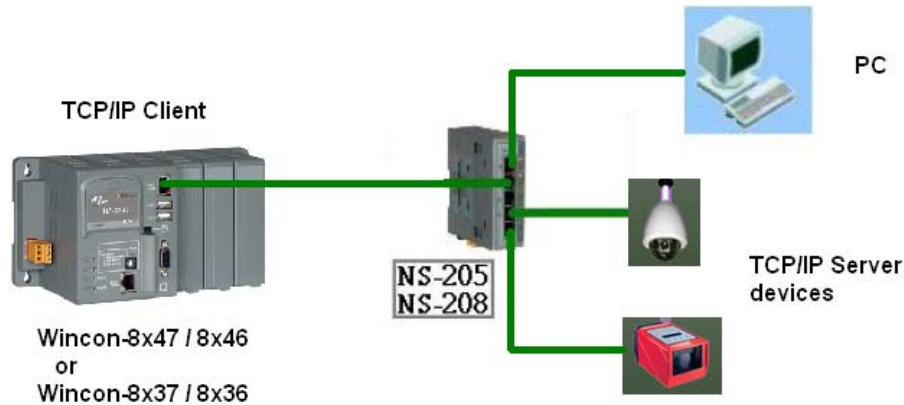
If you can not find “udp\_ip”, “udp\_recv” and “udp\_send” in your ISaGRAF, please visit <http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> to download “ICP DAS Utilities For ISaGRAF.zip”. For new driver please click “New Driver for I-8xx7, 7188EG/XG & W-8x37”

**Test Utility:** there is a useful utility “udp.exe” can be used on PC to receive message coming from UDP IP. Please run it in command shell.

W-8xx7 CD-ROM:\napdos\isagraf\some\_utility\udp\_test\udp.exe

## 19.3: Wincon Automatically Report data to Local PC or Remote Internet PC via TCP/IP . Or Wincon Communicate With Other TCP/IP Server device.

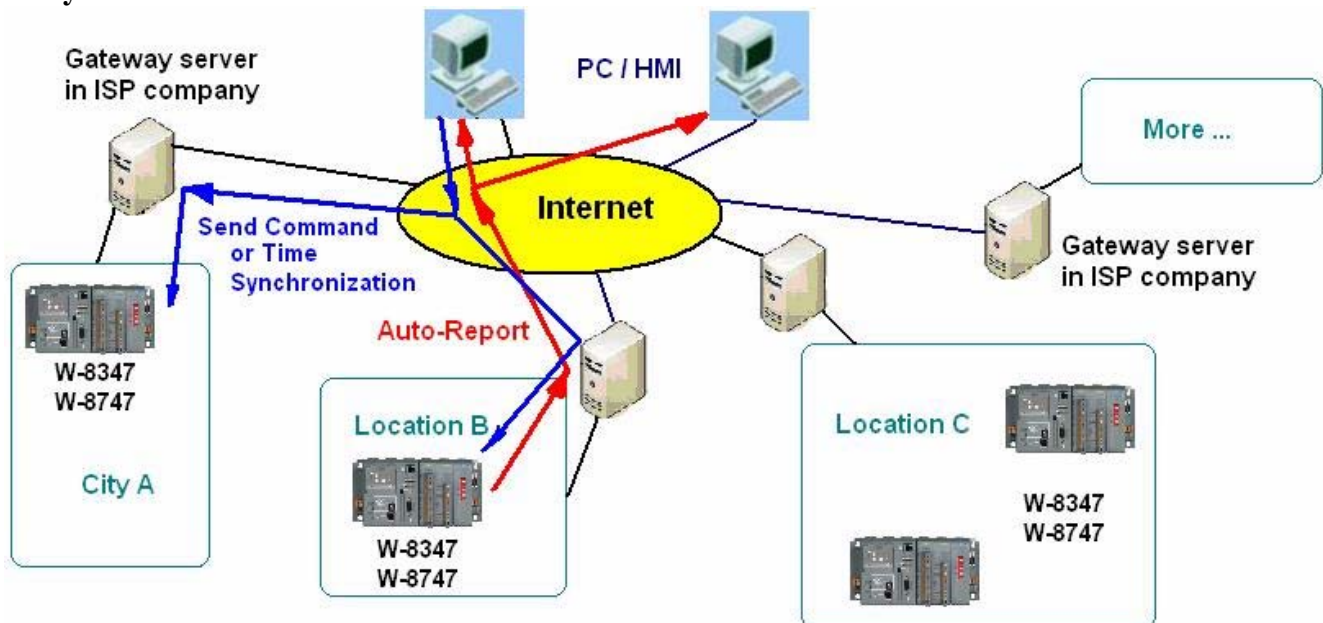
The Wincon-8xx7 / 8xx6 supports TCP/IP Client since its driver version of 3.30. Please visit <http://www.icpdas.com/products/PAC/i-8000/isagraf-link.htm> for new released driver. And please also visit <http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> to download the “ICP DAS Utilities For IsaGRAF.zip” to remove it and restore it to your ISaGRAF installed in PC. Then you will find IO connection - “Tcp\_Clie” & c-function - “Tcp\_send” & “Tcp\_recv”



The IsaGRAF demo program is Wdemo\_32.pia & Wdemo\_33.pia & Wdemo\_60.pia (It can be download at [ftp://ftp.icpdas.com/pub/cd/wincon\\_isagraf/napdos/isagraf/wincon/demo/](ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/))

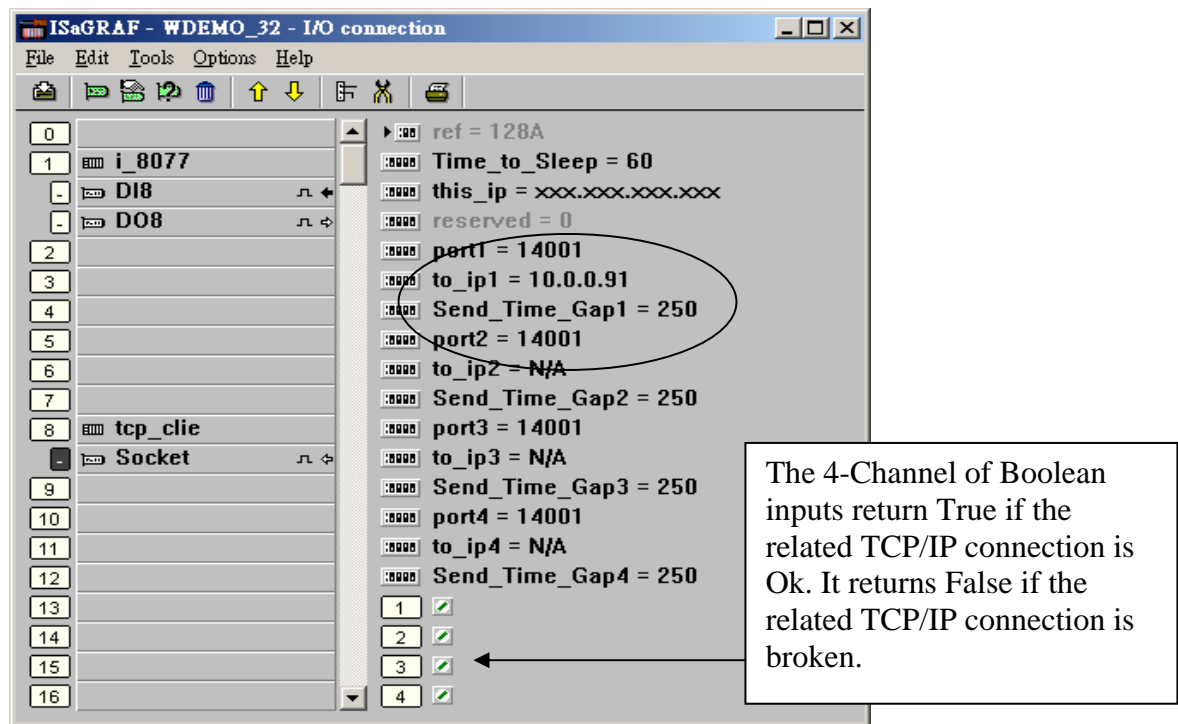
### \*\*\*\*\* Useful Application \*\*\*\*\*

(Please refer to [www.icpdas.com](http://www.icpdas.com) – FAQ – Software – ISaGRAF – 064 for demo program)  
W-8347 and W-8747 can use its TCP / IP Client to auto-report “acquisition data” or “control data” to local or remote Internet PC / HMI . The PC / HMI can send back “control commands” or “time synchronization command” to the Wincon after receiving the data from Wincon. **The advantage is every Wincon in different location doesn’t need a fixed “Internet IP” .**





To setup W-8xx7 as TCP/IP Client, please connect IO complex equipment - “Tcp\_clie” first in the IO connection windows as below. Max 4 TCP/IP Client can be setup in one Wincon-8xx7.



The “**Time\_to\_sleep**” setting ranges from 10 to 600, or 0. unit is second. If setting as 0, The TCP/IP sending connection is always connected unless the TCP/IP has communication problem. Setting as 10 to 600 sec, means if no message is sending persist for such a long time, the connection will be disconnected (TCP/IP connection will be disconnected). If there is new message requested to send, the TCP/IP connection will be connected again.

The “**this\_ip**” setting is only necessary if your Wincon is W-8x47 / W-8x46. Please enter the “IP address of the ethernet port in this controller”. For W-8x37/8x36 has only one Ethernet port (One IP address only), you don't need to specify it.

The “**port?**” specify the remote PC or device 's TCP/IP server's port number.

The “**to\_ip?**” setting default is “N/A”, which means “Not Available”. If setting an IP address to it, the related TCP/IP client will be setting up.

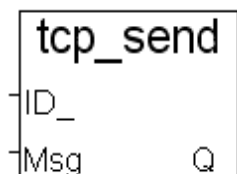
The “**Send\_Time\_Gap?**” setting is the minimum “Time Gap” in milli-second to send out each TCP/IP message one by one. For example, if setting as 250 , the 2nd message will not be sent out if the “Time Gap” since the first message sent out is less than 250 ms. The value can be 10 to 5000 ms depends on the remote device or PC 's TCP/IP server property.

After the “TCP\_clie” is well setup. Use may call “Tcp\_send” to send message out. To receive response from remote TCP/IP server, please call “Tcp\_recv”

## Tcp\_send

TCP Client send message to remote PCs or device's TCP/IP server (via ethernet)

\*\*\* Target : Wincon-8xx6/8xx7 (since driver version of 3.36)



Note:

The sending buffer for Wincon is 4096 bytes. That means max. 4096 bytes in one PLC scan can be send to remote IP. If sending buffer is full, the oldest message will be drop to release space for new "tcp\_send( )" request.

parameter :

|      |         |  |
|------|---------|--|
| ID_  | Integer | send to which connection, can be 1 to 4. The related "ip address" and "port No." is defined in "tcp_clie" (IO complex equipment) |
| Msg_ | Message | The message to send  |

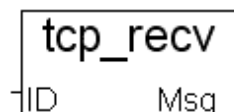
return value:

|    |         |   |
|----|---------|---|
| Q_ | Boolean | True: send OK , False: parameter error(For ex, setting ID_ as 8) or related connection is not defined in IO connection - "Tcp_clie" . |
|----|---------|---|

## Tcp\_recv

TCP Client receive message from remote PC or device's TCP/IP server (via ethernet)

\*\*\* Target : Wincon-8xx6/8xx7 (since driver version of 3.36)



Note:

The receiving buffer size is 4096 bytes. If the receiving buffer is full, the oldest message will be drop to release space for receiving new coming data.

Parameter:

|     |         |   |
|-----|---------|---|
| ID_ | Integer | send to which "Tcp_Clie" connection, can be 1 to 4. The related "ip address" and "port No." is defined in IO connection: "Tcp_clie" |
|-----|---------|---|

return value:

|      |         |  |
|------|---------|--|
| Msg_ | Message | the received message. If Msg_ = " (empty message), it means no message coming. |
|------|---------|--|

There is an useful "Tcp\_Server" tool can be run in PC to simulate the TCP/IP server device. It resides at Wincon-8xx7 CD-ROM:\napdos\isagraf\some\_utility\ (or visit

[ftp://ftp.icpdas.com/pub/cd/wincon\\_isagraf/napdos/isagraf/some\\_utility/](ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/some_utility/) to download), please copy "Tcp\_server folder to your PC", then open a command prompt, and key-in " tcp3 <port\_No> ", for example "tcp3 14001"

This utility wait Tcp/IP Client (W-8xx7) requesting to connect, and then receive message from it, and then reply a same message back to the Tcp/IP client.

The IsaGRAF demo program is Wdemo\_32.pia & Wdemo\_33.pia & Wdemo\_60.pia (It can be download at [ftp://ftp.icpdas.com/pub/cd/wincon\\_isagraf/napdos/isagraf/wincon/demo/](ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/) )



---

## Chapter 20. Redundancy Solutions

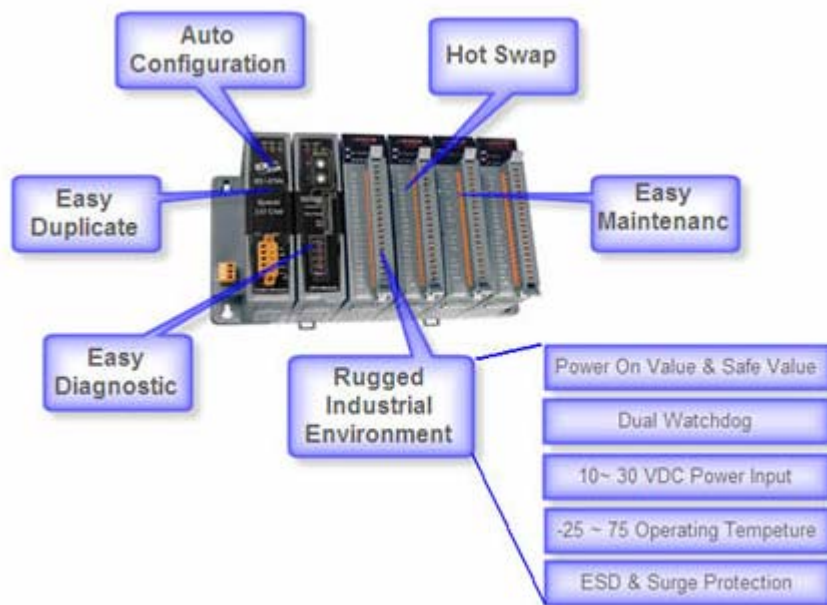
---

---

### New Products – RU-87P4 (4 slots) , RU-87P8 (8 slots) : RS-485 expansion unit

- Better than i-87K4/ i-87K8
- If using with i-87K high-profile I/O cards, it supports hot-swap
- Auto-configuration: Only need to config it once at design time, then at run time, just plug and play
- Communication Addr, Baud rate & Checksum are set by rotary and dip switch.
- All ISaGRAF controllers support them just like supporting the i-87K4/8 (please refer to Chapter 6)

More at [http://www.icpdas.com/products/io\\_expansion\\_unit/ru-87p/ru-87p4.htm](http://www.icpdas.com/products/io_expansion_unit/ru-87p/ru-87p4.htm)

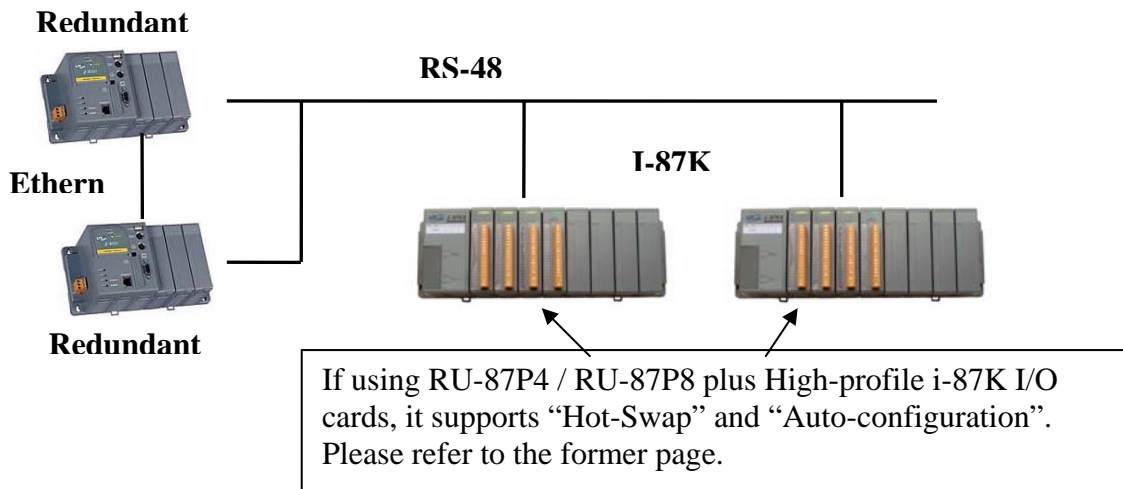


## 20.1: Wincon-8xx7 CPU Redundancy Plus I-87K I/O

### Note:

1. When using this function in controller of W-8347 / 8747, you may connect a cross ethernet cable between these two W-8x47/8x46's "LAN2" port. Then you don't need a Ethernet switch between them. (refer to Appendix F to Enable LAN2)
2. One or more PC/HMI can connect to the Modbus RTU slave RS485 port of these two redundant controllers at the same time (please refer to section 20.4). If setting Modbus RTU slave ports in COM5 to COM14 of the W-8347 / 8747 (Appendix G of the "Wincon Getting Start: ISaGRAF PAC" manual), these Modbus RTU slave ports will reply to the PC/SCADA or HMI's request only when the controller is "redundancy active". This means only one controller will response to the PC/HMI via Modbus RTU RS485 protocol at any time.

W-8347 / 8747 supports Redundancy CPU solution as below figure since driver version of 3.24. (Better to be version of 3.36 or later version)



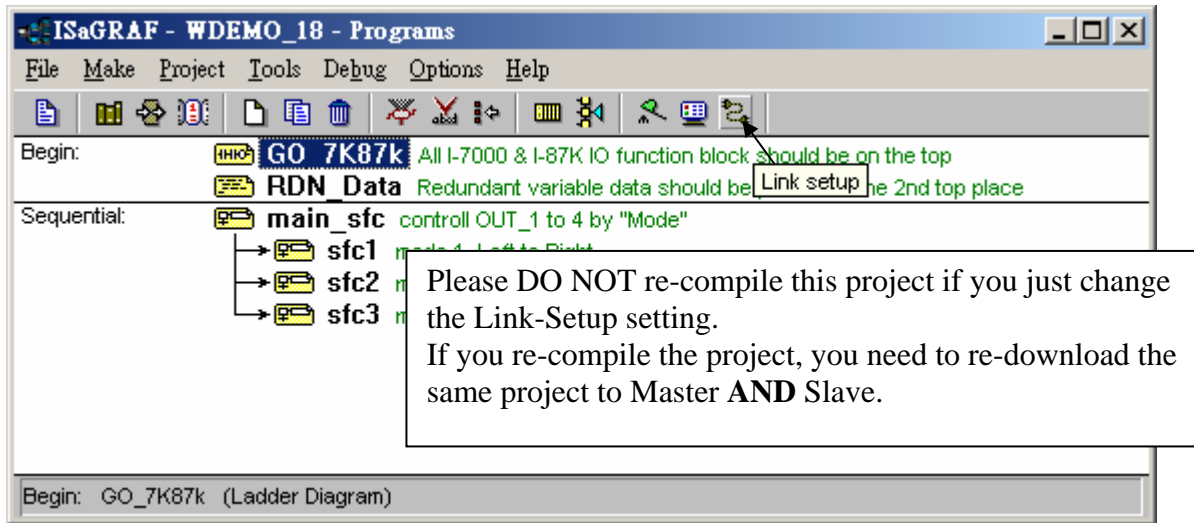
### Operations principle:

1. Two Wincons can use its COM3:RS485 to connect to one group of RS-485 remote IOs. The IO can be the I-87K4/5/8/9 extension base plus many I-87K IO modules or the I-7000 series remote IO. (Please refer to Chapter 6 for description of remote I/O)
2. **All outputs should be configured as RS-485 remote outputs**, while inputs can locate at slot 1 through slot 7 (I-8K or I-87K IO modules) or configured as RS-485 remote inputs.
3. **At least one I-7000 or I-87K Remote IO should be connected in COM3:RS485.**
4. At run time, only the Redundant Msater controller handles the RS485 command of the remote I/O. The slave controller just standby.
5. When Master controller is dead, the slave controller will take over the control to remote IO.
6. If Master is alive again, it will take back the control of remote IO .
7. The synchronous data is exchanged via the ethernet cable between the Master & slave controller. If you are using Wincon-8x47 (Wincon that has two ethernet ports), it is better to use one cross cable to link from Master controller's LAN2 port to Slave controller's LAN2 port.
8. Redundant change over time  $\leq 500\text{ms}$  , Data sychronization time  $\leq 75\text{ms}$ .

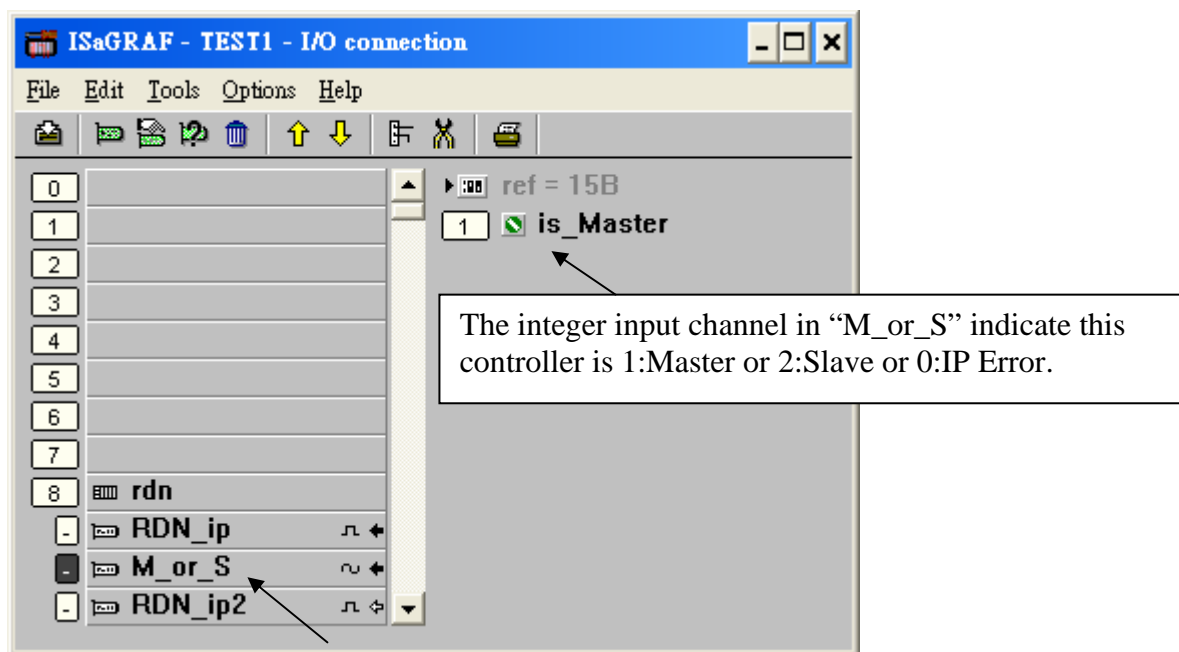
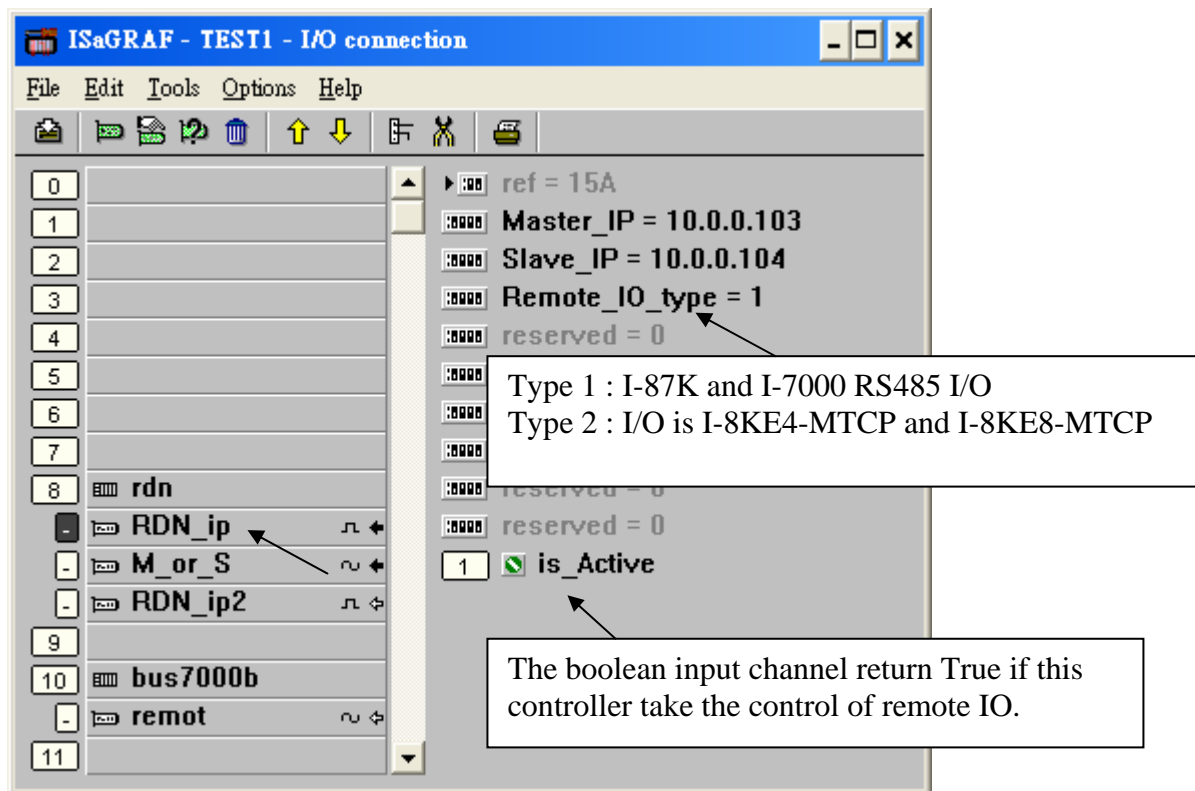
### Example program:

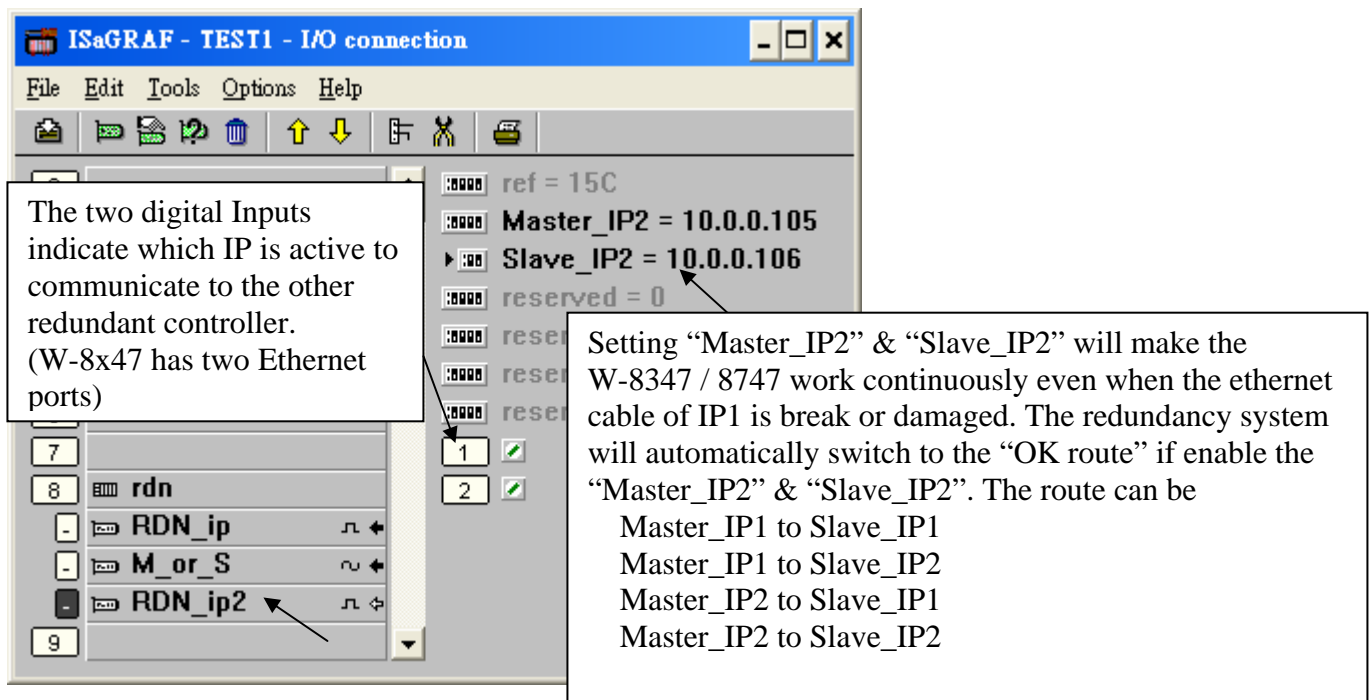
Wdemo\_18 for both Master (IP=10.0.0.103) & Slave (IP=10.0.0.104) controller.

