Reference Guide

for

TOPDOC® NexGen

&

SoftPLC Version 4.*x*

SoftPLC® CORPORATION

Because of the variety of uses of the information described in this user guide, the users of, and those responsible for applying this information must satisfy themselves as to the acceptability of each application and use of the information. In no event will SoftPLC Corporation be responsible or liable neither for its use, nor for any infringements of patents or other rights of third parties that may result from its use.

SOFTPLC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE.

SoftPLC Corporation reserves the right to change product specifications at any time without notice.

Printed in USA
Last Updated: JUNE 2006

SoftPLC Corporation
25603 Red Brangus Drive
Spicewood, Texas 78669 USA
Telephone: 512/264-8390 or 1-800-SoftPLC
FAX: 512/264-8399
http://www.softplc.com
Email: info@softplc.com

Table of Contents

Table of Figures

CHAPTER 1 - INTRODUCTION/OVERVIEW

NO FIGURES

CHAPTER 2 - TOPDOC INSTALLATION/CONFIGURATION

CHAPTER 3 - SOFTPLC CONFIGURATION

CHAPTER 4 - GATECRAFT FIREWALL SETUP

CHAPTER 5 - SOFTPLC OPERATION

CHAPTER 6 - MEMORY ORGANIZATION & DATA TABLE ADDRESSING

APPENDIX A - GATECRAFT LINUX BASIC SHELL COMMANDS

NO FIGURES

APPENDIX B - O·N·E TCP SETUP FOR THE MSG INSTRUCTIONS

Table of Tables

CHAPTER 1 - INTRODUCTION/OVERVIEW

CHAPTER 2 - TOPDOC INSTALLATION/CONFIGURATION

NO TABLES

CHAPTER 3 - SOFTPLC CONFIGURATION

CHAPTER 4 - GATECRAFT FIREWALL SETUP

NO TABLES

CHAPTER 5 - SOFTPLC OPERATION

NO TABLES

CHAPTER 6 - MEMORY ORGANIZATION & DATA TABLE ADDRESSING

APPENDIX A - GATECRAFT LINUX BASIC SHELL COMMANDS

NO TABLES

APPENDIX B - O·N·E TCP SETUP FOR THE MSG INSTRUCTIONS

CHAPTER 1 - INTRODUCTION/OVERVIEW

ABOUT THIS GUIDE

This Reference Guide is intended to help you quickly begin working with your SoftPLC and TOPDOC NexGen software. It describes software installation and configuration, setting up communications between computers, and provides a brief overview of the memory structure for SoftPLC.

SoftPLC based control systems can be configured to run on a wide variety of hardware platforms, and connect to an even wider variety of I/O devices. Therefore, detailed and/or specific information on even the most popularly used options is beyond the scope of this Guide. For complete system documentation, you may need to refer to additional SoftPLC Corp. manuals, computer hardware manuals, and the documentation provided by your I/O device manufacturers.

If you have purchased your SoftPLC system hardware from SoftPLC Corp. or one of our OEM Partners, your SoftPLC CPU will most likely already be installed and configured to a known default configuration for you. The hardware system documentation will describe the settings used in detail, and will also describe the procedure you will need to follow to make any changes.

Documentation updates as well as supplementary documentation can be found on SoftPLC's website at http://www.softplc.com. Our website's Knowledge Base also includes some add-on instructions and drivers for SoftPLC. In addition, you can find a list of our SoftPLC Partners, who provide hardware and/or software products that work with SoftPLC.

SOFTWARE COMPONENTS

APPLICATION SOFTWARE PRODUCTS

The **SoftPLC runtime** turns a computer into a controller. This software is what you use to run your control logic and drive I/O devices and it must run on a dedicated computer. If you purchased SoftPLC in hardware from SoftPLC Corp, the SoftPLC runtime software will already be installed and configured.

SoftPLC also includes an SSH and FTP Server, Java Virtual Machine (JVM), a number of utilities, I/O drivers, loadable instructions, and more. This Guide briefly describes the use of these auxiliary programs OR identifies where you can find more information on them.

The SoftPLC runtime runs on an embedded Linux operating system specially packaged for industrial control and called Gatecraft Linux. Most users do not need to be familiar with Linux to use SoftPLC. SoftPLC setup, programming and maintenance is done through TOPDOC NexGen.

As an option, the **SoftPLC Web Server** is available to allow SoftPLC to provide web server functionality. Use of the Web Server is beyond the scope of this Guide, and is detailed in other SoftPLC documentation.

TOPDOC NexGen is used for programming, program documentation, online maintenance, and troubleshooting of your SoftPLC controller. TOPDOC NexGen runs on any PC running a Win 32 or Linux operating system. With TOPDOC NexGen connected to your SoftPLC(s), you can perform functions such as online programming, data monitoring, I/O forcing, program upload/download, and more.

TOPDOC NexGen communicates to the SoftPLC via Ethernet.

PROGRAMMER'S TOOLKIT

SoftPLC Corp. provides a technology license for programmers who want to write C/C++ or Java language custom code to interface and/or work with SoftPLC. A Technology Transfer License must be executed prior to purchase of this product.

CHAPTER 2 -
TOPDOC NEXGEN INSTALLATION/CONFIGURATION

TOPDOC NEXGEN OVERVIEW

TOPDOC NexGen is a powerful ladder logic programming, documentation, troubleshooting, and SoftPLC configuration package. It is the only way to maintain and troubleshoot your SoftPLC based system.

TOPDOC NexGen (TDNG) provides:

- offline application development
- online maintenance/troubleshooting
- application documentation (e.g.: address labels and rung comments)
- hard-copy reports
- import/export utilities
- SoftPLC configuration

TOPDOC NexGen can run on any Windows 95/98/NT/2000/XP or Linux PC that meets the following *minimum* hardware requirements:

- Pentium 233 CPU
- 128 Mbytes of RAM (256 MB recommended)
- 800 x 600 screen resolution (1024 x 768 or better strongly recommended)
- Ethernet interface (configured for TCP/IP)
- Serial port (recommended) or USB-serial converter

You need to have a software protection key connected to any USB port on your computer to run TOPDOC NexGen.

The software protection key for your TOPDOC NexGen package is very valuable. You should treat it carefully. Lost key replacements cost the full package price!

INSTALLING TOPDOC NEXGEN AND OTHER DEVELOPMENT TOOLS

From the SoftPLC Product CD, select **SoftPLC Development Software and Tools**.

1. **Install Java Runtime Environment.** TOPDOC NexGen requires that you first install a Java Runtime Environment (JRE.) The JRE is required because TOPDOC NexGen is written in the Java language. Your JRE must be of the same version as that on the SoftPLC CD (or later.) If you do not have a JRE or you have an older version installed, first select **Install JRE**. Follow the installer prompts.

2. **Install TOPDOC NexGen.** Then select to install TOPDOC NexGen, which will be installed in a \softplc folder on the drive you specify. Follow the prompts on the screen to perform the installation.

3. **Install Auxiliary Tools.** If you do not already have favorite FTP and SSH client programs, you should install Filezilla and PuTTY from the SoftPLC CD, as you may need these tools to maintain your SoftPLC system(s.)

USING TOPDOC NEXGEN

When you install TOPDOC NexGen, an icon will be placed on your desktop that can be used to load TOPDOC NexGen (TDNG) at the main menu. Additionally, a SoftPLC folder will be created in your Programs folder that will include the TOPDOC NexGen icon, a PLC Configurator icon, and a Convert TDZ Apps icon. New users will not need the latter two icons.

Load TOPDOC NexGen using the TOPDOC NexGen icon. For information about how to use TOPDOC NexGen, refer to the help system within the program.

Table 1-1
TOPDOC NexGen Folders (Directories)

DIRECTORY	DESCRIPTION
\SOFTPLC	Home folder of the installation
\SOFTPLC\PLC	Specific local PLC defs, DEFAULT.LST file and any driver definition files (*.DEF)
\SOFTPLC\PLC\<name>	Local definitions and backup configurations for <name> (1 dedicated folder per PLC)
\SOFTPLC\BIN	TOPDOC NexGen executables, help files, etc.
\SOFTPLC\APP	Folder to store APP folders for <name> (1 dedicated folder per APP)
\SOFTPLC\APP\<name>	Specific APPs (ladder programs with documentation)
\SOFTPLC\CLIP	Program Clips (standard logic files)
\SOFTPLC\TLM	TOPDOC Loadable Modules
\SOFTPLC\USR	User settings and configurations

ETHERNET CONNECTIONS

TOPDOC NexGen communicates to SoftPLC on ethernet. You can also connect additional PC's, HMI's, or other devices.

For point-to-point Ethernet between only 2 systems, you need to use an Ethernet cross cable (Figure 2-1.) One 6-foot cable of this type is included with TOPDOC NexGen. Additional cables can be purchased from SoftPLC Corp. as Catalog Number ICO-CEC. Also, these type cables can be found at most computer stores.

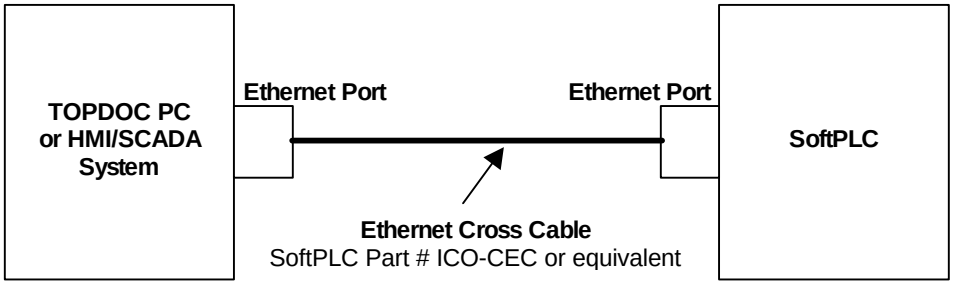

Figure 2-1
Point-to-Point Ethernet Connections Example

For communications through a network switch/hub, use standard ethernet patch cables (See Figure 2-2.)

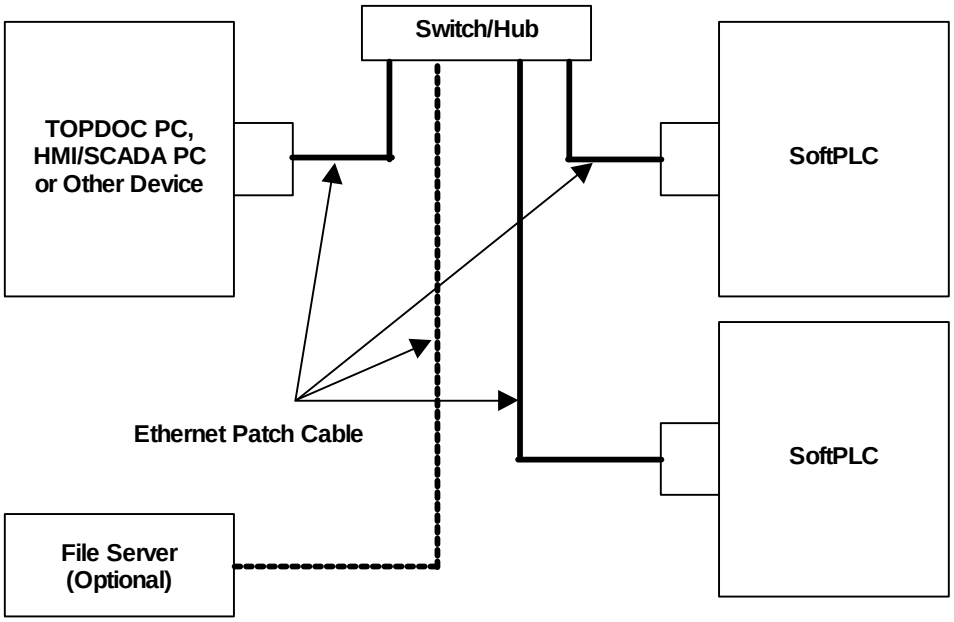

Figure 2-2
Multi-Node Ethernet Connections Example

TCP/IP COMMUNICATIONS SETUP (TOPDOC PC)

This section describes the process for installing and setting up TCP/IP communications on your TOPDOC PC.

