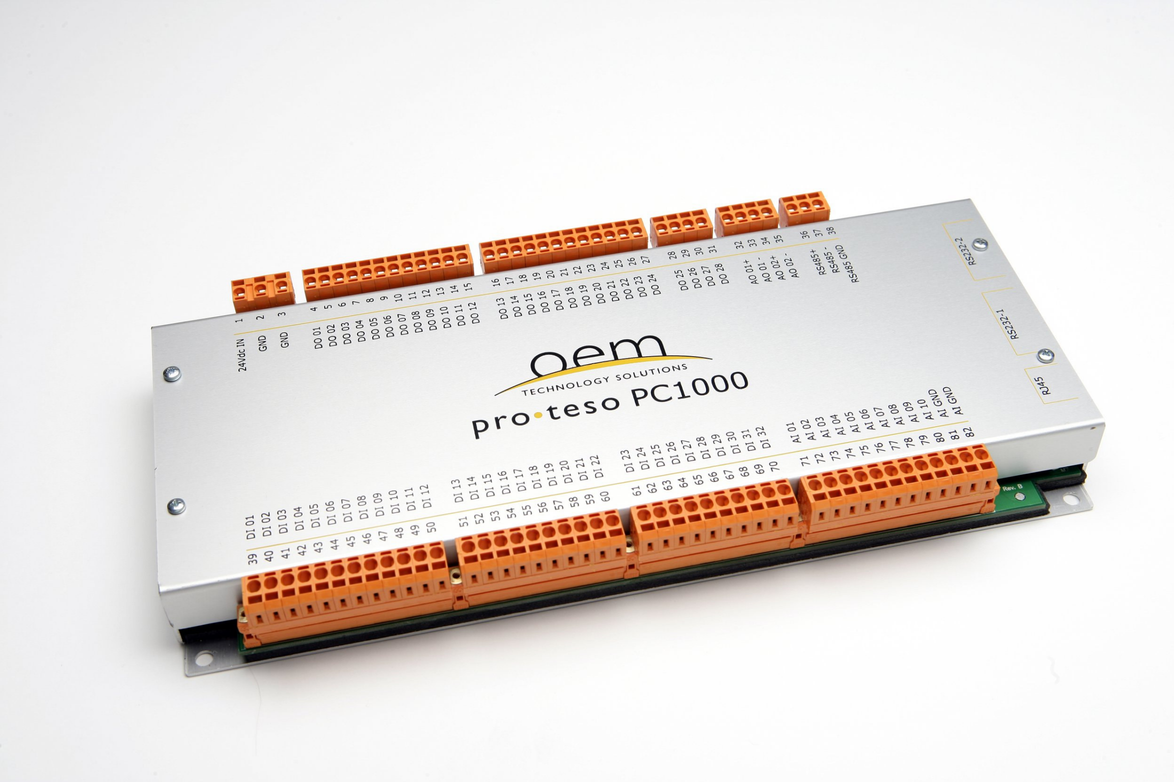
EMBEDDED PLC



OEM TECHNOLOGY SOLUTIONS

pro • teso PC1000 CONTROLLER

EMBEDDED PLC
USER’S MANUAL

oem
TECHNOLOGY SOLUTIONS

TURNING YOUR PRODUCT VISION INTO REALITY
www.oem.net.au

SYSTEM ENGINEERS | SOFTWARE ENGINEERS | ELECTRONICS ENGINEERS

Research | Specify | Design | Develop | Validate | Support | Custom Product Manufacture | Configuration | Warranty

COPYRIGHT

EMBEDDED PLC pro•teso PC1000 Controller User's Manual • Part Number: TD-0005-4

TRADEMARKS

EMBEDDED PLC is a registered trademark of OEM Technology Solutions Pty Ltd.

Microsoft, Windows, Windows XP, Windows 2000 and Windows NT are registered trademarks of Microsoft Corporation in the United States and/or other countries.

ISaGRAF is a registered trademark of ICS Triplex ISaGRAF Inc.

Rabbit and Rabbit 3000 are registered trademarks of Rabbit Semiconductor.

RabbitNet is a trademark of Z-World Inc.

Dynamic C is a registered trademark of Z-World Inc.

Indusoft is a registered trademark of Indusoft Ltd.

All other brand or product names are trademarks or registered trademarks of their respective holders.

NOTICE TO USERS

When a system failure may cause serious consequences, protecting life and property against such consequences with a back-up system or safety device is essential. The buyer agrees that protection against consequences resulting in system failure is the buyer's responsibility.

This device is not approved for life support or medical systems.

OEM Technology Solutions

Unit 13, 82 Reserve Road
Artarmon NSW 2064
AUSTRALIA

Telephone: +61 (2) 9966-9424
Fax: +61 (2) 9966-9429

www.oem.net.au

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

This page is intentionally left blank

1. INTRODUCTION

1.1 WHAT IS EMBEDDED PLC?

The EMBEDDED PLC system turns a Rabbit-based controller, such as the OEM pro•teso PC1000, into a high performance and yet inexpensive Programmable Logic Controller (PLC). The Target PLC can be programmed using the EMBEDDED PLC – ISaGRAF® Workbench in any of the standardized automation control programming languages IEC 61131-3:

- Sequential Function Chart (SFC)
- Function Block Diagram (FBD)
- Ladder Diagram (LD)
- Structured Text (ST)
- Instruction List (IL)

EMBEDDED PLC – ISaGRAF Workbench becomes an alternative to Dynamic C to program Rabbit-based controllers.

The EMBEDDED PLC system consists of two parts:

- The ***Target PLC***: The OEM Technology Solutions pro•teso PC1000 Controller with the RabbitCore® module RCM3100 (Rabbit 3000 microprocessor, 29.4 MHz clock speed, 512 kB Flash and 512 kB Battery-backed SRAM, OEM P/N 010-0232-001) or RabbitCore module RCM3360 (Rabbit 3000 microprocessor, 44.2 MHz clock speed, 512 kB Flash, 512 kB fast SRAM, 512 kB of battery-backed SRAM, 10/100Base-T Ethernet, 16 MB NAND flash and xD connector, OEM P/N 010-0232-001-E) and loaded with the EMBEDDED PLC Firmware Kernel (or refer to as the Kernel). The Firmware Kernel is distributed as a .bin file which is loaded into the Target through the EMBEDDED PLC Utility. The Kernel runs the PLC application in a loop (reading inputs → evaluate the logic → updating outputs) and also provides the communication interface between the Target and the Workbench.
- The ***ISaGRAF Workbench***: The Workbench provides a complete PLC programming environment, including an editor for each of the PLC programming languages, downloading PLC application to the target, simulation, program-debugging tools and on-line monitoring of the PLC application.

1.2 EMBEDDED PLC PRO•TESO PC1000 OVERVIEW

The pro•teso PC1000 is an advanced controller that incorporates flash memory, battery backed SRAM, 32 digital inputs, 28 FET digital outputs, 10 12-bit A/D inputs, 2 10-bit D/A outputs, RS232 and RS485 serial communications ports and an optional Ethernet interface (10/100 Mbps).

The pro•teso PC1000 has been designed to operate in tough industrial environments, and has been used in process control plants, distributed control systems and transportation monitoring and control applications. The pro•teso PC1000 controller has been designed and tested to the EN50155 Standard – Electronic Equipment used in rail vehicles.

The pro•teso PC1000 can be individually customised for Original Equipment Manufacturers (OEMs) to include your company name and logo if required.

Due to the low profile design, the pro•teso PC1000 controller has excellent shock and vibration performance capabilities and is ideally suited to the harsh environments where conventional rack mounted PLC's can fail.

Main features of the EMBEDDED PLC Firmware Kernel running on the Pro•teso PC1000 Target are shown in Table 1.

Table 1: Main Features of the EMBEDDED PLC Firmware Kernel V1.xx on the pro•teso PC1000 controller

Description	PC1000	PC1010	PC1020	PC1030
Microprocessor	Rabbit 3000			
Processor Speed	29.4 MHz	44.2 MHz		
I/Os supported	- 32 Digital Inputs - 28 Digital Outputs (with the option of reading their status individually as digital feedback inputs) - 10 Analogue Inputs: 12-bit resolution, configurable 0 – 10 VDC or 0 – 20 mA - 2 Analogue Outputs: 10-bit resolution, configurable 0 – 10 VDC or 0 – 20 mA			
Protocols[1]	Modbus Slave RTU over Serial Line (RS232 or RS485)[2]	Modbus Slave RTU over Serial Line or TCP/IP using static IP address (optional). Modbus Master over TCP/IP.		
Performance				
Digital Inputs scan time	Approx. 3.2 ms (all digital and feedback inputs).			
Analogue Inputs scan time	Approx. 6 ms (all analogue outputs).			
Digital Output update time	Approx. 140 µs per output.			
Analogue Output scan time	Approx. 560 µs per output.			
Boolean instruction execution time	Approx. 50 µs.			
Memory Space	Maximum size of ISaGRAF application database is 50000 bytes. Size of ISaGRAF real-time database (holds variables, SFC engine data, C function and function blocks data) is 10000 bytes. EMBEDDED PLC firmware kernel total code size is approx. 260000 bytes.			
ISaGRAF V3.5 features not supported by EMBEDDED PLC	- On-line modifications. This feature enables the user to modify the application while the process is running. - Uploading application stored in the target.			

1.3 ABOUT THIS MANUAL

This manual provides instructions for installing, testing, configuring and interconnecting the pro•teso PC1000 OEM Technology Solutions Controller running the EMBEDDED PLC™ Firmware Kernel. Instructions are also provided for using the ISaGRAF® IEC 61131-3 Soft Logic system.

This User's Manual is divided into the following chapters:

- This chapter provides and overview of the EMBEDDED PLC and information about this publication such as conventions used and reference documents.
- Chapter 2, Getting Started: Explains how to install the different EMBEDDED PLC products and run a sample application.

---

[1] Minimum polling period time is 1 cycle duration for Modbus TCP and 2 x cycle duration for Modbus RTU over RS232/RS485. Cycle duration corresponds to the time that kernel executes one cycle (read inputs → execute logic → update outputs)

[2] Baud Rates supported 600 to 19200 (configurable using EMBEDDED PLC Utility); No parity; 8 Data Bits; 1 Stop Bit; No flow control.

- Chapter 3, Running Sample Applications: Provides instructions how to run the different types of sample applications and also how to create a PLC application using the ISaGRAF Workbench.
- Chapter 4, Hardware Reference: Provides a detailed description of I/Os and communication ports present in the OEM Technology Solutions pro•teso PC1000 Controller and how to configure them in order to work with EMBEDDED PLC.
- Chapter 5, Software Reference: Provides a detailed description of the ISaGRAF library elements (Software configuration of I/O boards, C functions and C function blocks) available from the ISaGRAF Workbench.
- Chapter 6, Using EMBEDDED PLC Utility: Detailed description of the EMBEDDED PLC Utility.
- Chapter 7, FAQ and Troubleshooting: Describes the known issues with this release, frequently asked questions and troubleshooting.
- Chapter 8, Specifications: Describes the electrical, mechanical and environmental specifications of the pro•teso PC1000.
- Chapter 9, About Indusoft: A brief description of Indusoft Web Studio and how can be used in conjunction with EMBEDDED PLC.

1.4 TEXT CONVENTIONS

This manual uses special formatting to help you quickly identify certain items, as follows:

- Titles, labels, and new terms are indicated using *italic* text.
- File names, messages and screen text are indicated using bold, courier text (for example, **`C:\OEMTech\Install.exe`**).
- Variables and information you must provide are indicated using bold, italicized text enclosed in < and > brackets (for example, run the ***<driver name>.exe*** file indicates you must provide a driver name).
- Buttons, menu options and keyboard key are indicated in **bold**.
- Text requiring special emphasis is in ***bold italic***.
- Some information is segregated into **Note** and **Caution** boxes as follows:
    - ✍ **Notes** provide supplemental information related to the surrounding text, usually the text just preceding the note.
    - ✋ **Caution** provides information necessary to prevent damage or problems.

1.5 WINDOWS CONVENTIONS

This manual uses the following Windows conventions:

- Dialog boxes (or dialogs) are windows that allow you to enter information.
- Text boxes (or fields) are areas in dialogs where you can type in text.
- Radio buttons are white circles in which a black dot appears or disappears when you click on the button. Typically, the black dot indicates that the option or function is enabled (active). The absence of a black dot indicates the option or function is disabled (inactive).
- Check-boxes are white squares in which a check appears or disappears when you click on it with the cursor. Typically, the check indicates that the option or function is enabled (active). The absence of a check indicates the option or function is disabled (inactive).
- Buttons are icons in boxes that appear “pressed” when you click on them.
- Lists are panes (white boxes) in windows or dialog boxes containing two or more selectable options.
- Combo-boxes have arrows that, when clicked, show part or all of an otherwise concealed list.

✍ **Note:**

The dialogs and procedures described in this manual are valid for Windows 2000 and XP. Some terms may vary according to the operating system (type, language and version) you are using.

1.6 REFERENCE DOCUMENTS

[1] *ISaGRAF Version 3.5 User's Guide*, 1994 – 2004 ICS Triplex ISaGRAF Inc.

[2] Modbus Application Protocol Specification V1.1a.

1.7 ABBREVIATIONS AND TERMINOLOGY

The following abbreviations are terms used throughout the document.

Term	Description
A/D	Analogue to Digital
BSP	Board Support Package
D/A	Digital to Analogue
DDE	Dynamic Data Exchange
FAQ	Frequently Asked Questions
FBD	Function Block Diagram
FC	Flow Chart
HMI	Human Machine Interface
HVAC	Heating, Ventilation and Air-Conditioning. This is sometimes referred to as climate control.
I/O	Input/Output
IL	Instruction List
IWS	Indusoft Web Studio
LAN	Local Area Network
LD	Ladder Diagram
LED	Light Emitting Diode
ODBC	Open Database Connectivity
OPC	OLE (Object Linking and Embedding) for Process Control
PC	Personal Computer
PLC	Programmable Logic Controller
RAM	Random Access Memory
SBC	Single Board Computer
SCADA	Supervisory Control And Data Acquisition
SFC	Sequential Function Chart
SRAM	Static Random Access Memory
ST	Structured Text
TCP/IP	Transport Control Protocol/Internet Protocol
USB	Universal Serial Bus

2. GETTING STARTED

2.1 INTRODUCTION

This chapter will help you to quickly install the necessary software to run a PLC sample application on the pro•teso PC1000 target using the ISaGRAF Workbench. This chapter contains also important information about licensing your product.

To run the EMBEDDED PLC software products it is required to have a Personal Computer (PC) with at least a 133-MHz processor (550-MHz or faster processor is recommended). This PC will be referred as the *Host*. The PC must have the following hardware and software configuration:

- Windows 2000 or XP operating system
- 128 MB of RAM required; 256 MB or more recommended
- 1.25 to 2 GB of available hard-disk space
- CD-ROM or DVD-ROM drive
- VGA or hardware that supports console redirection required; Super VGA supporting 800 x 600 or higher-resolution monitor recommended
- 1 serial port (COM1 – COM4)
- Ethernet card (optional)

Note:

✍ The current version of ISaGRAF 3.5 software cannot be installed on Windows Vista PC.

The pro•teso PC1000 controller comes pre-loaded with the EMBEDDED PLC Firmware Kernel. If you require to load the Firmware Kernel to another controller or to upgrade the current one, go to Chapter 7, which describes in detail the installation of the additional software and how to load or upgrade the EMBEDDED PLC Firmware Kernel to the Target.

2.2 SOFTWARE INSTALLATION

The EMBEDDED PLC CD-ROM contains the following software:

- **`ISaGRAF_V3.5`**: EMBEDDED PLC – ISaGRAF V3.50 Workbench installer. This particular installation is intended to work in stand-alone mode or only with EMBEDDED PLC Rabbit-based targets. After installation of the Workbench, the software runs in a trial mode for 30 days. After period expires, license is required.
- **`OEM-PC1000_BSP_V1.xx_9-0004-xxx`**: EMBEDDED PLC Board Support Package (BSP) for the pro•teso PC1000 controllers. It contains the EMBEDDED PLC Firmware Kernel and the ISaGRAF I/O boards and sample files to be installed in the ISaGRAF directory.
- **`EmbeddedPLCUtility_V1.xx_3-0237x`**: Microsoft Windows Installer of the EMBEDDED PLC Utility. The EMBEDDED PLC Utility allows you to load or upgrade the EMBEDDED PLC Firmware Kernel to a Rabbit-based Target. The installation of this software is optional and is described in Chapter 7.

These software products can also be downloaded (in ZIP format) from the OEM Technology Solutions website http://www.oem.net.au.

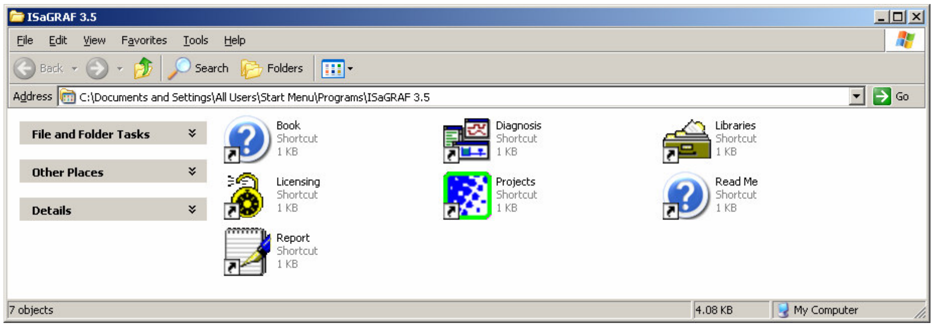
2.2.1 Installing ISaGRAF Workbench V3.5

Use the following procedure to install the EMBEDDED PLC - ISaGRAF Workbench V3.50 from the CD-ROM:

1. Insert the CD-ROM and open the **`ISaGRAF_V3.5`** folder. Alternatively, if you downloaded the zip file from the OEM Technology Solutions website, uncompress in a temporary folder.
2. Double-click on **`ISAGRAF.EXE`** to launch the ISaGRAF installation wizard. Follow the instructions provided by the wizard to proceed with the installation.

> ✍ **Note:**
>
> It is recommended that the ISaGRAF Workbench be installed on a new directory to avoid confusing files with files from other ISaGRAF versions.

3. Once all the ISaGRAF files have been copied, the ISaGRAF Projects Manager icon is added to your desktop and the following group is added to your main Programs group:



2.2.2 Installing EMBEDDED PLC Board Support Package (BSP) for pro•teso PC1000

Use the following procedure to install EMBEDDED PLC BSP for pro•teso PC1000 from the CD-ROM:

4. Go to **`OEM-PC1000_BSP_V1.xx_9-0004-xxx`** folder located in the root directory of the CD-ROM. Alternatively if you downloaded the zip file from the OEM Technology Solutions website, uncompress in a temporary folder.
5. Open the **`ISaGRAF`** folder and double-click on **`INSTALL.EXE`** to launch the ISaGRAF installation wizard. Enter the location where the ISaGRAF Workbench was installed (default **`C:\ISAWIN`**) and press **Install**. The installation wizard will update the ISaGRAF main directory with the I/O boards, C functions, C function blocks and sample applications for the EMBEDDED PLC pro•teso PC1000 target.

2.3 LICENSING THE ISAGRAF WORKBENCH

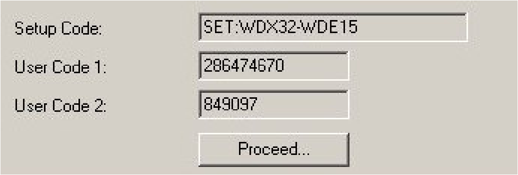
The ISaGRAF V3.5 Workbench you just installed runs for a trial period of 30-days before requiring licensing. When not licensed, you can not create, modify or export IEC programs to a library, or export variables to ASCII text file.

You can license ISaGRAF using either hardware or software license. When using a hardware license, a dongle which you place on the parallel or USB port of your PC is delivered pre-programmed with the selected feature set. When using software license, you need to obtain a set of registration keys.

The software license is specific to a PC (hardware configuration and Operating System). Therefore if you re-install the Operating System and then re-install ISaGRAF Workbench a new software license will be required. However, it is possible to transfer a software license from one PC to another. Section A.1.2.2 of the ISaGRAF User's Manual (See [1]) and the License Manager on-line help describe in detail how to transfer a software license.

To purchase and obtain software license follow these instructions:

1. Open the ISaGRAF License Manager (**Start** menu → **Programs** → **ISaGRAF 3.5** → **Licensing**). On the Available Components select **ISaGRAF 3 Software License** and press **Add** button. Click Option **32** (or more depending on the number of I/O points license to purchase) on **Number of IO Variables** box and press **OK** button. The Setup Code **SET: WDX*n*-WDE15** will be displayed in the **Setup Code** box.

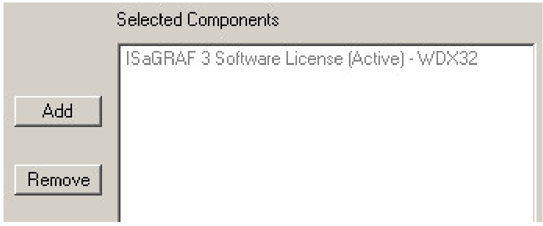

Setup Code: SET:WDX32-WDE15
User Code 1: 286474670
User Code 2: 849097

2. If you have access to the Internet and have an e-mail account:
    a) Press **Proceed…** button. A new mail message of your default e-mail client (such as Microsoft Outlook) will be displayed. Fill out the information on the message body: ***Contact details*** and ***Payment information***. Once you have filled out all the information, send the e-mail to support@oem.net.au. The e-mail client application must be running before pressing **Proceed…** button.
    b) Alternatively, if instead of e-mail client (such as Microsoft Outlook) you use web-based email or a command line e-mail (such as pine), compose manually a new message indicating the ***Payment information***, ***User Code 1 and 2***, and ***Contact details***, and send it to support@oem.net.au.
3. Alternatively, if you do not have an e-mail address or access to the Internet, send your ***Contact details***, ***Payment information***, ***User Code 1*** and ***User Code 2*** to OEM Technology Solutions by fax (Fax Number: +61 2 9966 9429).
4. OEM Technology Solutions will send you the ISaGRAF software license within two business days after receiving the payment. Press **Close** button to close the License Manager.

2.3.1 Adding the software license

The ISaGRAF software license consists of two sets of 4 Registration Keys. Upon reception of the Registration Keys, open License Manager and follow these instructions:

5. On the Available Components select **ISaGRAF 3 Software License** and press **Add** button. Select the purchased number of I/O points to license and press **OK** button.
6. Enter the first set of Registration Keys and press **Register** button. Setup Code will change to **SET: WDE15**. Enter the second set of Registration Keys and press **Register** button again. In the Selected Components **ISaGRAF 3 Software License** will be displayed as **Active**. Press **Close** button to close the License Manager. Congratulations! Your ISaGRAF Workbench is now licensed.


Selected Components
ISaGRAF 3 Software License (Active) - WDX32

✍ **Note:**

If you want to purchase a hardware license or upgrade your current software license, please contact OEM Technology Solutions (sales@oem.net.au) for pricing information and payment options. When you purchase a software license it is very important to include the following information in your Purchase Order:

- Setup Code, User Code 1 and User Code 2.

  To obtain these codes open the ISaGRAF License Manager (**Start** menu → **Programs** → **ISaGRAF 3.5** → **Licensing**). On the Available Components select **ISaGRAF 3 Software License** and press **Add** button. Select the desired number of I/O points to license (128, 256 or Unlimited) and press **OK** button. The Setup and User codes appear on the **License Manager** window. If you press **Proceed…** button, a new e-mail message window is displayed (if you have an e-mail client running such as Microsoft Outlook). You can fill out the required information and send that e-mail to obtain license pricing and payment information.

- Contact details (Company, Name, Address, Telephone, Fax and e-mail).

2.3.2 Adding a hardware license (optional)

When using a hardware license (dongle) on Windows 2000/XP systems, you must install the Sentinel driver in order for the dongle to be seen.

Open the **`ISaGRAF_V3.5\Sentinel`** folder (located in the root directory of the CD-ROM) and double-click on **`Setup.exe`**. Follow the on-screen instructions to install the driver. After the installation is completed, connect the dongle to a parallel or USB port and open the License Manager (**Start** menu → **Programs** → **ISaGRAF 3.5** → **Licensing**). On the Selected Components list, the **ISaGRAF 3 Hardware license** will be shown active.

2.4 PRO•TESO PC1000 CONNECTIONS

2.4.1 Connect a Power Supply

Connect the 24 VDC power supply to connectors 1 (24Vdc IN) and 2 (GND).

2.4.2 Connect an RS232 Cable

The pro•teso PC1000 comes with the EMBEDDED PLC Firmware Kernel loaded, licensed and configured. The default factory configuration is:

Modbus Slave Number = 1

Use Modbus RTU/RS232

Baud Rate = 19200

Connect a DB9 male to DB9 female serial cable to a COM port of the PC and to RS-232 Port 1 of the pro•teso PC1000 controller.

2.4.3 Connect an Ethernet Cable (optional)

There are two ways of connecting the Target PLC to your PC via Ethernet (see Figure 1):

- Using a crossover Ethernet cable to connect directly the Target PLC to your PC; or,
- If your PC is currently connected to an Ethernet hub, use a straight-through Ethernet cable to establish an Ethernet connection to the pro•teso PC1000 from the hub.

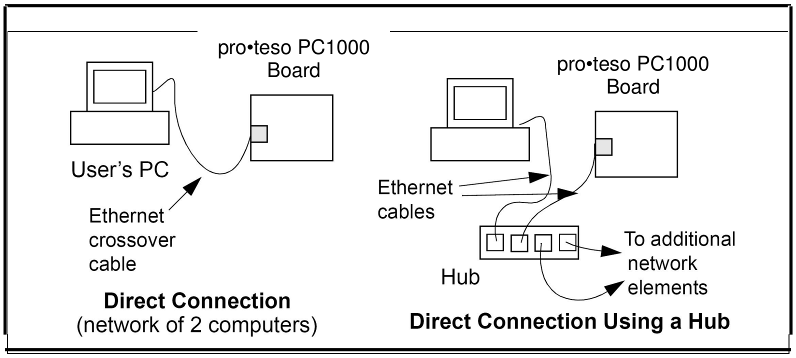
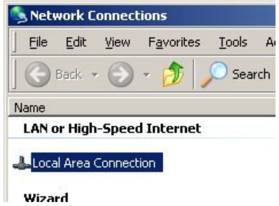


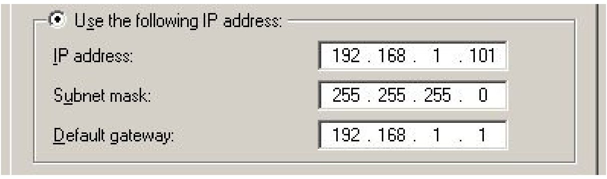
Figure 1: Ethernet connections

When your PC is connected directly to the pro•teso PC1000 via an Ethernet connection, you need to set (or change) the TCP/IP parameters of your computer in order to establish a proper connection with the pro•teso PC1000. If you require changing the TCP/IP configuration of the pro•teso PC1000, go to section 7.1.4 for instructions how to change the configuration of the Target PLC.