The program in the Master and Slave controllers are identical (wdemo\_18). Please DO NOT re-compile this project if you just change the Link-Setup setting, or the project's CRC value in Master and Slave may be different (Master & Slave 's project must be the same one)

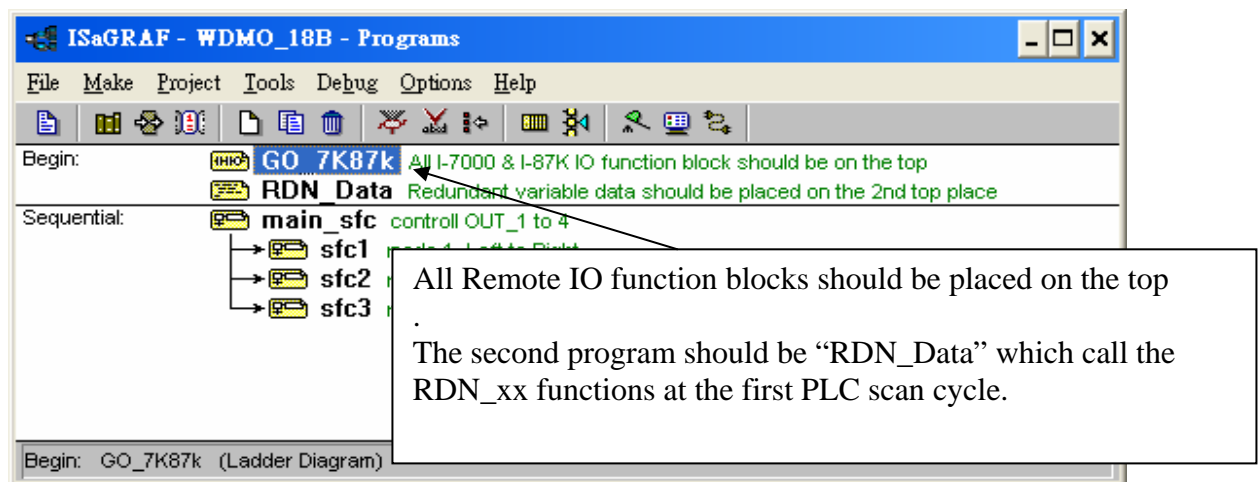


Please connect "rdn" in the IO connection window first as below. Please set the correct Master IP address and Slave IP address. For W-8x47, it is better to use IP address of the "LAN2" port. Please set "Remote\_IO\_type" to 1 if the remote IO is I-87K and I-7000 RS485 IO (**At least one Remote IO should be connected in COM3:RS485 when type=1**). (type 2 is reserved for future Modbus TCP/IP IO, not available before June.30,2006)





In the project , please must place the I-87xxx function blocks and the I-7xxx function blocks on the top. The second program should be “RDN\_Data” which call the RDN\_xx functions at the first PLC scan cycle.



All redundant synchronous data should be set in the first PLC scan cycle by using the following functions. However not necessary for the digital inputs & analog inputs in slot 1 to 7 or in the RS-485 I-7K & I-87K IO ,they are automatically updated. Only the output and other important internal data should be set as synchronous data.

RDN\_B( Boolean\_variable\_name )  
 RDN\_F( REAL\_variable\_name )  
 RDN\_N( Integer\_variable\_name )  
 RDN\_T( Timer\_variable\_name )  
 RDN\_A( for “Variable array” , please refer to Chapter 2.6 and Appendix A.4 )

For example,

```
if RDN_init then (* RDN_init is decalred with a initial value of "True" *)

    RDN_init := False ; (* only do it once *)

    (* Please set Output channels of I-7000 & I-87K IO as synchronous data *)
    (* Not necessary for Input channels of I-7000 & I-87K IO ,they are automatically updated *)
    TMP := RDN_B(OUT_1) ;
    TMP := RDN_B(OUT_2) ; (* Boolean *)
    TMP := RDN_B(OUT_3) ;
    TMP := RDN_B(OUT_4) ; (* TMP & RND_init is declared as Boolean internal variable *)
    TMP := RDN_B(OUT_5) ;
    TMP := RDN_B(OUT_6) ;
    TMP := RDN_B(OUT_7) ;
    TMP := RDN_B(OUT_8) ;

    (* set other synchronous data by using rdn_b(bool), rdn_n(integer), rdn_f(real), rdn_t (timer) *)
    TMP := RDN_N(Mode) ; (* Integer *)
    TMP := RDN_F(Real1) ; (* Real *)
    TMP := RDN_T(Timer1) ; (* Timer *)
    TMP := RDN_B(B1) ; (* Boolean *)

end_if ;
```

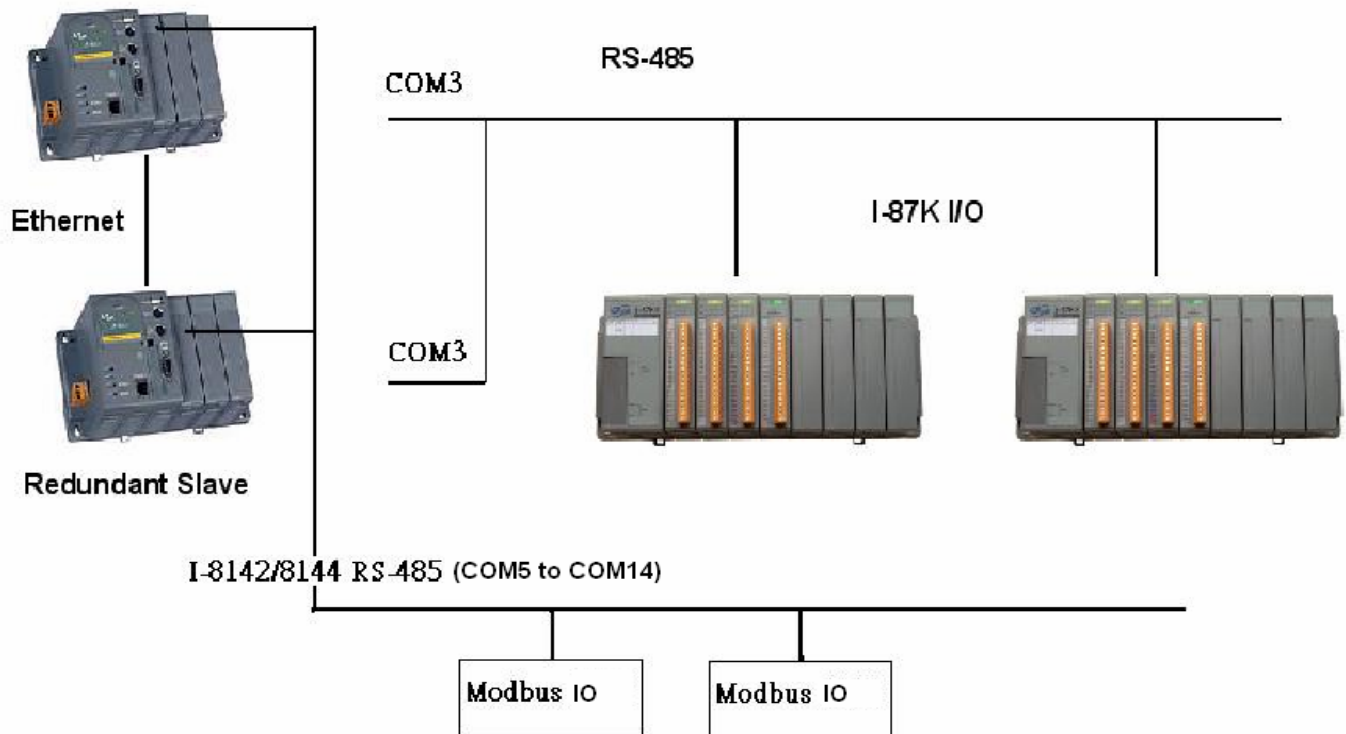
Please refer to “Wdemo\_18” in W-8xx7 CD-ROM:\napdos\isagraf\wincon\demo\ or  
<ftp://ftp.icpdas.com/pub/cd/winconcd/napdos/isagraf/wincon/demo/> “wdemo\_18.pia”

## 20.2: Wincon-8xx7 CPU Redundancy Plus I-87K I/O & Modbus RTU Devices

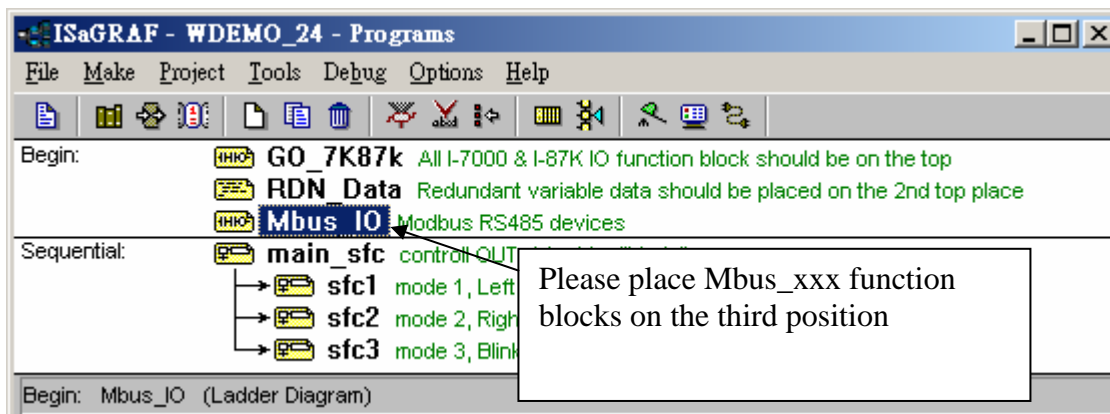
**Note:** When using this function in controller of W-8x47/8x46, it is better to connect a cross ethernet cable between these two W-8x47/8x46's "LAN2" port. Then you don't need a Ethernet switch between them. (please refer to Appendix F to Enable LAN2)

The W-8x47/8x37 Redundancy CPU solution can also support Modbus IO device as below. **At least one I-7000 or I-87K Remote IO should be connected in COM3:RS485**

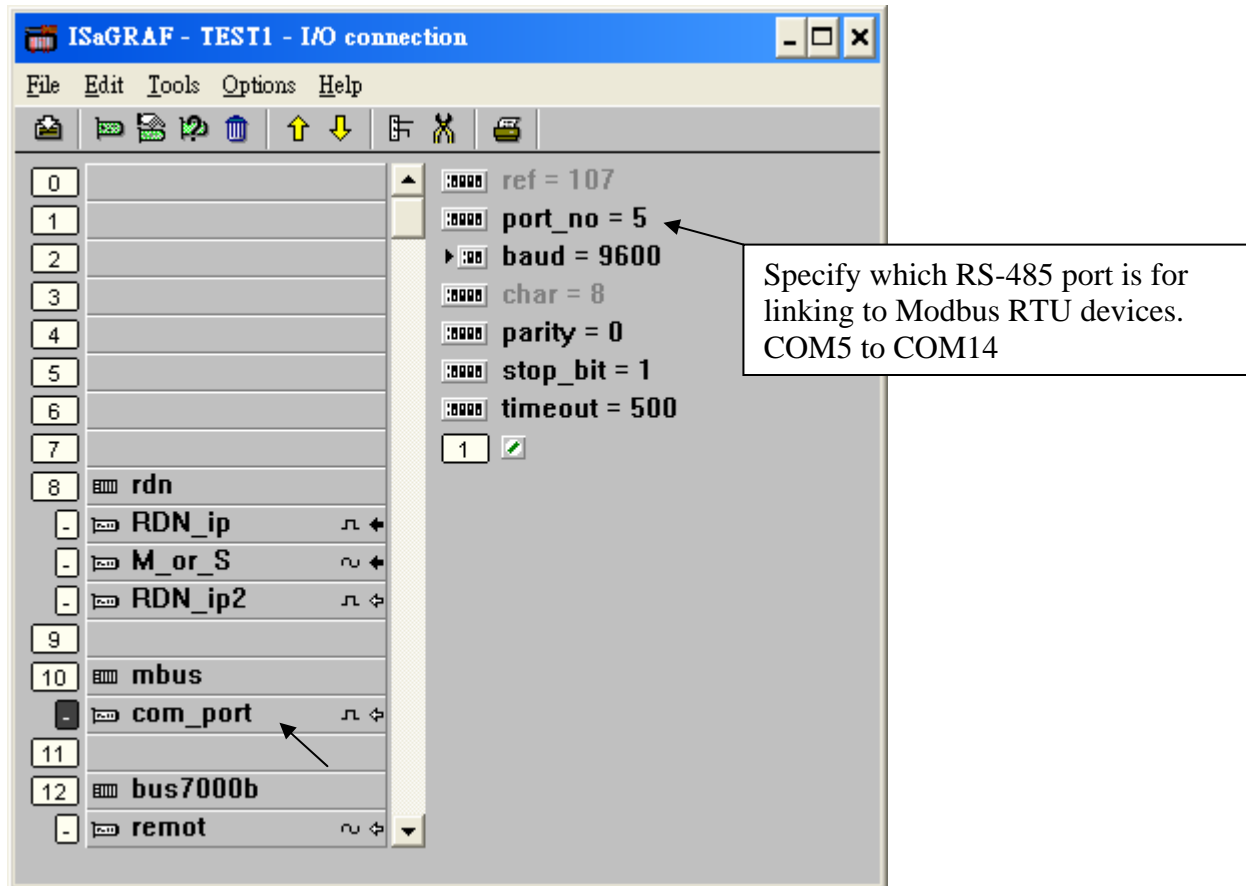
### Redundant Master



Please place Mbus\_XXX function blocks on the third position as below. Please refer to Chapter 8 for using Modbus RTU devices or Chapter 21 for connecting M-7000 I/O.



And please connect “mbus” or “mbus\_asc” in the IO connection windows.



Note:

1. Redundancy solution doesn't support Modbus RTU device in RS-232 ports since RS-232 is one-to-one connection (Two Wincons can not link to one Modbus RTU device by RS-232)
2. The Modbus device can be RTU or ASCII format listed as section 8.3.
3. Multi-ports Modbus IO can also work in redundancy solution. Please refer to section 8.4

Example:

Please refer to “Wdemo\_25” in W-8xx7 CD-ROM:\napdos\isagraf\wincon\demo\ or <ftp://ftp.icpdas.com/pub/cd/winconcd/napdos/isagraf/wincon/demo/> “wdemo\_25.pia”



## 20.3: Connecting PC / HMI to Modbus RTU RS485 ports

PC or HMI (with RS-485 Modbus RTU Master protocol supported, for example, Touch 506L / Touch 506TE) can link to the COM5 to COM14 RS-485 Modbus RTU slave port of the two redundant controllers at the same time as below. **Only the redundancy Active one will reply to the PC/HMI at any time.**

Please refer to Appendix E & G of the “Wincon Getting Start: ISaGRAF PAC” manual for setting up the Modbus RTU slave ports at COM5 to COM14. Or refer to W-8xx7

CD-ROM:\napdos\isagraf\wincon\english\_manu\“getting\_started\_w8337.pdf”

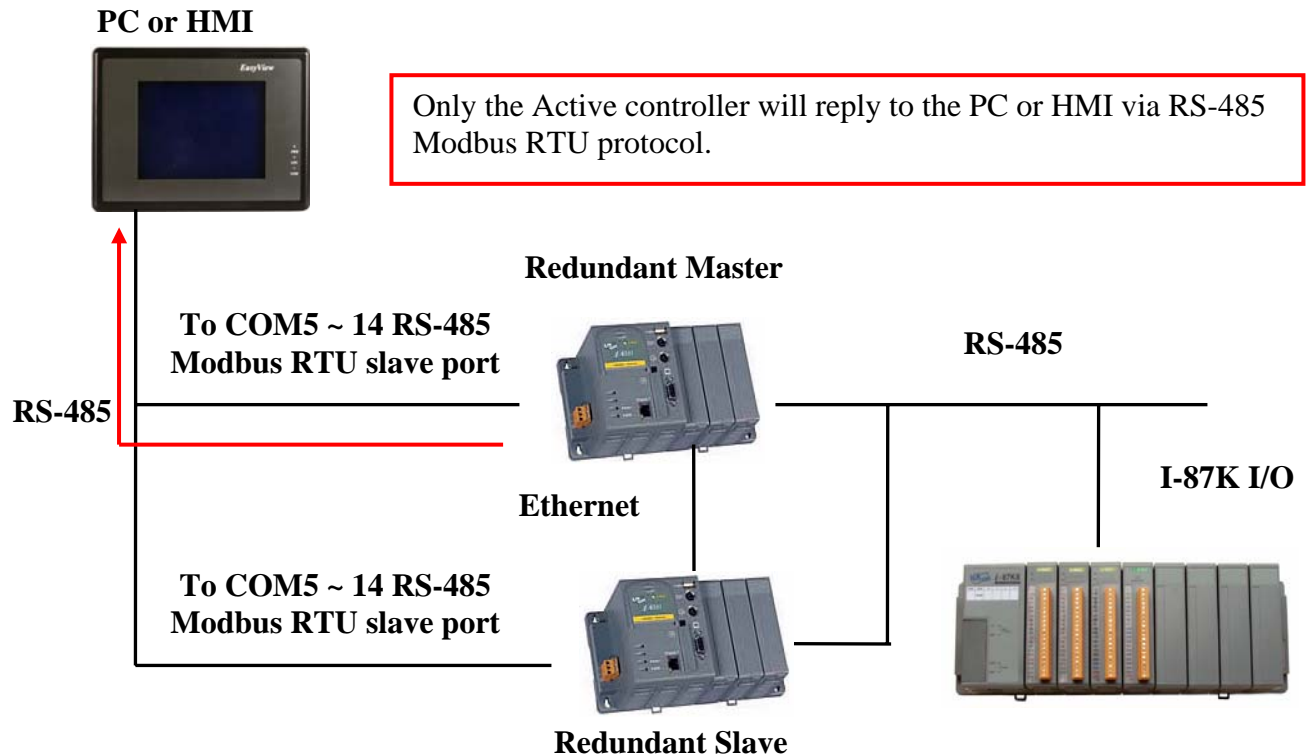
I-8142/8144/8142i: [http://www.icpdas.com/products/PAC/i-8000/8000\\_IO\\_modules.htm](http://www.icpdas.com/products/PAC/i-8000/8000_IO_modules.htm)

([http://www.icpdas.com/products/PAC/i-8000/8000\\_IO\\_modules.htm](http://www.icpdas.com/products/PAC/i-8000/8000_IO_modules.htm))

PC can connect one I-7520R converter to convert RS-232 to RS-485 and then connect to these two redundant controllers. [http://www.icpdas.com/products/Remote\\_IO/i-7000/i-7520r.htm](http://www.icpdas.com/products/Remote_IO/i-7000/i-7520r.htm)

### Important:

Please set the Net-ID of these two redundant controllers to the same number. Default is 1.



## 20.4: Connecting PC / HMI to a Redundancy system with a single IP address .

Please refer to Chapter 20.5 for setting the i-7188EX-MTCP.

Two Wincon-8347/8747 & Wincon-8346/8746 can setup as one redundancy system. The PC / HMI / SCADA software can connect to one I-7188EX-MTCP to convert Modbus TCP/IP protocol to become Modbus RTU RS-485 protocol and then to these two redundant controllers as below figures. Then the HMI software running on PC can use only one target IP address (7188EX-MTCP's IP) to link to the redundancy system. Related link:

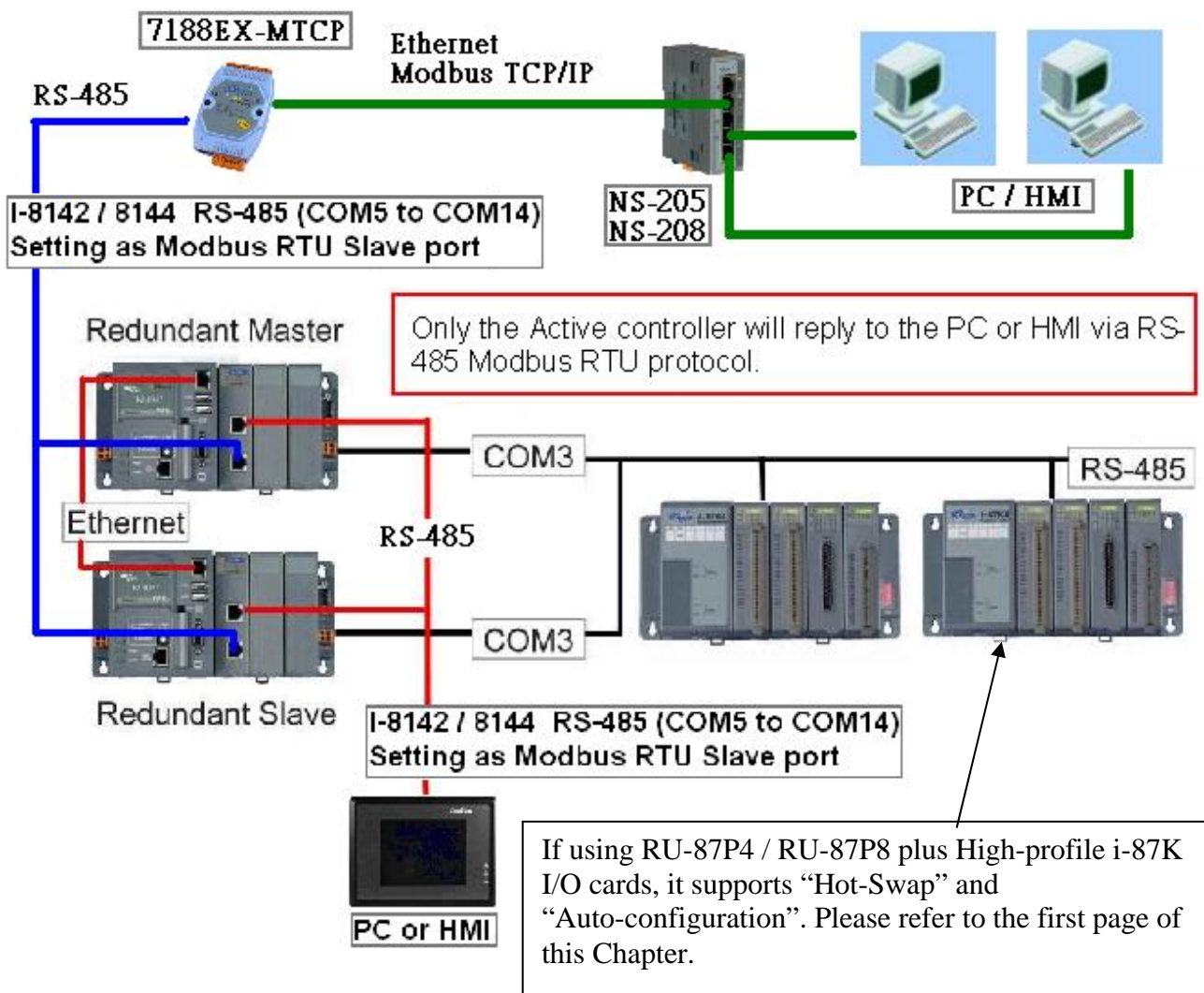
I-7188EX-MTCP : [http://www.icpdas.com/products/PAC/i-7188\\_7186/Modbus\\_PAC.htm](http://www.icpdas.com/products/PAC/i-7188_7186/Modbus_PAC.htm)

NS-205 / NS-208 : [http://www.icpdas.com/products/Switch/industrial/ethernet\\_switch.htm](http://www.icpdas.com/products/Switch/industrial/ethernet_switch.htm)

M-7000 I/O : [http://www.icpdas.com/products/Remote\\_IO/m-7000/m-7000\\_list.htm](http://www.icpdas.com/products/Remote_IO/m-7000/m-7000_list.htm)

Figure 1:

### PC / HMI Connecting to a Redundancy system with a single IP address

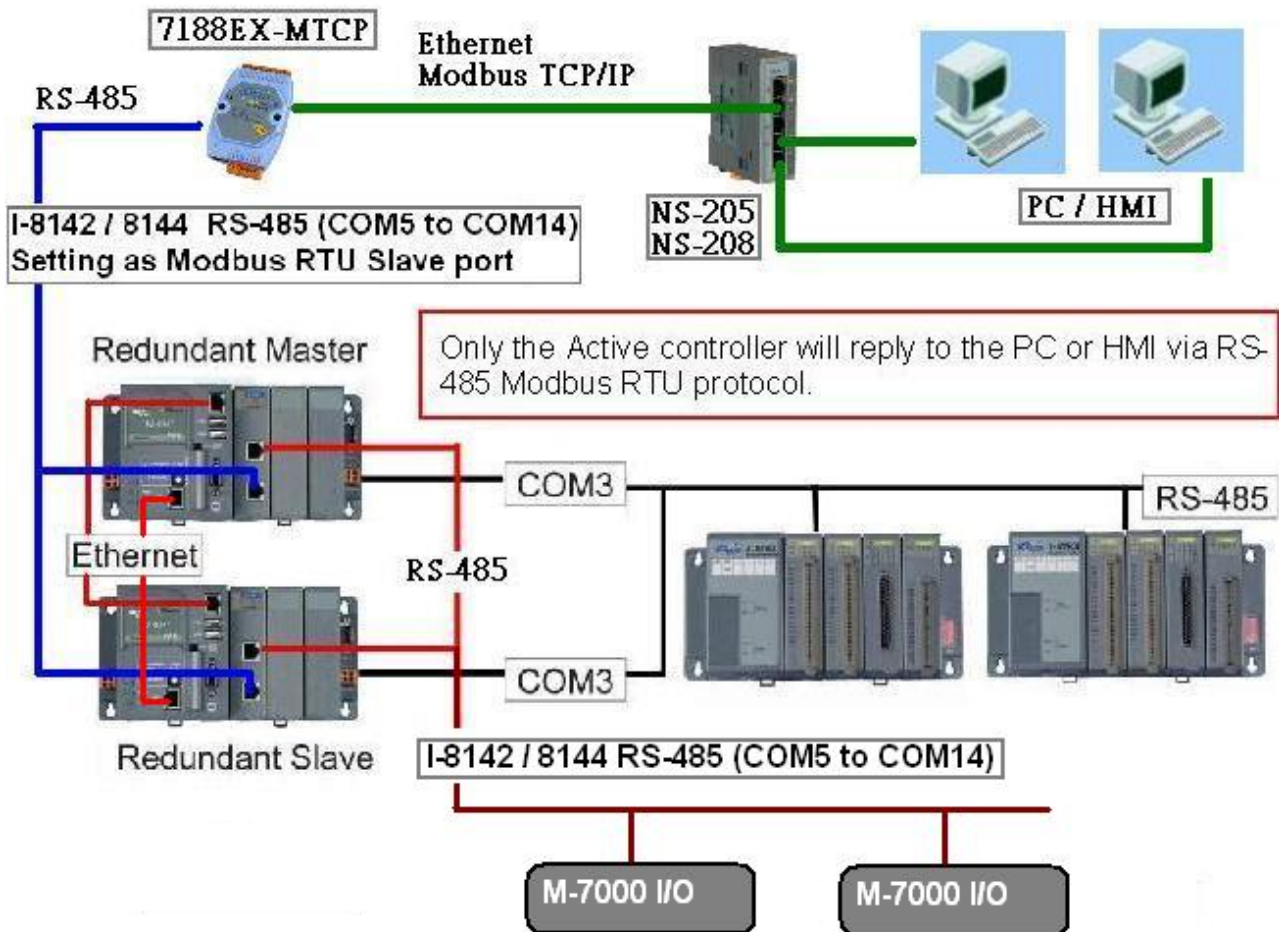


**Figure 2:**

The redundancy system supports not only I-87K I/O in I-87K4/5/8/9 base, but also M-7000 series I/O modules. [http://www.icpdas.com/products/Remote\\_IO/m-7000/m-7000\\_list.htm](http://www.icpdas.com/products/Remote_IO/m-7000/m-7000_list.htm)  
(Please refer to Chapter 21 for connecting M-7000 I/O)

Two redundant Wincon-8x47 / Wincon-8x46 can connect their LAN1 to LAN1 and also LAN2 to LAN2.  
If one cable is damaged, the other one still works.