TOPDOC NexGen is used to set up the networking and TCP/IP communications on the SoftPLC system. This is described in Chapter 3. To configure a new SoftPLC, you will need to set up the TCP/IP on your TOPDOC PC first.

Once you have configured both the SoftPLC and the TOPDOC PC, you can confirm that the TCP/IP channel is working using a utility that comes with Windows called PING.EXE (normally located in the WINDOWS directory of your Windows system.) Using the PING utility is described later in this section.

TCP/IP SETUP FOR WINDOWS 95/98/NT/2000/XP

If your TOPDOC PC is used on your company network, your TCP/IP is probably already configured as required by your company. Contact your network administrator before making any changes to your network settings, or if you have questions about the IP address and subnet mask you can use for your SoftPLC applications.

Each Windows operating system has a slightly different method for configuring TCP/IP. Refer to your Windows help files for instructions on your particular system.

CONFIGURING THE MICROSOFT TCP/IP STACK (EXAMPLE)

1. Select **Start** → **Settings** → **Control Panel** →**Network** →**Local Area Connection**
2. Select **TCP/IP,** then click on **Properties**.
3. Select the **Specify an IP address:** radio button. Enter an IP address (e.g. 192.100.100.44.) All network IDs on a network must be unique.
4. Enter a subnet mask (e.g. 255.255.255.0.)
5. If your network does not have a DNS server, disable DNS.
6. Select **OK**

TESTING TCP/IP COMMUNICATIONS

Before you try to connect from TOPDOC to SoftPLC:

1. Set up Windows TCP/IP
2. Confirm the Ethernet port on the SoftPLC is correctly configured (refer to the section on Network Editor in Chapter 3.)
3. Confirm the Ethernet cabling is correct (Figures 2-1 and 2-2.) With either method of Ethernet cabling, when both the SoftPLC and Windows computer are powered on there should be a solid ON Link LED near the Ethernet port confirming a correct cable configuration.
4. Test the Ethernet Configuration by using the Windows ping.exe utility. From the **Start** → **Run** menu or through a DOS command prompt window, type **ping** and then the SoftPLC's TCP/IP address. For example, **ping 192.100.100.101**. Ping.exe will attempt to communicate with the SoftPLC 4 times.
5. If ping.exe reports a timeout problem, TOPDOC will NOT be able to communicate. If this happens, consult your network administrator for specific help before continuing.

CHAPTER 3 -
SOFTPLC CONFIGURATION

INSTALLING SOFTPLC

If you have purchased a SoftPLC packaged system, the runtime software will already be installed in the hardware for you.

If you are using your own hardware, the SoftPLC runtime installation procedure will vary based on your hardware, and is beyond the scope of this Guide.

SoftPLC requires a copy protection key on a parallel or USB port in order to run. If you are setting up your own hardware, we recommend the key be installed INSIDE the enclosure to protect against theft or damage.

The Runtime software on the SoftPLC Installation CD is for version 3.x and is provided for customers using older software only.

Table 3-1
SoftPLC Version 4.x Default Directories

DIRECTORY	DESCRIPTION
/SoftPLC	SoftPLC directory
/SoftPLC/app	Directory for SoftPLC .app files
/SoftPLC/javalib	Modlet and Driverlet shared object files
/SoftPLC/run	SoftPLC binaries, O·N·E drivers/configuration, SoftPLC configuration files, runsplc shell script
/SoftPLC/tlm	I/O drivers and associated configuration files
/SoftPLC/web	Web server directory
/SoftPLC/web/htm/usr	Web pages
/SoftPLC/web/jar	Java servlets
/SoftPLC/web/lst	Web server configuration directory

CONFIGURING THE SOFTPLC CPU

TOPDOC NexGen is used to configure a SoftPLC CPU, and is described in this Chapter. There are 5 parts to complete a standard setup:

1. With the DEFINE EDITOR you configure TOPDOC NexGen to communicate with SoftPLC over TCP/IP

2. With the NETWORK EDITOR you configure SoftPLC's TCP/IP

3. With the O•N•E EDITOR you configure SoftPLC's Communications Channels to operator interfaces, A-B PLC's, etc.

4. With the MODULE EDITOR you configure SoftPLC's I/O and other Device Drivers and/or Loadable Instructions

5. With the STARTUP CONFIGURATION EDITOR you configure SoftPLC's Operation and Startup Functions

LOADING TOPDOC NEXGEN'S PLC CONFIGURATOR

Load TOPDOC NexGen using the icon on the desktop or in the SoftPLC folder, and then after logging in, select **PLC**. Figure 3-1 shows the initial TOPDOC NexGen display when you select PLC from the main TOPDOC NexGen menu.

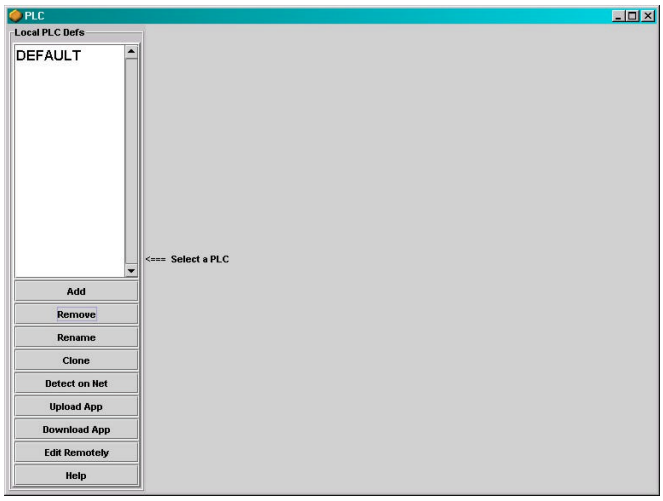

Figure 3-1
TOPDOC NexGen Configurator Opening Display

PLC CONFIGURATOR USER INTERFACE

The TOPDOC NexGen PLC Configuration Editor utilizes a simple interface that makes it easy for you to create and maintain the definition information for all your SoftPLC's. Figure 3-1 shows the initial opening display of the program. The applicable options in this window are:

- **Add** new PLCs
- **Remove** existing PLCs
- **Rename** PLCs
- **Clone** PLC definitions to create a new, similar PLC
- **Detect on Net** option will interrogate the TCP/IP network to find all the SoftPLC's on it.

Once you select a PLC to configure, the PLC Configuration Editors panel is displayed (Figure 3-2.) Across the top right side of the screen are buttons corresponding to the various configuration editors used to manage your SoftPLC configurations. Each of these is described in this Chapter.

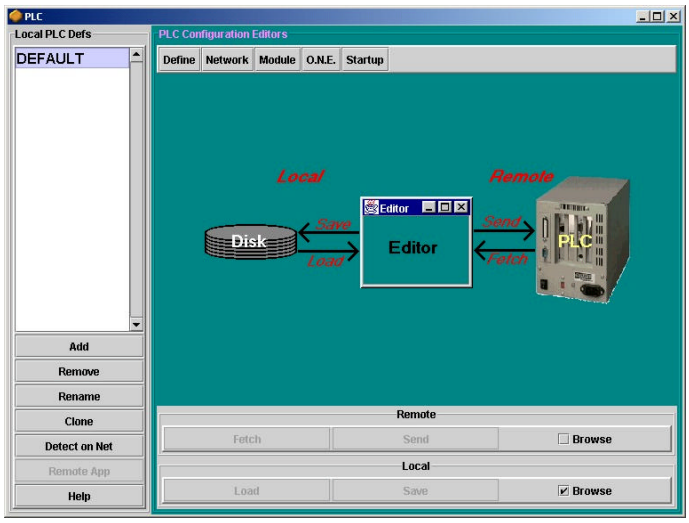

Figure 3-2
TOPDOC NexGen Interface Overview

The primary key to understanding the use of TOPDOC NexGen's PLC Configuration Editors is to first understand its use of the terms **local** and **remote**, and to know which definition you are editing at any point in time.

Refer to Figure 3-2. Before you load a configuration editor, decide which data you want to edit using the buttons in the bottom right window.

LOCAL CONFIGURATIONS

Local refers to data residing on your TOPDOC computer.

On your TOPDOC PC, you **Save** all of the configuration files for each PLC to disk, each PLC in its own dedicated folder. For example, the local configuration files for a PLC named **MYPLC** would be stored in the **\SOFTPLC\PLC\MYPLC** folder on the PC where you have TOPDOC NexGen installed and running.

To edit a configuration stored on disk, you **Load** it.

REMOTE CONFIGURATIONS

Remote refers to data residing in a SoftPLC, and is accessible from TOPDOC NexGen only through a TCP/IP link.

Send is used to transfer configuration information from TOPDOC NexGen to a SoftPLC. TOPDOC NexGen manages all the details of the configuration - where and how the information needs to be placed in the SoftPLC.

After doing a **Send**, it is always necessary to restart the SoftPLC for changes to take effect.

To retrieve existing configuration information from a SoftPLC into the TOPDOC PC for editing and/or to save it to the local disk, you **Fetch** it.

BROWSE CHECKBOXES

The **Browse** checkbox determines whether the data you are editing is going to be **Remote** or **Local**:

- **Local Browse** means that the data will automatically be Loaded from the TOPDOC PC disk into the respective editor when transitioning between either editors or PLC's. (This is the default mode.)

- **Remote Browse** means that the data will automatically be Fetched from the SoftPLC into the respective editor when transitioning between either editors or PLC's.

Editor data is only on-screen data. It may be different than what is on the local disk or in the remote PLC at any point in time. Displayed editor data becomes local or remote data only upon a **Save** or **Send** command or after a **Load** or **Fetch** command.

If neither Browse mode is enabled, no data is automatically provided for an editor. This is called **Non-Browse Mode**. In Non-Browse Mode, as you transition between PLC's, the data from the previous PLC's editor view is displayed in the new PLC's editor view. You can always select data to edit by pressing either the **Fetch** or **Load** button.

If you are not careful, using Non-Browse Mode could be potentially harmful in either your misinterpreting the data or in changing configurations by mistake.

Non-Browse Mode is seldom used, but can be helpful if you want to duplicate *part* of a configuration from one PLC to another. (Use **Clone** if you want to duplicate an entire configuration.)

DEFINE EDITOR

The TOPDOC NexGen **Define Editor** is only used locally. It is used to describe your SoftPLC, and most importantly to allow TOPDOC NexGen to communicate with the SoftPLC over TCP/IP.

When you Add a PLC, you must first identify in the Define Editor whether it is to be a version 2.*x*, 3.*x*, or 4.*x* system before attempting any other configuration.

The Define Editor also will identify to TOPDOC NexGen the software version of your SoftPLC (e.g.: Version 2.*x*, 3.*x*, or 4.*x*.) which in turn determines which folders and formats TOPDOC NexGen will use in the SoftPLC for the configuration files.

To load the editor, select a PLC from the leftmost listbox, then press **Define**.

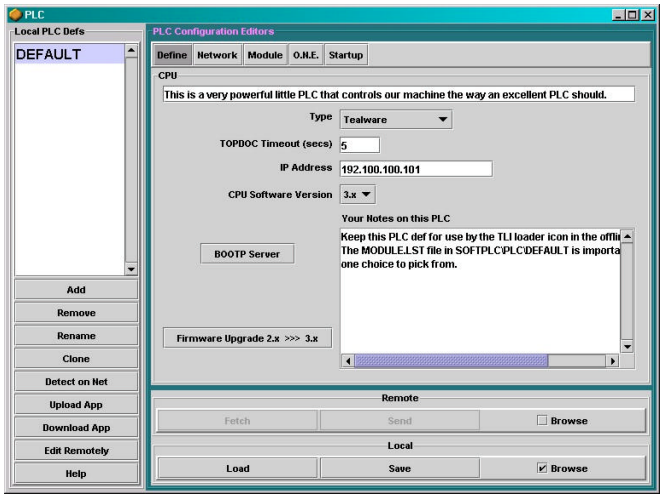

Figure 3-3
TOPDOC NexGen Define Editor Display

DEFINE EDITOR FIELDS

This information is stored in a text file called **DEFINE.LST** in the softplc\plc\<plcname> folder on your TOPDOC PC.