To configure your PC with an IP address and netmask suitable for establishing a TCP/IP connection with a pre-configured pro•teso PC1000 Target PLC, follow these instructions (you might need Administrator privileges to perform these steps):



1. Go to Control Panel (**Start** button → **Settings** → **Control Panel**) and start **Network Connections**.
2. Select **Local Area Connection** and choose **Properties** (**File** menu → **Properties**).
3. Select **Internet Protocol (TCP/IP)** and press **Properties** button.
4. Click on **Use the following IP Address** and fill in the following fields:



● Use the following IP address:
IP address: 192 . 168 . 1 . 101
Subnet mask: 255 . 255 . 255 . 0
Default gateway: 192 . 168 . 1 . 1

5. Press **OK** button to close the **Internet Protocol (TCP/IP) Properties** window. Press **OK** button to close the **Local Area Connection Properties** and to update the new TCP/IP values.
6. To verify the communication between your PC and the pro•teso PC1000 use the `ping` command. Open a **Command Prompt** window (**Start** button → **Programs** → **Accessories** → **Command Prompt**) and type

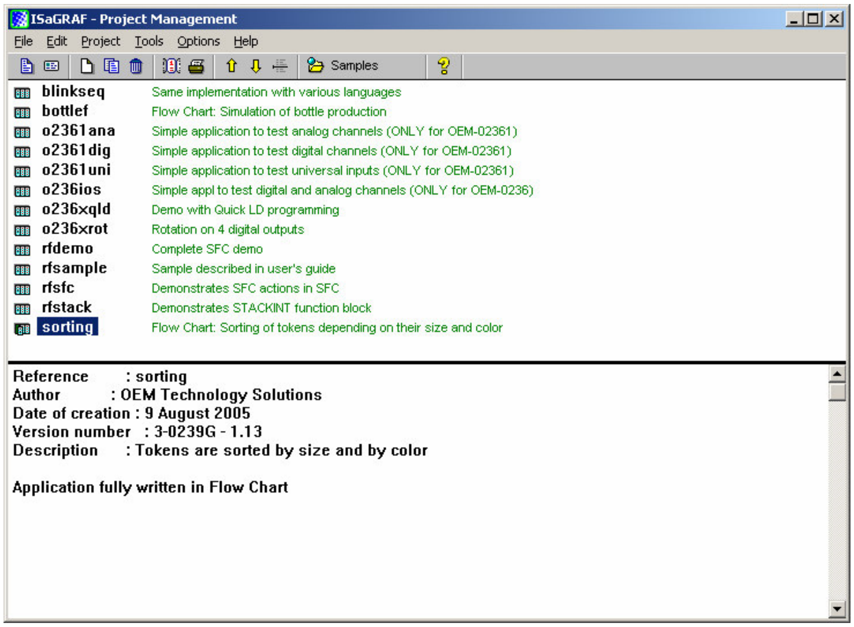
```
C:\ping 192.168.1.100
```

The command output should display the replies from the Target PLC.

2.5 RUNNING A SAMPLE APPLICATION

Once you have established connection between your PC and the Target PLC, follow these instructions to run the ***sorting*** sample application. This sample application displays a SpotLight which shows different tokens being sorted depending on size and colour (see the Project Description at the bottom of the Projects Management window). This PLC application was written in Flow Chart.

1. Start ISaGRAF Workbench by double-clicking on the ISaGRAF 3.5 shortcut on your desktop or **Start** button → **Programs** → **ISaGRAF 3.5** → **Projects**. On the **Projects Management** window, double-click on the ***sorting*** project.

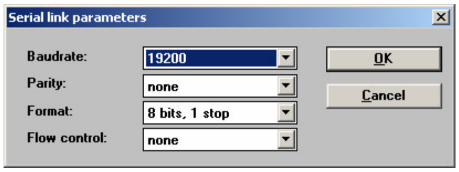

ISaGRAF - Project Management
File Edit Project Tools Options Help
Samples

blinkseq	Same implementation with various languages
bottlef	Flow Chart: Simulation of bottle production
o2361ana	Simple application to test analog channels (ONLY for OEM-02361)
o2361dig	Simple application to test digital channels (ONLY for OEM-02361)
o2361uni	Simple application to test universal inputs (ONLY for OEM-02361)
o236ios	Simple appl to test digital and analog channels (ONLY for OEM-0236)
o236xqld	Demo with Quick LD programming
o236xrot	Rotation on 4 digital outputs
rfdemo	Complete SFC demo
rfsample	Sample described in user's guide
rfsfc	Demonstrates SFC actions in SFC
rfstack	Demonstrates STACKINT function block
sorting	Flow Chart: Sorting of tokens depending on their size and color

Reference : sorting
Author : OEM Technology Solutions
Date of creation : 9 August 2005
Version number : 3-0239G - 1.13
Description : Tokens are sorted by size and by color

Application fully written in Flow Chart

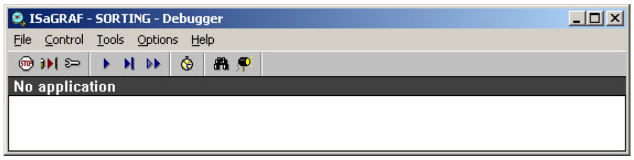
2. On the **SORTING - Programs** window select **Debug** menu → **Link setup**. On the **PC-PLC link parameters** window check that the **Target Slave Number** is set to 1 and COM port used by the PC is selected. Press **Setup** button and check the baud rate 19200 is selected. Press **OK** button to close this window. Press **OK** button to close the PC-PLC link parameters window.



Serial link parameters

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

3. Select **Debug** menu → **Debug** to open the **Debugger** window. The **Debugger** window is displayed with a ***No application*** message.

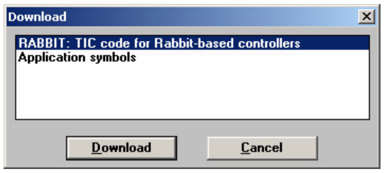


ISaGRAF - SORTING - Debugger

No application

✍ **Note:**

If the **Debugger** window shows that an application is currently running ('<application name>' active) on the Target PLC, press **Stop** button STOP on the toolbar to stop the application before downloading a new one. You can also stop the application from the **Debugger** window menu (**File** menu → **Stop application**).

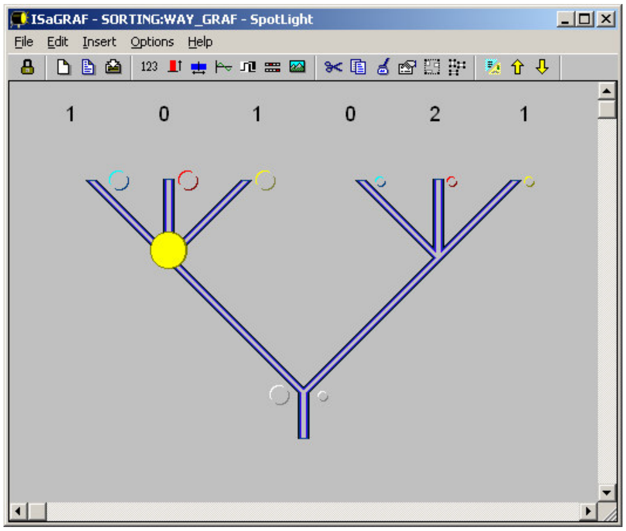


Download

RABBIT: TIC code for Rabbit-based controllers
Application symbols

4. On **Debugger** window, select **File** menu → **Download**. Select "RABBIT: TIC code for Rabbit-based controllers" on the Download window and press **Download** button.

5. A progress bar on the **Debugger** window will show the progress of the downloading process. Once the downloading is completed, the application runs immediately and the **SpotLight** window shown below is displayed.



ISaGRAF - SORTING:WAY_GRAF - SpotLight

Troubled? If you experienced problems running this application, such as an error message "Cannot install communication" or "Cannot start download – target not ready", follow this checklist:

1. Make sure you are using a correct serial cable (not NULL modem) between the PC and the controller.
2. If connection is alive, repeat accurately the step-by-step instructions in this Section. Remember to put correctly the Target Slave Number and the RS-232 parameters in the **PC-PLC link parameters** window (Step 2).
3. If the Target PLC has already an application running, stop it before downloading a new one.

If you still can not run this sample application, go to Chapter 8 (FAQ and Troubleshooting) which contains more information about how to troubleshoot your particular problem.

2.6 WHERE DO I GO FROM HERE?

The next chapter describes how to run the other sample applications and step-by-step instructions on how to create a simple PLC application.

The ISaGRAF User's Manual and the ISaGRAF Workbench on-line help contain a complete description of the Workbench features and capability, as well as a complete PLC programming languages reference. The User's Manual can be found in the Documentation folder of the CD-ROM (**`ISaGRAF.pdf`**).

This page is intentionally left blank

3. RUNNING SAMPLE APPLICATIONS

The ISaGRAF Workbench provides two modes of running PLC applications:

- Simulation mode. The PLC application runs entirely on the PC without the need to have a Target PLC connected. This mode is very useful to perform complete structural and functional tests prior to deploying the application to the target hardware.
- Real mode. The PLC application is first downloaded to a Target PLC and then the Target PLC executes the application. The Workbench monitors the program execution as well as the status of internal and I/O variables.

The EMBEDDED PLC – ISaGRAF V3.5 Workbench plus the EMBEDDED PLC OEM-PC1000 BSP, include several sample applications. There are three types of sample applications:

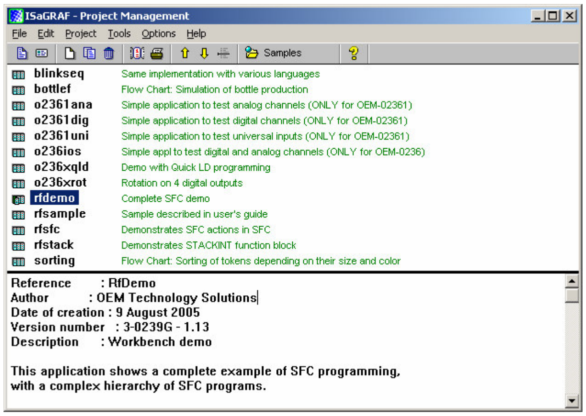
- Sample applications that use Virtual Boards (xai8, xao8, xbi8, xbo8, xmi8, xmo8). These applications can run ONLY in simulation mode:
  - ➢ ***blinkseq***: Same implementation with various languages.
  - ➢ ***rfdemo***: Complete SFC demo.
  - ➢ ***rfsample***: Sample described in ISaGRAF User's Guide (see [1]).
  - ➢ ***rfsfc***: Demonstrates SFC actions in SFC.
- Sample applications that use internal variables only. These applications can be simulated or downloaded to the Target PLC. The Target PLC does not need to be connected to physical I/Os:
  - ➢ ***bottlef***: Flow Chart: Simulation of bottle production.
  - ➢ ***rfstack***: Demonstrates STACKINT function block.
  - ➢ ***sorting***: Flow Chart: Sorting of tokens depending on their size and colour.
- Sample applications pre-configured to use pro•teso PC1000 I/O boards. These applications can be simulated or downloaded to the pro•teso PC1000 Target PLC and requires that the Target is connected to physical I/Os:
  - ➢ ***pc10qld***: Demo with Quick LD programming.
  - ➢ ***pc10rota***: Rotation on 4 digital outputs.
  - ➢ ***pc10cro***: Signal generation (pro•teso PC1000 Analogue Out O01) and SpotLight.
  - ➢ ***pc10ios***: Simple application which connects several inputs with output channels.

A description of each of the sample applications appears at the bottom of the ISaGRAF Project Manager.

3.1 RUNNING APPLICATIONS IN SIMULATION MODE

Use the following procedure to run any sample application in simulation mode:

4. Open the ISaGRAF Project Management by double-clicking on the ISaGRAF Projects icon on your Desktop or selecting **Start** menu → **Programs** → **ISaGRAF 3.5** → **Projects**.

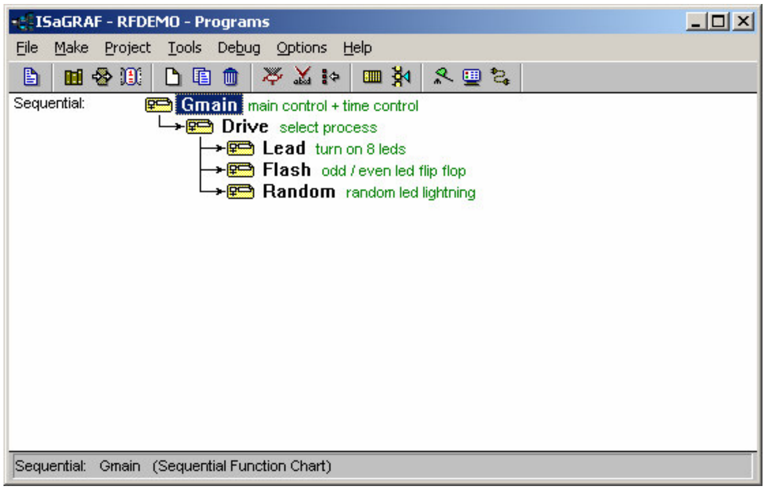

ISaGRAF - Project Management
File Edit Project Tools Options Help
Samples

blinkseq	Same implementation with various languages
bottlef	Flow Chart: Simulation of bottle production
o2361ana	Simple application to test analog channels (ONLY for OEM-02361)
o2361dig	Simple application to test digital channels (ONLY for OEM-02361)
o2361uni	Simple application to test universal inputs (ONLY for OEM-02361)
o236ios	Simple appl to test digital and analog channels (ONLY for OEM-0236)
o236xqld	Demo with Quick LD programming
o236xrot	Rotation on 4 digital outputs
rfdemo	Complete SFC demo
rfsample	Sample described in user's guide
rfsfc	Demonstrates SFC actions in SFC
rfstack	Demonstrates STACKINT function block
sorting	Flow Chart: Sorting of tokens depending on their size and color

Reference : RfDemo
Author : OEM Technology Solutions
Date of creation : 9 August 2005
Version number : 3-0239G - 1.13
Description : Workbench demo

This application shows a complete example of SFC programming, with a complex hierarchy of SFC programs.

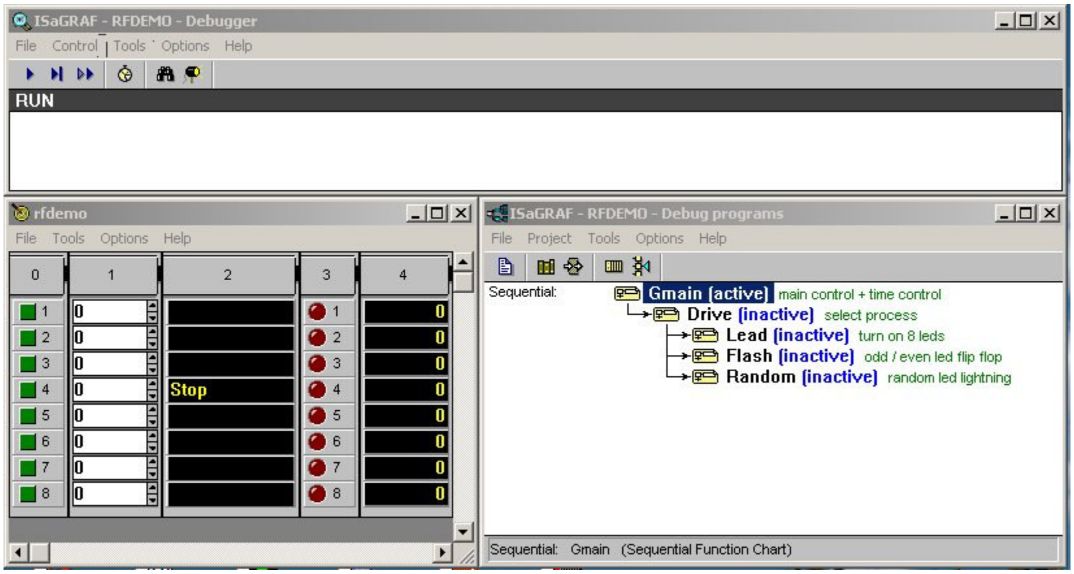
5. Double-click on the sample application (for example, ***rfdemo***) to open the **Programs Manager** window.


ISaGRAF - RFDEMO - Programs
File Make Project Tools Debug Options Help

Sequential:
Gmain main control + time control
Drive select process
Lead turn on 8 leds
Flash odd / even led flip flop
Random random led lightning

Sequential: Gmain (Sequential Function Chart)

6. On the **Programs Manager** window, select **Debug** → **Simulate** (or press **Simulate** button on the toolbar). The **Debugger**, **I/O simulation** and **Debug program** windows are displayed.


ISaGRAF - RFDEMO - Debugger
File Control Tools Options Help
RUN
rfdemo
File Tools Options Help
ISaGRAF - RFDEMO - Debug programs
File Project Tools Options Help

0	1	2	3	4
🟩 1	0		● 1	0
🟩 2	0		● 2	0
🟩 3	0		● 3	0
🟩 4	0	Stop	● 4	0
🟩 5	0		● 5	0
🟩 6	0		● 6	0
🟩 7	0		● 7	0
🟩 8	0		● 8	0

Sequential:
Gmain (active) main control + time control
Drive (inactive) select process
Lead (inactive) turn on 8 leds
Flash (inactive) odd / even led flip flop
Random (inactive) random led lightning

Sequential: Gmain (Sequential Function Chart)

7. The **I/O simulation** window shows the I/O boards used by the application (Boards 0 to 4). Board 0 (column 0) has 8 digital inputs; Board 1 (column 1) has 8 analogue inputs; Board 2 (column 2) has 8 message (strings) outputs; Board 3 (column 3) has 8 digital outputs and Board 4 (column 4) has 8 analogue outputs. On Board 0 (column 0) click on Input 1 and Input 4, a message **start** and **Lead** are displayed in message outputs 4 and 5 respectively; and Board 3 will show the LED outputs being turned ON in sequence. On the **Debug programs** window the **Drive** and **Lead** programs change to active state. Unset the digital input 4 and set any of the digital inputs 5 (Flash) or 6 (Random).

8. To close the simulation of the application, close the **Debugger** window. Close the **Programs** window.

9. Try with other sample applications such as the one listed in the introduction of this chapter. All the sample applications can run in simulation mode.

3.2 RUNNING APPLICATIONS THAT USE INTERNAL VARIABLES ONLY

The Target PLC and your PC must be connected through Ethernet, RS232 or RS485. Use the following procedure to run any sample application that uses only internal variables:

1. Open the ISaGRAF Project Management by double-clicking on the ISaGRAF Projects icon on your Desktop or selecting **Start** menu → **Programs** → **ISaGRAF 3.5** → **Projects**. On the **Projects Management** window, double-click on the ***bottlef*** project.

2. On the **BOTTLEF - Programs** window select **Debug** menu → **Link setup**. On the **PC-PLC link parameters** window enter the **Target Slave Number** (corresponds to the Modbus Slave number of the Target PLC) and the Communication port (**COM** for Modbus RTU over RS232/RS485 or **ETHERNET** for Modbus TCP):

    a) If serial **COM** port is selected (COM1 – COM4) press **Setup** button and select either 19200, 9600, 4800, 2400, 1200 or 600 baud rate depending on the current configuration of your EMBEDDED PLC target.

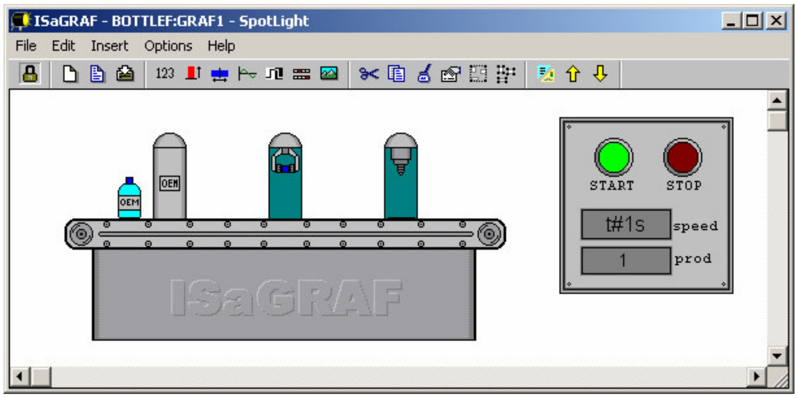
    > ✍ **Note:**
    >
    > The EMBEDDED PLC Kernel supports baud rates from 19200 to 600. The other parameters are fixed: no parity, 8 data bits, 1 stop bit and no flow control.
    >
    > Note also that ISaGRAF supports serial ports COM1 to COM4. If you are using a USB-to-serial converter cable check that the com port assigned is between COM1 and COM4. If not, change the serial port settings on your Windows Operating System (see Section 8.2). Many USB-to-serial converters are assigned to COM5 port or higher after its installation.

    b) If **ETHERNET** is selected press **Setup** button and enter the IP address of the target (default is 192.168.1.100). Press **OK** button. Press **OK** button to close the **PC-PLC link parameters** window

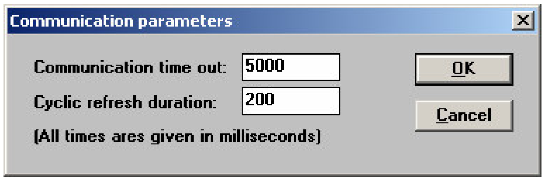
    > ✋ **Caution:**
    >
    > Do not change the port number from 1100 otherwise the Ethernet communication with the ISaGRAF Workbench will not work. This port is used exclusively by the communication between ISaGRAF Workbench and the EMBEDDED PLC target.

3. Select **Debug** menu → **Debug** to open the **Debugger** window. If **Debugger** window displays "<application name> active" message, stop the current application that is running by pressing the Stop button STOP on the toolbar or select **File** menu → **Stop application**. A "No application" message is displayed.

4. Download the bottlef application by selecting **File** menu → **Download**. Select "RABBIT: TIC code for Rabbit-based controllers" on the **Download** window and press **Download** button. A progress bar on the **Debugger** window will show the progress of the downloading process. Once the downloading is completed the application runs immediately and the **SpotLight** window is displayed. In the **SpotLight**, double-click on the **START** button and press **TRUE** button. The bottle production will start.



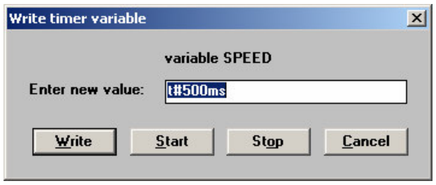
5. On the **Debugger** window, select **Options** menu → **Parameters**. Change the **Cyclic refresh duration** value to 200. This value corresponds to the refresh rate of the SpotLight. Press OK button.


Communication parameters

Communication time out: 5000

Cyclic refresh duration: 200

(All times ares given in milliseconds)


Write timer variable

variable SPEED

Enter new value: t#500ms

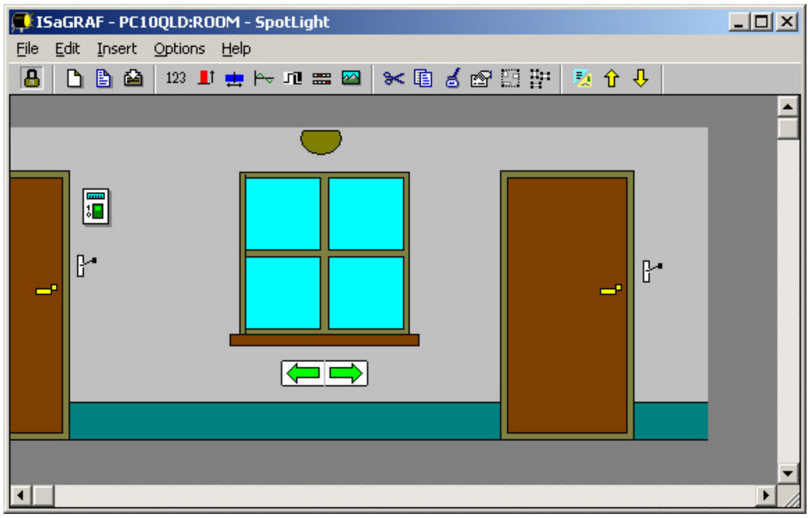
6. On the **SpotLight** window, double-click on the **t#1s** box. Change the variable SPEED value to **t#200ms** and press **Write** button. The bottle production will speed up by a factor of two.

Repeat this procedure for the ***rfstack*** application.

3.3 RUNNING APPLICATIONS THAT USE PRO•TESO PC1000 INPUTS AND OUTPUTS

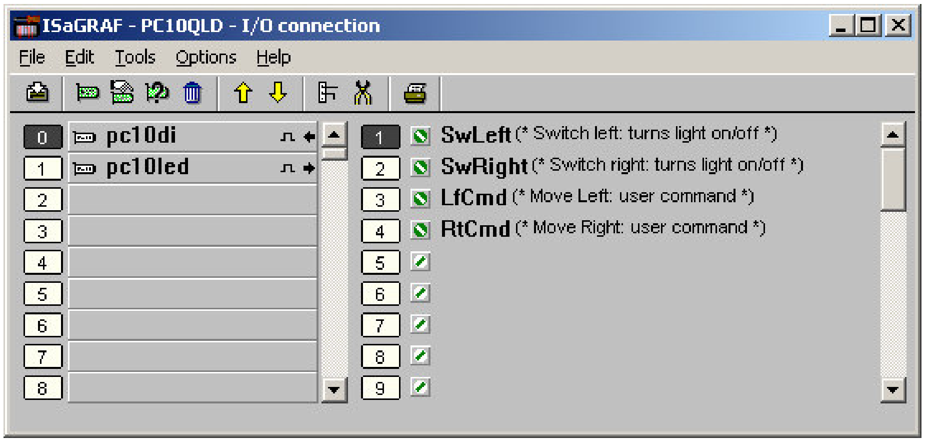
Use the following procedure to download a sample application (***pc10qld***) to the pro•teso PC1000 Target that uses physical I/Os. The Target PLC must be running a licensed EMBEDDED PLC Firmware Kernel and be connected to 4 digital inputs and 1 digital output. It also assumes that the Target is connected to the PC either via Ethernet or serial port.

1. Open the ISaGRAF Project Management by double-clicking on the ISaGRAF Projects icon on your Desktop or selecting **Start** menu → **Programs** → **ISaGRAF 3.5** → **Projects**. On the **Projects Management** window, double-click on the pc10qld project.
2. Select **Debug** → **Link setup** (or press **Link Setup**) and configure the Target PLC ↔ PC link. Follow the guidelines on Step 2 of the previous section.
3. Select **Debug** → **Debug** (or press **Debug**) to open the **Debugger** window. If an application is running, stop it (press **Stop** button on the toolbar). On the **Debugger** window select **File** → **Download** to download the pc10qld application to the target. A progress bar on the **Debugger** window shows the progress of the download process. Once the downloading is completed the application starts running and the following **SpotLight** window is displayed.



4. The Digital inputs DI01 and DI02 of the pro•teso PC1000 are wired as the Left and Right switches on the pc10qld application. The LED3 green is the output of the light. The pro•teso PC1000 digital inputs DI03 and DI04 move the window's screen to the left and right respectively.

5. To check the I/O connections, close the **Debugger** window and on the **PC10QLD – Programs** window select **Project** → **I/O connection**. Check the wiring of the channels.



0 pc10di
1 pc10led
2
3
4
5
6
7
8

1 SwLeft (* Switch left: turns light on/off *)
2 SwRight (* Switch right: turns light on/off *)
3 LfCmd (* Move Left: user command *)
4 RtCmd (* Move Right: user command *)
5
6
7
8
9

Follow the same procedure to run the pc10rota application. This application will turn ON 2 dual colour LEDs in sequence.

