TelePACE

Ladder Logic Training Manual

CONTROL
MICROSYSTEMS
SCADA products... for the distance

TelePACE Ladder Editor Training Manual

Printed in Canada.

Trademarks

TeleSAFE, TelePACE, TeleBUS, SmartWIRE and SCADAPack are registered trademarks of Control Microsystems Inc.

All other product names are copyright and registered trademarks or trade names of their respective owners.

Version 5.0.2006
2/1/2007

System Configuration Guide

CONTROL MICROSYSTEMS

SCADA products … for the distance

48 Steacie Drive	Telephone:	(613) 591-1943
Kanata, Ontario	Fascimile:	(613) 591-1022
K2K 2A9	Technical Support:	1- 888-226-6876
Canada	1-888-2CONTROL	

System Configuration Guide

Trademarks

TeleSAFE, TelePACE, SmartWIRE, SCADAPack, TeleSAFE Micro16 and TeleBUS are registered trademarks of Control Microsystems Inc.

All other product names are copyright and registered trademarks or trade names of their respective owners.

Material used in the User and Reference manual section titled SCADAServer OLE Automation Reference is distributed under license from the OPC Foundation.

Table of Contents

Index of Figures

Overview

This manual is a system configuration guide for SCADA, telemetry and control applications using the SCADAPack, TeleSAFE Micro16, or SmartWIRE controllers and 5000 Series I/O modules.

The configuration of a system begins with an understanding of the application requirements of the system. These requirements include; the programming capability of the controller; the types of serial communication needed; the number and type of I/O needed and power supply requirements. After determining the system requirements the selection of controller and 5000 Series I/O modules are made.

To aid with the selection of modules this manual contains an overview of the specifications of each type of controller and 5000 Series I/O module available. A section describing how to calculate system power requirements provides the information for selecting power supplies.

The installation section of this manual provides information for mounting and connecting the controller and 5000 Series modules. Diagrams of various types of system configurations are included to provide examples of correct system layout.

Important Safety Information

Power, input and output (i/o) wiring must be in accordance with Class I, Division 2 wiring methods Article 501-4 (b) of the National Electrical Code, NFPA 70 for installations in the U.S., or as specified in Section 18-1J2 of the Canadian Electrical Code for installations within Canada and in accordance with the authority having jurisdiction.

WARNING !
EXPLOSION HAZARD - SUBSTITUTION OF COMPONENTS MAY IMPAIR SUITABILITY FOR CLASS 1, DIVISION 2.

WARNING !
EXPLOSION HAZARD – WHEN IN HAZARDOUS LOCATIONS, TURN OFF POWER BEFORE REPLACING OR WIRING MODULES.

WARNING !
EXPLOSION HAZARD - DO NOT DISCONNECT EQUIPMENT UNLESS POWER HAS BEEN SWITCHED OFF OR THE AREA IS KNOWN TO BE NONHAZARDOUS.

Selecting System Modules

The selection of controller and 5000 Series I/O modules needed for an application requires an understanding of the application requirements and a knowledge of the controller and 5000 Series I/O modules available.

Section ***0-Application Requirements*** describes the general requirements of an application that will determine the type of controller and 5000 Series I/O modules needed. Once the application requirements are fully understood the process for selecting controller and 5000 Series I/O modules is straightforward.

- Section ***0-Controller Selection*** describes each type of controller available.
- Section ***0-Modem Selection*** describes the types of 5000 Series modems that are available.
- Section ***0-I/O Module Selection*** describes the types of 5000 Series I/O modules that are available.
- Section ***0-Power Supply Selection*** provides detailed information on system power requirements and how to supply the required power.

Application Requirements

The requirements of the application will determine which controller and 5000 Series I/O modules need to be selected. The application requirements may be separated into these general areas:

- Programming requirements
- Communication requirements
- I/O requirements
- Power supply requirements

Review each of these areas in the following paragraphs to ensure the appropriate controller and 5000 Series I/O modules are selected for the application.

Programming Requirements

A programmable controller, such as the SCADAPack or TeleSAFE Micro16, is required when:

- The controller must perform local, or remote, process control.
- The controller is a master station for more than one slave station.
- Data must be saved, manipulated or logged in the controller.

A non-programmable controller, such as the SmartWIRE, may be used when:

- The application is an end-to-end telemetry system.
- The controller is used as remote I/O or slave I/O for a master station.
- No local control or decision making capability is required.

Communication Requirements

The communication requirements include the number of communication ports the controller must have. The controller may typically communicate with one or more of the following:

- remote slave stations;

- remote master stations;
- a local HMI (Human Machine Interface);
- local, or remote, programming software;
- smart measurement devices;
- other controllers at the site.

In addition to the number of communication ports that are required the type of communication port may also be a requirement.

- An RS-485 serial communication port is available with all controllers except the SmartWIRE 5202 controller.
- One, or more, RS-232 serial communication ports are available on all controllers except the SmartWIRE 5201 controller.
- A 10baseT Ethernet port is available on the SCADAPack 32 and the SCADAPack 32P controllers.

The following table shows the communication port configurations for each type of controller.

Controller Type	RS-232	RS-232/RS-485	Ethernet	USB	Total
SmartWIRE 5201	0	1	0	0	1
SmartWIRE 5202	1	0	0	0	1
TeleSAFE Micro16	1	1	0	0	2
SCADAPack Light	2	1	0	0	3
SCADAPack	2	1	0	0	3
SCADAPack 2	1	2	1	2	6
SCADAPack Plus	3	1	0	0	4
SCADAPack 32	3	1	1	0	5
SCADAPack 32P	2	1	1	0	4
SCADAPack LP	2	1	0	0	3
SCADAPack 100	1	1	0	0	2

There may be a requirement that the controller communicate with other controllers, SCADA hosts or programming software using modem communication. There are two 5000 Series modems available.

- The 5901 dial up modem is used to connect to standard telephone lines.
- The 5902 Bell 202 modem is used with dedicated telephone lines, direct buried lines and radio transceivers.

I/O Requirements

The controller is normally required to monitor and control digital and analog inputs and outputs (I/O). These inputs and outputs may include transmitters indicating levels or flow, HOA switches, solenoids and VFD inputs.

The controller uses on board I/O, 5000 Series I/O modules, or a combination of both to monitor and control these devices. The size of the application, i.e. the number of I/O points, will determine what I/O the controller uses.

The following table shows the on board I/O for each type of controller.

	Digital Inputs[1]	Digital Outputs[2]	Analog Inputs	Analog Outputs
TeleSAFE Micro16	4	1	0	0
SCADAPack Light	4	3	5	0
SCADAPack	20	13	8	2[3]
SCADAPack2	8[4]	8[4]	8	2[3]
SCADAPack Plus	20	15	13	2[3]
SCADAPack 32	20	15	13	2[3]
SCADAPack 32P	4	1	0	0
SCADAPack LP	8[4]	8[4]	6	2[3]
SCADAPack 100	6[4]	6[4]	4	0

[1] Digital inputs include the interrupt input.
[2] Digital outputs include the status output.
[3] Optional, if ordered at time of purchase.
[4] Digital I/O is both input and output.

5000 Series I/O modules provide an expansion of the on board I/O. A maximum of forty 5000 Series I/O modules may be added to each controller. See section ***0-I/O Module Selection*** for a complete list of the 5000 Series I/O modules available.

Power Supply Requirements

The power requirements will depend on a number of factors. These factors include:

- The type of power available such as 115 or 220 VAC and 12 or 24 VDC.
- Limited power availability as in solar powered installations.
- The type of controller used and the number of 5000 Series I/O modules required for the application.
- The requirement of an uninterruptable power supply (UPS).

The SCADAPack and Micro16 controllers have an integrated power supply. This internal power supply will power the controller and a limited number of 5000 Series I/O modules.

The 5103 power supply provides operating power for controllers and 5000 Series I/O modules.

Section ***0-Power Supply Selection*** contains complete information for determining and supplying power requirements.

Controller Selection

Select a controller based on the application requirements determined in the preceding sections of this manual. The controller is selected using the following considerations:

- Is a programmable or non-programmable controller required for the application?
- What type and number of serial communication ports are required for the application?
- How many I/O points does application require?

Each type of controller available is described in the following paragraphs. For more detailed information on these controllers refer to the hardware manual for the controller.

SmartWIRE 5201 Controller

The SmartWIRE 5201 controllers provide an RS-485 serial communication interface to the 5000 Series I/O system. This is a non-programmable controller with configuration parameters such as station address, serial port settings and communication protocol set using DIP switches.

Use the 5201 RS-485 communication controller to:

- connect one or more 5000 Series I/O systems to a personal computer, PLC or SCADA system within 4000 feet (1219 m);
- connect remote I/O to a TeleSAFE controller;
- create a high speed end-to-end telemetry system using two 5201 controllers;
- connect to a device requiring very high speed communication;
- connect to existing RS-485 equipment.

The 5201 SmartWIRE controller supports a maximum of forty 5000 Series I/O modules.

Note: SmartWIRE controllers do not support sixteen point modules and can only use two points on the 5302 Analog Output module.

The 5201 Communication Controller accommodates:

- 64 digital inputs
- 64 digital outputs
- 64 analog inputs
- 32 analog outputs
- 32 counter inputs

SmartWIRE 5202 Controller

The SmartWIRE 5202 controllers provide an RS-232 serial communication interface to the 5000 Series I/O system. This is a non-programmable controller with configuration parameters such as station address, serial port settings and communication protocol set using DIP switches.

Use the 5202 RS-232 communication controller to:

- connect one 5000 Series I/O system to a personal computer, PLC or SCADA system within 50 feet (15.2m);
- create an end-to-end telemetry system using two 5202 controllers;
- connect to a modem for communication over telephone lines or dedicated lines longer that 4000 feet (1219 m);
- connect to any type of radio transceiver.

The SmartWIRE 5201 controller supports a maximum of forty 5000 Series I/O modules.

Note: SmartWIRE controllers do not support sixteen point modules and can only use two points on the 5302 Analog Output module.

The 5201 Communication Controller accommodates:

- 64 digital inputs
- 64 digital outputs

- 64 analog inputs
- 32 analog outputs
- 32 counter inputs

TeleSAFE Micro16 Controller

The TeleSAFE Micro16 controller has two serial communication ports. The model 5203 has two RS-232 communication ports and the model 5204 has one RS-232 and one RS-485 communication ports.

The TeleSAFE Micro16 controller is programmable using TelePACE Ladder Logic, IEC61131 or TelePACE C Tools.

Use the TeleSAFE Micro16 Controller to:

- provide local control or data acquisition;
- act as a data concentrator for remote TeleSAFE, SCADAPack or SmartWIRE systems;
- provide up to two RS-232, or one RS-232 and one RS-485, serial connections to host computers, SCADA systems or remote I/O;
- provide a multi-point telemetry system.

The TeleSAFE Micro16 controller has the following on-board I/O:

- 3 digital/counter inputs
- 1 interrupt digital input
- 1 status digital output
- 1 ambient temperature input
- 1 RAM voltage input

In addition to the on-board I/O the TeleSAFE Micro16 controller supports a maximum of forty 5000 Series I/O modules. Any combination of I/O modules may be used to a maximum of:

- 256 digital inputs
- 256 digital outputs
- 128 analog inputs
- 64 analog outputs
- 32 counter inputs

The TeleSAFE Micro16 controller has an integrated power supply.

SCADAPack Light Controller

The SCADAPack Light controller has three serial communication ports. With the 5203 controller board the SCADAPack Light has three RS-232 communication ports. With the 5204 controller board the SCADAPack Light has two RS-232 and one RS-485 communication ports.

The SCADAPack Light controller is programmable using TelePACE Ladder Logic, IEC61131 or TelePACE C Tools.

Use the SCADAPack Light Controller to:

- provide local control or data acquisition;
- act as a data concentrator for remote TeleSAFE, SCADAPack or SmartWIRE systems;

- provide up to three serial communication connections to host computers, SCADA systems or remote I/O;
- provide a multi-point telemetry system.

The SCADAPack Light has the following on-board I/O:

- 3 digital/counter inputs
- 1 interrupt digital input
- 2 mechanical relay outputs
- 1 status digital output
- 5 combination analog inputs / digital inputs
- 1 ambient temperature input
- 1 RAM voltage input

In addition to the on-board I/O the SCADAPack Light controller supports a maximum of forty 5000 Series I/O modules. Any combination of I/O modules may be used to a maximum of:

- 256 digital inputs
- 256 digital outputs
- 128 analog inputs
- 64 analog outputs
- 32 counter inputs

The SCADAPack Light controller has an integrated power supply.

SCADAPack Controller

The SCADAPack controller has three serial communication ports. With the 5203 controller board the SCADAPack has three RS-232 communication ports. With the 5204 controller board the SCADAPack has two RS-232 and one RS-485 communication ports.

The SCADAPack controller is programmable using TelePACE Ladder Logic, IEC61131 or TelePACE C Tools.

Use the SCADAPack Controller to:

- provide local control or data acquisition;
- act as a data concentrator for remote TeleSAFE, SCADAPack or SmartWIRE systems;
- provide up to three serial communication connections to host computers, SCADA systems or remote I/O;
- provide a multi-point telemetry system.

The SCADAPack controller has the following on-board I/O:

- 3 digital/counter input points
- 16 digital inputs
- 1 interrupt input point
- 12 mechanical relay outputs

- 1 status digital output
- 8 analog inputs
- 2 optional analog outputs
- 1 ambient temperature input
- 1 RAM voltage input

In addition to the on-board I/O the SCADAPack controller supports a maximum of forty 5000 Series I/O modules. Any combination of I/O modules may be used to a maximum of:

- 256 digital inputs
- 256 digital outputs
- 128 analog inputs
- 64 analog outputs
- 32 counter inputs

The SCADAPack controller has an integrated power supply.

SCADAPack2 Controller

The SCADAPack2 controller has three serial communication ports. There is a dedicated RS-485 port, a dedicated RS-232 port and a configurable RS-232/485 port.

The SCADAPack2 controller is programmable using TelePACE Ladder Logic, IEC61131 or TelePACE C Tools.

Use the SCADAPack2 Controller to:

- provide local control or data acquisition;
- act as a data concentrator for remote TeleSAFE, SCADAPack or SmartWIRE systems;
- provide up to three serial communication connections to host computers, SCADA systems or remote I/O;
- provide communication access through local and wide area networks.
- provide a multi-point telemetry system.

The SCADAPack2 controller has the following on-board I/O:

- 3 digital/counter input points
- 8 digital inputs/outputs
- 6 analog inputs
- 2 optional analog outputs
- 1 ambient temperature input
- 1 RAM voltage input

In addition to the on-board I/O the SCADAPack controller supports a maximum of forty 5000 Series I/O modules. Any combination of I/O modules may be used to a maximum of:

- 512 digital inputs
- 512 digital outputs
- 128 analog inputs

- 64 analog outputs
- 64 counter inputs

The SCADAPack2 controller has an integrated power supply.

SCADAPack Plus Controller

The SCADAPack Plus controller has four serial communication ports. With the 5203 controller board the SCADAPack Plus has four RS-232 communication ports. With the 5204 controller board the SCADAPack Plus has three RS-232 and one RS-485 communication ports.

The SCADAPack Plus controller is programmable using TelePACE Ladder Logic, IEC61131 or TelePACE C Tools.

Use the SCADAPack Plus Controller to:

- provide local control or data acquisition;
- act as a data concentrator for remote TeleSAFE, SCADAPack or SmartWIRE systems;
- provide up to four serial communication connections to host computers, SCADA systems or remote I/O;
- provide a multi-point telemetry system.

In addition to the on-board I/O the SCADAPack Plus controller supports a maximum of forty 5000 Series I/O modules. Any combination of I/O modules may be used to a maximum of:

- 256 digital inputs
- 256 digital outputs
- 128 analog inputs
- 64 analog outputs
- 32 counter inputs

The SCADAPack Plus has the following on-board I/O, in addition to the limits above:

- 3 digital/counter input points
- 16 digital inputs
- 1 interrupt input point
- 14 mechanical relay outputs
- 1 status digital output
- 13 analog inputs
- 2 optional analog outputs
- 1 ambient temperature input
- 1 RAM voltage input

The SCADAPack Plus controller has an integrated power supply.

SCADAPack 32 Controller

The SCADAPack 32 controller has four serial communication ports. Three of the serial communication ports are RS-232 ports. One of the serial ports is jumper configurable for either RS-232 or RS-485 serial communication.

The SCADAPack 32 has one10baseT Ethernet port. This Ethernet port supports Modbus\TCP communication protocol.

The SCADAPack 32 uses a 32-bit processor and has 8 Mbytes SDRAM, 4 Mbytes flash ROM, and 0.5 Mbytes CMOS SRAM memory. The CMOS SRAM is non-volatile (battery backed.).

A real time clock calendar provides for time of day operations and alarms. A hardware watchdog timer protects against application program failures.

The SCADAPack 32 controller is programmable using TelePACE Ladder Logic, IEC61131 or TelePACE C++ Tools.

Use the SCADAPack 32 Controller to:

- perform gas flow calculations for multiple meter runs.
- provide local control or data acquisition and logging;
- act as a data concentrator for remote TeleSAFE, SCADAPack or SmartWIRE systems;
- provide up to four serial communication connections to host computers, SCADA systems or remote I/O;
- provide a multi-point telemetry system.
- provide communication access through local and wide area networks.

In addition to the on-board I/O the SCADAPack 32 controller supports a maximum of forty 5000 Series I/O modules. Any combination of I/O modules may be used to a maximum of:

- 512 digital inputs
- 512 digital outputs
- 128 analog inputs
- 64 analog outputs
- 32 counter inputs

The SCADAPack 32 has the following on-board I/O, in addition to the limits above:

- 3 digital/counter input points
- 16 digital inputs
- 1 interrupt input point
- 14 mechanical relay outputs
- 1 status digital output
- 13 analog inputs
- 2 optional analog outputs
- 1 ambient temperature input
- 1 RAM voltage input

The SCADAPack 32 controller has an integrated power supply.

SCADAPack 32P Controller

The SCADAPack 32P controller has three serial communication ports. Two of the serial communication ports are RS-232 ports. One of the serial ports is jumper configurable for either RS-232 or RS-485 serial communication.

The SCADAPack 32P has one10baseT Ethernet port. This Ethernet port supports Modbus\TCP communication protocol.

The SCADAPack 32P uses a 32-bit processor and has 8 Mbytes SDRAM, 4 Mbytes flash ROM, and 0.5 Mbytes CMOS SRAM memory. The CMOS SRAM is non-volatile (battery backed.).

A real time clock calendar provides for time of day operations and alarms. A hardware watchdog timer protects against application program failures.

The SCADAPack 32P controller is programmable using TelePACE Ladder Logic, IEC61131 or TelePACE C++ Tools.

Use the SCADAPack 32P Controller to:

- perform gas flow calculations for multiple meter runs.
- provide local control or data acquisition and logging;
- act as a data concentrator for remote TeleSAFE, SCADAPack or SmartWIRE systems;
- provide up to four serial communication connections to host computers, SCADA systems or remote I/O;
- provide a multi-point telemetry system.
- provide communication access through local and wide area networks.

In addition to the on-board I/O the SCADAPack 32P controller supports a maximum of forty 5000 Series I/O modules. Any combination of I/O modules may be used to a maximum of:

- 512 digital inputs
- 512 digital outputs
- 128 analog inputs
- 64 analog outputs
- 32 counter inputs

The SCADAPack 32 has the following on-board I/O, in addition to the limits above:

- 3 digital/counter input points
- 1 interrupt input point
- 1 status digital output
- 1 ambient temperature input
- 1 RAM voltage input

The SCADAPack 32P controller has an integrated power supply.

SCADAPack LP Controller

The SCADAPack LP controller has three serial communication ports. One RS-485 port is designed for use with multivariable transmitters. One of the two RS-232 ports was designed for use with the SCADAPack Vision operator interface.

The primary microcontroller memory is 512KBytes flash ROM, and 1MBytes RAM. The CMOS RAM is non-volatile (battery backed). An EEPROM (1kBytes) stores configuration parameters.

Several power saving features are included in the SCADAPack LP. These power saving features include Sleep Mode, 24V-power shutdown, communication port power control and SCADAPack Vision power down.

The SCADAPack LP controller is programmable using TelePACE Ladder Logic, IEC61131 or TelePACE C Tools.

Use the SCADAPack LP Controller to:

- perform gas flow calculations for multiple meter runs in a very low power environment.
- provide local control or data acquisition and logging;
- act as a data concentrator for remote TeleSAFE, SCADAPack or SmartWIRE systems;
- provide up to four serial communication connections to host computers, SCADA systems or remote I/O;
- provide a multi-point telemetry system.

In addition to the on-board I/O the SCADAPack LP controller supports a maximum of forty 5000 Series I/O modules. Any combination of I/O modules may be used to a maximum of:

- 256 digital inputs
- 256 digital outputs
- 128 analog inputs
- 64 analog outputs
- 32 counter inputs

The SCADAPack LP has the following on-board I/O, in addition to the limits above:

- 3 high-speed counter inputs, two of which are designed for direct connection to the millivolt output of turbine meter transducers.
- 8 universal digital inputs / outputs, each capable of sinking 1A or monitoring a dry contact closure.
- 6 single-ended analog inputs with 15-bit resolution over the entire input range.
- 2 optional analog outputs
- 1 ambient temperature input.
- 1 RAM voltage input.
- 1 internal DC/DC converter voltage input.

The SCADAPack LP controller has an integrated power supply.

SCADAPack 100 Controller

The SCADAPack 100 controller has two serial communication ports. One serial port is either an RS-485 or an RS-232 port.

The SCADAPack primary microcontroller memory is 512KBytes flash ROM, and 256KBytes RAM. The CMOS RAM is non-volatile (battery backed). An EEPROM (1kBytes) stores configuration parameters. A real time clock calendar provides for time of day operations and alarms. A hardware watchdog timer protects against application program failures.

The SCADAPack 100 controller is programmable using TelePACE Ladder Logic, IEC61131 or TelePACE C Tools.

Use the SCADAPack LP Controller to:

- provide local control or data acquisition and logging;
- act as a data concentrator for remote TeleSAFE, SCADAPack or SmartWIRE systems;

- provide a multi-point telemetry system.

The SCADAPack 100 has the following on-board I/O, in addition to the limits above:

- 1 high-speed counter input.
- 6 universal digital inputs / outputs, each capable of sinking 1A or monitoring a dry contact closure.
- 4 single-ended analog inputs with 12-bit resolution over the entire input range.
- 1 ambient temperature input.
- 1 RAM voltage input.
- 1 internal DC/DC converter voltage input.

The SCADAPack LP controller has an integrated power supply.

Modem Selection

There are two 5000 Series modems available, a dial up modem for connection to standard public switched networks and a Bell 202 modem for connection to leased telephone lines, private dedicated lines and radio transceivers.

Model 5901 Dial-Up Modem

The 5901 modem provides high-speed dial-up telephone communication for SCADAPack, TeleSAFE Micro16 and SmartWIRE 5202 Communication controllers. The 5901 modem may be used with all TeleSAFE controllers except the 5201 RS-485 Communication controller.

The 5901 modem supports telephone line connection speeds up to 14400 baud. The 5901 modem performs complete handshake and data rate negotiations and supports error correction and data compression modes to ensure data transfer integrity. Non-error correcting mode is also supported. The 5901 modem supports DTE speeds up to 57600 baud with speed buffering at all speeds. XON/XOFF or RTS/CTS flow control between DTE and 5901 modem are supported.

Auto-recovery circuitry in the 5901 modem ensures controlled modem operation at all times.

Model 5902 Bell 202 Modem

The 5902 Bell 202 modem converts RS-232 signal levels to audio tones for transmission over leased telephone lines, private dedicated lines and radio transceivers. The 5902 modem may be used with all TeleSAFE controllers except the 5201 RS-485 Communication controller.

The 5902 modem can be used on two or four wire half duplex phone, or dedicated, lines (multi-point) or four wire full duplex phone or dedicated, lines (point-to-point).

Radio communication is facilitated by an isolated radio transmitter key output and a radio frequency carrier detect input. The 5902 modem is compatible with virtually any make and model of radio transceiver.

Model 5904 HART Interface Module

The 5904 HART Interface Module provides communication between SCADAPack and Mico16 controllers and intelligent field instruments using the industry standard HART (Highway Addressable Remote Transducer) communications protocol.

The 5904 HART module can operate as a primary master or a secondary master device. Four HART interface modules are supported per controller. Ladder Logic and C Tools application programs can read data from HART devices.

The 5904 HART Interface Module is a 5000 Series module and may be fully integrated into a DIN rail mounted controller system. Operating power is received from the I/O Bus and the 5904 HART Interface module requires no external power source.

Model 5906 Ethernet Gateway Module

The model 5906 Ethernet gateway converts RS-232/485 Modbus communication into Modbus/TCP on Ethernet. Applications include in-plant I/O and connectivity between SCADAPack, Micro16 and SmartWIRE controllers and operator workstations, other PLC devices, and SCADA systems communication, all using standard Modbus. The 5905 gateway allows interfacing of multiple or dual-redundant host computers that must be able to communicate to SCADA systems over a single communications channel.

Support for RS-232 using RTS/CTS handshaking, allows the 5906 to be used as a gateway between Ethernet LAN and commonly used half-duplex SCADA communications media such as the 5902 Bell 202 modem, radio modems, and various other half-duplex modems.

Multi-drop communication over RS-485, or half-duplex modems and radios, allows up to 255 Modbus slave devices to be connected to a 5905 Ethernet gateway. Each slave 5905 gateway is configured with an IP address and the master end has a configurable table defining which Modbus slave stations can be found at each 5905 slave gateway. Up to 255 Modbus slaves can be connected to each gateway, and up to 8 TCP/IP connections are supported by each 5905.

I/O Module Selection

The 5000 Series I/O System has a wide variety of I/O modules. Select the modules that meet the needs of your application from the list of available modules below. Refer to the installation and user manual for each module for specifications and configuration instructions.

Module	Number of Points	Description
5302	4	Analog Output Module
5304	4	Analog Output Module
5501	8	Analog Input Module
5502	8	Differential Input Analog Input Module
5503/5505	8	RTD Input Module
5504	8	Thermocouple Input Module
5506	8	Analog Input Module
5521	8	Potentiometer Analog Input Module
5401	8	Digital Input/Output Module
5402	16	Digital Input/Output Module
5403	8	Digital Input Module
5404	16	Digital Input Module
5405	32	Digital Input Module
5406	16	Relay Output Module
5407	8	Relay Output Module
5408	8	Triac Output Module
5409	8	FET Output Module
5410	4	High Speed Counter Input Module
5411	32	MOSFET Output Module
5421	8	Switch Input Module

Note: SmartWIRE controllers do not support sixteen and thirty-two point modules; and can only use two points on the 5302 and 5304 Analog Output modules.

A maximum of forty 5000 Series I/O modules may be connected to a SCADAPack, TeleSAFE Micro16 or SmartWIRE controller. Power supply and modem modules do not count as part of the forty module maximum.

Power Supply Selection

All 5000 Series I/O modules, SCADAPack and TeleSAFE Micro16 controllers and SmartWIRE communication controllers require 5V power. Some I/O modules require 24V power. Refer to the individual *Module Installation and User* manual for specific module the power requirements.

The model 5103 power supply provides 5V and 24V operating power for all controller and I/O modules. Model 5103 power supplies can be added anywhere in a system to provide the necessary power capacity for any size of system.

The SCADAPack and TeleSAFE Micro16 controllers have on-board power supplies and may provide the 5V and 24V power for themselves and a limited number of 5000 Series I/O modules.

The SCADAPack and TeleSAFE Micro16 controllers may be powered with the 5103 power supply. A jumper setting on the controller is required when it is powered with a 5103 power supply.

Note: Voltage referred to as Vrms (or VAC on some products) indicates AC power. Voltage referred to as V indicates DC power.

Calculating System Power Requirements

To calculate the power supply requirements the 5V supply and 24V supply current requirements for each module must be determined. The *Module Power Requirements* chart below summarizes the power requirements for all controllers and 5000 Series modules.

1. In ***Table 1: Module Power Requirements Table*** write the quantity of each type of module and in the ***Qty*** column beside the module.
2. For each module multiply the value in the ***Qty*** column by the value in the ***Module Current @ 5 V*** column and write the result in the ***Total Current @ 5 V*** column.
3. For each module multiply the value in the ***Qty*** column by the value in the ***Module Current @ 24 V*** column and write the result in the ***Total Current @ 24 V*** column.
4. Write the number of analog loops that will require power in the ***Qty*** column beside ***Analog Loops***. Multiply the value in the ***Qty*** column by the value in the ***Module Current @ 24 V*** and write the result in the ***Total Current @ 24 V*** column.
5. In the Totals section sum the ***Total Current @ 5 V*** and ***Total Current @ 24 V*** columns.

Table 1: Module Power Requirements Table

Module	**Qty**	**Module Current @ 5 V**	**ModuleCurrent @ 24 V**	**Total Current @ 5 V**	**TOTAL CURRENT @ 24 V**
SmartWIRE 5201		130mA	0mA		
SmartWIRE 5202		60mA	0mA		
TeleSAFE Micro16[1]		175mA	0mA		
SCADAPack[1]		175mA – 655mA[2]	20mA		

Module	Qty	Module Current @ 5 V	ModuleCurrent @ 24 V	Total Current @ 5 V	TOTAL CURRENT @ 24 V
SCADAPack2		See user manual[3]			
SCADAPack Light[1]		175mA – 255mA[2]	15mA		
SCADAPack Plus[1]		175mA - 755mA[2]	35mA		
SCADAPack 32		500mA - 980mA[2]			
SCADAPack 32P		500mA			
SCADAPack LP		55mA			
SCADAPack AOUTs		10mA	55mA		
5901		200mA	0mA		
5902		60mA	0mA		
5905		250A	0mA		
5904		20mA	0mA		
5401		90mA	0mA		
5402		150mA	0mA		
5403 - all versions		45mA	0mA		
5404		80mA	0mA		
5405		10mA	0mA		
5406		600mA	0mA		
5407		300mA	0mA		
5408		120mA	0mA		
5409		70mA	30mA		
5410		35mA	0mA		
5421		45mA	0mA		
5302		20mA	10mA		
5501- all versions		20mA	10mA		
5502		100mA			
5503		40mA	30mA		
5504		40mA	45mA		
5505		6mA	2.2mA		
5506		45mA	11mA		
5521		20mA	0mA		
Analog Loops		0mA	20mA		
TOTALS					

Notes:

[1] Add 75mA to Module Current @ 5 V when 5304 controller is used.

[2] The range shown is dependant on the number of digital output relays used. The minimum value shown is with no output relays used. Add 40mA per output relay to the maximum value shown.

[3] Current available for expansion I/O is limited to 600mA irrespective of controller board load

Supplying System Power Requirements

There are two sources of system power available. These sources may be combined to provide power for any size system. The sources available are:

- The model 5103 power supply provides 5V and 24V operating power for all controller and I/O modules.
- The SCADAPack and TeleSAFE Micro16 controllers have on-board power supplies and may provide the 5V and 24V power for themselves and a limited number of 5000 Series I/O modules.

The 5103 module will supply 1000mA at 5V and 500mA at 24V, independent of the loads. The TeleSAFE Micro16 and SCADAPack controllers will supply 1000mA at 5V and varying amounts of current at 24V, depending on the 5V load.

- The physical location of the power supply sources may require that more power will be required than was calculated, since the power supply outputs are not summed. Refer to section ***0-***

Note: ***Power Supply Installation*** Guidelines for more information.

The table below shows the power supply capacity for each power source. See section ***0-Calculating Controller 24 Volt Capacity*** for information on how to determine the 24V capacity for an installation.