### PC / HMI Connecting to a Redundancy system with a single IP address



**Figure 3:**

The redundancy system can support Ethernet I/O, please refer to Chapter 20.5.

## 20.5: Setting up a redundancy system with Ethernet I/O

### Step1:

The Wincon-8347 and Wincon-8747 can setup a redundancy system with Ethernet I/O as below. The demo program is “wdemo\_57.pia” at

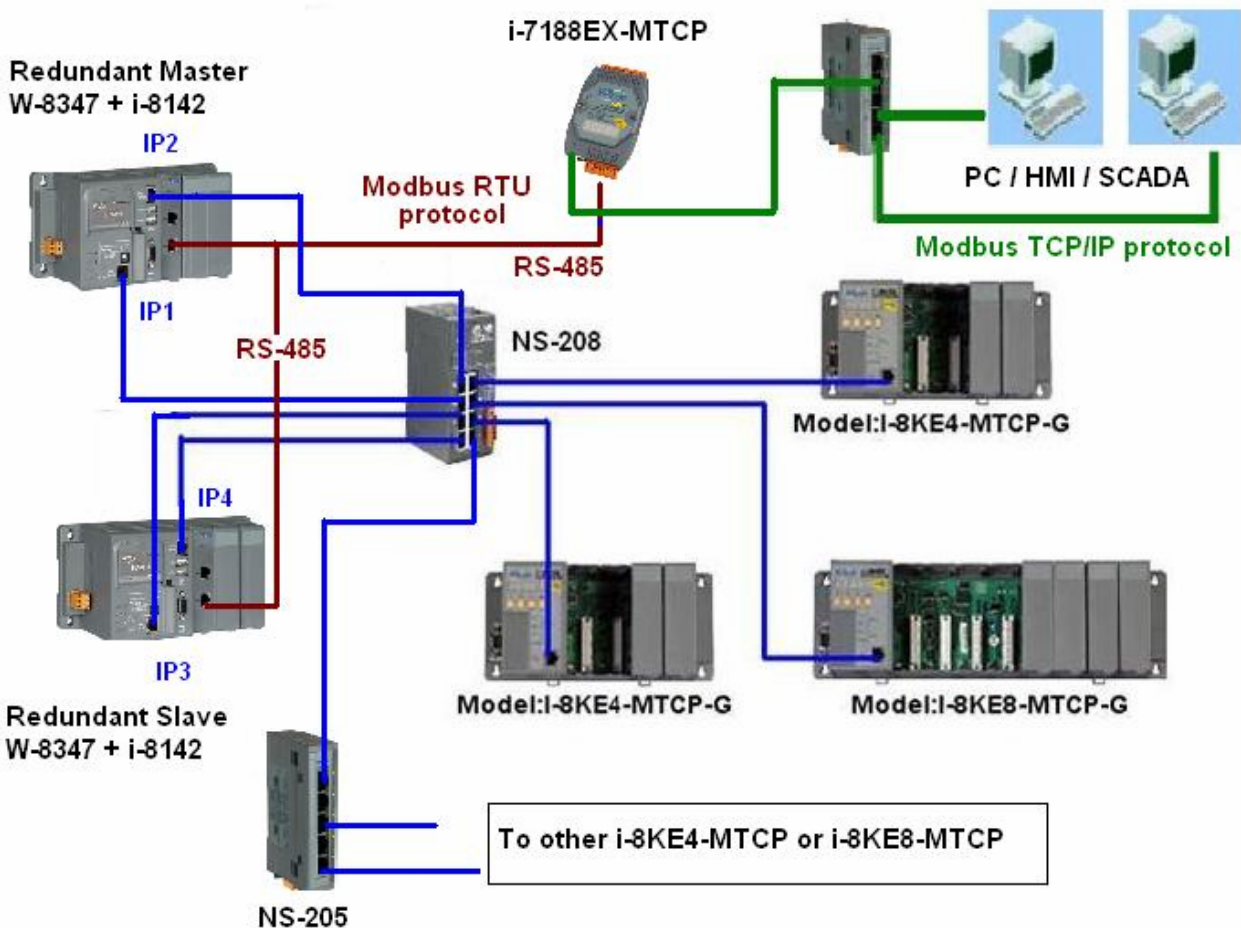
W-8xx7 CD-ROM: \napdos\isagraf\wincon\demo\ or  
[ftp://ftp.icpdas.com/pub/cd/wincon\\_isagraf/napdos/isagraf/wincon/demo/](ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/) or  
[www.icpdas.com](http://www.icpdas.com) – FAQ – Software – ISaGRAF - 062

Please refer to Chapter 22.2 to configure the Ethernet I/O – i8KE4-MTCP and i8KE8-MTCP by “Modbus utility” first or refer to FAQ-042. (It is better to enable the watchdog timer larger than 12 seconds). After all Ethernet I/O ‘s IP and configuration are configured well, please connect them as below. Then download the ISaGRAF program “wdemo\_57” to both “redundant Master” and “redundant Slave” controllers .

The advantage is

1. The scan of the ethernet I/O is much faster compared with the RS-485 i-7000 or i-87K I/O.
2. If one Ethernet cable is break or damaged, the other one will still handle the Ethernet I/O and exchange data with the other redundant controller.

PC / HMI / SCADA can connect to this redundant system with only one IP of the i-7188EX-MTCP



Wincon-8347 redundant system can connect max. 24 nodes of i-8KE4/8-MTCP I/O



Please refer to the former section for more information about the Wincon redundancy system. Or refer to [www.icpdas.com](http://www.icpdas.com) – FAQ – Software – ISaGRAF – 036)

Related link:

I-7188EX-MTCP : [http://www.icpdas.com/products/PAC/i-7188\\_7186/Modbus\\_PAC.htm](http://www.icpdas.com/products/PAC/i-7188_7186/Modbus_PAC.htm)

NS-205 / NS-208 : [http://www.icpdas.com/products/Switch/industrial/ethernet\\_switch.htm](http://www.icpdas.com/products/Switch/industrial/ethernet_switch.htm)

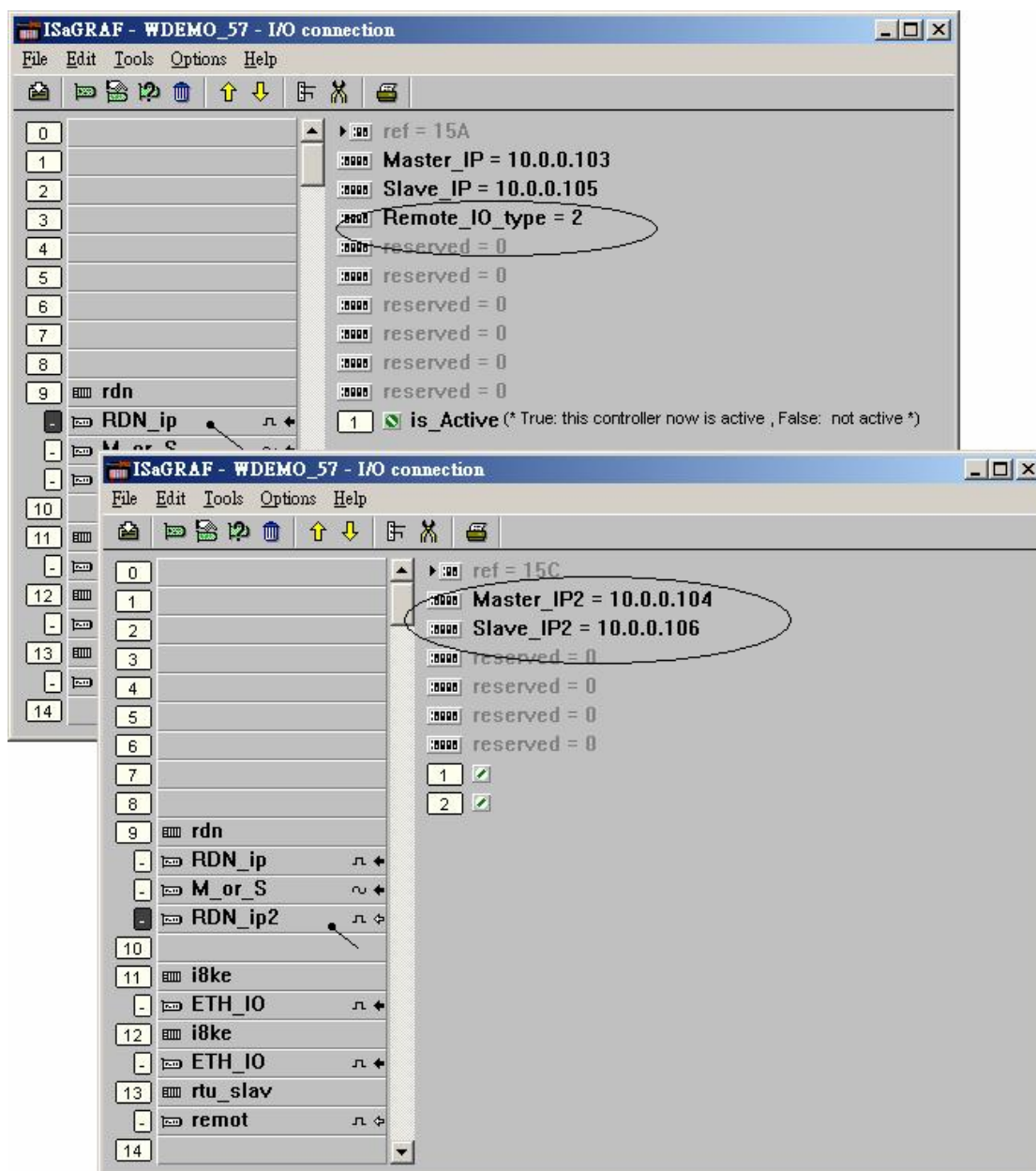
i-8142 / 8144: [http://www.icpdas.com/products/PAC/i-8000/8000\\_IO\\_modules.htm](http://www.icpdas.com/products/PAC/i-8000/8000_IO_modules.htm)

In this “wdemo\_57” demo, the ISaGRAF I/O connection window is as below.

(IP1 is 10.0.0.103 , IP2=10.0.0.104, IP3=10.0.0.105 and IP4=10.0.0.106 in this demo)

Master\_IP , Master\_IP2 , Slave\_IP and Slave\_IP2 are all enabled. (Please refer to Appendix F or Appendix D of W-8xx7 “Getting Started Manual” to enable W-8347 ’s LAN2)

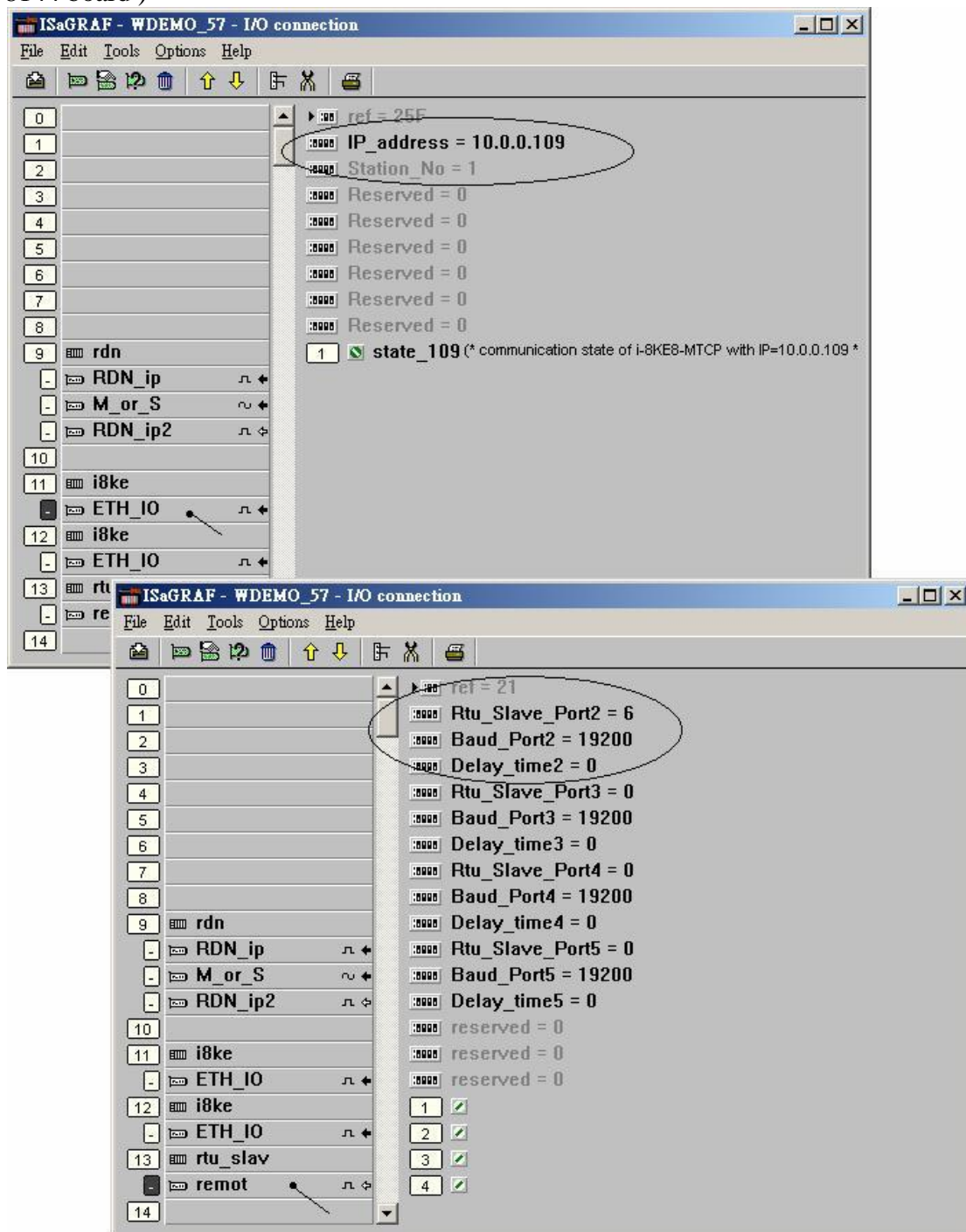
The “Remote\_IO\_type” is 2 (2 means I/O is i8KE4/8-MTCP)



There are two i8KE8-MTCP Ethernet I/O used in this demo program. One is with IP = 10.0.0.109. The other one is 10.0.0.110. If your application has more i8KE4/8-MTCP, you should connect more “i8ke” in the IO connection windows.

Please refer to Chapter 22 for more information about Ethernet I/O. (FAQ-042)

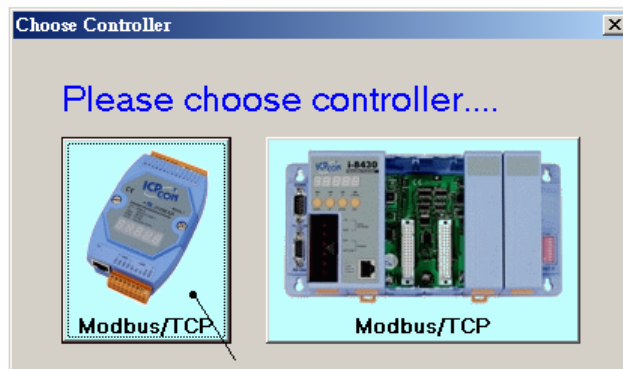
The “rtu\_slav” is to enable Wincon-8xx7 ‘s COM5 to COM8 as Modbus RTU slave port. Here we enable only COM6 in this example. (**Please refer to Appendix E of the Wincon ISaGRAF Getting Started Manual to setup expansion COM port first** . That is COM5 to COM14 in the expansion i-8142/8144 board )



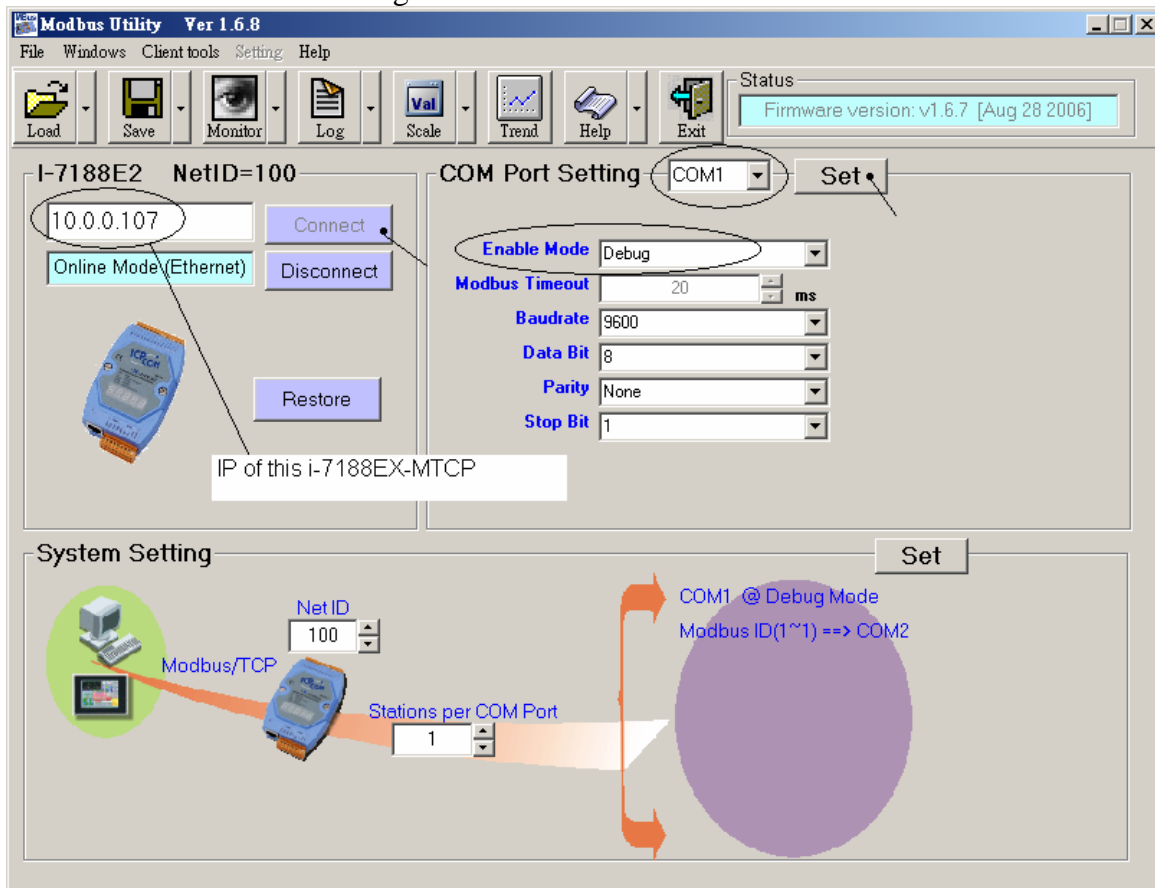
## Step2:

If you have installed i-7188EX-MTCP in this example, please set a fixed IP to it by the “7188xw.exe” utility. (Power off 7188EX-MTCP, short its “INIT\*” pin to “GND”, power it up, connecting one RS232 cable from i-7188EX-MTCP ‘s COM1 to PC ‘s COM1, PC running “7188xw.exe”, key-in “ip=10.0.0.107” to set ip address and “mask=255.255.255.0” to set mask address)

After i-7188EX-MTCP ‘s IP and Mask is well set. Please run “Modbus utility” to configure this i-7188EX-MTCP to become a Modbus TCP/IP to Modbus RTU Gateway as below steps.



Please set its COM1 as “Debug” mode.



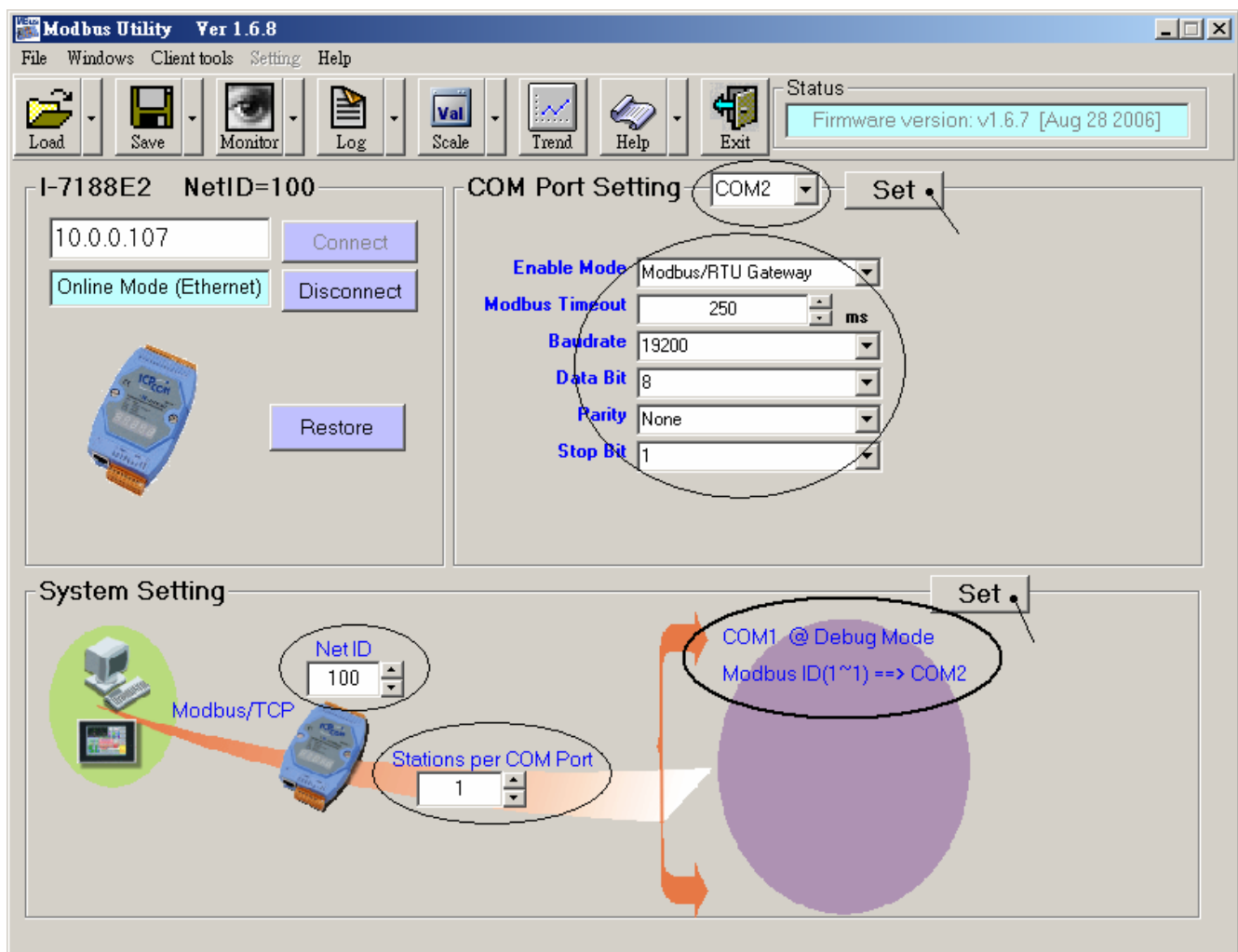
Then please set COM2 to “Modbus/RTU Gateway”, a proper timeout (250 ms), other parameters as 19200,8,None,1 . then Click on “Set”

In the “System Setting” area, please set this i-7188EX-MTCP ‘s Net-ID to a value not equal to 1, for example, 100. And “Stations per COM Port as 1” , then click on “Set”.

If it display “Modbus ID (1 ~ 1) ==> COM2” , the configuration is well done.

Note: If i-7188EX-MTCP ‘s COM1 is not set as “Debug” mode in the former page, the setting will become “Modbus ID (1 ~ 1) ==> COM1” . That is not correct, because here we want it send to i-7188EX-MTCP ‘ s COM2:RS-485.

Then when this i-7188EX-MTCP receives Modbus TCP/IP protocol to request Net-ID: 1, it will send this request to its COM2:RS-485 . And then if Wincon-8347 reply, it will reply back to the Ethernet port.





---

## Chapter 21. Connecting M-7000 Series I/O Modules

---

ISaGRAF controllers support M-7000 remote I/O since below driver version.

| <b>Controller</b>                         | <b>Driver version</b>              |
|---|------------------------------------|
| W-8037 / 8337 / 8737 / 8036 / 8336 / 8736 | 3.32B (better to be 3.36 or later) |
| W-8047 / 8347 / 8747 / 8046 / 8346 / 8746 | 3.32B (better to be 3.36 or later) |
| i-8417 / 8817 / 8437 / 8837               | 3.16 (better to be 3.19 or later)  |
| i-7188EG / 7188EGD                        | 2.14 (better to be 2.17 or later)  |
| i-7188XG / 7188XGD                        | 2.12 (better to be 2.15 or later)  |

Please visit <http://www.icpdas.com/products/PAC/i-8000/isagraf-link.htm> to download them & follows steps listed in “ReadMe.txt” or “Update\_w8xx7.pdf” to update them to your controller if your controller's driver is older.

The M-7000 series modules are RS-485 remote I/O modules which support Modbus RTU slave protocol. Please visit [http://www.icpdas.com/products/Remote\\_IO/m-7000/m-7000\\_list.htm](http://www.icpdas.com/products/Remote_IO/m-7000/m-7000_list.htm) for more information.

User can write ISaGRAF program to support Modbus RTU Master protocol to connect to M-7000 I/O or other Modbus RTU slave devices. Please refer to Chapter 8 of the “ISaGRAF user's Manual” . It resides at your I-8000 or W-8xx7 CD-ROM or can be download at <http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> .

There are some frequently asked question about the Modbus RTU Master. Please visit <http://www.icpdas.com/faq/isagraf.htm> for FAQ – 045, 046 , 047 , 049 & 050 .

One Modbus RTU Master port can connect up to 32 pcs. of M-7000 modules. I-8417/8817/8437/8837, iPAC-8447/8847 & I-7188EG/uPAC-7186EG & I-7188XG can setup max. 2 Modbus RTU Master ports, while W-8xx7/8xx6 can setup max. 10 Modbus RTU Master ports.

Please visit <http://www.icpdas.com/faq/isagraf.htm> FAQ-050 to download example program.

These programs all use COM3 port to connect to M-7000 I/O. You may change the “port\_no” setting of the “mbus” to fit your controller.