3.4 CREATING A NEW PLC APPLICATION

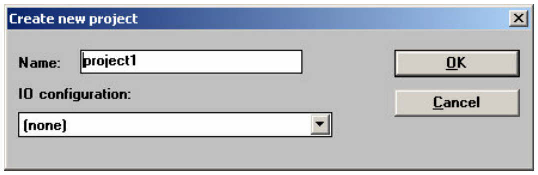
The following Section describes how to use the ISaGRAF Workbench to create simple PLC applications using the five IEC 611131-3 PLC programming languages (FBD, LD, ST, IL and SFC) and Flow Chart that later you will download to the pro•teso PC1000 Target PLC. The Target PLC must be connected to physical I/Os. It also assumes that the Target is connected to the PC either via Ethernet or serial port. ***To create (or modify) PLC applications the ISaGRAF Workbench must be licensed***.

The PLC application that is written using 6 different languages, implements a simple Boolean operation (XOR) between digital input DI01 and DI02 (SW1 and SW2 Boolean variables respectively). The digital output DO01 (LED1 Boolean output variable) is used as the output.

3.4.1 Using Function Blocks Diagrams (FBD)

1. Open the ISaGRAF Project Management by double-clicking on the ISaGRAF Projects icon on your Desktop or selecting **Start** menu → **Programs** → **ISaGRAF 3.5** → **Projects**. On the **Projects Management** window, select **File** menu → **New**.

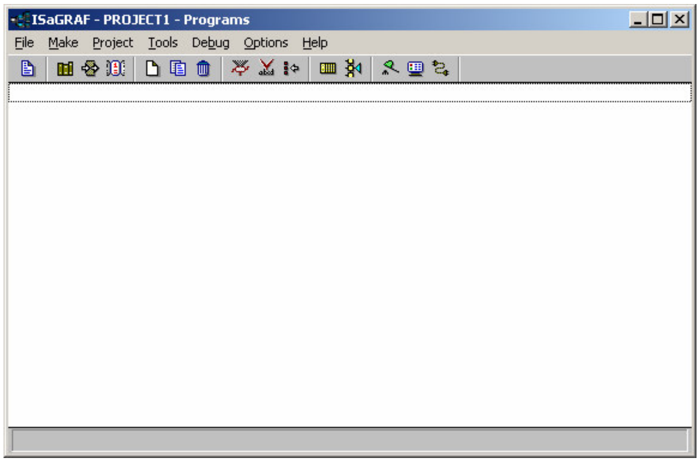
2. On the **Create new project** window, enter the name of the project (for example project1) and press **OK** button. The name of project should no exceed 8 characters. The **project1** project is then listed in the **Projects Management** window.

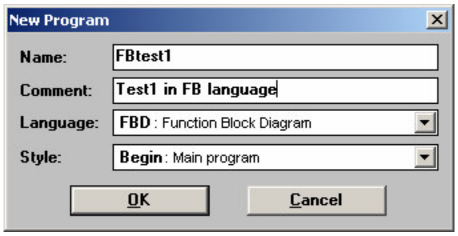


Name: project1
IO configuration: (none)

3. Select **Edit** menu → **Set comment text**. Enter a brief description of this project and press **OK** button. This description will be displayed next to the project name in the **Projects Management** window.

4. Double-click on **project1** project. An empty **PROJECT1 – Programs** window is displayed.

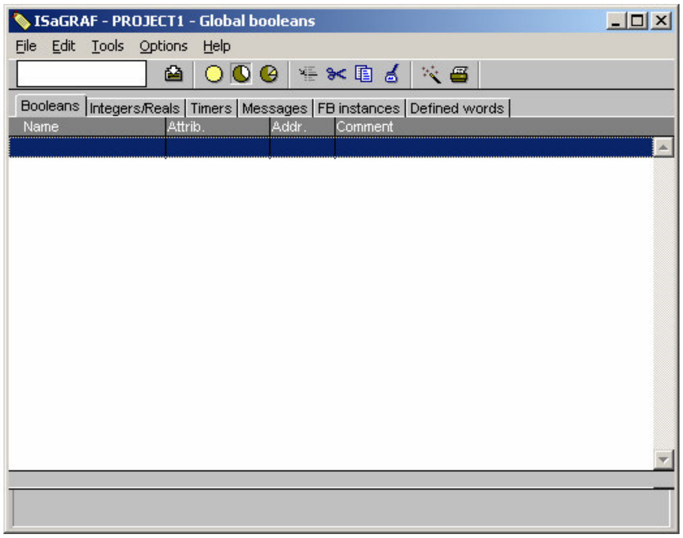

ISaGRAF - PROJECT1 - Programs
File Make Project Tools Debug Options Help



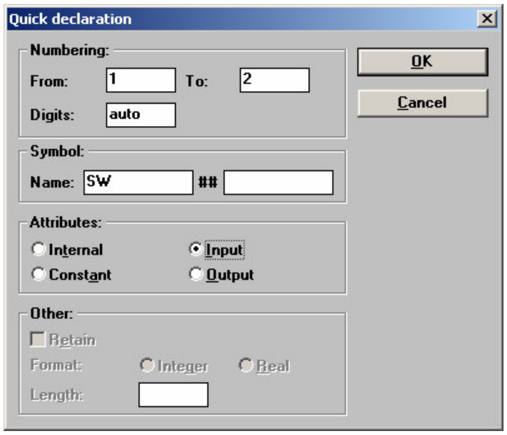
Name: FBtest1
Comment: Test1 in FB language
Language: FBD : Function Block Diagram
Style: Begin : Main program

5. Select **File** menu → **New**. On the **New program** window enter the name of the program (for example, FBtest1); a brief comment; select the Language to **FBD**; select **Begin** Style and press **OK** button. The FBtest1 program is displayed in the **PROJECT1 – Programs** window.

6. On the **PROJECT1 – Programs** window select **File** menu → **Dictionary** to open the **Dictionary** window.



7. Select **Tools** menu → **Quick declaration**. Enter Numbering from 1 to 2, Symbol name SW and Attributes = Input as shown in figure below. Press **OK** button.



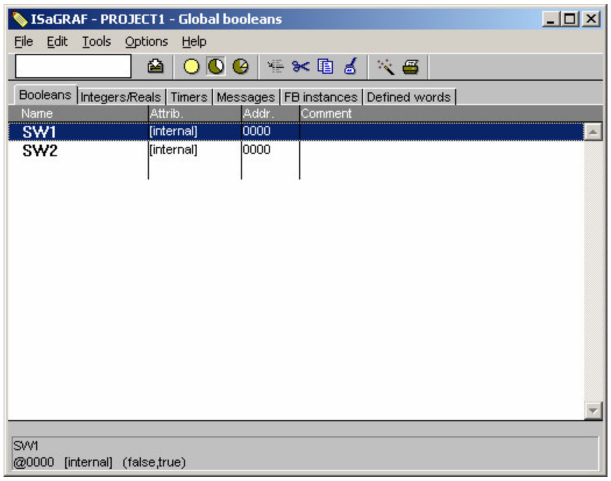
Numbering:
From: 1 To: 2
Digits: auto

Symbol:
Name: SW ##

Attributes:
○ Internal ● Input
○ Constant ○ Output

Other:
☐ Retain
Format: ○ Integer ○ Real
Length:

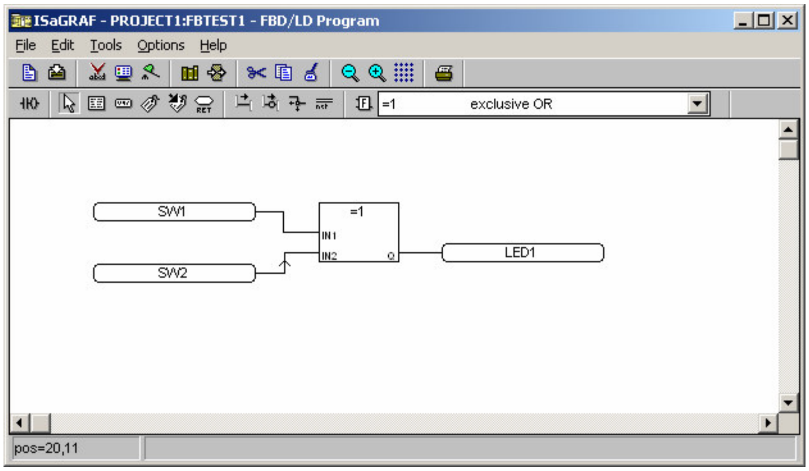
8. On the **Dictionary** window, the SW1 to SW2 variables are listed.


ISaGRAF - PROJECT1 - Global booleans
File Edit Tools Options Help
Booleans | Integers/Reals | Timers | Messages | FB instances | Defined words

Name	Attrib.	Addr.	Comment
SW1	[internal]	0000	
SW2	[internal]	0000	

SW1
@0000 [internal] (false,true)

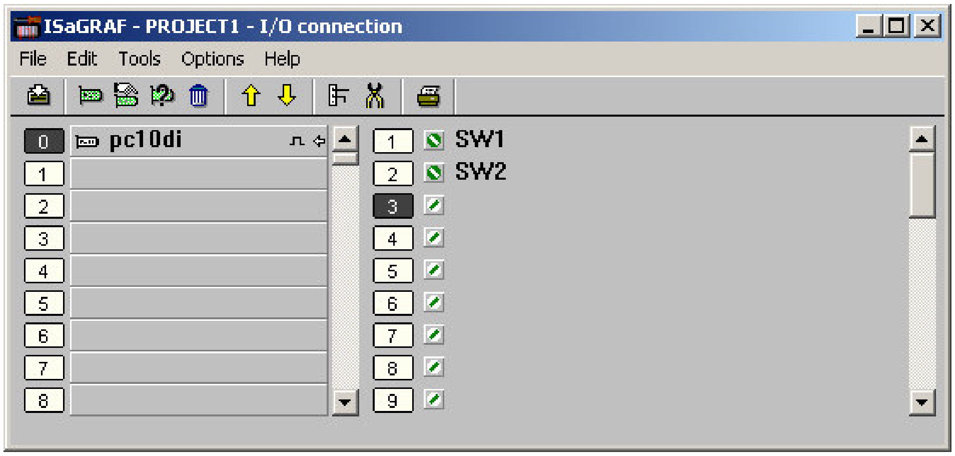
9. On the PROJECT1 Dictionary double-click on the empty row. Enter the name LED1 and select it as output.
10. Save the current dictionary by selecting **File** menu → **Save**. Close the **Dictionary** window.
11. On the **PROJECT1 – Programs** window double-click on **FBtest1** program. Press **Insert variable** button on the toolbar and click on the working area. On the Select variable window select SW1 and press **OK** button. SW1 variable box is drawn on the working area. Place SW2 and LED1 variables following the same procedure.
12. Select on the **Function blocks** combo box =1 exclusive OR the exclusive OR (=1) function block and click on the working area. The exclusive OR (=1) function block is displayed.
13. Select button on the toolbar and move SW1 and SW2 variables to the left of the function block; also move the LED1 variable to the right of the function block.
14. Click on **Draw connection line** button on the toolbar and press left mouse button on the SW1 variable and drag the line towards the function block. A connection line will be drawn between SW1 and IN1 of the function block. Repeat the same procedure to connect SW2 with IN2 of the function block and LED1 with Q of the function block (connection lines are dragged always from left to right).


ISaGRAF - PROJECT1:FBTEST1 - FBD/LD Program
File Edit Tools Options Help
=1
exclusive OR
SW1
SW2
=1
IN1
IN2
Q
LED1
pos=20,11

15. Save the program by selecting **File** menu → **Save**. Press **OK** button on the **Update diary** window. Close the **FBtest1 – FBD/LD Program** window.

16. On the **PROJECT1 – Programs** window click on I/O connection button on the toolbar.

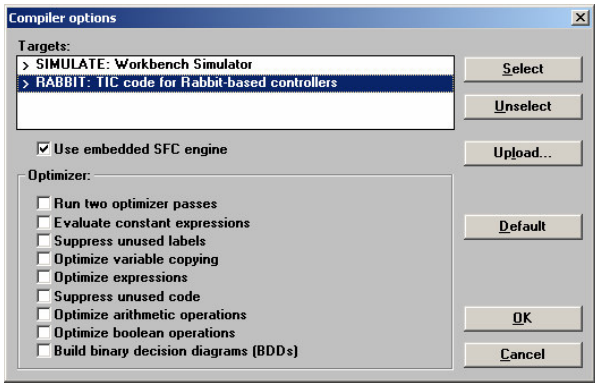
17. Double-click on Slot **0** or press **Set board/equipment** button on the toolbar. Select **pc10di** board (pro•teso PC1000 Digital Inputs) and press **OK** button. The pc10di I/O board is placed in Slot 0. Double-click on Channel **1**. On the **Connect I/O channel** window press **Connect** button several times until all the free variables are connected. Press **Close** button.


ISaGRAF - PROJECT1 - I/O connection
File Edit Tools Options Help
0 pc10di
1
2
3
4
5
6
7
8
1 SW1
2 SW2
3
4
5
6
7
8
9

18. Double-click on Slot **1** and select **pc10do** (pro•teso PC1000 Digital Outputs) board. Double-click on Channel **1** and connect the free variable LED1. Press **Close** button.

19. Before downloading the PLC application to the Target PLC, the application needs to be compiled. On the **PROJECT1 – Programs** window, select **Make** menu → **Compiler Options** to check that **RABBIT: TIC code for Rabbit-based controllers** is selected. Press **OK** button.

    If RABBIT target is not selected, click on RABBIT target and press **Select** button. Alternatively you can double-click on RABBIT target.

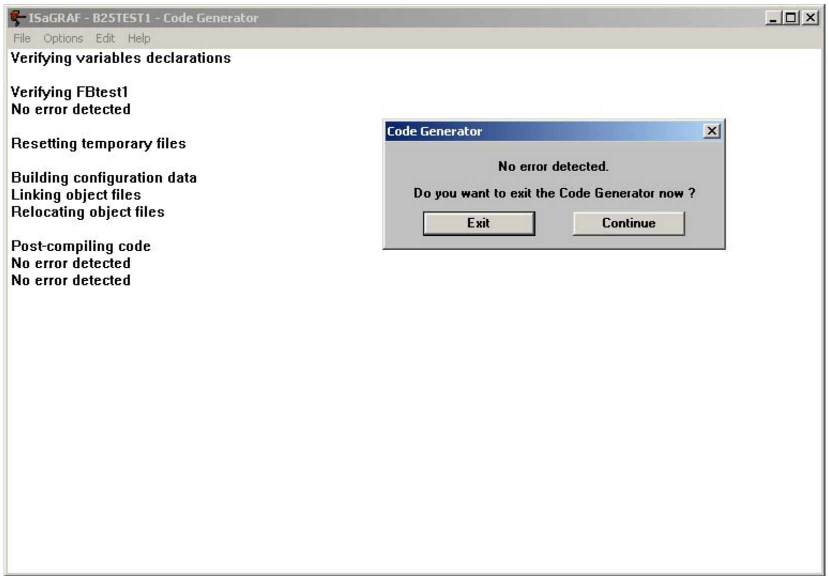


Targets:
> SIMULATE: Workbench Simulator
> RABBIT: TIC code for Rabbit-based controllers

☑ Use embedded SFC engine

Optimizer:
☐ Run two optimizer passes
☐ Evaluate constant expressions
☐ Suppress unused labels
☐ Optimize variable copying
☐ Optimize expressions
☐ Suppress unused code
☐ Optimize arithmetic operations
☐ Optimize boolean operations
☐ Build binary decision diagrams (BDDs)

20. Select **Make** menu → **Make application** to compile the PLC application. The **Code generator** window is displayed. Press **Exit** button.



```
Verifying variables declarations

Verifying FBtest1
No error detected

Resetting temporary files

Building configuration data
Linking object files
Relocating object files

Post-compiling code
No error detected
No error detected
```

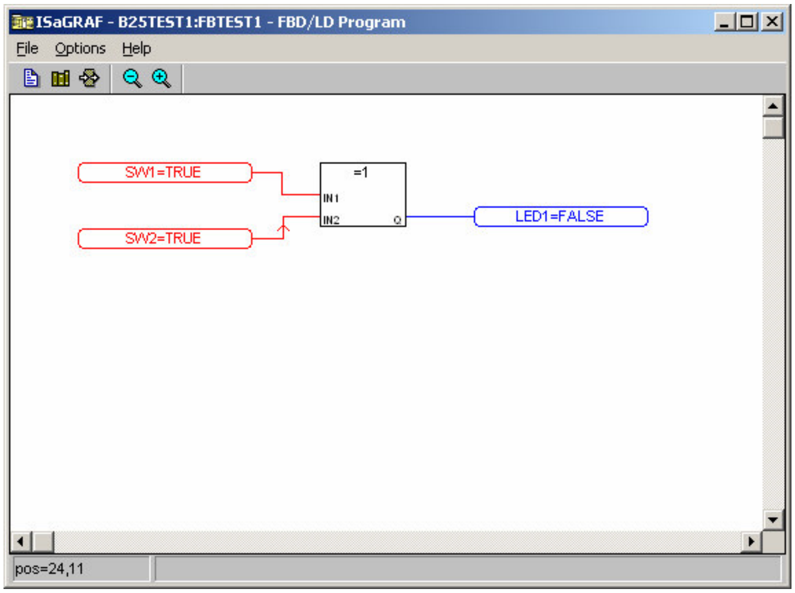
No error detected.

Do you want to exit the Code Generator now ?

21. Select **Debug** → **Link setup** (or press **Link Setup**) and configure the Target PLC ↔ PC link. Follow the guidelines on Step 2 of Section 3.2.

22. Select **Debug** → **Debug** (or press **Debug**) to open the **Debugger** window. If an application is running, stop it (press **Stop** button STOP on the toolbar). On the **Debugger** window select **File** → **Download** to download the PROJECT1 application to the Target PLC. A progress bar on the **Debugger** window shows the progress of the download process. Once the downloading is completed the **PROJECT1 – Debug** programs window is displayed (On the **Debugger** window the application state will be **RUN**).

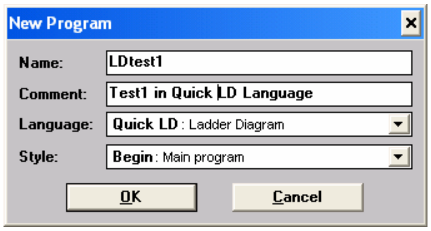
23. Double-click on **FBtest1** program. The **FBTEST1 – FBD/LD Programs** window is displayed with the variables in BLUE or RED colour depending on its current state, FALSE or TRUE respectively. If the digital input DI01 of the pro•teso PC1000 is short circuited, SW1 is FALSE and digital output DO01 will be turned ON as well as the state of LED1 in the Programs window will change. The PLC program implements a Boolean XOR operation on SW1 and SW2 with the result set on the LED1.



24. To stop the monitoring of the PLC application, close the **Debugger** window. The programs and dictionary windows in debug mode will be closed.

3.4.2 Using Quick Ladder Diagrams (LD)

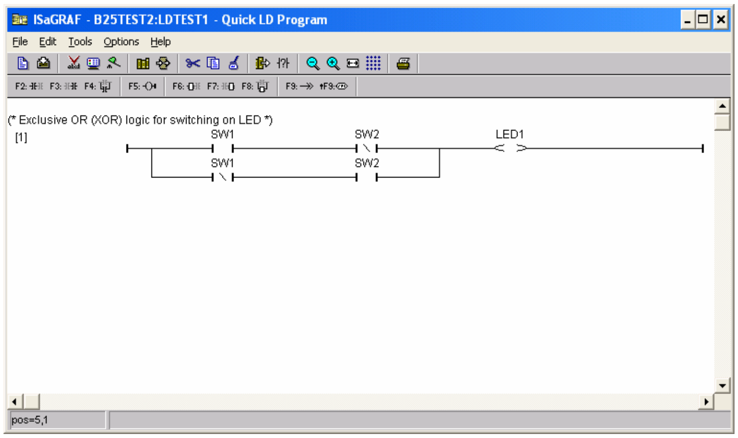
1. Open the ISaGRAF Project Management by double-clicking on the ISaGRAF Projects icon on your Desktop or selecting **Start** menu → **Programs** → **ISaGRAF 3.5** → **Projects**. On the **Projects Management** window, select **File** menu → **New**.
2. On the **Create new project** window, enter the name of the project (for example project2) and press **OK** button. The name of project should not exceed 8 characters. The project2 project is then listed in the **Projects Management** window.
3. Select **Edit** menu → **Set comment text**. Enter a brief description of this project and press **OK** button. This description will be displayed next to the project name in the **Projects Management** window.
4. Double-click on **project2** project. An empty **PROJECT2 – Programs** window is displayed.



Name: LDtest1
Comment: Test1 in Quick LD Language
Language: Quick LD : Ladder Diagram
Style: Begin : Main program

5. Select **File** menu → **New**. On the **New program** window enter the name of the program (for example, LDtest1); a brief comment; select the Language to **Quick LD**; select **Begin** Style and press **OK** button. The LDtest1 program is displayed in the **PROJECT2 – Programs** window.
6. Follow Steps 6 to 9 in Section 3.4.1 to create and populate the application's Dictionary with SW1 and SW2 Boolean input variables (connected to Channel 1 and 2 of the pc10di I/O board respectively); and LED1 Boolean output variable connected to Channel 1 of the pc10do I/O board.
7. On the **PROJECT2 – Programs** window double-click on **LDtest1** program. The Quick Ladder Diagram Editor contains an **editing grid,** a logical matrix where each cell of the matrix may contain up to one LD symbol. Press **F2: Contact on the left** button F2: ⊣⊢ on the toolbar to insert a **new rung**. A new rung with one contact and one coil is created.
8. Double click the **rung comment** (* *) cell to insert a comment such as (*Exclusive OR logic for switching on LED*).
9. Double click the **contact** —| |— on the rung and select SW1 for an input.

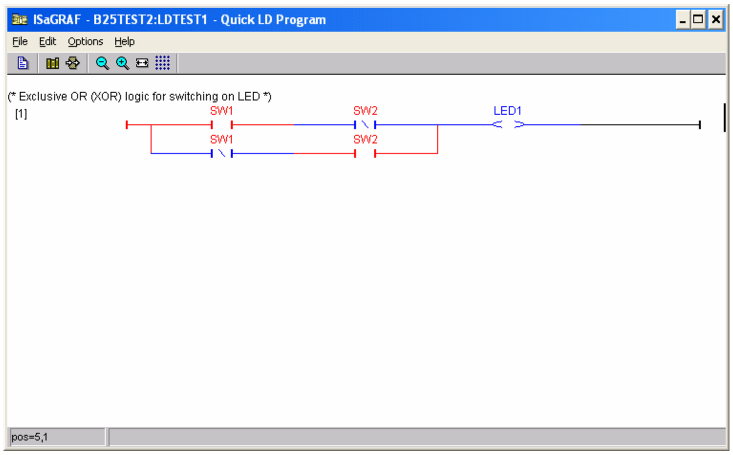
10. With the SW1 contact selected, press **F3: Contact on the right** button F3: on the toolbar.
11. Double click the new contact and select SW2 variable. Click once on the **Coil/contact type** button to negate this contact.
12. Select the **Select** button on the toolbar. Select both contacts on the rung by clicking on one contact and dragging the mouse over to the next contact before releasing the mouse button. Press **F4: Parallel contact** button F4: to create two new contacts in parallel with the selected contacts.
13. Repeat steps 9 and 11 except negate the SW1 contact with the **Coil/contact type** button this time instead of the SW2 contact.
14. Double click on the **coil** —< >— on the rung and select LED1 for an output.
15. The **LDtest1 program** should now appear as follows:


ISaGRAF - B25TEST2:LDTEST1 - Quick LD Program
File Edit Tools Options Help
F2: F3: F4: F5: F6: F7: F8: F9: sF9:
[1]
SW1
SW2
LED1
SW1
SW2

(* Exclusive OR (XOR) logic for switching on LED *)

pos=5,1

16. Save the program by selecting **File** menu → **Save**. Press **OK** button on the **Update diary** window. Close the **LDtest1 – QuickLD Program** window.
17. Follow Steps 19 to 22 in Section 3.4.1 to compile the application and download it to the Target PLC.
18. Double-click on **LDtest1** program. The **LDTEST1 – Quick LD Programs** window is displayed with the variables in BLUE or RED colour depending on its current state, FALSE or TRUE respectively. Press the SW1 (short-circuit digital input DI01) and the LED1 will be turned ON as well as the state of LED1 in the Programs window will change. The PLC program implements a Boolean XOR operation on SW1 and SW2 with the result set on the LED1.



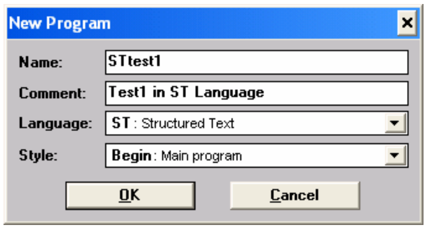
(* Exclusive OR (XOR) logic for switching on LED *)

pos=5,1

19. To stop the monitoring of the PLC application, close the **Debugger** window. The programs and dictionary windows in debug mode will be closed.

3.4.3 Using Structured Text (ST)

1. Open the ISaGRAF Project Management by double-clicking on the ISaGRAF Projects icon on your Desktop or selecting **Start** menu → **Programs** → **ISaGRAF 3.5** → **Projects**. On the **Projects Management** window, select **File** menu → **New**.
2. On the **Create new project** window, enter the name of the project (for example project3) and press **OK** button. The name of project should not exceed 8 characters. The project3 project is then listed in the **Projects Management** window.
3. Select **Edit** menu → **Set comment text**. Enter a brief description of this project and press **OK** button. This description will be displayed next to the project name in the **Projects Management** window.
4. Double-click on **project3** project. An empty **PROJECT3 – Programs** window is displayed.

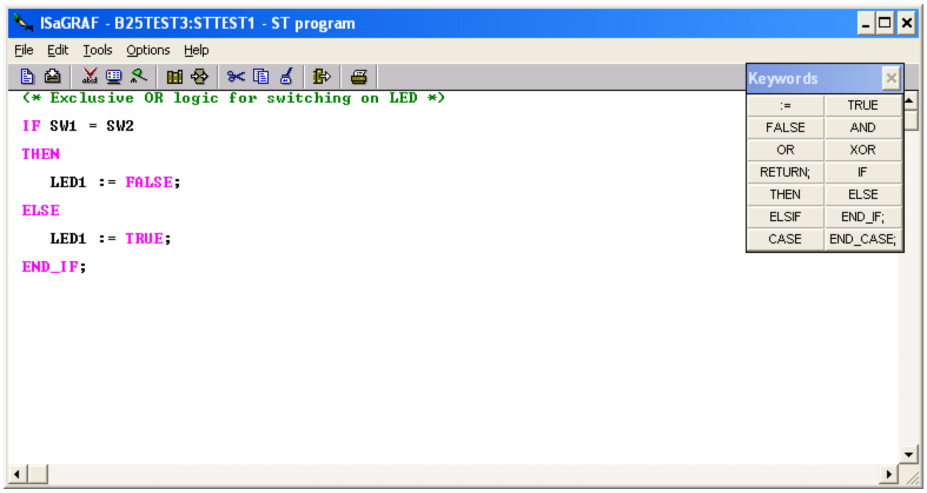


Name: STtest1
Comment: Test1 in ST Language
Language: ST : Structured Text
Style: Begin : Main program

5. Select **File** menu → **New**. On the **New program** window enter the name of the program (for example, STtest1); a brief comment; select the Language to **ST**; select **Begin** Style and press **OK** button. The STtest1 program is displayed in the **PROJECT3 – Programs** window.
6. Follow Steps 6 to 9 in 3.4.1 to create and populate the application's Dictionary with SW1 and SW2 Boolean input variables, and LED1 Boolean output variable.
7. On the **PROJECT3 – Programs** window double-click on **STtest1** program.
8. Type in a comment in **comment parentheses** (* *) such as (* Exclusive OR logic for switching on LED *). When entered correctly comments are highlighted green.
9. Press **Enter** to move to the next command line in the **ST program** editor.
10. To enter **Keyword commands**, either click the desired button in the Keywords tool box or type it, such as for an 'if' statement, either click **IF** in the Keyword tool box or type IF on the command line. When correctly entered Keyword commands are highlighted pink.
11. To enter **inputs, outputs, or variables**, for example input switch SW1, either select Edit → Insert variable → SW1 or type SW1 on the command line.
12. Type the following command lines under the entered comment:

    ```
    IF SW1 = SW2
    THEN
          LED1 := FALSE;
    ELSE
          LED1 := TRUE;
    END_IF;
    ```

13. The **LDtest1 program** should now appear as follows:

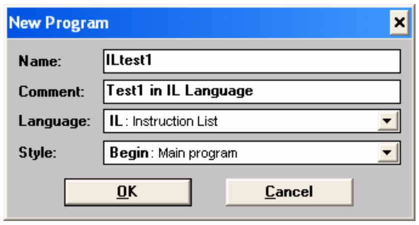


```
(* Exclusive OR logic for switching on LED *)
IF SW1 = SW2
THEN
   LED1 := FALSE;
ELSE
   LED1 := TRUE;
END_IF;
```

14. Save the program by selecting **File** menu → **Save**. Press **OK** button on the **Update diary** window. Close the **STtest1 – ST Program** window.
15. Follow Steps 19 to 22 in 3.4.1 to compile the application and download it to the Target PLC.
16. Double-click on **STtest1** program. The **STTEST1 – ST Programs** window is displayed. Press the SW1 (short-circuit digital input DI01) and the LED1 will be turned ON as well as the state of LED1 in the Programs window will change. The PLC program implements a Boolean XOR operation on SW1 and SW2 with the result set on the LED1.
17. To stop the monitoring of the PLC application, close the **Debugger** window. The programs and dictionary windows in debug mode will be closed.