- **Single Line Description** is at the very top of the panel and is a place to describe your PLC with a brief summarizing comment.

- **Type** is a combobox giving you a choice of Tealware, SoftPLC Processor, or Other (for customer provided hardware.)

- **TOPDOC Timeout** specifies the amount of time in seconds to wait for a TCP/IP connection with a SoftPLC, and is also the time to wait for a reply to any communications request before reporting a communications problem.

- **IP Address** must be set to a valid IP Address or DNS registered name of the SoftPLC on your TCP/IP network.

- **CPU Software Version** must be set to the proper version for your runtime system software.

- **Your Notes on this PLC** gives you a free format text area where you can put any text you would like: serial numbers, maintenance notes, etc.

BOOTP SERVER

As a factory default, SoftPLC's are shipped to start up using BOOTP protocol as a means of obtaining their IP Address the first time.

Prerequisites to using BOOTP Server

- TCP/IP installed on your TOPDOC computer
- Valid ethernet cable between the TOPDOC and SoftPLC systems (refer to Chapter 2 for cabling information)
- SoftPLC configured to boot using BOOTP client protocol. A new SoftPLC (from the factory) should automatically boot using BOOTP client protocol.

You must shut down any other BOOTP Servers running on your network, as they will interfere with the TOPDOC BOOTP Server.

If this is NOT a new SoftPLC, you will need to manually change the existing IP address of the SoftPLC.

- If the SoftPLC's IP address is on the same subnet as your PC, you will use TOPDOC's Network Editor to change it as described in this chapter.
- If the SoftPLC's IP address is NOT on the same subnet as your PC:
  - You will need to establish a Serial Connection (see Chapter 4 **Serial Connection**.)
  - Use the **setupcatecraft** NETWORK.LST option (see Chapter 4 **Network.LST**.)

BOOTP Step by Step Instructions

For first time configuration of new SoftPLCs you should first click on the **BOOTP Server Button** in the **Define Editor** panel to establish a first time IP Address and Subnetmask within the SoftPLC. You can also use BOOTP, if later, you want to change the IP Address or Subnetmask, although the **NETWORK EDITOR** will normally be used.

The information displayed on the dialog box that opens after clicking on BOOTP Server from within the Define Editor, applies to version 2.*x* and 3.*x* only. For version 4.x, refer to the instructions in this chapter.

From the Define Editor screen, click on **BOOTP SERVER** button. It will open up the BOOTP Server Configuration screen (Figure 3-4.)

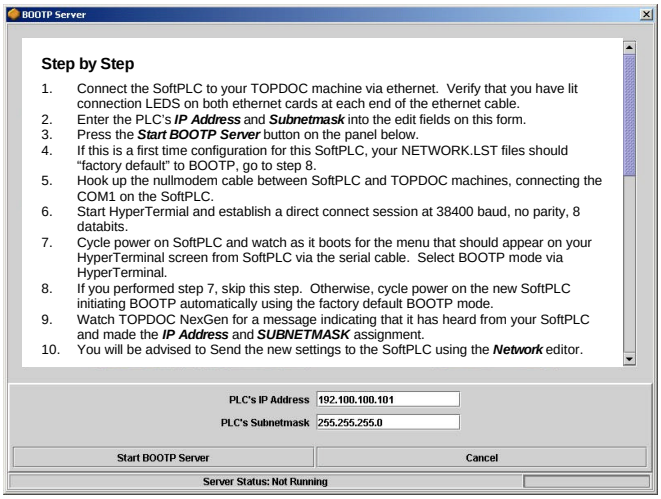

Figure 3-4
BOOTP Server Configuration Screen Example

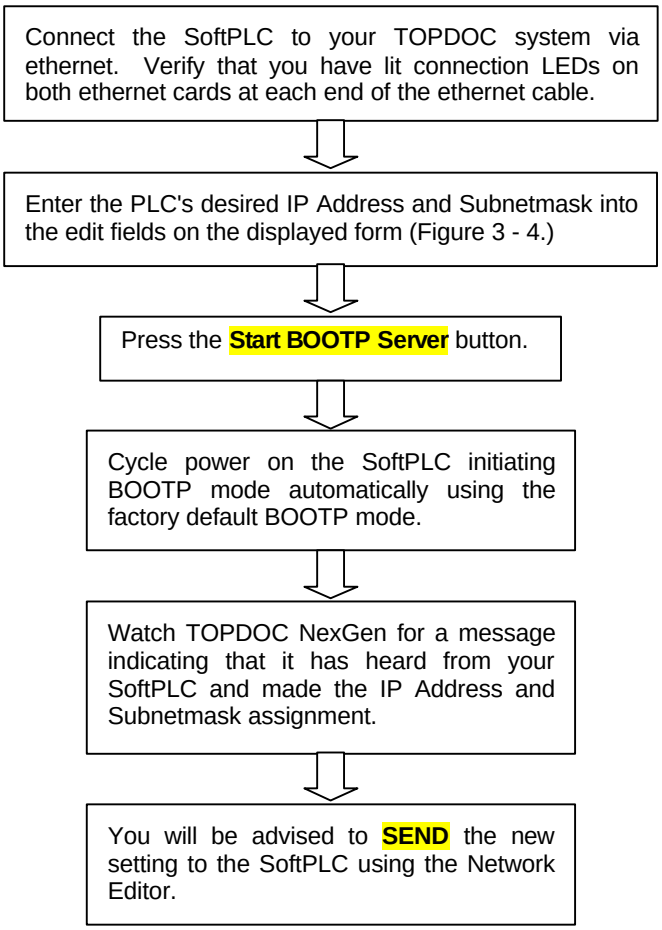

Figure 3-5
BOOTP-Step by Step Instructions

NETWORK EDITOR - TCP/IP CONFIGURATION

You use the Network Configuration Editor panel in TOPDOC NexGen to configure a SoftPLC's TCP/IP support. This configuration data is kept in a file named NETWORK.LST. The remote copy on the SoftPLC is in /SoftPLC/run/NETWORK.LST. The local copy is in \SoftPLC\plc\<PLCNAME>\NETWORK.LST. The Network Editor edits these files. Figure 3-6 is a sample Network editor panel.

The Network Configuration Editor panel, as shown in Figure 3-6, consists of the following sections:

- Interfaces
- FTP
- DNS
- ROUTING

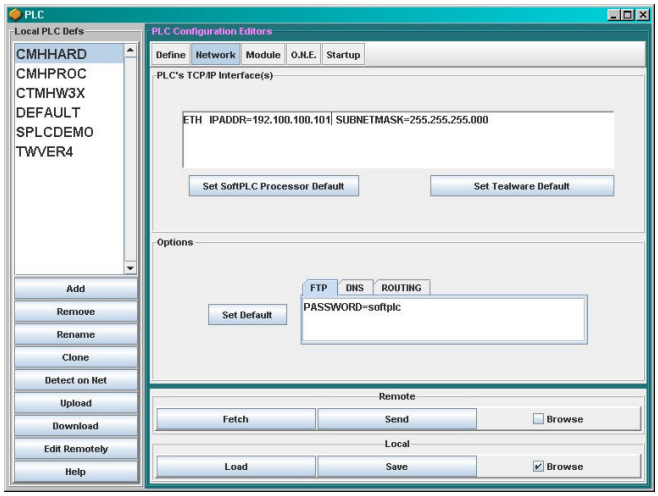

Figure 3-6
Network Configuration Editor Panel

INTERFACES

SoftPLC supports up to four external TCP/IP ethernet and one internal loopback interfaces. Normally only one external TCP/IP interface is configured as explained here.

If you are using a SoftPLC Processor or Tealware CPU, you should either click on **Set Tealware Default** or on **Set SoftPLC Processor Default**. Change the values described below within the text string displayed:

- For either type of card you should set the IPADDR and SUBNETMASK values. If this is a new SoftPLC, you should have obtained the IPADDR using the BOOTP procedure described in the previous section.

On any TCP/IP subnet, the first three parts of the address (NetID) must match in order for one node to communicate with another node. The last part of the address (HostID) must be unique for each node.

If you need to use your TOPDOC or SoftPLC computer on an existing TCP/IP network, consult your network administrator for the addresses you should use.

FTP

This section establishes the FTP password, if any, which can be used for security with SoftPLC's embedded FTP Server (described in Chapter 5.) The FTP password, if entered, is required to login into the FTP server. It may not contain any spaces. If not present, then no password is required to login and the FTP server will be available to anyone. The default password is softplc.

DNS

This section establishes the IPADDR of your DNS (Domain Name Server), if any. Do not set a DNS server unless you are using DNS services from Java to initiate connections to other TCP/IP nodes from this PLC. Setting a DNS server when you actually don't have one can slow down some TCP/IP communications badly. One example of when you would set a DNS server is if you were using the Java based SENDMAIL Modlet to send email out onto the Internet.

If provided, the Domain Name Server must have an IP address that uses a network number common with any one of the provided interface definitions; i.e.: the name server must be accessible via one of the external interfaces. SoftPLC's TCP/IP support is not capable of being a name server - normally a Unix, Netware, or NT machine will play that role.

ROUTING

This optional section establishes the default TCP/IP gateway that is used to access other TCP/IP nodes that are not found on the same subnet as this SoftPLC. The GATEWAY parameter is used to provide an IP Address, which should be supplied only if you have a router between this SoftPLC and some other TCP/IP node that you want to talk to from this SoftPLC and you are initiating those communications from this SoftPLC.

TLI'S FOR SOFTPLC'S TCP/IP SUPPORT

There are also 2 ladder logic instructions, NETSTATS, and NETZERO, in SoftPLC's TCP/IP support that are implemented as TLI's. They manage some diagnostic counters that are specific to the ethernet interfaces.

NETSTATS can be used to read a block of ethernet statistics from one of the external interfaces and used to write diagnostics counters to a file of data table words when energized. Table 3-3 lists Diagnostic Counter Definitions by Word Number.

NETZERO will reset the counters to zero, except for lowest_free packets, which gets set to a high number temporarily.

Table 3-2
Diagnostic Counter Definition by Word Number

CTR	NAME	DESCRIPTION
0	lowest_free_buffers	Low water mark for available ethernet packet buffers. This number should reach a minimum value after several minutes of high traffic and stop. If it continues to decline over several hours or days, this is an indication of a memory leak.
1	errors_out	Number of errors while transmitting packets.
2	packets_lost	Number of packets that have been dropped upon reception.
3	errors_in	Number of errors while receiving packets.
4	no_memory	Number of times the TCP/IP stack was unable to obtain a packet buffer.
5	rx_overwrite_errors	Number of receive overwrite errors.
6	rx_frame_errors	Number of times the ethernet chip rejected a packet due to improper framing, such as too many bytes, or invalid CRC.
7	rx_crc_errors	Number of times the ethernet chip detected a CRC error on packet reception.
8	rx_fifo_errors	Number of receive fifo errors.
9	tx_delayed	Number of times where the chip wanted to transmit a packet, but had to wait for a small bit of time due to existing wire traffic.
10	tx_carrier_errors	Number of transmit carrier errors.
11	one_collision	Transmit process started, but had to back off one time due to a collision on the wire.
12	multiple_collisions	Transmit process started, but had to back off more than once for a given packet, due to a collision on the wire.
13	buffers_in_use	Count of packet buffers currently holding live communications data. This count of buffers has been removed from the free list.
14	packets_out	Count of packets/frames that have been sent out on the wire.
15	packets_in	Count of packets/frames that have been received from the wire.
16	current_free_buffers	Current count of free packet buffers. lowest_free_buffers, counter 0, is the low water mark of this value. If the sum of buffers_in_use plus current_free_buffers changes significantly over time, this is an indication of a memory leak.
17	isr	Count of total interrupt service routine dispatches that have occurred due to ethernet card interrupts on this interface. This value will change rapidly on a healthy communicating interface.