Power Supply Capacity			
Module	**Current Capacity @ 5V**	**Minimum Current Capacity @ 24V**	**Maximum Current Capacity @ 24V**
5103 POWER SUPPLY	** 2000mA	500mA	500mA
TeleSAFE Micro16	1000mA	80mA	360mA
SCADAPack Light	1000mA	80mA	330mA
SCADAPack	1000mA	80mA	190mA
SCADAPack Plus	1000mA	80mA	155mA
SCADAPack 32	1300mA	100mA	300mA
SCADAPack 32P	1300mA	100mA	190mA
SCADAPack LP	400mA	140mA	140mA
SCADAPack 100	250mA	0mA	0mA

** Applies only to newer models. Current rating for older models is 1000mA at 5V.

Calculating Controller 24 Volt Capacity

To determine the maximum current capacity at 24 volts the following formulae are used.

For all Modules except SCADAPack 32 and SCADAPack 32P:

24V current = 80 + ((1000 – Total Current @ 5V) x 0.35)

where:

80 = Minimum current @ 24V

1000 = Maximum current @ 5V

Total Current @ 5V = Sum of 5V current required for system

0.35 = Derating factor

For SCADAPack 32 and SCADAPack 32P Modules:

24V current = 100 + ((1300 – Total Current @ 5V) x 0.35)

where:

80 = Minimum current @ 24V

1000 = Maximum current @ 5V

Total Current @ 5V = Sum of 5V current required for system

0.35 = Derating factor

Power Supply Installation Guidelines

Multiple power supplies may be required in a system. The power requirements of the I/O modules determine the locations of the supplies.

5103 Power Supplies

Follow these guidelines when connecting 5103 power supplies:

- The I/O bus connector on the right side of the 5103 power supply is the output connector. The I/O bus connector on the left side of the module is the input connector. The 5V power from the 5103 power supply is connected to the output connector only.
- All modules connected to the output connector of the 5103 power supply are powered by it, until another power supply is installed.
- The first (leftmost) module in a SmartWIRE system must be a power supply. An additional power supply may be installed at any point. It powers all the modules connected to its output side. Add another power supply when the current requirements of the modules reach the capacity of the previous supply.

WARNING: Never connect the outputs of two power supplies together. The power supplies may be damaged.

Some modules require a 24V-power supply. The 24V supply is connected to I/O modules with field wiring.

- The 24V supply is isolated from the 5V supply. I/O functions operating from the 24V supply are isolated from the I/O bus, but not from other I/O modules sharing the 24V supply. Use additional 24V supplies, if necessary, to isolate I/O modules.
- The 24V supply powers isolated I/O functions such as analog inputs and outputs. If current loop transmitters do not require a 24V supply, a 12V supply may be used to lower power consumption. Refer to the individual module manuals for minimum operating voltage specifications.

Controller Power Supplies

Always refer to the controller Hardware Manual for detailed information when using the integrated power supply in a SCADAPack or TeleSAFE Micro16 controller. Follow these guidelines when using the controller integrated power supply:

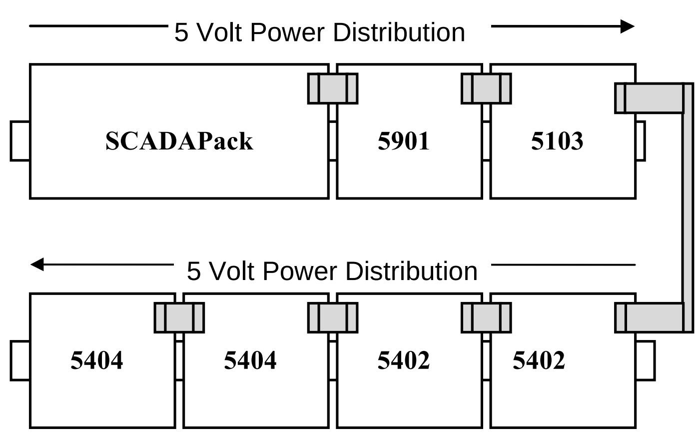
- The integrated power supply in the TeleSAFE Micro16 and SCADAPack controllers powers modules to the left and right of it. It requires a special jumper setting before a 5103 power supply can be connected to the left (input side) of it.
- If a TeleSAFE Micro16 or SCADAPack controller is used without an external power supply, it should be the first module.
- The SCADAPack 100 controller does not support the use of a 5103 power supply.

Example Power Requirement Calculation

Suppose the system has a SCADAPack controller with a 5204 controller board and using three analog inputs. The system also uses a 5901 modem, two 5402 and two 5404 modules. The power requirements are shown in the following table.

Module Power Requirements for Example Calculation					
Module	**Qty**	**Module Current @ 5 V**	**Total Current @ 5 V**	**Module Current @ 24 V**	**Total Current @ 24 V**
SCADAPack	1	675mA	675mA	20mA	20mA
5204 Controller	1	75mA	75mA	0mA	0mA
5901	1	200mA	200mA	0mA	0mA
5402	2	150mA	300mA	0mA	0mA
5404	2	80mA	160mA	0mA	0mA
Analog Loops	3	0mA	0mA	20mA	60mA
Other Loads	0	0mA	0mA	0mA	0mA
TOTALS			1410mA		80mA

From the power supply capacity table it is seen that the SCADAPack will supply 1000mA at 5V. Since the requirement is 1410mA a 5103 power supply will be required in addition to the SCADAPack power supply. The following diagram shows how this system would be laid out. Note that the SCADAPack can supply the required 24V power for the analog loops.



System Layout

System layout is the process of mounting and connecting the controllers and 5000 Series I/O modules required for an application. The modules mount on DIN rails for ease of installation. All field wiring attaches to one side of the modules. Extended inter-module cabling allows multiple DIN rail installations in restricted areas.

WARNING:	**EXPLOSION HAZARD**
	SUBSTITUTION OF COMPONENTS MAY IMPAIR SUITABILITY FOR CLASS I, DIVISION 2.

Mounting Modules

All SCADAPack controllers, TeleSAFE Micro16 controllers, SmartWIRE communication controllers and 5000 Series modules mount on 7.5mm by 35mm DIN type rails.

Mounting Guidelines

Follow these guidelines for mounting modules:

- DIN rails mount horizontally or vertically. Where possible mount DIN rails horizontally. Modules are easier to install on horizontal rails. Cooling is optimal when mounted horizontally.
- The model DIN17 rail supplied by Control Microsystems holds four modules. It is 17 inches (432 mm) long. If you use another length of DIN rail, cut it to a multiple of the module width. See the figure *Module Dimensions* for more details.
- Mount modules tightly together on the rail, to avoid straining the inter-module cable. The spacing dimples on the sides of the modules should touch.

SmartWIRE, TeleSAFE, SCADAPack and 5000 Series I/O modules are certified as components for use within an enclosure.

Mounting Procedure

Use the following procedure to mount each module on the DIN rail.

- Unscrew the two cover screws on each module. These screws are permanently attached to the cover for ease of handling.
- Remove the cover.
- There are two clamp screws accessible through notches on either side of the circuit board. Loosen the clamp screws until there is a 1/16 inch (1.5mm) clearance between the back of the module and the clamp. Do not remove the screws from the clamp, as they can be lost inside the case.
- Slide the clamp out past the bottom of the module.
- There are two hooks on the back of the module. Place the hooks over the top lip of the DIN rail.
- Slide the clamp upward until it meets the lower lip of the rail. The bottom of the clamp will be flush with the bottom of the module.
- Tighten the clamp screws.

CAUTION:	If you route an inter-module cable (IMC) behind the module, take care when installing the module. Avoid pinching the cable in the clamp and ensure that there is adequate clearance between the module and the hardware used to mount the DIN rails.

Connecting Modules

All SCADAPack controllers, TeleSAFE Micro16 controllers, SmartWIRE communication controllers and 5000 Series modules connect together with inter-module cables (IMC). The IMC distributes power and communication signals to the 5000 Series modules. These power and communication signals are referred to as the I/O Bus.

The IMC is a 16-conductor ribbon cable. TeleSAFE Micro16 controllers, SmartWIRE communication controllers and 5000 Series modules ship with a 2 inch (5cm) IMC cable. SCADAPack and SCADAPack Plus controllers ship with a 3 inch (7.5cm) IMC. This cable connects adjacent modules on a DIN rail. The IMC12 (12 inch) and IMC30 (30 inch) and longer cables connect modules on separate DIN rails.

To install the IMC:

- Remove the module cover.
- Connect the IMC to the I/O bus connector on the left or right side of the module. Align the key on the IMC with the slot in the connector. Press the IMC firmly into the connector.
- Connect the other end of the IMC to an adjacent module.

Always use the shortest length IMC that is practical. This minimizes voltage drops and interference from electrical noise. Keep the IMC away from field wiring and other sources of electrical noise.

Shielded Inter-Module Cables

All IMC cables longer than 12 inches (30 cm) are shielded for maximum physical protection and protection from electrical noise. Shielded inter-module cables have a foil and braid shielding. The shielding is connected to a terminal lug at one end of the cable.

Note:	When using a shield for an IMC cable between modules, the shield should be fastened to one module only, and that module should be the one closest to the controller. Always connect the shield to the enclosure using the self-tapping screw provided.

Module Field Wiring

Field wiring attaches to the field wiring terminal blocks on each module. Refer to the individual module Installation and User Manual for complete wiring details.

WARNING:	**EXPLOSION HAZARD** WHEN IN HAZARDOUS LOCATIONS, TURN OFF POWER BEFORE REPLACING OR WIRING MODULES.

WARNING:	**EXPLOSION HAZARD** DO NOT DISCONNECT EQUIPMENT UNLESS POWER HAS BEEN SWITCHED OFF OR THE AREA IS KNOWN TO BE NON HAZARDOUS

Field Wiring Guidelines

Follow these guidelines when planning for field wiring:

- When modules are installed in a Class I Division 2 area then power, input and output (I/O) wiring must be in accordance with Class I, Division 2 wiring methods (article 501-4(b) of the National Electrical Code, NFPA 70) and in accordance with the authority having jurisdiction.
- Arrange the modules so field wiring is accessible.
- Field wiring termination connectors are removable. Leave enough slack in the wiring for the connector to be removed.
- Arrange the modules and wiring so the status LEDs are visible.
- Keep modules switching loads away from communication controllers and analog input modules.
- Route low-level signals separate from high-level and AC power signals.
- No bonding is usually required by Electrical Safety Standards between modules carrying primary power and AC Ground since these modules are certified as components to be installed within an enclosure thereby preventing access by unauthorized personnel. Consult individual module manuals for special bonding instructions should they be required.
- Consult individual module manuals for the connection of external fuses should they be required in particular by primary voltage carrying modules. They must be installed where required.

Suggested Layouts

Controller and 5000 Series I/O, power supply and modem modules may be mounted in many ways. The best possible layout places all modules on one DIN rail. This minimizes the IMC length, provides the best protection against electrical noise and has no module placement restrictions.

Figure 1: Preferred SmartWIRE Layout shows the preferred multiple rail layout for a SmartWIRE communication controller system. Two IMC12 cables distribute power and communication signals from rail to rail. I/O modules designated 5xxx refer to any 5000 Series I/O modules except the 5103.

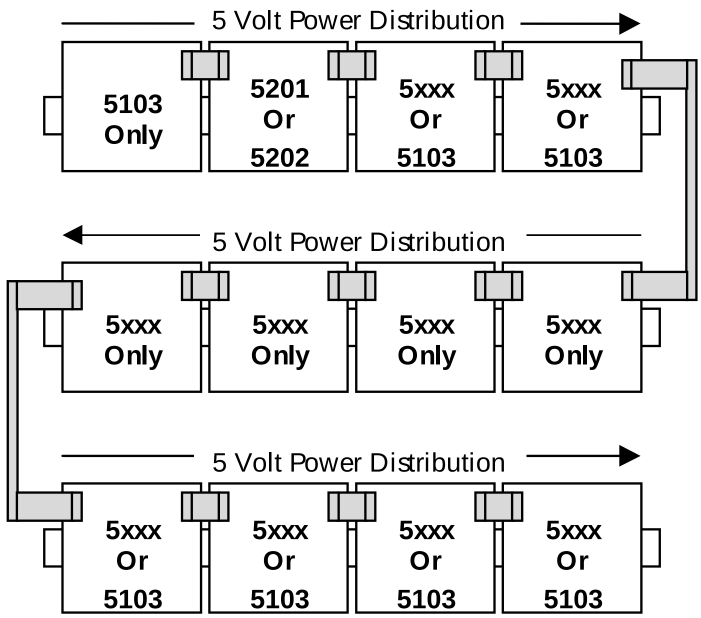


Figure 1: Preferred SmartWIRE Layout

Figure 2: Preferred SCADAPack or TeleSAFE Micro16 Layout shows the preferred multiple rail layout for a SCADAPack or TeleSAFE Micro16 controller system. Two IMC12 cables distribute power and communication signals from rail to rail. I/O modules designated 5xxx refer to any 5000 Series I/O module, except the 5103 power supply. The 5V Power Distribution in the top row of modules Indicates that the SCADAPack or TeleSAFE Micro16 controllers may supply power to I/O modules to their left. This is not true for 5103 power supply modules, they supply power to the right only.

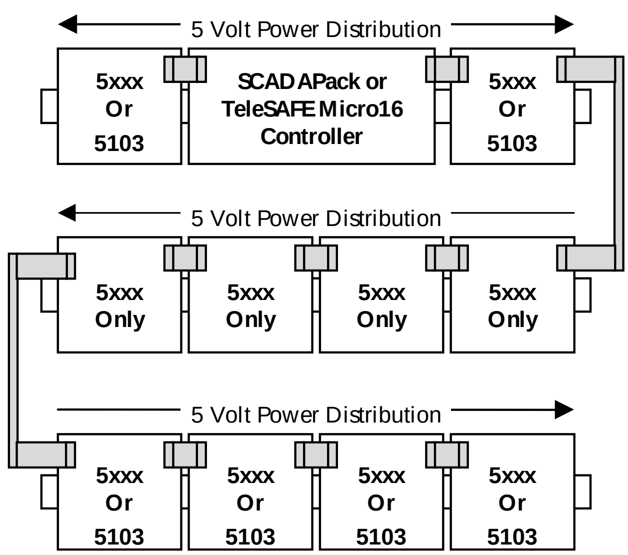


Figure 2: Preferred SCADAPack or TeleSAFE Micro16 Layout

The layouts shown in ***Figure 1: Preferred SmartWIRE Layout*** and ***Figure 2: Preferred SCADAPack or TeleSAFE Micro16 Layout*** are the recommended system layouts as they minimize the IMC length. However, they restrict power supply placement, as 5103 power supply modules cannot be placed in the middle row.

The following diagrams show alternate system layouts. These layouts require the use of IMC30 cables and should be used only if it is essential that a 5103 power supply module be placed in the middle or bottom rows.

Note:	It is always recommended that the system layout use the shortest possible IMC length.

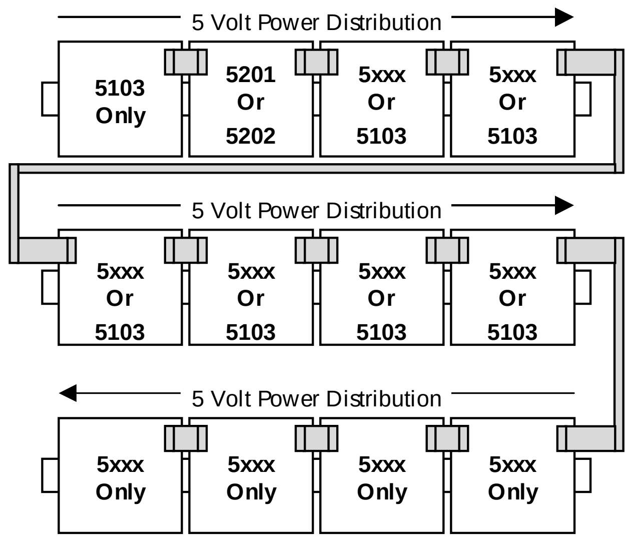


Figure 3: Alternate SmartWIRE Layout

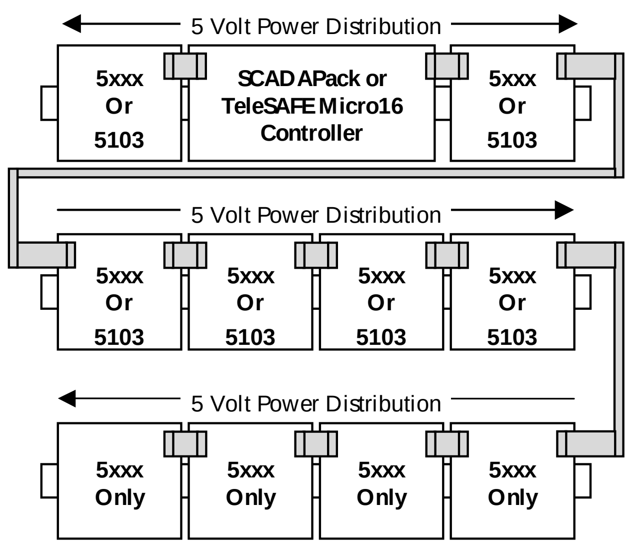


Figure 4: Alternate SCADAPack or TeleSAFE Micro16 Layout

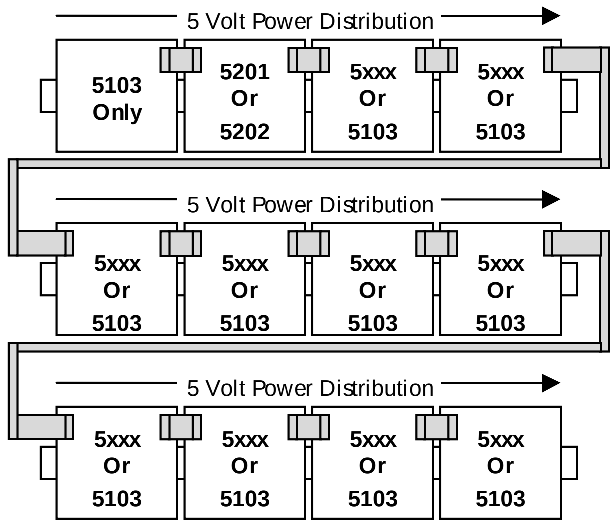


Figure 5: Alternate SmartWIRE Layout

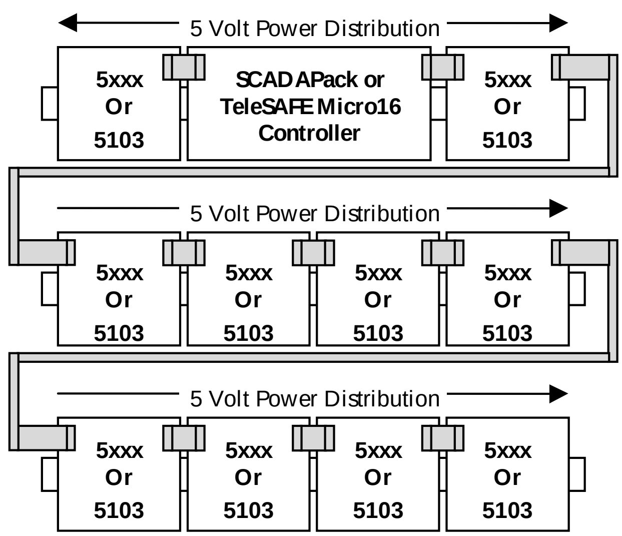


Figure 6: Alternate SCADAPack or TeleSAFE Micro16 Layout

Table of Contents

Table of Figures

Relay Ladder Logic

Relay Ladder Logic has been used in the control of industrial processes for many years. Early control circuits were created by wiring input devices such as pushbuttons and limit switches to output devices such as motors. As the processes became increasingly complicated mechanical relays were added to the control circuits. The use of control relays enabled one area of a process to be controlled by another area.

Control Relays

A typical industrial relay is shown in *Figure 7: Basic Relay*. A relay is a coil of wire wrapped around a core. When the switch is closed current flows through the coil and an electromagnet is created. The armatures, 1 and 4, are drawn from the Normally Closed (NC) connection to the Normally Open (NO) connection.

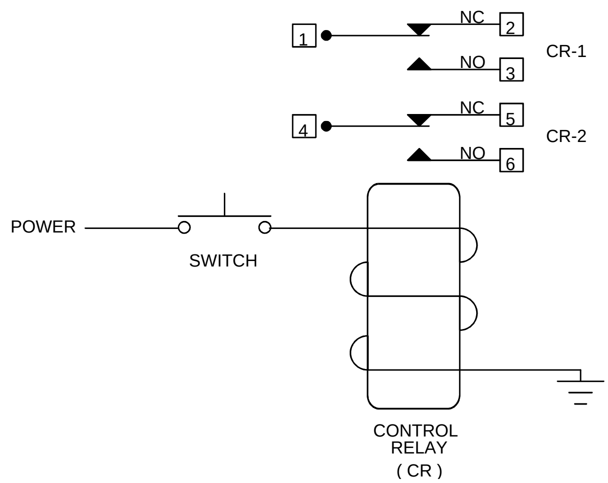


Figure 7: Basic Relay

The relay shown in figure 1 has two sets of contacts, CR-1 and CR-2. When power is applied to the coil, the connection between contacts 1 and 2 of CR-1 is broken and the connection between contacts 1 and 3 is made. The status of the contacts is directly controlled by the current flow through the coil.

► ***Industrial control relays can, and many times do, have large numbers of contact sets. All the contacts of a relay are controlled by the presence or absence of current through the coil of the relay.***

The use of mechanical relays, timers, delay timers and counters enabled the creation of very sophisticated hardwired control circuits.

Simplified Relay Ladder Logic Circuit

An example of a simplified relay ladder logic circuit is shown in *Figure 8: Relay Ladder Logic Circuit*. Each element in the circuit is described as follows:

- SW 1 through SW 4 are switches.
- CR 1 is a control relay with a NO contact.
- START is a normally open pushbutton switch.
- STOP is a normally closed pushbutton switch.
- M1 is a motor with NO and NC contacts.

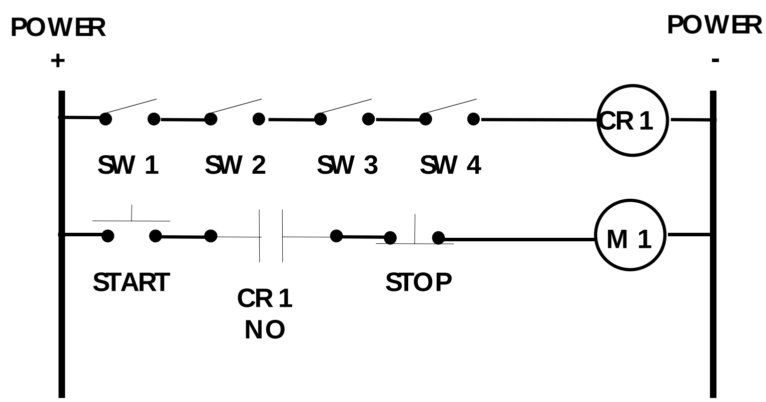


Figure 8: Relay Ladder Logic Circuit

Early Relay Ladder Logic systems had some important limitations:

- The physical size of the control and wiring panels became larger and larger as the complexity of the control systems increased.
- Changes to the control system would involve large scale rewiring of the control and wiring panels.

Programmable Relay Ladder Logic

The TelePACE Relay Ladder Logic, used with Control Microsystems controllers, overcomes the limitations of early relay ladder logic through programmable logic functions. The simplified controller diagram in *Figure 9* shows how the Relay Ladder Logic circuit from exercise 1 would be wired. The physical inputs are wired to digital inputs on the controller. The TelePACE Relay Ladder Logic program operating in the controller will determine whether the MOTOR is started or stopped.

Changing the control system for this example would involve changing the Ladder Logic program. Any combination of switches and pushbuttons can be programmed to turn on or stop the motor. Logic functions such as delay start, motor runtime, number of starts etc. can all be easily added to the control system, without additional wiring.

SW1	**+**	Input		Output	**+**	**MOTOR**
	-	10001		00001	**-**	
SW2	**+**	Input		Output	**+**	
	-	10002		00002	**-**	
SW3	**+**	Input		Output	**+**	
	-	10003		00003	**-**	
SW4	**+**	Input	**TelePACE**	Output	**+**	
	-	10004	**RELAY**	00004	**-**	
START	**+**	Input	**LADDER**	Output	**+**	
	-	10005	**LOGIC**	00005	**-**	
STOP	**+**	Input		Output	**+**	
	-	10006		00006	**-**	
	+	Input		Output	**+**	
	-	10007		00007	**-**	
	+	Input		Output	**+**	
	-	10008		00008	**-**	

Figure 9: TelePACE Relay Ladder Logic

TelePACE Ladder Logic Program

The major components of a ladder program consist of ladder networks, ladder function elements, and associated comments. It is important to fully understand each of these major elements and how they are executed as a program within the controller.

Networks

A network is a diagrammatic representation of control logic similar to a wiring schematic, showing the interconnection of relays, timers, contacts and other control elements. The number of networks in a program is only limited by the memory available.

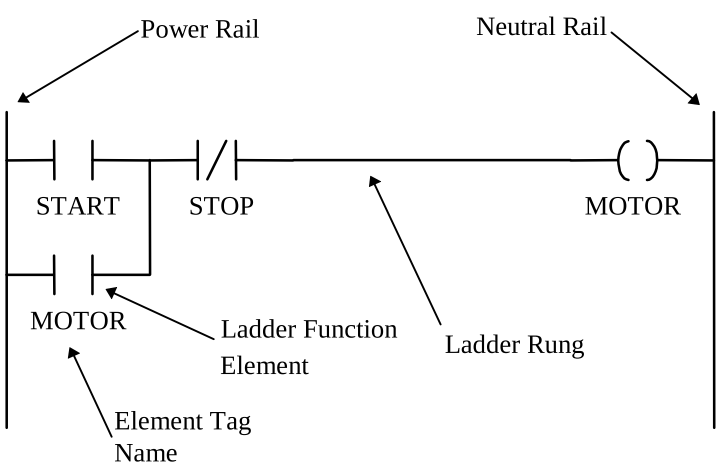


Figure 10: A Ladder Network

Network Elements

Network elements are contacts, coils, and function blocks. Shunts are used to interconnect elements. Coils are always found connected to the neutral rail and represent either physical outputs in the controller or internal (memory only) outputs in the I/O database. Contacts represent either physical status inputs from the controller or internal (memory only) status inputs in the I/O database. Function blocks are used to perform specific functions, such as moving data, manipulating data or communicating data.

Network Comments

The comment editor enables the programmer to fully document an application. Each network may have up to three pages of text documentation. The network title allows for quick access to networks when editing or monitoring programs.

TelePACE Ladder Editor Environment

TelePACE is a powerful programming and monitoring environment, which enables the user to:

- Create ladder logic programs.
- Edit ladder logic programs On or Off line with a Controller.
- Read and write programs to a Controller.
- Configure the Controller serial communication ports.

TelePACE Display

The TelePACE display provides the interface between the programmer and the TeleSAFE controller. The major sections of the TelePACE display are shown in the following diagram.

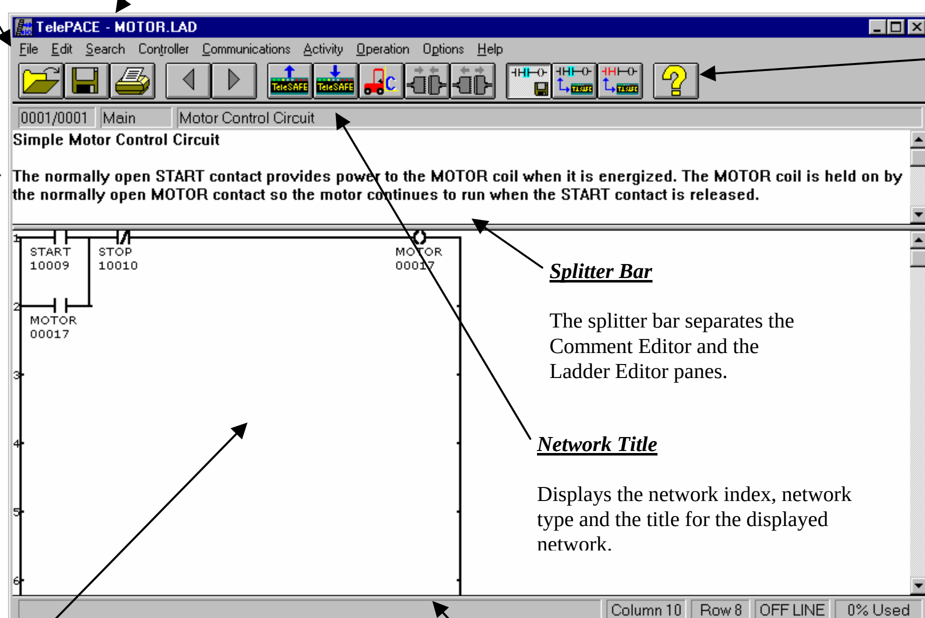
► ***The TelePACE program is fully Windows compatible and as such all normal Windows functions are active when using the TelePACE program.***

Menu Bar

Displays the programming, editing, monitoring and utility functions available.

Title Bar

TelePACE title and the name of the open program file. The file name area is blank if there is no file in use or the file is new and has not been saved.



Tool Bar

File Open
Save
Print
Previous Network
Next Network
Read From Controller
Write To Controller
C Program Loader
Connect To Controller
Disconnect From Controller
Edit Off line
Edit On line
Monitor On line
Help

Comment Editor

Each network in the program may contain approximately three pages of text comments.

Splitter Bar

The splitter bar separates the Comment Editor and the Ladder Editor panes.

Network Title

Displays the network index, network type and the title for the displayed network.

Network Display

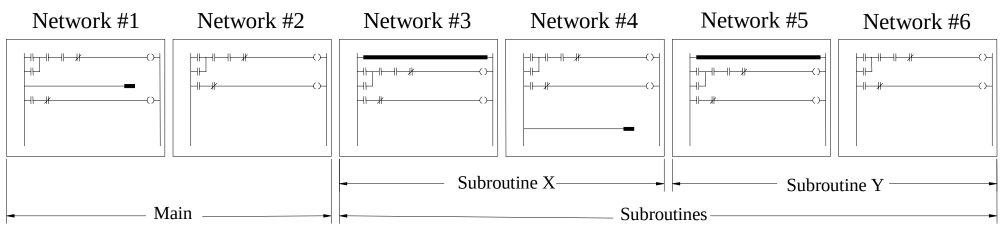
The Ladder Editor Pane displays the current network of the open program. This area is used for creating, editing and monitoring of the ladder logic program.

Status Bar

The status bar displays command and programming status of the Network Editor.

Program Network Layout

The Ladder Logic program consists of the main program followed by a number of subroutines. The main program is defined as all the logic networks up to the start of the first subroutine, or until the end of the program if no subroutines exist. A subroutine is defined as all the logic networks from a subroutine element until the next subroutine element, or the end of the program if there are no more



subroutines.

Figure 11: Program Network Layout

In *Figure 11: Program Network Layout*, the main program consists of networks 1 and 2. The next four networks make up the subroutines, with networks 3 and 4 comprising subroutine X and networks 5 and 6 comprising subroutine Y. If subroutines were not needed, the entire program would consist of a main section only.

Program Execution Order

Networks in the TelePACE Ladder Editor may contain up to 8 rungs. Each rung can contain a maximum of 10 logic elements. The highlighted cursor in the Ladder Editor pane occupies one logic element position.

The controller evaluates each element, or function block, in the network in a sequence that starts at the top left hand corner; or ROW 1, COLUMN 1. The ladder evaluation moves down column 1 until it reaches row 8. The evaluation then continues at the top of column 2 and moves down to row 8 again.