3.4.4 Using Instruction List (IL)

1. Open the ISaGRAF Project Management by double-clicking on the ISaGRAF Projects icon on your Desktop or selecting **Start** menu → **Programs** → **ISaGRAF 3.5** → **Projects**. On the **Projects Management** window, select **File** menu → **New**.
2. On the **Create new project** window, enter the name of the project (for example project4) and press **OK** button. The name of project should not exceed 8 characters. The project4 project is then listed in the **Projects Management** window.
3. Select **Edit** menu → **Set comment text**. Enter a brief description of this project and press **OK** button. This description will be displayed next to the project name in the **Projects Management** window.
4. Double-click on **project4** project. An empty **PROJECT4 – Programs** window is displayed.



Name: ILtest1
Comment: Test1 in IL Language
Language: IL : Instruction List
Style: Begin : Main program

5. Select **File** menu → **New**. On the **New program** window enter the name of the program (for example, ILtest1); a brief comment; select the Language to **Quick IL**; select **Begin** Style and press **OK** button. The ILtest1 program is displayed in the **PROJECT4 – Programs** window.
6. Follow Steps 6 to 9 in 3.4.1 to create and populate the application's Dictionary with SW1 and SW2 Boolean input variables, and LED1 Boolean output variable.
7. On the **PROJECT4 – Programs** window double-click on **ILtest1** program.
8. Type in a comment in **comment parentheses** (* *) such as (* Exclusive OR logic for switching on LED *). When entered correctly comments are highlighted green.
9. Press **Enter** to move to the next command line in the **IL program** editor.

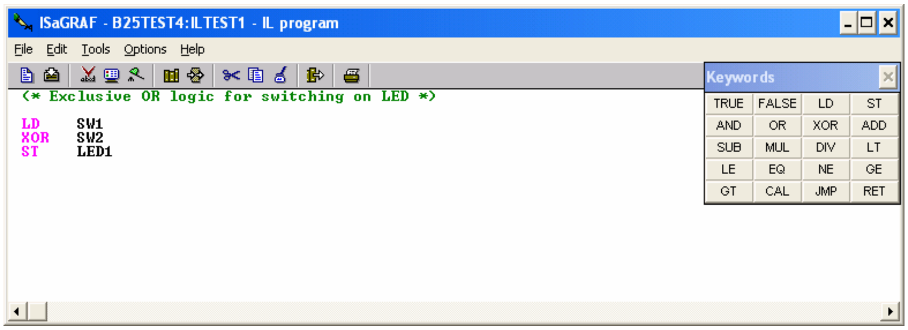
10. To enter **Keyword commands**, either click the desired button in the Keywords tool box or type it, such as for an ‘AND’ statement, either click **AND** in the Keyword tool box or type AND on the command line. When correctly entered Keyword commands are highlighted pink.
11. To enter **inputs, outputs, or variables**, for example input switch SW1, either select Edit → Insert variable → SW1 or type SW1 on the command line.
12. Type the following command lines under the entered comment:

```
LD     SW1

XOR    SW2

ST     LED1
```

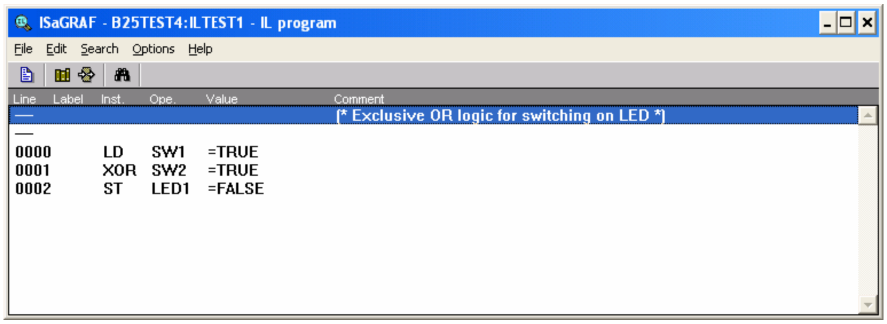
13. The **ILtest1 program** should now appear as follows:



```
(* Exclusive OR logic for switching on LED *)

LD      SW1
XOR     SW2
ST      LED1
```

14. Save the program by selecting **File** menu → **Save**. Press **OK** button on the **Update diary** window. Close the **ILtest1 – ST Program** window.
15. Follow Steps 19 to 22 in 3.4.1 to compile the application and download it to the Target PLC.
16. Double-click on **ILtest1** program. The **ILTEST1 – ST Programs** window is displayed, with the line numbers visible and the TRUE or FALSE current state for each variable displayed in the Value column. Short circuit digital input DI01 to set SW1 to FALSE, the LED1 will be turned ON as well as the state of LED1 in the Programs window will change. The PLC program implements a Boolean XOR operation on SW1 and SW2 with the result set on the LED1.


ISaGRAF - B25TEST4:ILTEST1 - IL program
File Edit Search Options Help

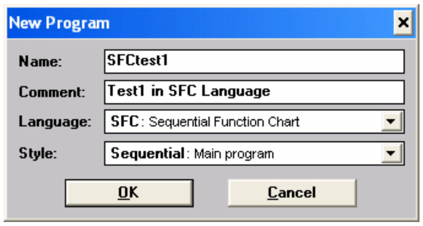
```
Line  Label  Inst.  Ope.   Value    Comment
—                                   (* Exclusive OR logic for switching on LED *)
—
0000         LD     SW1    =TRUE
0001         XOR    SW2    =TRUE
0002         ST     LED1   =FALSE
```

17. To stop the monitoring of the PLC application, close the **Debugger** window. The programs and dictionary windows in debug mode will be closed.

3.4.5 Using Sequential Function Chart (SFC)

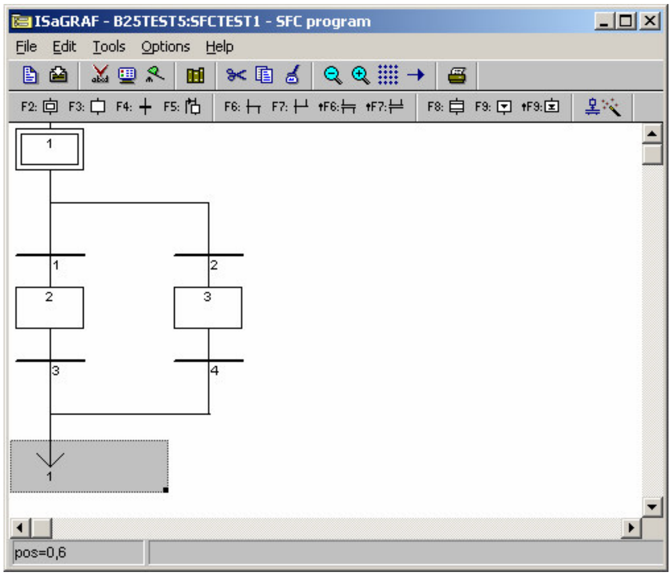
1. Open the ISaGRAF Project Management by double-clicking on the ISaGRAF Projects icon on your Desktop or selecting **Start** menu → **Programs** → **ISaGRAF 3.5** → **Projects**. On the **Projects Management** window, select **File** menu → **New**.
2. On the **Create new project** window, enter the name of the project (for example project5) and press **OK** button. The name of project should not exceed 8 characters. The project5 project is then listed in the **Projects Management** window.

3. Select **Edit** menu → **Set comment text**. Enter a brief description of this project and press **OK** button. This description will be displayed next to the project name in the **Projects Management** window.
4. Double-click on **project5** project. An empty **PROJECT5 – Programs** window is displayed.

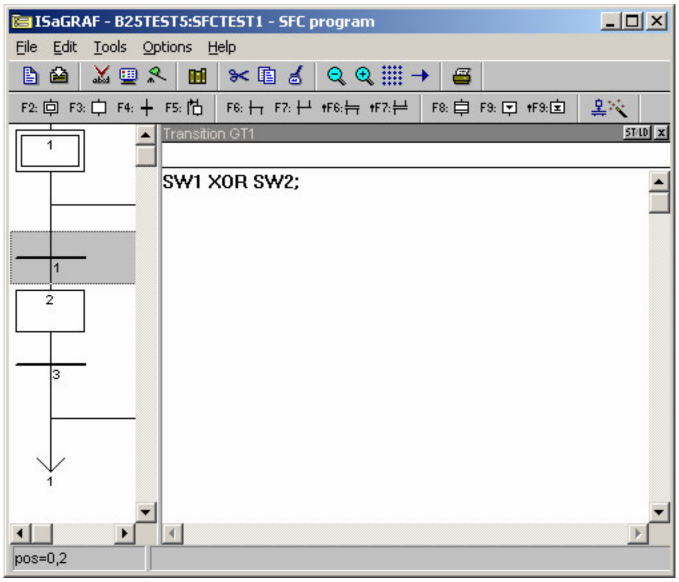


Name: SFCtest1
Comment: Test1 in SFC Language
Language: SFC : Sequential Function Chart
Style: Sequential : Main program

5. Select **File** menu → **New**. On the **New program** window enter the name of the program (for example, SFCtest1); a brief comment; select the Language to **SFC**; select **Begin** Style and press **OK** button. The SFCtest1 program is displayed in the **PROJECT5 – Programs** window.
6. Follow Steps 6 to 9 in Section 3.4.1 to create and populate application's Dictionary with three Boolean variables (SW1, SW2 and LED1).
7. On the **PROJECT5 – Programs** window double-click on **SFCtest1** program.
8. Click on below Step 1. Click on **OR divergence** F6: ⊢ to add a single divergence. Click on below single divergence and then on **Transition** F4: + to add a transition after the single divergence. Click on below the transition and add Step 2 by pressing **Step** F3: ☐ button. Repeat the same procedure to add Transitions 3, 4 and 5, Step 3, a single divergence and a Jump to Step 1. The final program screen should be as shown below.

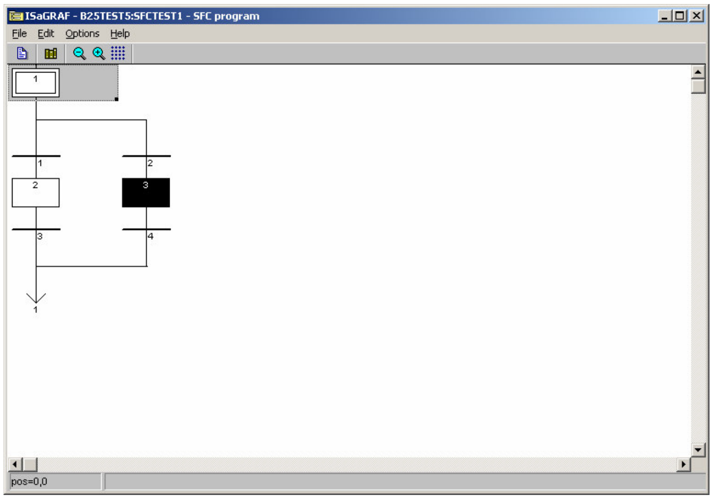


9. Double-click on Transition 1 and write the expression `SW1 XOR SW2;` on the **Transition GT1 ST/LD** editor that is displayed.


ISaGRAF - B25TEST5:SFCTEST1 - SFC program
File Edit Tools Options Help
F2: F3: F4: F5: F6: F7: ↑F6: ↑F7: F8: F9: ↑F9:
Transition GT1
ST LD
1
1
2
3
1
pos=0,2

```
SW1 XOR SW2;
```

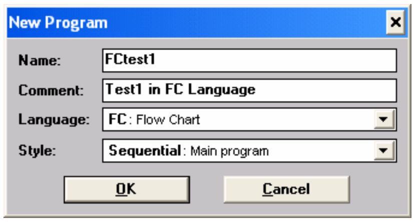
10. Double-click on Transition 2 and write the expression `not(SW1 XOR SW2);` on the **Transition GT2 ST/LD** editor that is displayed.
11. Double-click on Step 2 and write the action `LED1(S);` on the **Step GS2** that is displayed. Add the action `LED2(R);` on Step 3.
12. Save the program by selecting **File** menu → **Save**. Press **OK** button on the **Update diary** window. Close the **SFCtest1 – Program** window.
13. Follow Steps 19 to 22 in 3.4.1 to compile the application and download it to the Target PLC.
14. Double-click on **SFCtest1** program. The **SFCTEST1 – SFC Program** window is displayed. The step currently active is shown with a black box. Short circuit digital input DI01 to change the activity to Step 2 which set the LED1 (digital output DO01) output to TRUE.



15. To stop the monitoring of the PLC application, close the **Debugger** window. The programs and dictionary windows in debug mode will be closed.

3.4.6 Using Flow Chart (FC)

1. Open the ISaGRAF Project Management by double-clicking on the ISaGRAF Projects icon on your Desktop or selecting **Start** menu → **Programs** → **ISaGRAF 3.5** → **Projects**. On the **Projects Management** window, select **File** menu → **New**.
2. On the **Create new project** window, enter the name of the project (for example project6) and press **OK** button. The name of project should not exceed 8 characters. The project6 project is then listed in the **Projects Management** window.
3. Select **Edit** menu → **Set comment text**. Enter a brief description of this project and press **OK** button. This description will be displayed next to the project name in the **Projects Management** window.
4. Double-click on **project6** project. An empty **PROJECT6 – Programs** window is displayed.



Name: FCtest1
Comment: Test1 in FC Language
Language: FC : Flow Chart
Style: Sequential : Main program

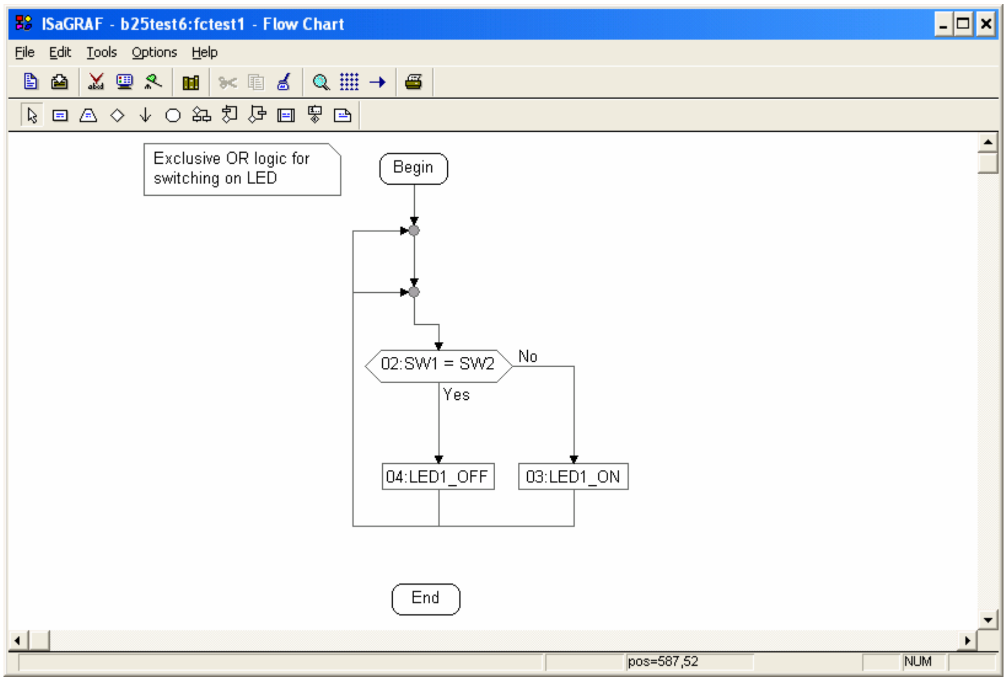
5. Select **File** menu → **New**. On the **New program** window enter the name of the program (for example, FCtest1); a brief comment; select the Language to **FC**; select **Begin** Style and press **OK** button. The FCtest1 program is displayed in the **PROJECT6 – Programs** window.
6. Follow Steps 6 to 9 in 3.4.1 to create and populate the application's Dictionary with SW1 and SW2 Boolean input variables, and LED1 Boolean output variable.
7. On the **PROJECT6 – Programs** window double-click on **FCtest1** program.
8. Select the **Insert a comment** button and place the comment box in the **FCtest1 Flow Chart** editor. Double click on the box to type in a comment such as "Exclusive OR logic for switching on LED".
9. Select the **Insert a test** button ◇ and place it between the "Begin" symbol and the "End" symbol. Double click on it to display the editing box for that symbol. Rename it SW1 = SW2, and type in the code SW1 = SW2.
10. Select the **Insert an action** button ▤ and place one on each of the two flow links from the test. Double click on the action symbol placed on the "Yes" flow link to display the editing box. Rename it LED1_OFF and type in the code:

    ```
    LED1 := FALSE;
    ```

    Double click on the action symbol placed on the "No" flow link to display the editing box. Rename it LED1_ON and type in the code:

    ```
    LED1 := TRUE;
    ```

11. Select the **Insert a Repeat-Until** button and place it in between the "Begin" symbol and the "SW1 = SW2" test symbol. Press Delete to just leave the flow line connector behind .
12. Select the **flow link** between the "Begin" symbol and the "End" symbol and delete it to allow for the program to run continuously without ending.
13. Click on the flow line connector that both the "LED_OFF" and the "LED_ON" action symbols connect to. Select the **Insert a flow** button ↓ and connect both the "LED_OFF" and the "LED_ON" action symbols to the flow line connector remaining from **Step 11.** This allows the "SW1 = SW2" test to repeat continuously, so if either of SW1 or SW2 change it can test again and change the LED1 output.
14. The **FCtest1 Flow Chart** should now appear as follows:


ISaGRAF - b25test6:fctest1 - Flow Chart
File Edit Tools Options Help
Exclusive OR logic for switching on LED
Begin
02:SW1 = SW2
No
Yes
04:LED1_OFF
03:LED1_ON
End
pos=587,52
NUM

15. Save the program by selecting **File** menu → **Save**. Press **OK** button on the **Update diary** window. Close the **FCtest1 – Flow Chart** window.

16. Follow Steps 19 to 22 in 3.4.1 to compile and download it to the Target PLC.

17. Double-click on **FCtest1** program. The **FCTEST1 – Flow Chart** window is displayed. Short circuit the digital input DI01 (SW1 will be FALSE), the LED1 will be turned ON as well as the state of LED1 in the Programs window will change. The PLC program implements a Boolean XOR operation on SW1 and SW2 with the result set on the LED1.

18. To stop the monitoring of the PLC application, close the **Debugger** window. The programs and dictionary windows in debug mode will be closed.

The ISaGRAF User's Manual (See [1]) and the on-line help of the ISaGRAF Workbench describes in detail all the Workbench functionalities (editor, simulation, debugger, etc.) including also a reference manual of the PLC programming languages (IEC 61131-3).

4. MODBUS PROTOCOL

4.1 EMBEDDED PLC TARGET AS A MODBUS SLAVE

The EMBEDDED PLC Target implements Modbus slave RTU over serial line (RS232 or RS485) or Modbus slave on TCP protocols to exchange data with other systems.

Modbus is an application layer messaging protocol for client/server communication between devices connected on different types of buses or network. For more details about the protocol specification see [2] available from the Modbus-IDA Organization website http://www.modbus.org/specs.php.

The following Modbus function codes are supported by the EMBEDDED PLC target:

Function Code (Decimal)	**Action**
1	Read coils (or digital outputs)
2	Read discrete inputs (or digital inputs)
3	Read holding registers (or analogue outputs)
4	Read input registers (or analogue inputs)
5	Write single coil (1 bit)
6	Write single register (1 word)
15	Write multiple coils (N bits)
16	Write multiple registers (N words)

4.1.1 Configuring Modbus Slave

The EMBEDDED PLC BL2500 comes pre-configured as a Modbus slave TCP. Use the EMBEDDED PLC Utility to change the Modbus slave parameters of the target. Go to section 7 for instructions on how to install the EMBEDDED PLC Utility. Section 7.1.4 describes step-by-step how to change the configuration of the EMBEDDED PLC Modbus slave interface.

✍ **Note:**

The Modbus TCP interface implemented in the EMBEDDED PLC target supports up to 5 simultaneous socket connections, excluding the ISaGRAF Workbench.

When using Modbus over RS232/RS485, the ISaGRAF Workbench and the Modbus master share the same physical port; therefore only one can be connected to the target at any given time.

4.1.2 Defining Variable Addresses for access via Modbus

To make data available from the EMBEDDED PLC target to a Modbus master, you must first define the variable with a "Network Address". The variable must be declared with a network address that is in the Modbus format. All the addresses are expressed in this document in decimal format, unless other format is explicitly specified.

✍ **Note:**

Normally, the ISaGRAF Workbench always formats network addresses in hexadecimal. There is a new undocumented option in ISaGRAF V3.5 to set all network address displays and edit controls to decimal. This option is stored in isa.ini file located in <ISaGRAF folder>\EXE:

```
[EDIT]
NwAddrDecimal=1 ; 1=Decimal / 0=Hexadecimal
```

Table 2 describes the Modbus address mapping supported by the EMBEDDED PLC target.

Table 2: Modbus slave addresses mapping

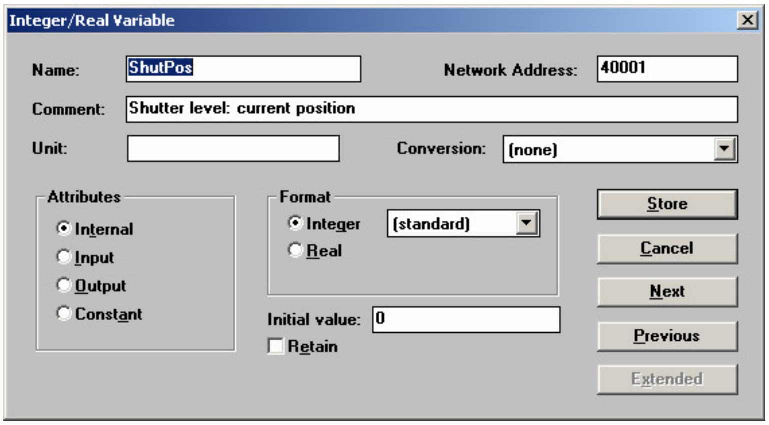
Modbus Data Type	Network Address range (Decimal)
Coils (Read/Write)	1 – 9999
Discrete Inputs (Read Only)	10001 – 19999
Input Registers (16-bit Integer, Read Only)	30001 – 34999
Input Registers (32-bit Integer/Float, Read Only)	35001 – 39997 (odd address)
Holding Registers (16-bit Integer, Read/Write)	40001 – 44999
Holding Registers (32-bit Integer/Float, Read Only)	45001 – 49997 (odd address)

In order to exchange 32-bit data types via Modbus, each value uses two words (32-bit) of data via read or write multiple registers function codes. The data value is grouped as less significant word first. Since it uses two words, two contiguous addresses are needed to access the 32-bit value, therefore the starting address of the register must be odd: 35001, 35003, etc for input registers; and 45001, 45003, etc. for holding registers.

From the Modbus master point of view, the data address of each type starts from 1. For example, if the Modbus master needs to access the Coil 21, the network address defined in the target must be equal to 21. If the Modbus master needs to access the Discrete Input 21, the network address defined in the target must be equal to 10021. If the Modbus master needs to access the Input Register 21 reported as 16-bit integer (1 word), the network address defined in the target must be equal to 30021. If the Modbus master needs to access the Holding Register 21 reported as 16-bit integer (1 word), the network address defined in the target must be equal to 40021. If the Modbus master needs to access a register reported as 32-bit integer or float, the address defined in the master must be odd and start from 5001; and the network address defined in the target must start from 35001 (read only) or 45001 (read or write).

There are two methods in ISaGRAF Workbench to declare a variable for Modbus access:

1. Open ISaGRAF Workbench and open the desired program. Click on Dictionary icon (or click on **File** menu → **Dictionary**). Double click on the variable to assign the Modbus address. The following dialog window is displayed.



Integer/Real Variable

Name: ShutPos

Network Address: 40001

Comment: Shutter level: current position

Unit:

Conversion: (none)

Attributes: ● Internal ○ Input ○ Output ○ Constant

Format: ● Integer (standard) ○ Real

Initial value: 0

☐ Retain

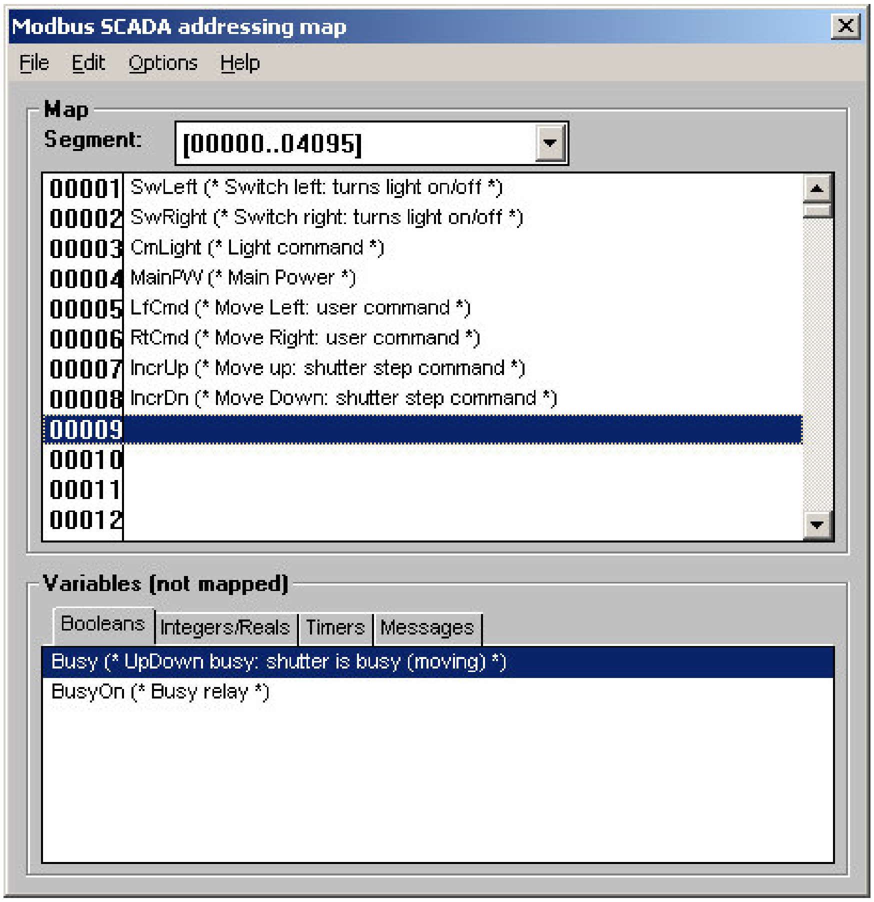
Enter the complete Modbus address in the **Network Address** field and press Store button. See the previous Note on how to change the format (Decimal or Hexadecimal) of the Network Address field of the above window.