O•N•E CONFIGURATION EDITOR (COMMUNICATIONS)

O•N•E is an acronym for SoftPLC's communications software **O**ptimum **N**etwork **E**xecutive. By defining O•N•E channels, you are establishing peer to peer messaging paths that enable a SoftPLC to talk with other PLC's and HMI devices. Common examples are the ladder MSG instruction, SoftPLC's, A-B PLC's, or SoftPLC communications to an HMI.
SoftPLC's O•N•E supports 4 different types of peer-to-peer media types:

- DataHighway (DH)
- DataHighway Plus (DH+)
- A-B TCP/IP ethernet
- RS-232/422/485 (using A-B DF1 full/half duplex protocol.)

The protocols on these media types appear to other devices and/or applications identical to an Allen-Bradley PLC-5's, making SoftPLC extremely compatible with a wide variety of products from many different vendors. Table 3-4 shows O•N•E channel types and the media types that they support. Figures 3-7(a) thru 3-6(e) describe configuration of each Type.

Table 3-3
O•N•E Channel Media Types for Version 4.x SoftPLC

O•N•E CHANNEL TYPE	MEDIA TYPE(S)
Serial DF1	RS-232, 422, 485
A-B KTx	DH+
Client	A-B TCP/IP ethernet
Server	A-B TCP/IP ethernet

You use TOPDOC NexGen's O•N•E Editor (Figures 3-7(a)) to configure SoftPLC's peer-to-peer communications. Up to 32 O•N•E channels may be used simultaneously.

In the O•N•E Editor, there are buttons to **Add** and **Remove** O•N•E Channels. When you **Add** a new Channel, you first select a Type, and then you customize the parameters for the Channel (defaults shown in Figures 3-7(a).)

You can define more channels than you actually use. The **Channels Used** setting determines how many channels SoftPLC will actually load. This can be less than the number defined. In order for a defined channel to be eligible for use, it must be part of a **contiguous** block of channels numbered from 0.

The entry order of O•N•E Channels is important as they are loaded in the order listed and only the number of **Channels Used** (from 0 up) as you specify are active.

If any of the Used O•N•E channels are not configured properly or there is an error when SoftPLC tries to open the channel (ie: disconnected cable), then SoftPLC will not start up.

SERVER TYPE CHANNEL

There are no parameters for the Server type channel.

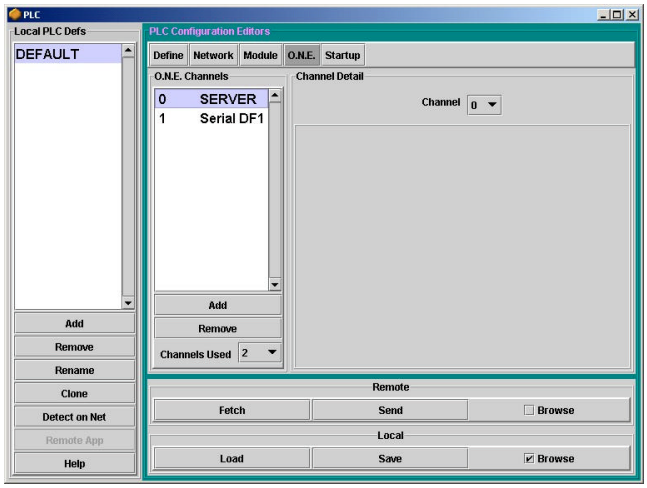

Figure 3-7 (a)
O•N•E Editor – Server Channel Panel

You always need at least one O•N•E channel defined as type Server in order to support communications with TOPDOC NexGen, SoftPLC's FTP Server, and HMI/SCADA products. This is set by default when you Add a new PLC.

SERIAL DF1 TYPE CHANNELS

Serial O•N•E Channels are normally used when SoftPLC is to connect via the DF1 full duplex or half-duplex slave protocols to an operator interface, SCADA package, DCS, or a wide variety of devices that support the DF1 protocol for peer-to-peer communications. Normally a null modem cable is used between the SoftPLC serial port and the other computer/device. Cabling requirements can be found later in this section.

If you are using serial ports other than COM1 and COM2 on your SoftPLC system, you will need to ensure the hardware you select is compatible with SoftPLC's extended COM support. SoftPLC Corp. sells compatible 2 port and 8 port serial add-on cards.

For the O•N•E Serial DF1 driver you can configure the following parameters:

- **Port** is the number of the SoftPLC serial port, COM1, and up. The O•N•E Serial DF1 driver supports up to 16 COM ports, for numbers COM1 to COM16

SoftPLC systems are factory set to use COM1 as a CONSOLE for troubleshooting purposes. We do not recommend you remove the CONSOLE function, without carefully considering the consequences of losing this functionality. Therefore, your first choice for serial O•N•E communications should be COM2, which is the default.

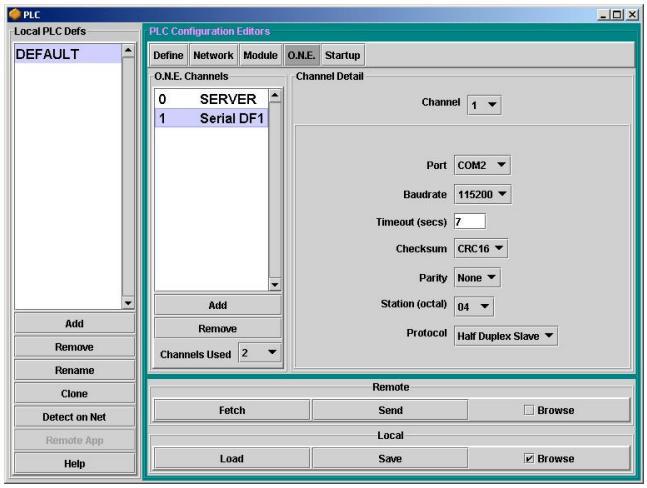

Figure 3-7 (b)
O•N•E Editor – Serial DF1 Type Channel

- **Baudrate** is the speed of the serial communications link, from 300 to 115200 bits/second. This must match the setting of the device at the other end of the link.

- **Timeout** is the length of time in seconds SoftPLC should wait before issuing a communications timeout error.

- **Checksum** is the type of error checking method to be used. We recommend CRC16. This must match the setting of the device at the other end of the link.

- **Parity** is normally set to None. This must match the setting of the device at the other end of the link.

- **Station** is the octal station number of the SoftPLC on the network. This value is only important if you are connecting via DF1 protocol to devices on a Data Highway network, where each station number must be defined and be unique. For point-to-point communications, this value is not important.

- **Protocol** allows you to select Full Duplex or Half-Duplex Slave protocol. This must match the setting of the device at the other end of the link.

SERIAL CONNECTIONS

When communicating to a SoftPLC from another computer or HMI, through one of its serial (COM) ports, you will need to use a standard null modem cable, as shown in Figure 3-8.

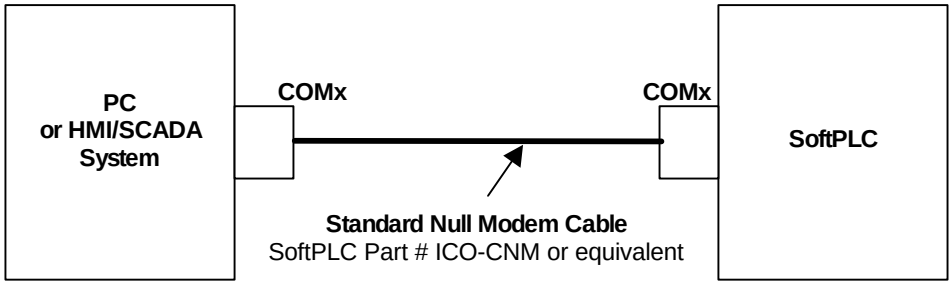

Figure 3-8
Serial Connection

The cable pinout for 9-pin serial ports is shown in the next figure (Figure 3-9.) One 6-foot cable of this type is included with TOPDOC. Additional cables can be purchased from SoftPLC Corp. as Catalog Number ICO-CNM. Also, these cables or 25 pin versions of them (if required) can be found at most computer stores.

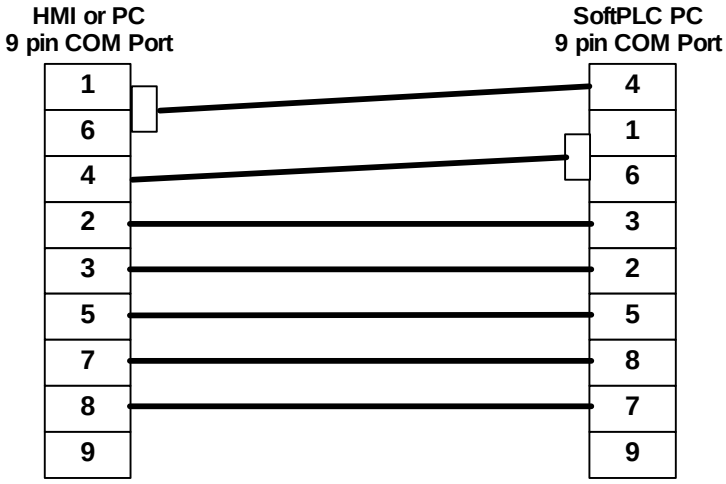

Figure 3-9
Standard Serial Null Modem Cable

CLIENT TYPE CHANNELS

O•N•E Client channels are used when the SoftPLC is to communicate to another SoftPLC, an Allen-Bradley PLC on Ethernet, or another device that supports the MSG instruction or other peer-to-peer functions. One Client channel is required for each unique TCP/IP address to which SoftPLC will be communicating. More detail on setting up Client Channels for use with the MSG instruction can be found in the TOPDOC Help File, MSG Instruction information, and Appendix C, **O·N·E TCP setup for the MSG Instruction**.

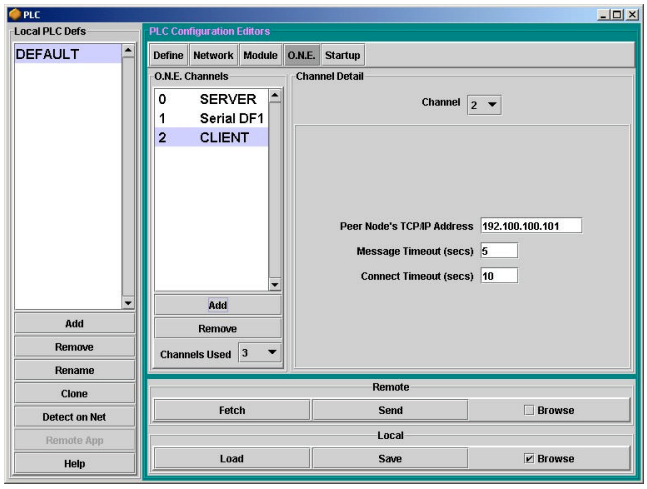

Figure 3-7 (c)
O•N•E Client Type Editor Panel

Figure 3-7(c) shows the panel for Client Channel configuration. The parameters to be configured are:

- **Peer Node's TCP/IP Address**, which is the TCP/IP address or host name of the other station.

- **Message Timeout** is the number of seconds from the time the MSG is initiated before setting the MSG instruction error bit.

- **Connect Timeout** is the number of seconds to wait before reporting a TCP/IP connection error.

KT*X* (DH+) TYPE CHANNELS

KT*x* Type Channels are used for Data Highway Plus communications using Allen-Bradley 1784-KT, -KTX, -KTXD, -PKTX, or -PKTXD cards in the SoftPLC. Up to 8 cards per SoftPLC are supported. Refer to the hardware documentation on setting up and installing the interface card(s.)