This process continues until the entire network has been evaluated. If there is more than one network in the main program, evaluation continues to the next sequential network in the program until the entire main program has been evaluated. Then the evaluation returns to the first network in the main program.

If a subroutine call function block is encountered, execution transfers to the start of the called subroutine and continues until the end of the subroutine. Execution then returns to the element after the subroutine call element.

Subroutine execution can be nested. This allows subroutines to call other subroutines. Subroutine execution cannot be recursive. This prevents potential infinite loops in the ladder logic program.

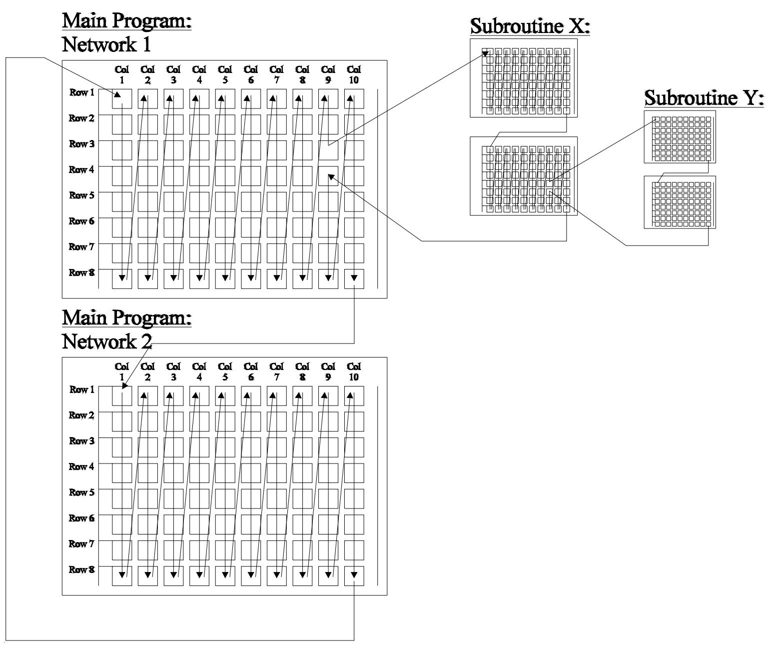


Figure 12: TelePACE Ladder Editor Network Execution

Ladder Logic Memory Usage

Memory usage in a Ladder Logic application program is based on the number of networks and number of elements used by a program. There are 12k words of program space available in all SCADAPack models.

Limiting the number of unnecessary horizontal shunts can reduce the program size. Optimizing other logic, such as math calculations, can also ensure that efficient use is made of the available memory.

Networks	Each network created in an application requires one word of memory, whether the network is used or not.
Columns	Each column that is occupied in a network requires one word of memory.
Single Elements	Each single element requires one word of memory.
Double Elements	Each double element requires two words of memory.
Triple Elements	Each triple element requires three words of memory.

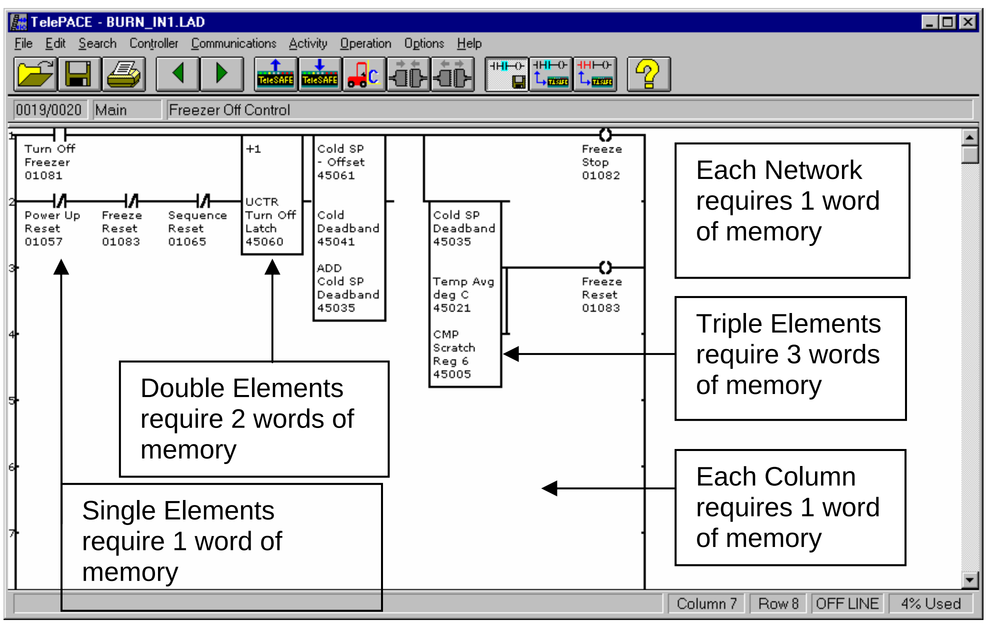


Figure 13: Ladder Logic Memory Usage

I/O Database

The I/O database allows data to be shared between C programs, Ladder Logic programs and communication protocols. A simplified diagram of the I/O Database is shown in *Figure 14: I/O Database Block Diagram.*

The I/O database contains general purpose and user-assigned registers. General-purpose registers may be used by Ladder Logic and C application programs to store processed information and to receive information from a remote device. Initially all registers in the I/O Database are general-purpose registers.

User-assigned registers are mapped directly from the I/O database to physical I/O hardware, or to controller system configuration and diagnostic parameters. The mapping of registers from the I/O database to physical I/O hardware and system parameters is performed by the Register Assignment.

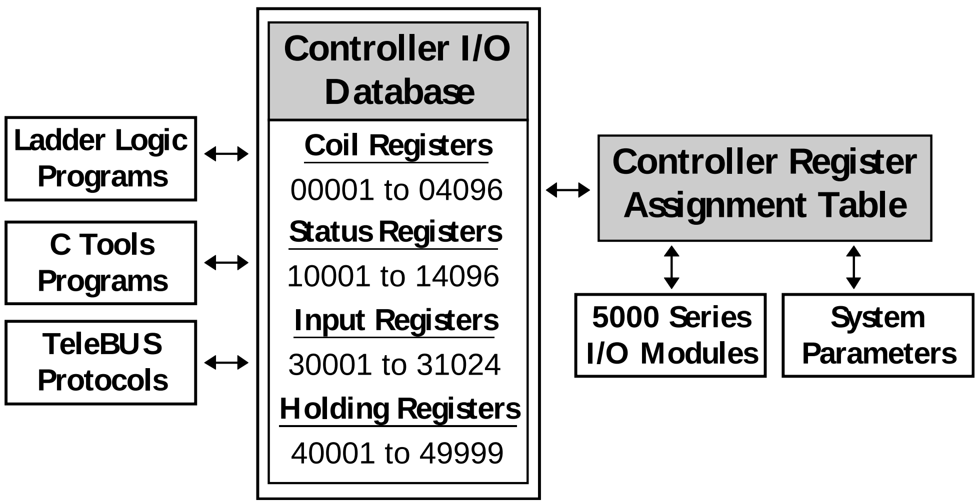


Figure 14: I/O Database Block Diagram

User-assigned registers are initialized to the default hardware state or system parameter when the controller is reset. Assigned output registers do not maintain their values during power failures. Assigned output registers do retain their values during application program loading.

General-purpose registers retain their values during power failures and application program loading. The values change only when written by an application program or a communication protocol.

The TeleBUS communication protocols provide a standard communication interface to the controller. The TeleBUS protocols are compatible with the widely used Modbus RTU and ASCII protocols and provides full access to the I/O database in the controller.

I/O Database register types

The I/O database is divided into four types of I/O registers. Each of these types are initially configured as general purpose registers by the controller.

Coil Registers

Coil registers are single bit registers located in the digital output section of the I/O database. Coil, or digital output, database registers may be assigned to 5000 Series digital output modules or SCADAPack I/O modules through the Register Assignment. Coil registers may be also be assigned to controller on board digital outputs and to system configuration modules.

► There are 4096 coil registers numbered 00001 to 04096. Ladder logic programs, C language programs, and the TeleBUS protocols can read from and write to these registers.

Status Registers

Status registers are single bit registers located in the digital input section of the I/O database. Status, or digital input, database registers may be assigned to 5000 Series digital input modules or SCADAPack I/O modules through the Register Assignment. Status registers may be also be assigned to controller on board digital inputs and to system diagnostic modules.

► There are 4096 status registers are numbered 10001 to 14096. Ladder logic programs and the TeleBUS protocols can only read from these registers. C language application programs can read data from and write data to these registers.

Input Registers

Input registers are 16 bit registers located in the analog input section of the I/O database. Input, or analog input, database registers may be assigned to 5000 Series analog input modules or SCADAPack I/O modules through the Register Assignment. Input registers may be also be assigned to controller internal analog inputs and to system diagnostic modules.

► There are 1024 input registers numbered 30001 to 31024. Ladder logic programs and the TeleBUS protocols can only read from these registers. C language application programs can read data from and write data to these registers.

Holding Registers

Holding registers are 16 bit registers located in the analog output section of the I/O database. Holding, or analog output, database registers may be assigned to 5000 Series analog output modules or SCADAPack analog output modules through the Register Assignment. Holding registers may be also be assigned to system diagnostic and configuration modules.

► There are 9999 input registers numbered 40001 to 49999. Ladder logic programs, C language programs, and the TeleBUS protocols can read from and write to these registers.

Summary of Modbus Register Types

Register Type	Typical Usage *	Address Range	No. of Registers	Register Size	Value Range	Access
Coil Registers	Digital Outputs	00001 to 04096	4096	1 bit	1 or 0 (ON or OFF)	read and write
Status Registers	Digital Inputs	10001 to 14096	4096	1 bit	1 or 0 (ON or OFF)	read only
Input Registers	Analog Inputs	30001 to 39999 **	9999	16 bits	0 to 65535 in Unsigned format *** -32768 to 32767 in Signed format	read only
Holding Registers	Analog Outputs	40001 to 49999	9999	16 bits	0 to 65535 in Unsigned format *** -32768 to 32767 in Signed format	read and write

* All registers may also be used as general purpose registers to simply save data in memory (e.g. memory for program data, internal variables).

** 30001 to 39999 for SCADAPack 32

30001 to 31024 for SCADAPack controllers

*** When 2 sequential 16-bit registers are used together, the following 32-bit registers are possible:

32 bit Register Format	Value Range
Unsigned Double	0 to 4,294,967,295
Signed Double	-2,147,483,648 to 2,147,483,647
Floating Point	-3.402×10^{38} to 3.402×10^{38} Six digits of precision

Register Assignment

The Register Assignment is used to assign I/O database registers to user-assigned registers using I/O modules. An I/O Module can refer to an actual I/O hardware module such as a ***5401 Digital Input Module*** or it may refer to a set of controller parameters, such as serial port settings.

5000 Series I/O modules used by the controller must be assigned to I/O database registers in order for the I/O points to be used by the ladder program.

The configuration and diagnostic I/O modules used by the controller must be assigned to I/O database registers in order for the I/O points to be used by the ladder program.

► Ladder logic programs may read data from, or write data to the physical I/O only through user-assigned registers in the I/O database.

I/O Module Types

AIN ***Analog Input*** modules are used to assign data from physical analog inputs to input registers in the I/O Database. Physical analog inputs are specific 5000 Series analog input modules, generic analog input modules, and internal controller data such as RAM battery voltage and board temperature.

AOUT ***Analog Output*** modules are used to assign data from the I/O Database to physical analog outputs. The physical analog outputs are specific 5000 Series analog output modules or generic analog output modules.

DIN ***Digital Input*** modules are used to assign data from physical digital inputs to input registers in the I/O Database. Physical digital inputs are specific 5000 Series digital input modules, generic digital input modules, and controller digital inputs.

DOUT ***Digital Output*** modules are used to assign data from the I/O Database to physical digital outputs. Physical digital outputs are specific 5000 Series digital output modules or generic digital output modules.

CNTR ***Counter Input*** modules are used to assign data from physical counter inputs to input registers in the I/O Database. Physical counter inputs are specific 5000 Series counter input modules and controller counter inputs.

DIAG ***Controller Diagnostic*** modules are used to assign diagnostic data from the controller to input or status registers in the I/O Database. The diagnostic data is used to monitor internal controller data such as controller status code, the force LED, serial port communication status and serial port protocol status.

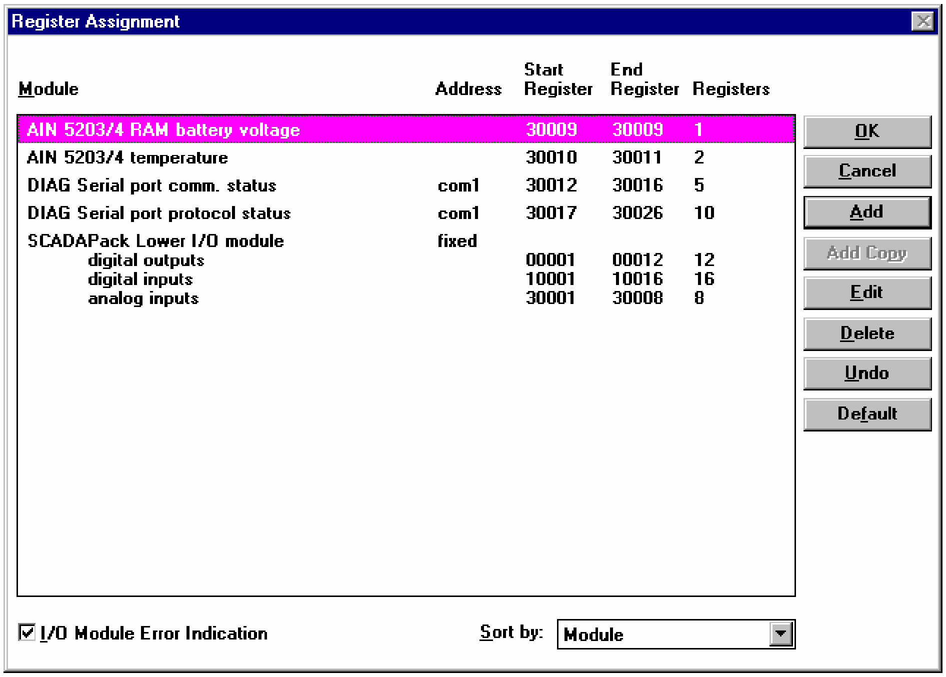
CNFG ***Controller Configuration*** modules are used to assign data from I/O Database coil and holding registers to controller configuration registers. The configuration data is used to configure internal controller settings such as clearing protocol and serial counters, real time clock settings and PID control blocks.

SCADAPack modules are used to assign data to registers in the I/O Database, from physical SCADAPack digital and analog I/O. Physical inputs and outputs are specific SCADAPack I/O modules.

Register Assignment Display

The Register Assignment display provides the interface between the I/O Database and the physical I/O devices used by the TeleSAFE controller. The major sections of the Register Assignment display are shown in the following diagram.

► Only the I/O Modules that are defined in the Register Assignment have a connection to Database I/O registers.



I/O Module Error Indication

The Register Assignment dialog contains a check box selection for *I/O Module Error Indication*. The *I/O Module Error Indication* check box determines if the controller displays I/O module communication errors. If enabled, the controller will blink the Status LED if there is an I/O error. If disabled, the controller will not display the module communication status.

The module communication status is always checked. This option controls only the indication on the Status LED.

I/O Module Sorting

The Register Assignment listing may be sorted by Module, Start Register or Address. The sort order is selected from the drop-down box at the right of the dialog.

Default Register Assignment Selection

The default register assignment selection replaces the current Register Assignment with the Default Register Assignment for the controller. The default selection allows for backward compatibility with controllers that do not support register assignment.

Assigning Registers to I/O Modules

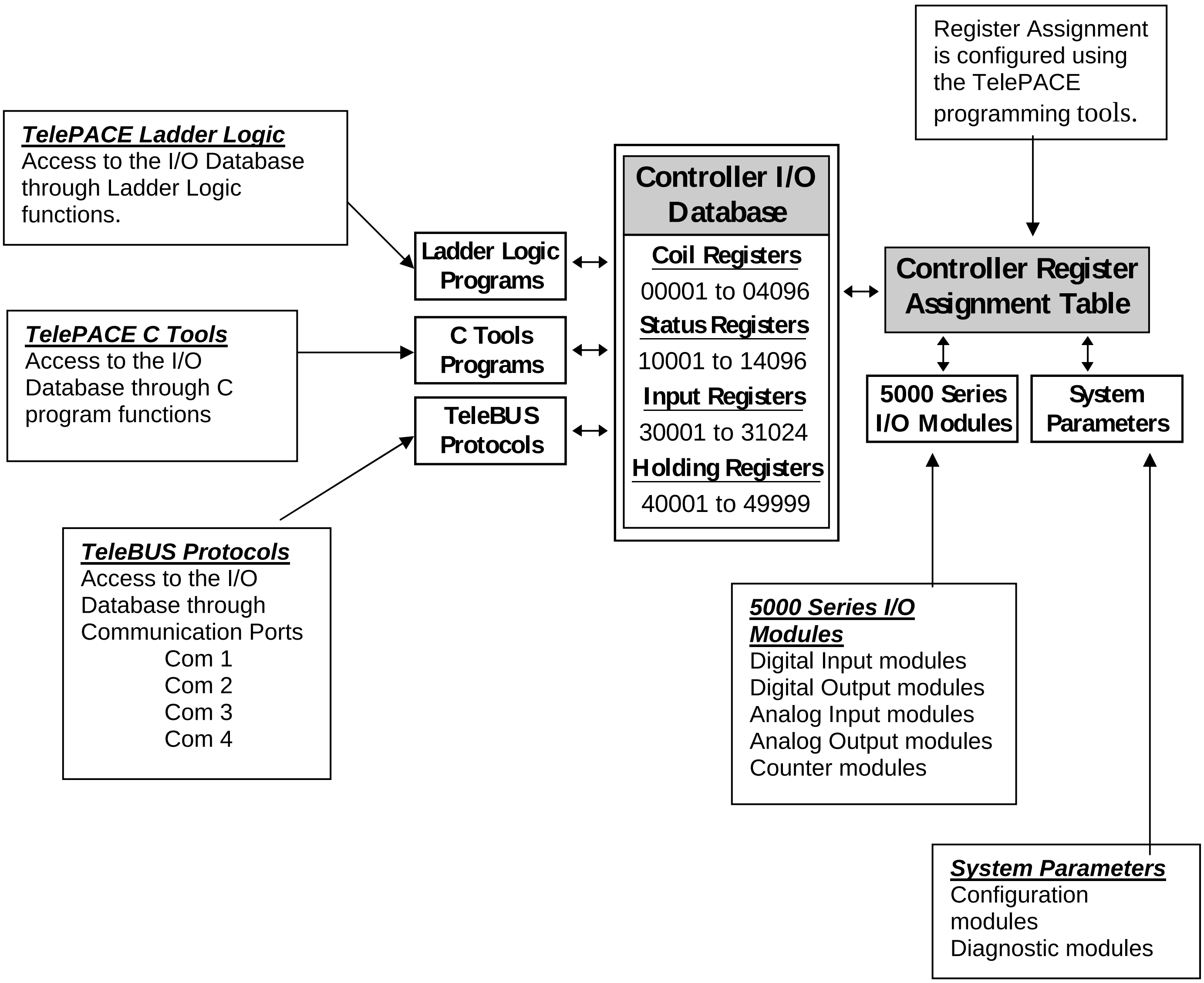
The I/O module reference is found in the TelePACE Ladder Logic User Manual or in the Help menu of the TelePACE Ladder Editor. The register assignment table in the I/O module reference is used to describe the information required in the ***Add Register Assignment*** dialog.

The information required for each section of the ***Add Register Assignment*** dialog is explained in the following table. The shaded column at the left of the table indicates the information that is required for the I/O module in the ***Add Register Assignment*** dialog. The requirements vary depending on the I/O module.

Each part of the register assignment table is explained in the following table.

Module	I/O module highlighted in the **Module** window of the **Add Register Assignment** dialog.
Address	For hardware I/O modules this is the physical address of the 5000 Series I/O module. This address is set to be equal to the address of 5000 Series I/O module. For Configuration and Diagnostic I/O modules this is the I/O module address or the communication port, depending on the I/O module selected. For I/O modules that do not require an address this selection is grayed out and no entry is allowed.
Type	The I/O database register type that the I/O module will assign data to. The I/O database register types are: Coil Register **0xxxx** Status Register **1xxxx** Input Register **3xxxx** Holding Register **4xxxx**
Start	The I/O database register address where the I/O module begins the register assignment.
End	The I/O database register where the I/O module ends the register assignment. This value is automatically set to **Start + number of registers required** by the I/O module.
Registers	Number of I/O database registers that the I/O module assigns.
Description	The register type description for I/O modules that assign multiple types of registers.

Register Assignment Overview



Ladder Logic Program Development

To demonstrate the features of the TelePACE Ladder Editor a simple Ladder Logic program will be created. This program development will lead you through the steps required to create a TelePACE Ladder Logic application program. Each of the following steps will be explained in the following chapters.

Configure 5000 Series I/O modules used in the application.

Configure controller for Service Mode.

Start a new Ladder Logic Program.

Configure TelePACE Program Editor.

Install serial connection between PC and Controller.

Select Controller type.

Initialize the Controller.

Configure Controller Register Assignment.

Add Configuration and Diagnostic I/O modules.

Configure Outputs-On-Stop settings.

Configure Controller serial port settings.

Create and comment the Ladder Logic program.

Save the TelePACE Ladder Logic program.

Write the Ladder Logic program to the controller.

Run and test the Ladder Logic program.

Motor Control Circuit Application

To demonstrate the steps required developing a ladder logic program using TelePACE we will develop a simple ladder logic application. *Figure 15* shows a simple motor control circuit for this application.

Closing the START contact applies power to the motor control coil. A contact on the motor coil holds on the power when the START contact is released. Pressing the STOP switch opens the normally closed STOP contact, and removes power from the motor control coil.

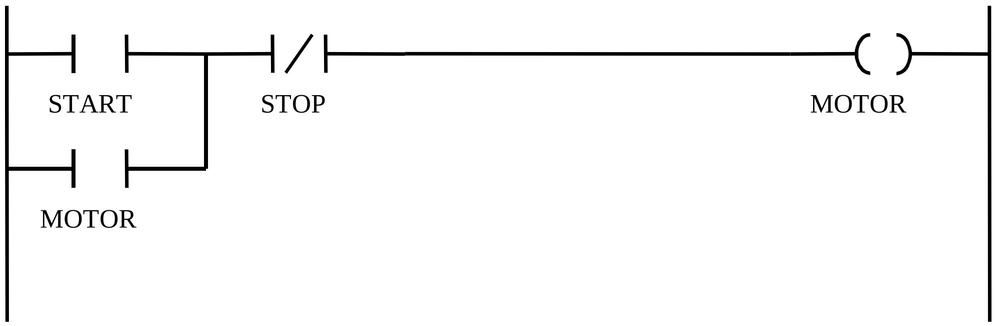


Figure 15: Motor Control Circuit

? ***This Motor Control circuit requires ___digital inputs and ___ digital outputs?***

Configure 5000 Series I/O modules used in the application.

5000 Series I/O modules used in an application must be assigned an individual hardware address for each type of module. This addressing is selected using DIPswitch settings on the 5000 Series I/O module. Depending on the type of I/O module the module address will be 0 through 8 or 0 through 16.

The 5000 Series I/O bus will support a maximum of forty I/O (input/output) modules. 5000 Series I/O module types may be combined in any manner to the maximum supported by the controller used. The types of input and output modules available are:

- Digital Input modules
- Digital Output modules
- Analog Input modules
- Analog Output modules
- Counter Input modules

Each type of I/O module, connected to the I/O bus, must have a unique I/O module address. Different types of I/O modules may have the same module address.

For this training session the 5699 I/O simulator is used with the SCADAPack controller. The SCADAPack I/O is configured using the Register Assignment and does not require physical addressing.

Configure Controller for Service Mode

When a new controller is first used it should be started in Service mode. Service mode is selected by performing a ***Service Boot*** using the following procedure:

- Remove power from the controller.
- Press down on the LED POWER button.
- Apply power to the controller.
- Continue holding the LED POWER button until the STAT LED starts turns on.
- Release the LED POWER button.

► If the LED POWER button is released before the STAT LED turns on, the controller will start in RUN mode.

When the controller starts in SERVICE mode:

- default serial communication parameters of Modbus RTU, 9600 baud rate, no parity, 8 data bits, 1 stop bit and station address 1 are used for all communication ports;
- the Ladder Logic program is stopped;
- the C program is stopped;
- all programs are retained in non-volatile memory;

? ***What is the reason for performing a service boot?***

Open a New TelePACE Ladder Editor File

To start a new TelePACE Ladder Logic program:

- From the **File** menu command select **New**.

The comment editor window and the ladder editor window are cleared and the network display shows network one of one.

The TelePACE Ladder Network Editor retains all information from the previous editor settings. When the TelePACE Ladder Editor is first used after it has been installed a tple.ini file is created. This file contains all the Editor settings that were in use when the program closed. These settings are used when the TelePACE Ladder Editor is next used. Any changes are again used on the next start up of the program.

Configure TelePACE

The configuration of TelePACE involves setting the serial port parameters for the target controller and selecting options to customize the Ladder Editor environment.

PC Communication settings

Configure Ladder Editor communications to satisfy the requirements of the communication media between the PC running the TelePACE software and the target controller. Enter the parameters used by your personal computer (PC) to communicate with the controller using the **PC Communication Settings** command on the **Communications** menu.

- Select the **No Flow Control** radio button to select an RS-232 connection that does not require the hardware control lines on the serial ports.
- Select the serial **Port** your PC will use for communication to the TeleSAFE controller.
- Select **Modbus RTU** for the communication **Protocol**.
- Select **9600** for the **Baud** rate.
- Select **none** for **Parity** type.
- Select **8 bits** for the number of **Data Bits**.
- Select **1 Bit** for the number of **Stop Bits**.
- Enter address **1** in the **Station** address selection box.
- Enter **3** seconds in the TeleSAFE response **Time Out** selection box.
- Enter **3** attempts in the number of **Attempts** selection box.

► The default settings of Modbus RTU protocol, 9600 baud rate, no parity, 8 data bits, 1 stop bit and a station address of 1 are the same settings the Controller will have when it is started in Service Mode.

Options

Set the Ladder Editor environment options to suit the display resolution and color capability of your computer.

- From the **Options** menu select **Screen Font** to change the Editor screen font and **Colors** to change the Ladder Logic display color.
- The **Tool Bar**, **Title Bar** and the **Status Bar** may be removed from the Editor Display by selecting or de-selecting the option in the **Option** menu.
- Select tag names display options to meet your personal preference. The selections are **Single Tag Names**, **Double Tag Names**, **Tag and Address** and **Numeric Address**.

In most cases it is recommended that the **Allow Multiple Coils** and **Warning Messages** be selected.

Install serial connection between PC and Controller

An RS-232 serial communication cable is required to connect the PC serial port to any of the controller serial ports.

The RS-232 communication ports on the controller are 9 pin male D-sub-miniature connector (DE-9P) configured as Data Terminal Equipment (DTE).

Generally the RS-232 communication port on the PC will either be a 9 pin or a 25 pin male connector configured as Data Terminal Equipment (DTE).

A **null modem** cable is required to connect two RS-232 DTE devices.

Select Controller Type

The type of controller used in the application is selected from the ***Types*** selection in the ***Controller*** menu. There are nine types of controllers available for selection:

Micro16
SCADAPack
SCADAPack Light
SCADAPack Plus
SCADAPack LP
SCADAPack 32
SCADAPack 32P
SCADAPack 100
4202GFC

A check mark beside the controller type indicates the selection.

Initialize the controller

The initialize command is used to restore a controller to default settings. This is typically done when starting a new project with a controller.

- From the **Controller** menu select **Initialize**.
- Select the items to initialize in the **Initialize Controller** dialog.
- Check **Erase Ladder Logic Program** to clear the ladder logic program memory in the controller.
- Check **Erase C Program** to clear C application program memory in the controller.
- Check **Initialize Controller** to reset all values in the I/O database to default settings.
- Check **Erase Register Assignment** to remove all register assignments.

► The **Initialize Controller** selection:

- Removes all register forces in the controller.
- Sets controller serial port configurations to the default settings.
- Sets all I/O database registers used for Element Configuration to zero.

Configure Controller Register Assignment

Register assignments are stored in the user configured Register Assignment and are downloaded with the ladder logic application program. Using register assignment is simply a process of adding and configuring I/O modules.

The steps required to configure the register assignment differ depending on the controller used.

SCADAPack

The SCADAPack controller may use on-board I/O or 5000 Series I/O modules. The 5699 I/O simulator is used in this training session to simulate the on-board I/O of the SCADAPack controller.

To configure the SCADAPack register assignment:

- From the **Controller** menu select **Register Assignment.**

The **Register Assignment** dialog is displayed. The Register Assignment is initially empty when a new file is started.

- From the buttons at the right of the Register Assignment dialog select the **Add** button.

The **Add Register Assignment** dialog appears. The first I/O module in the I/O module list is displayed and highlighted in the **Module** window.

- Click on the drop down arrow at the right of the **Module** window.

The **Module** window now displays alphabetically all I/O modules that are available.

- Use the scroll bar control to scroll down the list to the **SCADAPack 5601 I/O module.**
- Click the mouse button on this module. This closes the module list and the **Add Register Assignment** now displays the entries required for the **SCADAPack 5601 I/O module**.
- Enter 1 into the 0xxxx **Start** box.

The number 12 will automatically appear in the **End** column.

? ***What does this number represent?***

- Enter 10001 into the 1xxxx **Start** box.

The number 10016 will automatically appear in the **End** column.

? ***What does this number represent?***

- Enter 30001 into the 3xxxx **Start** box.

The number 30008 will automatically appear in the **End** column.

? ***What does this number represent?***

- Click on the **OK** button to close the **Add Register Assignment** dialog box.
- Click on the **OK** button to close the **Controller Register Assignment** dialog box.

Add Configuration and Diagnostic I/O Modules

The controller contains a number of internal configuration and diagnostic I/O modules that may be required by the application.

Controller configuration I/O modules are used to assign data from I/O Database coil and holding registers to controller configuration registers. The configuration data is used to configure controller settings such as clearing protocol and serial counters, real time clock settings and PID control blocks.

Controller diagnostic I/O modules are used to assign diagnostic data from the controller to input or status registers in the I/O Database. The diagnostic data is used to monitor internal controller data such as controller status code, the force LED, serial port communication status and serial port protocol status.

For each configuration or diagnostic module used in the application:

- Select the desired I/O module from the module list by clicking the left mouse button on the module text. The selected module is highlighted and the module list is closed.
- Enter the required information for the I/O module in the **Add Register Assignment** dialog.

Refer to the Register Assignment section of this manual, or to the Help menu, for details on the information required by the **Add Register Assignment** dialog.

Configure Outputs On Stop

The controller analog and digital outputs are configured to maintain their value, or go to the default state when the ladder logic program execution is stopped.

To configure the outputs on stop setting:

- Select **Outputs on Stop** from the **Controller** menu and the **Output Conditions on Program Stop** dialog appears. This dialog controls the state of the controller analog and digital outputs when the ladder logic program is stopped.
- The state of the digital outputs may be set to **Hold** their last value or to turn **Off** when the ladder program is stopped.
- The state of the analog outputs may be set to **Hold** their last value or to go to **Zero** when the ladder program is stopped.