If the variable is of Boolean type and its attribute is internal or output, the Network Address must be defined within the range of Modbus Coils (1 – 9999). If the variable is of Boolean type and its attribute is internal or input, the Network Address must be defined within the range of Discrete Inputs (10001 – 19999).

If the variable is of Integer type and its attribute is internal or input, the Network Address must be defined within the range of Input Registers (30001 – 39997). If the variable if of Integer type and its attribute is internal or output, the Network Address must be defined within the range of Holding Registers (40001 – 49997). If the variable is of Real type it must be exchanged as 32-bit value, therefore its address must be in the range of 35001 – 39997 (internal or input attribute) or 45001 – 49997 (internal or output attribute).

2. The second method of assigning network addresses to variables requires that you define the variables BEFORE assigning addresses. This method allows you to assign several addresses before you link them to an ISaGRAF program.

   On the Dictionary window, select **Tools** menu → **Modbus SCADA Addressing Map**. The following window is displayed.



Segment: [00000..04095]

00001 SwLeft (* Switch left: turns light on/off *)
00002 SwRight (* Switch right: turns light on/off *)
00003 CmLight (* Light command *)
00004 MainPW (* Main Power *)
00005 LfCmd (* Move Left: user command *)
00006 RtCmd (* Move Right: user command *)
00007 IncrUp (* Move up: shutter step command *)
00008 IncrDn (* Move Down: shutter step command *)
00009
00010
00011
00012

Busy (* UpDown busy: shutter is busy (moving) *)
BusyOn (* Busy relay *)

In the **Options** menu you can select the displayed address format (Hexadecimal or Decimal).

To assign the unmapped variables, position the cursor in the desired address and then double click on the variable to assign. To assign continuous network addresses to several variables, click on the starting address in the Map segment section, select the variables in the not-mapped section using the SHIFT key and click on **Edit** menu → **Map selected variable**.

4.2 EMBEDDED PLC TARGET AS A MODBUS MASTER

The EMBEDDED PLC Target can also be used as a Modbus master over TCP. A typical Modbus TCP/IP distributed communication architecture is shown in next figure.

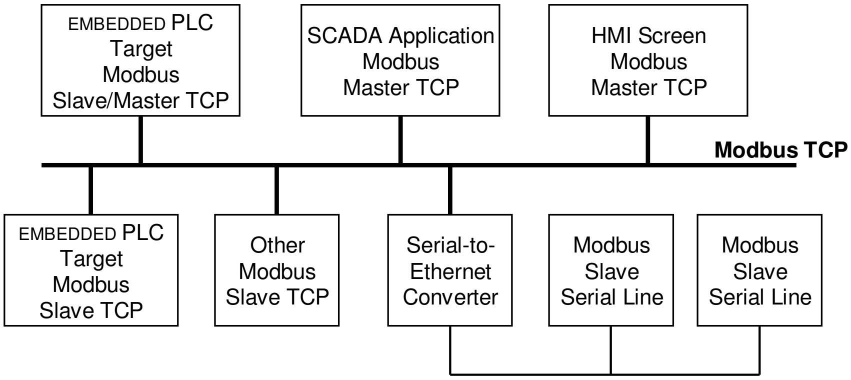

EMBEDDED PLC Target Modbus Slave/Master TCP
SCADA Application Modbus Master TCP
HMI Screen Modbus Master TCP
Modbus TCP
EMBEDDED PLC Target Modbus Slave TCP
Other Modbus Slave TCP
Serial-to-Ethernet Converter
Modbus Slave Serial Line
Modbus Slave Serial Line

Figure 2: Modbus TCP communication architecture

The EMBEDDED PLC Modbus Master implementation supports the following functions through IEC 61131-3 programming environment:

- Write 8 or 16 consecutive coils (0x).
- Read 8 or 16 consecutive discrete inputs (1x).
- Read 8 or 16 consecutive input registers (3x) as 16-bit integer value or 32-bit integer/float value.
- Write 8 or 16 consecutive holding registers (4x) as 16-bit integer value or 32-bit integer/float value.

The EMBEDDED PLC target acting as a Modbus master can communicate with up to one Modbus slave over TCP or up to 16 Modbus slaves over serial via Serial-to-Ethernet converter.

4.2.1 Configuration of Modbus Master over TCP

The configuration of the Modbus master interface is done using ISaGRAF I/O boards and I/O complex equipments. The first step is to use the Interface Configuration I/O board (**mbtcpcfg**). This board must be defined in the I/O connection list before the data groups are declared. The following parameters are setup through this I/O board:

- **slave_ipAddr** (Data type: String): IP Address of the Modbus TCP slave or the IP address of the serial-to-Ethernet converter (default 192.168.1.100).
- **slave_port** (Data type: Integer): This is a read-only parameter and corresponds to the TCP port used by Modbus which is equal to 502.
- **polling_period** (Data type: Long): The Modbus master request every **polling_period** (in milliseconds) a reading of all the data. In case of an output data group (coils or holding registers) it uses this information for data quality.
- **timeout** (Data type: Long): Connection timeout in milliseconds (default 1000).
- **retries** (Data type: Integer) Number of retries to establish connection before setting the status of the interface as BAD (default 2).

Once the Interface Configuration board is connected in one available slot, the Data Groups I/O complex equipments can be connected. These equipments are used to configure and exchange the data available in the Modbus slave. The following Data Groups are available (through ISaGRAF I/O complex equipments):

- **mtcpNai**: Read N consecutive input registers (3x) where N can be 8 or 16.
- **mtcpNdi**: Read N consecutive discrete inputs (1x) where N can be 8 or 16.
- **mtcpNao**: Write N consecutive holding registers (4x) where N can be 8 or 16.
- **mtcpNdo**: Write N consecutive coils (0x) where N can be 8 or 16.

Each I/O complex equipment consists of a configuration board (**cfgstat**) and the data board. The following parameters are common to all the I/O complex equipments and are setup through the configuration board (**cfgstat**):

- **slave_ipAddr** (Data type: String): IP address defined in the Modbus Master TCP Interface Configuration I/O Board (Default 192.168.1.100). This value is used to link this data group to the main TCP interface and must be the same as defined in the TCP Interface Configuration board (**mbtcpcfg**). Multiple data groups can be linked to one TCP Interface board.
- **slave_address** (Data type: Integer): Modbus slave address used in multiple slaves configuration (Default 1). This parameter is meaningful only when a serial-to-Ethernet converter is used to connect multiple Modbus slaves over serial line.

The configuration board contains an input channel (Channel 1) which returns the quality of the data (GOOD or BAD).

The data board usually contains a parameter with the starting address of the data group. It also contains the channels on which the ISaGRAF variables are connected to exchange the data values with the Modbus slave. All writing operations (coils and holding registers) require variables with output attribute. All reading operations (discrete inputs and input registers) require variables with input attribute.

The next section contains a couple of examples on how to configure the Modbus master interface. Section **Error! Reference source not found.** describes in details each parameter of the Modbus master I/O boards and complex equipment.

4.2.2 Examples

4.2.2.1 Modbus Master communicating to one Modbus Slave

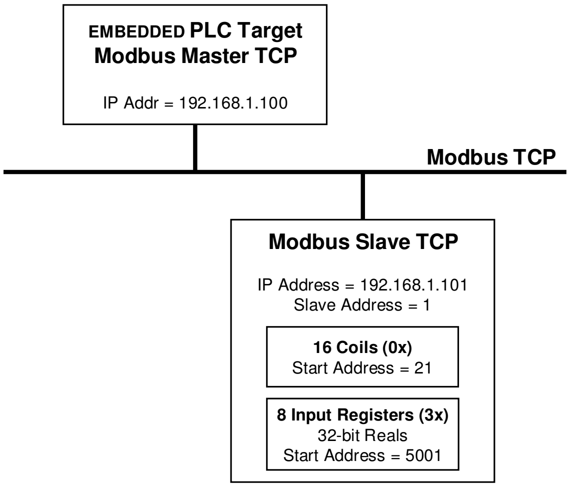
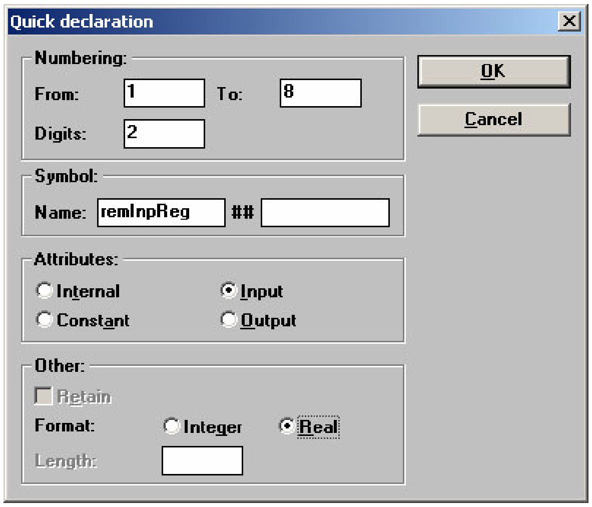


Figure 3: Example 1: Modbus Master to one Modbus Slave over TCP

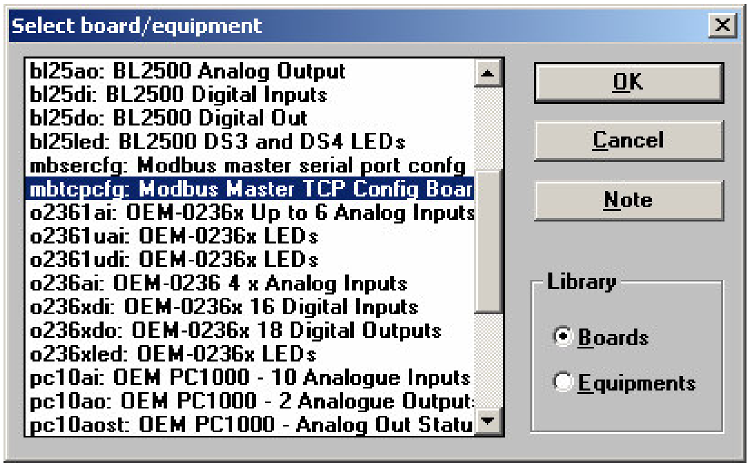
There is one EMBEDDED PLC Target configured as a Modbus Master over TCP and one Modbus Slave over TCP which has 16 Coils (or digital outputs) starting from address 21 and 8 Input Registers (analogue inputs, 32-bit Real) starting from address 5001. All addresses are expressed in Decimal format.

3. The EMBEDDED PLC Target (Modbus Master) must be configured as TCP with the IP Address 192.168.1.100. See Section 7.1.4 for instructions on how to configure the EMBEDDED PLC firmware kernel.
4. Open ISaGRAF Workbench and create a new project called mdbMast1. Open the Dictionary and define 16 Boolean output variables (remCoil01 – remCoil02) and 8 Analogue input variables (remInpReg01 – remInpReg08) as Reals. You can use the Quick Declaration tool to quickly define those variables (**Tools** menu → **Quick Declaration**). Save and close the Dictionary.



From: 1 To: 8
Digits: 2
Name: remInpReg ##
○ Internal ● Input
○ Constant ○ Output
Format: ○ Integer ● Real

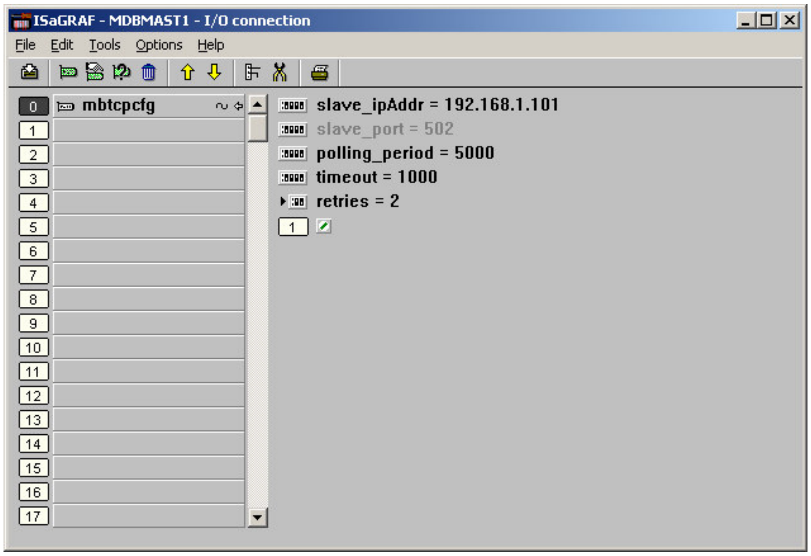
5. In the ISaGRAF Programs window click the I/O connection tool icon. On the I/O Connection tool click the Set board/equipment icon. The following window is displayed.



bl25ao: BL2500 Analog Output
bl25di: BL2500 Digital Inputs
bl25do: BL2500 Digital Out
bl25led: BL2500 DS3 and DS4 LEDs
mbsercfg: Modbus master serial port confg
mbtcpcfg: Modbus Master TCP Config Boar
o2361ai: OEM-0236x Up to 6 Analog Inputs
o2361uai: OEM-0236x LEDs
o2361udi: OEM-0236x LEDs
o236ai: OEM-0236 4 x Analog Inputs
o236xdi: OEM-0236x 16 Digital Inputs
o236xdo: OEM-0236x 18 Digital Outputs
o236xled: OEM-0236x LEDs
pc10ai: OEM PC1000 - 10 Analogue Inputs
pc10ao: OEM PC1000 - 2 Analogue Output
pc10aost: OEM PC1000 - Analog Out Statu

● Boards
○ Equipments

6. Select **mbtcpcfg** I/O configuration board and press OK. The following information is displayed in the I/O connection tool. This board is the main configuration of the Modbus master interface. All the parameters displayed here affect all the data groups linked to this board. This configuration board MUST be defined before the Modbus data groups boards.

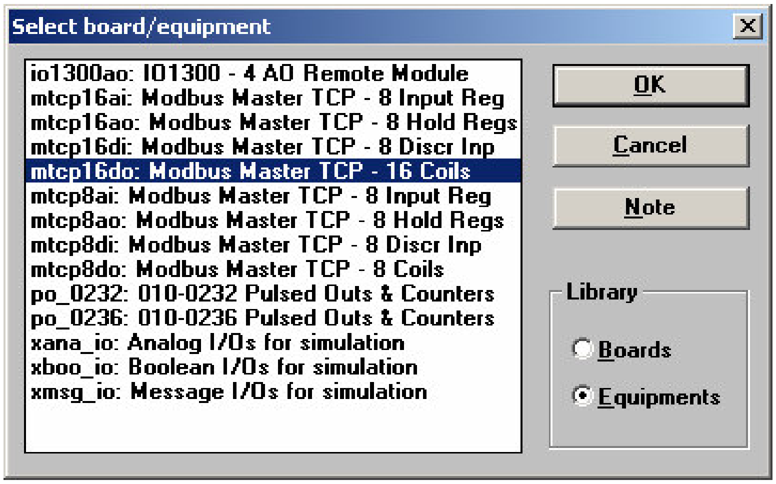

ISaGRAF - MDBMAST1 - I/O connection
File Edit Tools Options Help
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
1

mbtcpcfg

slave_ipAddr = 192.168.1.101
slave_port = 502
polling_period = 5000
timeout = 1000
retries = 2

7. Double click on **slave_ipAddr** configuration parameter and enter the IP address of the Modbus slave device 192.168.1.101. The **polling_period** and **timeout** are in millisecond. The status of the Modbus master interface is returned as an integer value in Channel 1 of this board.

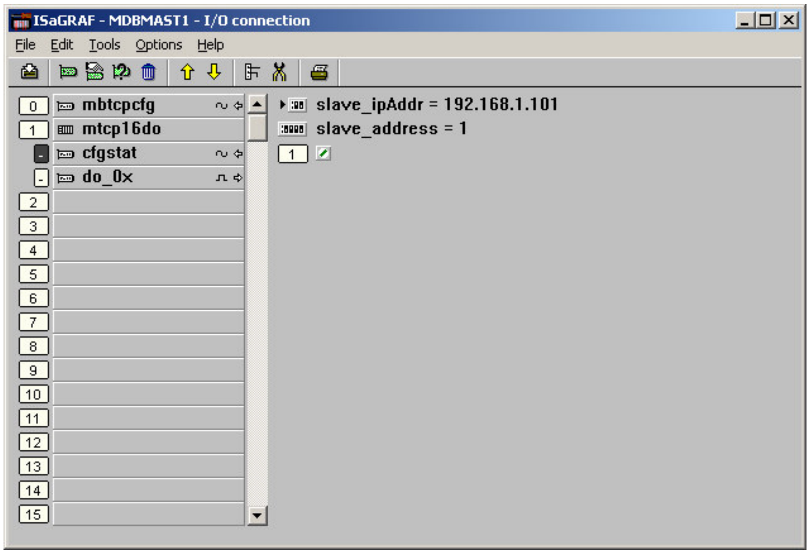
8. On the left side of the I/O connection tool double click on Slot 1 and select Equipment option on the window displayed below.



io1300ao: IO1300 - 4 AO Remote Module
mtcp16ai: Modbus Master TCP - 8 Input Reg
mtcp16ao: Modbus Master TCP - 8 Hold Regs
mtcp16di: Modbus Master TCP - 8 Discr Inp
mtcp16do: Modbus Master TCP - 16 Coils
mtcp8ai: Modbus Master TCP - 8 Input Reg
mtcp8ao: Modbus Master TCP - 8 Hold Regs
mtcp8di: Modbus Master TCP - 8 Discr Inp
mtcp8do: Modbus Master TCP - 8 Coils
po_0232: 010-0232 Pulsed Outs & Counters
po_0236: 010-0236 Pulsed Outs & Counters
xana_io: Analog I/Os for simulation
xboo_io: Boolean I/Os for simulation
xmsg_io: Message I/Os for simulation

9. Select the Modbus data group I/O equipment **mtcp16do** which corresponds to the 16 Coils the Modbus slave support.

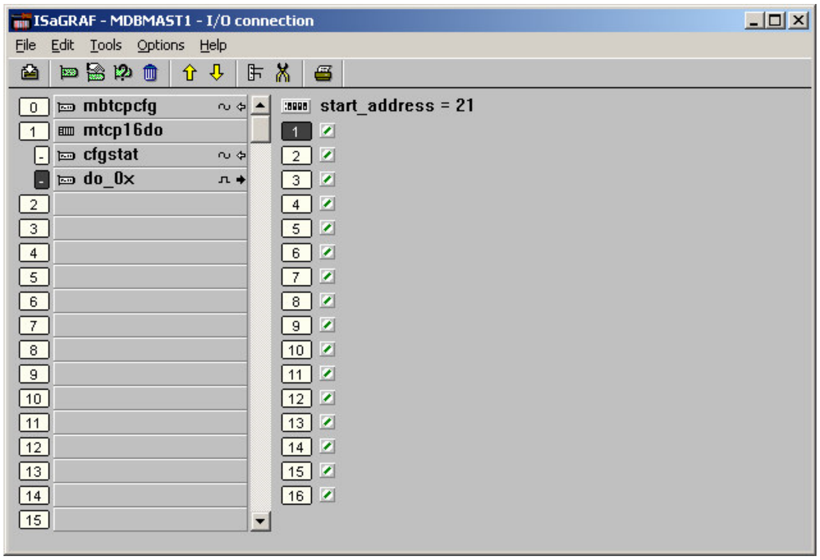
10. Under **mtcp16do** equipment, double click on **cfgstat** board. Enter the **slave_ipAddr** defined previously in the **mbtcpcfg** board 192.168.1.101. It is important that the IP address defined in the TCP Interface Configuration board (**mbtcpcfg**) and the configuration board of the **mtcp16do** data group are the same. The **slave_address** parameter is used if you have a TCP/IP to RS232/485 converter where multiple slaves can be connected to one IP address. The status of the data quality of the group is returned in Channel 1 of the **cfgstat** board (integer input).



mbtcpcfg
mtcp16do
cfgstat
do_0x

slave_ipAddr = 192.168.1.101
slave_address = 1

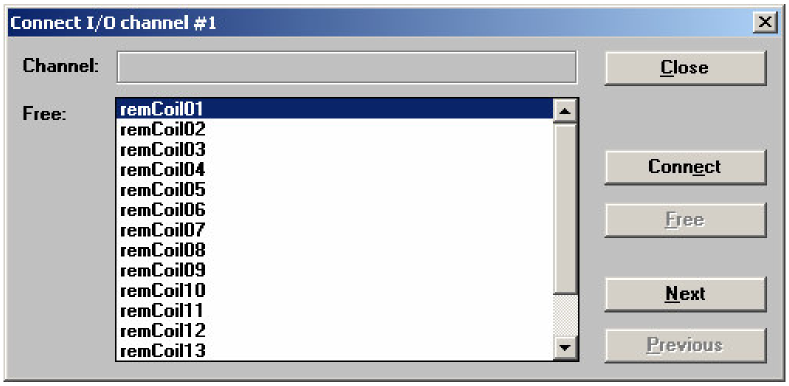
11. Under **mtcp16do** equipment double click on **do_0x** board. Double click on **start_address** parameter to modify the starting address of the Modbus data group.



mbtcpcfg
mtcp16do
cfgstat
do_0x

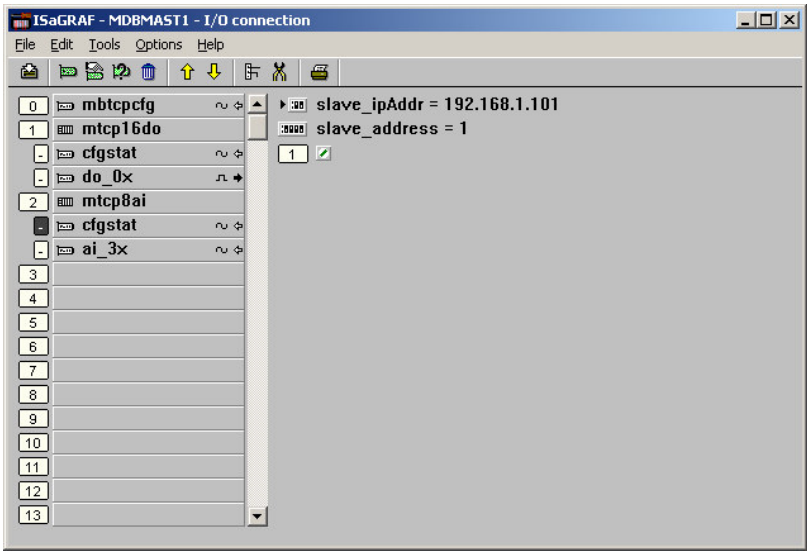
start_address = 21

12. On the right section of the I/O Connection tool double click on Channel 1. On the Connect I/O channel window select the desired free output variable and press **Connect**. Repeat the same for the other channels. Close the Connect I/O channel window after all the output variables have been connected.



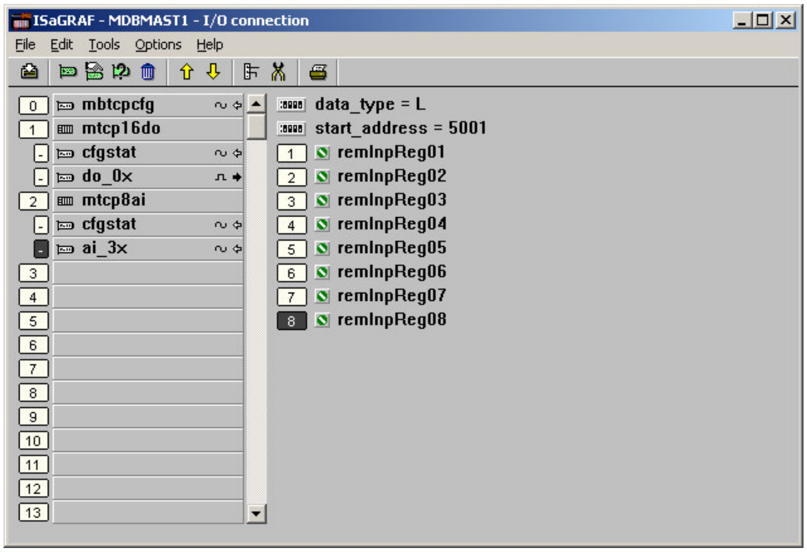
remCoil01
remCoil02
remCoil03
remCoil04
remCoil05
remCoil06
remCoil07
remCoil08
remCoil09
remCoil10
remCoil11
remCoil12
remCoil13

13. Double-click on Slot 2 of the I/O Connection window and select the **mtcp8ai** Equipment. Set the **slave_ipAddr** to the same value defined in the TCP Interface Configuration board (192.168.1.101).



slave_ipAddr = 192.168.1.101
slave_address = 1

14. Click on ai_3x board and set the **data_type** value to 'L' (Long data type for 32-bit Reals or Integers) and **start_address** to 5001. Connect the analogue input real variables remInpReg01 to remInpReg02 to the corresponding channels. Save and close the I/O Connection tool.



data_type = L
start_address = 5001
1 remInpReg01
2 remInpReg02
3 remInpReg03
4 remInpReg04
5 remInpReg05
6 remInpReg06
7 remInpReg07
8 remInpReg08

When using input or holding registers, there is an extra parameter which defines the format of the data value: 'S' for 16-bit or 'L' for 32-bit. If the variable connected to the channel is integer, the value will be treated as integer. If the attribute of the variable connected to the channel is Real, the value will be treated as floating point.

If this project is compiled and downloaded to the target, the Boolean output variables **remCoil01** to **remCoil16** are used to write Boolean values to the Coils in the slave. The Analogue input variables hold the values of the Input Registers in the slave.