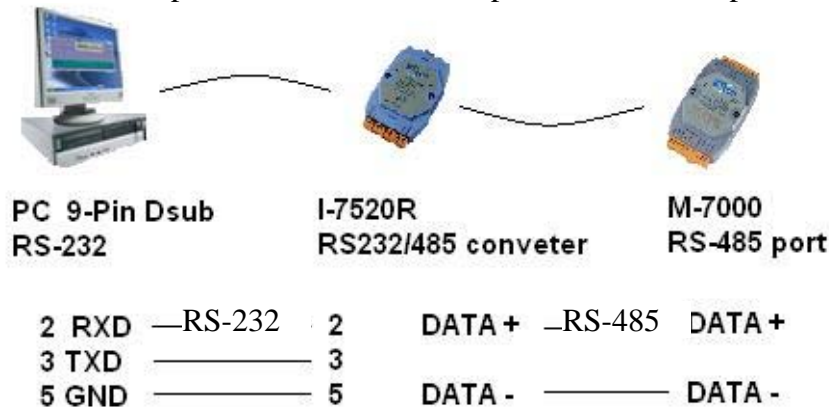
|          |   |
|----------|---|
| Wdemo_41 | COM3 connecting 1:M-7053D (16-Ch. D/I) + 2:M-7045D (16-Ch. D/O)   |
| Wdemo_42 | COM3 connecting 1: M-7053D to get D/I counter value (16-bit, 0- 65535)  |
| Wdemo_43 | COM3 connecting 1:M-7017R (8-Ch. A/I) + 2:M-7024 (4-Ch. A/O)  |
| Wdemo_44 | COM3 connecting 1: M-7017RC (8-Ch. Current Analog Input)  |
| Wdemo_45 | COM3 connecting 1: M-7019R (8-Ch. Universal A/I, <a href="#">thermocouple</a> or voltage input or current input) to get temperature value |
| Wdemo_46 | COM3 connecting 1:M-7080 (2-Ch counter or frequency)  |

## 21.1: Using DCON utility to do initial setting for M-7000

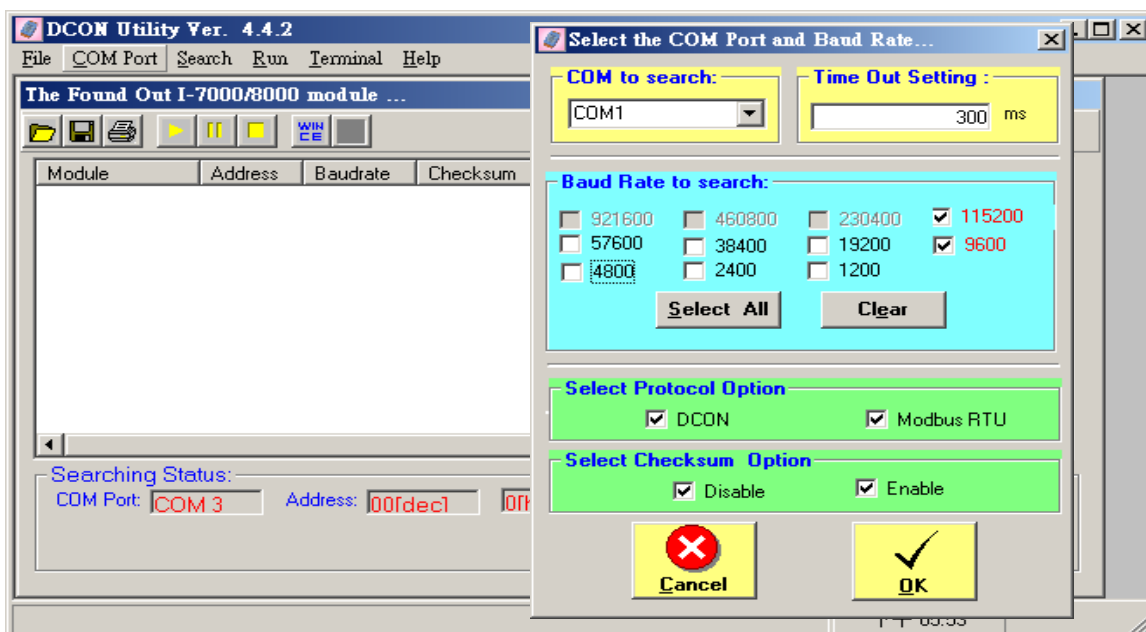
Before we starting at programming Modbus Master port, please run “DCON utility” to well configure M-7000's “Slave No” (or called Address), “Baudrate” for every D/I/O & A/I/O module and channel range or type setting for Analog input & output module. The “Procotol” setting should be “Modbus”. You may install “Dcon Utility” from the I-8000 CD-ROM or visit <http://www.icpdas.com/download/7000/7000.htm> to download and then install it.

### Steps to configure each M-7000 module.

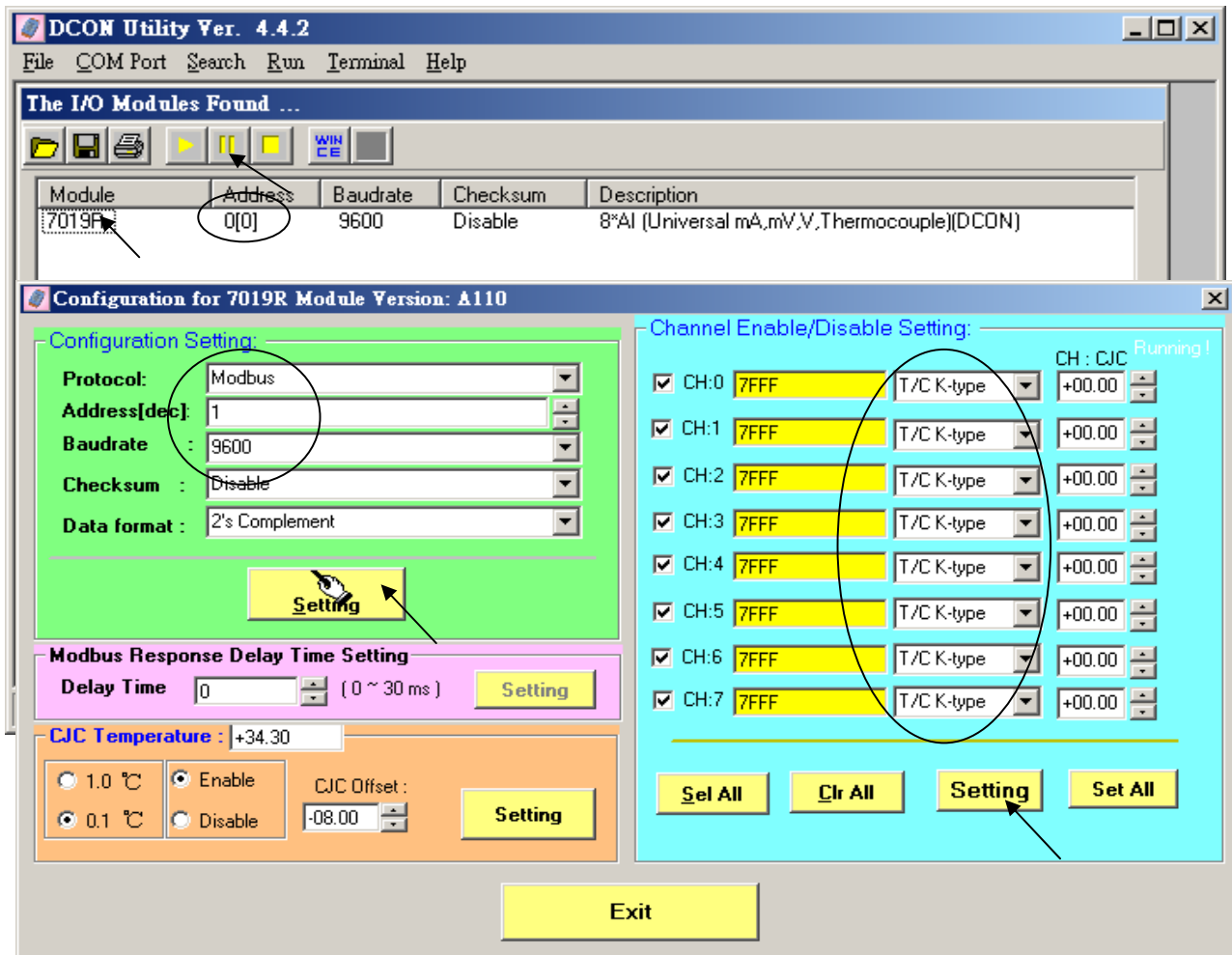
**Step 1:** Power off M-7000. Connect one RS232 cable from PC's COM1 (or other COM port) to one RS232/485 converter, for example i-7520R at [http://www.icpdas.com/products/Industrial/communication\\_module/communication\\_list.htm](http://www.icpdas.com/products/Industrial/communication_module/communication_list.htm), then connect this converter to the M-7000 module. **Please short “INIT\*” to “GND”**. This means to make the M-7000 to be in initial state (Address will be 0, baudrate=9600). Some M-7000 module provide a “Init – Normal” dip switch on its back to replace the “INIT\*” pin.



**Step 2:** Power on M-7000. Run “Dcon utility”, click “COM Port” menu to select proper COM port, baudrate, check on “DCON” & “Modbus RTU”, ... Then click “start Search” to search M-7000.



**Step 3:** The only one connected M-7000 should be found at Address=0 (because it is in initial state), Click “Stop” to stop searching when found it. Please set protocol as “Modbus”, proper “Address” (Slave no) , “Baudrate” . And if the M-7000 is Analog I/O, please set proper type & range, then click on “Setting”



**Step 4:** Power off M-7000. Remove connection between “INIT\*” and “GND” . **Then power it ON again . Run DCON utility to search & then check if the setting is correct or not.**

If the setting is not correct, modify them and click on “setting” again.

If this M-7000 Module is M-7041 or M-7044 or M-7050 or M-7053 or M-7060 or M-7063 or M-7065 (or M-7041D or M-7044D or M-7050D or M-7053D or M-7060D or M-7063D or M-7065D), please go to step 5. If the module is not in the above item numbers, then this M-7000 is well configured.

**Note:**

1. Every M-7000 must be configured to a unique “Address number” (1 to 247) and the same “Baudrate” and other proper setting before using it.
2. User may refer to the attached manual in the product box, or visit [http://www.icpdas.com/products/Remote\\_IO/m-7000/m-7000\\_list.htm](http://www.icpdas.com/products/Remote_IO/m-7000/m-7000_list.htm) to get each M-7000 Module's Manual to find their “Analog Input Type and Data Formate Table” (Type code setting).

## Important Step5:

After the initial configuration is completed (Step 1 to 4), please give below Modbus command to below M-7000 modules 's Digital input channels to invert them.

01 46 29 01 (4-byte command, each byte is 2 Hex-number)

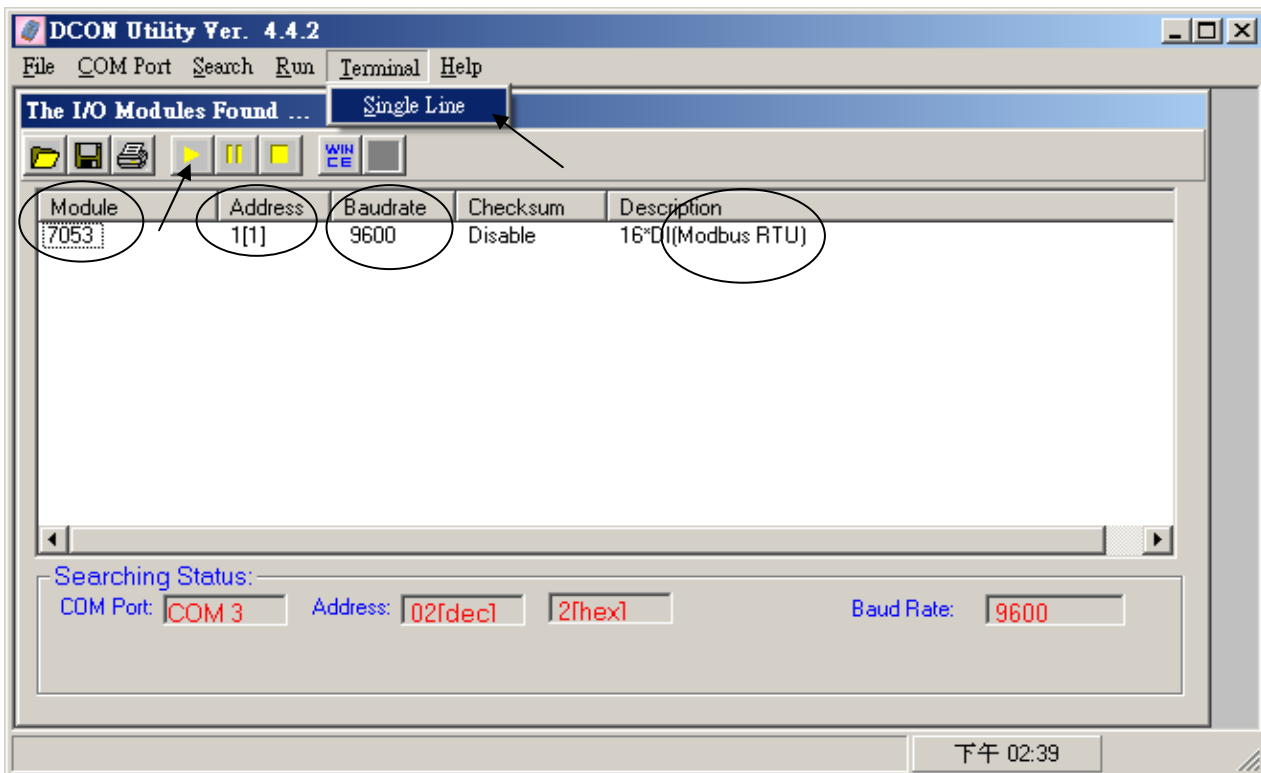
The first byte is the M-7000 Address number been set by DCON utility, it may be 01, 02, 03, ..., 0F, ... to F7 depends on your setting of the related M-7000. The other 3 bytes "46 29 01" should be always same.

| <i><b>M-7000 Modules should be inverted</b></i> |   |         |   |         |
|---|---|---------|---|---------|
| M-7041  | , | M-7044  | , | M-7050  |
| M-7053  | , | M-7060  | , | M-7063  |
| M-7065  |   |         |   |         |
| M-7041D   | , | M-7044D | , | M-7050D |
| M-7053D   | , | M-7060D | , | M-7063D |
| M-7065D   |   |         |   |         |

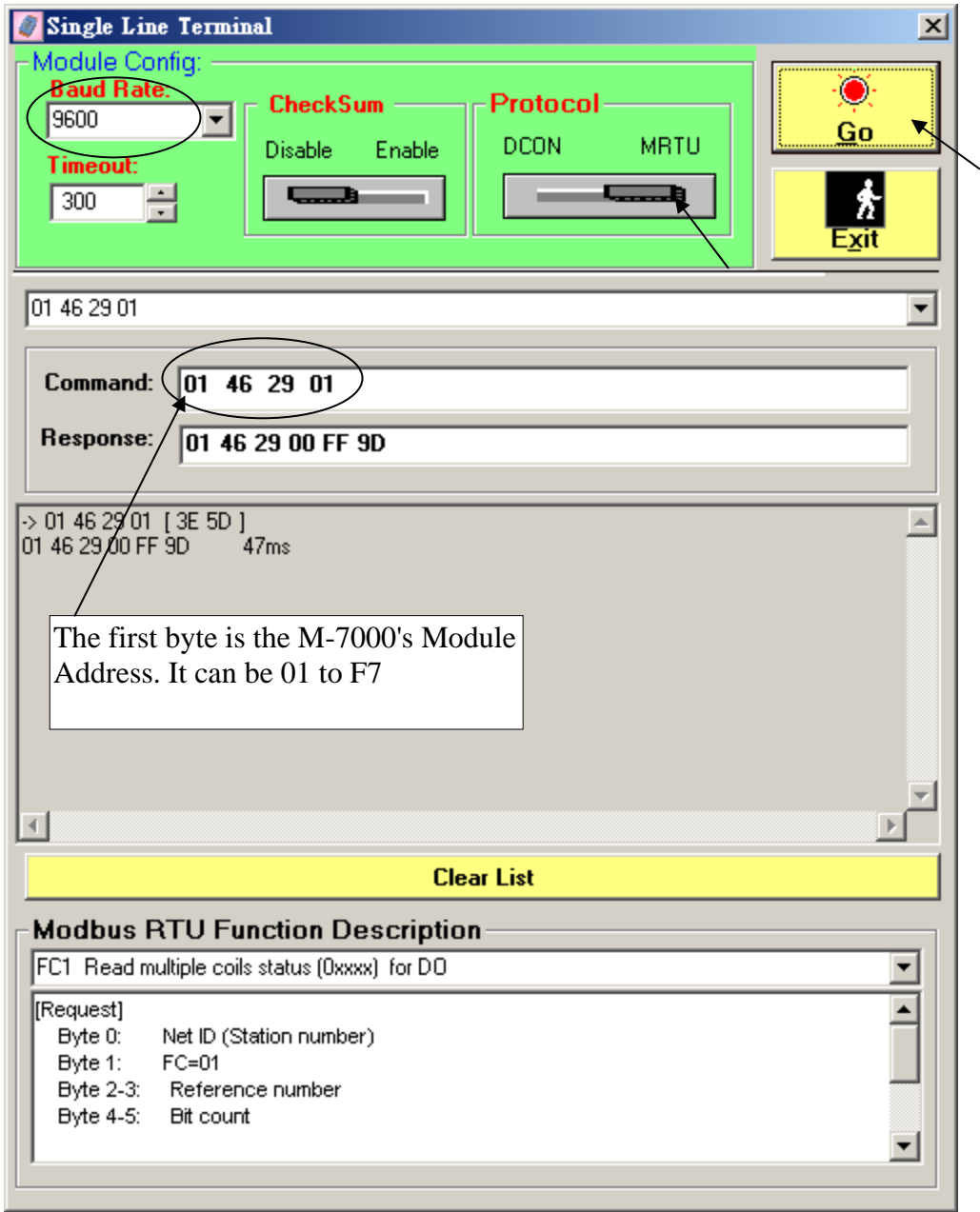
**Please Do Not give the upper command to other M-7000 modules which are not in the above lists.**

Steps to invert the digital input channels:

After Step 4 is finished, power on M-7041 or M-7044 or M-7050 or M-7053 or M-7060 or M-7063 or M-7065 again. Run DCON utility to search the module first. If the module is found. Stop search. Make sure the Module name is one of M-7041 or M-7044 or M-7050 or M-7053 or M-7060 or M-7063 or M-7065. Then goto "Terminal" - "Single Line"



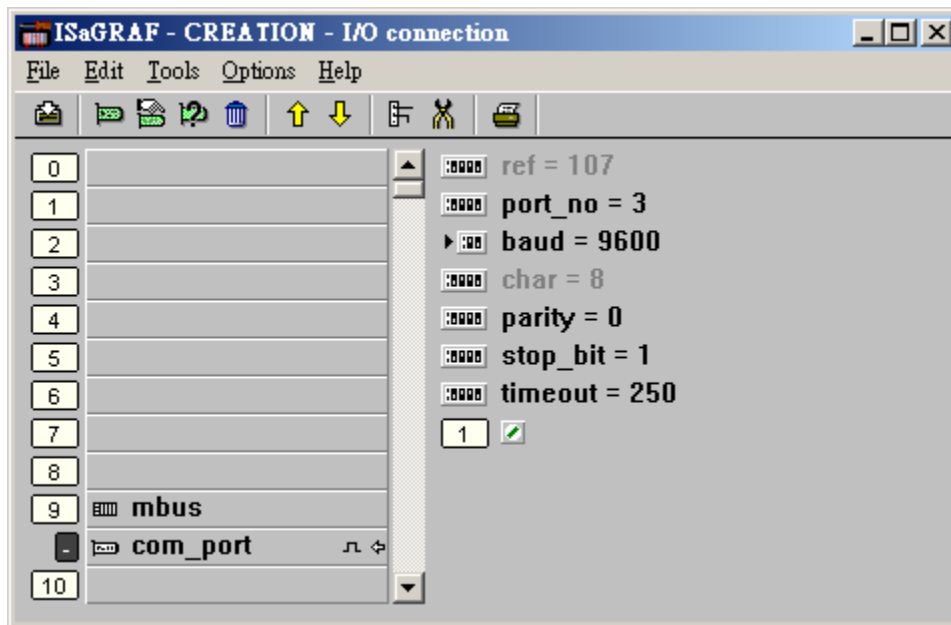
Select the correct baudrate, Protocol should be set to “MRTU”. Then type the inverted command as below, the first byte should be the Module's Address number. It can be 01 to F7. And then click “Go” . If the response is “01 46 29 ...” , it means command succeed. Power off this M-7000 modules. And it is well configured.



## 21.2: Writting program to connect to I-7000 modules

**Important** : If your M-7000 is **M-7041** or **M-7044** or **M-7050** or **M-7053** or **M-7060** or **M-7063** or **M-7065** (or **M-7041D** or **M-7044D** or **M-7050D** or **M-7053D** or **M-7060D** or **M-7063D** or **M-7065D**), please follow the former section 's Step5 to invert their digital input channels.

To program Modbus RTU Master, please connect “mbus” in the IsaGRAF IO connection windows as below. Please set proper “port\_no”, “baud” & “timeout”. “timeout” setting default is 500 ms, you can specify 250 ms if connecting only M-7000 I/O modules.



Then please create an Ladder program or function block program to access to each M-7000 I/O channels. ICP DAS IsaGRAF controllers can access to M-7000 modules by using “Mbus\_r” , “Mbus\_r1” , “Mbus\_b\_w” , “Mbus\_wb” & “Mbus\_n\_w” .

|          |  |
|----------|--|
| Mbus_R   | A. Read max. 12 word-value (-32768 ~ +32767) using Modbus function call <b>4</b> to read M-7000 <b>Analog input</b> channels or read D/I counter value.<br>B. Read max.192 bit-value using Modbus function call <b>2</b> to read M-7000 <b>Digital input</b> channels. |
| Mbus_R1  | Same as Mbus_R but with one extra setting – Period.<br>Read words or bits with a specified period time (unit is second)  |
| MBUS_N_W | Write max. 4 word-value (-32768 ~ +32767) using Modbus function code <b>6</b> or <b>16</b> to write M-7000 <b>Analog output</b> channels.<br>(write 1 word using code <b>6</b> , write 2 ~ 4 words using code <b>16</b> )  |
| MBUS_B_W | Write max. 4 bit-value using Modbus function code <b>5</b> or <b>15</b> to write M-7000 <b>Digital output</b> channels.<br>(write 1 bit using code <b>5</b> , write 2 ~ 4 bits using code <b>15</b> )  |
| MBUS_WB  | Write max. 16 bit-value using Modbus function code <b>15</b> to write M-7000 <b>Digital output</b> channels.   |

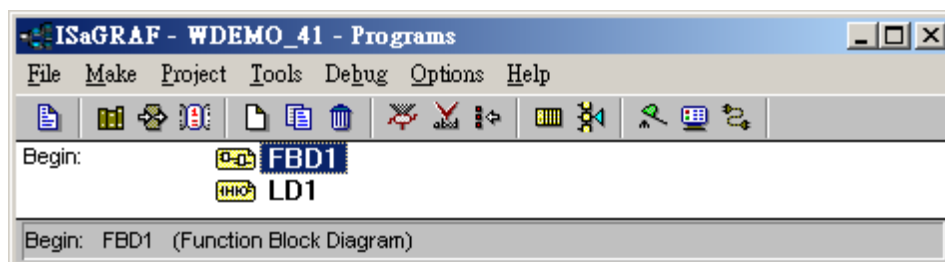
**Example 41:** Connecting 1:M-7053D (16-Ch. D/I) + 2:M-07045D (16-Ch. D/O)  
(This example is “Wdemo\_41”).

Please follow former section ‘s step 5 to invert M-7053's D/I channel's signal.

Variables:

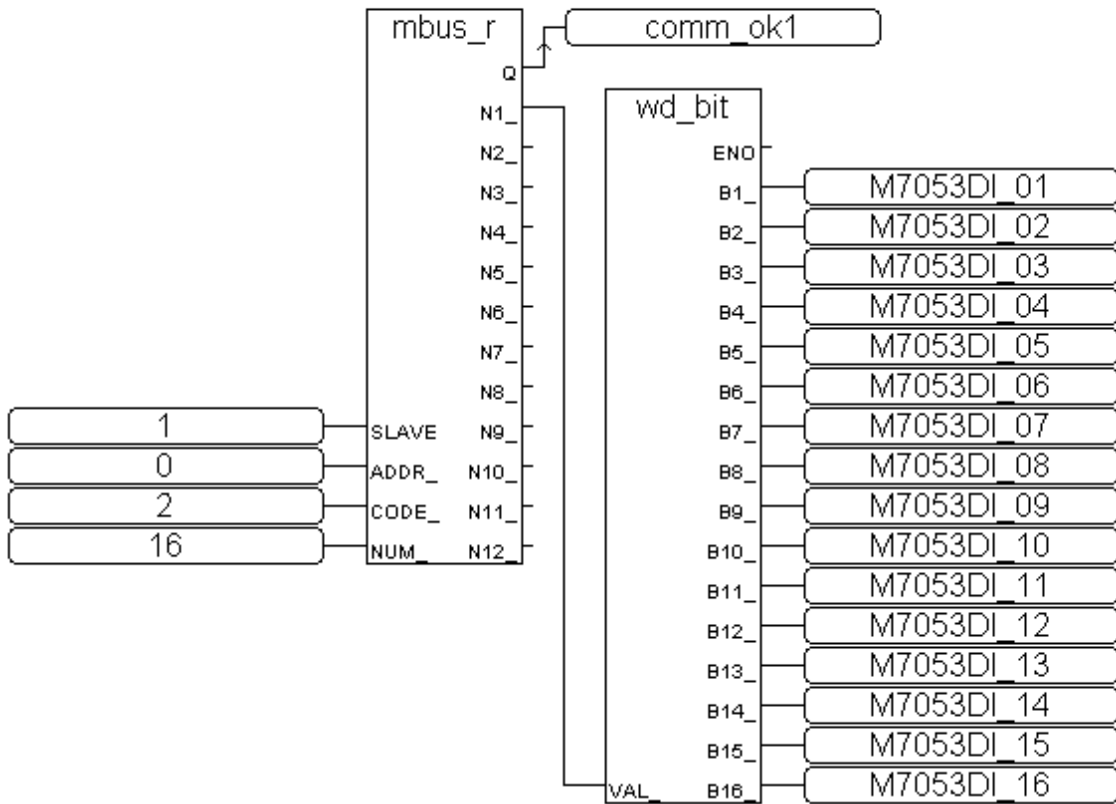
| <b>Name</b>                 | <b>Type</b> | <b>Attribute</b> | <b>Description</b>  |
|-----------------------------|-------------|------------------|---|
| comm_ok1                    | Bool        | Internal         | Communication state of the related M-7053D                        |
| comm_ok2                    | Bool        | Internal         | Communication state of the related M-7045D                        |
| M7053DI_01 to<br>M7053DI_16 | Bool        | Internal         | Total 16 boolean internal variables<br>D/I Ch. 1 to 16 of M-7053D |
| M7045DO_01 to<br>M7045DO_16 | Bool        | Internal         | Total 16 boolean internal variables<br>D/O Ch. 1 to 16 of M-7045D |

Project: One function block program + one Ladder program



Function block program:

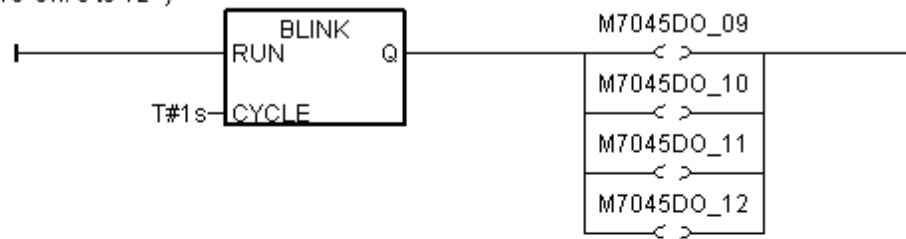
Request 16 bits from Slave=1 (M7000 Address=1)  
Using code=2 , starting Modbus ADDR\_ No. is 0  
If CODE=1 or 2, each returned N1 to N12 contains one word (or 16 bits)