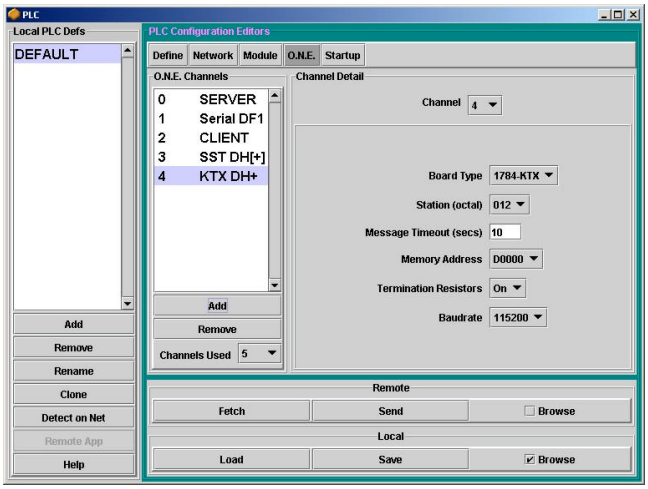

Figure 3-7 (d)
KTX (DH+) Editor Panel

Figure 3-7(d) shows the configuration panel. The parameters to be configured are:

- **Board Type** is the old style 1784-KT or new style 1784-PKTX, -PKTXD, -KTX, and -KTXD.
- **Station** is the octal DH+ station number from 0 to 077 of this SoftPLC, for this interface card. Each DH+ station number must be unique for a given Data Highway network.
- **Message Timeout** is the number of seconds to wait before setting the error bit in the MSG instruction.
- **Memory Address** is the hexadecimal location in memory for the card to be loaded. Your SoftPLC hardware will determine which of the values in the list box are allowed for your system.
- **Termination Resistors** is a selection (ON or OFF) for the resistors used at each end of the Data Highway network connection. The resistors are on the KTx card and are turned on/off through software. Refer to a Data Highway manual for more information on their use.
- **Baud Rate** is the network speed, either 57.6 Kbaud, 115.2 Kbaud, or 230.4 Kbaud.

MODULE CONFIGURATION EDITOR

SoftPLC supports loadable software modules that are written in C, C++, or Java. These loadable software modules are often referred to as TLMs (TOPDOC Loadable Modules.) TLMs can be for I/O, or other Devices (**Drivers**,) or they can be a container for user written ladder instructions (**Modules**), or both.

Drivers Driver TLMs are used to communicate to I/O or device networks. A DRIVER type TLM automatically gets CPU control on each PLC scan, whereas a MODULE type TLM does not get CPU control each scan unless called by ladder instruction. Forcing of discrete I/O points can be supported.

Modules

Module TLMs are normally a container for ladder instructions, but can also be loaded and can run as a separate task within SoftPLC. Module TLMs can be computations, data manipulators, or device interfaces (e.g.: motion control cards, serial communications protocols.) A Module TLM can contain a number of TLIs (TOPDOC Loadable Instructions.) If you load the module you have access to all the TLIs contained within it. TOPDOC NexGen treats TLIs, once loaded, like other ladder instructions.

TLMs are loaded by SoftPLC at runtime when you select to **Use** a module in the Module Configuration Editor (Figure 3-10), which adds an entry to a text file called MODULE.LST. Local TLM data is stored in \softplc\plc\<plcname>\MODULE.LST and the remote TLM data is in the file /SoftPLC/run/MODULE.LST within each SoftPLC. The Module Editor edits these MODULE.LST files.

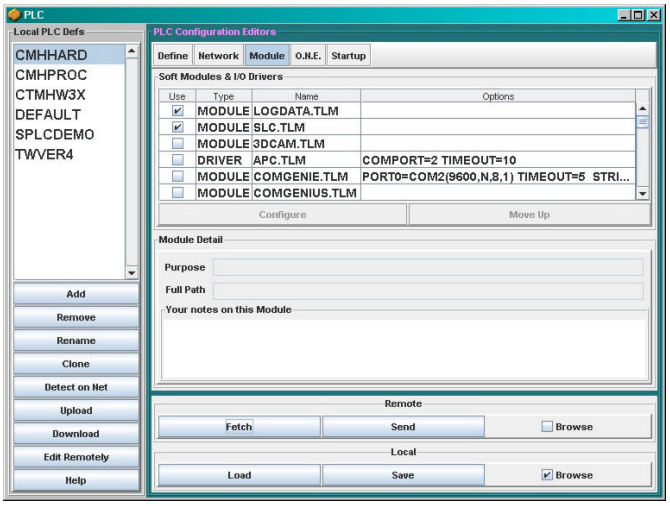

Figure 3-10
TOPDOC NexGen Module Configuration Editor Panel

You use the TOPDOC NexGen Module Editor to install/uninstall and to configure TLMs for your SoftPLCs (Figure 3-10.) At the top of the Module Editor panel is a table with 4 columns:

- **Use**, if checked, indicates that this TLM is to be loaded at runtime by SoftPLC. TLMs that are listed, but where the Use box is not checked, are installed on the disk, but will not be loaded.

Some TLM's are SoftPLC runtime version specific. Before selecting to USE a TLM, make sure it will operate with your SoftPLC.

- **Type** is established by the TLM developer to be either DRIVER or MODULE for each TLM and is not user definable. TLM developers indicate the Type in the \softplc\plc\DEFAULTS.LST file.

- **Name** is the filename of the TLM without path information. See Full Path in the lower panel for the full remote name including path.

Options is a user definable text field that customizes the way the corresponding TLM operates. Each TLM sets its own rules for its command line options, but defaults are supplied when appropriate. When you Use a TLM, you can edit the options.

Module Detail is displayed in the bottom of the panel:

- **Purpose** is provided by the TLM developer and is a brief description of what the TLM does.
- **Full Path** is where the TLM resides in the remote SoftPLC. It is chosen by the TLM developer and displayed here.
- **Your Notes on this Module** can be used to save text that might be helpful to you in using this module.

MODULE EDITOR OPERATIONS (BUTTONS)

Configure will be enabled when a TLM is in Use (selected in the table) and that TLM uses a configuration file. Usually all DRIVER type TLMs need to be configured, and you will need to refer to separate documentation on how to configure them. For SoftPLC provided TLMs, this documentation is found in a specific .pdf file on SoftPLC Product CD.

Most TLMs' configuration files are text based, and when you press the **Configure** button, TOPDOC NexGen presents you with a text editor that can **Send**, **Fetch**, **Load**, or **Save** the corresponding text based configuration file. The TEALWARE DRIVER TLM is an exception to this. When you **Configure** Tealware, you are launched into the IOMAP editor for Tealware I/O.

Move Up lets you move a particular module up in the list. This lets you establish an order of loading within SoftPLC. Most TLMs have no preference with regard to load order.

AVAILABLE MODULES LIST

The list of available TLM's in the Module Editor is based on the *.def files in the local (TOPDOC) PC's **\SoftPLC\PLC** folder. If you want to USE a TLM that is not listed, you will need to obtain or create the proper *.def file for that TLM and install it in the TOPDOC PC **\SoftPLC\PLC** folder.

Some TLM's require configuration or other auxiliary files. The *.TLM and any auxiliary files, editors, etc. must also be installed in the TOPDOC PC's **\SoftPLC\tlm** folder (or the location referenced in the *.def file) to be able to use the TLM in offline APP editing. Additionally, for offline APP editing, a local PLC definition must also be created that has the *.TLM **Checked for Use** in the Module Editor.

When you select to USE a TLM, make sure the *.TLM and any required configuration or other auxiliary files are installed in the SoftPLC's **/SoftPLC/tlm** directory (or the location required by the TLM.) Otherwise, SoftPLC will not start. You can use FTP to transfer files to the SoftPLC.

HOW SOFTPLC INTERACTS WITH I/O DRIVERS

Regardless of mode, I/O Drivers are called from SoftPLC once each SCAN. If SoftPLC is in PROGRAM, or FAULT mode, however, no I/O access takes place. I/O Drivers are also called from SoftPLC whenever a mode change takes place either via the Keyswitch or remotely via TOPDOC NexGen. I/O Drivers that have Block Transfer capability are also called once every execution of 100 rungs of ladder logic to service any pending Block Transfer requests.

In addition, each I/O Driver is called once when SoftPLC is being started and once when SoftPLC is being shutdown. This allows I/O Drivers to perform one time initialization tasks such as reading the command line arguments specified in MODULE.LST file, loading "smart" cards, allocating memory, or installing interrupt vectors. Being called prior to SoftPLC's shutdown lets the I/O Drivers do clean up such as shutting down outputs, freeing memory or restoring interrupt vectors.

Several I/O drivers are included with a standard SoftPLC installation. The most popular of these are briefly described in this section. SoftPLC Corp. sells the interface card(s) for these drivers, or you can purchase them from a variety of other sources.

Most drivers require a configuration file and may also include ladder instructions to help perform functions such as smart card configuration. User's Guides for each driver are available on the SoftPLC Product CD or from the Knowledge Base on SoftPLC's website.

I/O vendors, systems integrators, and other SoftPLC Partners have written other I/O drivers. A complete list of drivers and compatible hardware products can be found on our website.

I/O Device Drivers support I/O forcing for all digital points mapped to the SoftPLC I/O image tables. Some drivers have been written as ladder instructions for more time-sensitive control, however, these might not support forced I/O. Again, the detailed driver User's Guide describes the full capabilities and requirements.

TEALWARE I/O DRIVER (TEALWARE.TLM)

The Tealware™ I/O Driver is used to control local and/or SoftPIPE remote configurations of SoftPLC in Tealware I/O for either a Tealware CPU based system, or when Tealware I/O is used with the SPO-TWRIO (PCI bus), or PCCARD10 (ISA bus) interface cards. Up to 16K digital I/O can be supported from a single channel, at a distance of up to 20K feet. A separate Windows program, found on the SoftPLC CD, called **SoftPLC I/O Map Editor** is used for configuration.

ALLEN-BRADLEY REMOTE I/O DRIVER (KTX.TLM)

The Allen-Bradley KTx I/O Scanner Driver is used to support up to eight (8) Allen-Bradley interface cards. The cards operate in scanner mode and allow SoftPLC to access up to 32 Allen-Bradley remote I/O adapters (8K logical racks) per channel. A-B Interface cards supported and their SoftPLC Catalog number are:

Allen Bradley Part Number	SoftPLC Part Number
1784-KTX	ICO-KTX
1784-KTXD	ICO-KTXD
1784-KTS	ICO- KTS
1784-PKTX	SPO-PKTX
1784-PKTXD	SPO-PKTXD
1784-PKTS	SPO-PKTS

Supported devices include A-B models 1771, 1747, flex and block I/O, as well as compatible I/O systems such as WRC's SmartMUX, Digital and analog “smart” I/O modules, and other devices are all supported (e.g.: motion controllers, robot controllers, drives) via block transfer instructions.

HILSCHER I/O DRIVER (HILSCHER.TLM)

The Hilscher I/O Driver supports Profibus, Interbus, DeviceNet, SERCOS, CANOpen, and other cards manufactured by Hilscher. SoftPLC resells all these cards. SoftPLC supports up to eight (8) cards simultaneously. Both master and slave modes are supported. A separate I/O configuration program (called SYCON) must be used, which is provided with the card at no additional charge.

STARTUP CONFIGURATION EDITOR – SOFTPLC LOADING OPTIONS

A file called STARTUP.LST contains runtime options that are used to specify the PLC Name, start-up operating mode, the ladder program file to run, and more.

TOPDOC NexGen’s Startup Editor is used to edit STARTUP.LST. The local version is in \SoftPLC\PLC\<PLCNAME>\STARTUP.LST and the remote version is in /SoftPLC/run/STARTUP.LST. Figure 3-11 shows the Startup Editor Panel.