4.2.2.2 Modbus Master communicating to two slaves via RS485-to-Ethernet converter

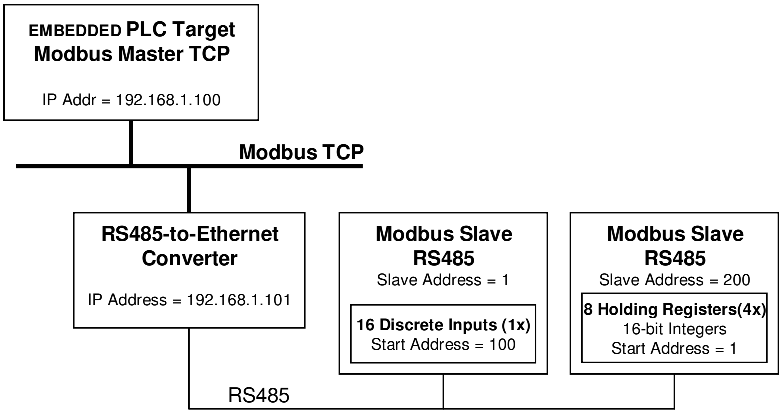
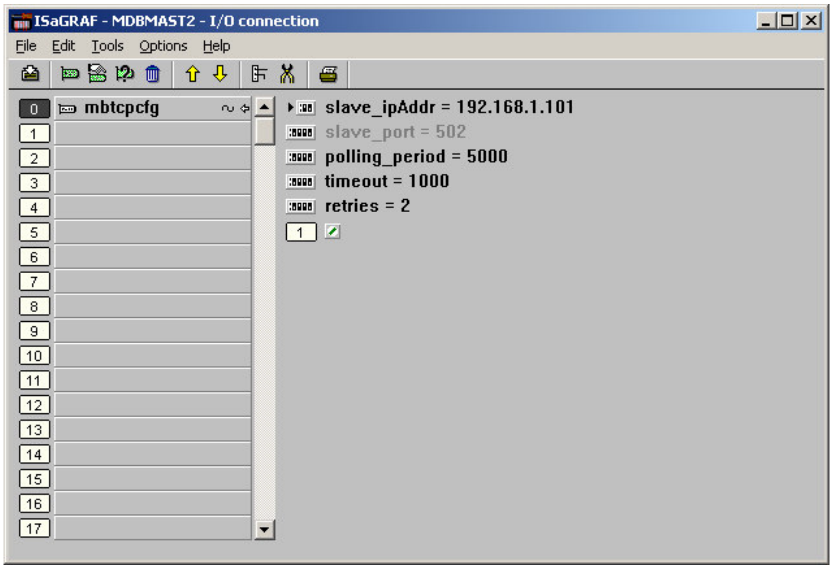


Figure 4: Example 2: Modbus Master TCP to two Modbus Slaves over Serial

There is an EMBEDDED PLC target acting as the Modbus Master over TCP communicating to two Modbus slaves over RS485 via an RS485-to-Ethernet converter. One Modbus slave has 16 Discrete Inputs (or digital inputs) starting from address 100. The other Modbus slave has 8 Holding Registers (or analogue outputs) starting from address 1 and its values are reported as 16-bit Integer. The IP Address of the Modbus Master is 192.168.1.100 and the IP address of the RS485-to-Ethernet Converter is 192.168.1.101.

The following step-by-step instructions describe how to configure the Modbus master within an ISaGRAF project:

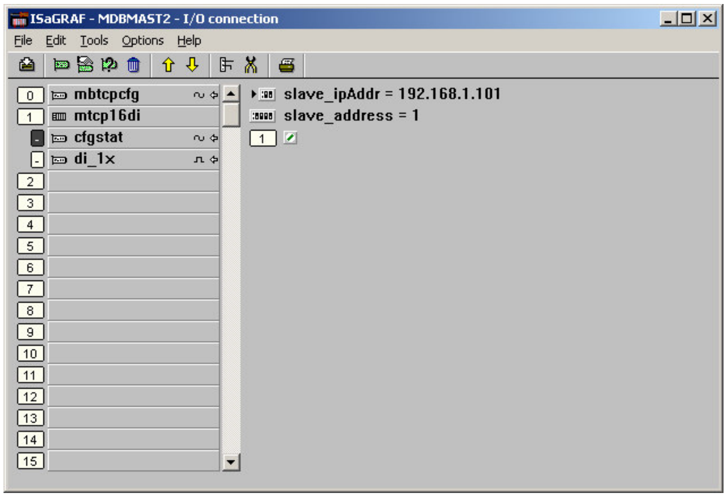
15. The EMBEDDED PLC Target (Modbus Master) must be configured as TCP with the IP Address 192.168.1.100. See Section 7.1.4 for instructions on how to configure the EMBEDDED PLC firmware kernel.
16. Open ISaGRAF Workbench and create a new project called **mdbMast2**. Open the Dictionary and define 16 Boolean input and 8 Analogue output variables.
17. Open the I/O Connection tool and connect the Modbus TCP Interface Configuration board (**mbtcpcfg**) as described in Steps 3 to 5 of the previous example.


ISaGRAF - MDBMAST2 - I/O connection
File Edit Tools Options Help
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

0 mbtcpcfg

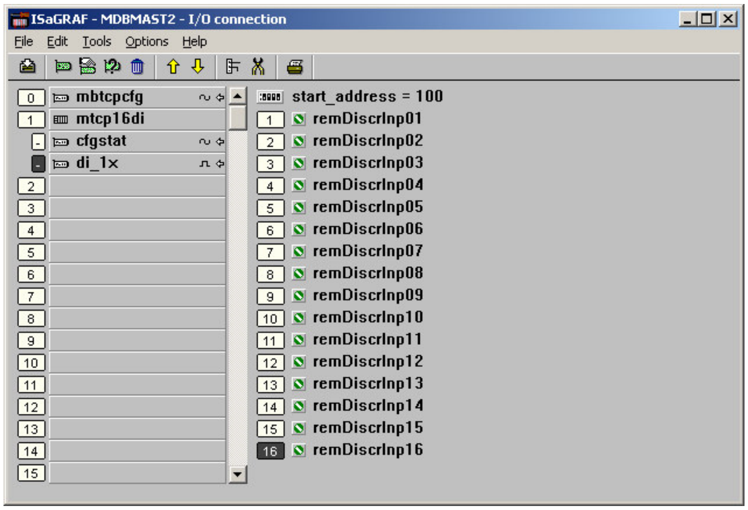
slave_ipAddr = 192.168.1.101
slave_port = 502
polling_period = 5000
timeout = 1000
retries = 2

18. Double-click on Slot 1, select Equipments and connect **mtcp16di** I/O equipment. Select **cfgstat** board and set the **slave_ipAddr** to 192.168.1.101 and **slave_address** to 1.



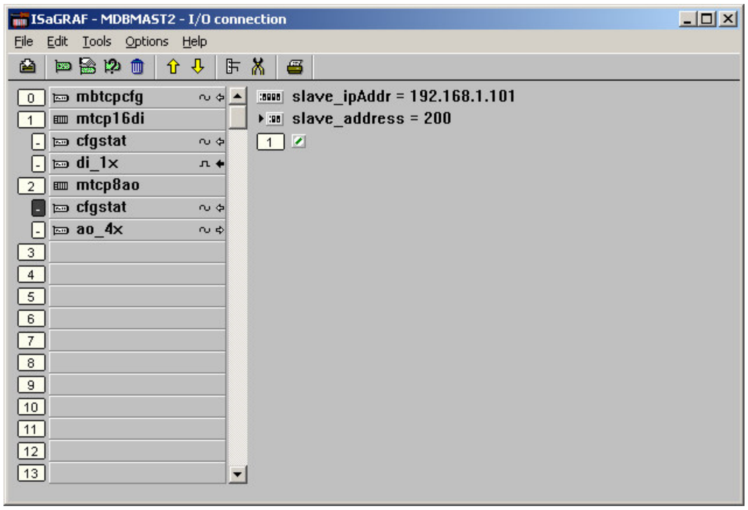
slave_ipAddr = 192.168.1.101
slave_address = 1
1

19. Select **di_1x** board, set the **start_address** to 100 and connect all the Boolean input variables.



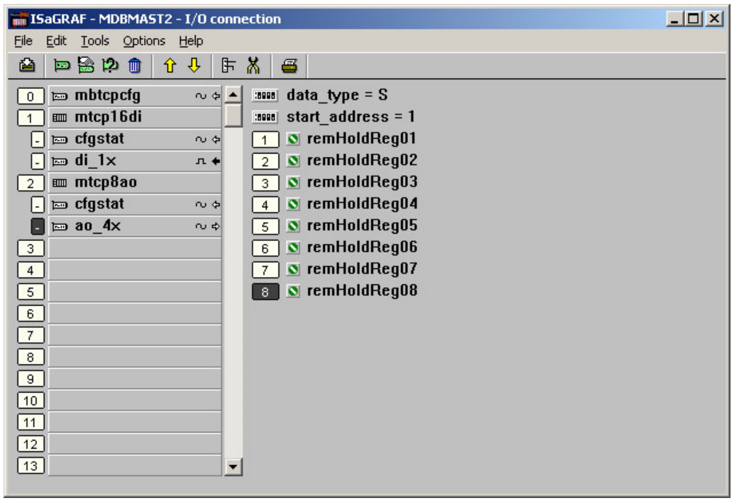
start_address = 100
1 remDiscrInp01
2 remDiscrInp02
3 remDiscrInp03
4 remDiscrInp04
5 remDiscrInp05
6 remDiscrInp06
7 remDiscrInp07
8 remDiscrInp08
9 remDiscrInp09
10 remDiscrInp10
11 remDiscrInp11
12 remDiscrInp12
13 remDiscrInp13
14 remDiscrInp14
15 remDiscrInp15
16 remDiscrInp16

20. Double-click on Slot 2, select Equipments and connect **mtcp8ao** I/O equipment. Select **cfgstat** board and set the **slave_ipAddr** to 192.168.1.101 and **slave_address** to 200.



slave_ipAddr = 192.168.1.101
slave_address = 200
1

21. Select **ao_4x** board, set the **data_type** to ‘S’ (Short or 16-bit value) and **start_address** to 1. Connect all the Analogue output variables. Save and close the I/O Connection tool.



0 mbtcpcfg
1 mtcp16di
- cfgstat
- di_1x
2 mtcp8ao
- cfgstat
- ao_4x
3
4
5
6
7
8
9
10
11
12
13

data_type = S
start_address = 1
1 remHoldReg01
2 remHoldReg02
3 remHoldReg03
4 remHoldReg04
5 remHoldReg05
6 remHoldReg06
7 remHoldReg07
8 remHoldReg08

5. HARDWARE REFERENCE

5.1 JUMPER SETTINGS

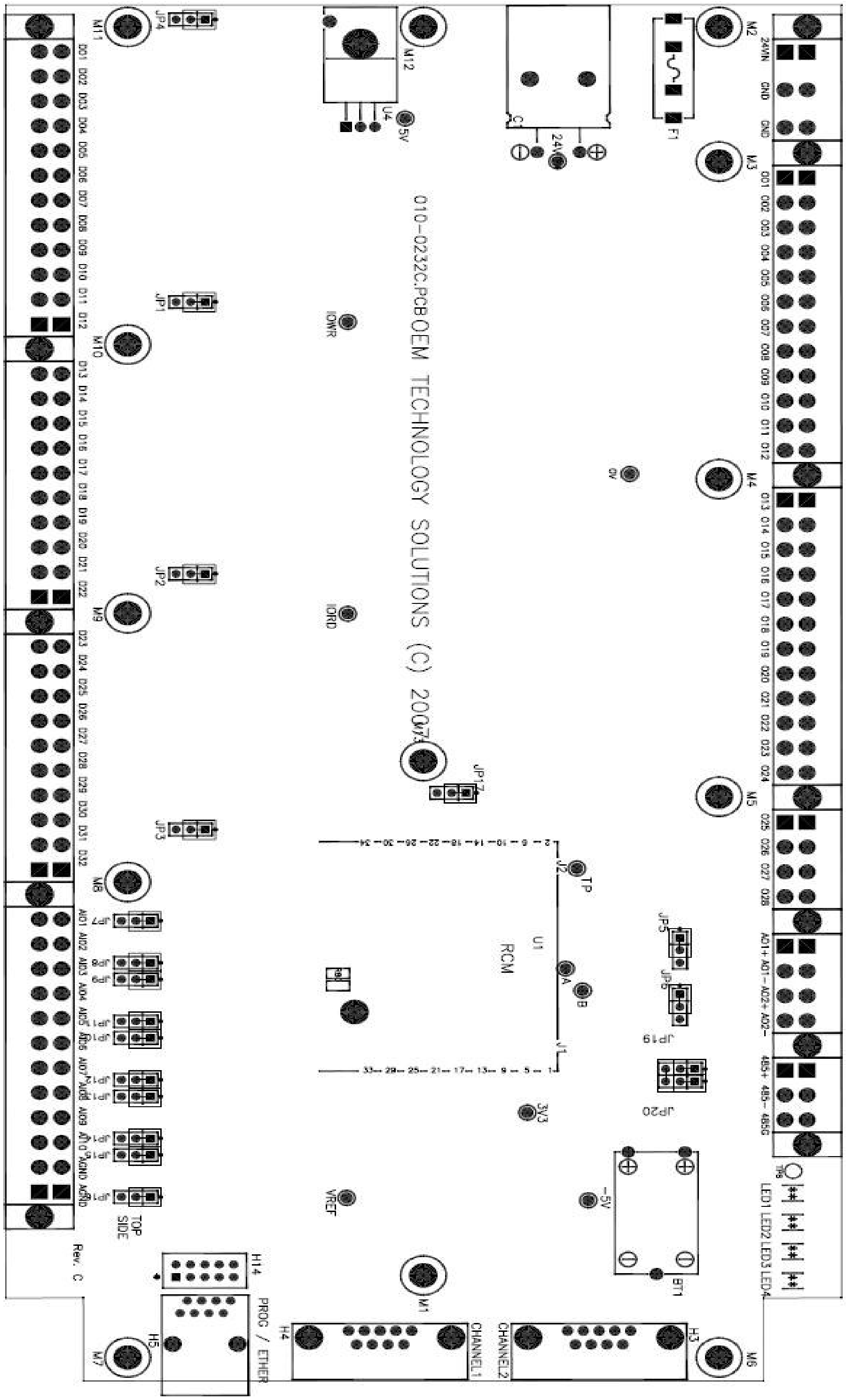

M11
JP4
M12
U4
5V
C1
24V
F1
M2
24VIN GND GND
M3
DI1 DI2 DI3 DI4 DI5 DI6 DI7 DI8 DI9 DI10 DI11 DI12
M10
DI13 DI14 DI15 DI16 DI17 DI18 DI19 DI20 DI21 DI22
M9
DI23 DI24 DI25 DI26 DI27 DI28 DI29 DI30 DI31 DI32
M8
AD1 AD2 AD3 AD4 AD5 AD6 AD7 AD8 AD9 AD10 AGND AGND
DO1 DO2 DO3 DO4 DO5 DO6 DO7 DO8 DO9 DO10 DO11 DO12
M4
DO13 DO14 DO15 DO16 DO17 DO18 DO19 DO20 DO21 DO22 DO23 DO24
M5
DO25 DO26 DO27 DO28
AI1+ AI1- AI2+ AI2-
485+ 485- 485G
JP1
IOWR
010-0232C.PCB OEM TECHNOLOGY SOLUTIONS (C) 2007
JP2
IORD
0V
M1
JP17
JP3
JP7
JP8
JP9
TOP SIDE
J2
TP
U1
RCM
A
B
J1
JP5
JP6
JP19
JP20
3V3
-5V
VREF
BT1
TP8
LED1 LED2 LED3 LED4
Rev. C
H14
H5
M7
PROG / ETHER
H4
CHANNEL1
CHANNEL2
H3
M6

Figure 5: pro•teso PC1000 Jumper Settings

Table 3: pro•teso PC1000 Jumper Settings

Jumper	Position	Description	Default
JP4	1 – 2 2 – 3	Pull-up D01 – D08 to 24 VDC Pull-down D01 – D08 to 0 VDC	1 – 2
JP1	1 – 2 2 – 3	Pull-up D09 – D16 to 24 VDC Pull-down D09 – D16 to 0 VDC	1 – 2
JP2	1 – 2 2 – 3	Pull-up D17 – D24 to 24 VDC Pull-down D17 – D24 to 0 VDC	1 – 2
JP3	1 – 2 2 – 3	Pull-up D25 – D32 to 24 VDC Pull-down D25 – D32 to 0 VDC	1 – 2
JP5	1 – 2 2 – 3	Analogue Output 1: 0 – 10 VDC Analogue Output 1: 0 – 20 mA	1 – 2
JP6	1 – 2 2 – 3	Analogue Output 2: 0 – 10 VDC Analogue Output 2: 0 – 20 mA	1 – 2
JP7 (*)	1 – 2 2 – 3	Analogue Input 1: 0 – 20 mA Analogue Input 1: 0 – 10 VDC or Thermistor	1 – 2
JP8 (*)	1 – 2 2 – 3	Analogue Input 2: 0 – 20 mA Analogue Input 2: 0 – 10 VDC or Thermistor	1 – 2
JP9 (*)	1 – 2 2 – 3	Analogue Input 3: 0 – 20 mA Analogue Input 3: 0 – 10 VDC or Thermistor	1 – 2
JP10 (*)	1 – 2 2 – 3	Analogue Input 4: 0 – 20 mA Analogue Input 4: 0 – 10 VDC or Thermistor	1 – 2
JP11	1 – 2 2 – 3	Analogue Input 5: 0 – 20 mA Analogue Input 5: 0 – 10 VDC or Thermistor	1 – 2
JP12	1 – 2 2 – 3	Analogue Input 6: 0 – 20 mA Analogue Input 6: 0 – 10 VDC or Thermistor	1 – 2
JP13	1 – 2 2 – 3	Analogue Input 7: 0 – 20 mA Analogue Input 7: 0 – 10 VDC or Thermistor	1 – 2
JP14	1 – 2 2 – 3	Analogue Input 8: 0 – 20 mA Analogue Input 8: 0 – 10 VDC or Thermistor	1 – 2
JP15	1 – 2 2 – 3	Analogue Input 9: 0 – 20 mA Analogue Input 9: 0 – 10 VDC or Thermistor	1 – 2
JP16	1 – 2 2 – 3	Analogue Input 10: 0 – 20 mA Analogue Input 10: 0 – 10 VDC or Thermistor	1 – 2

(*) Fixed to 1 – 2 (0 – 10 VDC) for the pro•teso PC1000 with Ethernet support (OEM P/N 010-0232-001-E and 010-0232-001-EX).

5.2 DIGITAL INPUTS

The pro•teso PC1000 controller has 32 digital inputs, labelled as DI 01 to DI 32. The digital inputs comply with the standard IEC 61131-2 for Type 1 Digital Input Specification. Table 4 details the specification of the digital inputs.

Table 4: pro•teso PC1000 Digital Inputs Specifications

Number of Digital Inputs	32 (with transorb and inductor protection)
Number of Groups	4
Points per Group	8
Input Tolerance Range	-48 to +48 VDC
Minimum High Logic Level Input Voltage	+16.0 VDC
Maximum Low Logic Level Input Voltage	+14.0 VDC
Signal Types Supported	Each group of 8 digital inputs is configured via jumper settings (see Table 3) as: Pulled up internally to +24 VDC or pulled down internally to 0 VDC
Fast Electrical Transient Burst	Standard IEC 61000-4-4, Level 4 Criteria A 2 kV Voltage peak 5kHz Repetition Rate (As per BS EN50155 Section 10.2.7)

5.3 DIGITAL OUTPUTS

The pro•teso PC1000 controller has 28 digital outputs, labelled DO 01 – DO 28. Each output is capable of simultaneously sinking 1 A loads over the operating temperature range of the controller (-40 ºC to 70 ºC). Table 5 details the specification of the digital outputs.

Table 5: pro•teso PC1000 Digital Outputs Specifications

Number of Digital Outputs	28
Number of Groups	14
Points per Group	2
Output Type	35 V Low-Side N-Channel MOSFET driver output
Individual Load Output	Each driver can sink up to 1 A over the operating temperature range of the controller (-40 ºC to 70 ºC).
Fast Electrical Transient Burst	Standard IEC 61000-4-4, Level 4 Criteria A 2 kV Voltage peak 5kHz Repetition Rate (As per BS EN50155 Section 10.2.7)

It is important that all inductive loads connected to the digital outputs of the pro•teso PC1000 controller have an external back EMF diode, such as 1N4004, installed across the coil of the inductive device. This is industry practice. Failure to install back EMF diodes may result in damage or degradation of a PLC digital output due to the associated EMC generated when the inductive load is switched on and off.

The pro•teso PC1000 controller input power is wired via a 3-way, 7.5 mm pitch power connector. This connector has one terminal for 24 VDC-IN and two terminals for GND. Each terminal is rated at 15 A current. Due to the potentially large currents that can be sunk by the controller (up to 24 x 1 A) it is necessary for both GND terminals to be wired and appropriately rated cables used.

5.4 ANALOGUE INPUTS

Table 6: pro•teso PC1000 Analogue Inputs Specifications

Number of Digital Inputs	10 (with transorb and inductor protection)
Number of Groups	1
Points per Group	10
Signal Types Supported	Each point individually configured via jumper settings as (see Table 3): 0 to 10 VDC, 0 to 20 mA or NTC Thermistor (optimised for 2 kΩ resistance at 25 ºC). For controller with Ethernet support AI 01 – 04 are fixed to voltage type (0 to 10 VDC).
Resolution	12 bits
Current Input Capabilities	Range: 0 to 20 mA Absolute Maximum Current: 30 mA Input Impedance: 499 Ω
Voltage Input Capabilities	Range: 0 to 10 VDC Absolute Maximum Voltage: +15 VDC Input Impedance: > 1 MΩ
Fast Electrical Transient Burst	Standard IEC 61000-4-4, Level 4 Criteria A 2 kV Voltage peak 5kHz Repetition Rate (As per BS EN50155 Section 10.2.7)

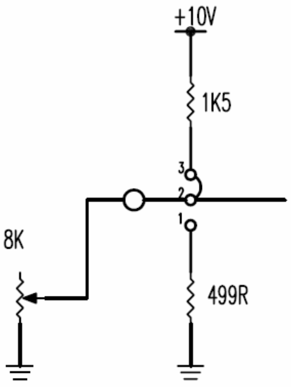


Figure 6: Analogue Input Configuration - Measuring Resistance

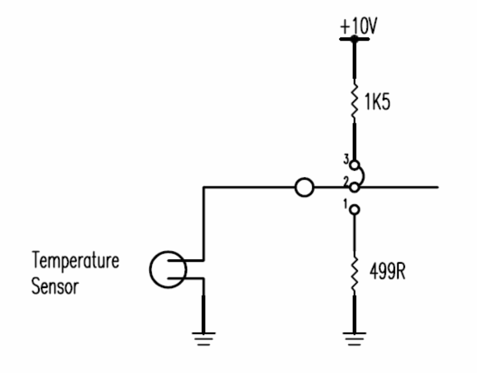


Figure 7: Analogue Input Configuration – Temperature Sensor (optimised for NTC Thermistors)

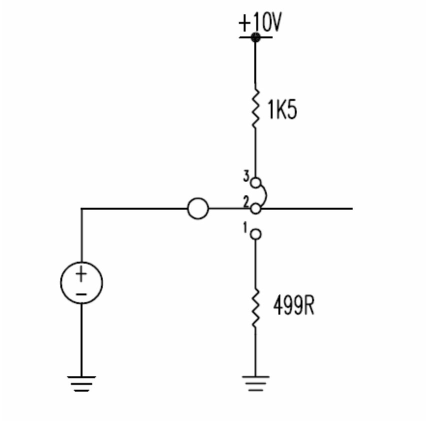


Figure 8: Analogue Input Configuration - Voltage Source

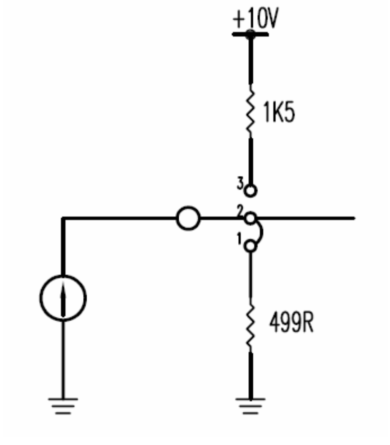


Figure 9: Analogue Input Configuration - Current Source

5.5 ANALOGUE OUTPUTS

Table 7: pro•teso PC1000 Analogue Outputs Specifications

Number of Analogue Outputs	2 (with transorb and inductor protection)
Number of Groups	1
Points per Group	2

Signal Types Supported	Each point individually configured via jumper settings as (see Table 3): 0 to 10 VDC or 0 to 20 mA
Resolution	10 bits
Current Input Capabilities	Range: 0 to 20 mA Maximum Load Impedance: 600 Ω
Voltage Input Capabilities	Range: 0 to 10 VDC Maximum Load Impedance: 20 kΩ
Fast Electrical Transient Burst	Standard IEC 61000-4-4, Level 4 Criteria A 2 kV Voltage peak 5kHz Repetition Rate (As per BS EN50155 Section 10.2.7)

5.6 COMMUNICATION INTERFACES

5.6.1 Serial Ports

The pro•teso PC1000 has two RS-232 (via DB9 female connector) and one RS-485 (via plug/socket connector) serial ports. In addition, there is a serial channel (Rabbit port) that is used to program and debug the controller using the programming cable, and to configure the target.

Table 8: pro•teso PC1000 Serial Port Configuration

Serial Port Number	Description
Rabbit Port	Programming port RS232 with handshaking. Connection type RJ45 female.
1	RS232 Full duplex 5 wire port with hardwire handshaking (RTS/CTS). Connection type DB9 female.
2	RS232 Full duplex 3 wire port. Connection type DB9 female.

✍ **Note:**

The pro•teso PC1000 can be configured (using EMBEDDED PLC Utility) to use either one RS232 (serial port number 1) or the RS485. The serial port chosen will be used as the communication channel with either the ISaGRAF Workbench or a Modbus master. If there is a special need to use serial port number 2, please contact OEM Technology Solutions.

5.6.1.1 RS-485

The pro•teso PC1000 comes with a 120 Ω termination resistor and two 560 Ω bias resistors. Jumpers J19 and J20 enable or disable the resistor according to the following table:

Table 9: RS485 Jumper Settings

Description	Jumper	Enabled	Disabled	Default
RS485 Terminating Resistor	J19:1 – J20:1	Install	Remove	Installed
RS485 Biasing Resistor	J19: 2 – 3 J20: 2 – 3	Install	Remove	Installed

6. SOFTWARE REFERENCE

6.1 MORE ABOUT ISAGRAF

6.1.1 Project Manager

The ISaGRAF methodology begins with the preparation of a detailed project specification. This step is common to all PLC programming techniques and is usually the result of a thorough analysis of the project and consultation between the integrator and the user.

Distributed applications can be represented as several ISaGRAF projects grouped together in the project list. Distributed applications can be edited, simulated or debugged on the same Windows screen.

6.1.2 Program Manager

With the program manager the application specification is divided into smaller functional modules. The exact operation that is to be performed within each module is defined at that stage.

ISaGRAF's program management facilities allow the user to define each of these modules, their operations and their interaction to form the complete application.

6.1.3 Variable Definition

All variables are declared or imported in the ISaGRAF dictionary. When this step is achieved, during programming, a mouse click will insert the variable in the program.

Any external database can be imported to build the ISaGRAF dictionary. A quick declaration allows many variables to be declared with one command and an easy to use mapping tool is provided for Modbus-based SCADA systems such as Indusoft®. For a brief description about Indusoft and how to use it in conjunction with EMBEDDED PLC see Chapter 10.

The EMBEDDED PLC Firmware Kernel implements three types of Modbus communication: Modbus over TCP/IP, Modbus RTU over RS232 and Modbus RTU over RS485. The EMBEDDED PLC Kernel can run only one protocol at one time.

✍ **Note:**

EMBEDDED PLC has the TCP/IP port 502 reserved (this port can not be modified) to support Modbus over TCP/IP communications. Since ISaGRAF Workbench uses a different TCP/IP port for communication with the Target PLC (port 1100), the Workbench and a Modbus master (HMI or SCADA) can communicate with the Target at the same time, i.e. when the Target is configured as Modbus over TCP/IP the ISaGRAF Workbench and a Modbus master can share the same physical line (Ethernet). However, when the Target is using either RS232 or RS485 only the ISaGRAF Workbench or a Modbus master can be connected to the target at any one time.