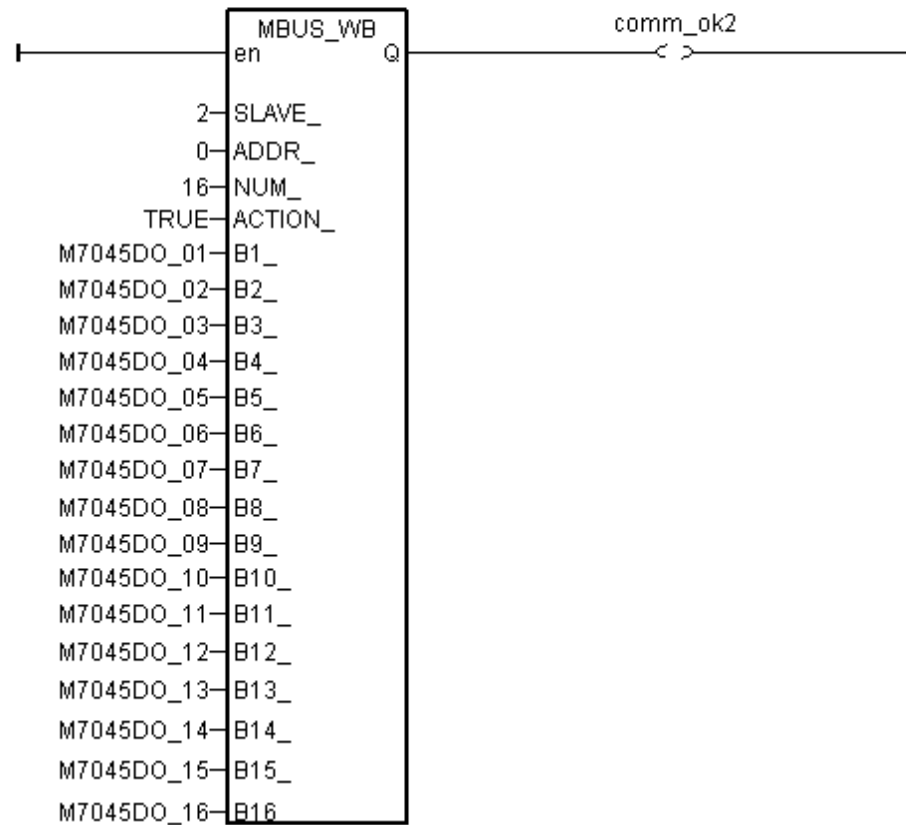


## Ladder Program:

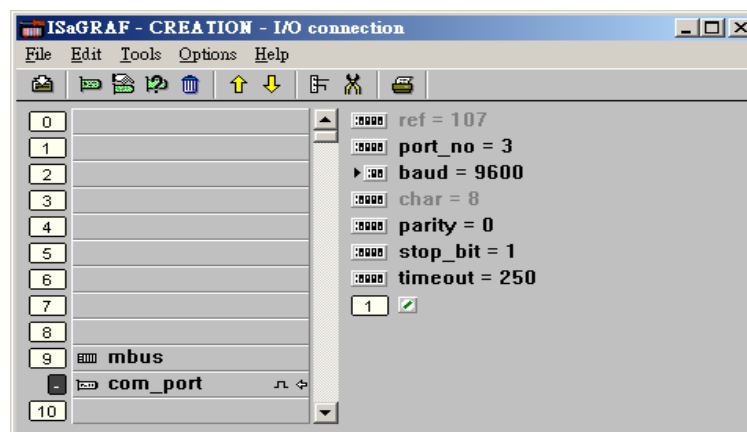
(\*Blinking D/O Ch. 9 to 12 \*)



(\* Write 16 bits to Slave=2 (M-7000 Address=2), starting Modbus ADDR\_ No. is 0, this block automatically uses code=15 \*)



## I/O connection:

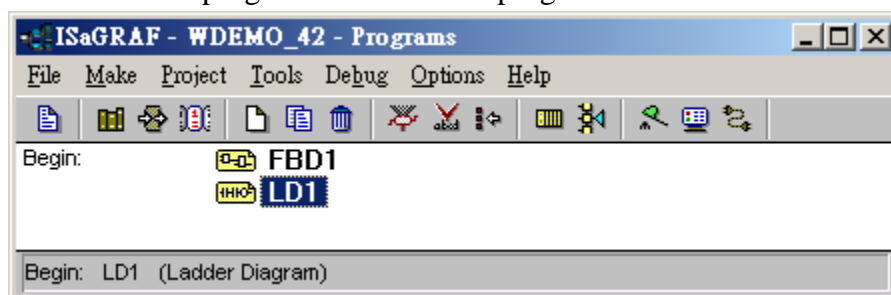


**Example 42:** Connecting 1:M-7053D to get D/I counter value (This example is “Wdemo\_42”)

Variables:

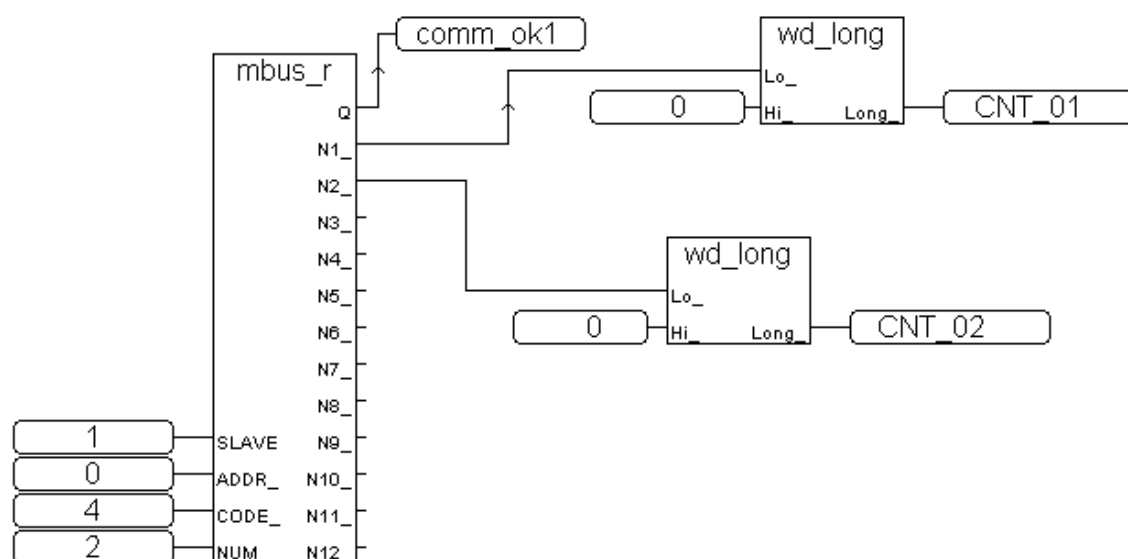
| <b>Name</b> | <b>Type</b> | <b>Attribute</b> | <b>Description</b>                               |
|-------------|-------------|------------------|--|
| comm_ok1    | Bool        | Internal         | Communication state of the related M-7053D       |
| RS1         | Bool        | Internal         | Set as True to reset Ch1. D/I counter value to 0 |
| RS2         | Bool        | Internal         | Set as True to reset Ch2. D/I counter value to 0 |
| CNT_01      | Integer     | Internal         | Ch1 D/I counter value                            |
| CNT_02      | Integer     | Internal         | Ch2 D/I counter value                            |

Project: One Function block program + one Ladder program



Function block program:

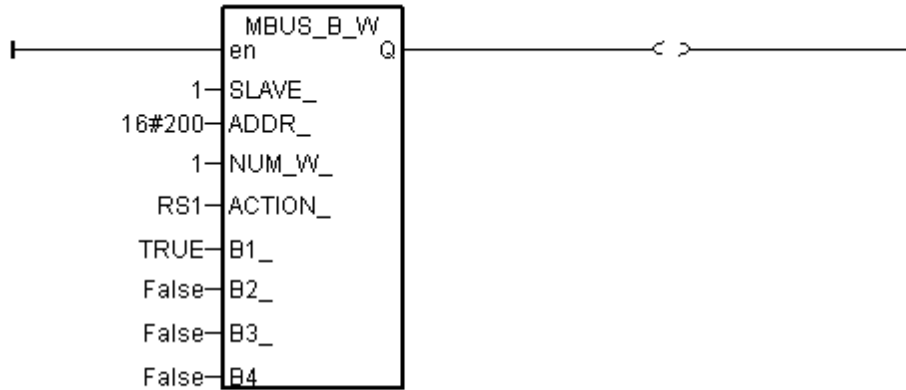
Using Code=4 to request M-7000 D/I counter value, Starting from Modbus ADDR No=0  
 NUM can be 1 to 12 depends on how many D/I counter channel in the M-7000 to be read  
 The M-7000 D/I counter value is from 0 ~ 65535 contained in one word.  
 Since Mbus\_r & Mbus\_r1 can only return word value as -32768 to +32767, so please  
 use "wd\_long" to convert this word to become a long integer value. Then the converted  
 counter value will be 0 to 65535



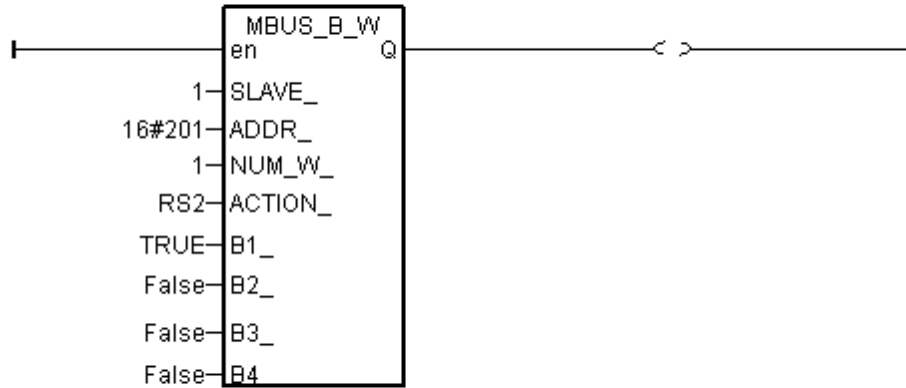
To reset M-7000's D/I counter value to 0, please write bit value 1 (TRUE) to coil Modbus No. 16#200 to 16#21F . Reset Ch1. Is to write to No. 16#200, Ch2 is 16#201, ..., Ch.32 is 16#21F.

#### Ladder Program:

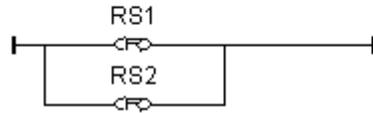
(\* Set RS1 to True to clear D/I counter 1 (ADDR 16#200), The "Clear D/I counter" 's Modbus ADDR is from 16#200 to 16#21F depends on the total D/I channel number of the M-7000 \*)



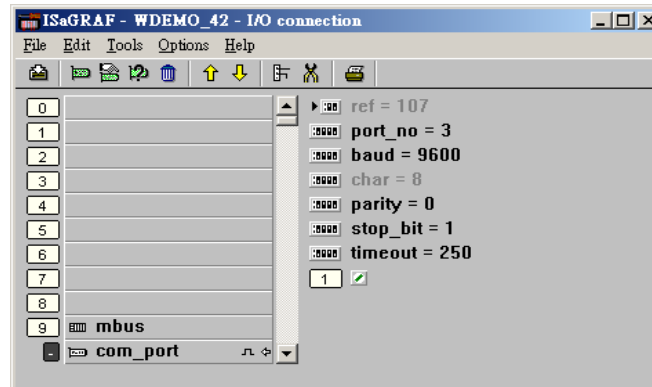
(\* Set RS2 to True to clear D/I counter 2 (ADDR 16#201), The "Clear D/I counter" 's Modbus ADDR is from 16#200 to 16#21F depends on the total D/I channel number of the M-7000 \*)



(\* alsway reset RS1 & RS2 to False at the end \*)



#### I/O connection:



**Example 43:** Connecting 1:M-7017R & 2:M-7024 (This example is “Wdemo\_43”)

Please set M-7017R's Input range & Type to +/- 10V

M-7024 Output range & Type to +/- 10V

User may refer to the attached manual in the product box, or visit

[http://www.icpdas.com/products/Remote\\_IO/m-7000/m-7000\\_list.htm](http://www.icpdas.com/products/Remote_IO/m-7000/m-7000_list.htm) to get each M-7000 Module's Manual to find their “Analog Input Type and Data Format Table” (Type code setting)

Please refer to below for using variable array.

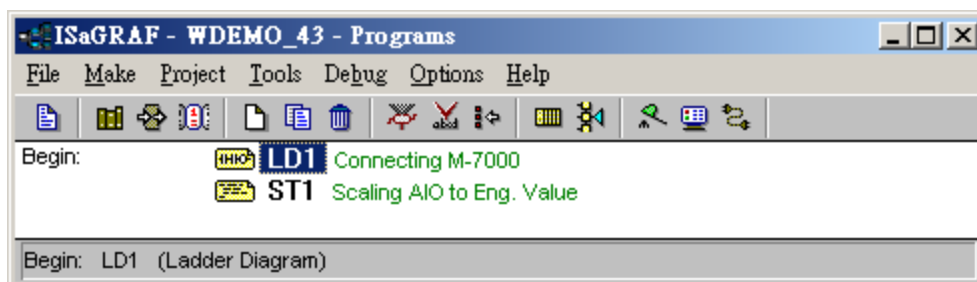
W-8xx7 CD-ROM: \napdos\isagraf\wincon\english\_manu\ variable\_arrays.pdf

[www.icpdas.com](http://www.icpdas.com) – FAQ – Software – IsaGRAF - FAQ039

Variables:

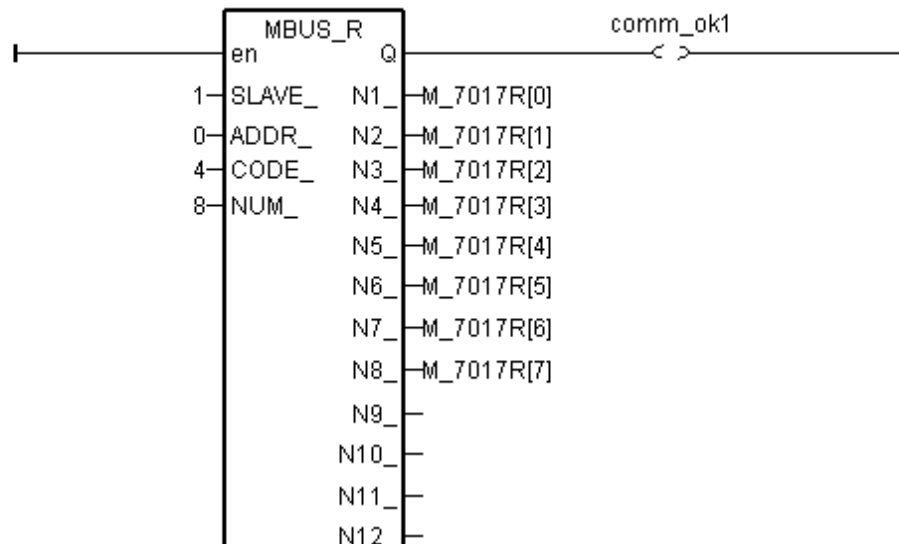
| <b>Name</b>   | <b>Type</b> | <b>Attribute</b> | <b>Description</b>   |
|---------------|-------------|------------------|--|
| comm_ok1      | Bool        | Internal         | Communication state of the related M-7053D   |
| comm_ok2      | Bool        | Internal         | Communication state of the related M-7045D   |
| M_7017R[0..7] | Integer     | Internal         | <b>Variable Array</b><br>M-7017R's Analog Input value<br>(-32768 to +32767) means (-10 to +10) V                               |
| M_7024[0..3]  | Integer     | Internal         | <b>Variable Array</b><br>M-7024's Analog Output value<br>(-16384 to +16383) means (-10 to +10) V                               |
| In_Val[0..7]  | Integer     | Internal         | <b>Variable Array</b><br>Engineering value converted from M_7017R[0..7]<br>(-32768 to +32767) converter to (-10000 to +10000)  |
| Out_Val[0..3] | Integer     | Internal         | <b>Variable Array</b><br>Engineering value to be converted to M_7024[0..3]<br>(-1000 to +1000) converter to (-16384 to +16383) |
| ii            | Integer     | Internal         | index  |

Project: One Ladder program + one ST program

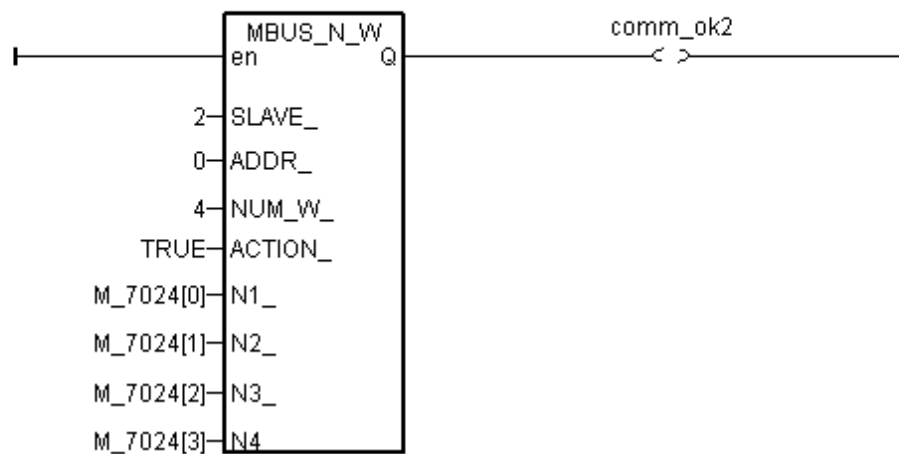


## Ladder program:

(\* Read 8 words from Slave=1 (M-7000 Address=1) using code=4. starting Modbus ADDR No. is 0  
Please set M-7017R's range to +/-10V by DCON utility (type code=8) \*)



(\* Write 4 words to Slave=2 (M-7000 Address=2) , starting Modbus ADDR No. is 0  
Please set M-70124's range to +/-10V by DCON utility (type code=33) \*)



ST program:

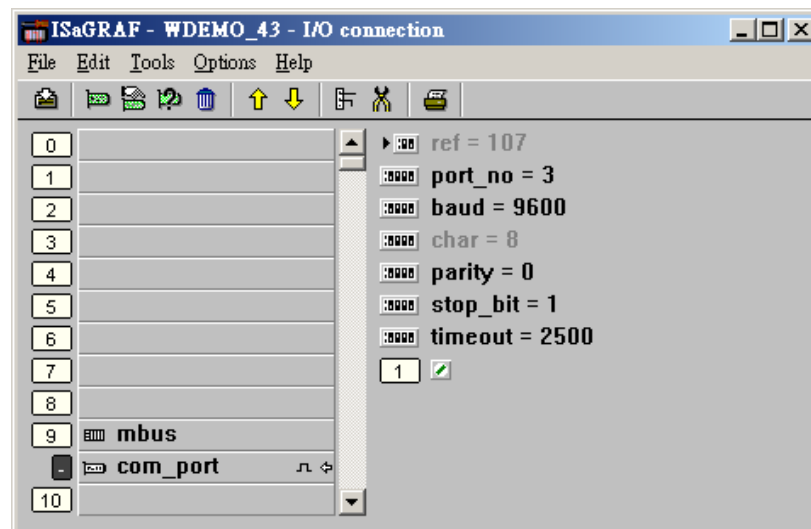
```
(* Please configure this M-7017R as +/- 10V range (type code=8) *)  
(* convert M-7017R's A/I value (-32768 to +32767) to become engineering value  
of (-10000 to +10000) *)
```

```
for ii := 0 to 7 do  
  IN_Val[ii] := Bin2Eng( M_7017R[ii] , 32767 , -32768 , 10000 , -10000 ) ;  
end_for ;
```

```
(* Please configure this M-7024 as +/- 10V range (type code=33) *)  
(* convert OUT_Val of (-1000 to +1000) to become M-7024's A/O value  
of (-16384 to +16383) *)
```

```
for ii := 0 to 3 do  
  
  if OUT_Val[ii] > 1000 then  
    M_7024[ii] := 16383 ;  
  
  elsif OUT_Val[ii] < -1000 then  
    M_7024[ii] := -16384 ;  
  
  elsif OUT_Val[ii] >= 0 then  
    M_7024[ii] := (OUT_Val[ii] * 16383) / 1000 ;  
  
  elsif OUT_Val[ii] < 0 then  
    M_7024[ii] := (OUT_Val[ii] * -16384) / 1000 ;  
  
  end_if ;  
  
end_for ;
```

I/O connection:



**Example 44:** Connecting 1:M-7017RC (This example is “Wdemo\_44”)

Please set M-7017RC 's Input range & Type to +/- 20 mA

User may refer to the attached manual in the product box, or visit

[http://www.icpdas.com/products/Remote\\_IO/m-7000/m-7000\\_list.htm](http://www.icpdas.com/products/Remote_IO/m-7000/m-7000_list.htm) to get each M-7000 Module's Manual to find their Analog I/O Value mapping to physical I/O (Type code setting)

Please refer to below for using variable array.

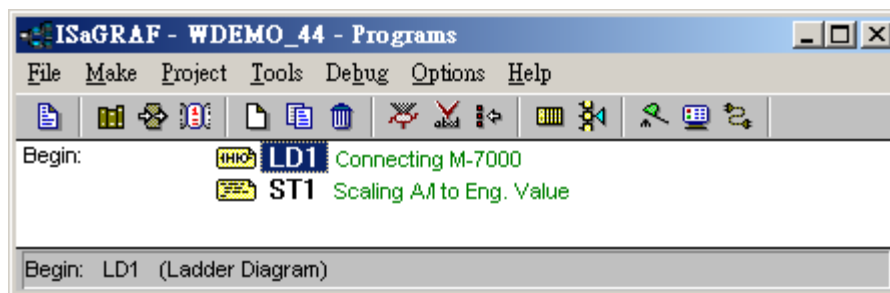
W-8xx7 CD-ROM: \napdos\isagraf\wincon\english\_manu\ variable\_arrays.pdf

[www.icpdas.com](http://www.icpdas.com) – FAQ – Software – IsaGRAF - FAQ039

Variables:

| <b>Name</b>   | <b>Type</b> | <b>Attribute</b> | <b>Description</b>   |
|---------------|-------------|------------------|--|
| comm_ok1      | Bool        | Internal         | Communication state of the related M-7053D   |
| M7017RC[0..7] | Integer     | Internal         | <b>Variable Array</b><br>M-7017RC 's Analog Input value<br>(-32768 to +32767) means (-20 to +20) mA if setting<br>Input range & Type to +/- 20 mA  |
| In_Val[0..7]  | <b>REAL</b> | Internal         | <b>Variable Array (REAL format)</b><br>Engineering value converted from M7017RC[0..7]<br>4 to 20 mA converting to (0.0 to 1000.0) psi              |
| VAL[0..7]     | Integer     | Internal         | <b>Variable Array (Integer format)</b><br>Engineering value converted from M7017RC[0..7]<br>4 to 20 mA converting to (0 to 10000), unit is 0.1 psi |
| ii            | Integer     | Internal         | index  |

Project: One Ladder program + one ST program



Analog input Table of M-7017RC:

+/- 20 mA type (type code=16#D) : -32768 to +32767

4 to 20 mA type (type code=16#7) : 0 to +32767

If the input sensor type is 4 to 20 mA , it is better to set M-7017RC as +/- 20 mA type .  
 (It is not good to set M-7017RC as "4 to 20 mA" type. )

The reason is, when the sensor is broken, the analog input of M7017RC[0..7] will be near to 0.  
 If setting M-7017RC 's range type as 4 to 20 mA type, the value near 0 can mean 4 mA, and also can mean sensor broken. So no way to distinguish them.

However if setting M-7017RC as +/- 20 mA type, the value near 0 only means sensor broken if the communication of M-7017RC is well.

Because if sensor is well, the input is 4 to 20 mA, value should be (6553 to 32767).

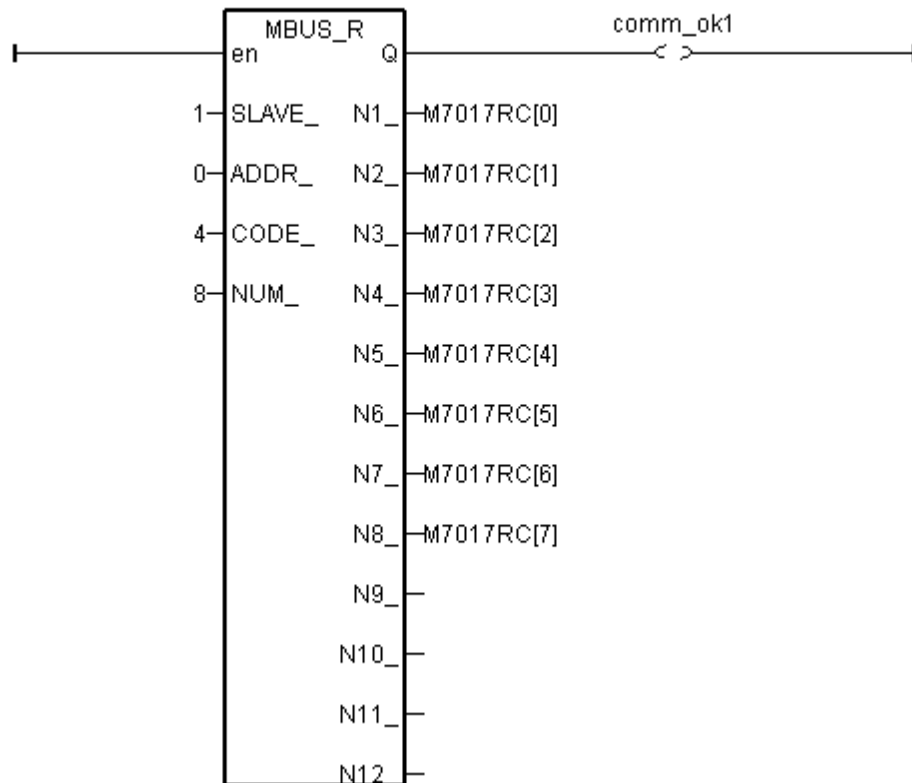
Value near 0 means sensor broken if the communication of M-7017RC is well.

For safe reason, please set M-7017RC as +/- 20 mA type.

So you can say if the value of M7017RC[0..7] < 5000 " or < 4000" , then it means sensor broken.