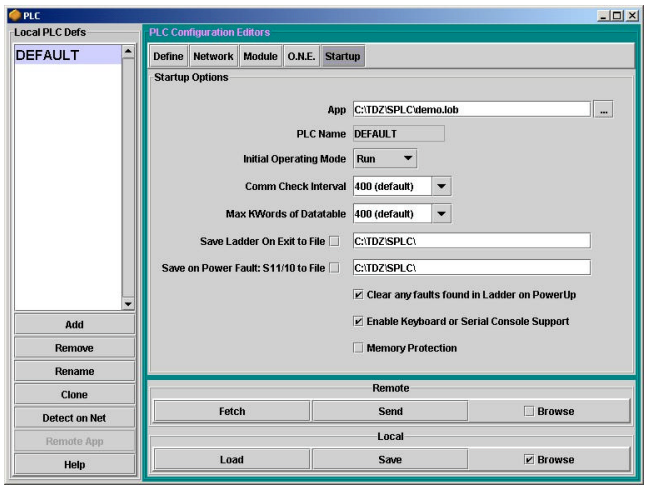

Figure 3-11
Startup Configuration Editor Panel

CONFIGURATION FIELDS

- **App** gives the name of the ladder application SoftPLC should load and run at startup. If you are connected to a SoftPLC, pressing the [...] button to the right of the text area will bring a popup showing the names of all the Apps in that SoftPLC. The specified APP file can reside on any SoftPLC recognizable drive including flash, ram, floppy, or hard disk. If a file is not specified, SoftPLC will start with a blank application called **default**.

- **PLC Name** should always agree with the local name of the PLC as given in the listbox on the left. To change this field, simply **Save** or **Send** the Startup data and then **Load** or **Fetch** the changed data back into the editor. If you **Save**, then **Load**. If you **Send**, then **Fetch**.

- **Initial Operating Mode** can be set to **Run**, **Program**, or **Test**. If your SoftPLC has a hardware keyswitch, the keyswitch position will override this setting. However, if the keyswitch is in REMOTE mode, this setting will determine which Remote mode is set at SoftPLC startup.

  If you select **Program** mode here, and have no keyswitch, then you would need to use TOPDOC NexGen to change the operating mode before SoftPLC would start running the ladder logic and controlling the I/O.

- **Comm Check Interval** adjusts the frequency that a SoftPLC checks its peer-to-peer messaging queues. A smaller number is a faster frequency but may slow down the overall scan time somewhat under a heavy messaging load.

  Checking a queue means seeing if any HMI application(s) have sent SoftPLC communication commands on any O•N•E channels that are currently in use. Each channel has its own queue. During each check, all active O•N•E channels are serviced, but only one packet is serviced from the front of each queue.

  This option can be used to tune the amount of CPU time SoftPLC gives to these communications requests. The default is to check after each 400 rungs, plus at the end of program scan. If your program has less than this setting's number of rungs, then only the end of program check is done and this setting will have no effect.

- **Max KWords of Data Table** sets a high limit for your data table size, where one kword is 1024 sixteen bit words. If you are running out of system RAM you can reduce this value. If you are running out of data table you can increase this value. The default is 200 kwords.

  If TOPDOC NexGen tries to expand the data table beyond this limit, the expansion attempt will fail and TOPDOC will report an error code. Actual APP files must always use less than this limit in order to be loaded into this SoftPLC, either through TOPDOC NexGen or from disk via the STARTUP.LST that loads SoftPLC.

- **Save Ladder on Exit to File** gives the name of an APP file that the current RAM contents will be written to whenever you exit SoftPLC via a keyboard attached to the SoftPLC (or via SSH or the CONSOLE utility.) This setting has no effect if you just power off a SoftPLC.

  This option can be used for retentive memory -- to track online programming changes. This filename may or may not be the same as the name given as the startup file on the command line.

- **Save on Power Fault** gives the name of an APP file that will be written to whenever bit S11/10 is turned on. The SoftPLC faults itself and saves the current SoftPLC RAM contents to this file. This support is also used by the APC TLM (see section on Retentive Memory on Power Loss.)

  Additionally, you can also set bit S11/10 via the ladder program to snap shot a ladder image and shut SoftPLC down. (e.g.: when a SoftPLC sees this bit on, it goes into Fault Mode and saves the memory image to the given filename. The SoftPLC will need to be restarted in order to resume operation.)

- **Clear any Faults in Ladder on Powerup** should normally be set. If a ladder image was saved by either Save Ladder on Exit to File or Save on Power Fault it is possible that data table fault bits were set into the image before saving. In order to be able to automatically restart after such a faulted saved image, SoftPLC needs to clear all the faults one time at startup.

- **Memory Protection**, if set will disable data table size changes and ladder program rung changes attempted by TOPDOC, regardless of the PLC operating mode. This is similar to RUN mode through a hardware keyswitch.

RETENTIVE MEMORY ON POWER LOSS

Through the use of an Uninterruptible Power Supply (UPS), you can configure SoftPLC so that online ladder logic and data table changes not yet saved to flash disk will be retained in the event of an unexpected loss of power.

SoftPLC provides a driver for American Power Conversion (APC) smart UPS's. The driver (APC.TLM) uses the UPS-Link control language that enables the UPS to communicate with SoftPLC through a serial port, using a special UPS-Link cable (SoftPLC Cat. No. ICO-CAPC.) The cable is included with UPS's sold by SoftPLC Corp.

When loaded, the driver will automatically cause the UPS to enter **smart signaling mode**, which allows the driver to determine if the UPS is operating in ON-LINE or ON-BATTERY mode. If the UPS goes into ON-BATTERY mode, SoftPLC will be put into Fault Mode. This is done by the driver setting S11:10 after a **TIMEOUT** period (user specifiable or default=10 seconds.) Once SoftPLC has been Faulted, the UPS will be automatically shut down within twenty (20) to thirty (30) seconds unless the **NOSHUTDOWN** command line option is specified. When normal power is restored, SoftPLC will re-boot and restart.

UPS DRIVER CONFIGURATION AND USE

To use the APC.TLM driver, select to Use it in the Module Editor. The Driver has 3 available configuration parameters:

COMPORT = 1 thru 4 (COM1=1, COM2=2, COM3=3, COM4=4)

The COM port used by the UPS cable (Cat. No. ICO-CAPC) must be reserved for this driver, and not used for any other purpose (e.g.: O•N•E, Console, COMGENIUS, or other TLMs.)

TIMEOUT = Number of seconds (5 to 1800) to run on battery before faulting SoftPLC. Default is 10 seconds.

NOSHUTDOWN = Add this keyword to disable automatic UPS shutdown after SoftPLC is faulted.

No ladder logic programming is required for the APC.TLM driver to interact with SoftPLC. Once the APC.TLM is selected to be used, it will automatically set the **Power Loss** major fault bit in SoftPLC (S11/10) if the UPS remains **On-Battery** past the **TIMEOUT** period. When the driver sets this fault bit, it causes SoftPLC to go into **Fault** mode immediately. During runtime, the **Low Battery** (S10/0) and/or **On Battery** (10/1) status bits are also set or reset by the TLM (where TRUE=1.)

If you have activated the Save on Power Fault feature (described earlier in this Chapter), before SoftPLC shuts down, the ladder logic program and data table in RAM memory will be saved to the specified disk and file. Note that if SoftPLC has been put into Fault mode, that status will be saved with the program.

If the **UPS-Link** serial cable is disconnected, UPS control will be bypassed. This enables you to service the UPS without faulting SoftPLC or having to shut down the SoftPLC system first. Once the serial communication cable is re-attached, the driver will re-activate automatic UPS control.

RESTARTING SOFTPLC AFTER POWERLOSS

If the **NOSHUTDOWN** option is used, there are 3 ways you can restore operation of SoftPLC:

If you have specified in the Startup Editor to not clear faults at start-up, only method 3 below can be used to clear the Fault and resume operation.

1. Re-boot the system by cycling power.

2. From a console prompt, manually shutdown SoftPLC, then re-load SoftPLC. This is described in Chapter 5.

3. Using TOPDOC's Data table Editor, manually clear the fault using the **Clear Faults** option in the **Modes** button menu. This causes SoftPLC to change from **FAULT** to **PROGRAM** mode. Then, using the **Modes** button, change the SoftPLC mode to **RUN**.

DETECT ON NET (WHO FUNCTION)

Detect On Net is quite powerful. It lets a qualified person walk into a new plant and in minutes have a complete local record of all SoftPLCs in the plant, including their Names, IP Addresses, and configurations.

This panel (Figure 3-12) shows SoftPLC's that were detected on a subnet using Detect On Net. It provides an easy way to automatically create or update your Local PLC Definitions to the correct IP Addresses and Names. It also provides an easy way to automatically Fetch and Save all your configuration files for all your connected SoftPLCs.

The Detect On Net panel is called up by pressing the Detect On Net button found on the main PLC Configuration Manager panel.

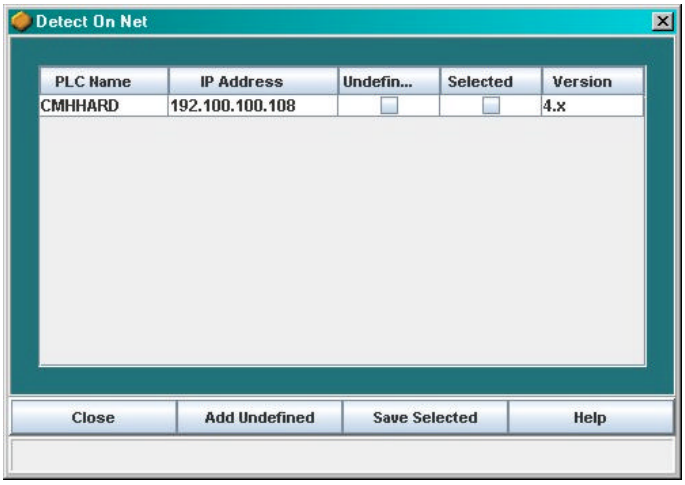

Figure 3-12
Example Detect on Net Panel

DETECT ON NET TABLE COLUMNS

- **PLC Name** is the name assigned by the Startup editor and should be unique across all SoftPLCs. This should be the same name shown in the left listbox of the PLC Configuration Manager. If it is not unique, you will need to correct this using the Rename button and the Startup Editor.

- **IP Address** is the actual current TCP/IP Address in the SoftPLC. This should agree with the one shown on the Define editor panel for this SoftPLC.

- **Undefined** will be checked by TOPDOC NexGen if either:
  - there is no entry in the PLC Configuration Manager for this PLC, **or**
  - the detected IP Address is in disagreement with the IP address recorded in the Define editor

- **Selected** is user changeable. SoftPLCs may be selected in support of the Save Selected button (see below.)

OPERATIONS (BUTTONS)

- **Add Undefined** will create or update definitions for all SoftPLCs shown as Undefined. A folder below \softplc\plc will be created if one does not exist already, and the Define data's IP Address will be made current.

- **Save Selected** operates only on Selected SoftPLCs. If any Selected SoftPLCs are also Undefined, it automatically does Add Undefined on those SoftPLCs. Next it does a Fetch and a Save for all Selected SoftPLCs for the following editors: Network, Module, O•N•E, and Startup. This is a simple way to get backups of all your PLC configurations for all your SoftPLCs.

CHAPTER 4 -
GATECRAFT FIREWALL SETUP

INTRODUCTION

This chapter will cover the setup of the firewall technology included with SoftPLC version 4.*x*. Since the firewall is integrated into Gatecraft Linux, the term Gatecraft will used throughout this chapter. Use of the firewall is optional.

Connection to Gatecraft may be achieved using a serial or ethernet port (the preferred method,) however the serial port will be required if the default ethernet address (IP) is not compatible with your existing network. This chapter will discuss how to connect using both methods.

Configuration of the Gatecraft may be done using the **setupgatecraft** shell script: one of the many shell scripts Gatecraft Linux uses internally when booting and shutting down the operating system. A shell script is similar to a MSDOS batch file, but much more powerful.

ESTABLISHING A CONNECTION TO GATECRAFT

HARDWARE REQUIREMENTS

- 9 pin din female/female Standard Null Modem cable
- Ethernet Cross Cable or ethernet switch with two (2) Ethernet Patch Cables.