? ***What is the effect of these settings?***

Configure Controller Serial Port Settings

The communication settings for each serial port on the controller must be configured. There is a set of port settings for each serial port on the controller. The port settings selected will depend on the type of application the controller is being used.

► The controller serial port that is used by TelePACE for programming must be configured with the same settings as the PC Serial Port Settings.

To configure the serial port settings:

- From the **Controller** menu select **Serial Ports**.
- Enter the required serial port settings for each serial port in the **Controller Serial Ports Settings** dialog.
- If the application does not require the serial port it is recommended that the default settings for the serial port be used.

Create and Comment Ladder Logic Program

Develop the TelePACE Ladder Logic program to meet the requirements of the application. It is strongly recommended that program comments be added as the program is developed. When the TelePACE Ladder Editor is started it is initially in the **Edit Off line** mode.

The **Edit Off line mode** enables editing of a ladder logic program without a connection to a controller. The editing commands affect the program file in use in the TelePACE Network Ladder Editor. All editing commands can be used in this mode.

Creating Tag Names

Tag names describe I/O points by names that are familiar to you. They make a ladder logic program much easier to read.

The first tag for the motor control example is START. The tag name can have a maximum of 16 characters. The **Tag Display** box will show 8 characters on each of two lines. This box shows how the tag name will be displayed throughout the Ladder Logic program.

- Select **Tag Names** from the **Edit** menu. The **Edit Tag Names** dialog appears (see *Figure 16*).

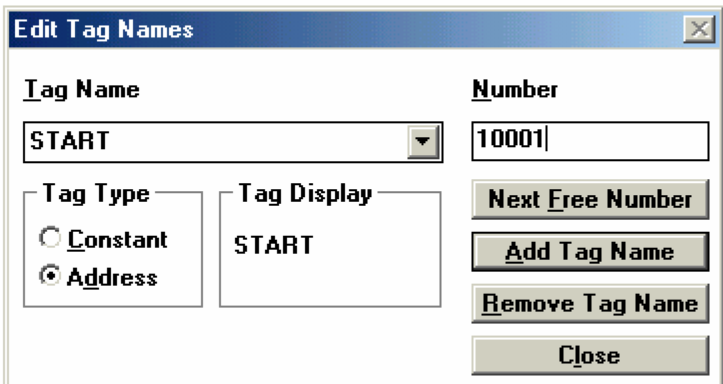


Figure 16: Edit Tag Names Dialog Box

- Type a tag name into the **Tag Name** edit box.

The address of START can be any of the input addresses. To assign START to the first Digital Input address enter 10001 in the **Number** edit box.

- Move to the **Number** edit box and enter the address number.

The tag name START is to be assigned an I/O address of 10001. Select Address by clicking the left mouse button on the **Address** radio button.

- The **Tag Type** selection box assigns the value in the Number edit box as an **Address**.
- Click on the **Add Tag Name** button.

The tag name STOP is to be assigned an I/O address of 10002. Select Address by clicking the left mouse button on the **Address** radio button.

- The **Tag Type** selection box assigns the value in the Number edit box as an **Address**.
- Click on the **Add Tag Name** button.

The tag name MOTOR is to be assigned an I/O address of 00001. Select Address by clicking the left mouse button on the **Address** radio button.

- The **Tag Type** selection box assigns the value in the Number edit box as an **Address**.
- Click on the **Add Tag Name** button.
- Click on the **Close** button to close the dialog box.

Entering Ladder Logic Networks

Ladder logic networks define the control actions of the controller. Networks are entered using the mouse and the keyboard.

- Move the mouse pointer to the Ladder Editor pane.
- Click the left button on the mouse and the Ladder cursor is displayed. Move the cursor to the position of Column 1 and Row 1.
- Click on the position with the mouse **OR** use the arrow key on the keyboard.

? ***The cursor occupies __________ position (s) in the Ladder Editor pane?***

Double click the left mouse button on this position or press the insert key on the keyboard with the cursor in this position. The *Insert/Edit Network Element* Dialog (*Figure 17*) appears.

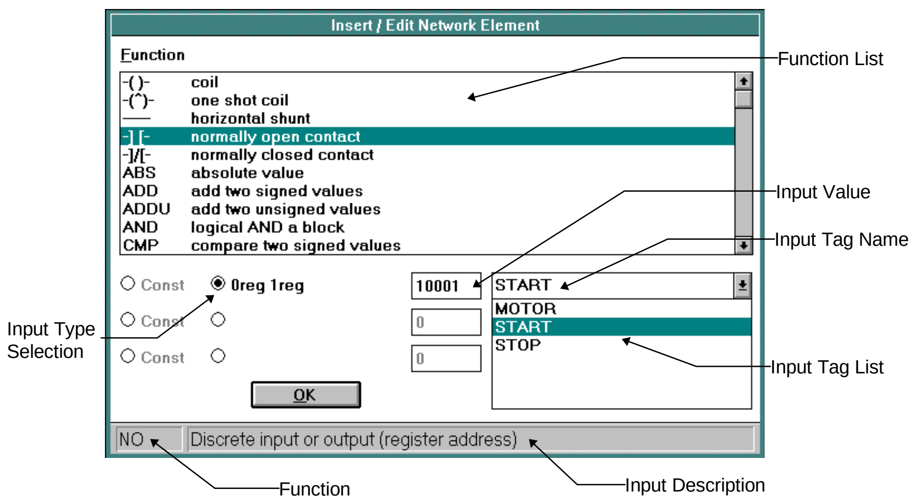


Figure 17: Insert/Edit Network Element

The **Function** list box shows all functions that are valid for the current position. The functions are listed alphabetically, ***except for the _____________________ functions***.

Select a function from the list. The first element for the example is a normally open (NO) contact (see *Figure 15*).

- Click on **normally open contact** in the **Function** list.
- Click on the drop down arrow on the **Input Tag Name** list box to display tags that apply to the current input type. Select a tag from the list. The first element in the example is the START input. Select START from the list box.

Click on the **OK** button to insert the element.

Repeat these steps for each element of the logic network.

- Position the mouse pointer over the position to the right of the first element. Follow the steps above to insert a **normally closed contact** with the address STOP.

Elements may also be entered from the keyboard.

Press the **right arrow** key. The edit cursor will move one element to the right. Press the **Insert** key. The *Insert/Edit Network Element* Dialog (*Figure 17*) appears.

- Follow the steps above to insert a coil with the address MOTOR.
- Position the mouse pointer over the position under the first element. Follow the steps above to insert a normally open contact with the address MOTOR.

To insert vertical shunts, position the mouse pointer over the element where a shunt is required, and click on the right mouse button.

- For the motor control example, insert a vertical shunt on the START contact element.

Adding Comments

Text can be inserted into the Comment Editor pane that describes the logic in the network. This is a valuable documentation tool that helps make programs easier to understand in the future. The comments entered are for each network. The comment editor is blank when a new network is created.

To enter network comments:

- Position the mouse pointer in the comment editor pane and click the mouse button.
- Size the comment editor pane to view the amount of text you require.
- Enter the text you wish. The comment editor uses the Windows system font for the text.

Another important documentation feature is the Network Tile. The Network Title describes the network and is useful when using the Go To Network command.

To enter network titles:

- Double click on the Network Title field in the Network Status Bar. This field is the second from the left, beside the network display.
- The Edit Network Title box appears and a 28-character name for the network can be entered.

Save the Ladder Logic Program

You can save your ladder logic program to disk at any time. Saving your programs as you work protects you against power failures. We recommend making backup copies of your completed programs and storing them in a safe place.

To save the program:

- Position the mouse pointer over the Open File button on the tool bar and click the left mouse button or select **Save** from the **File** menu. The standard Windows **Save As File** dialog (Figure 18) appears because the file has no name. If the file has a name, it is saved, and no dialog appears.

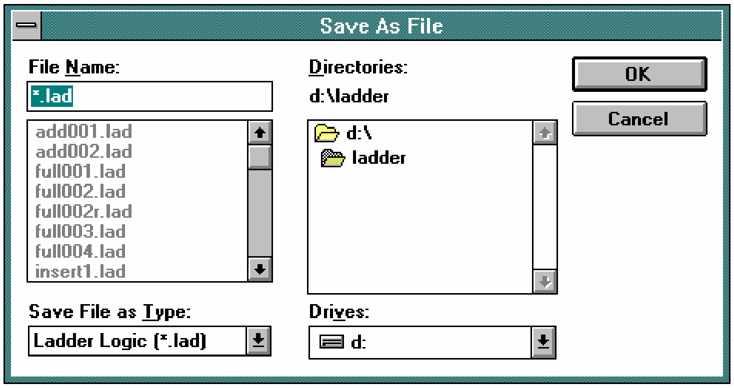


Figure 18: Save As File Dialog

- Enter the file name. For this example enter **MOTOR**. It is not necessary to enter the extension LAD. It is supplied automatically, if you do not type it.

Click on the **OK** button. The file is saved a **MOTOR.LAD**.

Write Ladder Logic Program to the Controller

The ladder logic program created in the TelePACE Ladder Editor must be written to the target controller before program execution can occur.

- When a ladder logic program is written to a controller it replaces any ladder logic program in the controller.

If the program in the controller is executing a dialog box will request whether to stop execution of the new program when the write is complete or to continue execution of the new program after the write is complete.

The controller executes the ladder logic program. To write a program to the controller:

- Select **Write to Controller** from the **Communications** menu or click on the Write-to-TeleSAFE icon on the tool bar. The **Communication Progress** dialog *(Figure 19)* appears.

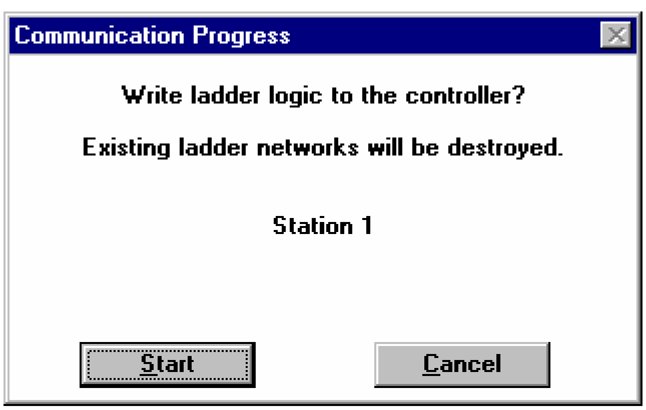


Figure 19: Write to TeleSAFE Communication Progress Dialog

- Click on the **Start** button.
- The program is written to the TeleSAFE controller. The download progress is shown in the **Serial Communication** dialog.

Run the Ladder Logic Program

Once the program is written to the controller the controller must be put in the **Run** mode.

- From the **Operation** menu select **Run**.
- The RUN LED on the controller will indicate that the controller is in the **Run** mode.

► The ladder logic program in a SCADAPack or TeleSAFE Micro16 controller is not executing if the RUN LED is not on.

Monitoring Execution

The last step in completing a ladder logic application program is to test the program execution to ensure it meets the requirements of the application. The Monitor On line mode enables the real time monitoring of a program executing in a controller.

The TelePACE Ladder Editor displays the power-flow through the network on the screen. No changes can be made to the program while in monitor mode. The Register Editor is displayed on top of the Ladder Editor.

To monitor a program on line :

- Select **Monitor On line** from the **Activity** menu or click on the monitor on line button on the toolbar.

A **Communication Progress** dialog similar to the figure below appears.

Figure 20: Communication Progress Dialog

- Click on the **Start** button.

When the start button is clicked the controller checks the program that is loaded into memory with the program in TelePACE. If the programs are not exactly similar an error message is displayed.

To test the motor control program:

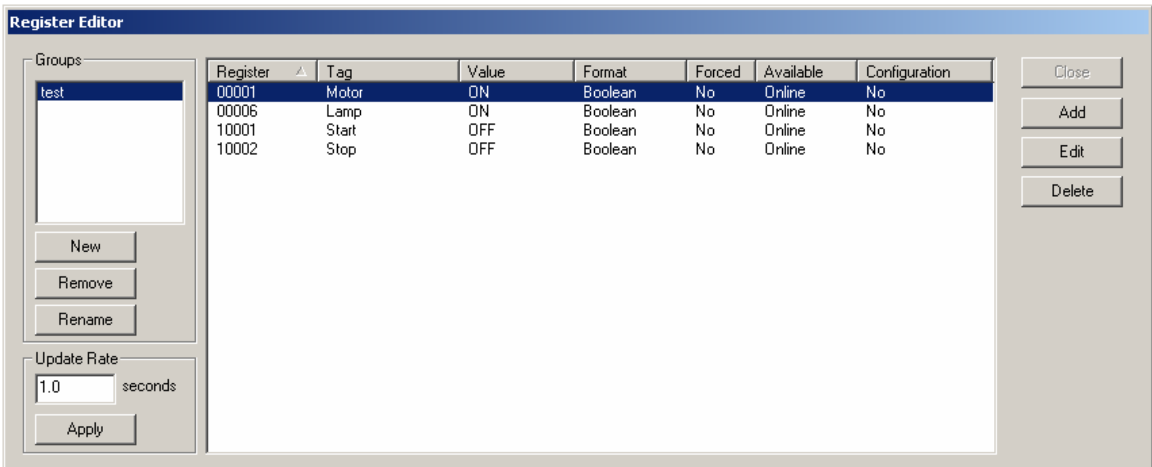
- Apply power to the START input. Observe that the MOTOR output is on. Observe the power-flow through the contacts to the MOTOR coil.
- Remove power from the START input. The MOTOR hold in contact maintains the power-flow to the MOTOR coil. Observe that the output of the START contact is not powered, indicating that power is not flowing through the contact.
- Apply power to the STOP input. Observe that the MOTOR output is off. Observe there is no power-flow through the contacts to the MOTOR coil.

- Remove power to the STOP input.

The motor control circuit is functioning properly.

Monitoring Registers On line

The **Register Editor** enables you to create a list of discrete points or registers that will be displayed in real time on top of the Ladder Editor pane when the Ladder Editor is in the **Debug** operation and **Monitor On line** activity.



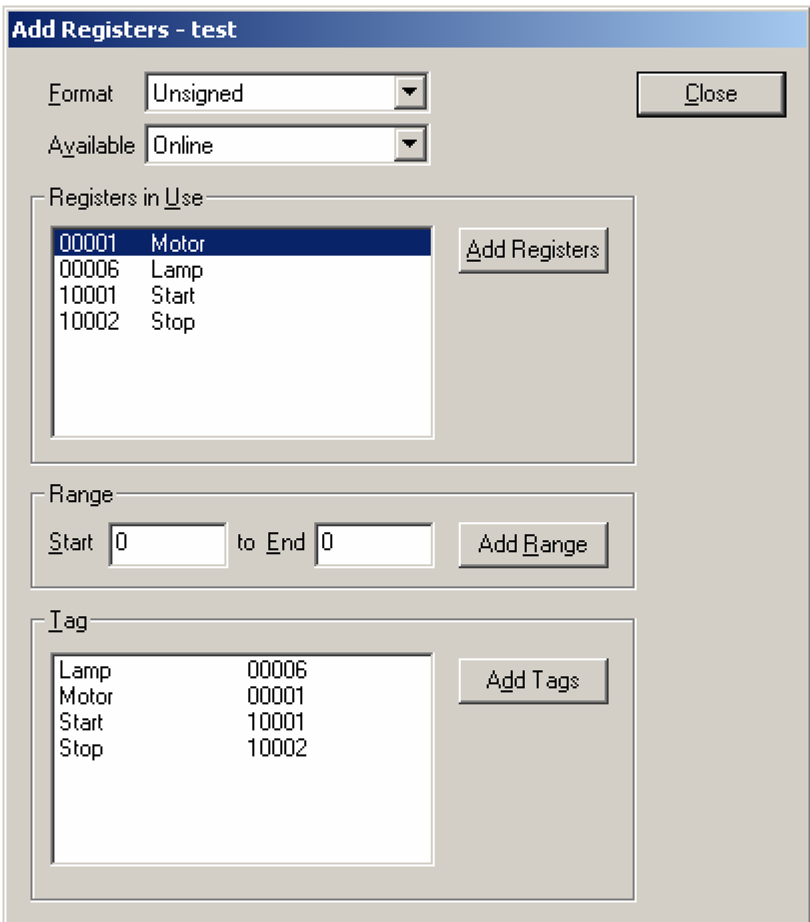
- Select **Monitor On line** from the **Activity** menu or click on the monitor on line button on the toolbar.
- From the **Register Editor** dialog select the **New** button from the **Groups** list box.

The **Groups** dialog box is displayed. Enter a name to identify a group of registers to monitor and select **OK**.

- From the **Register Editor** dialog select the **Add** button.

The **Add Registers** dialog adds registers to the current group in the Register Editor dialog. Registers may be selected in one of three ways:

1. **Register in Use** – a list of all registers that are used in the Ladder Logic program. To select registers from this list, highlight one or more registers and select the **Add Registers** button.
2. **Range** – The **Start** and **End** edit boxes specify a range of registers to add.
3. **Tag** – The **Tags** list box displays all tags and the corresponding register number that are defined in the Ladder Logic program. To select tags from this list, highlight one or more tags and select the **Add Tags** button.



Use the **Format** list box to select a register format other than the default before adding a selection of registers.

Contact Power-flow Monitoring

It is often necessary, while in Monitor On line mode, to determine whether a contact would pass power if power were supplied to it. This feature is extremely useful when testing the operation of ladder logic programs.

In Monitor On line mode, TelePACE shows the power-flow through the network. It also colors contacts that are not powered to show how power would flow if they were.

Figure 21: Contact Power-flow Monitoring shows how power-flow information is displayed in Monitor On line mode.

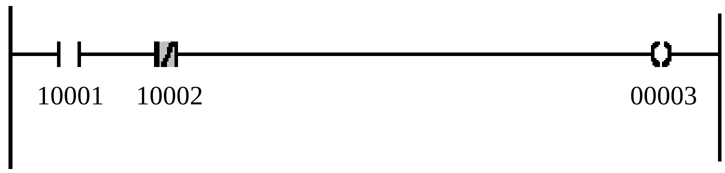


Figure 21: Contact Power-flow Monitoring

In *Figure 21*, normally open contact 10001 and normally closed contact 10002, are not energized. The shaded background, in contact 10002, indicates power would flow through the contact if the input side were powered.

- The background of a normally open contact is colored when it is energized and its input is not powered.
- The background of a normally closed contact is colored when it is not energized and its input is not powered.

The background is only displayed on contacts if the input side of the contact is not powered.

The color used for contact monitoring can be changed using the *Options* | *Colors* command.

Editing a Program On line

Changes are often required to a program already executing in a controller. You could make the changes off line, and then write the entire program into the controller. However, it is often more convenient to make changes on line.

On line editing changes the program in the controller and the editor at the same time. The program can continue to execute in the controller while changes are made, or you can stop the program.

For example, assume a lamp is added to the motor control circuit, as shown in *Figure 22*. The lamp contact is on when the motor is running.

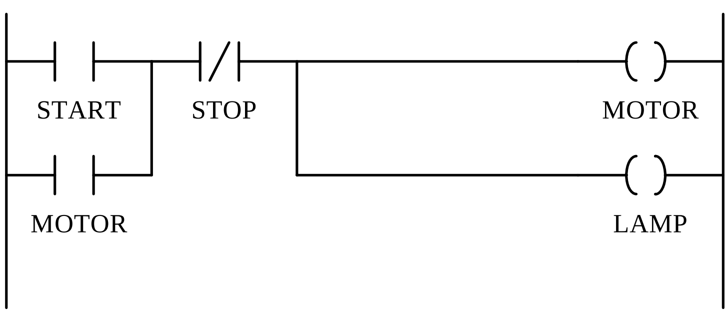


Figure 22: Revised Motor Control Circuit

To edit the program on line:

- Select **Edit On line** from the **Activity** menu or click on the edit on line button on the tool bar.

To enter the new tag name:

- Select **Tag Names** from the **Edit** menu. The edit tag names dialog appears.
- Enter the tag name LAMP. The output address is 00006, and the tag type is address.
- Click on the **Add Tag Name** button to add the tag.
- Click on **Close** to close the dialog.

Insert the element by positioning the mouse pointer in row 2, column 3 and double clicking the left mouse button. The **Insert / Edit Network Element** dialog *(Figure 23)* appears.

- Select coil from the **Function** list.
- Click on the **Input Tag List** drop down box. Select the tag LAMP from the list.
- Click on the **OK** button. The program in the controller is executing, so the dialog box in *Figure 23* appears. Select **Stop** if you want to stop the program execution. Select **Continue** is you want to keep executing the program after the change is made. Please read the warning below. It is safe to select **Continue** in this example.

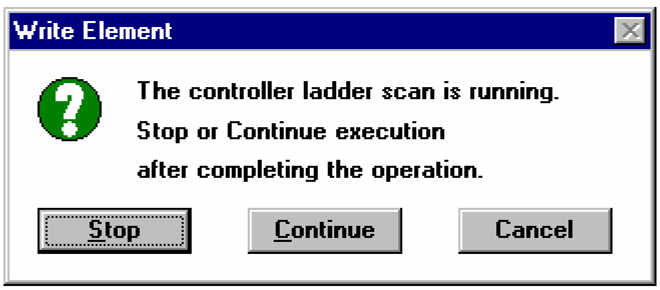


Figure 23: Insert / Edit Network Element

► Be careful when selecting the **Continue** option. The program will execute with the changes you make, even if the changes are not complete. This may cause undesired operation. Select **Stop** if you are making multiple changes.

- To insert the vertical shunt, position the mouse pointer over the STOP contact and right click to toggle the vertical shunt. Select **Continue** when the program execution query dialog appears.

The change to the program is now complete. Test it as described in the *Monitoring Execution* section.

Editing Off line

Once the program is completed and tested, all or parts of the program can be used to control a similar process. To demonstrate this procedure we will use motor control to control a motor in another network. This will also demonstrate the copy and paste features of the editor.

To edit the program off line:

- Click on the Edit Off line button on the tool bar, or select **Edit Off line** from the **Activity** menu.
- From the **Edit** Menu select the **Insert** command. The **Insert** dialog (see *Figure 24*) appears.

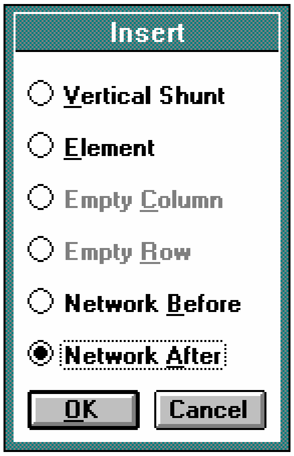


Figure 24: Insert Dialog Box

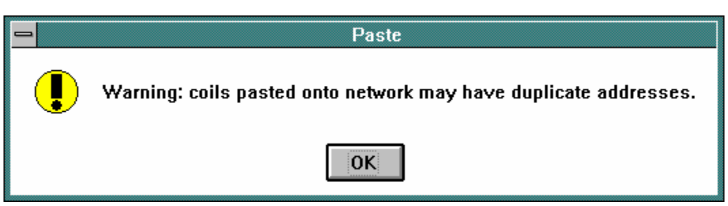
- Click on the **Network After** selection.

A new network is inserted into the program after network 1. The Network displayed in the Editor Window is now Network 2. The Network Display **current network/total networks** display (left side) now shows **0002/0002** indicating the second network of two networks.

- The **left arrow** icon on the Toolbar is now green. Click on this icon to go to the previous network, network 1. The motor control network is now displayed.

We want to copy this entire network except the LAMP.

- To do this position the cursor at the top left position of the motor control network, over the START contact. Click and hold the left mouse button and drag it down one position. The START contact and the MOTOR contact are now both highlighted by the cursor.
- Drag the mouse to the right one position. The START contact, MOTOR contact and STOP contact are now highlighted. Release the mouse and the selected items stay highlighted.
- From the **Edit** menu select the **Copy Selected** command. The selected elements are copied to the clipboard. Using **Ctrl+C** also copies the selected elements to the clipboard.
- Click on the **right arrow** icon in the Toolbar to move to network 2. Position the cursor in the upper left most position in the network.
- From the **Edit** menu select the **Paste** command, or use **Ctrl+V**. The elements selected from Network 1 are now pasted into the position of the cursor in Network 2. The START contact, MOTOR contact and STOP contact and the vertical shunt to the right of the STOP contact are now in Network 2.
- Delete the vertical shunt by positioning the cursor over the STOP contact and clicking the right mouse button.
- Repeat this procedure to copy and paste the MOTOR coil from Network 1 to Network 2. The following warning will be displayed.



The warning alerts you to the fact that there are two circuits controlling the same output coil. Click on the **OK** button to remove the message from the screen. We will change the address of the coil later.

Network 1 and Network 2 are now similar except Network 2 does not have the LAMP coil. We want the control in Network 2 to control a different motor so the Tag Names and Addresses for the START, and STOP contacts and the MOTOR coil and contacts must be changed.

To change the tag name and address of the START contact in Network 2.

- Add the Tag name START2 with the Number 10003.
- Double click on the START contact in Network 2 and select START2 from the Input Tag List.

Repeat this procedure for each element in Network 2 changing the Tag Names to STOP2 and MOTOR2. (Tag Name STOP2 will be Number 10004 and Tag name MOTOR2 will be Number 00007).

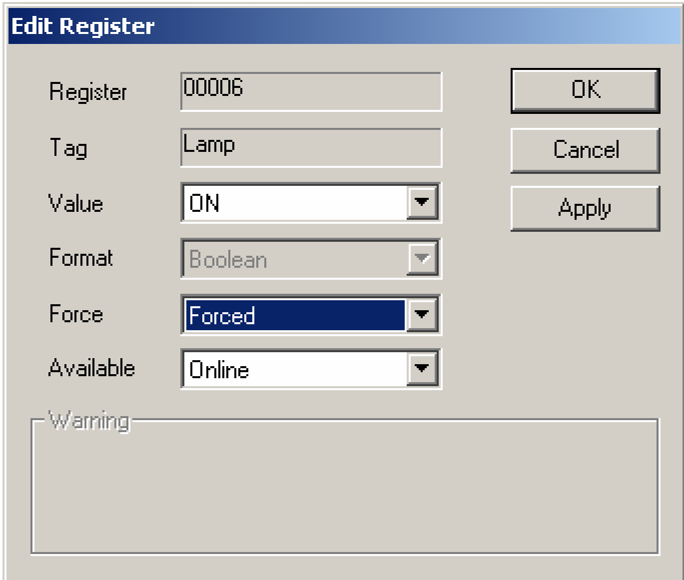
Forcing Registers

It is often convenient and necessary to be able to change the value of a register while the program is in operation. Toggling a digital output or changing the value contained in an analog holding register, for example, enables the programmer to simulate a particular condition and then test the operation of the ladder program.

To force a register:

- Select **Monitor On line** from the **Activity** menu or click on the **Monitor On line** button on the toolbar.
- With the mouse, highlight the element that corresponds to the register which is to be forced; in this example, the Lamp output 00006.
- Select **Monitor Element** from the **Controller** menu or activate the right mouse button and select **Monitor Element.** The registers used by highlighted element are added to the Register Editor.
- Highlight the register 00006 on the Register Editor and select the **Edit** button.
- The **Edit Register** dialog is displayed. Change the **Value** list box to *On,* change the **Force** list box to *Forced* and select the **Apply** button.
- Close the **Edit Register** dialog and observe output 00006.

Applying a new Value without forcing will work only if the ladder logic program is not writing to the register. Forcing will override any ladder logic programming and the register will remain forced until the force is removed



To remove the force from a particular register:

- Edit the register again and change the **Force** list box to *No*.

OR

- Select **List Forced Registers** from the **Controller** menu. The **Forced Registers** dialog is displayed. Highlight the forced register from the list and select **Remove Force** button. This command is only available in the *Edit On Line* or *Monitor On Line* mode.

To remove all forcing:

- Select **Remove All Forces** from the **Controller** menu**.** A confirmation prompt is displayed.
  - Select the **Yes** button. This command is only available in the *Edit On Line* or *Monitor On Line* mode.

Using Help

The on line help for the TelePACE ladder editor contains a complete reference to the editor and all ladder logic function blocks. Help opens the *TelePACE User Manual*. You can refer to the help, while you edit your programs.

To display help:

- Select **Contents** from the **Help** menu or click on the **Help** button on the tool bar. The *TelePACE User Manual* is opened.
- You may also select the *TelePACE User Manual* directly under the *TelePACE* program group from the *Start* menu.

When Downloading a new Program to Controller

When downloading a new program to the controller over a serial port, the following steps are recommended:

1. **Open Program File**: Open the program file in TelePACE.
2. **Service Boot**: If your PC is not currently communicating with the controller you may need to perform a Service Boot to set the controller serial ports to a known state: i.e. their default settings.
3. **PC Communication Settings:** Confirm the *PC Communications Settings* on the *Communications* menu match the settings of the controller's serial port connected to your PC.
4. **Controller | Initialize**: This dialog presents initialization options. The program download will erase and replace ladder program and the register assignment, so these items do not need to be erased. To set all the Modbus registers to zero before downloading the new program and to set the controller serial ports to their defaults, check the item *Initialize Controller* from this dialog. Check any remaining items from the dialog as required if applicable.
5. **Write to Controller**: To download the TelePACE program, select *Write to Controller* from the *Communications* menu. The ladder program, register assignment, and controller settings are written to the controller. Also, any registers in the Register Editor (with *Available* flag set to *Always*) will be preset in the controller to the values shown on this dialog.

Ladder Logic Functions

The Ladder Logic Functions used in TelePACE programs are summarized below. For details and examples using each function see the *Function Block Reference* of the *TelePACE User Manual* on your PC under the TelePACE Program group of your Start Menu.

Math Functions using Unsigned Registers

ADDU – Add Unsigned Values
CMPU – Compare Unsigned Values
DIVU** – Divide Unsigned Values
MODU – Modulus of Unsigned Values
MULU* – Multiply Unsigned Values
PUTU – Put Unsigned Value into Registers
SUBU – Subtract Unsigned Values

*MULU – result is an Unsigned Double (32 bit register); a 32 bit register = two consecutive 16 bit registers.

**DIVU – input is an Unsigned Double (32 bit register); result is an Unsigned (16 bit register)

Math Functions using Signed Registers

ABS – Absolute Value
ADD – Add Signed Values
CMP – Compare Signed Values
DIV** – Divide Signed Values
MUL* – Multiply Signed Values
PUT – Put Signed Value into Registers
SUB – Subtract Signed Values

*MUL – result is a Signed Double (32 bit register)

**DIV – input is a Signed Double (32 bit register); result is a Signed (16 bit register)

Math Functions using Floating Point Registers

ABSF – Floating-Point Absolute Value
ADDF – Add Floating-Point Values
DIVF – Divide Floating-Point Values
GTEF – Floating-Point Greater Than or Equal
LTEF – Floating-Point Less Than or Equal
MULF – Multiply Floating-Point Values
POWR – Raise Floating-Point Value to a Power
PUTF – Put Floating-Point Value
SUBF – Subtract Floating-Point Values
SQRF – Square Root of Floating-Point Value

Convert Register Type

FTOS – Floating-Point to Signed Integer
FTOU – Floating-Point to Unsigned Integer
STOF – Signed Integer to Floating-Point
UTOF – Unsigned Integer to Floating-Point

Boolean Operations

AND – And Block
Coil
CMPB – Compare Bit
GETB – Get Bit from Block
Normally Closed Contact
Normally Open Contact
NOT – Not Block
One Shot Coil
OR – Or Block
PUTB – Put Bit into Block
ROTB – Rotate Bits in Block
XOR – Exclusive Or Block

Counters and Timers

DCTR – Down Counter
PULM – Pulse Minutes
PULS – Pulse Seconds
UCTR – Up Counter
Timers – 0.01, 0.1, and 1 second Timers

Controlling Block of Registers

L–>L – List to List Transfer
L–>R – List to Register Transfer
MOVE – Move Block
OVER – Override Block of Registers
R–>L – Register to List Transfer

SCADA Application Functions

FLOW – Flow Accumulator
PID – PID Controller
PIDA – Analog Output PID
PIDD – Digital Output PID
SCAL – Scale Analog Value
SLP – Put Controller into Sleep Mode
TOTL – Analog Totalizer

Data Logging

DLOG – Data Logger
FIN – FIFO Queue Insert
FOUT – FIFO Queue Remove
GETL – Data Logger Extract

Communications

DIAL – Control Dial-Up Modem
HART – Send HART Command
INIM – Initialize Dial-Up Modem
MSTR – Master Message
MSIP – Master TCP/IP Message

Program Execution

CALL – Execute Subroutine
SUBR – Start of Subroutine

Using the Network Display

The network display is the area of the TelePACE program where all ladder logic programming and editing occurs. All programming and editing functions are done from the ***cursor*** position. The cursor is the highlighted square inside the network display. The cursor position is always highlighted and will highlight any element that it is positioned on.