One Ladder program:

(\* Read 8 words from Slave=1 (M-7000 Address=1) using code=4. starting Modbus ADDR No. is 0  
 Please set M-7017RC 's range to +/-20 mA by DCON utility (type code=D) \*)





One ST program:

```
(* Please configure this M-7017RC as +/- 20 mA range (type code=D) *)  
(* We will convert (4 , 20 mA) to become (0.0 , 1000.0 Psi), Real format *)
```

```
for ii := 0 to 7 do  
  IN_Val[ii] := A4_20_To( M7017RC[ii] , 16#D , 1000.0 , 0.0 );  
end_for ;
```

```
(* or you may use Bin2Eng( ) to convert (4 to 20mA) to become (0 to 10000) as below, unit  
is 0.1 psi *)
```

```
(* Please declare Val[0..7] as Integer format *)  
for ii := 0 to 7 do  
  Val[ii] := Bin2Eng( M7017RC[ii] , 32767 , 6553 , 10000 , 0 );  
end_for ;
```

```
(* You can do something if the sensor is broken or communication is break *)  
if comm_ok1 and ( M7017RC[ii] < 5000 ) then
```

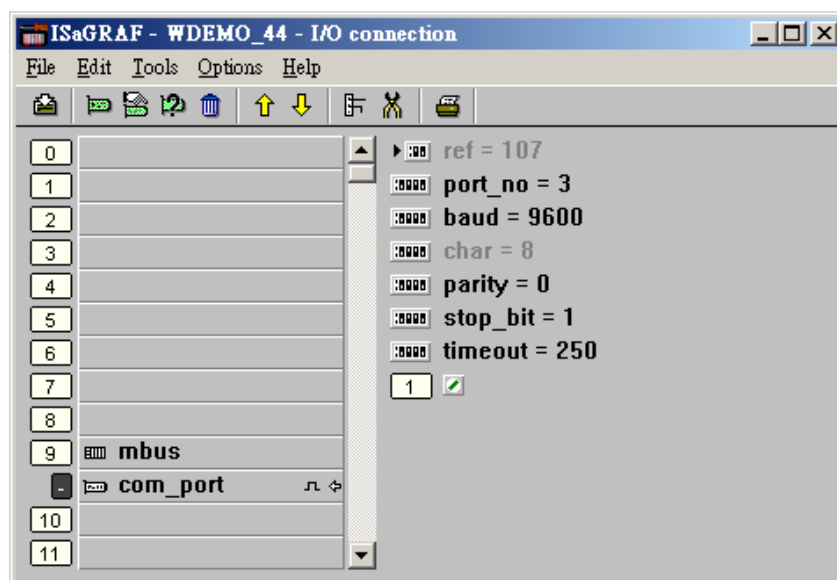
```
  (* You may do something if 4-20 mA sensor is broken *)
```

```
elseif comm_ok1=False then
```

```
  (* You may do something if communication between controller & M-7017RC is break *)
```

```
end_if ;
```

I/O connection:



**Example 45:** Connecting 1:M-7019R to get temperature val (This example is “Wdemo\_45”)

Please use DCON utility to configure M-7019R's range & type to Thermocouple, K-Type (Type code=0F)

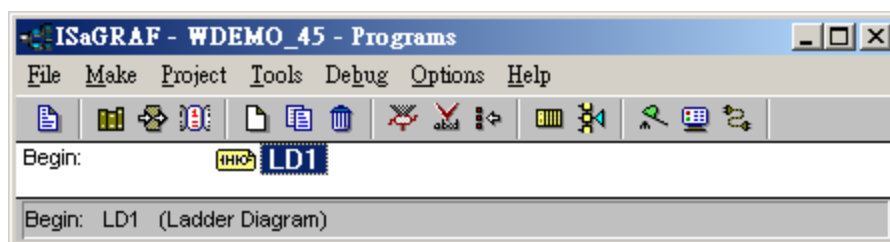
User may refer to the attached manual in the product box, or visit

[http://www.icpdas.com/products/Remote\\_IO/m-7000/m-7000\\_list.htm](http://www.icpdas.com/products/Remote_IO/m-7000/m-7000_list.htm) to get each M-7000 Module's Manual to find their “Analog Input type and data Format Table” (Type code setting)

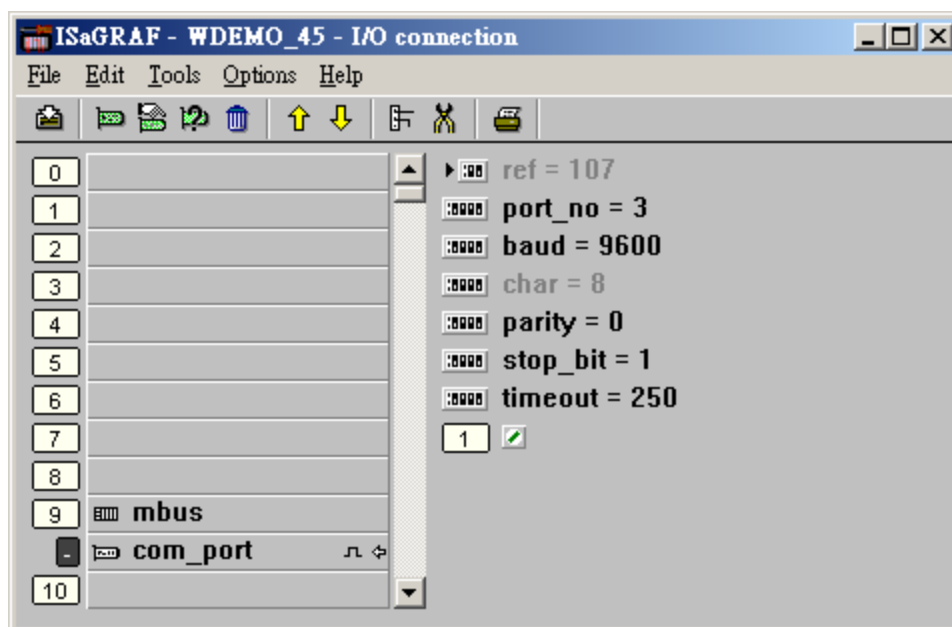
Variables:

| <b>Name</b>             | <b>Type</b> | <b>Attribute</b> | <b>Description</b>                                 |
|-------------------------|-------------|------------------|--|
| comm_ok1                | Bool        | Internal         | Communication state of the related M-7000          |
| Temper_1 to<br>Temper_8 | Integer     | Internal         | Temperature input value of Ch1. To 8 of<br>M-7019R |

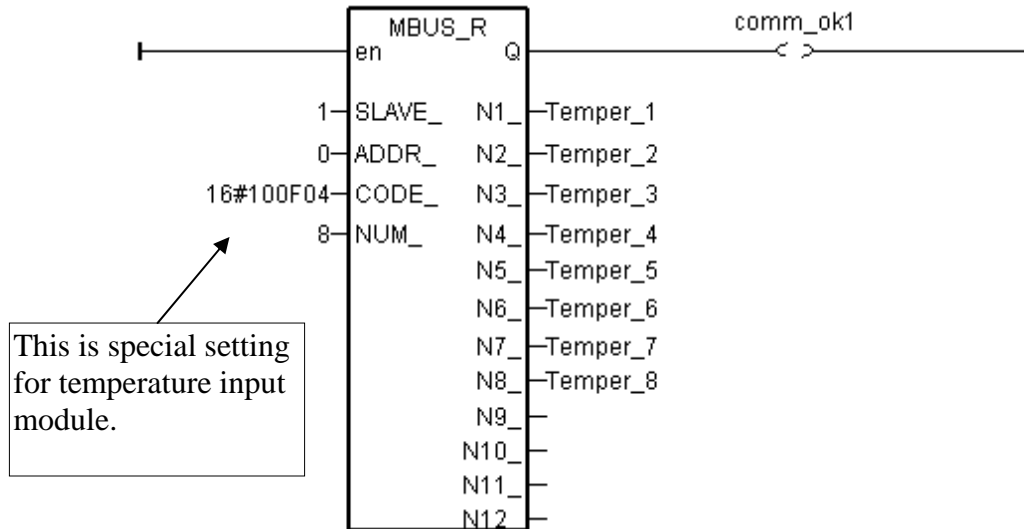
Project: One Ladder program



I/O connection:



Ladder program:



The "CODE\_" parameter of "MBUS\_R" & "MBUS\_R1" can be "standard" or "special" setting.

In the **"standard" setting case**,

Setting "CODE\_" as 1 or 2, each returned N1\_ to N12\_ contains 16-bits data (or 16 boolean)

Setting "CODE\_" as 3 or 4, each returned N1\_ to N12\_ is normally from -32768 to +32767.

The **"special" setting case** is for M-7000 temperature input modules like M-7015, M-7018R & M-7019R, Please set "CODE\_" to a special value defined as below.

Format : TTRRCC (Hex.)

TT=10 (Convert to "Degree Celsius")

TT=20 (Convert to "Degree Fahrenheit")

TT=00 (standard setting, -32768 to +32767. RR should be set as 00 if TT=00)

RR : "Type Code" setting of the related temperature input module

CC : Modbus function code 1 to 4 of the related Modbus device

The temperature input value unit is 0.01 degree. For ex, if returned "3012", it means 30.12 degree. If returned 999990, it means "sensor broken line"

For example, setting "CODE\_" as below to read the temperature value of M-7019:

**A. 16#100F04 :** (TT=10, RR=0F CC=04, Hex) the input value will be "Degree Celsius", unit is 0.01 degree, range= "0F: Thermocouple K Type, -270 ~1372 degree Celsius", code=04(Dec.). That results input value of "2356" = 23.56 Degree Celsius, "-489" = -4.89 Degree Celsius, "999990" = sensor broken-line.

**B. 16#200F04 :** (TT=20, RR=0F, CC=04, Hex)) the input value will be "Degree Fahrenheit", unit is 0.01 degree, range= "0F: Thermocouple K Type, -270 ~1372 degree Celsius", code=04(Dec.). That results input value of "4512" = 45.12 Degree Fahrenheit, "500" = 5.00 Degree Fahrenheit, "999990" = sensor broken line.

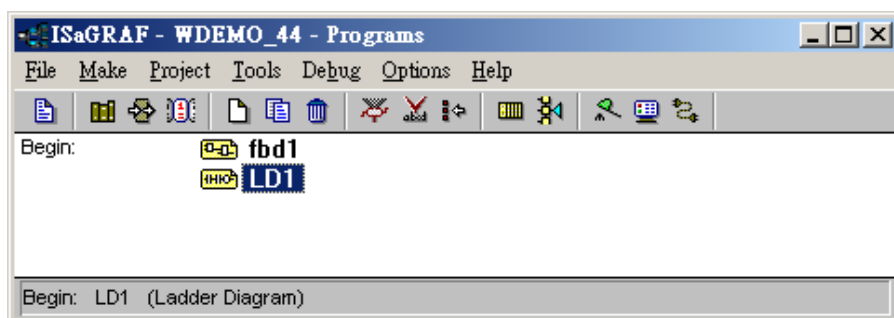
**C. 16#04 :** (TT=00, RR=00, CC=04) standard setting, the input value will be , -32768 to +32767, code=4

**Example 46:** Connecting 1: M-7080-D to get counter value (This example is “Wdemo\_46”)

Variables:

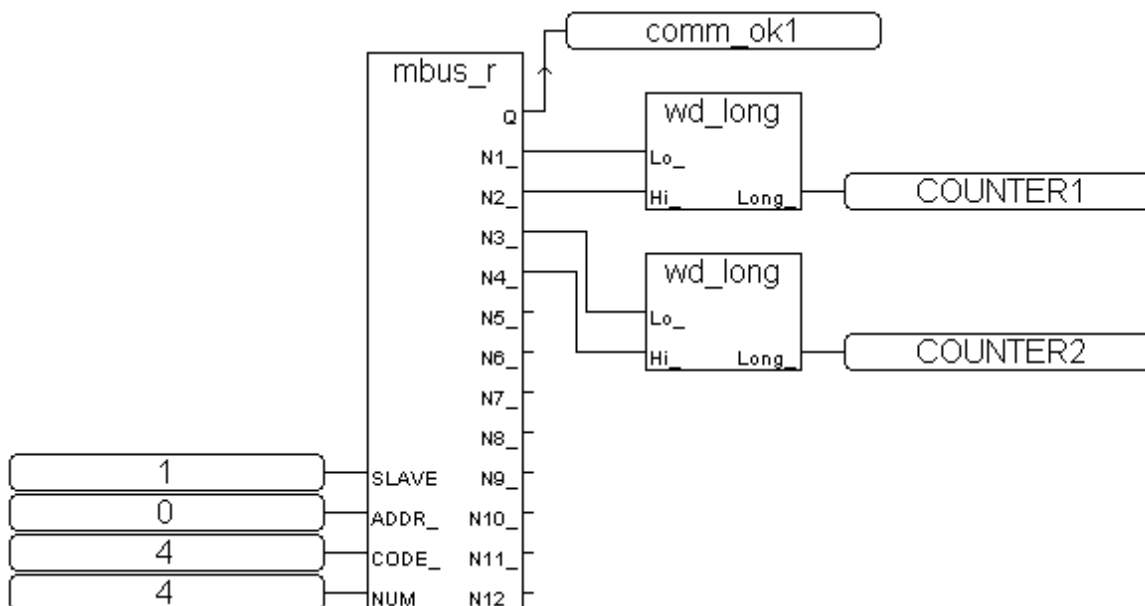
| <b>Name</b> | <b>Type</b> | <b>Attribute</b> | <b>Description</b>                        |
|-------------|-------------|------------------|---|
| comm_ok1    | Bool        | Internal         | Communication state of the related M-7000 |
| RS1         | Bool        | Internal         | set as True to reset counter 1 as 0       |
| RS2         | Bool        | Internal         | set as True to reset counter 2 as 0       |
| COUNTER1    | Integer     | Internal         | 1st Counter or frequency value of M-7080D |
| COUNTER2    | Integer     | Internal         | 1st Counter or frequency value of M-7080D |

Project: One function block program + one Ladder program



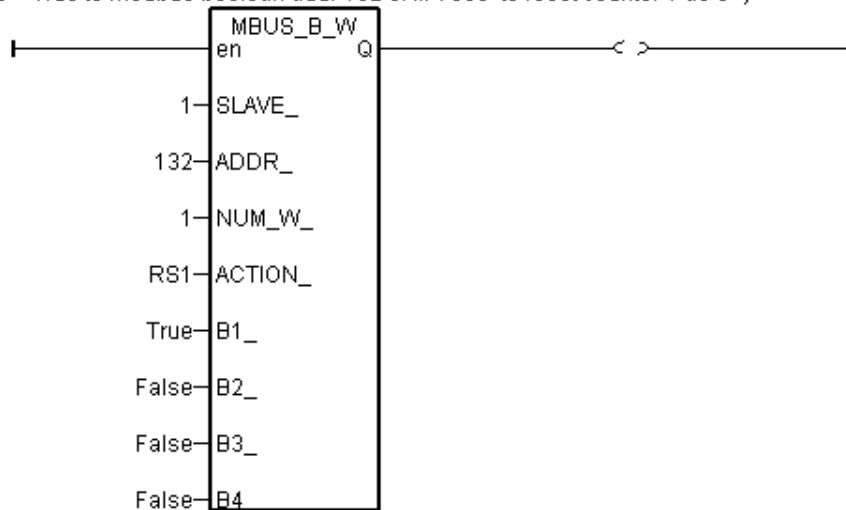
Function block program:

Request 4 words using Modbus code=4 from "Slave=1" (M-7000's Address=1)  
 The starting Modbus ADDR\_No. is 0  
 Then convert 2 words to become one long integer

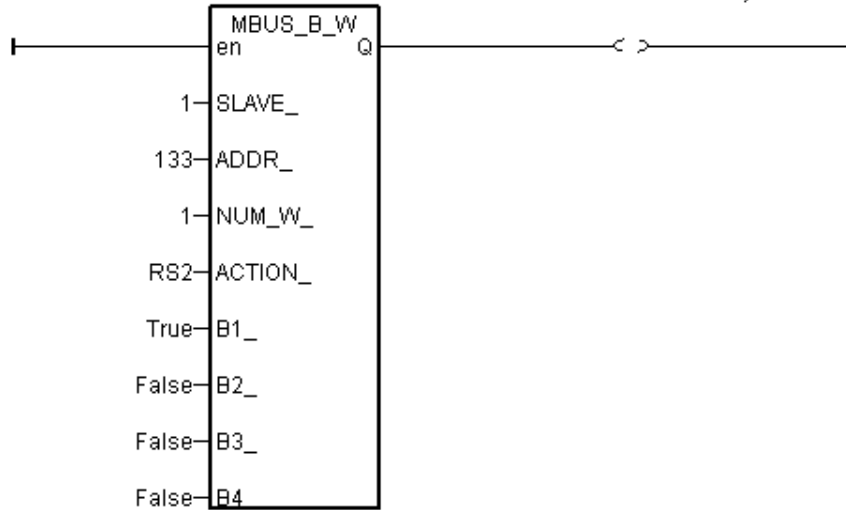


## Ladder program:

(\* Write value = True to modbus boolean addr 132 of M-7080 to reset counter 1 as 0 \*)



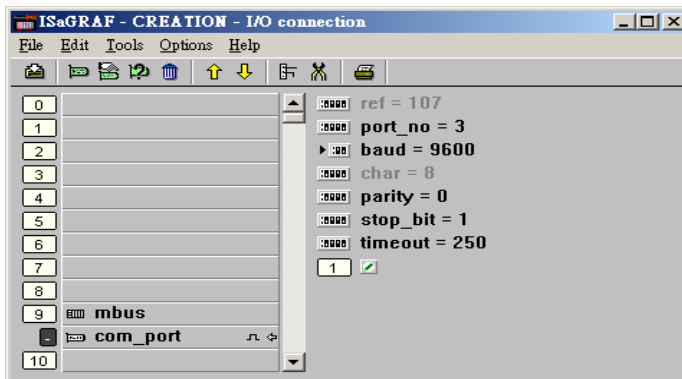
(\* Write value = True to modbus boolean addr 133 of M-7080 to reset counter 2 as 0 \*)



(\* reset RS1 & RS2 to False at the end \*)



## I/O connection:



## Chapter 22. Connecting i-8KE4-MTCP And I-8KE8-MTCP Ethernet I/O

Wincon-8347 / 8747 supports i-8KE4-MTCP & i-8KE8-MTCP ethernet I/O since Wincon ISaGRAF driver version 3.32B (it is better to use version 3.36 or later version).

i-8KE4/8-MTCP: [http://www.icpdas.com/products/PAC/i-8000/i-8KE4\\_8KE8\\_MTCP.htm](http://www.icpdas.com/products/PAC/i-8000/i-8KE4_8KE8_MTCP.htm)

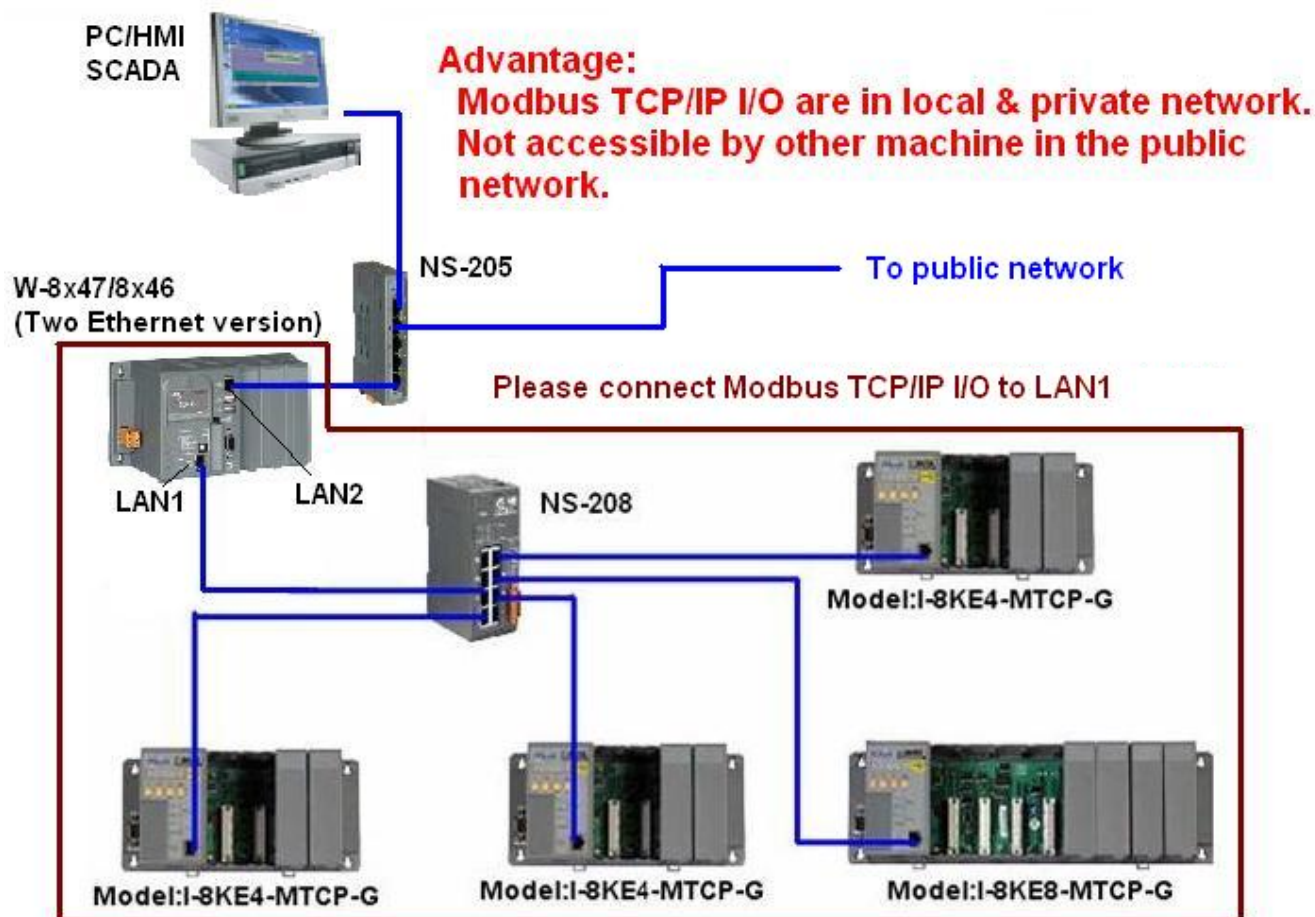
Wincon ISaGRAF driver: <http://www.icpdas.com/products/PAC/i-8000/isagraf-link.htm>

NS-205 / NS-208 : [http://www.icpdas.com/products/Switch/industrial/ethernet\\_switch.htm](http://www.icpdas.com/products/Switch/industrial/ethernet_switch.htm)

### 22.1: Induction of the I-8KE4-MTCP and I-8KE8-MTCP

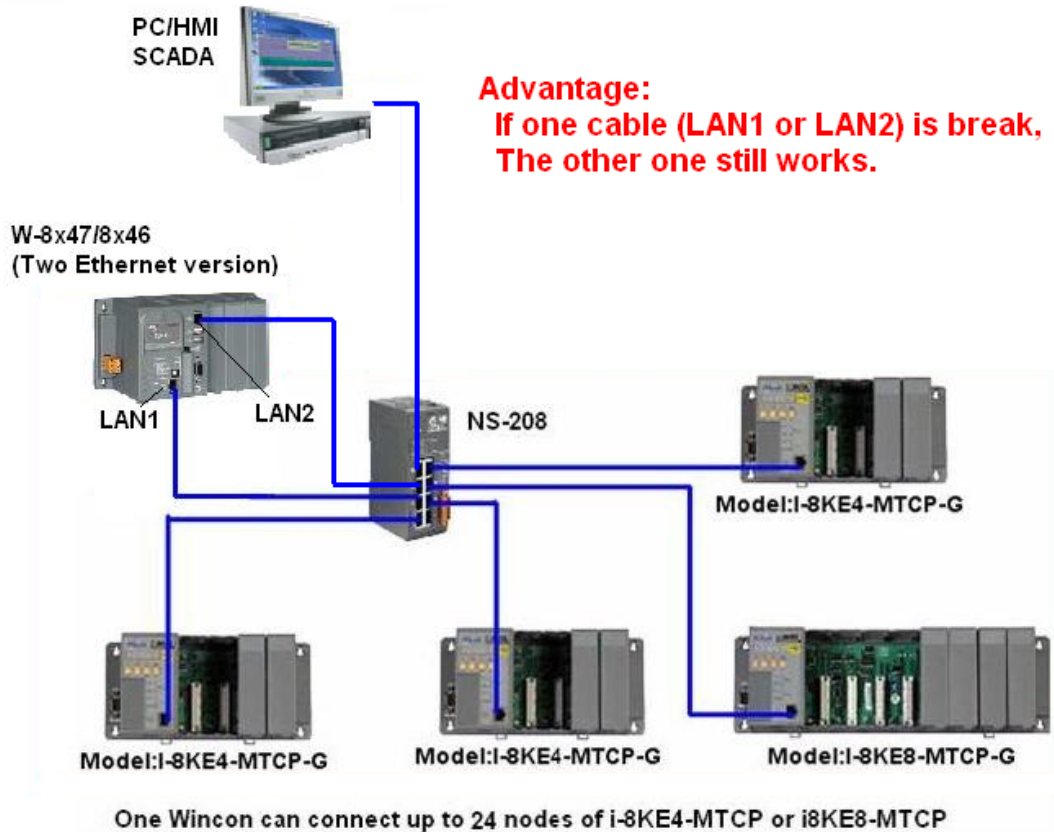
One Wincon-8xx7/8xx6 can connect max. 24 nodes of i-8KE4-MTCP and I-8KE8-MTCP. The Ethernet I/O scan time of the 3000 ~ 6000 I/O Channels for one W-8347 / 8747 is about 30 to 40 ms. If connecting less than 10 nodes of I-8KE4/8-MTCP, the Ethernet I/O scan time is about 20 ms. However, it still depends on how big (complex) of your logic program. (**The Ethernet I/O scan time of one W-8x36 / 8x37 is about twice of the W-8347 / 8747. That means W-8x36 / 8x37 is slower than W-8347 / 8747 when connecting Ethernet I/O**)

**Configure1:** W-8347 / 8747 (Dual Ethernet version) connecting Modbus TCP/IP I/O in a safe-local-private network .



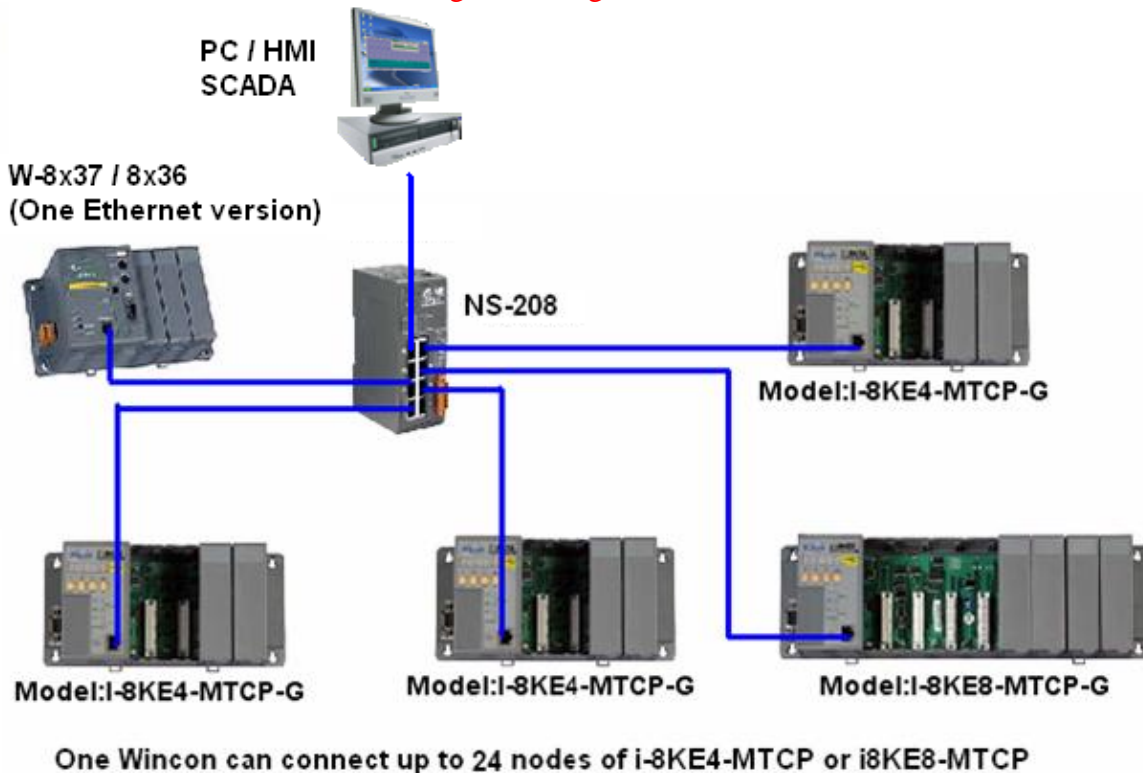
One Wincon can connect up to 24 nodes of i-8KE4-MTCP or i8KE8-MTCP

**Configure2:** W-8x47/8x46 (Dual Ethernet) connecting Modbus TCP/IP I/O in both two ports.



**Configure3:** W-8x37/8x36 (One Ethernet version) connecting Modbus TCP/IP I/O.

This configure doesn't have the advantage of configure 2 . And if the NS-208 is connected to public network, then it doesn't have the advantage of configure 1.



## 22.2: Programming the I-8KE4-MTCP and I-8KE8-MTCP

---

### Step 1.

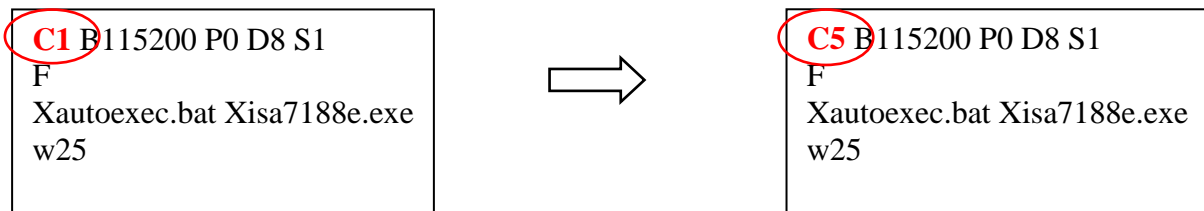
The first step is to assign an unique IP address to all of the i-8KE4-MTCP and i-8KE8-MTCP. Please power off the i-8KE4/8-MTCP, short its “INIT” to “INIT \* COM”, and then power it up.