6.1.4 Functional Module Programming

The next step in the ISaGRAF methodology is the actual programming of the various functional modules. This can be done using any one of the six supported languages:

- Sequential Functional Chart (SFC)
- Function Block Diagram (FBD)
- Ladder Diagram (LD)
- Structured Text (ST)
- Instruction List (IL)
- Flow Chart (FC)

6.1.5 IEC 61131-3 Language Editors

In February 1993, responding to the need for standards to reduce training costs and guaranteed portability, the IEC issued the IEC 61131-3 standard: a specification of five PLC programming languages that can be freely mixed to define automation and control procedures.

In August 1996, ISaGRAF version 3.2 received the certificate of PLC open compliance class of IEC 61131-3 (base level IL).

6.1.5.1 Sequential Function Chart Editor

SFC divides the process cycle into a number of well defined steps, separated by transitions. SFC is the core language of the IEC 61131-3 standard. The other languages are used to describe the actions performed within the steps and the logical conditions for the transitions. Parallel processing can easily be described using SFC.

6.1.5.2 Function Block Diagram Editor

The FBD is a graphical language that allows the user to build complex procedures by taking existing function blocks from the ISaGRAF library and wiring them together on the screen.

ISaGRAF includes a library with more than 60 blocks ready to use. Users can enlarge this library by writing functions and function blocks in LD/FBD/ST/IL or C.

6.1.5.3 Ladder Diagram Editor

The ladder diagram is one of the most familiar methods of representing logical equations and simple actions. The ISaGRAF ladder diagram editor offers the best compromises between high level graphic capabilities and easy to use keyboard driven programming.

6.1.5.4 Structured Text Editor

Structured text is a high level structured language with a syntax similar to Pascal, but more intuitive to the automation engineer.

This language is mainly used to implement complex procedure that cannot be easily expressed with graphical languages (IF/THEN/ELSE, FOR, WHILE, etc.).

6.1.5.5 Instruction List Editor

Instruction list is a low level language similar to the simple textual PLC languages.

6.1.6 Flow Chart Editor

In addition to the five IEC 61131-3 languages ISaGRAF implements another graphical language, Flow Chart. Flow Chart is a decision diagram, which can also be used to describe sequential operations.

6.1.7 Document Generator

ISaGRAF features a self-documentation capability that can automatically generate the project's most current project description, project architecture, history of modifications, I/O wiring lists, dictionaries and cross references.

6.1.8 Simulation

Without any target hardware platform, the programmer can validate the complete application in the office. With ISaGRAF's powerful simulator on the workbench, the user can perform structural and functional tests of each module separately or on the global application.

The simulator makes it easy to trace the program execution and to see the status of any internal variable. The I/O hardware can be fully simulated and internal status and variables manual forced by the user.

During simulation, editors can be opened in debug mode to see how programs are executed.

> ✍ **Note:**
>
> Not all C functions and C function blocks are available for simulation. Section 6.2.2 and 6.2.3 describes C functions and C function blocks that are specific to the pro•teso PC1000 and are not available for simulation.

6.2 ISAGRAF SOFTWARE REFERENCE

This section presents information about the specific elements of the ISaGRAF Workbench Library for the EMBEDDED PLC pro•teso PC1000 (I/O boards, C functions and C function blocks).

6.2.1 I/O Boards

ISaGRAF I/O board library for the pro•teso PC1000 is included as part of the pro•teso PC1000 BSP to make it easy to communicate with the pro•teso PC1000 supported inputs and outputs. These I/O boards are described in the following sub-sections.

6.2.1.1 pro•teso PC1000 Digital Inputs Board

Name:

pc10di

Description:

Monitors DI 01 – DI 32 of the pro•teso PC1000 Digital Inputs

Number of Channels:

32

Connection Terminals:

DI 01 to DI 32 – Connection for positive signal channel 1 to 32

GND – Connection for negative signal all channels

Channel Value Correspondence:

TRUE – When digital input pulled up to greater than 16 V (V_{INHMin})

FALSE – When digital input pulled down to less than 14 V (V_{INLMax})

6.2.1.2 pro•teso PC1000 Digital Outputs Board

Name:

pc10do

Description:

Controls DO 01 – DO 28 of the pro•teso PC1000 digital sinking outputs

Number of Channels:

28

Connection Terminals:

DO 01 to DO 28 – Connection of channels 1 to 28

Channel Value Correspondence:

TRUE – Digital output is short-circuit to 0 V

FALSE – Digital output is open circuit

6.2.1.3 pro•teso PC1000 Digital Output Feedback

Name:

pc10dof

Description:

Monitors the pro•teso PC1000 28 Digital Output Feedback

Number of Channels:

28

Connection Terminals:

None

Channel Value Correspondence:

TRUE – Corresponding digital output channel is set to TRUE

FALSE – Corresponding digital output channel is set to FALSE

6.2.1.4 pro•teso PC1000 On-board LEDs

Name:

pc10led

Description:

Controls the pro•teso PC1000 on-board 2 dual colour LEDs (LED3 and LED4). Channel 1 corresponds to the GREEN LED3 and channel 2 corresponds to the RED LED3. Channel 3 corresponds to the GREEN LED4 and channel 4 corresponds to the RED LED4.

Number of Channels:

2

Connection Terminals:

None

Channel Value Correspondence:

TRUE – LED is ON

FALSE – LED is OFF

6.2.1.5 pro•teso PC1000 Analogue Input Board

Name:

pc10ai

Description:

Monitors AI 01 – AI 10 Analogue Inputs of the pro•teso PC1000 (12-bit resolution over 0 – 10 VDC or 0 – 20 mA range). Channels AI 11 and AI 12 are used to monitor the Input Power Supply and Analogue Reference Voltage respectively

Parameters:

Voltage ('V' or 'v') or current ('C' or 'c') type for each channel

Number of Channels:

12

Connection Terminals:

AI 01 – AI 10 – Connection for positive signal channel 1 to 10

AI GND – Connection for negative signal channel 1 to 10

Channel Value Correspondence:

Analogue REAL type variable: Voltage input in the range 0 – 10 VDC or Current input in the range 0 – 20 mA

6.2.1.6 pro•teso PC1000 Analogue Output Board

Name:

pc10ao

Description:

Controls AO 01 and AO 02 Analogue Outputs of the pro•teso PC1000 (10-bit resolution over 0 – 10 VDC or 0 – 20 mA).

Parameters:

Voltage ('V' or 'v') or current ('C' or 'c') type for each channel

Number of Channels:

2

Connection Terminals:

AO 01+ – AO 02+ – Connection for positive signal channel 1 and 2

AO 02- – AO 02- – Connection for negative signal channel 1 and 2

Channel Value Correspondence:

Analogue REAL type variable: Voltage input in the range 0 – 10 VDC or current 0 – 20 mA

6.2.1.7 pro•teso PC1000 Analogue Output Status Board

Name:

pc10aost

Description:

Monitors AO 01 and AO 02 Analogue Output status of the pro•teso PC1000.

Number of Channels:

2

Connection Terminals:

N/A

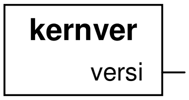
Channel Value Correspondence:

FALSE – Output channel is in NORMAL operation

TRUE – Output channel is shutdown due to high temperature

6.2.2 ISaGRAF C Functions

6.2.2.1 KERNVER



Arguments:

`versi`	MSG	EMBEDDED PLC kernel version number (minimum message length 8)

Description:

Returns the kernel version number.

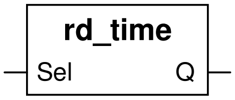
(* ST Equivalence *)

```
result := kernver();
```

Remarks:

Not available for simulation.

6.2.2.2 RD_TIME



Arguments:

`Sel`	INT	format of the time and date in the output message (Q)
`Q`	MSG	time, date and day of the week (depending of Sel value)
		If Sel = 0, Q format is YYYY/MM/DD;
		If Sel = 1, Q output format is HH:MM:SS
		If Sel = 2, Q output format is the day of the week

Description:

Gives the time, date and the day of the week depending on the value of argument sel.

(* ST Equivalence *)

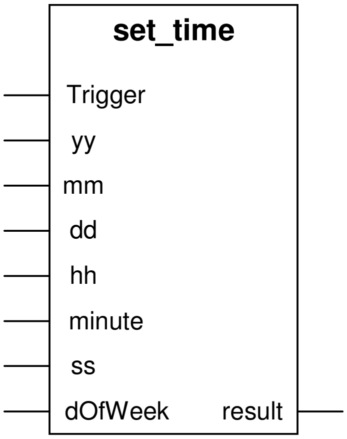
```
result := rd_time(sel);
```

Remarks:

Not available for simulation.

6.2.3 ISaGRAF C Function Blocks

6.2.3.1 SET_TIME



Arguments:

`Trigger`	BOO	if Rising edge, sets the time and date of the pro•teso PC1000
`yy`	INT	Year (1900 - 2050)
`mm`	INT	Month (1 - 12)
`dd`	INT	Day of month (1 -31)
`hh`	INT	Hour (0 - 23)
`minute`	INT	Minute (0 - 59)
`ss`	INT	Second (0 - 59)
`DOFWEEK`	INT	Day of the week (0 - 6, 0 means Sunday)
`result`	BOO	1 if operation was successful; 0 otherwise

Description:

Set the time and date of the pro•teso PC1000 Real Time Clock.

(* ST Equivalence *)

```
result := set_time(trigg, yy, mm, dd, hh, minute, ss, dOfWeek);
```

Remarks:

Not available for simulation.

This page is intentionally left blank

7. LOADING OR UPGRADING THE FIRMWARE KERNEL

7.1 INTRODUCTION

The EMBEDDED PLC Utility is used to load the EMBEDDED PLC Firmware Kernel to the Rabbit-based Target such as the pro•teso PC1000 controller. The Firmware Kernel, once loaded into the Target, requires a license key to run. The EMBEDDED PLC Utility does not check for an existing license key before loading the Kernel. Thus, a new license key is required after a Kernel is loaded to a Target.

Upgrading the EMBEDDED PLC Firmware Kernel means also loading the newer version of the Firmware kernel file into the Target. Therefore after loading a newer version of the Firmware Kernel file into a licensed Target PLC, a new license key will also be required.

Caution:

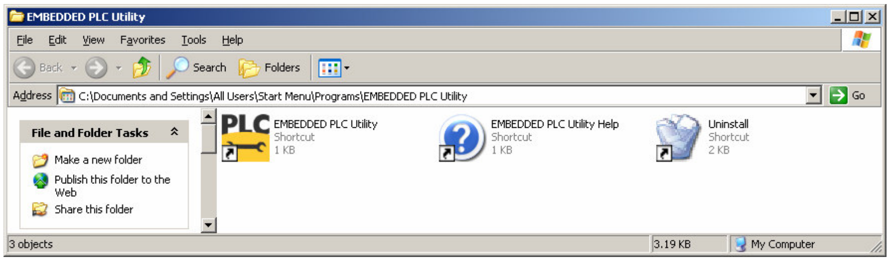
Loading of the Firmware Kernel file is **NOT** needed if you purchased the pro•teso PC1000 controller pre-loaded with the Firmware Kernel. If you re-load the Firmware Kernel to a pre-loaded Target, the license information will be lost and a new license key will be required.

This chapter describes how to install the EMBEDDED PLC Utility, load and license the EMBEDDED PLC Firmware Kernel. Also includes a complete reference of the Utility software.

7.1.1 Installing EMBEDDED PLC Utility

Use the following procedure to install EMBEDDED PLC Utility from the CD-ROM:

1. Insert the EMBEDDED PLC pro•teso PC1000 CD-ROM and open the root folder. Alternatively, if you download the zip file from the OEM Technology Solutions website, uncompress into a temporary folder.
2. Double-click on the `EmbeddedPLCUtility_V1.xx_3-0237x.msi` file to launch the EMBEDDED PLC Utility Setup Wizard. Follow the on screen instructions to proceed with the installation.
3. Once all the EMBEDDED PLC Utility files have been copied, the EMBEDDED PLC Utility icon is added to your Desktop and the following group is added to your main Programs group:



Address C:\Documents and Settings\All Users\Start Menu\Programs\EMBEDDED PLC Utility

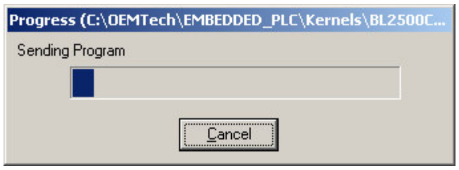
7.1.2 Connecting to the pro•teso PC1000 controller

1. Connect the programming cable (yellow cable) to the RJ-45 socket labelled as **Rabbit Port** (do not use the diagnostic connector, which is used for licensing and configuring only). Connect the other end of the programming cable to a COM port on your PC. Make a note of the port to which you connect the cable, as the EMBEDDED PLC Utility will need to have this parameter configured. Note that COM1 on the PC is the default COM port used by the EMBEDDED PLC Utility.
2. Connect the power supply to the pro•teso PC1000 controller.

7.1.3 Loading and licensing the Firmware Kernel

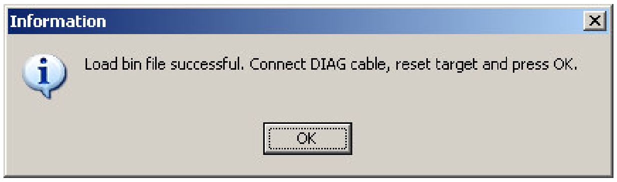
The EMBEDDED PLC pro•teso PC1000 Firmware Kernel file (`PC1000_V1.xx_3-0256x.bin`) is located in the `OEM-PC1000_BSP_V1.xx_9-0004-xxx\Kernels` directory of the CD-ROM (or in the ZIP file, if downloaded from the website). It is recommended to copy this file from the CD-ROM to a local PC folder, referred in this procedure as ***<EMBEDDED PLC Kernel Dir>***. Use the following procedure to load the EMBEDDED PLC Firmware Kernel to the Target:

3. Open the EMBEDDED PLC Utility and select **Setup** → **Communications…** from the main menu bar. Select the serial COM port where the programming cable is connected. Check the **Use USB to Serial converter** check-box if you have the RS232-to-USB converter cable. Press **OK**.
4. Select **File** → **Load Flash Image…** from the main menu bar. On **Choose Flash Image** window press **...** button. On the **Open** window select the EMBEDDED PLC Firmware Kernel file ***<EMBEDDED PLC Kernel Dir>*****\PC1000_V1.xx_3-0256x.bin** and press **Open**. Press **OK** on the **Choose Flash Image** window.



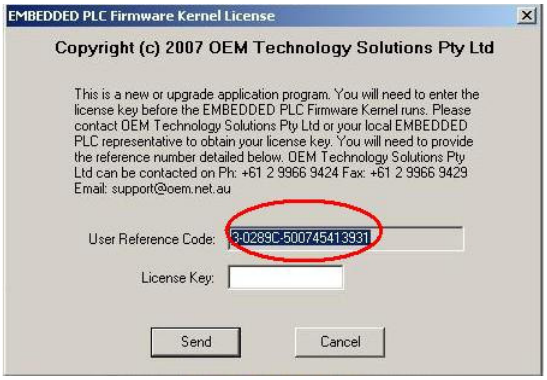
Sending Program

5. A **Progress** window is displayed with the status of the loading process. The process consists of four stages: Sending Coldloader → Sending Pilot BIOS → Erasing Flash → Sending Program.
6. After loading is completed a message dialog box is displayed. Follow the instructions in the same order: Connect **DIAG** cable → Reset Target → Press **OK**. To reset the target.



Load bin file successful. Connect DIAG cable, reset target and press OK.

7. On the **Kernel License** window, write down the User Reference Code (you can select the code in the User Reference Code and copy it to the clipboard using **Ctrl-C**). Contact OEM Technology Solutions or your local distributor to obtain a valid license key. Remember to quote the User Reference Code. Press **Cancel**. An error message dialog is displayed, press **OK** and close the EMBEDDED PLC Utility.



Copyright (c) 2007 OEM Technology Solutions Pty Ltd

This is a new or upgrade application program. You will need to enter the license key before the EMBEDDED PLC Firmware Kernel runs. Please contact OEM Technology Solutions Pty Ltd or your local EMBEDDED PLC representative to obtain your license key. You will need to provide the reference number detailed below. OEM Technology Solutions Pty Ltd can be contacted on Ph: +61 2 9966 9424 Fax: +61 2 9966 9429 Email: support@oem.net.au

User Reference Code: 3-0289C-500745413931

License Key:

✍ **Note:**

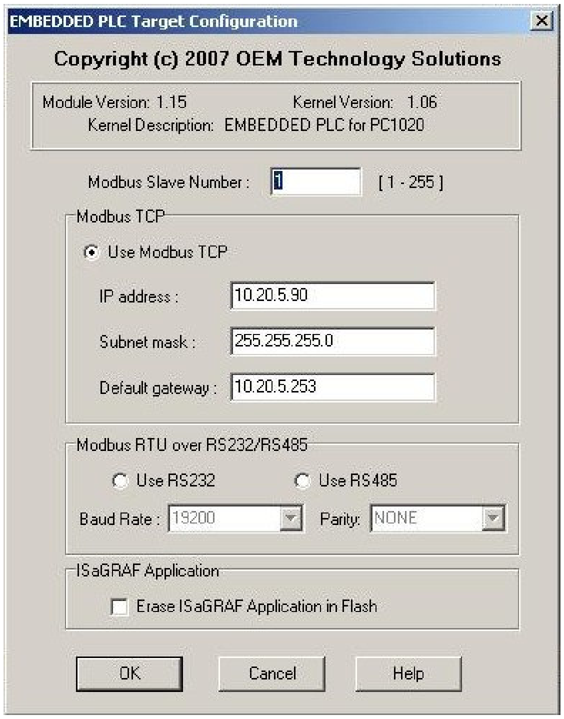
Even if your Target is not active (without valid license key) you can run several samples in the ISaGRAF Workbench in simulation mode. To run those sample files go to Section 0.

8. Once you have a valid license key, open the EMBEDDED PLC Utility. Select **Setup** → **Kernel License…** Follow the instructions on the message dialog box in the same order: Press **OK** → Reset Target. After a few seconds the License Window will be displayed.
9. Enter the valid license key in the **License Key** entry box and press **Send**. A confirmation message is displayed. Save the User Reference Code and License Key for future reference and technical support. Press **OK**.

7.1.4 Configuring the Firmware Kernel

The pro•teso PC1000 controller comes pre-configured for ***Use Modbus RTU/RS-232*** and the following serial parameters: Baud rate = 19200. If you want to check or change the Target PLC configuration, follow these instructions:

1. Open the EMBEDDED PLC Utility and select **Setup** → **Target Configuration…** Follow the instructions on the message dialog box in the same order: Press **OK** → Reset Target. After a few seconds the **Target Configuration** Window is displayed.



EMBEDDED PLC Target Configuration

Copyright (c) 2007 OEM Technology Solutions

Module Version: 1.15 Kernel Version: 1.06
Kernel Description: EMBEDDED PLC for PC1020

Modbus Slave Number : 1 [1 - 255]

Modbus TCP
● Use Modbus TCP
IP address : 10.20.5.90
Subnet mask : 255.255.255.0
Default gateway : 10.20.5.253

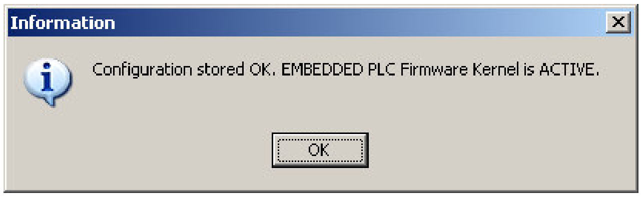
Modbus RTU over RS232/RS485
○ Use RS232 ○ Use RS485
Baud Rate : 19200 Parity: NONE

ISaGRAF Application
☐ Erase ISaGRAF Application in Flash

2. Enter the desired Modbus Slave Number (Default is 1). Select **Use Modbus TCP** radio button and fill the Modbus TCP configuration parameters if you want to use Modbus TCP as communication protocol. Otherwise select **Use RS232** or **Use RS485** radio buttons. Once all the configuration parameters have been entered press **OK**. The communication protocol and slave number are used to communicate the Target PLC with the ISaGRAF Workbench and Modbus master.

> ✍ **Note:**
>
> The Modbus Slave Number parameter corresponds to the Target Slave Number on ISaGRAF Workbench (Default is 1). In order to communicate the ISaGRAF Workbench with a specific Target PLC, both numbers must be the same. This is also valid for communication between the Target PLC and a Modbus master using Modbus protocol.

3. After pressing **OK** the following message dialog box is shown. Press **OK** and your EMBEDDED PLC Firmware Kernel will be active and ready to be used with the ISaGRAF Workbench.



Information

Configuration stored OK. EMBEDDED PLC Firmware Kernel is ACTIVE.

7.2 EMBEDDED PLC UTILITY SOFTWARE REFERENCE

To use the EMBEDDED PLC Utility you must have an EMBEDDED PLC Firmware Kernel (.bin file) to load to your Rabbit-based controller. Click on **File** → **Load Flash Image**; enter the .bin file's location in the dialog box and click **OK**.

This Utility works only with serial connection and it requires the programming cable for the Rabbit-based controller (1.27mm programming cable for Rabbit 3000 microprocessor based controllers or a yellow programming cable for the pro•teso PC1000 controllers).

This Utility downloads ONLY EMBEDDED PLC Firmware Kernels. If you want to download another binary file generated with Dynamic C use the Rabbit Field Utility (RFU) that comes in the Dynamic C installation.

7.2.1 Menu Commands

7.2.1.1 File → Load Flash Image

Allows a EMBEDDED PLC Kernel (.bin file) to be chosen as the file to be loaded. Clicking the ellipses button (**…**) brings up an Open File dialog box to browse for a file's location.

7.2.1.2 Setup → Communications

This version of the EMBEDDED PLC Utility only supports Serial Connection.

The default com port is COM1.

The default baud rate is 115200 bps.

If the host PC is using a different com port, change the default to match the port being used by the PC. If the BIOS is using a different baud rate, change the default to match the baud rate being used by the BIOS.

7.2.1.3 Setup → File Locations

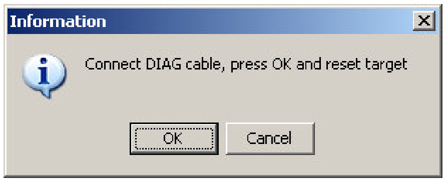
The first time EMBEDDED PLC Utility runs, it assumes that `coldload.bin`, `pilot.bin` and `flash.ini` are located in the EMBEDDED PLC Utility subdirectory. Those files are installed by the EMBEDDED PLC installation process.

The Rabbit Field Utility can be run from a different location if the location of `coldload.bin` and `pilot.bin` are specified using **Setup** → **File Locations**.

7.2.1.4 Setup → Target Configuration

The configuration parameters can be changed at any time once the EMBEDDED PLC kernel is running but requires the Target to be reset (or powered cycle).

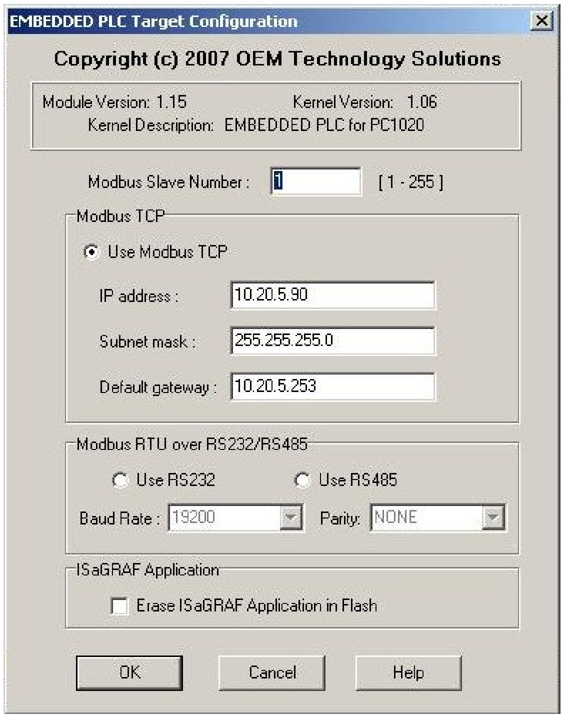
Select **Setup** → **Target Configuration…** Follow the instructions that appear in the message dialog box and press **OK**.



Information

Connect DIAG cable, press OK and reset target

The EMBEDDED PLC Utility will establish connection with the target to get the configuration parameters. If connection is established the following window will appear.



EMBEDDED PLC Target Configuration

Copyright (c) 2007 OEM Technology Solutions

Module Version: 1.15 Kernel Version: 1.06
Kernel Description: EMBEDDED PLC for PC1020

Modbus Slave Number: 1 [1 - 255]

Modbus TCP
● Use Modbus TCP
IP address: 10.20.5.90
Subnet mask: 255.255.255.0
Default gateway: 10.20.5.253

Modbus RTU over RS232/RS485
○ Use RS232 ○ Use RS485
Baud Rate: 19200 Parity: NONE

ISaGRAF Application
☐ Erase ISaGRAF Application in Flash

Enter the desired configuration and press **OK**.

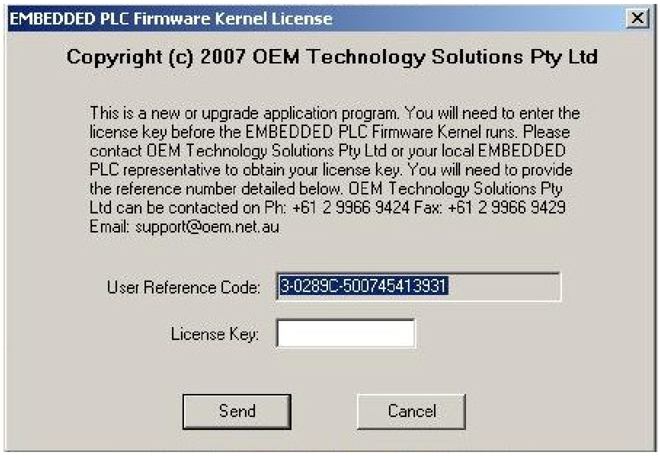
Configuration parameters:

- ***Modbus Slave Number***: This is slave number used to communicate with the ISaGRAF Workbench and a Modbus Master.
- ***IP Address***: IP address of the target (must be static, DHCP is not supported).
- ***Netmask***: Netmasks are used to identify which part of the address is the Network ID and which part is the Host ID. For example: 255.255.255.0.
- ***Default Gateway***: IP address of the default gateway.
- ***Use Modbus TCP***: If this option is selected, the Modbus RTU over RS232/RS485 box is disabled. Select this option if Modbus TCP is required.
- ***Use RS232***: If this option is selected, the Modbus TCP configuration box is disabled. Select this option if Modbus RTU over RS232 is required.
- ***Use RS485***: If this option is selected, the Modbus TCP configuration box is disabled. Select this option if Modbus RTU over RS485 is required.
- ***Baud Rate***: Select between 600 to 19200 bits per second.
- ***Parity:*** Select the parity type NONE, EVEN or ODD. If, *Use RS232 / RS485* option selected
- ***Erase ISaGRAF Application in Flash:*** If this option is selected, any previous ISaGRAF application in Flash memory of the controller will be erased.

This operation can be repeated at any time but does require resetting (or power cycle) the Target.