SOFTWARE REQUIREMENTS

Terminal emulation program	HyperTerminal (Windows) minicom (Linux)
SSH client program	PuTTY (Windows) ssh (Linux)
FTP client	WSFTP or FileZilla (Windows) gFTP (Linux)
PLC Configurator	TOPDOC NexGen (Windows or Linux) (Hardbook only)

SYSTEM DEFAULTS

1st Ethernet Port IP Address	192.100.100.101
System login	login = root password = softplc
FTP login	login = anonymous password = softplc
Serial	baud=38400 No parity 8 bits 1 stop bit

SERIAL CONNECTION

To establish a serial connection using a terminal emulation program and the Gatecraft's serial console port:

1. Connect the client computer to the Gatecraft's serial console using a Null Modem cable.

2. Start the terminal emulation program with the communication parameters set to

38400 baud

8 bits

No Parity

1 Stop bit

3. Login to Gatecraft Linux

login=root

password=softplc

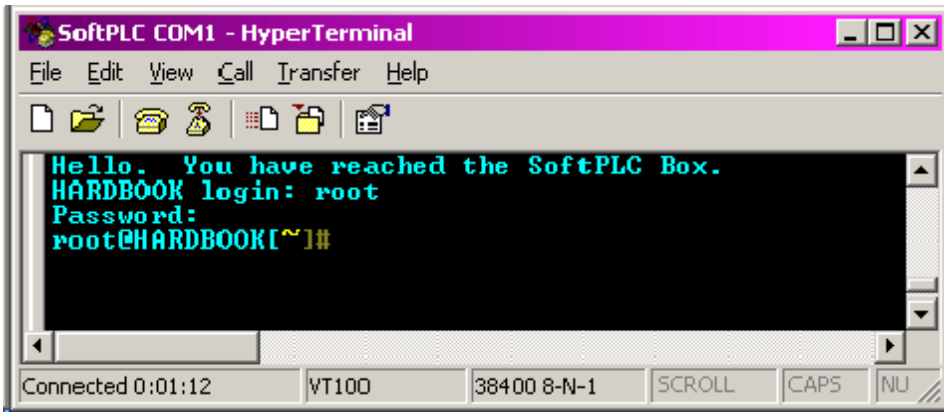

Figure 4-1
Serial Console Login

ETHERNET CONNECTION

To establish a Secure Shell (SSH) connection to the Gatecraft on ethernet:

1. Connect the client computer to the proper Gatecraft ethernet port (normally the first or lowest numbered port) using an ethernet cross cable. If both the Gatecraft computer and the client computer are connected to an ethernet switch, use an ethernet patch cable.

2. Make an SSH connection. The instructions differ depending on the Operating System of the client machine:

(Windows Option) **Start PuTTY. Select** SSH**, enter the IP address (default=192.100.100.101) of the Gatecraft, and press Open. Now login to Gatecraft Linux.**

login as: root

root@192.100.100.101's password: softplc

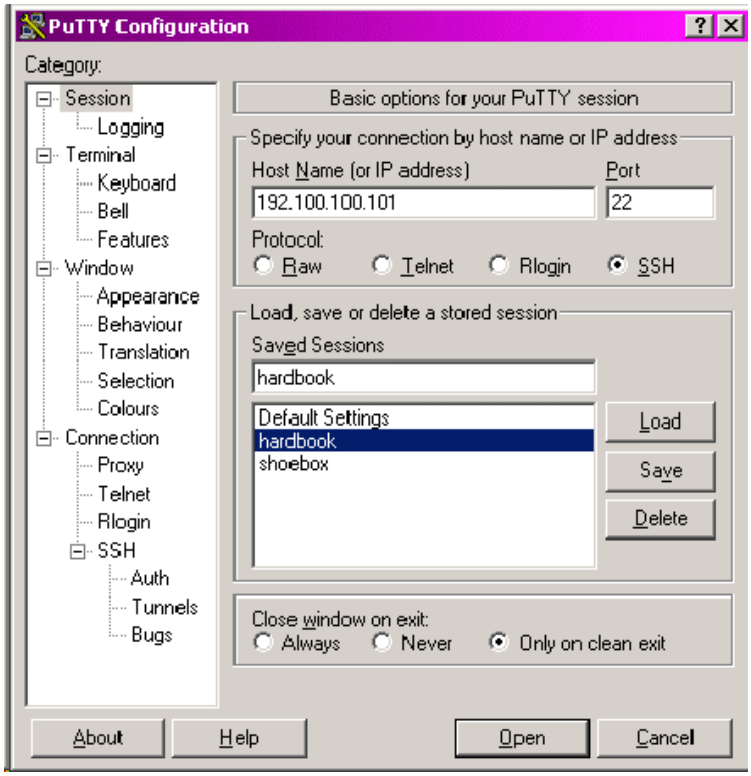

Figure 4-2
PuTTY SSH Login

(Linux Option) **Open an xterm window and connect to the Gatecraft using** ssh **(e.g.: ssh -l root 192.100.100.101.) When asked for the password, respond with** softplc.

GATECRAFT SETUP PROGRAM

Once you have a console prompt, start the **Gatecraft Setup Program** by typing **setupgatecraft** and then pressing the enter key.

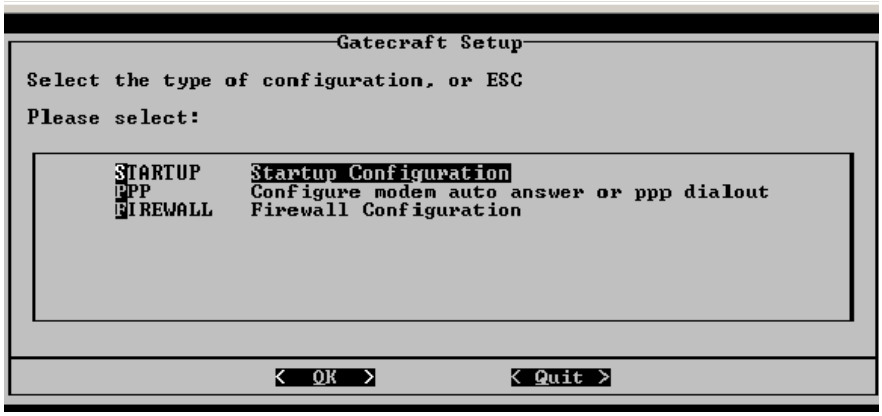

Figure 4-3
Gatecraft Setup Menu

The setupgatecraft menu is divided into 3 main sections:

Startup - Startup Configuration is required for all systems

PPP - The PPP menu is for Gatecraft Junction or a SoftPLC system that will use a serial modem.

Firewall - The Firewall may be configured and enabled for all systems.

The menu structure is shown below:

Startup - Startup Configuration
|------**S**TARTUP.LST - change Hostname, etc
|------**N**ETWORK.LST - define local IP address, DNS, default gateway
|------**R**unlevel.conf - init.d startup configuration
PPP - Configure modem auto answer or ppp dial-out
|------**P**PPCLIENT.LST - configure modem for ppp dial-out
|------**P**PPSERVER.LST -configure modem for dial-in and pppserver
|------**i**nittab - add/remove modem auto answer (#mgetty)
Firewall - Firewall Configuration
|------**S**oftPLC_protector
|------**i**p_masquerade
|------**r**aw
|------**E**dit
|------**C**lone

ACTIVATING YOUR CHANGES

There are several ways to activate specific changes but the simplest way is to reboot the system by cycling power or by using one of the following commands at the Gatecraft Linux command prompt:

reboot
init 6

STARTUP CONFIGURATION

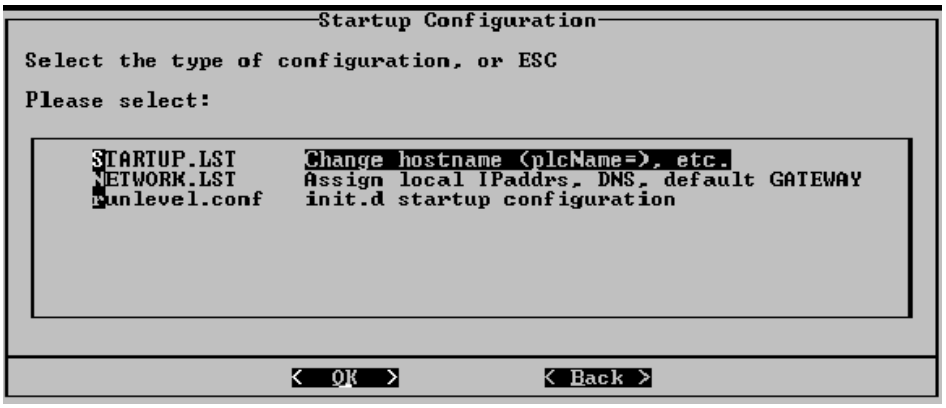

Figure 4-4
Startup Configuration Menu

STARTUP.LST

SoftPLC users should normally use TOPDOC NexGen's PLC Configuration **Startup Editor** rather than configuring through this program.

The **hostname** is set in STARTUP.LST. The text string assigned to **plcName=** will be read during the initialization phase occurring in the boot process and will overwrite the hostname string in the standard Linux configuration file **/etc/hostname**. The other options are specific to SoftPLC and do not apply to firewall operation.

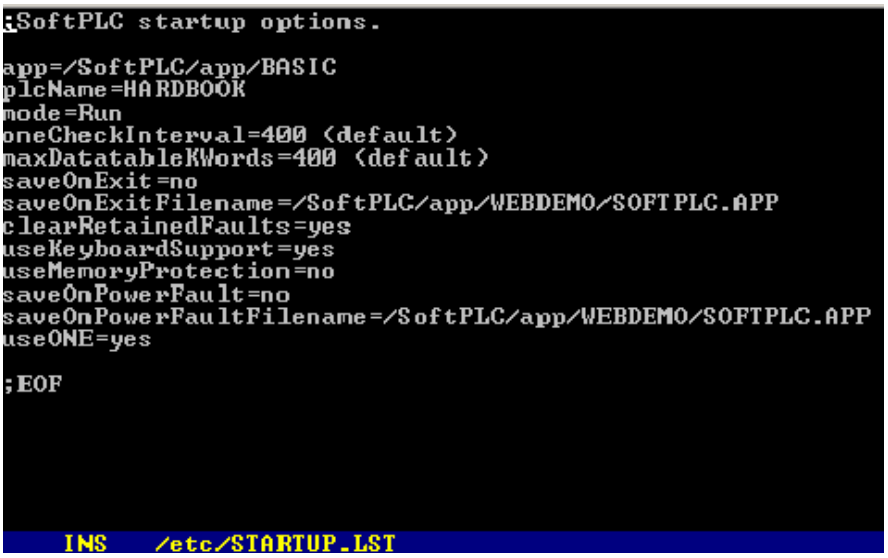

Figure 4-5
Startup Settings

To exit the editor, press **CTRL-q** and then **y** to save your changes (not case sensitive.)

NETWORK.LST

	SoftPLC users should normally use TOPDOC NexGen's PLC Configuration **Network Editor** rather than configuring through this program.

The ethernet adapter(s) IP address, DNS, default GATEWAY, and FTP password are set in NETWORK.LST.

The first entry under the **INTERFACES** section will be assigned to the first network interface adapter. Subsequent entries will be assigned to the next interface adapter(s.)

For dynamic permanent address assignment by an external BOOTP Server such as TOPDOC NexGen, the IPADDR and SUBNETMASK are replaced with the keyword BOOTP.

```
;This is the configuration file for NETWORK.DLL
;Change IPADDR, and SUBNETMASK as needed.
;See http://www.softplc.com/JAVAUSER.PDF for more details


[INTERFACES]
ETH IPADDR=192.100.100.101  SUBNETMASK=255.255.255.000
#ETH IPADDR=192.100.101.101  SUBNETMASK=255.255.255.000
#ETH BOOTP


[FTP]
PASSWORD=softplc

[ROUTING]
;GATEWAY=192.100.100.9
IPFORWARDING=YES

[DNS]
IPADDR=205.230.159.9
IPADDR=205.230.159.7

; EOF
 *  INS    /etc/NETWORK.LST
```

Figure 4-6
Network Settings

To exit the editor, press **CTRL-q** and then **y** to save your changes (not case sensitive.)