Positioning the cursor within the Network Display

- Move the mouse pointer to any position in the network display and click the left mouse button.
- The cursor can be moved about the network using the keyboard directional arrow keys.

Inserting Elements in a Network

- Position the cursor at the network position where the element is to be inserted and double click the left mouse button.

Or

- Elements can be inserted using the right mouse button and selecting **Edit Element** from the drop down menu.

Or

- Elements can be inserted from the keyboard by positioning the cursor at the desired network position and pressing the Insert key.

When any of these actions are taken the ***Insert/Edit Network Element*** dialog box pops up and an element or function may be selected to insert.

► The **Insert/Edit Network Element** dialog box only displays the elements and functions that will fit from the cursor position.

Editing Elements in a Network

- Position the cursor on the element to be edited and double click the left mouse button.

Or

- Elements can be edited using the right mouse button and selecting ***Edit Element*** from the drop down menu.

Or

- Elements can be edited from the keyboard by positioning the cursor at the desired element the Insert key.

When any of these actions are taken the ***Insert/Edit Network Element*** dialog box pops up and an element configuration may edited.

Inserting Networks in the TelePACE Program

- To insert an empty network, before or after the current network, select ***Insert*** from the ***Edit*** menu.

The ***Insert*** dialog pops up. Selections are made in this dialog by moving the mouse pointer to the required radio button and clicking the left mouse button.

Deleting Elements and Networks from TelePACE

- Position the cursor on the element to be deleted and press the ***Delete*** key.

Or

- Select ***Delete*** from the ***Edit*** menu.

Or

- Click the right mouse button and selecting ***Delete*** from the drop down menu.

The **Delete** dialog pops up. Selections are made in this dialog by moving the mouse pointer to the required radio button and click the left mouse button.

Copying Elements to the Clipboard

- Position the cursor on the element to be copied and select ***Copy Selected*** from the ***Edit*** menu.

Or

- Click the right mouse button and selecting ***Copy Selected*** from the drop down menu.

Copying Networks to the Clipboard

- Position the cursor on the element to be copied and select ***Copy Networks*** from the ***Edit*** menu.

Or

- Click the right mouse button and selecting ***Copy Networks*** from the drop down menu.

Selecting Multiple Elements

- Position the cursor on the first element of the selection and press the left mouse button. Hold the button down and drag the pointer to the last element or function.

Or, to select disconnected elements

- Position the cursor on the first element. Hold the keyboard shift key down and using the left mouse button select elements to be copied to the clipboard.

Or, to use the keyboard

- Use the keyboard directional arrow keys to move the cursor to the first element. Hold the shift key down while using the directional arrow keys to move the cursor to the last element.

The elements selected will be highlighted in the same way as the cursor. Releasing the mouse button does not change the highlighting.

Using Keyboard Shortcuts

The following function keys or control key shortcuts are available:

Ctrl-A	Save As File	**F3**	Repeat Last Find
Ctrl-S	Save File	**F5**	Toggle Vertical Shunt
Ctrl-O	Open File	**F6**	Insert Horizontal Shunt
Ctrl-Q	Exit	**F7**	Previous Network
Ctrl-Z	Undo	**F8**	Next Network
Ctrl-X	Cut Selected	**Delete**	Delete Menu
Ctrl-C	Copy Selected	**Insert**	Insert/Edit Element
Ctrl-V	Paste	**PgDn**	Scroll Down one page
		PgUp	Scroll Up one page

Controller Lock

Locking a controller prevents unauthorized access. The controller will reject commands sent to the unit when it is locked. A controller that is unlocked operates without restriction.

The *Lock Controller Dialog,* see *Figure 25,* specifies a password to be used to lock the controller and the commands that are to be locked.

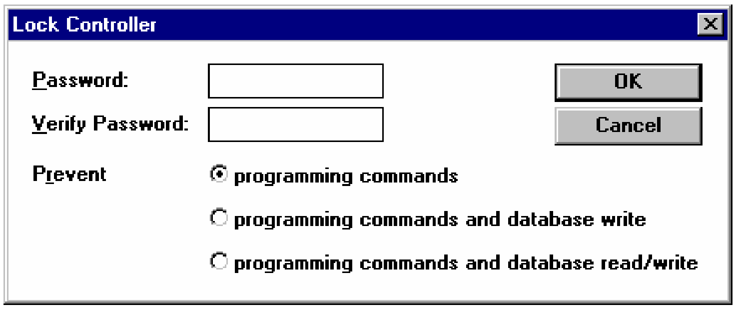


Figure 25: Lock Controller Dialog

Enter a password in the *Password* edit box. Re-enter the password in the *Verify Password* edit box. Any character string up to eight characters in length may be entered. Typing in these edit boxes is masked. An asterisk is shown for each character typed.

The *Prevent* radio buttons select the commands that are locked.

- Locking the programming commands prevents modifying or viewing the program in the controller. Communication protocols can read and write the I/O database.
- Locking programming and database write commands prevents modifying or viewing the program and prevents writing to the I/O database. Communication protocols can read data from the I/O database, but cannot modify any data.
- Locking programming and database commands prevents modifying or viewing the program and prevents reading and writing the I/O database. Communication protocols cannot read or write the I/O database.

The *OK* button verifies the passwords are the same and sends the lock controller command to the controller. The dialog is closed. If the passwords are not the same an error message is displayed. Control returns to the dialog.

The *Cancel* button closes the dialog without any action.

If the controller is already locked, a message indicating this is shown instead of the dialog.

Unlock Controller

The *Unlock Controller Dialog,* see *Figure 26*, prompts the user for a password to be used to unlock the controller. If the controller is locked, the following dialog is displayed.

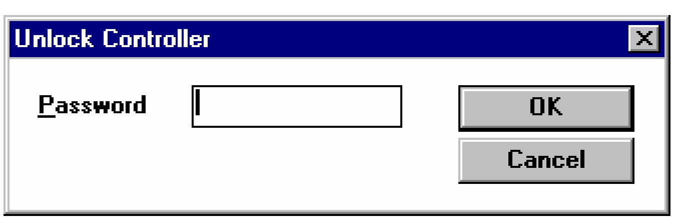


Figure 26: Unlock Controller Dialog

Enter the password that was used to lock the controller in the *Password* edit box. Any character string up to eight characters in length may be entered. . Typing in this edit box is masked. An asterisk is shown for each character typed.

The *Cancel* button closes the dialog without any action.

The *OK* button sends the *Unlock Controller* command to the controller. If the password is correct the controller will be unlocked. If the password is not correct, the controller will remain locked.

Override Controller Lock

The *Override Controller Lock Dialog,* see *Figure 27*, allows the user to unlock a controller without knowing the password. This can be used in the event that that the password is forgotten.

To prevent unauthorized access to the information in the controller, the C and Ladder Logic programs are erased. Use this command with caution, as you will lose the programs in the controller.

Selecting the *Override Controller Lock* command displays this dialog.

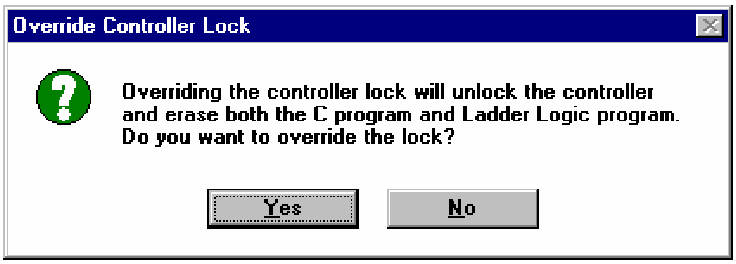


Figure 27: Override Controller Lock Dialog

The *Yes* button unlocks the controller and erases all programs.

The *No* button closes the dialog without any action.

Show Lock Status

The *Show Lock Status* command displays the controller lock state. It opens a dialog showing one of the following states:

- unlocked
- locked against programming commands
- locked against programming commands and database write
- locked against programming commands and database read/write

The *OK* button closes the dialog.

Cold Boot Mode

COLD BOOT mode is used after installing new controller firmware or when the controller lock password has been forgotten. When the SCADAPack controller starts in COLD BOOT mode:

- The default serial communication parameters are used.
- The Ladder Logic program is erased.
- The C program is erased.
- The registers in the I/O database are initialized to default values.

- The Register Assignment is erased.
- The controller is unlocked.

COLD BOOT mode is selected by performing a COLD BOOT using the following procedure:

1. Remove power from the SCADAPack controller.
2. Hold down the LED POWER button.
3. Apply power to the SCADAPack controller.
4. Continue holding the LED POWER button for 25 seconds until the STAT LED begins to flash on and off continuously.
5. Release the LED POWER button.

► If the LED POWER button is released before the STAT LED flashes, the controller will start in either the SERVICE or RUN modes.

Lead Lag Pump Control Example

In this section of the TelePACE Ladder Logic training course a practical example of a simple lead / lag pump control program will be developed. A program will be developed from a very simple single pump start / stop switch control into a more complex two pump automatic lead / lag control system.

The development of the lead / lag pump control is divided into five parts. Each part of this section emphasizes one or more function blocks that are available for use when programming the Controller using the TelePACE Ladder Logic.

In Part One of this section the Controller is programmed to use the Series 5000 I/O as a means of controlling the turning on and off of a digital output. Digital inputs and then analog inputs are used to turn the output on and off. This process simulates the starting and stopping of a pump connected to the digital output.

In Part Two math functions are used to scale the analog input value used to control the pump operation. The types of numbers used in the TelePACE Ladder Logic are explained and the use of an initialization network is introduced.

In Part Three counter and timer functions are added to the program to store the accumulated pump run time and to store the number of pump starts.

Part Four of this section adds a lag pump to the program and a lead / lag pump control program is developed.

The last section involves using an automatic switching sequence to control the lead / lag pump operation in the program.

Part One: Digital Output Control

The first step in the program development will be to create a Ladder Logic program that is similar to the Motor.lad program.

Step 1 Create tag Names

- Open a new file in the TelePACE Ladder Network Editor.
- Enter the Tag names using the Edit Tag Name dialog.

Tag Name	***Register Address***
Pump 101 Start	10001
Pump 101 Stop	10002
Pump 101 Run Lead	00001
Power Up Reset	01057
LeadStop	01058
LeadStrt	01059
Pump 101 TimerRst	01071
Analog Input 00	30001
Multiply Low Word	42200
AIN 00 0_100.0%	42202
P101 Lo Minutes	42206
P101 Hi Minutes	42207
P101 Run Hours	42210
Pump 101 Starts	42224
LeadStrt Setpoint	43000
LeadStop Setpoint	43001

? ***How can these Tag Names be viewed in the Ladder Editor?***

Step 2 Create Motor Start Program

The Motor Start program is very similar to the program created in the Using the TelePACE Ladder Editor section of the course.

- Insert the elements as shown in *Figure 28: Pump Start Program.*
- Load the program into the Controller and monitor the program execution.

The output coil 00001 can be turned on and off using the input switches 10001 and 10002. Using digital inputs to control digital outputs is a very common occurrence when using relay ladder logic. Pumps are usually started and stopped based on the level of the tank or reservoir they are being used to fill. In this case an analog input is used as a control for starting and stopping the pump.

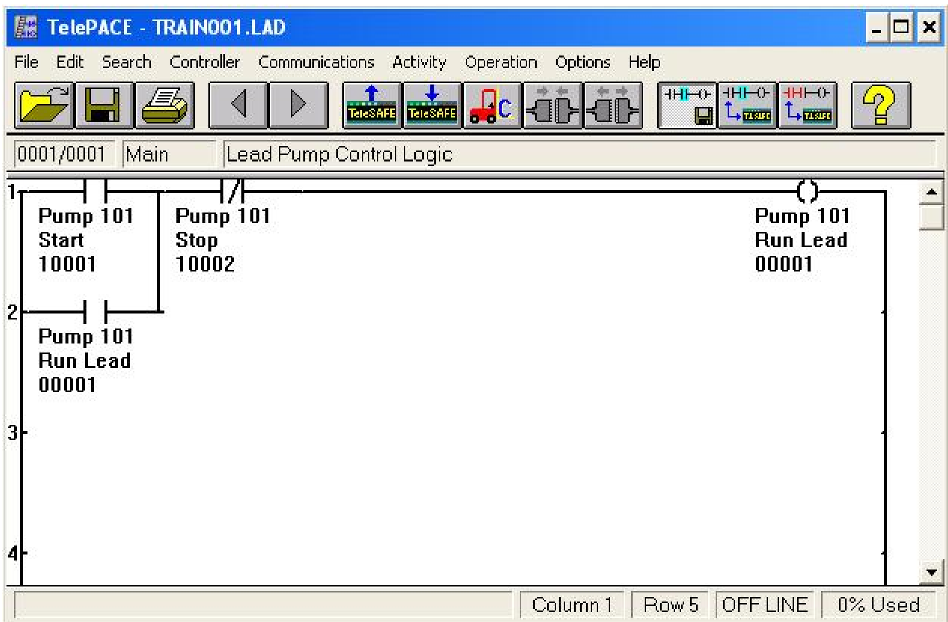


Figure 28: Pump Start Program

- Save the Ladder Logic program as TRAIN001.LAD.

Step 3 Analog Input to Start and Stop the Pump

In order to use an analog input to control the starting and stopping of the digital output the value of the analog input must be compared to some setpoint value. The TelePACE Ladder Logic functions used for comparing values are the CMP and CMPU. The CMP function is used to compare signed values and the CMPU is used to compare unsigned values.

- Delete the elements that were inserted in step 2.
- Select the Help menu and then select the CMPU function from the function block reference section.

The Compare function allows two registers or constants to be compared and an output from the function that is the result of the comparison.

This function will be used to compare the raw value of analog input 0 with a constant representing one half of the maximum raw value of analog input 0. This constant value is referred to as the setpoint.

? ***What is the range of a raw analog input on the SCADAPack 5601 I/O module?***

- To insert a CMPU function position the cursor in Row 1 Column 1 and double click the left mouse button.
- Select the CMPU function from the function list and configure the function as shown in *Figure 29: CMPU Compare Function*.

Complete the program as shown in

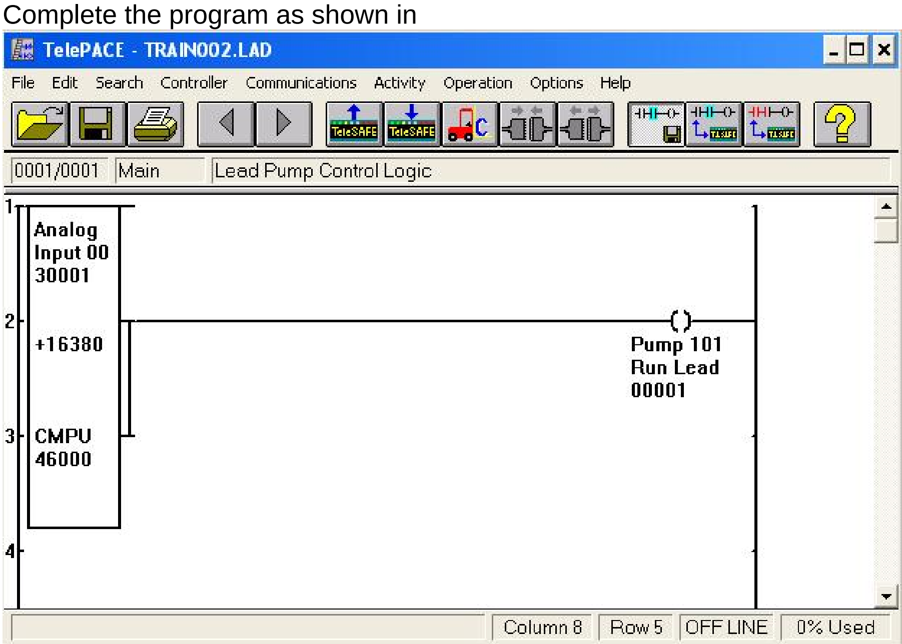


Figure 30: Using CMPU Function.

? ***What is the function of the vertical shunt used on the CMPU output?***

- Load the program into the Controller and monitor the program execution.

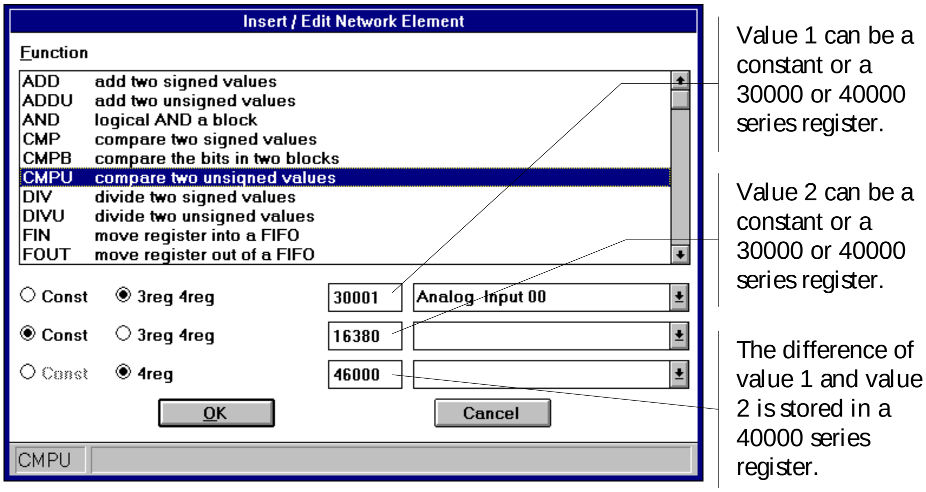


Figure 29: CMPU Compare Function

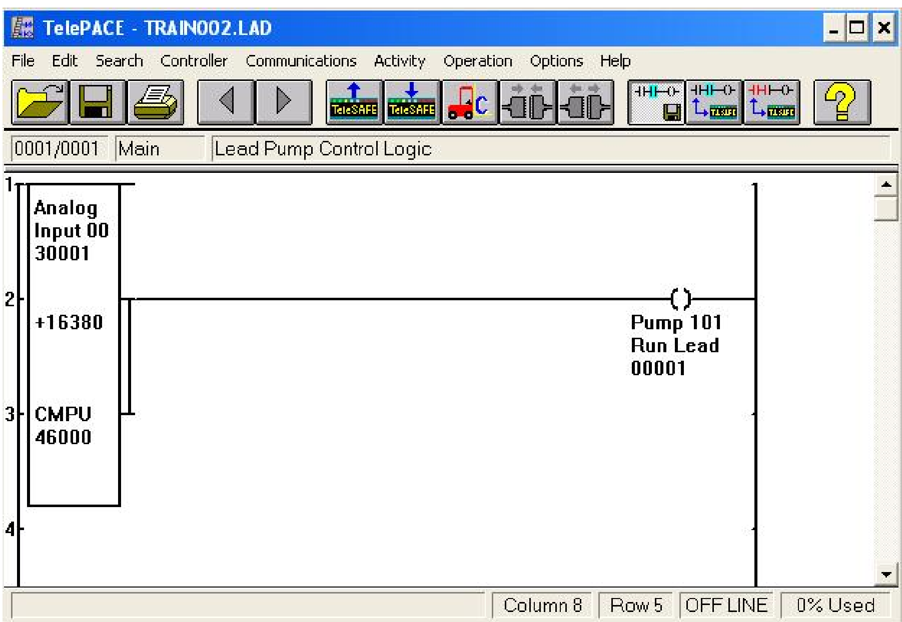


Figure 30: Using CMPU Function

? ***What is the limitation of using this type of control?***

- Save the Ladder Logic program as TRAIN002.LAD.

Step 4 Adding Hysteresis to the Pump Control

Generally a pump will not be started and stopped at one analog input level. The pump will continually be starting and stopping as the level rises above and falls below the setpoint. Hysteresis is simply having one setpoint to start the pump and another setpoint to stop the pump.

Examine the network shown in Figure 31: Hysteresis Example and answer the following questions.

? ***What do the two numbers 29484 and 22932 represent?***

? ***What type of coils are 1058 and 1059?***

? ***At what analog input level will the coil 01059 be energized?***

? ***At what analog input level will the coil 01058 be de-energized?***

- Modify the program as shown in *Figure 31: Hysteresis Example*.
- Load the program into the Controller and monitor the program execution.

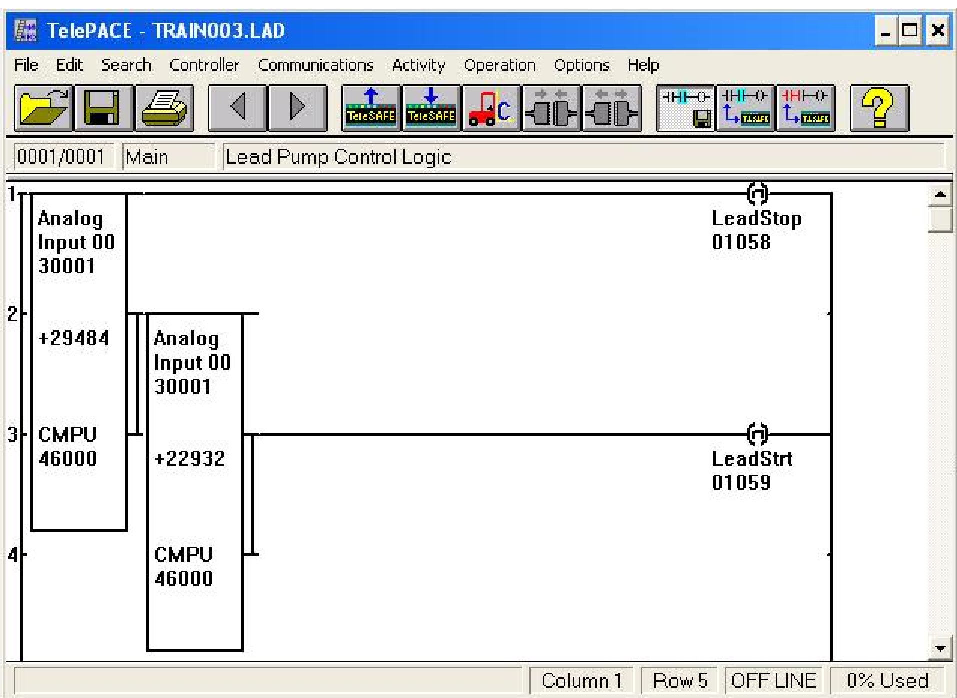


Figure 31: Hysteresis Example

Step 5 Pump Control Using Hysteresis

The limitations of the above example are quickly seen when the program is monitored on line. To overcome these limitations and to control the pump start and stop the program needs to be modified to that shown in *Figure 32: Working Hysteresis Example*.

- Modify the program as shown below.
- Load the program into the Controller and monitor the program execution.

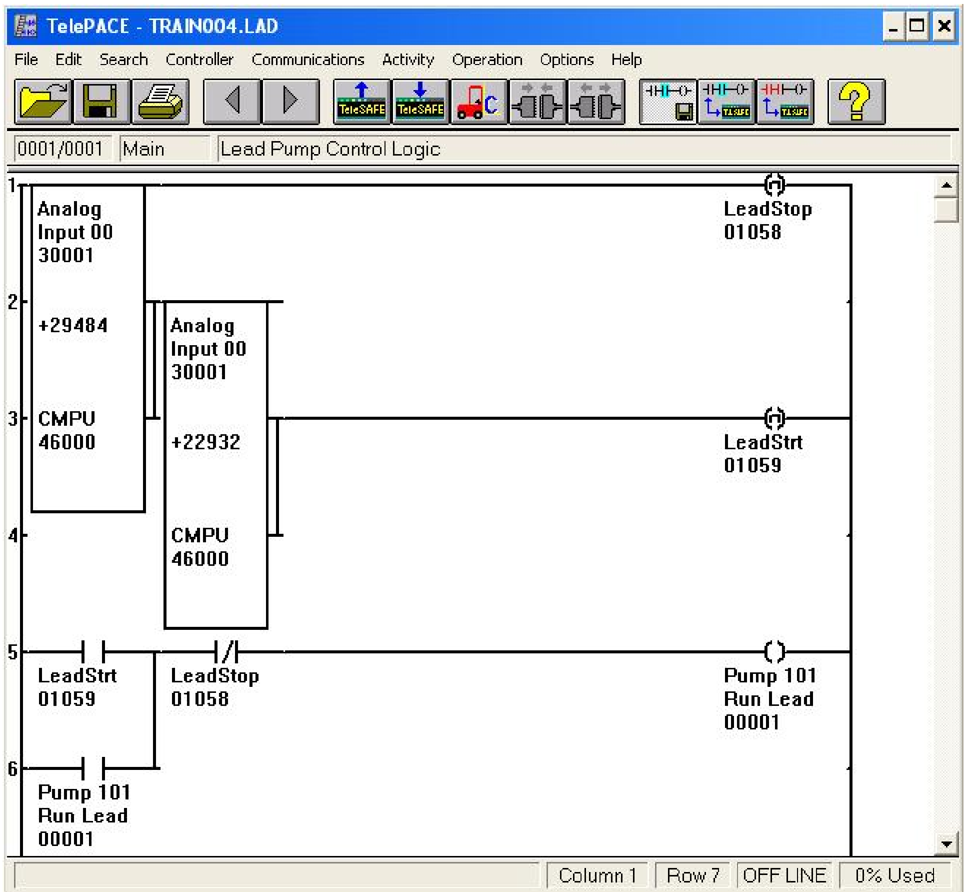


Figure 32: Working Hysteresis Example

- Save this program as TRAIN004.LAD**.**

Part Two: Scaling Analog Inputs

In this part of the course logic will be added to the program to scale the analog input, register 30001. A field Transmitter is commonly used to convert a process value, temperature, pressure, level etc., into voltage or current level. This level is input to a 5000 Series analog input module. The current or voltage value at the analog input is analog to digital (A/D) converted into a raw I/O count of between 0 and 32760.

The Ladder Logic program running in the controller often must convert the raw I/O count back into the units of measurement of the process value, or to a percentage of the maximum input. This process is referred to as scaling the analog input.

Before the program can be modified to scale the analog input, the way in which decimal numbers are represented in the TelePACE Ladder Logic needs to be understood.

Representing Numbers in the Controller

The type of numbers used in TelePACE Ladder Logic are called integers or fixed point numbers. Integers are numbers that do not have decimal places and are commonly called whole numbers. Integer numbers can be positive or negative.

The controller stores integer numbers in 16 bit words. Each bit in the word is either a 1 or a 0 and the combination of 1's and 0's represent the binary equivalent of the integer number stored in the word. The bits in a word are numbered from 0 to 15 starting at the right side of the word. Bit 0 is referred to as the least significant bit and bit 15 is the most significant bit. Each bit in a word has a binary and a decimal value as shown in the following table:

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

Using the above table it can be seen that if all 16 bit positions are 1's the word will have a value of 65535. This is the largest number that a single word can contain.

Each 16-bit word can contain an unsigned integer or a signed integer. Unsigned integers are positive and have the range of 0 to 65535. Signed integers may be positive or negative. When a word contains a signed integer the most significant bit, bit 15, is used for the sign extension, positive or negative. This leaves 15 bits to represent signed integers. Signed integers have a range of -32768 to +32767.

The 16 bit words in the controller are used to store constants, input registers, output registers and blocks of 16 status and coil registers.

? ***What would the value be for a register with the following bit combination?***

`0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1`

? ***What would the bit structure be for the integer value 65435?***

? ***What is a limitation of using 16 bit words?***

Double Words

Often the result of a multiplication function is much larger than the maximum value of 65535 available using a single word. The same is true when a divide function is used, the number to be divided is often larger than 65535.

The TelePACE Ladder Logic uses double words to store the result of a multiplication function, and as the value to be divided in a divide function. A double word is simply two sequential 16 bit words. The largest value that can be stored using a double word is 4,294,967,295.

Scaling Analog Input 30001

The training.lad program will be modified to scale the raw analog input, register 30001, into a percentage of the maximum analog input value, 32760. The calculation used for the scaling will be:

$$\frac{\text{Value of Input Register 30001} \quad \times \quad 100\%}{\text{Maximum Value of Input Register 30001}}$$

Step 1 Multiply Input Register by 100

- Open the Ladder Logic program saved in Part One.
- Insert a network before network 1.

There are two multiplication function that are available in the TelePACE Ladder Editor, the MUL function for multiplying signed numbers and MULU function for multiplying unsigned numbers. Since the value of input register 30001 will always be between 0 and 32760 the MULU function can be used.

- Double click the mouse in position Row 1, Column 1 of network 1 and select the MULU function from the Insert / Edit Network Element Dialog box.
- Complete the Dialog box as shown in *Figure 33: Insert / Edit MULU Function*.