7.2.1.5 Setup → Kernel License

This option can be used to establish connection with the target to check the current license status. If license is required the following window will appear.



Copyright (c) 2007 OEM Technology Solutions Pty Ltd

This is a new or upgrade application program. You will need to enter the license key before the EMBEDDED PLC Firmware Kernel runs. Please contact OEM Technology Solutions Pty Ltd or your local EMBEDDED PLC representative to obtain your license key. You will need to provide the reference number detailed below. OEM Technology Solutions Pty Ltd can be contacted on Ph: +61 2 9966 9424 Fax: +61 2 9966 9429 Email: support@oem.net.au

User Reference Code: 3-0289C-500745413931

License Key:

The User Reference Code can be copied into the Clipboard by double-clicking on the box (this will select the whole text) and pressing **Ctrl-C**. The text can be pasted in any document.

Enter the valid license key in the entry box and press **Send**. If license is valid a message confirming the validation will appear and the EMBEDDED PLC will be active.

If license is already valid, a message dialog box will show the current Reference Code and License Key. Keep this numbers for your records and for customer support.

7.2.1.6 Setup → Utility Settings

The **Target Communication Timeout** corresponds to the maximum amount of time waiting for a message coming from the EMBEDDED PLC target during licensing or configuring the kernel. The value is expressed in seconds and has to be an integer. Minimum value is 1 seconds.

7.2.2 Loading Shortcuts

7.2.2.1 Right Mouse Click

Clicking the right mouse button anywhere on the main window will bring up a list of the most recently loaded programs. Selecting any of the programs in the list will automatically load that program to the attached controller.

7.2.2.2 Drag and Drop

Once the EMBEDDED PLC Utility is open, if a file is dragged from Windows Explorer, the Desktop, etc, and dropped into the main window, the file will be automatically loaded to the controller.

8. FAQ AND TROUBLESHOOTING

8.1 KNOWN ISSUES

8.1.1 ISaGRAF Floating Point Arithmetic

The floating point arithmetic implemented by Dynamic C for the Rabbit 3000 microprocessor is not 100% compliant with the IEEE Standard. When doing arithmetic operations with numbers close to the upper limit (3.4E38) instead of causing an overflow the result goes beyond the limits set by the standard. External applications, such as ISaGRAF Debugger, monitoring those values will fail because those values are invalid floating point (NaN).

When this condition happens, the ISaGRAF Debugger stops but leaves some processes running. If you experience this problem, close all ISaGRAF applications (including the Project Management). Re open the Project Management to resume ISaGRAF development.

8.1.2 Reading/writing 32-bit integer/real analogue variables via Modbus

EMBEDDED PLC supports reading and/or writing 32-bit integer/real analogue variable through Modbus. Requests are handled by using two consecutive words (multiple registers) in little endian format, i.e. the less significant word goes first. The constraint is the Modbus address for analogue variables must be odd number.

8.2 FAQ

Q1. I already have a version of ISaGRAF Workbench V3.5 installed and licensed. What should I do or install to make it EMBEDDED PLC compatible?

If you already have an ISaGRAF Workbench V3.50 running on you Host PC, use the following instructions to make your ISaGRAF Workbench compatible with EMBEDDED PLC:

(a) Download the **`ISaGRAF_V3.55_3-0239x.zip`** file from OEM Technology Solutions website (http://www.oem.net.au).

(b) Extract all files to a local directory ***<ISaGRAF Install Dir>***.

(c) Copy <*ISaGRAF Install Dir*>\EXE\RABBIT.RSC and <*ISaGRAF Install Dir*>\EXE\ISATXT.ERR to where your ISaGRAF main directory resides <*ISaGRAF Main Dir*>\EXE (for example, C:\ISAWIN\EXE). Confirm yes at overwriting ISATXT.ERR file.

(d) Edit **`<ISaGRAF Main Dir>\EXE\RSCDEF.RSC`** and add the following line:

```
Resource=RABBIT.rsc
```

This line will add the "RABBIT: TIC code for Rabbit-based controllers" in the Targets list in the Compiler Options window of the ISaGRAF Programs Manager. You will have to recompile all your programs against this TIC code type in order to work with the EMBEDDED PLC pro•teso PC1000.

Q2. What happens if I try to download the EMBEDDED PLC firmware kernel to a Target already running a run-time license of EMBEDDED PLC?

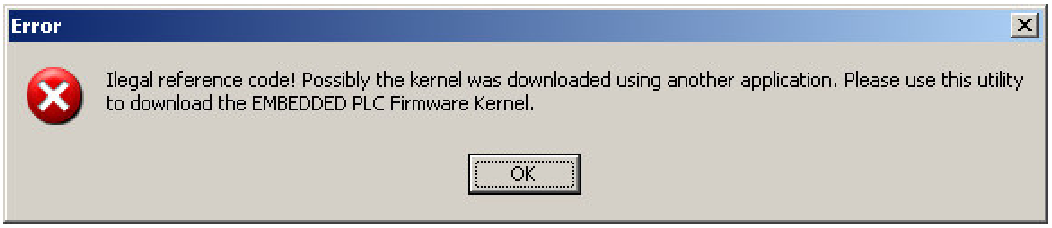
The EMBEDDED PLC Utility does not check for an existing license before performing a download and a new license is required after a Kernel is loaded to a Target. This means that after downloading a new Kernel to a Target, a new User Reference Code is generated and you will need to contact OEM Technology Solutions or your local distributor to obtain a new license key to activate the EMBEDDED PLC Kernel.

Q3. Can I use the EMBEDDED PLC Utility to download binary files compiled with Dynamic C?

No. The EMBEDDED PLC Utility works only with EMBEDDED PLC firmware kernels. Use Rabbit Field Utility instead.

Q4. Can I use the Rabbit Field Utility to download the EMBEDDED PLC Firmware Kernel to the Target?

No. If you do so, when you tried to enter the run-time license using the EMBEDDED PLC Utility the following message shall appear:



Ilegal reference code! Possibly the kernel was downloaded using another application. Please use this utility to download the EMBEDDED PLC Firmware Kernel.

Q5. Which are the Modbus function codes supported by EMBEDDED PLC?

The EMBEDDED PLC pro•teso PC1000 supports the following Modbus function code:

1	Read coils
2	Read discrete inputs
3	Read holding registers
4	Read input registers
5	Write single coil
6	Write single register
15	Write multiple coils
16	Write multiple registers

Q6. How can I change the assigned serial port number to the RS-232-to-USB converter cable on Windows XP?

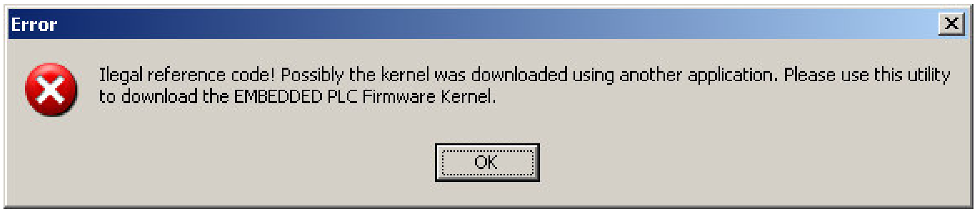
Open Control Panel → System. Click on **Device Manager**. On the Device Manager window expand the Ports (COM & LPT) category and select the **USB Serial Port (COMx)**. Select **Action** → **Properties** on the menu bar. On the USB Serial Port (COMx) Properties window click on **Port Setting** tab and click on **Advanced…** On the Advanced window change the COM port number to within COM1 to COM4 range. For more information about device settings contact your local system administrator.

Q7. Can I use the other serial ports on the pro•teso PC1000 from an ISaGRAF application?

The EMBEDDED PLC Firmware Kernel uses one RS232 (serial channel 1) or one RS485 (port 2) to communicate with the ISaGRAF Workbench or Modbus master. Contact OEM Technology Solutions or your local EMBEDDED PLC distributor for information about custom solutions for the EMBEDDED PLC Kernel.

Q8. Do I need to keep the programming cable connected to the Target all the time?

No. The programming DIAG cable is used only for entering license key or Target configuration parameters values. After that it can be disconnected.



Ilegal reference code! Possibly the kernel was downloaded using another application. Please use this utility to download the EMBEDDED PLC Firmware Kernel.

8.3 TROUBLESHOOTING

8.3.1 Common Errors

Listed below are some solutions of common errors that you may encounter with EMBEDDED PLC products.

8.3.1.1 Communications

If you experienced problems with the communication between the Target and either the ISaGRAF Workbench or a Modbus master check the items below:

- Disconnect all the physical I/Os connected to the Target and try again.
- Check that the Modbus Slave Number value in the Target is the same as the Target Slave Number in the ISaGRAF Workbench.
- If you are using Ethernet link, check that IP address, Netmask and Default Gateway values are correct in the Target. Check also that the port in the ISaGRAF Workbench is set to 1100.
- If you are using RS232 or RS485 link remember that either the ISaGRAF Workbench or the Modbus master can be connected to the Target at a certain time. Check that the serial port parameters on the ISaGRAF Workbench and/or Modbus master are the following: Baud rate (19200 or 9600, depending on the Target configuration), no parity, 8 data bits, 1 stop bits and no flow control. If you are using a RS232-to-USB converter cable check that the COM port assigned to it is within COM1 to COM4 range. If not follow the instructions described in Q8 of the FAQ section.
- The minimum value of the polling period from ISaGRAF Workbench or a Modbus master must be greater than the PLC cycle execution time. In any case the absolute minimum values for the polling period must be greater than the following:

  Modbus over TCP/IP: 50 ms

  Modbus over RS232 or RS485: 200 ms

8.3.1.2 ISaGRAF Workbench for EMBEDDED PLC

If you experienced problems downloading an application to the Target check the items below:

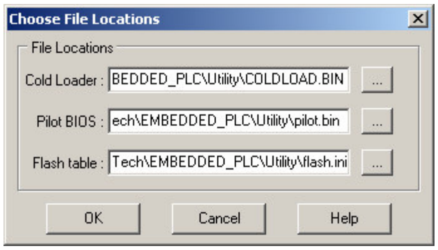
- Check that you have communication with the Target (check items above).
- Every new project created in ISaGRAF will have only the SIMULATE compiler option selected. In order to work with EMBEDDED PLC make sure that the "RABBIT: TIC code for Rabbit-based controllers" is selected in the Compiler options list before making and downloading the application.

8.3.1.3 EMBEDDED PLC Utility

Check the items below if you have problems downloading the EMBEDDED PLC Firmware Kernel:

- Check that the programming cable is connected through DIAG connection to the Target.
- Check that no other application is using the selected COM port.
- Check the location of the cold loader, pilot BIOS and Flash initialization files. Select **Setup** → **File Locations…** from the main menu bar. Check that the Cold loader (`COLDLOAD.bin`), Pilot BIOS

(`pilot.bin`) and Flash table (`flash.ini`) files locations are correct. It should point to where the EMBEDDED PLC Utility is installed (default `C:\OEMTech\EMBEDDED PLC\Utility`). Press **OK**.


Choose File Locations

File Locations

Cold Loader : BEDDED_PLC\Utility\COLDLOAD.BIN

Pilot BIOS : ech\EMBEDDED_PLC\Utility\pilot.bin

Flash table : Tech\EMBEDDED_PLC\Utility\flash.ini

8.3.2 Before Contacting Technical Support

Some things you should try before you contact Technical Support are:

- Check out the documentation.
- Consider recent changes on your system. If something used to work, think about what may have changed. New software installation or general system changes can affect performance and general functionality of other software on your system.
- Verify that the entire system has good, low-impedance, separate grounds for analogue and digital signals. The pro•teso PC1000 is often connected between the host PC and another device. Any differences in ground potential can cause serious problems that are hard to diagnose.
- Do not connect analogue ground to digital ground anywhere.
- Verify that the power supply has enough capacity and filtering to support the pro•teso PC1000.

If you cannot find an answer to your technical question in the product documentation or help system, send your request to OEM Technology Solutions technical support via email support@oem.net.au.

Please try to define the problem before you contact Technical Support so that you can repeat the steps that led to the problem and specifically identify when and how the problem occurred. The Technical Support representative will need to know exactly what the problem is in order to provide help. These steps will help us pinpoint and solve your problem more quickly.

Please have the following information available:

- Hardware environment: Target details (pro•teso PC1000 model), Host PC details and a brief description of the physical I/Os used.
- Software environment: Windows Operating System.
- EMBEDDED PLC filenames used.
- EMBEDDED PLC Kernel version: You can obtain this information from the Target Configuration Window (Module Version, Kernel Version, Kernel Description).
- EMBEDDED PLC license information.
- Description of the problem or error, including specific text or error messages.
- Steps to reproduce the problem, if it is reproducible.

9. SPECIFICATIONS

9.1 ELECTRICAL AND MECHANICAL SPECIFICATIONS

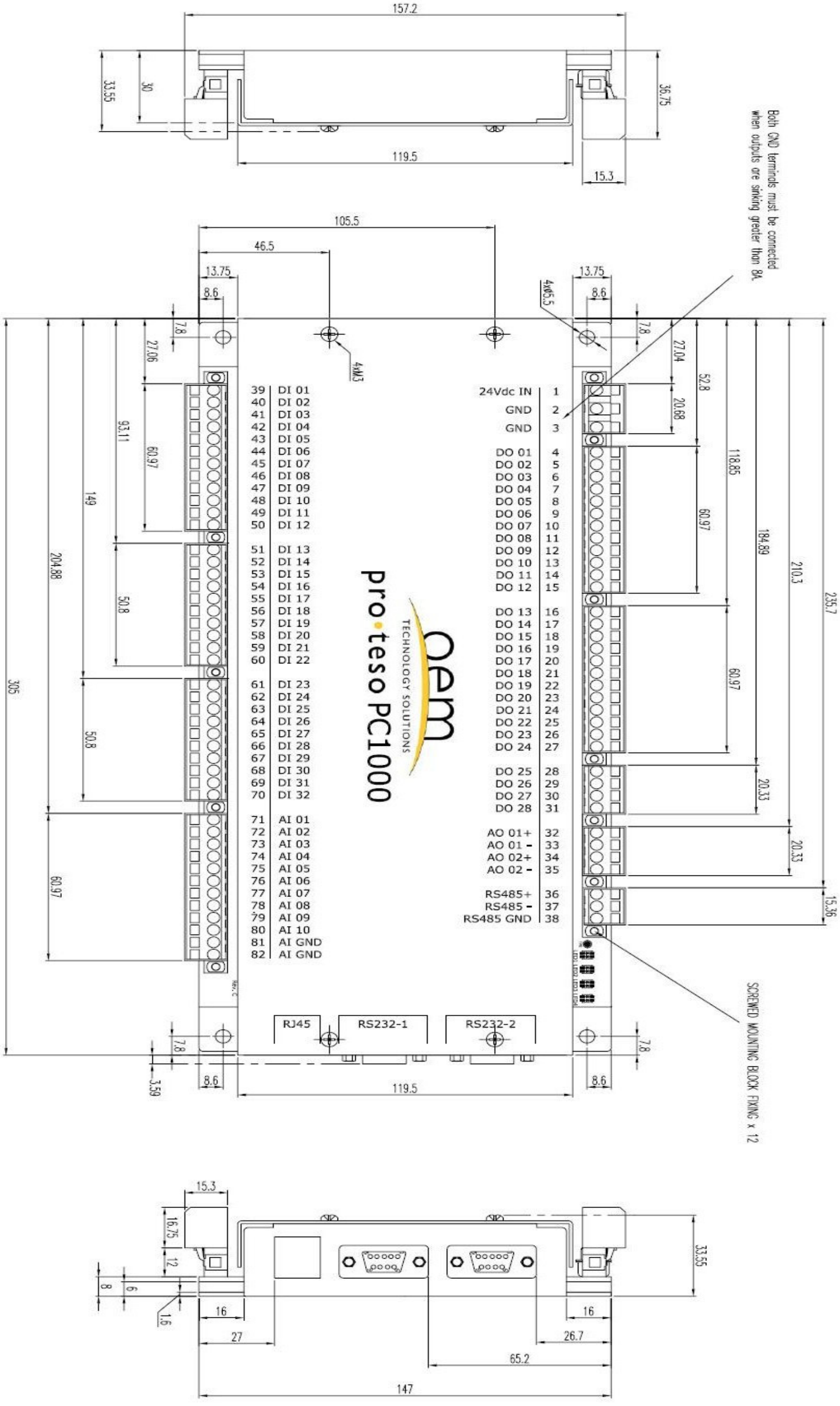


Figure 10: pro•teso PC1000 Dimensions

Table 10: pro•teso PC1000 Electrical, Mechanical and Environmental Specifications

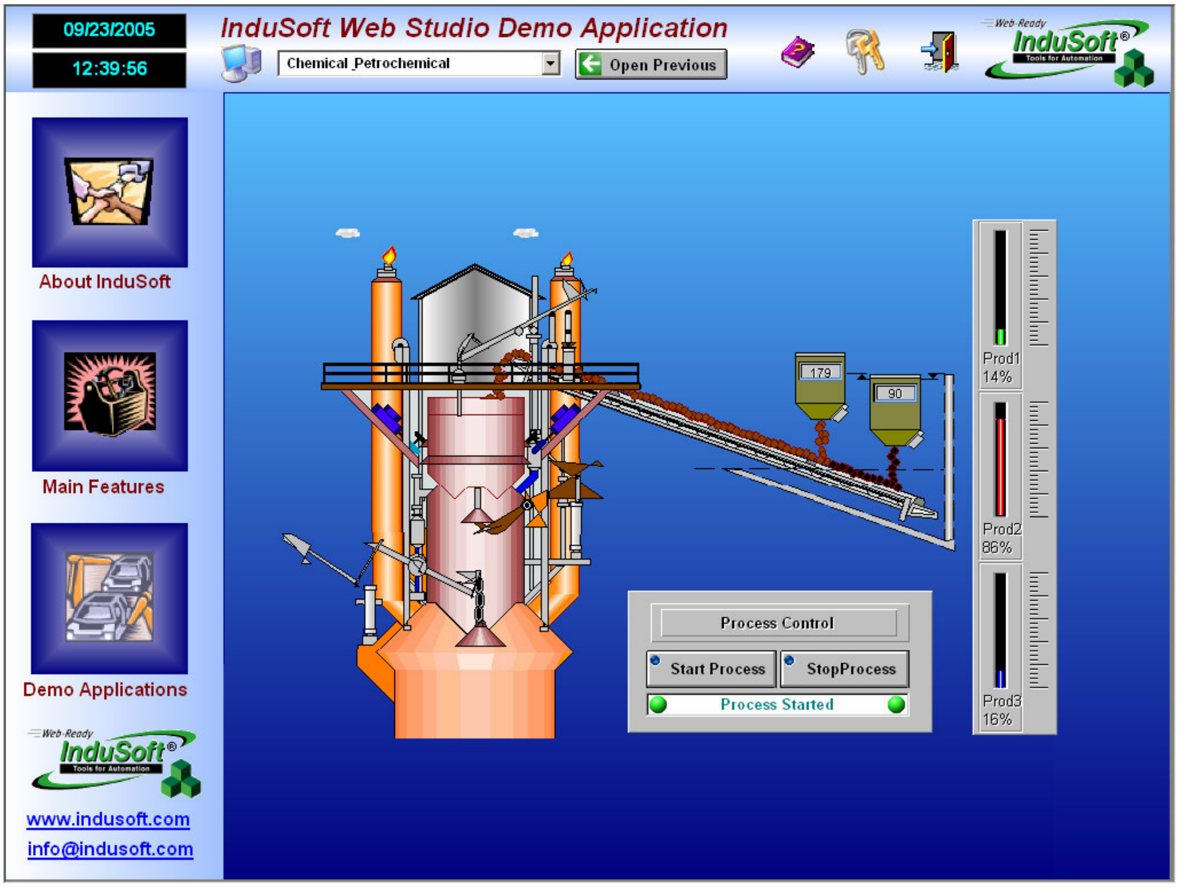
Description	PC1000	PC1010	PC1020	PC1030
Input Voltage	+18 to +32 VDC (Nominal 24 VDC at 250mA)			
Environmental	Operational -40ºC to +70ºC 5% to 95% Relative Humidity			
Processor Speed	29.4 MHz	44.2 MHz		
SRAM Memory	512 kB Data	512 kB program and 256 kB data	512 kB program and 512 kB data	
Flash Memory	(2 x 256 kB)	512 kB		
Serial Flash Memory	None		16 MB NAND Flash	None
Removable Memory	None		xD card up to 128 MB	
Backup Battery	Data RAM and RTC battery backed via 950mAH battery			
Real-Time Clock	Yes			
Watchdog	CPU hardware watchdog			
Digital I/O	32 inputs: Configurable in banks of eight as pull up to 24 V or pull down to 0 V. Input tolerance -48 to 48 VDC 28 outputs: 35 V Low-Side N-Channel MOSFET driver output. Each output can sink 1 A at 24 VDC over the temperature range -40℃ to +70℃			
Analogue Inputs	10 12-bit resolution. Each channel individually configurable as 0 to 10 VDC, 0 to 20 mA or NTC Thermistor (optimised for 2 kΩ resistance at 25 ℃)			
Analogue Outputs	Two 10-bit resolution. Each channel individually configurable as 0 to 10 VDC or 0 to 20 mA			
Serial Ports	3 serial ports[3]: - 1 x 2-wire RS-485 port via plug/socket connector - 2 x 5-wire RS-232 via DB-9 connectors (channel 1 and 2)			
Ethernet	None	One (1) – Ethernet port 10/100 Mbps via RJ-45 socket		
Protocols	Modbus RTU Serial (Slave)	Modbus RTU Serial and TCP/IP (Master and Slave)		
Terminations	Weidmuller plug/socket cage clamp connectors with V0 flammability rating. Cage clamping range maximum 1.5 mm^2 machine tool (MTW)			
Enclosure	Anodised Aluminium 305 mm x 147 mm x 34 mm (without plug connectors)			
Compliance	EN50155:1996 – Compliance with Visual Inspection, Performance Test, Cooling Test, Dry Test, Supply Related Surge and Transient Susceptibility Test, Transient Burst Susceptibility Test and Vibration, Shock and Bump Test			
EMC Testing	Fast Electrical Transient Burst. All Digital I/Os, Analogue I/Os and Serial Ports comply with IEC1000-4-4, Level 4 Criteria A. 2 kV voltage peak 5 kHz repetition rate (as per EN50155 Section 10.2.7)			
I/O Expansion	Analogue and digital I/O expansion modules connected via the RS485 port			
Part Number	010-0232-001	010-0232-002	010-0232-003	010-0232-004

---

[3] Only 1 RS232 (serial channel 1) and the RS485 serial ports are supported in EMBEDDED PLC at this present time.

10. ABOUT INDUSOFT

Indusoft® Web Studio (IWS) is a powerful, integrated collection of automation tools that include all the building blocks needed to develop modern HMI, SCADA, instrumentation and embedded applications running on native Microsoft Windows XP, 2000, NT, CE and CE.NET or in Internet and intranet environments. Indusoft Web Studio provides a simple drag and drop, point and click development environment which mimics the most complex behaviours of you live processes. Indusoft Web Studio is ideal e-automation solution for industry.



IWS applications consist of animated operator-interface screens, communication drivers (for PLCs and I/O protocols such as Modbus, DFS, Profibus, etc.), tags database, and additional modules such as alarm monitors, logic, trend charts, recipes, schedulers and security system.

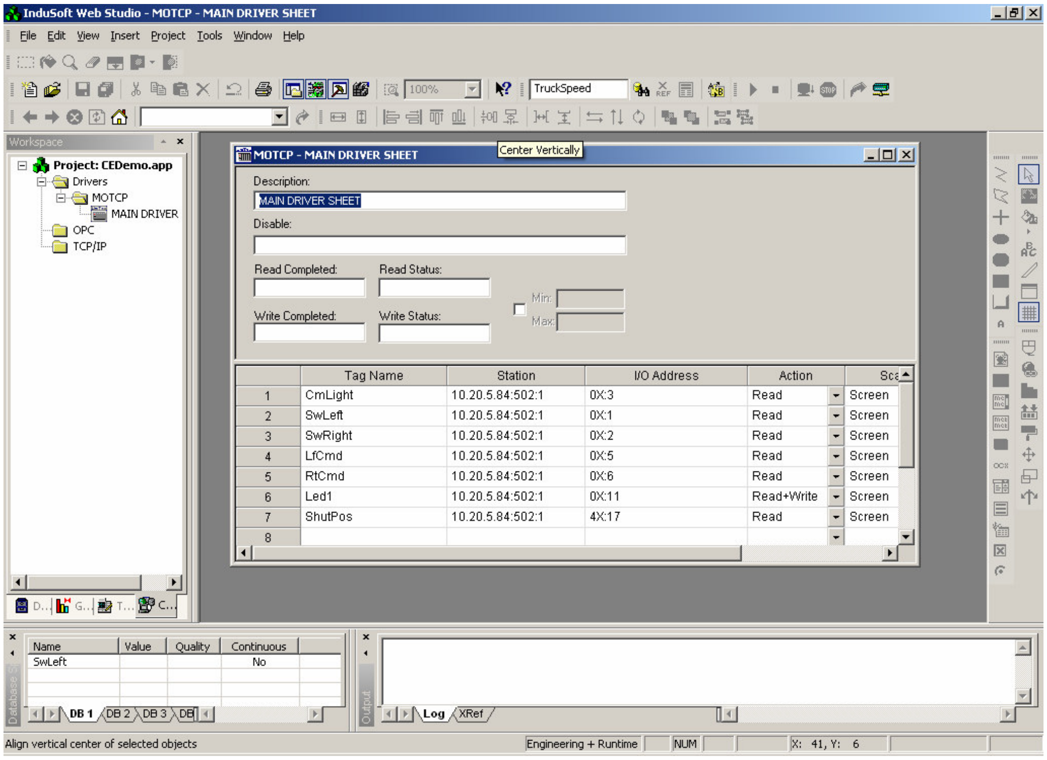
IWS applications interface with industrial I/O systems and other Windows applications in the run-time environment using ODBC, DDE, NetDDE, OPC or TCP/IP protocols.

For more information about IWS contact OEM Technology Solutions (http://www.oem.net.au).

10.1 USING INDUSOFT WEB STUDIO WITH EMBEDDED PLC

The EMBEDDED PLC implements Modbus over TCP/IP or RTU over RS232/RS485 to communicate with Modbus masters such as Indusoft Web Studio.

Indusoft Web Studio (IWS) applications can be quickly developed to be used in conjunction with EMBEDDED PLC since it has built-in communication drivers for Modbus.



Description: MAIN DRIVER SHEET

Disable:

Read Completed:

Read Status:

Write Completed:

Write Status:

☐ Min: Max:

	Tag Name	Station	I/O Address	Action	Sca
1	CmLight	10.20.5.84:502:1	0X:3	Read	Screen
2	SwLeft	10.20.5.84:502:1	0X:1	Read	Screen
3	SwRight	10.20.5.84:502:1	0X:2	Read	Screen
4	LfCmd	10.20.5.84:502:1	0X:5	Read	Screen
5	RtCmd	10.20.5.84:502:1	0X:6	Read	Screen
6	Led1	10.20.5.84:502:1	0X:11	Read+Write	Screen
7	ShutPos	10.20.5.84:502:1	4X:17	Read	Screen
8					

Name	Value	Quality	Continuous
SwLeft			No

Align vertical center of selected objects

Engineering + Runtime NUM X: 41, Y: 6

You can quickly add the Modbus driver on the Workspace, define its operational parameters and define the I/O variables (tags) that will be exchanged between the EMBEDDED PLC and the IWS application.