Then connect a RS-232 cable from your PC ‘s COM1 to the i-8KE4/8-MTCP ‘s COM1.

Then please run “7188xw.exe” on PC ( “7188xw.exe” is burned in I-8000 CD-ROM or can be download at <ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/minios7/utility/>).

If your computer has no COM1/COM2 or you use other COM (like COM5) to link the i-8KE4/8-MTCP, you can change the “C number” in the first line of “7188xw.ini” file.

EX: Using computer’s COM5 to link to i-8KE4/8-MTCP



Press some <Enter> in the “7188xw.exe” windows , and then type “ip” to view the current IP setting. Then type in for example, “**ip 192.168.2.70**” to set an IP address to it.

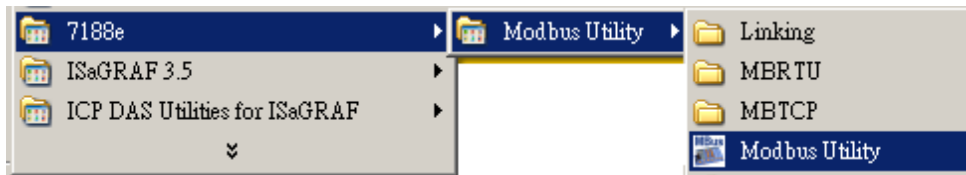
To set mask address, please type in for example, “**mask 255.255.255.0**”

PLEASE make sure to remove the connection between the “INIT” and the “INIT \* COM” pin on the i-8KE4/8-MTCP ‘s front panel after setting successfully. And then re-cycle its power.

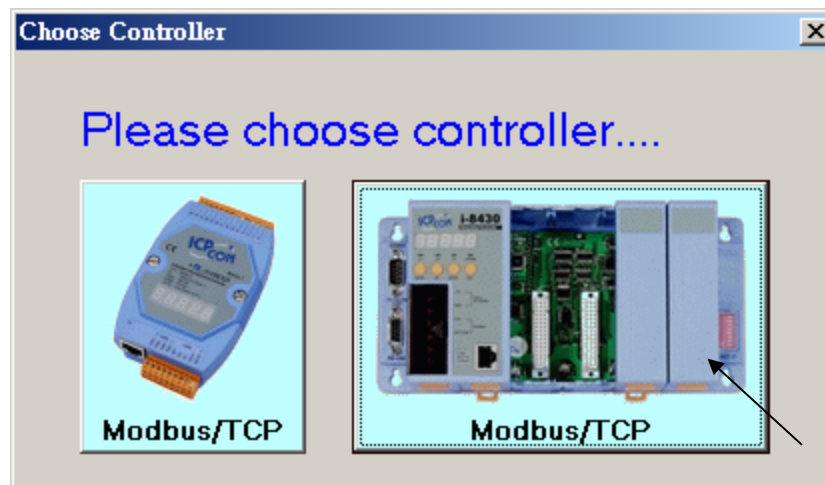


## Step 2.

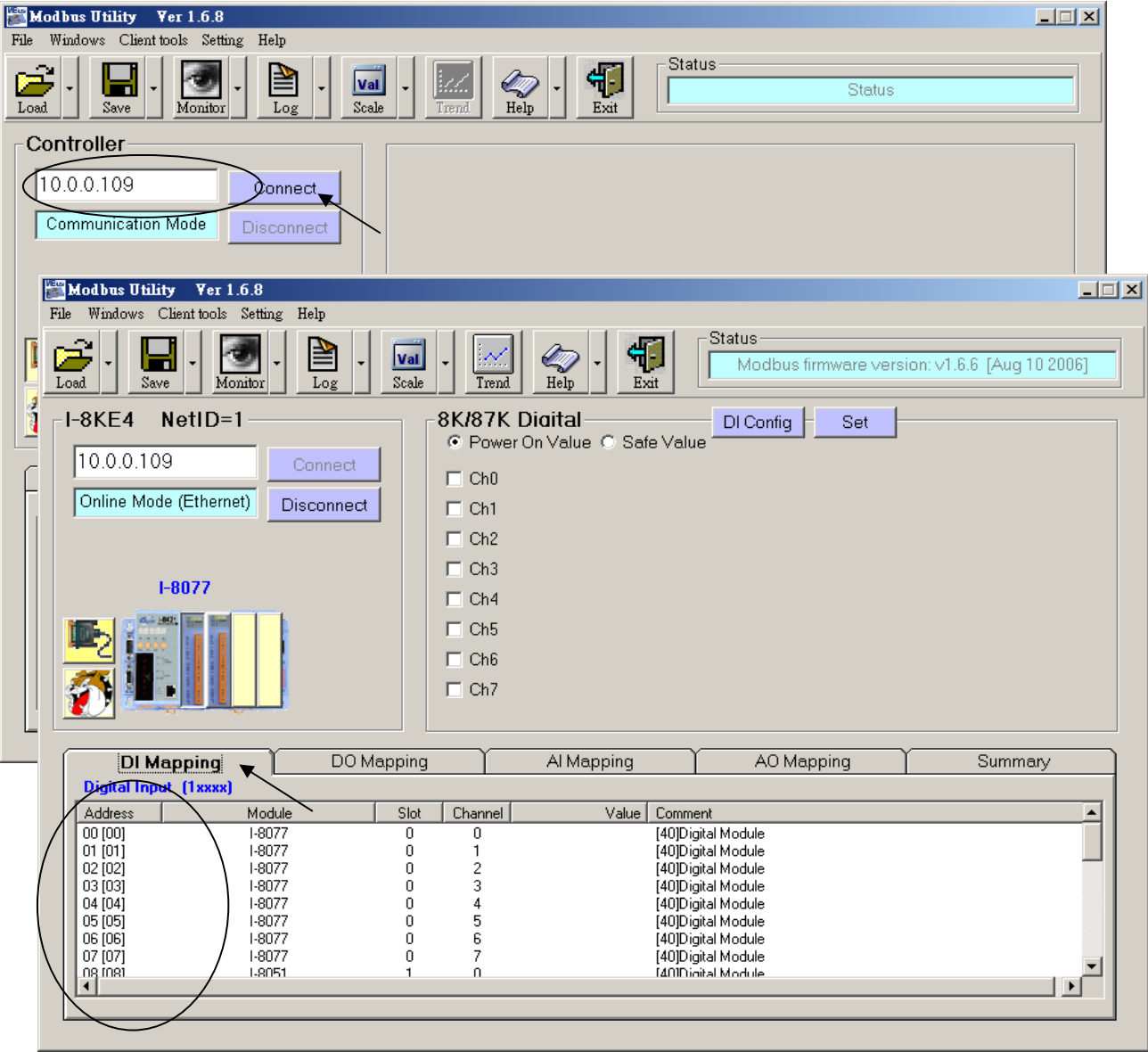
The second step is to configure all of the connecting i-8KE4/8-MTCP by running “Modbus utility”. The Modbus utility is burned in the I-8000 CD-ROM or can be download at [http://www.icpdas.com/products/PAC/i-8000/modbus\\_web\\_download.htm](http://www.icpdas.com/products/PAC/i-8000/modbus_web_download.htm)



Every i-8KE4/8-MTCP with new plugged IO board should be configured at least once by “Modbus utility”. **If the 2nd & 3rd Leds below the Five 7-Segment-Led is always blinking, it means this i-8KE4/8-MTCP is not configured well by the “Modbus utility”**

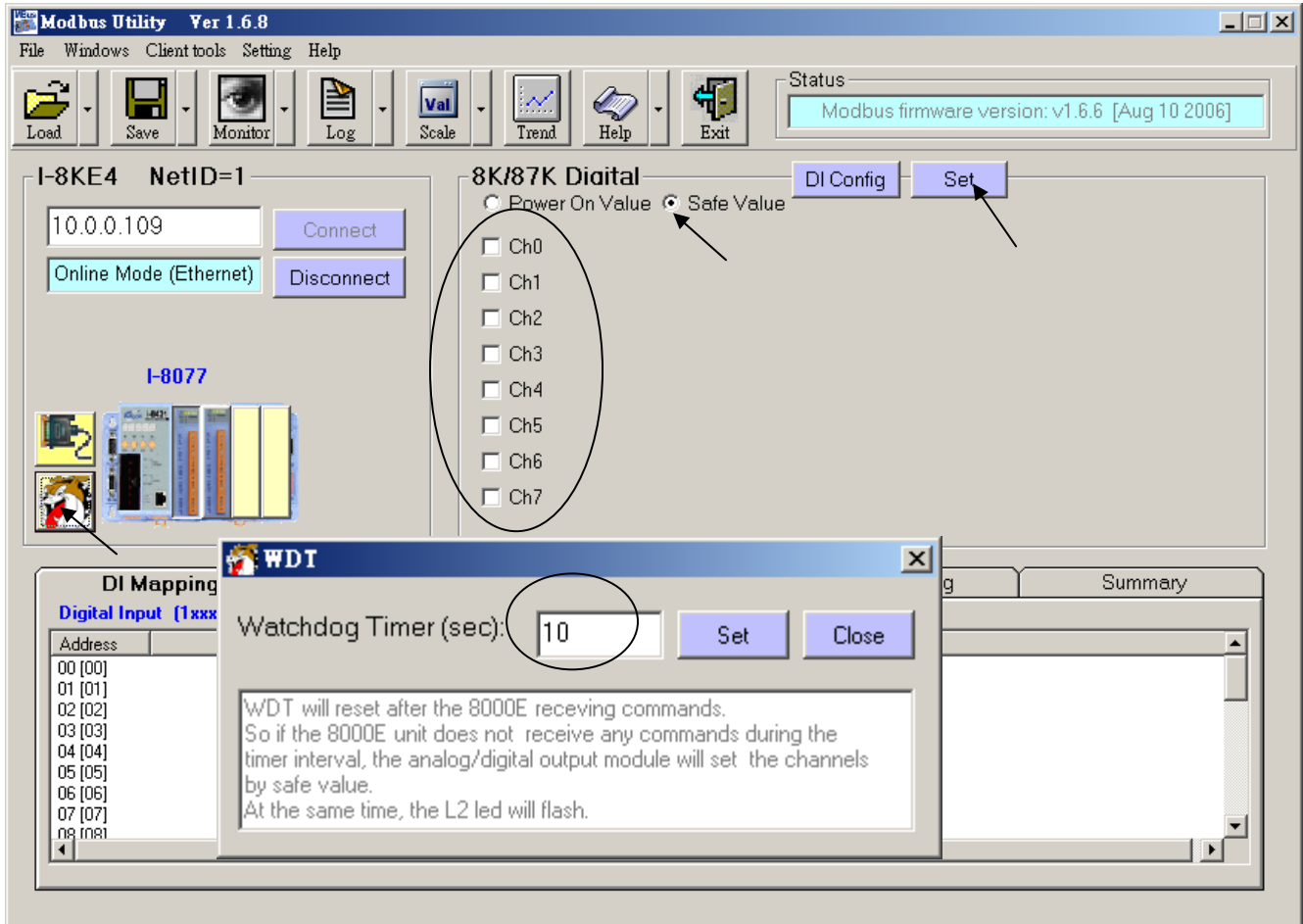


Enter the correct IP of the i-8KE4/8-MTCP on the Modbus utility, and then click on “Connect”. If the i-8KE4/8-MTCP is well connected, You will see the Modbus address assigned in the i-8KE4/8-MTCP. For example, D/I starting from 0 to ... , A/I starting from 0 to ...

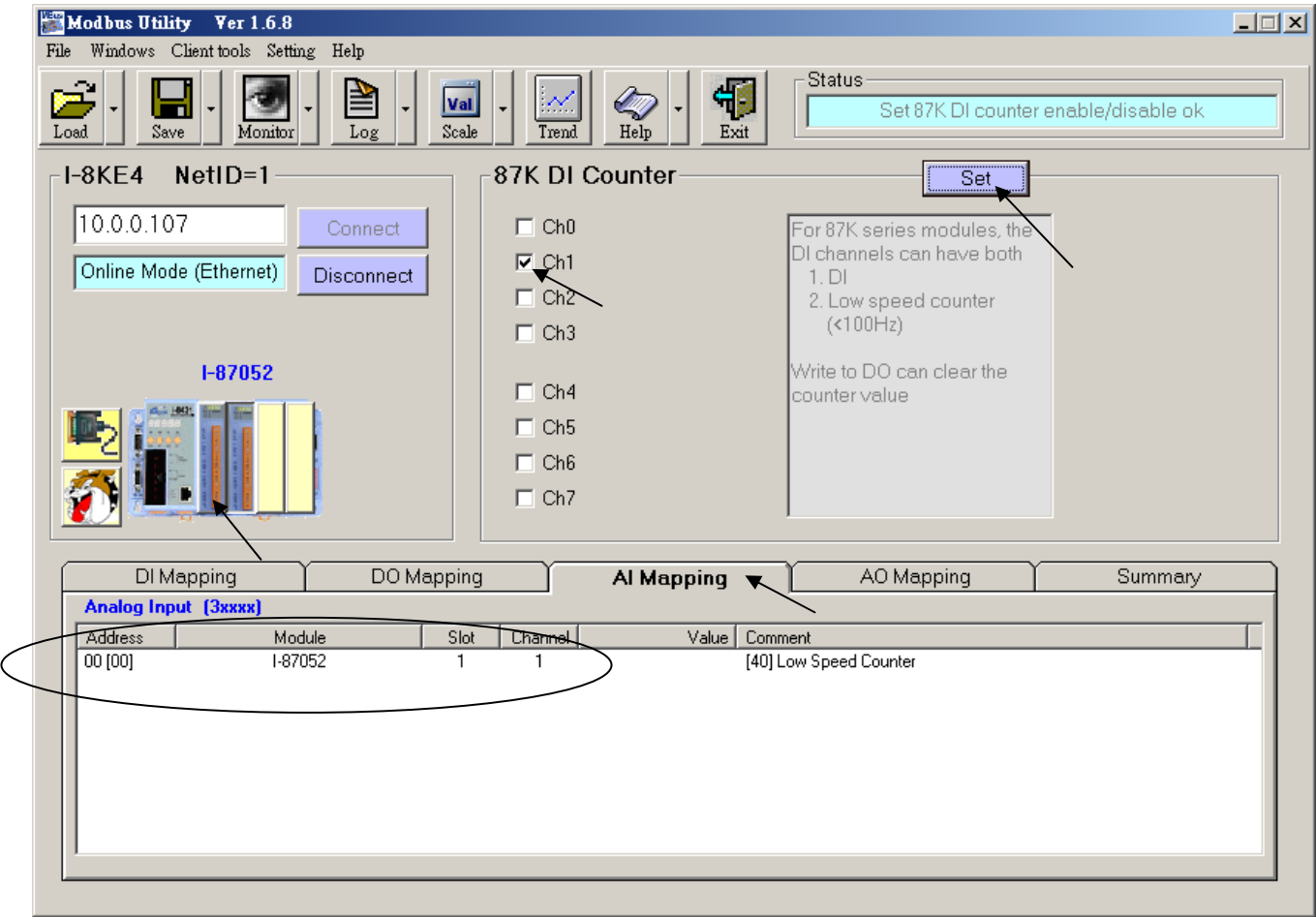


There is a Watchdog setting for the i-8KE4/8-MTCP. The default value is “disable the watchdog” . You may enable it by assigning a “Watchdog timer” value , for example from 10 to 60 seconds. This will automatically set the Digital outputs and Analog outputs of the i-8KE4/8-MTCP to a pre-defined “Safe Value” when the Ethernet communication between the Wincon and the i-8KE4/8-MTCP is break. If you check the Ch0 to ... of the “Safe Value”, it means to set these Channels to have a safe value of “ON” , if you un-check it, it means the related channel has safe value of “OFF”. The “Safe value” function only works when you assign a value larger than zero to the “Watchdog timer” and the communication is break.

Set 0 to the “Watchdog timer” is to disable the Watchdog function.



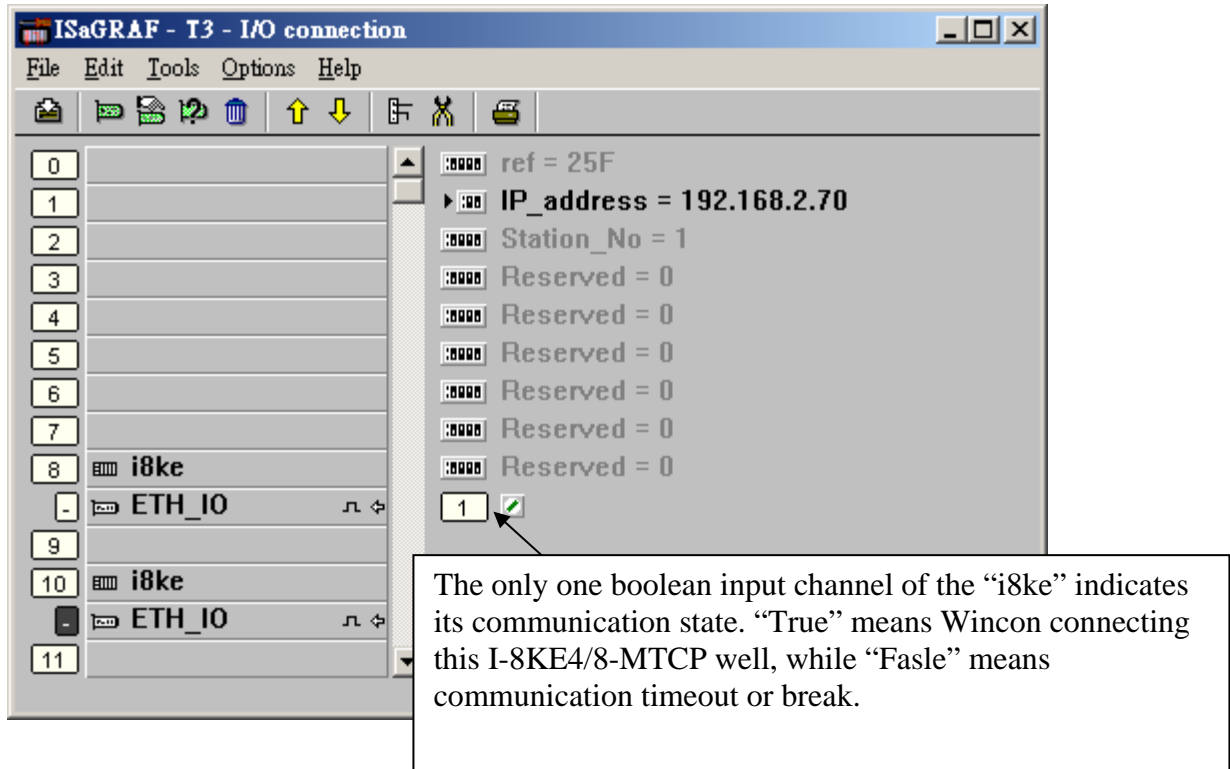
User may enable the “87K DI Counter” if you have plugged i-87K D/I module in the i-8KE4/8-MTCP. Every Digital Input channel of the i-87K module have a D/I Counter value. The max. rate can be accepted is 100 Hz . The DI Counter value is a 16-bits value (0 to 32767, and then from -32768 to -1, Hex is from 0000, 0001 ... to 7FFF , 8000 , 8001, ... to FFFF, then back to 0000, ...). The DI Counter value is a A/I value with a Modbus address.



**Note:**  
Every i-8KE4/8-MTCP with new plugged IO board should be configured at least once by “Modbus utility”. **If the 2nd & 3rd Leds below the Five 7-Segment-Led is always blinking, it means this i-8KE4/8-MTCP is not configured well by the “Modbus utility”**

### Step 3.

Please connect “i8ke” I/O complex equipment in your IsaGRAF project. Please enter the IP address of the related i-8KE4/8-MTCP. If the Wincon has connected more than one i-8KE4/8-MTCP, you should connect more “i8ke” as below.



**If the i8ke , i8ke\_b, i8ke\_n , i8ke\_f , i8ke\_b\_a , i8ke\_n\_a , i8ke\_f\_a is not found in your IsaGRAF, please visit <http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> to download “TCP DAS Utilities For IsaGRAF.zip” to install it again to the IsaGRAF workbench.**

### Step 4.

Please map IsaGRAF internal variables to the related Modbus I/O address of the i-8KE4/8-MTCP by using below functions. (Example program “Wdemo\_30” & “Wdemo\_31” at W-8xx7 CD-ROM :

\napdos\isagraf\wincon\demo\ or

[ftp://ftp.icpdas.com/pub/cd/wincon\\_isagraf/napdos/isagraf/wincon/demo/](ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/)

Please refer to Appendix A.4 for more information about the below functions and refer to Chapter 2.6 for “Variable Array”

|          |   |
|----------|---|
| I8KE_B   | Set Boolean variable as an I-8KE4/8-MTCP Ethernet I/O                 |
| I8KE_B_A | Set Boolean 'Variable Array' as several i-8KE4/8-MTCP Ethernet I/O(s) |
| I8KE_F   | Set REAL variable as an I-8KE4/8-MTCP Ethernet I/O                    |
| I8KE_F_A | Set REAL 'Variable Array' as several i-8KE4/8-MTCP Ethernet I/O(s)    |
| I8KE_N   | Set Integer 'Variable Array' as an I-8KE4/8-MTCP Ethernet I/O (s)     |
| I8KE_N_A | Set Integer 'Variable Array' as several i-8KE4/8-MTCP Ethernet I/O(s) |

## Chapter 23. Wincon and iPAC-8x47 Connecting Fast FRNET Remote I/O

Wincon-8xx7 / 8xx6 (since its driver version of 3.42) and iPAC-8447 / 8847 support FRNET I/O. Please visit below web site for more information.

i-8172 & FRNET I/O module: [http://www.icpdas.com/products/Remote\\_IO/frnet/frnet\\_list.htm](http://www.icpdas.com/products/Remote_IO/frnet/frnet_list.htm)

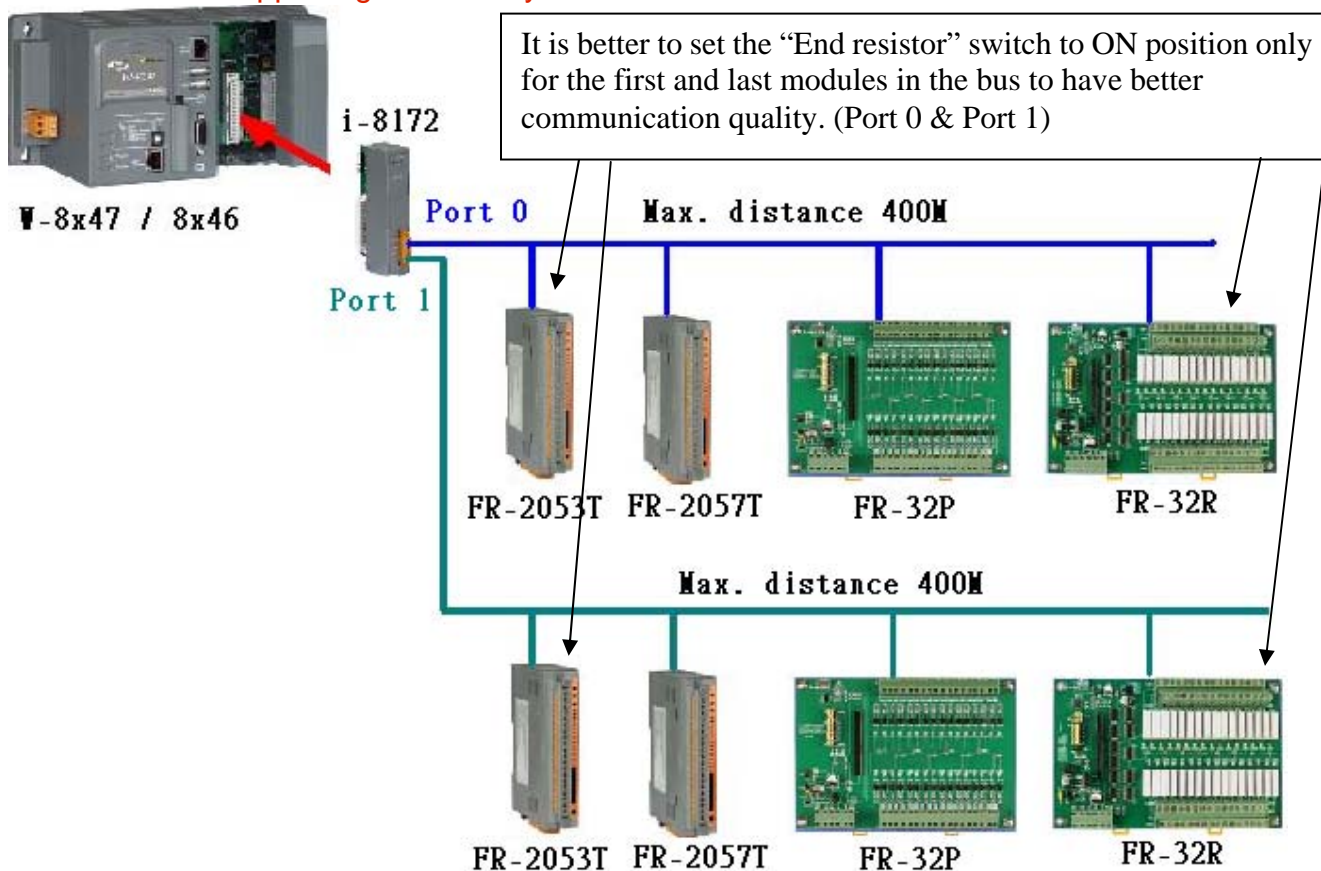
Wincon ISaGRAF driver: <http://www.icpdas.com/products/PAC/i-8000/isagraf-link.htm>

### Advantage of FRNET I/O:

Fast I/O scan, it is about 3 ms / per FRNET I/O scan. (This depends on your program's PLC scan time, for ex, if the ISaGRAF PLC program scan time is about 9ms, then the scan time for all will be 9 ms, not 3 ms. **Below table shows the approximate PLC scan time of an ISaGRAF project which runs only the FRNET setup code (without other codes)**

|                | With 1<br>i-8172 | With 2<br>i-8172 | With 3<br>i-8172 | With 4<br>i-8172 | With 5 to 7<br>i-8172 |
|----------------|------------------|------------------|------------------|------------------|-----------------------|
| W-8xx7/8xx6    | 3 ms             | 4 ms             | 4 ms             | 4 ms             | 5 ms                  |
| iPAC-8447/8847 | 3 ms             | 6 ms             | 9 ms             | 12 ms            | -                     |

FRNET I/O not supporting A/I & A/O yet.



## Important:

1. Every FRNET Output module has a 'RESET' or called 'HOLD' dip on its Dip switch or a special Jumper. User may set it to 'ON' position (or enable it), this will reset the output channels to OFF state when the communication is broken between the i-8172 and the FRNET D/O module. For example, set 8th Dip to ON of FR-2057 means enable it.
2. The communication state of D/I modules can be detected in the 8-Ch. D/I of "i-8172" in the IO connection window. However FRNET Output module doesn't support communication detection.
3. W-8xx7 supports max. 7 pcs. of i-8172 in its Slot 1 through 7, while iPAC-8447 / 8847 supports only max. 4 pcs. of i-8172 in its Slot 0 through 7.

Wincon-8xx7/8xx6 or iPAC-8447 / 8847 plus i-8172 boards (plugged in slot 1 thru. 7) can connect to FRNET I/O modules, for example, FR-2053, FR-2057, FR-32P, FR-32R listed in [http://www.icpdas.com/products/Remote\\_IO/frnet/frnet\\_list.htm](http://www.icpdas.com/products/Remote_IO/frnet/frnet_list.htm)

One i-8172 board has two FRNET ports, ID is port 0 & port 1. Each FRNET port can connect up to 8 FRNET D/O "Module Address" and up to 8 D/I "Module Address". It is very important. The "Module Address" for D/O modules can only be set as 0 to 7, while D/I "Module Address" can only be set as 8 to 15.