? ***What are the values of registers 42200 and 42201 when the value of input register 30001 is 32760?***

? ***What decimal value do registers 42200 and 42201 represent when input register 30001 is 32760?***

? ***How do the values in registers 42200 and 42201 represent this value?***

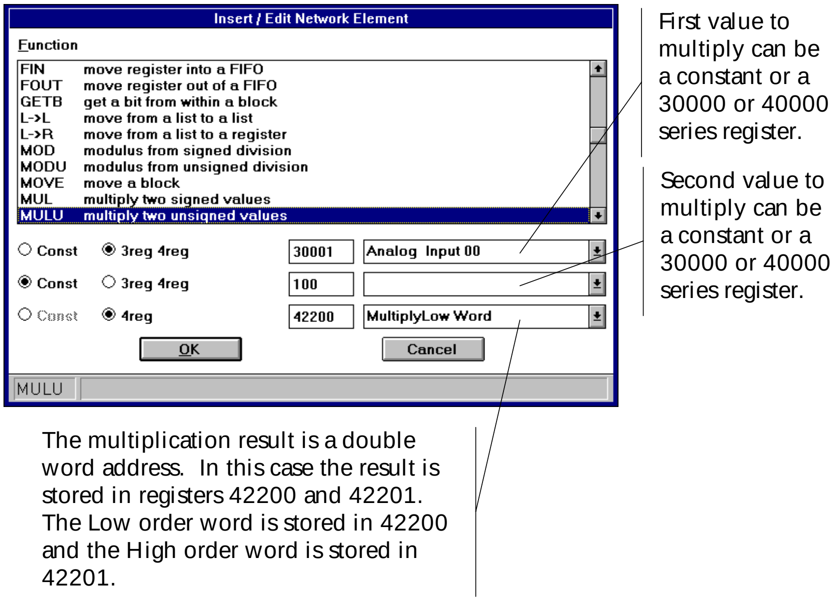


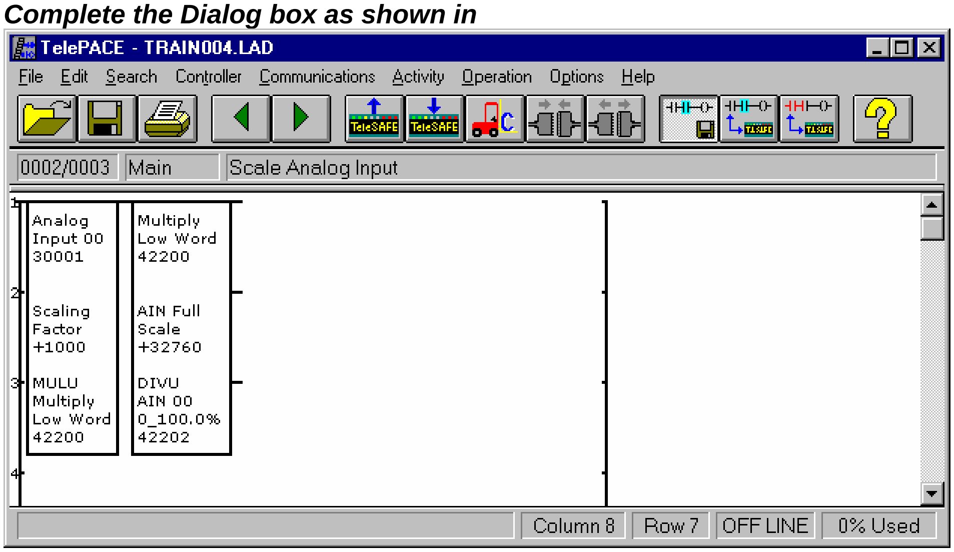
Figure 33: Insert / Edit MULU Function

Step 2 Divide the result by 32760

There are two division functions that are available in the TelePACE Ladder Editor, the DIV function for dividing signed numbers and DIVU function for dividing unsigned numbers. Since the result of the multiplication in step 1 will always be positive the DIVU function can be used.

- Double click the mouse in position Row1, Column 2 of network 1 and select the DIVU function from the Insert / Edit Network Element Dialog box.

Complete the Dialog box as shown in



The network should now look like the network in *Figure 35: Completed Scaling Network.*

- Load the modified program into the controller and run it.
- Using the Monitor On line function monitor registers 30001 and 42202.

- Adjust the input pot for different values in register 30001.

? ***Why is register 42200 used as the value to divide in the DIVU function?***

? ***How many I/O counts are represented with each percent change in register 42202?***

? ***How can this network be modified for better resolution?***

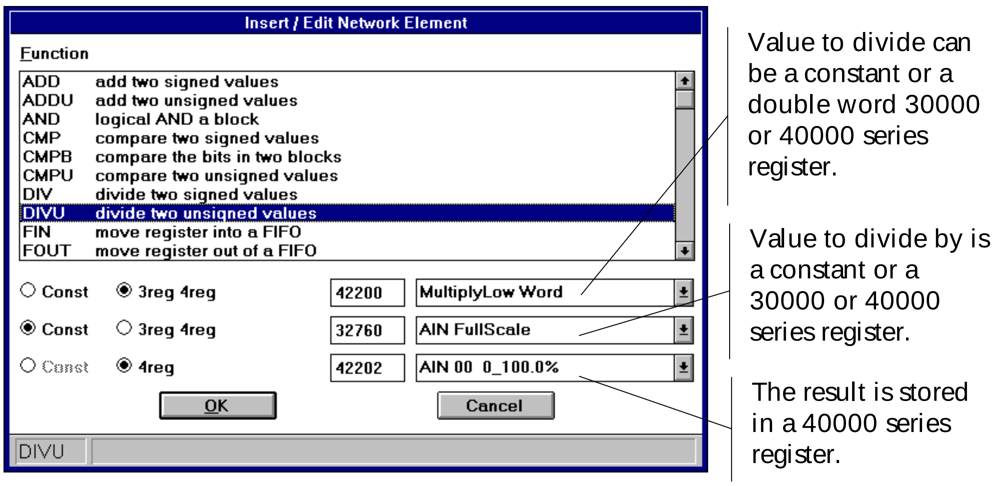


Figure 34: Insert / Edit DIVU Function

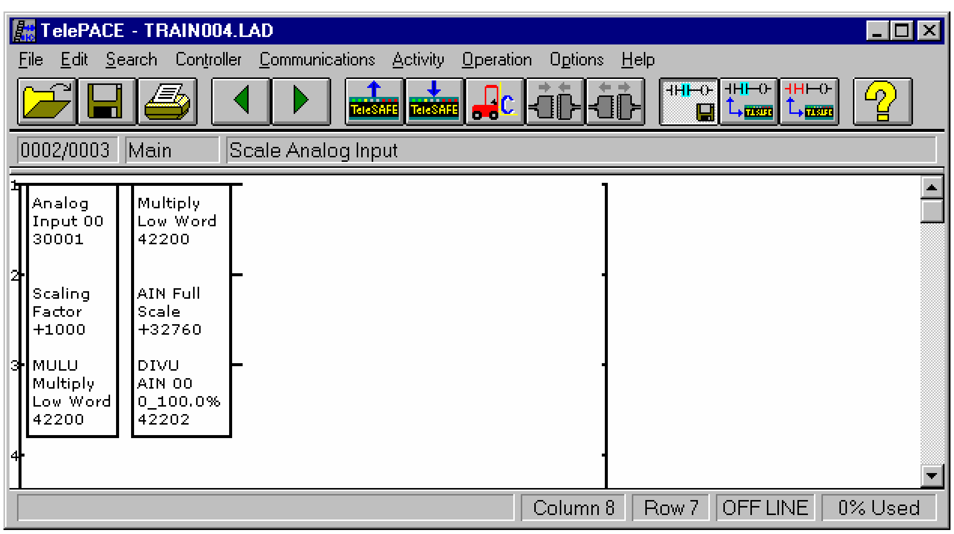


Figure 35: Completed Scaling Network

To increase the resolution of the scaling calculation change the constant +100 in the second value to multiply in the MULU function to +1000. This change will cause the value in register 42202 to have a range of 0 to 1000. This range will represent a scaled value of 0.0 to 100.0%.

- Modify the MULU function for a constant of +1000 instead of +100.
- Modify the CMPU function in network 2 to use the scaled input values.
- Save the Ladder Logic program as TRAIN005.LAD.

Step 3 Change Setpoints to Registers from Constants

In this step the start and stop setpoint values in network 2 will be changed from constants to analog output registers. When a constant is used in a Ladder Logic function, it can only be changed by editing the function using the TelePACE Ladder Editor. To enable the start and stop Setpoints to be changed through the I/O Database an analog out put register must be used.

? ***What is the advantage of changing these setpoints through the I/O Database?***

- Modify the network 2 as shown in *Figure 36: Modified Setpoint Network*.

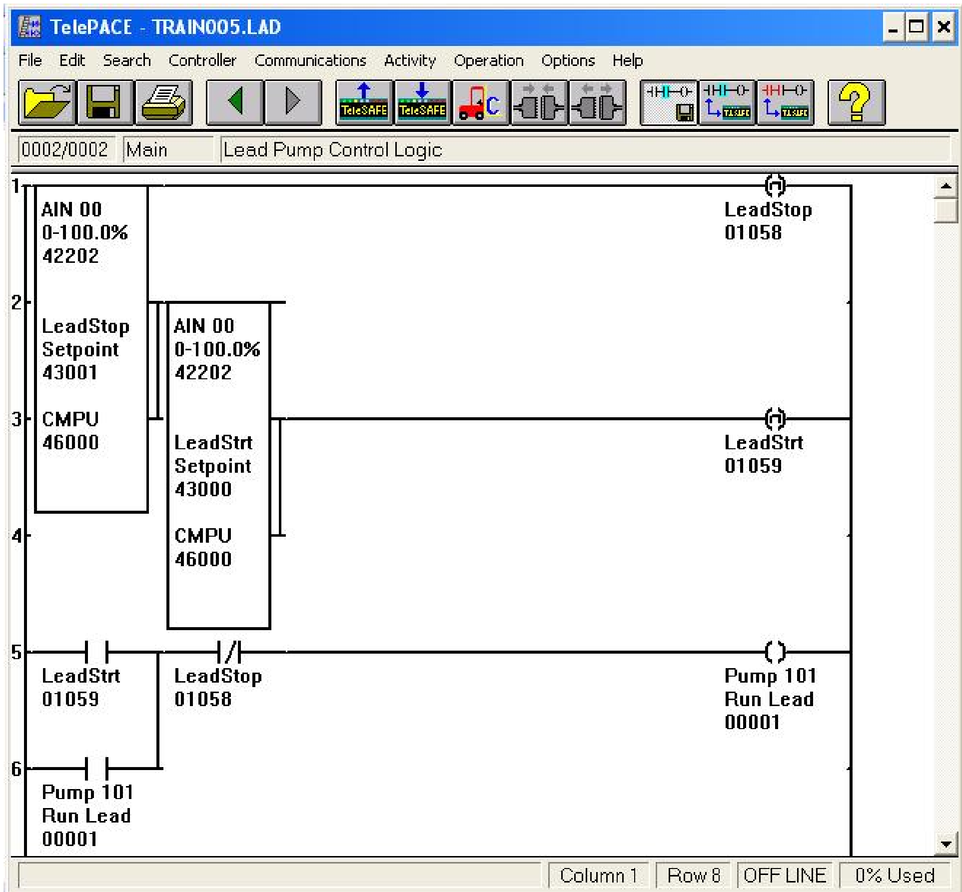


Figure 36: Modified Setpoint Network

Step 4 Put start up values into the setpoint registers

The TelePACE Ladder Logic functions that are used to put values into registers are the PUT and PUTU functions. The PUT function is used for signed numbers and the PUTU function is used for unsigned numbers.

In this program the setpoint values are stored in analog output registers so that they may be modified through the I/O Database. For this reason the Ladder Logic program should put values into the start and stop setpoint registers when the program initially starts.

? ***Why would the setpoint values only be put into the analog output registers when the program initially starts?***

- Go to network 1 of the program and insert a network before network 1.
- Insert the elements as shown in *Figure 37: One Shot Coil and Contact.* Note that the one-shot coil is inserted in row 1 column 1.

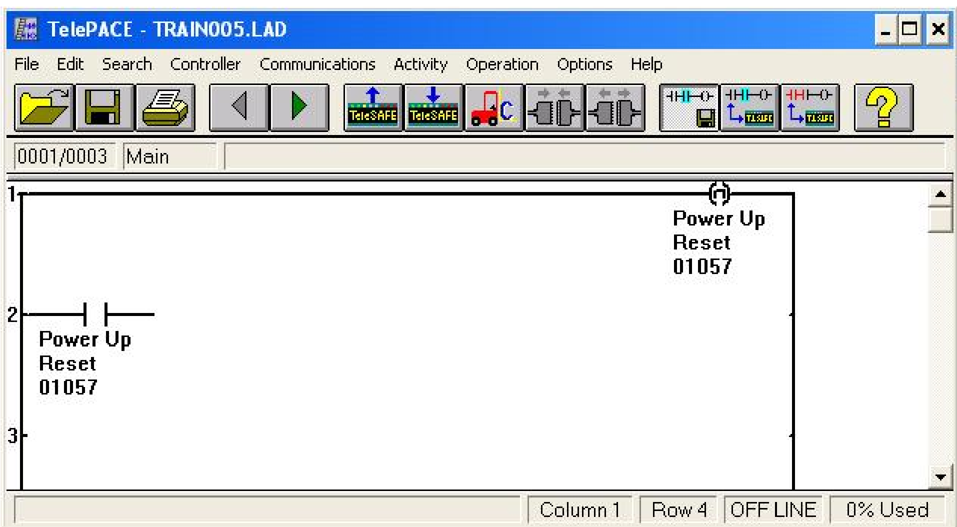


Figure 37: One Shot Coil and Contact

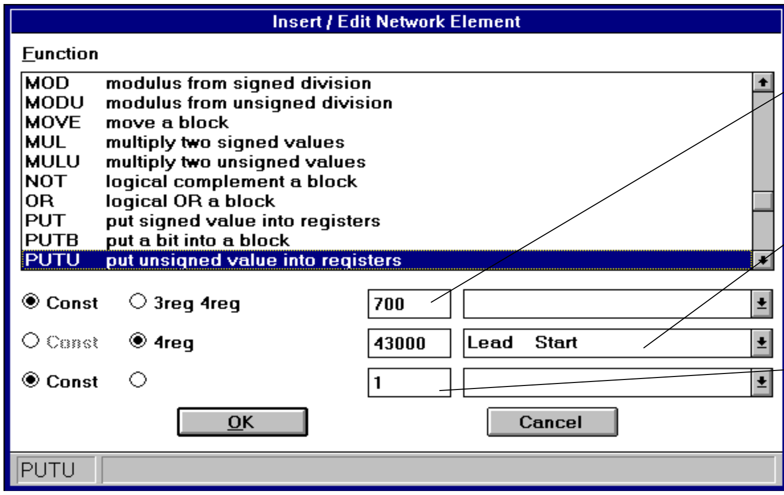
? ***When will the normally open contact 01057 be closed?***

- To the right of the contact 01057 insert a PUTU function. Configure the function as shown in *Figure 38: Insert / Edit PUTU Function*.

? ***Why is the value 700 used as the source register?***

? ***When will this value be PUTU into register 43000?***

- Insert another PUTU function for the stop setpoint register. The network should look as shown in *Figure 39: Completed Start Up Network*.
- Save the Ladder Logic program as TRAIN006.LAD.



The source can be a constant or a 30000 or 40000 series register.

The destination block is a 30000 or 40000 series register.

The number of registers in the destination block may be 1 to 9999.

Figure 38: Insert / Edit PUTU Function

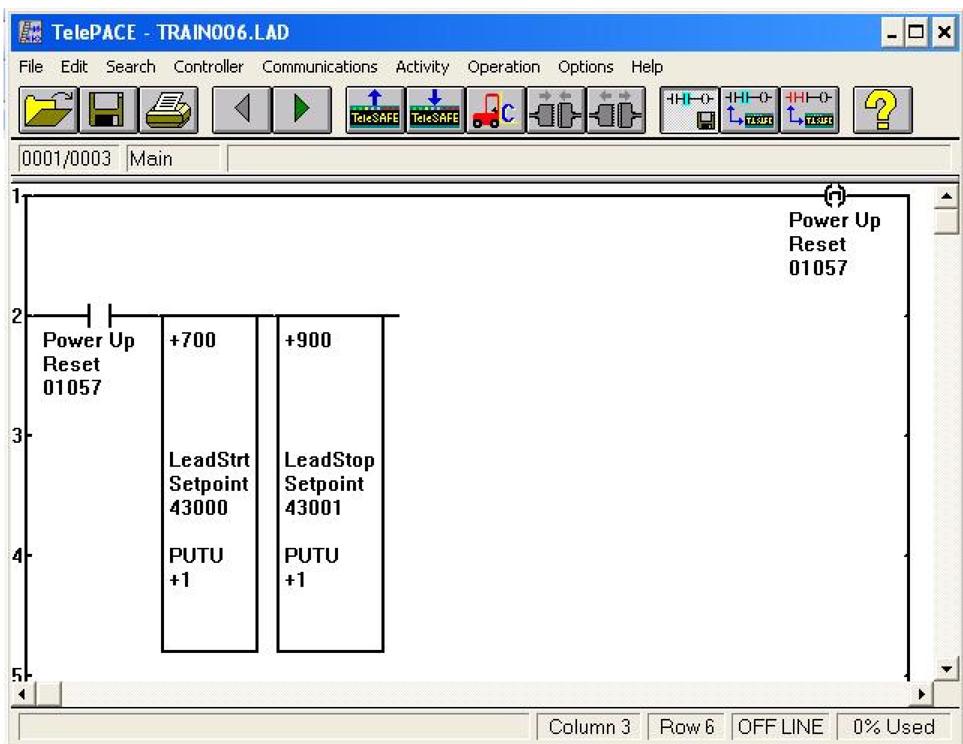


Figure 39: Completed Start Up Network

Part Three: Run Timer, Start Counter

In this part of the training course the program is going to be modified to add a pump run timer and a number of starts counter for Pump 101.

Step 1 Set Up a One Minute Timer

For maintenance purposes it is useful to know how many hours a pump has run. The first step in creating a run timer for the pump is to create a timer that will run in one minute intervals. To do this a T1 Timer function will be used.

- Display the Timer function in the TelePACE Ladder Editor Help.

The graphic of the T1 timer in the Help screen shows that it has four connections, two on the left side and two on the right side. This graphic is a representation of how the Time function appears in the Ladder Editor. The connections on the left of the block are referred to as inputs and the connections on the right side of the block are referred to as outputs.

? ***What connections to the timer are required for the accumulator register to increment?***

? ***How is the Timer enabled for timing?***

? ***When the Timer is enabled what is being timed?***

? ***In this program what would the limit variable contain?***

? ***How is the Timer reset?***

- Insert a network after network 3.
- In row 1 column 2 insert a T1 Timer function. Configure the timer as shown in *Figure 40: Insert / Edit Timer Function*.

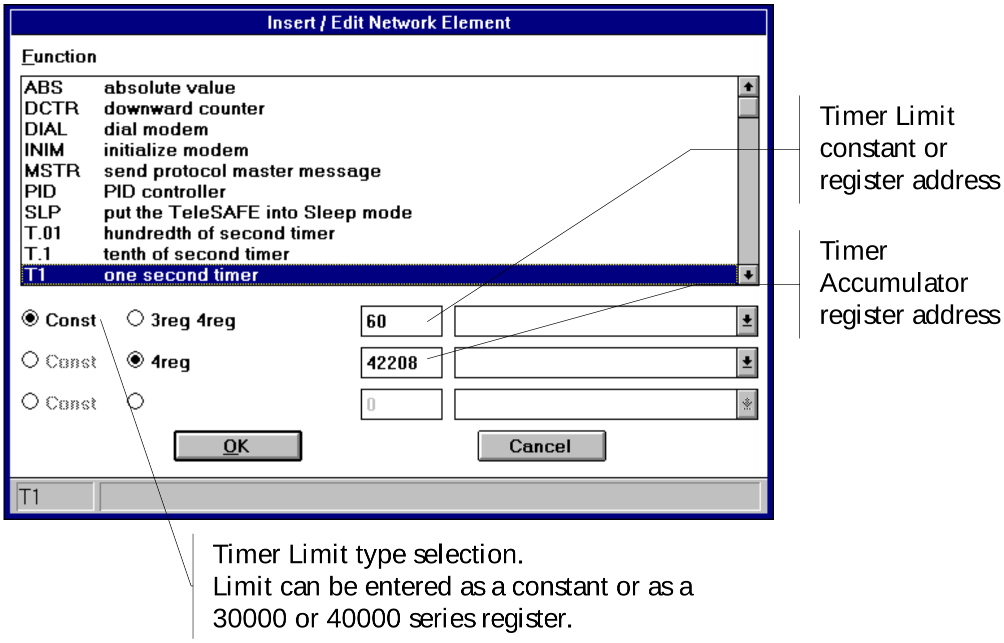


Figure 40: Insert / Edit Timer Function

Complete the network by inserting the elements as shown in *Figure 41: Timer Network*.

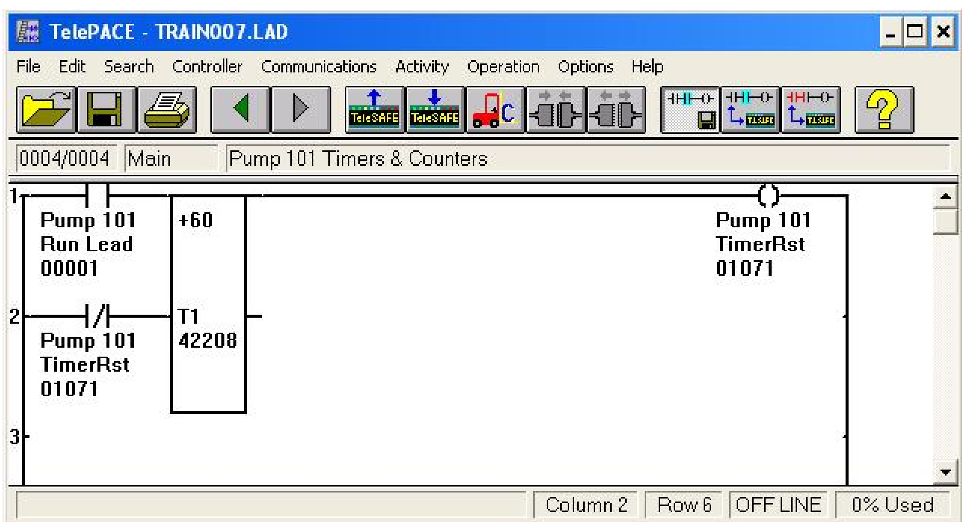


Figure 41: Timer Network

- Load the modified program into the and run it.
- Adjust the analog input such that coil 00001 is turned on.
- Use the Monitor On line function and monitor register 42208.

? ***When is coil 01071 energized?***

? ***Why is contact 01071 normally closed?***

- Adjust the pot simulating the tank level such that coil 00001 is turned off.

? ***What happens to the value in the accumulator register when coil 00001 is turned off?***

? ***What happens to the value in the accumulator when the program is stopped and then restarted?***

Step 2 Accumulate the Pump On Time

In this step the one-minute intervals will be accumulated. There are several ways to accumulate the intervals. In this program an ADDU function will be be used to demonstrate how a totalizer can be constructed using the TelePACE Ladder Logic.

? ***What are some other ways that the intervals could be accumulated?***

To accumulate the number of minutes that the pump is on each one minute output from the timer will added to a register. The value in the register used will increment by one on each one-minute count.

The TelePACE Ladder Logic functions used for addition are the ADDU function for unsigned addition and the ADD function for signed addition. In this program unsigned addition will be used to accumulate the one-minute increments.

- Insert an ADDU function to the right of the Timer function, between the Timer function and coil 01071.
- Configure the ADDU function as shown in *Figure 42: Insert / Edit ADD Function*.
- The network should appear the same as shown in *Figure 43: Accumulate One-Minute Outputs*.
- Load the program into the controller and run it.
- Using the Monitor On line function monitor the accumulator register.

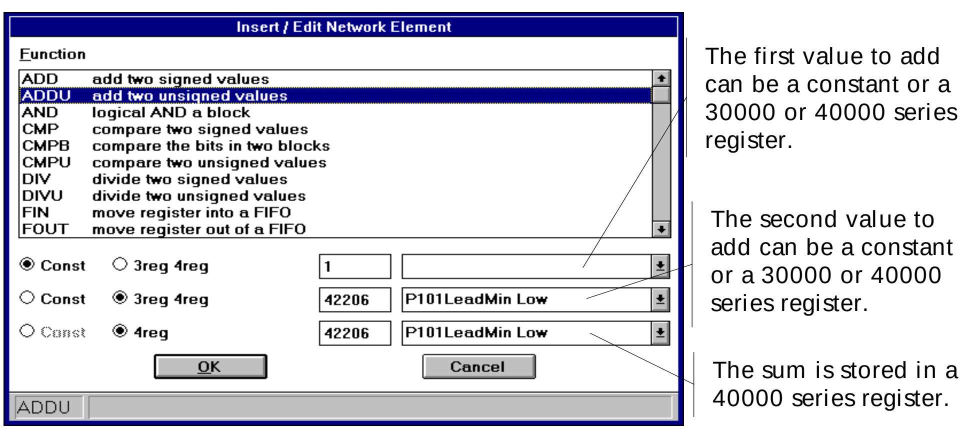


Figure 42: Insert / Edit ADD Function

? ***How does the ADDU function work as an accumulator?***

? ***Why is only one analog output register required in the ADDU function?***

? ***What is the limitation of this accumulator?***

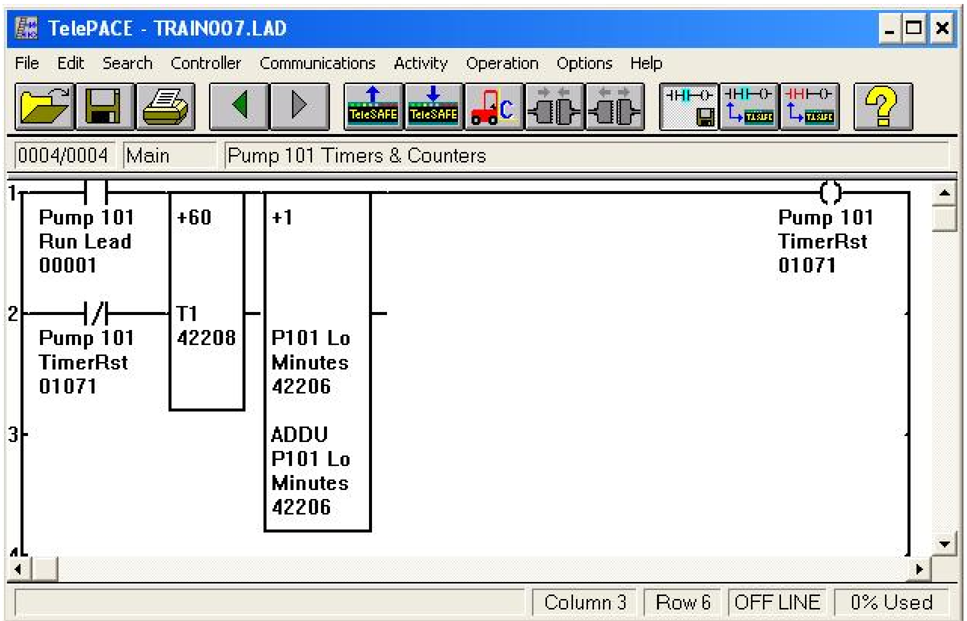


Figure 43: Accumulate One-Minute Outputs

- Insert another ADDU function and connect the *enable* input to the *result out of range* output of the first ADDU function.

? ***What does this change in the network do?***

? ***How can the operation of the accumulator be tested.***

- Insert a DIVU function at the start of rung 3. This will divide the minutes totalizer (low word | high word) by 60 to generate a Run Hours value.

? ***Why does the DIVU function accept a double word value as its first input.***

- The network should appear the same as shown in *Figure 44: Completed Pump Run Timer Network*.

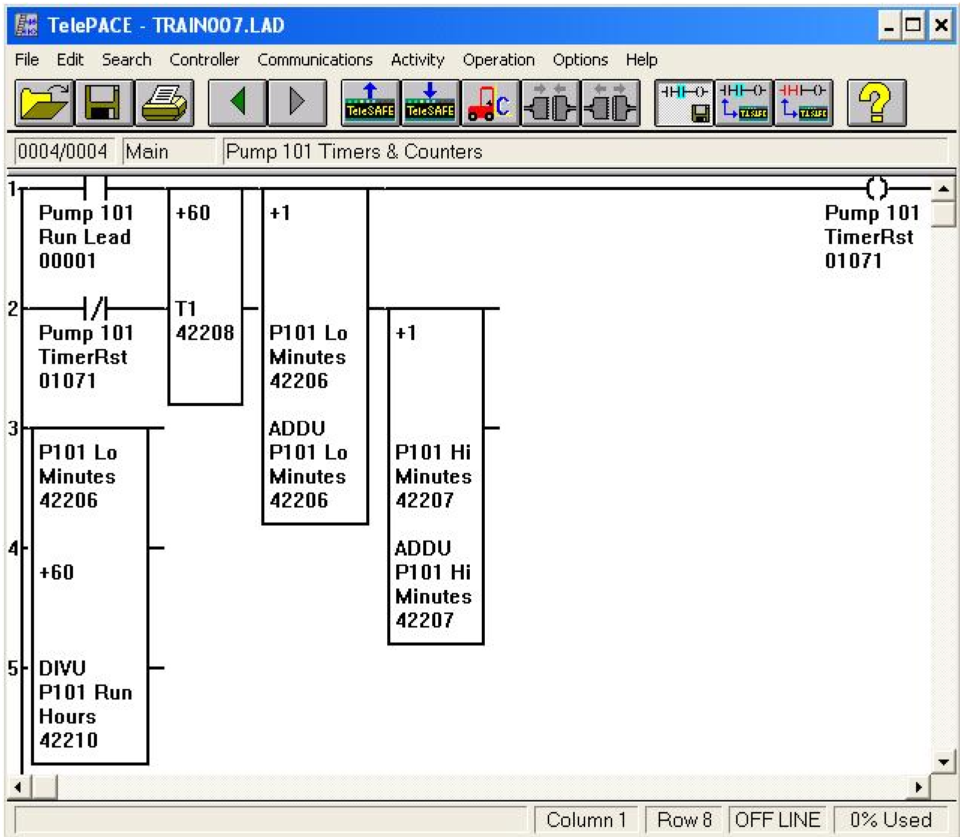


Figure 44: Completed Pump Run Timer Network

- Save the program as TRAIN007.LAD.

Step 3 Count the number of Pump Starts

The last step of this part of the training course is to modify the program to count the number of pump starts. There are two functions in the TelePACE Ladder Editor that are used to count. The UCTR is an up counter and is used to count up to a limit value. The DCTR is a down counter and is used to count down from a preset value. The UCTR function will be used to count the number of pump starts.

- Insert a network after network 4.
- In row 1 Column 1 insert the normally open contact 00001.
- In row 1 Column 2 insert a UCTR function and configure it as shown in figures 45 and 46.
- In row 2 Column 1 insert a horizontal shunt.
- Load the program into the controller and monitor the operation of the program.

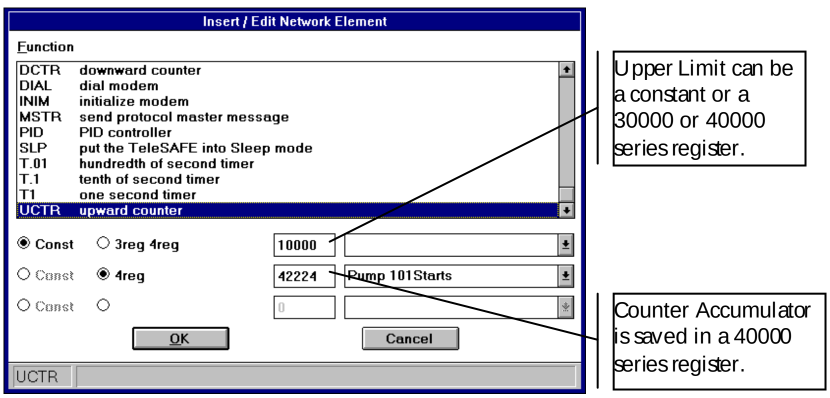


Figure 45: Insert / Edit UCTR Function

? ***What is a limitation of this upwards counter.***

- Add a coil to the upper output (counter = limit) of the UCTR as shown in Figure 46.
- Replace the horizontal shunt with a normally closed contact, with the same address as the coil.