The max. I/O channel number for one FRNET "Module Address" is 16. That means one i-8172 can connect max. 2 (ports) x 8 x 16 = 256 ch. of digital output plus max. 2 x 8 x 16 = 256 ch. of digital input. You may plug up to 7 pcs. of i-8172 in the W-8xx7/8xx6 depends on your application. (Max. 4 i-8172 can plug in the iPAC-8447 / 8847)

**Note:** ISaGRAF 3.x can program FRNET I/O by using "i-8172" I/O complex equipment & "fr\_16di", "fr\_16do" & "fr\_b\_a" functions. If your ISaGRAF doesn't support them, please visit

<http://www.icpdas.com/products/PAC/i-8000/isagraf.htm>

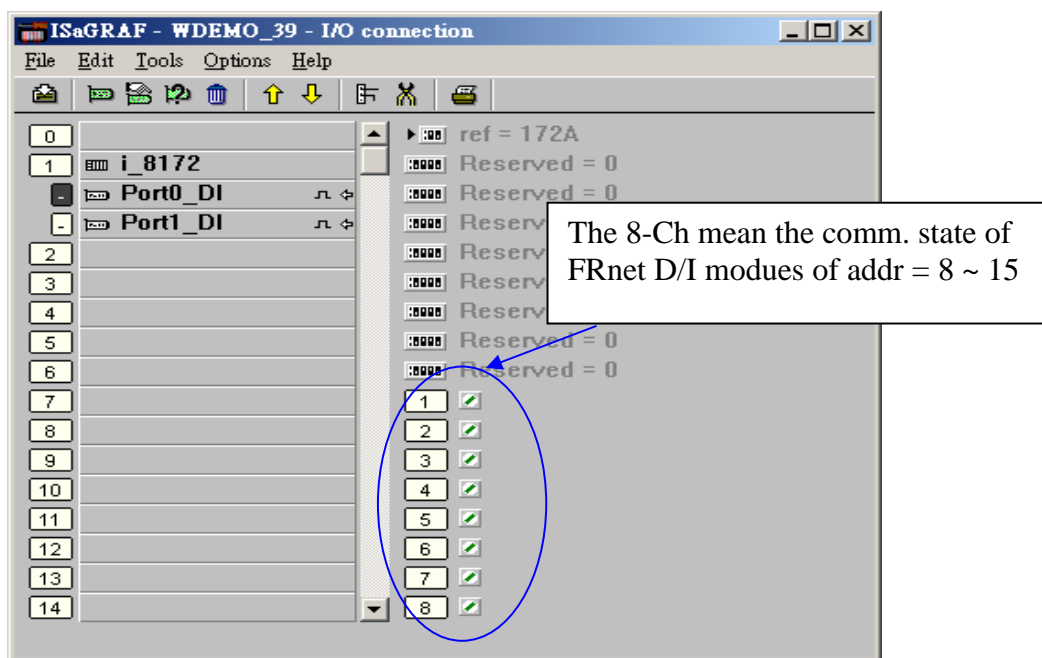
to download the "ICP DAS Utilities For ISaGRAF.zip" to install it again to the ISaGRAF workbench. Or refer to Appendix A.2 of the ISaGRAF User's Manual to restore - "i\_8172", "fr\_16di", "fr\_16do" & "fr\_b\_a" into your ISaGRAF Workbench.

Below is a demo program show you how to program FRNET I/O. This ISaGRAF example program can be download at

[www.icpdas.com](http://www.icpdas.com) > FAQ > Software > ISaGRAF > 082 or visit

[ftp://ftp.icpdas.com/pub/cd/wincon\\_isagraf/napdos/isagraf/wincon/demo/ to download](ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/ to download)  
"Wdemo\_070.pia"

Step 1: Connecting i-8172 in the related slot in the IO connection windows. W-8xx7/8xx6's slot No. is from 1 to 7 (Slot 0 to 7 for the iPAC-8447 / 8847).



Step 2: Declaring ISaGRAF variable

| Name                   | Type    | Attribute       | Description                                      |
|------------------------|---------|-----------------|--|
| INIT                   | Boolean | <b>Internal</b> | Init as True                                     |
| Dump_di                | Boolean | <b>Internal</b> | boolean variable for non-using FRNET D/I channel |
| Dump_do                | Boolean | <b>Internal</b> | boolean variable for non-using FRNET D/O channel |
| FR_IN01 ~<br>FR_IN16   | Boolean | <b>Internal</b> | Will map to 16-chanel FRNET DI channels          |
| FR_OUT01 ~<br>FR_OUT16 | Boolean | <b>Internal</b> | Will map to 16-chanel FRNET DO channels          |

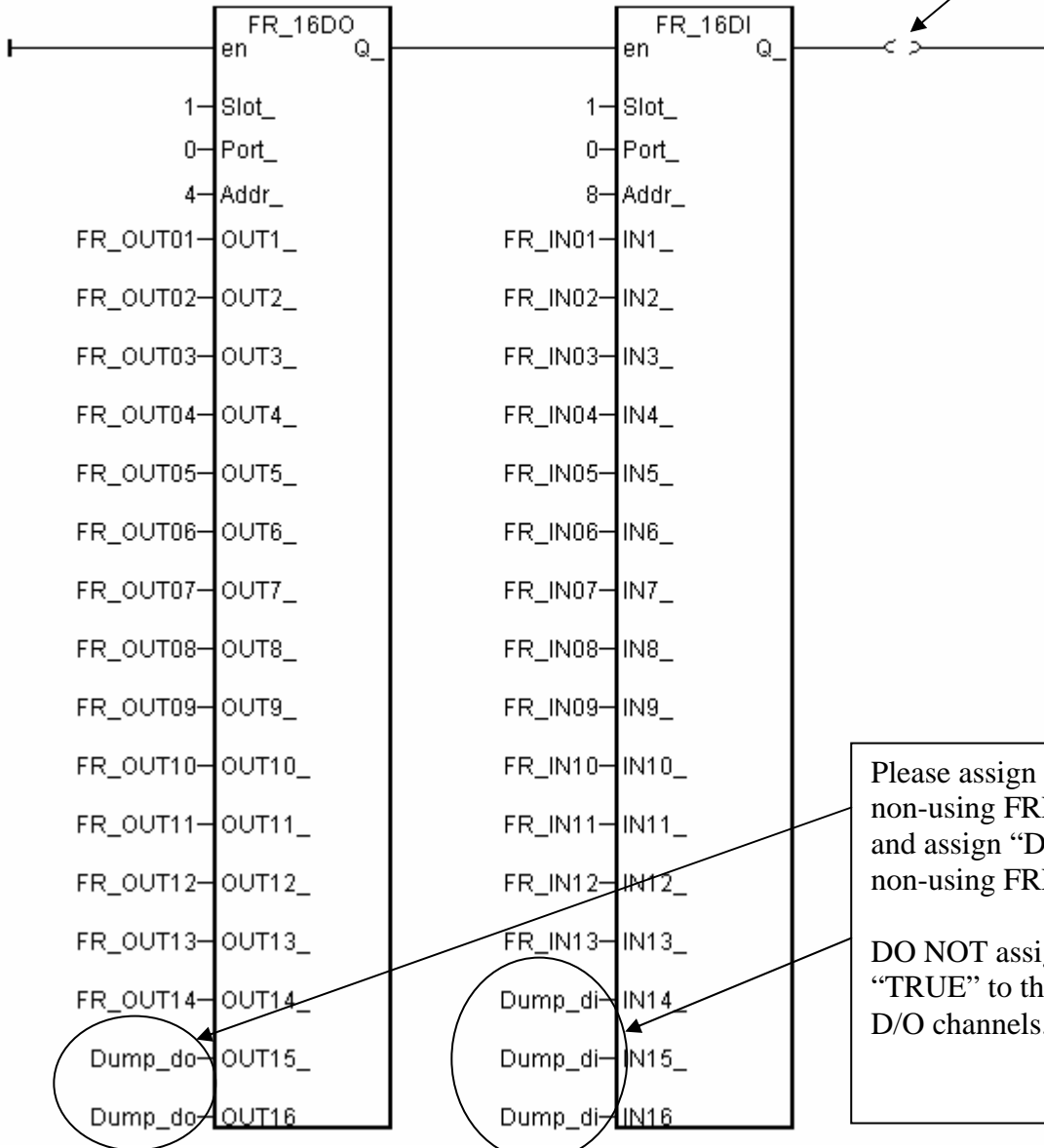


### Step 3: writing LD program to map ISaGRAF boolean variable as FRNET I/O

(\* INIT should be declared with an initial value of TRUE. The below code can only run once in the first PLC scan cycle, please don't use them in other PLC scan cycles \*)



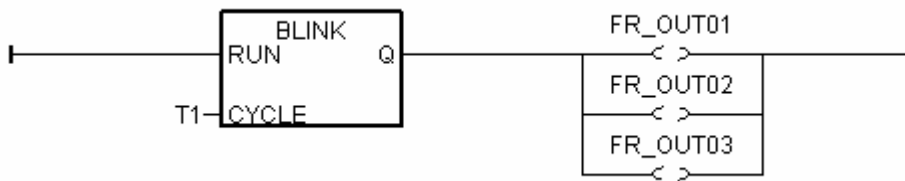
The “Q\_” output of the “Fr\_16di” and “Fr\_16do” is always TRUE, so many “Fr\_16di” and “Fr\_16do” can be used in the same LD rung.



Please assign “Dump\_di” to non-using FRNET D/I channels and assign “Dump\_do” to non-using FRNET D/O channels.

DO NOT assign “False” or “TRUE” to the non-using D/I and D/O channels.

Step 4: Write a Ladder program to blink output of FR\_OUT01 to 03



Step 5: How to test ?

Please plug i-8172. into slot 1 of the W-8xx7/8xx6 (make sure the Wincon ISaGRAF driver version is 3.31 or later) or in the iPAC-8447 / 8847 .

Please connect the i-8172 's Port 0 in this demo to one FR-2053 (16-Ch. DI) and one FR-2057 (16-Ch. DO).

The FR-2053's ADDR = 8 (dip switch 4 = ON, other dips 1,2,3, ,5,6,7,8 is OFF)

The FR-2057's ADDR = 4 (dip switch 3 = ON, other dips 1,2, ,4,5,6,7,8 is OFF)

Note:

-----  
FRNET DO module, for example FR-2057 can only set module ADDR as 0 to 7  
FRNET DI module, for example FR-2053 can only set module ADDR as 8 to 15  
-----

Then after you download this ISaGRAF project into W-8xx7/8xx6, you will see the DO1 to 3 is blinking in the period of 0.5 second.

This ISaGRAF example program can be download at

[www.icpdas.com](http://www.icpdas.com) > FAQ > Software > ISaGRAF > 082 or visit

[ftp://ftp.icpdas.com/pub/cd/wincon\\_isagraf/napdos/isagraf/wincon/demo/](ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/) to download  
"Wdemo\_070.pia"

---

## Chapter 24. Some Skill To Operate The RS-232 / 422 / 485 serial COM Port By COM functions

---

ICP DAS ISaGRAF controllers support below Serial COM Port (RS-232/422/485) protocols:

|                               |   |
|-------------------------------|---|
| Modbus RTU Slave              | Refer to Chapter 4 of the ISaGRAF user's manual & respective getting started manual |
| I-7000 and I-87xxx RS-485 I/O | Refer to Chapter 6 of the ISaGRAF user's manual                                     |
| Modbus RTU Master (M-7000)    | Refer to Chapter 8 and 21 of the ISaGRAF user's manual                              |
| Modbus ASCII Master           | Refer to Chapter 8 of the ISaGRAF user's manual                                     |
| Modem Link                    | Refer to Chapter 13 of the ISaGRAF user's manual                                    |
| MMICON                        | Refer to Chapter 16 of the ISaGRAF user's manual                                    |
| SMS : Short Message Service   | Refer to Chapter 17 of the ISaGRAF user's manual                                    |

User can apply below COM functions to operate other protocols or 3rd party protocols. (Please refer to Appendix A.4 of the ISaGRAF user's manual for description of these COM functions )

|           |   |
|-----------|---|
| COMOPEN   | Open Serial COM Port (without "Flow control" parameter)   |
| COMOPEN2  | Open Serial COM Port (with "Flow control" parameter, not for I-8xx7)                                  |
| COMREADY  | Test if any byte come in  |
| COMARY_R  | Read all bytes which already come in to a byte array  |
| COMARY_W  | Write many bytes in a byte array to COM Port  |
| COMREAD   | Read one bytes (Please call "COMREADY" to test first, if there is data, then "COMREAD" can be called) |
| COMCLEAR  | Clear all received bytes in the receiving buffer  |
| COMARY_NW | Write one signed long Integer to COM Port, format is Binary, 4-byte                                   |
| COMARY_WW | Write one signed Word to COM Port, format is Binary, 2-byte   |
| COMSTR_W  | Write one string to COM Port  |
| COMWRITE  | Write one byte to COM Port  |
| COMCLOSE  | Close Serial COM Port   |

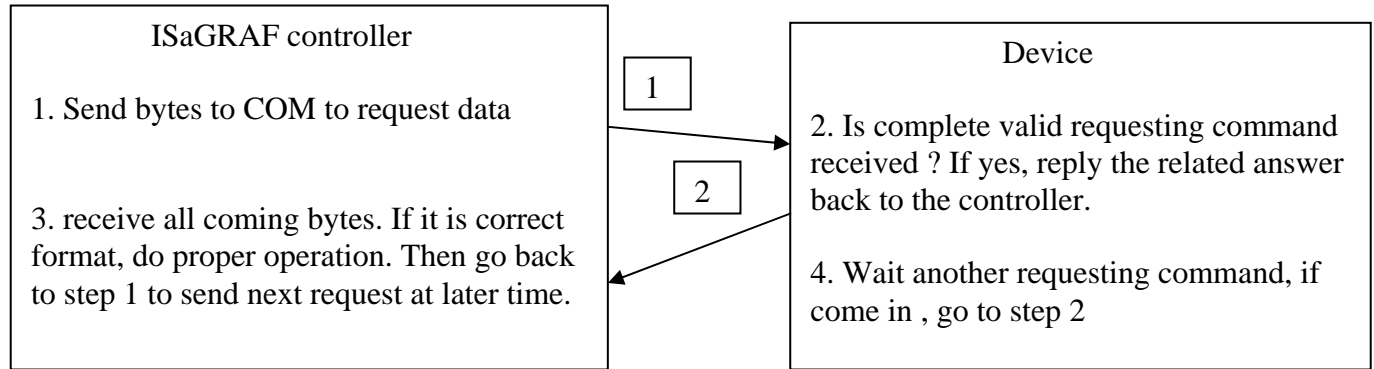
Note:

1. The default shipping of I-8xx7 controller has set its COM1 and COM2 (COM2:RS-485 is only for I-8417/8817) as Modbus RTU Slave Port. User can choose to switch off the COM1:Modbus RTU Slave function to become a freely used COM port by the above listed COM functions. (Please refer to Appendix C.1). To use i-8xx7 's COM5 to COM20, Please refer to Chapter 1.8 to install I-8112/8114/8142/8144 serial expansion boards.
2. Wincon-8xx7 / 8xx6 's COM2 / COM3 can be switched ON as a Modbus RTU Slave Port. Or Switch Off for freely used. (Refer to Appendix A.2 of its Getting Started manual delivered with the hardware). To use Wincon 's COM5 to COM14 at I-8112/8114/8142/8144 serial expansion boards, please refer to Appendix E of the "Getting Started:Wincon ISaGRAF PAC" manual.
3. COM1 of I-7188EG / 7186EG is set as Modbus RTU Slave port when shipping. User may switch it OFF to freely use it by COM port functions. (Please refer to its "Getting Started Manual" delivered with its hardware). However I-7188XG 's COM1 can not be switch OFF, it is always Modbus RTU Slave port. If user want to use COM3 to COM8 of I-7188EG/XG and 7186EG, please plug one extra X-5xx expansion I/O board inside it .

[http://www.icpdas.com/products/PAC/i-o\\_expansion/x\\_list.htm](http://www.icpdas.com/products/PAC/i-o_expansion/x_list.htm)

## 24.1: Controller send 1 request and get 1 reply from device

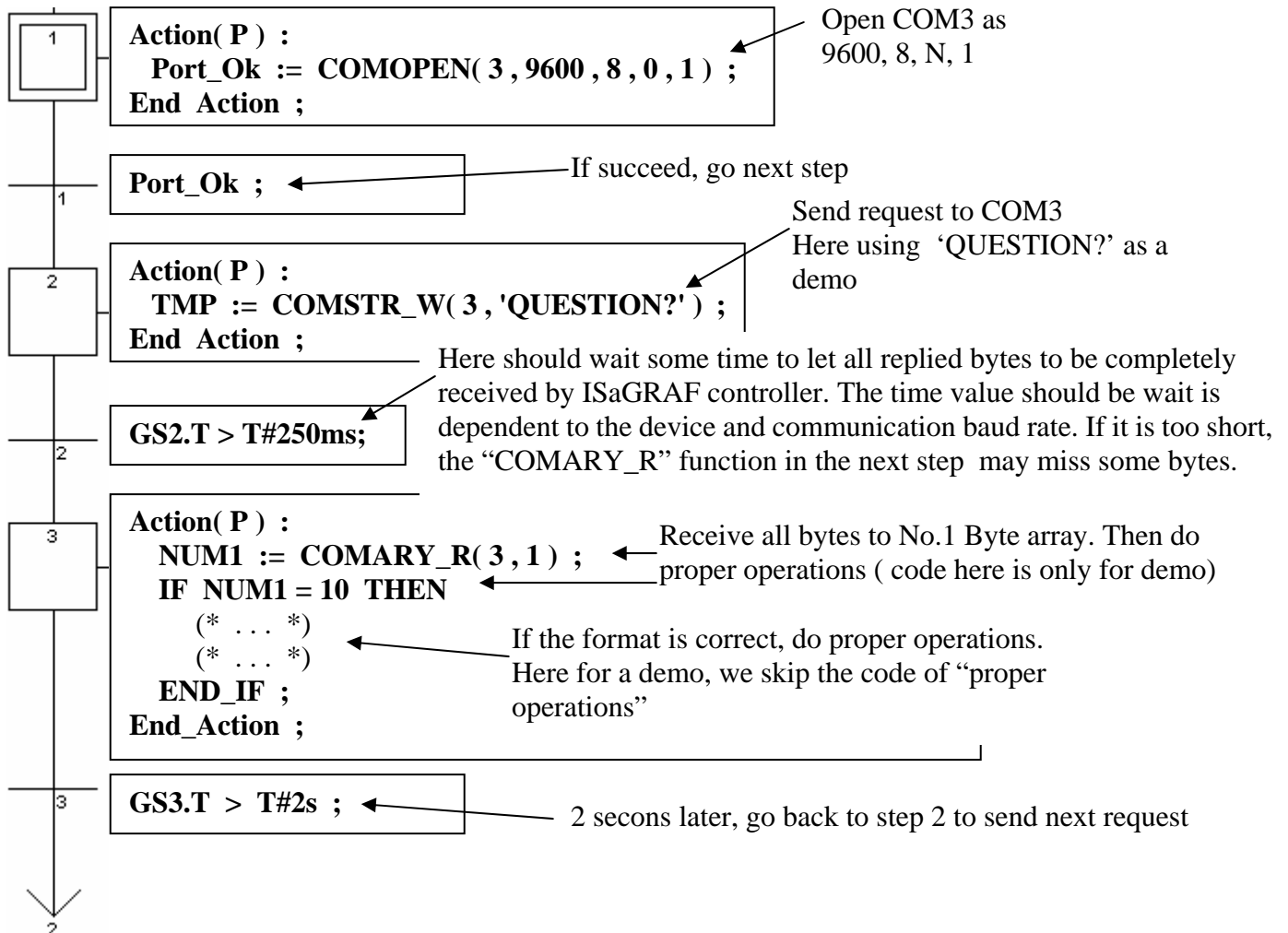
Below figure lists the most common RS-232 / 422 / 485 application.



User can use the below code or similar code to do it.

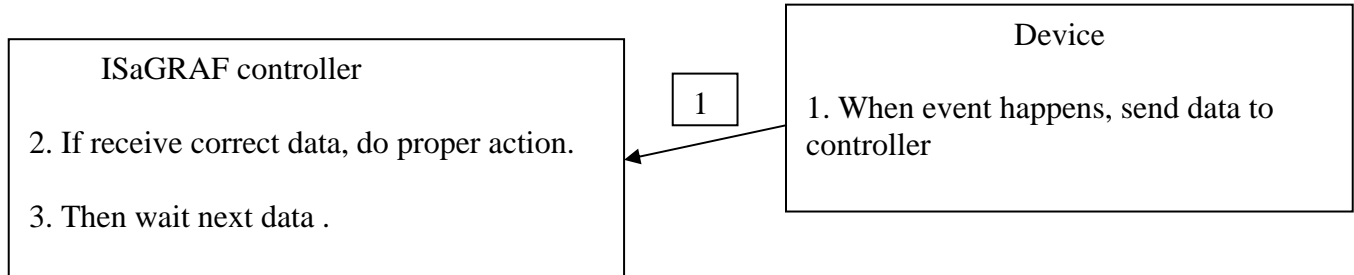
Below example will send a string "QUESTION?" to device via COM3 , then waiting device to reply the related answer. And then 2 seconds later, send next same question to device , ...

SFC program: ("Port\_OK" & "TMP" is Boolean Internal , "NUM1" is Integer Internal )



## 24.2: Controller just wait data from the remote device

This kind of application is very common in the store. Like the device of “Bar code reader”, when it reads bar code on the product, it will send the related data to the controller via RS-232 / 422 / 485. The controller just receive it, not necessary to send any byte to device. (Please visit [www.icpdas.com](http://www.icpdas.com) – FAQ – Software\_ISaGRAF for demo program and more information.)



ST program:

```

IF INIT THEN
  INIT := FALSE ;
  Port_Ok := COMOPEN( 3 , 9600 , 8 , 0 , 1 ) ;
  T1 := T#0s ;
  STEP := 0 ;
END_IF ;
  
```

Variable declaration

“INIT” is Boolean Internal, init as TRUE  
 “Port\_Ok” & “TMP” is Boolean Internal  
 “T1” is Timer Internal  
 “STEP” & “NUM1” is Integer Internal

Open COM3 as 9600, 8, N,1

```

IF Port_Ok = False THEN
  Return ;
END_IF ;
  
```

If open fail, just quit this ST program

**CASE STEP OF**

```

0 : IF COMREADY(3) THEN
  STEP := 1 ;
  T1 := T#0s ;
  Tstart(T1) ;
END_IF ;
  
```

STEP=0 means “waiting state”, it should check if any byte come in. If “COMREADY” returns TRUE, it means there is at least one byte come in. Then set “STEP” to 1 and start timer “T1” to tick.

```

1 : IF T1 > T#250ms THEN
  Tstop(T1) ;
  T1 := T#0s ;
  STEP := 0 ;
  NUM1 := COMARY_R(3, 1) ;
  
```

STEP=1 means “data is coming”, when “T1” tick to the specified waiting time, call “COMARY\_R” to receive all bytes to No.1 byte array (this will make sure all bytes from the “device” all well received). This waiting time is dependent with the “Device” and the communication baud rate. Here we use 0.25 second as a demo. (If setting too short than your device should have, some bytes may be missing) Remember to set “STEP” to 0 to wait next data.

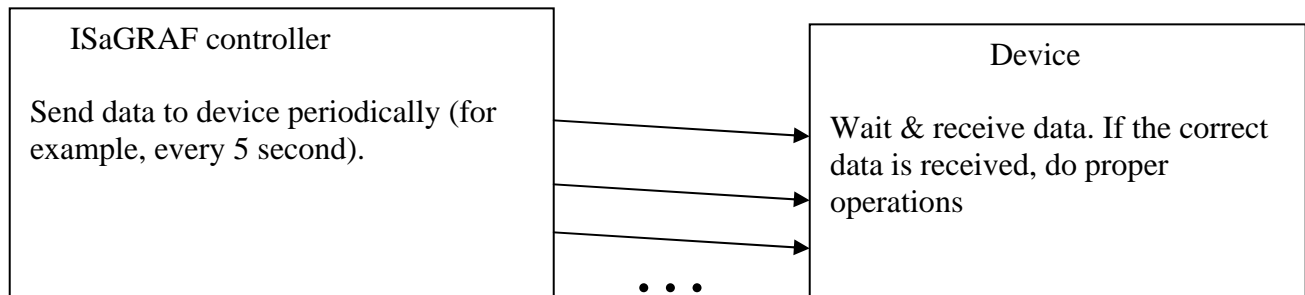
```

  IF NUM1=10 THEN
    (* ... *)
  END_IF ;
END_IF ;
  
```

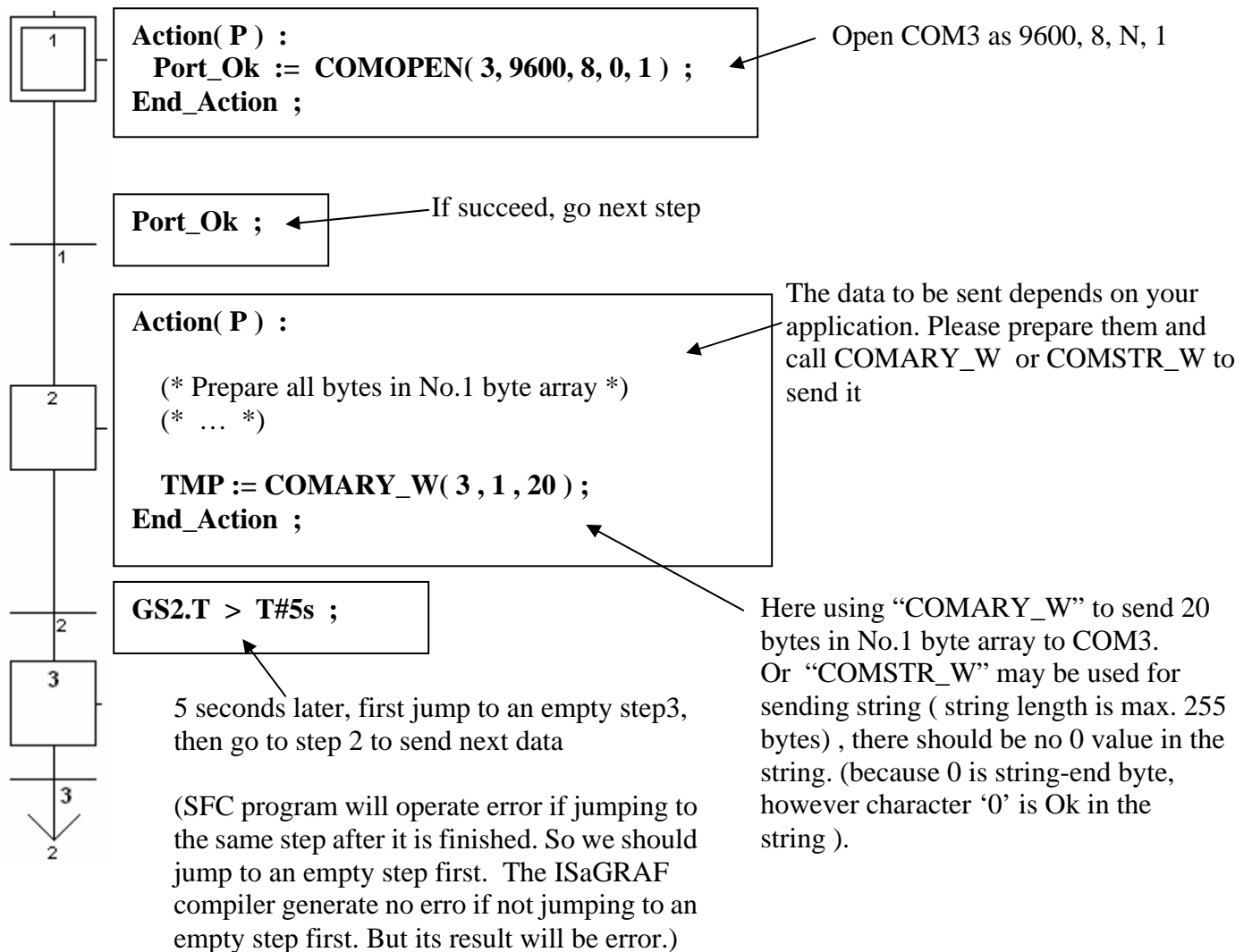
Do proper operation if receive correct data.  
 The code listed here is only for demo. (It is depend with your application, we just assume the correct data has 10 bytes here.)

**END\_CASE ;**

## 24.3: Report data to remote device periodically



SFC program: (Please declare “TMP” & “Port\_Ok” as Boolean Internal)



## **24.4: Controller send data when event happens**

---

Please refer to Chapter 11.3.5