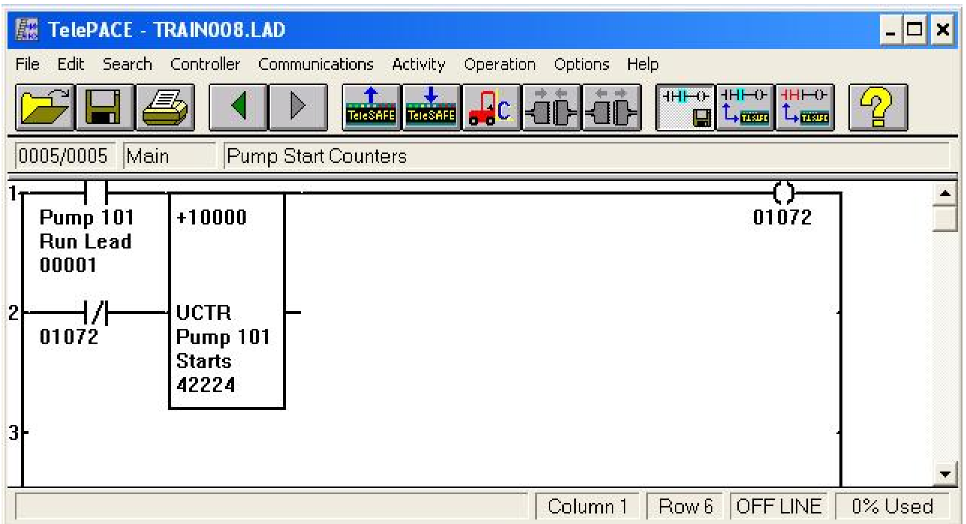


Figure 46: Completed Pump Start Counter Network

- The network should appear as shown in Figure 46: Completed Pump Start Counter Network.
- Load the program into the controller and monitor the operation of the program.
- Save the program as TRAIN008.LAD.

Part Four: Add Lag Pump Control

In parts one, two and three of this programming example the training.lad program has been developed to control the operation of a pump.

In the program developed so far:

- The pump starts when the analog input 30001 value is 70.0% of maximum.
- The pump stops when the analog input 30001 value is 90.0% of maximum.
- The run time of the pump is measured and accumulated.
- The number of times the pump starts is saved.

In many applications using pumps it is often required to have a second pump to back up the first pump if the first pump cannot maintain the setpoint. In this configuration the pump that is started first is the lead pump and the back up pump is the lag pump. This type of control is called Lead / Lag pump control.

In this part of the training course the training.lad program is going to be modified to add Lead / Lag pump control. The sequence for starting and stopping the Lead and the Lag pumps will be:

Stop Lead Pump	When the value of analog input register 30001 is equal to or greater than 90.0% of the maximum value.
Stop Lag Pump	When the value of analog input register 30001 is equal to or greater than 80.0% of the maximum value.
Start Lead Pump	When the value of analog input register 30001 is equal to or less than 70.0% of the maximum value.
Start Lag Pump	When the value of analog input register 30001 is equal to or less than 60.0% of the maximum value.

Modifications to the program will include a Lag pump run timer and start counter. Use the Following steps to modify the training.lad program.

1. Modify network 1 to include the Lag pump start and stop setpoints.
2. Add a network to control the Lag pump start and stop. Insert this network immediately after the Lead pump start and stop control network.
3. Add a network to accumulate the Lag pump run time. Insert this network immediately after the Lead pump run timer network.
4. Add logic to count the number of Lag pump starts into the timer network.
5. Load the modified program into the controller.
6. Use the Monitor On line function to test the program.
7. Save the tested program as TRAIN009.LAD.

Part Five: Alternating Lead / Lag

In this last section of the TelePACE Ladder Logic programming example the program is modified to switch the lead and lag pumps every 50 hours. When a pump is selected for lead operation it will run as the lead pump for 50 hours and then the lag pump will become the lead pump and it will run for 50 hours. The pumps will cycle as lead and lag, every 50 hours, continuously.

To begin this part of the programming example the Ladder Logic program TRAIN010.LAD has been created. In this program the lead and lag run timers have been changed to increment the accumulated minutes registers every 2 seconds. The lead and lag switching will occur every 10 seconds.

- Open a new file in the TelePACE Ladder Editor.
- Initialize the I/O Database in the controller.
- Load the program TRAIN010.LAD into the controller.
- Run the program and monitor the program execution.
- Adjust the analog input to 65% and monitor program execution.
- Adjust the analog input to 50% and monitor program execution.
- Adjust the analog input to 85% and monitor program execution.

Using the tools available determine how the program works.

The instructor will explain the program execution after the class has had the opportunity to thoroughly examine the program.

- Save the Ladder Logic program as TRAIN010.LAD.
- Pat yourself on the back for a job well done!

Implementing Subroutines in TelePACE

A subroutine is a group of ladder logic networks that may be executed conditionally. Each subroutine begins with a network containing an **SUBR** element on its first rung. A subroutine ends when another subroutine element is encountered, or when the end of the ladder logic program is reached. In order to use a subroutine, a **CALL** element with the same subroutine number is enabled by the program.

Ladder logic programs can grow to include many networks, causing the scan time to become longer than may be desired. The use of subroutines allows the programmer to decide when any part of the logic will be executed. One subroutine might need to be executed once per second, while another only once per hour. This can dramatically reduce the controller's scan time.

If a given piece of code needs to be executed repeatedly during a single scan cycle, one subroutine may be called repeatedly with multiple CALL elements instead of wasting memory on multiple instances of the logic. With only 12k words available for ladder logic, this can become important.

Digital and analog outputs that are set in a subroutine remain in their last state when that subroutine is not called. For example, a coil that is turned on by a subroutine remains on when the subroutine is disabled. The output will only turn off when the subroutine is called and it turns the output off.

Subroutines do not have to be programmed in any particular numerical order. For example, subroutine 1 can follow subroutine 2, subroutine 200 can follow subroutine 400.

A subroutine can call other subroutines. This is called nesting. The maximum level of nesting is 20 calls. In practice, more than two or three levels of nesting can become quite confusing.

Subroutine calls cannot be recursive. For example, subroutine 1 cannot call itself or call another subroutine that calls subroutine 1. This prevents potential infinite loops in the ladder logic program.

See Figure 12 for a visualization of the scan cycle of a program with subroutines.

The following exercise will provide a simple demonstration of the use of a subroutine in a program. It will then be expanded to include a second nested subroutine.

Subroutine Exercise

Step 1 Create Tag Names

- Open a new file in TelePACE.
- Enter the following tag names using the Edit Tag Name dialog or import them from the supplied SUBRDEMO.CSV file.

Tag Name	***Register Address***
Motor 1 Start	10001
Enable Sub 100	10002
Enable Sub 200	10003
Alarm Trigger	10004
Motor 1 Run	00001
Pulse Output	00002
Alarm Output	00003

Step 2 Create Main Program and Subroutine

The main program will simply allow the user to control a motor with a switch. The main program could be many networks in length, but this short example will suffice.

- Select the appropriate Controller Type in the Controller menu.
- Create a Default Register Assignment with the appropriate lower I/O board.
- Insert the single rung of main program logic shown below, along with the CALL in rung 2.
- Note that a warning is generated when the CALL is inserted before the matching SUBR. In an offline edit this is acceptable. In an online edit, however, it is not allowed to insert the CALL first as a jump to a non-existent subroutine could cause the controller to lock up.
- Note that the network type is “Main.”

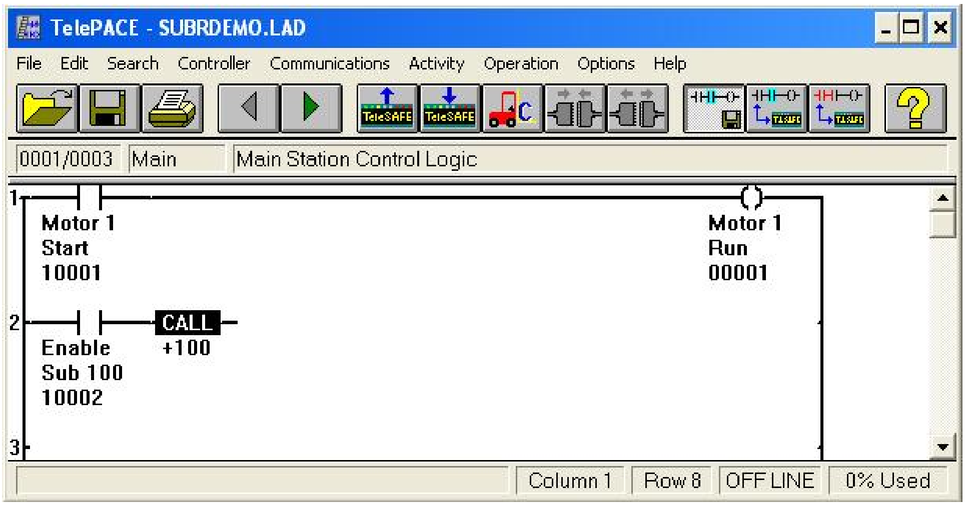


Figure 47: Main Program Control Network

- Go to Edit | Insert and create a new Network After… the current network.
- Note that this network's type is currently "Main."
- Insert the logic as shown in Figure 48.
- Note that the network's type is now shown to be "Sub 100."
- Save the program as SUBRDEMO.LAD.
- Download the program to the controller and monitor program operation.

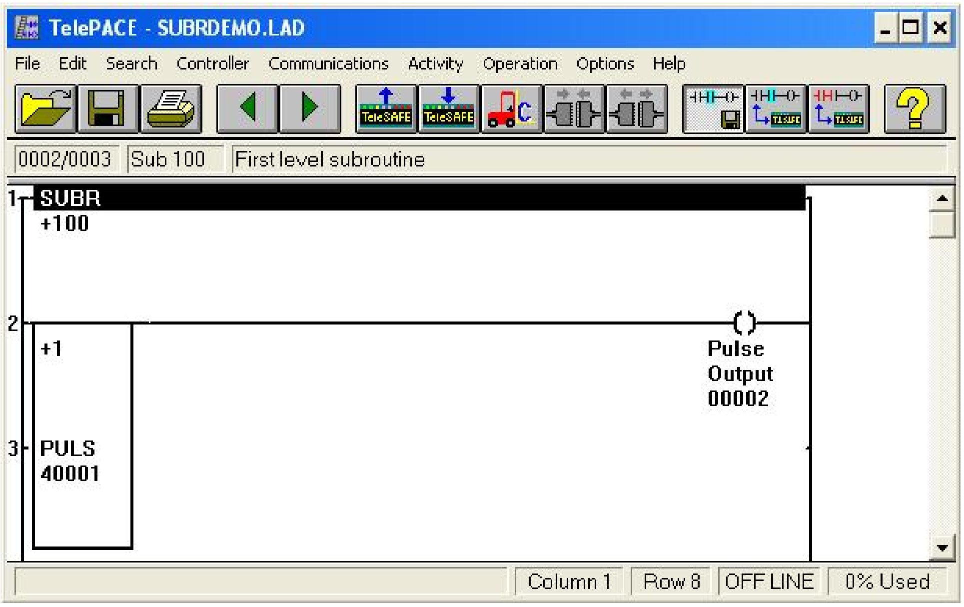


Figure 48: Subroutine 100 Network

- Note that the main program logic will execute at all times. (ie switch 1 should always be able to control the first LED)

- Note that the second LED will only flash when Subroutine 100 is enabled with switch 2.
- See that the power rail and SUBR label only turn red (active) when the subroutine is enabled.
- Also note that if the subroutine is disabled (switch 2 turned off) while the second LED is on, that LED will remain on until the subroutine is again enabled and the PULS is activated.

Step 3 Create a Nested Subroutine

- Go to Edit | Insert and create a new Network After… network two.
- Note that the network type states "Sub 100" at this time.
- Place an SUBR instruction in rung 1 of the new network, which now names this subroutine to be "Sub 200."
- Add a fourth switch controlling the third LED, as shown in Figure 49.
- Go back to the first subroutine in network 2, and add a second CALL instruction controlled by a switch, as shown in Figure 50.
- Note that the warning seen when the previous CALL was inserted did not occur this time. This is because the subroutine being referenced did exist before the CALL was added.
- Download the program to the controller and go online to monitor execution.

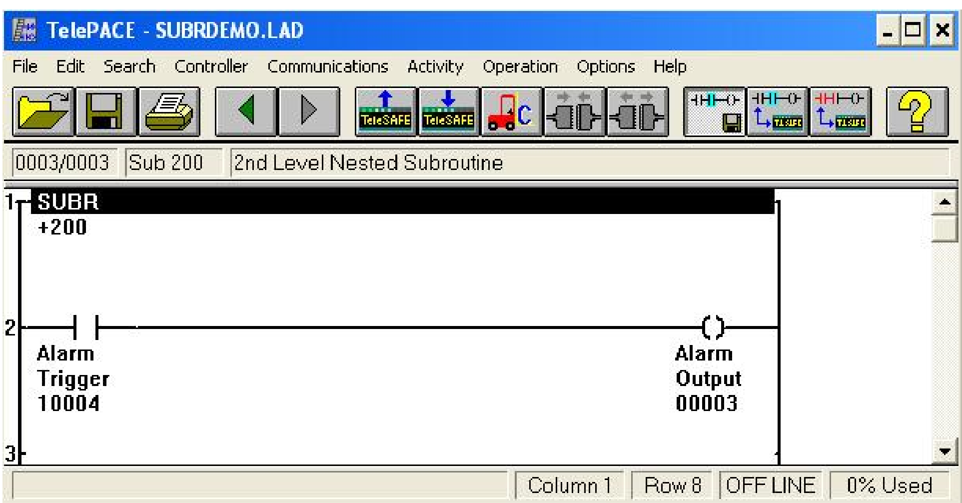


Figure 49: Subroutine 200 Nested Network

- Save the program as SUBRDEMO.LAD.

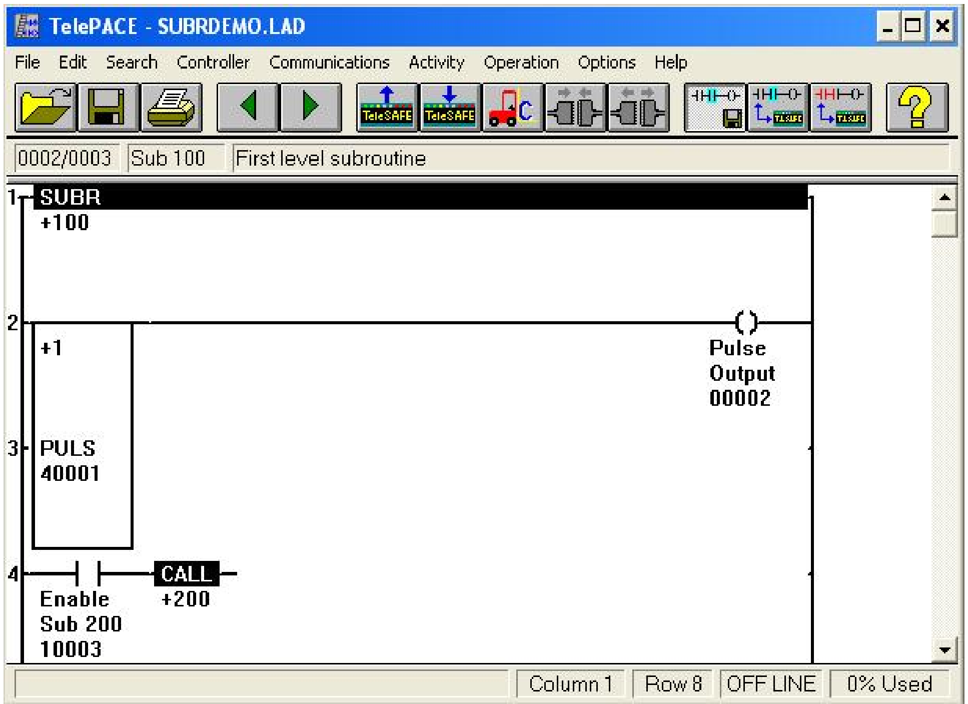


Figure 50: Subroutine 100 Network with CALL

- Note that the main program logic still works, as it always will.
- With Subroutine 100 disabled (switch 2 turned off), note that turning on the third switch does not cause Subroutine 200 to be executed. This is normal. Subroutine 100 is not being called, and therefore Subroutine 200 will not operate either.
- Turn on switch 2, which enables Subroutine 100. Notice that the second LED flashes.
- Turn on switch 3, which now enables Subroutine 200. Note that the power rail and SUBR label in the subroutine only now become red. (active)
- Only at this time should switch 4 be able to turn on the third LED, as both subroutines are now enabled.
- Note that if either or both of switches 2 and 3 are turned off, Subroutine 200 will be disabled. LED 3 will remain fixed in its current state until the subroutine is reactivated.

Flow Accumulation

The FLOW function accumulates flow from pulse-type devices such as turbine flow meters. This function is designed to be used in the measurement of products that do not require a complex flow calculation. If any correction factors must be applied, such as for temperature or pressure, that must be done in separate logic in the controller.

The only correction that is applied is the K Factor. This is a value supplied by the manufacturer of the meter, which specifies the number of pulses per unit of product. For example, "103.45 pulses per gallon."

The turbine meter or other pulse source is wired to a counter input on the SCADAPack. An appropriate entry must be added to the Register Assignment, which provides two registers for a double unsigned integer (32 bit) value to store the pulse accumulator. Incoming pulses are detected by the counter circuitry, and each time a pulse is seen the count value is incremented by one.

The pulse accumulator register is monitored by the FLOW function, which divides by the K Factor to calculate the current flow rate. This is done on each scan of the controller logic. The flow rate updates once per second if there is sufficient flow. If the flow is insufficient, the update slows to as little as once every ten seconds to maintain resolution of the calculated rate.

The FLOW function then accumulates the measured flow over a period of time. When the Log Data input goes from OFF to ON, the accumulated volume, flow time and the time at the end of the period is saved in the history registers. Older history is pushed down and the oldest record is discarded. The Log Data input must be triggered at least once every 119 hours (at the maximum pulse rate). Otherwise the volume accumulator will overflow, and the accumulated volume will not be accurate.

Typically the Log Data input will be toggled once per day, to capture the accumulated flow during that day. It can, however, be toggled at any desired time if the program is accumulating flow for other purposes. There are a maximum of 35 sets of history records, for 35 days or periods of accumulation.

In the following exercise, the turbine meter will be simulated by a pulse generator on a 5699 demonstrator I/O board. If the 5699 board is not available, the class may instead add a T.01 timer with a Limit of 65535 to the program. This timer will need to be self-resetting. The Accumulator register may then be monitored as a 16 bit accumulator.

FLOW Exercise

Step 1 Create Tag Names

- Open a new file in TelePACE.
- Enter the following tag names using the Edit Tag Name dialog or import them from the supplied FLOWDEMO.CSV file.

Tag Name	***Register Address***
Error	01001
Accum Flow	10001
Log Data	10002
Enable Accum	10003
Counter Input	30011
K Factor	41001
Input Register	41003
Input Type	41004
Rate Period	41005
Status	41006
Flow Rate	41007
Number Records	41017
Volume Period 1	41018
End Time Period1	41020
FlowTime Period1	41022

Step 2 Build Ladder Logic

The required ladder logic will be developed in this step, and the FLOW function will be configured. The FLOW function has an Element Configuration dialog where required setup parameters may be entered.

- Select the appropriate Controller Type in the Controller menu.
- Create a Default Register Assignment with the appropriate lower I/O board.

- Delete the entry DIN Controller Digital Inputs, then replace it with CNTR Controller Counter Inputs. Assign a Start Register of 30011.
- Enter the ladder logic as shown in Figure 51.

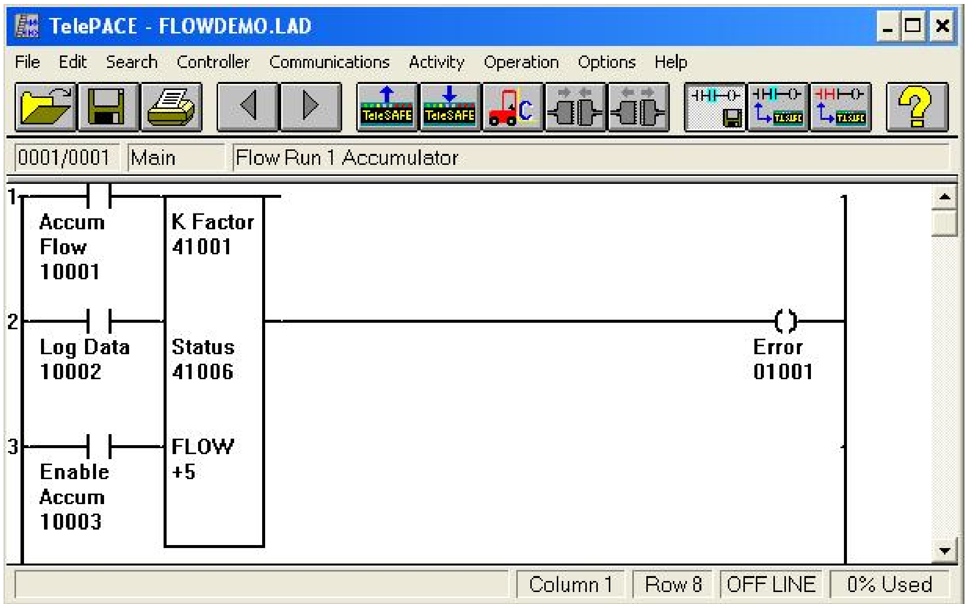


Figure 51: Ladder logic for FLOW demonstration

- Select the FLOW function by clicking on it once with the left mouse button.
- Right click, then select Element Configuration.
- Complete the FLOW function Element Configuration as shown in Figure 52.
- Save the program as FLOWDEMO.LAD.
- Ensure that switch 3 is turned on, to Enable the Flow function, and that switches 1 and 2 are both off before running the program.
- Download the program to the controller, then go online to monitor execution.

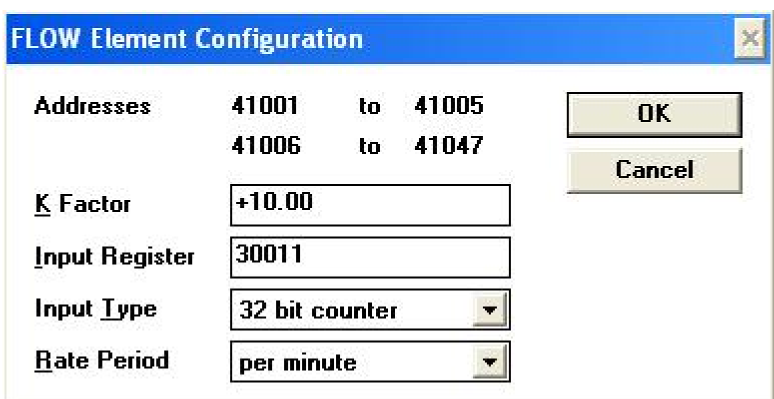


Figure 52: FLOW function configuration

Step 3 Monitor Program Operation

With a pulse signal being accumulated by the SCADAPack, it will be possible to demonstrate the operation of the FLOW function. The student may vary the measured flow rate by adjusting the pulse frequency. Use the Register Editor to monitor the flow rate and accumulation.

- Once online with the controller, create a new Group in the Register Editor.
- Add register 30011, the pulse accumulator, to the group using the Unsigned Double format.
- Select the FLOW function, then right click on it. Select Monitor Element. This will add all of the configuration registers and the output registers to the Register Editor group, with each register in its correct data format.
- Registers 41009, 41011, 41013 and 41015 are not required for this demonstration. Select them and hit the Delete button to remove them from the Register Editor group.
- Check register 30011. Its value should be incrementing at a rate set by the Counter Freq potentiometer on the 5699 I/O board. Adjust this pot and verify that the pulse accumulator rate also varies. It should vary beween approximately 5 and 150 pulses per second (Hz).
- Note that registers 41001 through 41005 are the configuration registers, as entered in the FLOW function's Element Configuration dialog.
- Note that the Flow Rate, in register 41007, should be 0 at this point. The Accumulate Flow switch is still off, so the FLOW function is not operational.
- Set the Counter Freq potentiometer fully clockwise. This will set the input pulse rate to approximately 150 Hz.
- Turn on switch 1. This causes the FLOW function to begin accumulating flow.
- Monitor register 41007, the Flow Rate. Note that the flow should be approximately 900, if the pulse rate is 150 Hz and the K Factor is 10. The Rate Period is set to Per Minute. Thus the flow rate is calculated as: (150Hz / 10) * 60 seconds = 900. This may mean 900 gpm, 900 m3/min, or perhaps 900 barrels/min.
- Set the Counter Freq potentiometer fully counter-clockwise. This will be about 5 Hz. If the pulse rate is too slow, the FLOW function will generate Status code 5, "Pulse rate is too low for accurate flow rate calculation." The Flow Rate should read approximately 30. Note that at this low pulse rate the flow calculation will only update once every 10 seconds.
- Increase the flow rate until the Status register returns to 0. (No error)
- Note that register 41018, (Volume Period1) is incrementing. This is the current period, and so the value is live. It only stops if the flow stops or the Accumulate Flow input is turned off.
- Note that the End Time Period1 register is showing the current time in Unix time format.
- Note that the FlowTime Period1 register counts the number of seconds of flow in the period.
- Simulate the end of a day by toggling the Log Data input on and off with switch 2.
- Note that the three Period 1 values are frozen at this point, then pushed down to period 2.
- New Period 1 values begin to accumulate as the new day begins.
- Once the Log Data input has toggled 5 times the original day's data will be pushed out the bottom of the history. This data is now lost.
- Experiment with new values for the K Factor and the Rate Period. Note that the K Factor should not be changed while flow is being accumulated. First turn the Accumulate Flow input off to avoid flow calculation errors.

PID Feedback Loop Control

A PID controller compares a desired value (Setpoint) against a real-world measured value (Process Value). Any error detected will cause a corrective output from the controller to occur, in an attempt to reduce the error. This is called a feedback loop. The PID controller uses three tuning parameters called Proportional, Integral and Derivative to adjust how the controller reacts. These are also known as Gain, Reset Time and Rate.

The PID controller to be used in the following exercise is the PIDA function. This controller has an analog-type floating point output, which can be sent to one of a SCADAPack's analog out channels. This will send a 4-20 mA or 1-5 V signal to a control device such as a flow control valve.

The PIDA Process Value is taken from an external device such as a flow meter, a pressure transmitter or an RPM indicator. This value is typically input to the controller through a 4-20 mA analog input, or perhaps through a modbus message from the measurement device. The Setpoint is typically entered from an HMI screen, if its value needs to be changed. The tuning parameters are not normally available to the site operator, other than perhaps after entering a password.

A PID controller typically also has a manual mode, in which the operator can take control of the output and set it to any desired value. In the PIDA function a digital input is available to allow switching between Auto and Manual modes, and a Manual mode value may be entered from an HMI.

In the following demonstration, no actual feedback is possible. It will be up to the student to simulate feedback by adjusting the analog input potentiometer on their demonstrator I/O board.

Tuning of a PID loop can be a complex and involved process, beyond the scope of this document. However, included here is a brief discussion of some of the key parameters.

Proportional Gain – Increasing Gain tends to reduce the Error. If Gain is increased too far, however, the loop will begin to cycle or oscillate. Gain alone can never reduce the Error to zero.

Integral or Reset Time – Reset is added to eliminate the Error. As Reset time is decreased the controller output becomes more likely to oscillate. The oscillation period also tends to increase. In the PIDA function it is measured in Seconds per Repeat.

Derivative or Rate Time – This parameter is typically not used in applications such as flow control. If not used, its value must be set to 0 (zero) to disable it. Rate allows a degree of predictive action, in order to reduce dead time. This is the time between a PID output change and a resulting change in the process.

Introductory Tuning – The tuning of a PID loop can be very complex, particularly if Rate is involved. Below is a simple introductory technique to set the tuning. This brief discussion will not include Rate.

- Initially set the Reset value to a very high setting, for example 1000 seconds.
- While the PIDA is controlling the process in Auto mode, increase the Gain slowly until the PID Output begins to oscillate.
- Measure the oscillation time. This is the Natural Period of the loop.
- Divide the Gain used at this point by 3, and use this value for Gain.
- Use the Natural Period of oscillation as the Reset Time.

CAUTION: The procedure above is a very basic guide for training purposes **ONLY**. Anyone performing a PID tuning procedure under actual field conditions must understand the process and follow all appropriate safety precautions.

PIDA Exercise

Step 1 Create Tag Names

- Open a new file in TelePACE.
- Enter the following tag names using the Edit Tag Name dialog or import them from the supplied PIDADEMO.CSV file.

Tag Name	***Register Address***
Auto/Man Mode	10001
FlowRate Raw	30001
PID Out Raw	40001
SCAL Config	41001
SCAL Output	41010
Raw Out Range	41101
Raw Out Float	41103
Process Value	42001
Setpoint	42003
Gain	42005
Reset Seconds	42007
Rate Seconds	42009
Deadband	42011
Full	42013
Zero	42015
Cycle Time	42017
Manual Output	42019
PIDA Output	42021

Step 2 Build Ladder Logic

This program will take an analog input from the demo I/O board and convert it to engineering units. The input will represent flow through a pipeline, with a minimum scaled value of 0 at 6552 and a maximum of 300 at 32760. The PIDA function will calculate a floating point output, with a range of 0 – 100%. This will be used to drive a control valve. The floating point number will then be converted back to a raw value and sent to one of the controller's analog outputs.

- Select the appropriate Controller Type in the Controller menu.
- Create a Default Register Assignment with the appropriate lower I/O board.
- Click Add, then insert the SCADAPack AOUT module. Assign a Start register of 40001.
- Enter the ladder logic as shown in Figure 53.

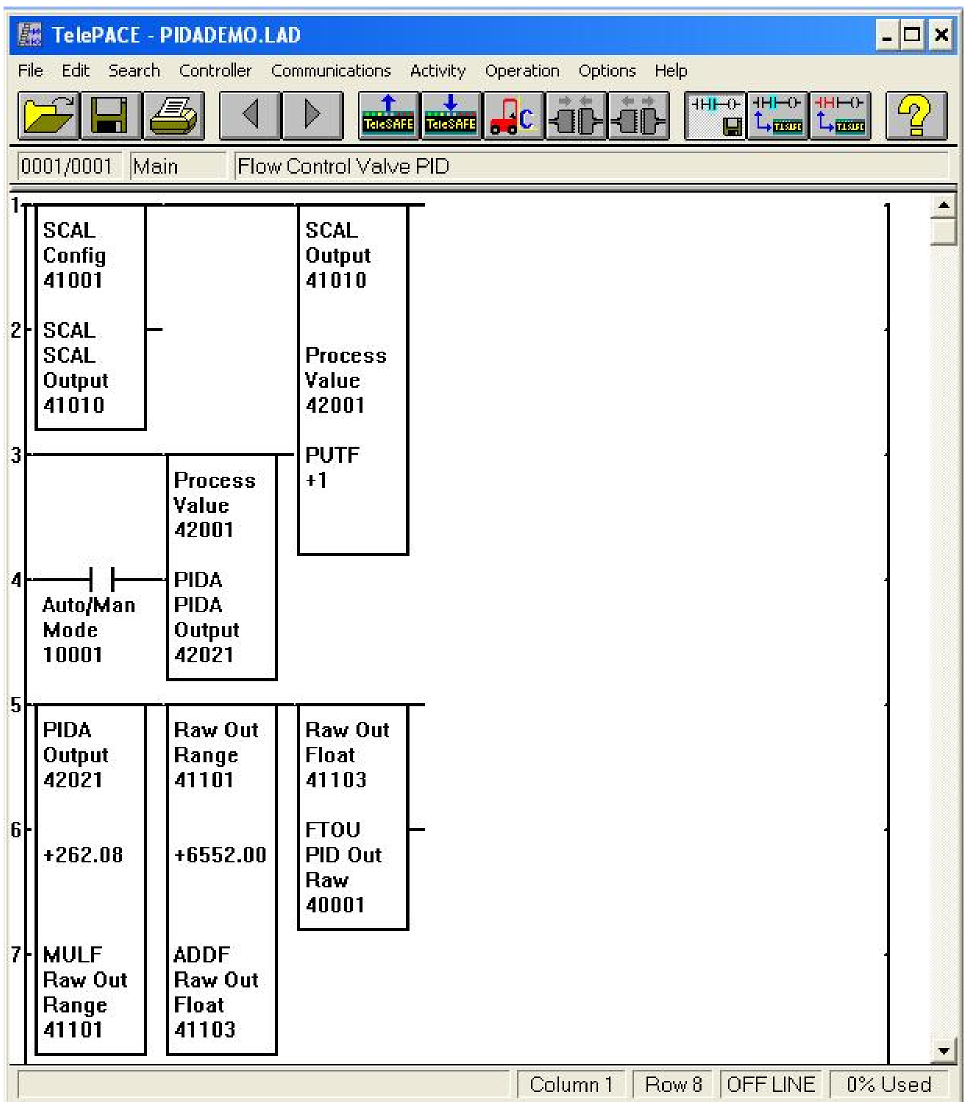


Figure 53: Ladder logic for PIDA demonstration

- Select the SCAL function by clicking on it once with the left mouse button.
- Right click, then select Element Configuration.
- Complete the SCAL function Element Configuration as shown in Figure 54.

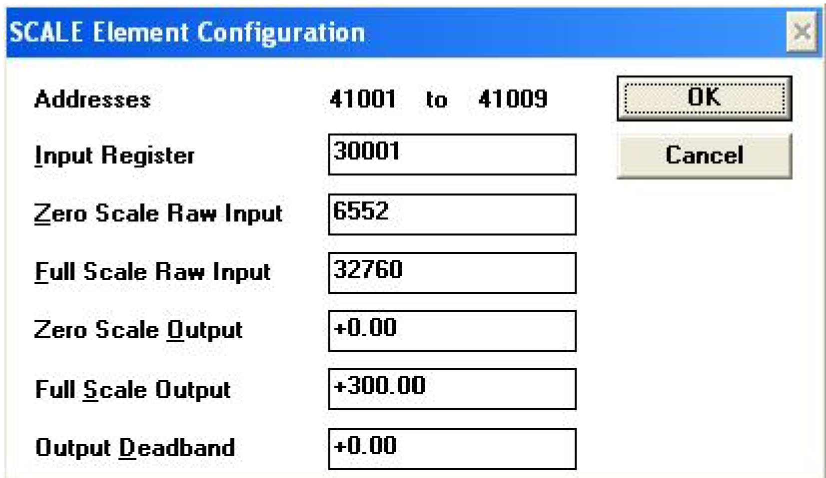


Figure 54: SCAL function configuration

- The analog input circuitry will convert a 4-20 mA signal into a raw analog value between 6552 at 4 mA and 32760 at 20 mA. The offset is because the circuitry will read as low as 0 mA, providing a raw value of 0 at that point.
- The SCAL function converts the raw values between 6552 and 32760 into scaled values of between 0 and 300 units of flow. The student may presume whatever flow measurement units they are familiar with. (eg gal/min, m3/hr, e3m3/day, etc)

- Select the PIDA function by clicking on it once with the left mouse button.
- Right click, then select Element Configuration.
- Complete the PIDA function Element Configuration as shown in Figure 55.

- Save the program as PIDADEMO.LAD.
- Ensure that switch 1, the Auto/Man Mode selector, is OFF for Manual mode.
- Download the program to the controller, then go online to monitor execution.

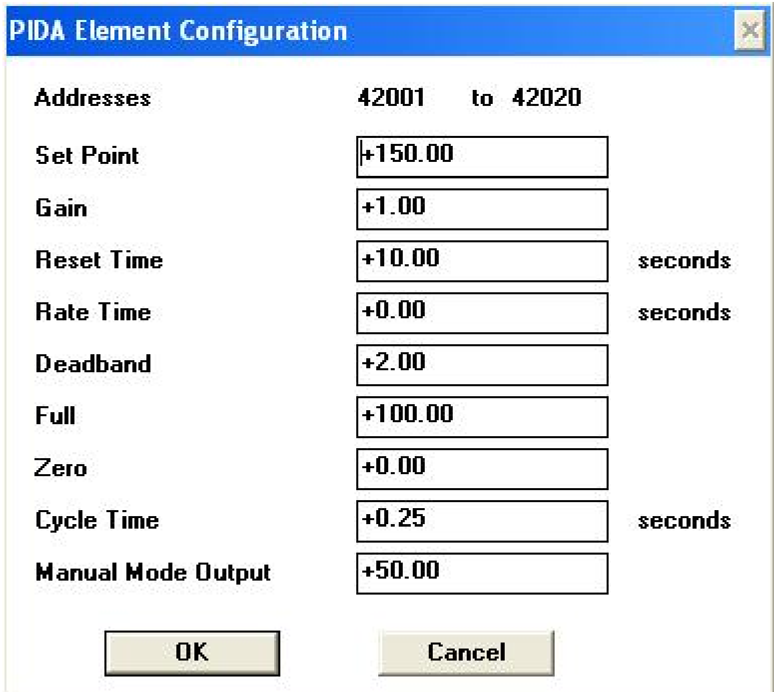


Figure 55: PIDA function configuration

- The PIDA function, when they Auto/Man Mode switch is On, calculates a PID Output value. When the switch is off, the Manual Output value is sent directly to PID Output.
- The entered Setpoint value specifies the desired flow rate. The PID Output will increase, to open the valve, if the flow rate drops below the setpoint. The PID Output will decrease, to close the valve, if the flow rate goes above the setpoint.

- The PID Output value is a floating point number, ranging from 0 – 100%. It is multiplied in a MULF by 262.08, to scale it up to the raw range. (32760 – 6552 = 26208)
- A value of 6552, equal to the 4 mA point or 0% valve rotation, is added to the raw range value. This produces a value at 0% of 6552, and at 100% of 32760.
- The last step is to convert the floating point raw value to integer, and send it to register 40001. This was assigned to Analog Out 0 in the Register Assignment.

Step 3 Monitor Program Operation

With an analog input signal coming from the demo I/O board, simulating flow rate as the Process Value, it is possible to demonstrate a PID loop. The student may vary the flow rate and then watch the PID Output change in reaction. Use the Register Editor to monitor the Process Input (flow rate), the Setpoint and the resulting PID Output value.

- Once online with the controller, create a new Group in the Register Editor.
- Press Add to manually add register 30001, the raw analog input representing flow rate, and 40001, the raw analog output to the control valve.
- Select the PIDA function, then right click on it. Select Monitor Element. This will add all of the configuration registers and the output registers to the Register Editor group, with each register in its correct data format.
- Registers 42023 through 42029 are not required for this demonstration. Select them and hit the Delete button to remove them from the Register Editor group.
- Monitor register 30001, the raw flow rate, and 42001, the scaled Process Input values at the same time as the analog input potentiometer is adjusted. Note that at a raw value of 6552 the scaled value should be 0, and at a raw value of 32760 the scaled value should be 300. Also note that if the raw value is reduced to zero, the scaled value is -75. Why is this?
- With the Auto/Man Mode switch in Manual mode, the PID Output will immediately follow the value entered into the Manual Mode Output register. If it is desired to have the value sent to the valve change more slowly, additional logic will be required to ramp the value up or down.
- Note that when a Manual Mode Output value of 0 is entered the PID Out Raw value, register 40001, goes to 6552. When 100 is entered the raw output value goes to 32760.
- Enter a Manual Mode Output value of 110. Note that the PID Output goes to 100. Try a value of -10. The PID Output will go to 0. It is clamped at these points by the Full and Zero settings.
- Adjust the analog input pot so that the difference between the Process Input (flow rate) and the Setpoint (the Error) is less than the Deadband.
- Now switch the PID controller into Automatic mode by turning on switch 1.
- Note that the PID Output does not change. This is because the error is less than the Deadband, thus no new calculation is done.
- Turn the potentiometer counter-clockwise to reduce the Process Value to a value slightly outside the Deadband range. Once the Error exceeds the Deadband, note that the PID Output begins to slowly rise. If the Process Value is flow rate, and the flow drops below the Setpoint, then the flow control valve must begin moving open. This allows more product through the valve, thus increasing the flow rate.
- Turn the pot farther counter-clockwise, and note that the PID Output begins to rise more quickly. If the student does not reduce the error by turning the pot clockwise, the PID Output will continue to rise until it is clamped by the Full value (100%). This would cause the control valve to rotate to its full open position.
- Turn the pot back clockwise until the Process Value rises above the Setpoint. Now the flow rate on the pipeline is too high, and as a result the valve needs to begin closing to allow less product through. As a result the PID Output should begin dropping.
- At this point, try changing the values of Gain and Reset. Only change one at a time. Note that increasing Gain or reducing Reset will both cause the PID loop to react more vigorously.

Data Logging and SCADALog

The DLOG function block is used to populate a data log in a SCADAPack controller. Each time a low to high transition occurs on its Grab Data input a record is generated. A record consists of a number of data fields. As many as 16 data logs may exist in a controller at any time, and each data log may contain anywhere from one to eight fields. This will allow for as many as 128 data points to be captured in a single SCADAPack.

The log contents may be deleted at any time by toggling the Delete Log input low then high again. This may be done after data is read out of the log using the GETL function if desired. If the SCADALog software is instead used to read logs, the log contents may be automatically deleted after the logs are read if desired.

Each data log field may be assigned to one of six data types. These include 16 bit unsigned and signed integer, 32 bit unsigned and signed integer, 32 bit floating point and Time & Date. There is no option to log boolean values, but if this is desired a block of booleans may be transferred into a holding register using the MOVE function.

The Time & Date data type is stored in two 32 bit unsigned integer registers. If the Time & Date is read using the GETL function the user will need to interpret the data manually. SCADALog will convert it to actual time and date information automatically. The first 32 bit register contains the number of complete days since 01/01/97. The second 32 bit register contains the number of hundredths of a second since the start of the current day.

Each data log record is automatically assigned a sequence number, starting at 0. This is a 32 bit unsigned value. If SCADALog is used, there is no need to deal with the sequence number. If GETL is used, the programmer must track the sequence number to know its current value. (using a counter triggered by the Grab Data source) The programmer must specify the sequence number of each record to be retrieved.

The programmer must be aware of the maximum available memory for data logs in the controller they are working with. This is documented in the DLOG function Help section of the TelePACE manual. For example, a data log record containing Time & Date (4 words), two floats (2 words each) and two integers (1 word each) would require 10 words. Each time the Grab Data input is toggled this amount of memory is used.

DLOG Exercise

Step 1 Create Tag Names

- Open a new file in TelePACE.
- Enter the following tag names using the Edit Tag Name dialog or import them using the supplied DLOGDEMO.CSV file.

Tag Name	***Register Address***
Enable Get Log	10001
Read Log Data	10002
Analog Input 0	30001
Status Inputs	41001
Data Log Config	41201
Data Log Status	41219
Get Log Sequence	41301
Get Log Error	41303
Sequence Echo	41304
Record Length	41306
DaysFrom 01/01/97	41307
100ths Of Secs	41309
AIN 0 Field	41311
DINs Field	41312

Step 2 Build Ladder Logic

This program will gather data every two seconds, triggering the DLOG's Grab Data input with a PULS function. The DLOG has three fields: Time & Date, Analog Input 0, and Status Inputs. As there is no boolean data type available, a MOVE function is used to copy the first 16 digital inputs into a holding register. The GETL function is included to demonstrate its use in retrieving log records and placing them into modbus registers.

- Select the appropriate Controller Type in the Controller menu.
- Create a Default Register Assignment with the appropriate lower I/O board.
- Enter the ladder logic as shown in Figure 56.

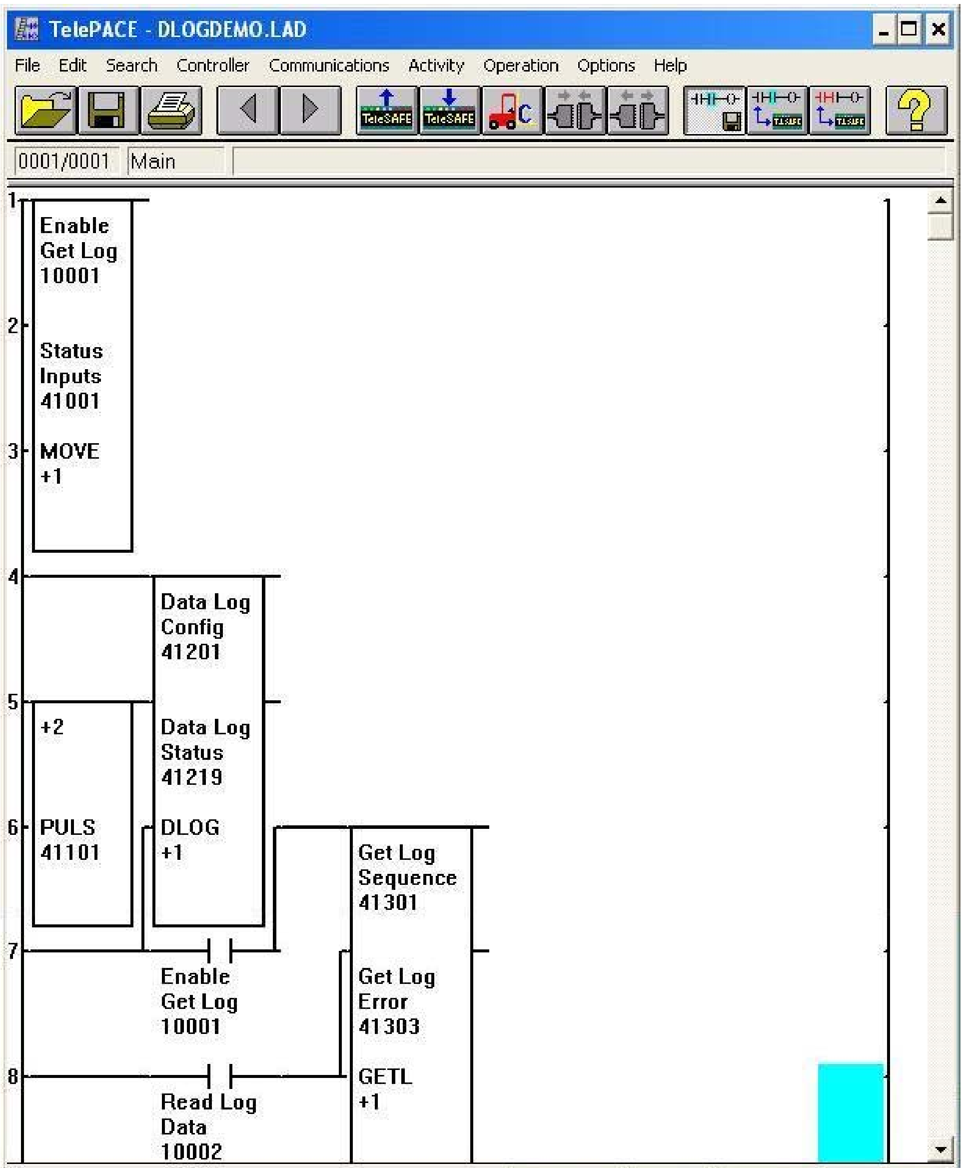


Figure 56: Ladder logic for DLOG demonstration

- Select the DLOG function by clicking on it once with the left mouse button.
- Right click, then select Element Configuration.
- Complete the DLOG function Element Configuration as shown in Figure 57.

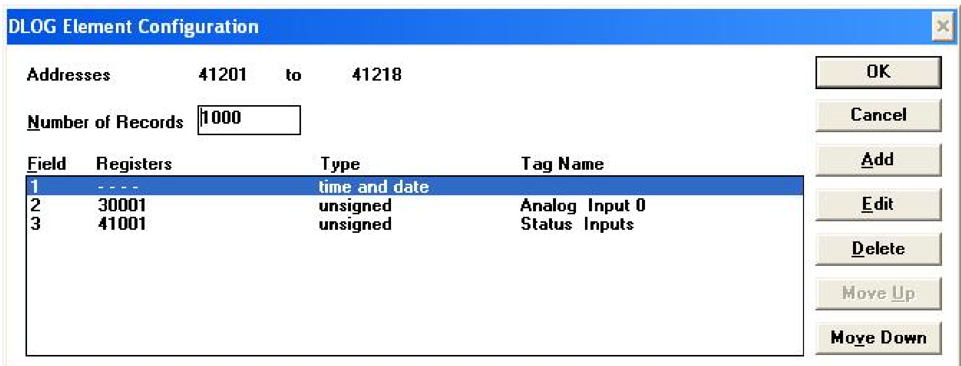


Figure 57: DLOG function configuration

- The DLOG will gather a total of 1000 records in a first in – first out buffer.
- Three fields will be gathered in each record. The first is Time & Date. This does not require modbus registers – just select the appropriate data type. Second is register 30001, which is Analog Input 0. Third is register 41001, which contains the first 16 digital inputs.

- Save the program as DLOGDEMO.LAD.
- Ensure that switch 1, the Enable GETL input, is ON to enable the GETL function.
- Download the program to the controller, then go online to monitor execution.

Step 3 Monitor Program Operation

With an analog input signal and digital inputs coming from the demo I/O board, the program will begin to gather data every two seconds. The student may adjust the analog input and toggle the 3rd and 4th switches to generate data. (switches 1 and 2 are used in the demo, and should only be toggled as required) Use the Register Editor to monitor and test the operation of the GETL function. Once this is done, SCADALog will be used to demonstrate automated reading of log contents.

- Once online with the controller, create a new Group in the Register Editor.
- Press Add, then add the following registers in order to test the GETL function:
  - 41303 – Unsigned 41301 – Unsigned Double
  - 41306 – Unsigned 41304 – Unsigned Double
  - 41311 – Unsigned 41307 – Unsigned Double
  - 41312 – Unsigned 41309 – Unsigned Double

- The DLOG has been gathering data every two seconds, starting with sequence # 0. Enter a desired sequence # to view into register 41301.
- With switch 1 on to enable the GETL function, toggle switch 2 (Read Data) on and off again.
- If a 23 error code is generated, the sequence # does not yet exist. Enter another value, then toggle the Enable input (switch 1) off and on again to clear the error.
- If no error is generated the log will return the field values for Time & Date, AIN 0 and DINs.
- Enter another valid sequence number, then toggle the Read Data input to retrieve data from another record. Note that the sequence number will need to be tracked in order to retrieve data in future.
- Once you are satisfied with the operation of the DLOG and GETL functions, close TelePACE.
- Open SCADALog, and start a new configuration by clicking on File | New.
- Set up comms to the controller by clicking on Communication | PC Communication Settings. The Modbus RTU protocol is used as with TelePACE.
- Read the controller's data log configuration by clicking on Data Log | Configuration, then clicking the Read button. This will read the setup of all logs in the controller.
- For each data field a descriptive title may be added. Click on the field, then click Edit Title.
- Click Ok. The three columns are listed in the order the fields were added in the DLOG.
- Click the Save button, and give the file a descriptive name. This saves the controller's log configuration and comm settings into an *.slc file.
- To read log data from the controller, click on Data Log | Read Logs. Select which logs are to be read - in this case log 1 is the only log, so leave it at All Logs. Also choose whether the log contents are to be purged (deleted) after being read. Then click Ok to start the read.
- Click Save after the read is done. This again saves the *.slc file, but also saves a *-01.csv file. The -01 part specifies which log it is. It is saved as a comma-separated variable file.
- If it is desired to manipulate the log contents afterwards, the contents must be exported to another csv file. Editing the *-01.csv file will corrupt it, causing SCADALog to not be able to open it afterwards.
- To perform an Export, open the appropriate log. (click on the log selection button at the top of the screen) Go to the File menu, then click Export. The Export file will be saved as

another csv file, but name it in such a way as to prevent confusion with the SCADALog log file.

Modbus Serial Master Communications

The MSTR function block is used to initiate a master message to exchange data with another device via the serial port of a SCADAPack controller. The message may use either the Modbus or DF1 protocols. Modbus is an open-source, non-proprietary protocol which is implemented by many equipment manufacturers in order to maximize inter-operability. DF1 is a protocol primarily used by Allen-Bradley and anyone wishing to communicate with their equipment. The following exercise will demonstrate only the Modbus protocol.

Each time a low to high transition occurs on the master's Enable input a message is sent. A timer starts at the same time. If a valid reply arrives before the timer's Time Out delay is reached, the Message Complete output goes true and the message succeeds. If no reply has arrived after the Time Out delay, or if the reply is corrupted, the Message Error output goes true.

Only one master message may be active on each serial port at any one time. The modbus protocol must receive a reply to each message before the next message is sent. A separate message may be sent on a different port at the same time, as it will be on a different network. If it is desired to send multiple messages simultaneously, the ethernet port of a SCADAPack 2 or 32 must be used, along with the MSIP function. As many as 20 MSIP messages may be active on an ethernet port at any time.

Modbus is a low-level protocol which does not include such refinements as Retries and Report By Exception. If these features are desired then additional ladder logic code must be written to implement them. An example of Retry logic may be shown after the following demonstration. It is possible to add this functionality and more by instead using the DNP3 protocol.

Within the Modbus protocol there are eight primary functions. The programmer may choose to send a Read or a Write message. It is possible to read any of the four Modbus data types: coil, input status, analog input or holding register. When writing, it is only possible to write to the output registers – the coil and holding register types. A message type sending either one or multiple registers may be chosen. It is also possible to send Enron Modbus messages, though this will not be covered.

The desired protocol, baud rate, station number and other parameters must be set for each port in the Controller | Serial Ports dialog before any given port is used for Modbus communications. This information will be downloaded to the controller along with the ladder logic.

Choosing the desired serial port (along with its previously set parameters) is done in the MSTR element configuration. The slave device's serial port modbus station number must also be specified, along with the data source and destination register addresses.

The MSTR function section of the TelePACE manual also discusses such issues as the maximum number of registers that can be read or written in a single message, and how to use the DF1 protocol in a MSTR message.

MSTR Exercise

Step 1 Create Tag Names

- Open a new file in TelePACE.
- Enter the following tag names using the Edit Tag Name dialog or import them using the supplied MSTRDEMO.CSV file.

Tag Name	***Register Address***
Complete	00101
Error	00102
Master Enable	10001
Analog Input 0	30001
Slave AIN 0	41001
Master Config	42010

Step 2 Build Ladder Logic

For this demonstration two student's SCADAPacks will be connected together to create a simple network. As only one controller may be the Master at any time, the Master Enable switch is used to select this. The PULS function in this program will turn on once every ten seconds. With the Master Enable switch on, the off – on transition generated by the PULS will trigger the MSTR function to send a message. If the message succeeds the upper output will go true very quickly. If it fails, the lower output will go true after two seconds.

- Select the appropriate Controller Type in the Controller menu.
- Create a Default Register Assignment with the appropriate lower I/O board.
- Enter the ladder logic as shown in Figure 58.

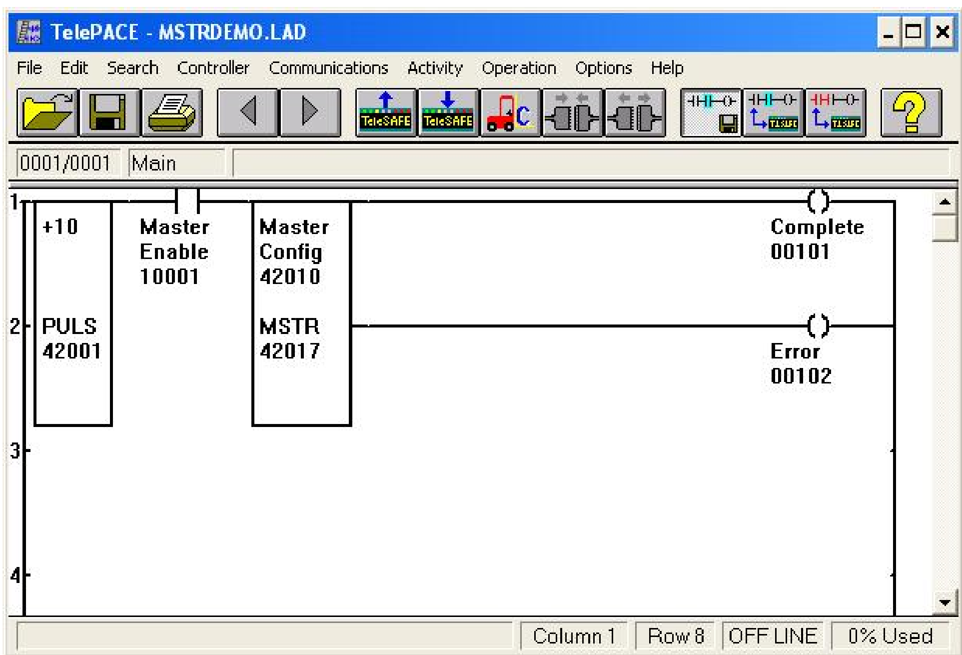


Figure 58: Ladder logic for MSTR demonstration

- Go to Controller | Serial Ports and ensure com1 is selected.
- Set the com1 parameters as shown in Figure 59.
- The Station number will be YOUR controller's station number. The slave controller's station number will be set later, in the MSTR element configuration. Both of these will be assigned by the instructor.

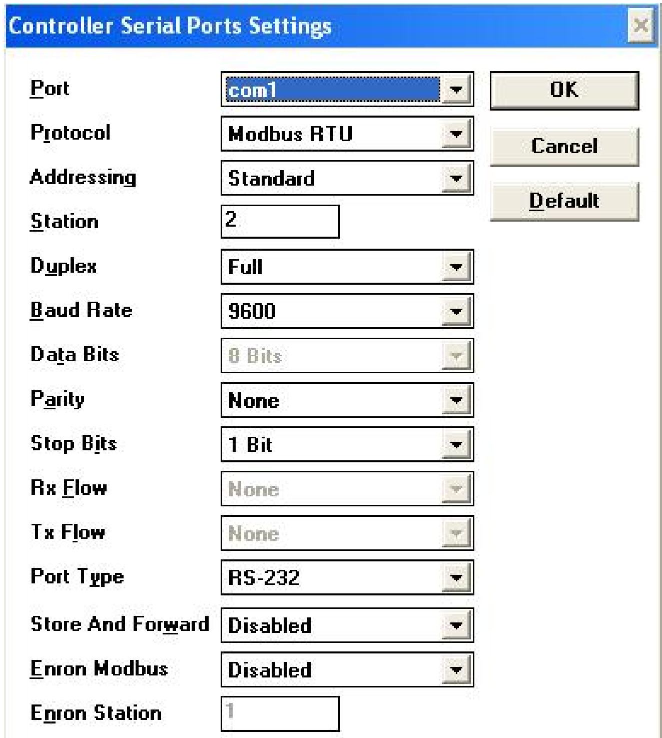


Figure 59: Controller Serial Port Settings

- Select the MSTR function by clicking on it once with the left mouse button.
- Right click, then select Element Configuration.
- Complete the MSTR function Element Configuration as shown in Figure 60.
- The Slave RTU Address will be the modbus station number of the other student's controller.

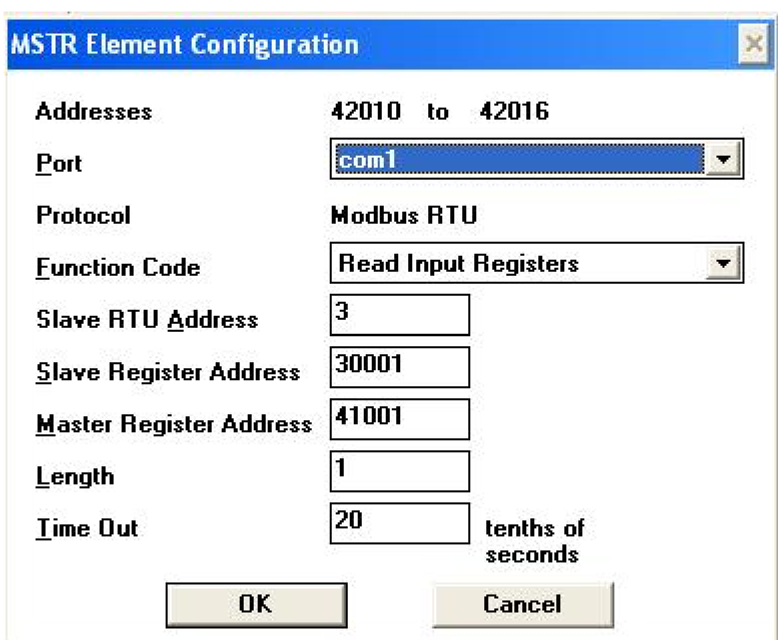


Figure 60: MSTR configuration to Read Input Registers

- Save the program as MSTRDEMO.LAD.
- Ensure that switch 1, the Master Enable input, is OFF to disable the MSTR function.
- Download the program to the controller, then go online to monitor execution.

Step 3 Monitor Program Operation

Each student will take a turn being the Master station. The other student in each pair will be the Slave station. Com port 1 of the two controllers will be connected with a null modem cable. In its simplest this could be a 3 wire cable, though a commercially built cable will also work. When the PULS output goes true on the Master station it will send a message to the Slave, either requesting an analog input value or setting the state of a digital output.

- Once online with the controller, create a new Group in the Register Editor.
- Press Add, then add the following registers in order to test the MSTR function:
    - 30001 – Unsigned 41001 – Unsigned
- The instructor will state who in each team is to start out as the Master station. That person should turn on their Master Enable switch. A master message will be sent the next time that controller's PULS function sets its output True. The message will be re-sent every 10 seconds until the Master Enable switch is turned off.
- The student who is Master should watch their Complete and Error outputs. If Complete goes true then the message succeeded. If Error goes true then some configuration problem exists.
- If the message has succeeded, then the Master controller's register 41001 will now display the same value as the slave controller's register 30001. (Analog Input 0)
- If the message fails, the Error output will go on two seconds after the MSTR was enabled. This is the value the Time Out parameter has been set to. If this occurs, check the Controller | Serial Port settings for com1 in both controllers. The station number must be 2 in one and 3 in the other. Then check the MSTR configuration in the controller which is currently the Master. It should have the station number of the other controller set in it. Also check the serial cable. It must be a null cable, connected to com1 on both controllers.
- The student at the Slave controller should adjust the value of Analog Input 0, and then the student at the Master should check to see if this new value is properly received.
- Once the first student's controller is working as Master, that person should turn Off their Master Enable switch and the other student should turn their switch On to reverse roles.
- Now ensure that the new Master station is receiving live data every 10 seconds into its register 41001 from the other station's register 30001.
- Once both students have been able to read data from the other controller, it's time to try to write data to the other controller.
- Each student should go to Edit Online mode.
- Select the MSTR function, right click and go to Element Configuration.

- Change the Function Code from Read Input Registers to Write Single Coil.
- Change the Slave Register Address from 30001 to 00006. (6th coil, tied to an LED)
- Change the Master Register Address from 41001 to 10004. This is the 4th switch.
- Click ok, which sends the changes to the controller. Save the changes.
- The 4th (right-most) switch on the controller whose Master Enable switch is on will be the data source. Whether that switch is on (1) or off (0) its state will be written to the Slave controller once every 10 seconds. Ensure the slave's 6th LED goes on within 10 seconds when the Master's 4th switch is turned on, and goes off when the switch is turned off.
- If this fails, perform the same troubleshooting as before.
- Once the Write message is working, turn the Master Enable switch Off for the first student, and turn it on for the second. Ensure that the new Master can also write to its slave's LED.
- Try increasing the pulse rate. It can go as fast as every 1 second. If this is done, remember that the Time Out period must be shortened as well. At a 1 second pulse, the On period is only 500ms. To make the Error output work the Time Out value must therefore be less than 500ms. To see the Error output go true the Update Rate in Register Editor must be faster.
- Also try having BOTH Master Enable switches on at the same time. This technically is a bad idea. But due to the very short duration of the MSTR messages any collisions will be quite infrequent. At lower data rates, or if the message is being triggered more often, or if the message is longer, collisions will become more likely