RUNLEVEL.CONF

The RUNLEVEL.CONF file determines which initialization scripts are executed in each of the seven (7) run levels (0-6.) The user may enable or disable scripts and services within the **Configuration** section by removing or adding the leading **#** (number symbol character.) The following scripts and services are available:

- Firewall
- BOOTP Server
- Kernel Logging
- BOOTP Relay
- SoftPLC Runtime
- PPP Client
- SoftPLC Webserver
- PPP Server
- SSH Server
- HotPlug

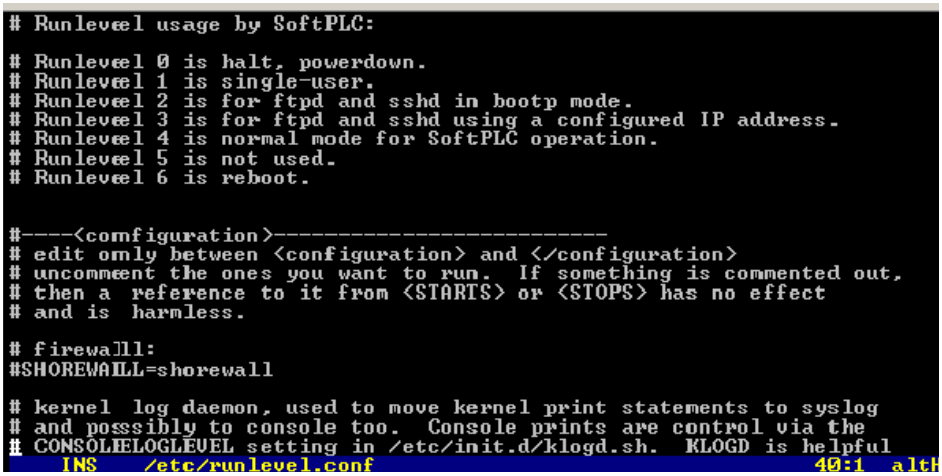

Figure 4-7
Run Level Settings

To exit the editor, press **CTRL-q** and then **y** to save your changes (not case sensitive.)

PPP CONFIGURATION

Point to Point Protocol (PPP) is the Internet standard for transmission of IP packets over serial lines. Gatecraft Junction supports both PPP dialout (pppclient) and Auto Answer (pppserver.) From the **setupgatecraft** main menu, Gatecraft Setup, select PPP to configure the pppclient and/or the pppserver.

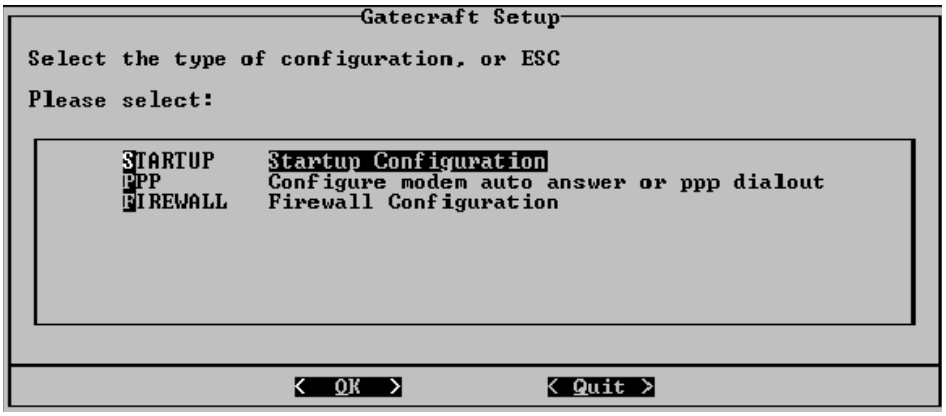

Figure 4-8
Gatecraft Setup Menu

GLOSSARY

CHAP	Challenge Handshake Authentication Protocol
PAP	Password Authentication Protocol
PPP	Point to Point Protocol

PPP CLIENT CONFIGURATION

To configure the PPP Client, select PPPCLIENT.LST from the PPP Setup menu.

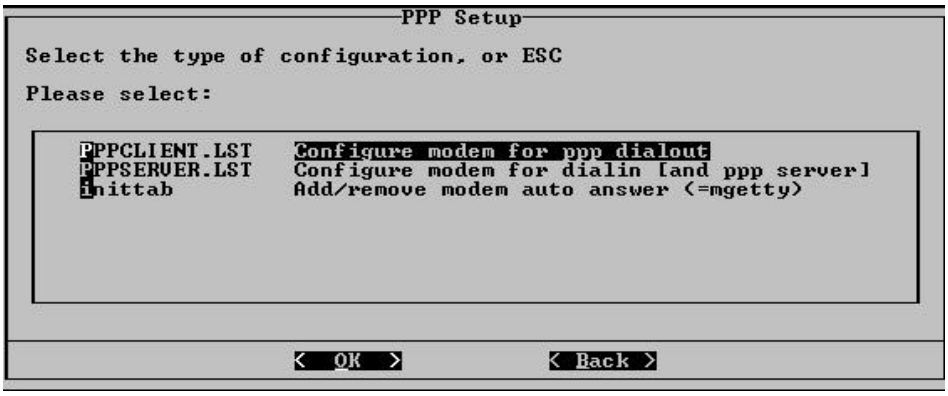

Figure 4-9
PPP Setup Menu

Figure 4-10 is a screen shot showing the configuration parameter screen.

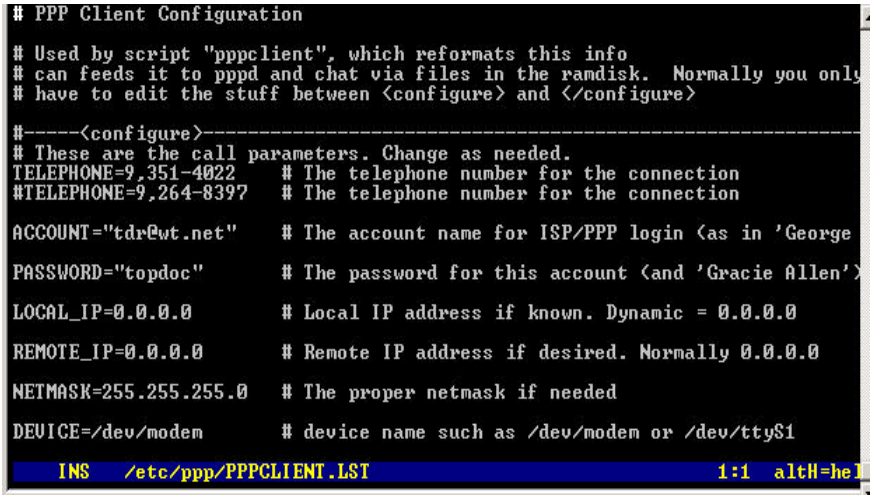

Figure 4-10
PPP Client Settings

To configure PPP Client, you must enter the information specific to your ISP

TELEPHONE=	Internet Service Provider's (ISP) telephone number, use a comma (,) for pause. ex. 9,351-4022
ACCOUNT=	Account ID used to log on to your ISP
PASSWORD=	Password for that account

The default configuration is for the PPP Client to dial the ISP and create a persistent connection. However if you require a dial-on-demand connection, set the idle (IDLE=) to a positive integer for the amount of time you want the connection to remain on before timing out (e.g. 120 seconds) and the demand mode to yes (DEMAND=yes.)

Additional PPP Client call parameters are listed in the following table. Seldom would you need to change these parameters.

LOCAL_IP=	Set to 0.0.0.0 (default) for Dynamic address
REMOTE_IP=	Remote IP, normally 0.0.0.0 (default)
NETMASK=	If needed Netmask, default is: 255.255.255.0
DEVICE=	Modem device name; default is **/dev/modem**
BAUDRATE=	9600, 19200, 38400, 57600 (default), 115600
DIALMETHOD=	Dial method where: T=Tone, P=Pulse
AUTHMETHOD=	PAP, LOGIN, or CHAP, use PAP if not sure!
IDLE=	Seconds to wait before hanging up. **-1** means never (default)
DEMAND=	**no** (default), dial immediately! **yes** dial when packet sent ppp0
CHATDEBUG=	turns off/on chat debugging default is Off (default=0)
FIREWALL=	(Optional) Firewall name to be loaded dynamically, default is **" "** (none) Valid firewall names are in **/etc/shorewallcfg**

We recommend that the firewall not be defined in the PPP Client configuration, but rather be loaded during the boot cycle by enabling the firewall option in runlevel.conf. (see **runlevel.conf** earlier in this chapter)

The PPP Client support is started by the shell script **/etc/init.d/pppclient.sh** which in turn starts the **Point to Point Protocol daemon** (**pppd**.) To start the pppclient automatically during the boot cycle, enable the pppclient script in **runlevel.conf**.

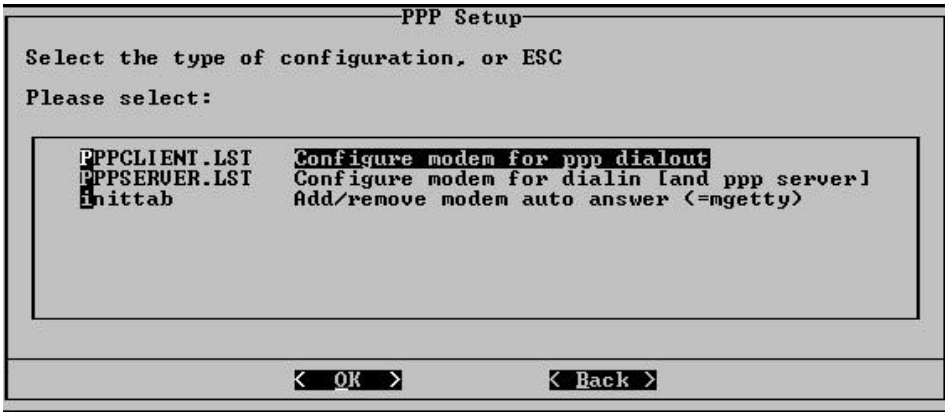

Figure 4-11
PPP Setup Menu

To start the pppclient immediately without rebooting, execute the pppclient script in **/etc/init.d** at the Gatecraft Linux command prompt. For example:

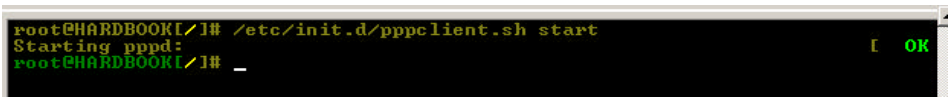

Figure 4-12
PPP Client Init Script

Gatecraft Linux will assign the dialout modem as network interface **ppp0.** This can be confirmed using the Linux command **ifconfig**.

```
root@HARDBOOK[~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:42:03:20:17
          inet addr:192.100.100.66  Bcast:192.100.100.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:42ff:fe03:2017/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:372 (372.0 B)  TX bytes:378 (378.0 B)
          Interrupt:9 Base address:0xc000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

ppp0      Link encap:Point-Point Protocol
          inet addr:10.64.64.64  P-t-P:10.112.112.112  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:552  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
```

Figure 4-13
Confirming PPP0 with ifconfig

PPP SERVER CONFIGURATION

Select **PPPSERVER.LST** to edit the remote dial in call parameters.

```
# These are the configurable parameters. Change as needed.

MYIPADDR=10.200.1.48     # IP address of the _local_ ppp interface

NETMASK=255.255.255.0    # The proper netmask if needed

DEVICE=/dev/modem        # device name such as /dev/modem or /dev/ttyS1

AUTHMETHOD=PAP           # PAP or CHAP supported

IDLE=-1                  # seconds to wait before hanging up.  "-1" means never

FIREWALL=""
                         # a shorewall config dir under /etc/shorewallcfg/,
                         # or "" if no firewall required.

                         # add new line for each ACCOUNT as needed.
                         # assignedIpAddr must use same subnet as MYIPADDR
ACCOUNTS='\
#user            server          password        assignedIpAddr
#-------------------------------------------------------------
"oscar"          *               "meyer"         10.200.1.50
"user"           *               "softplc"       10.200.1.50
    INS   /etc/ppp/PPPSERVER.LST                              9:1   altH=help NE
```

Figure 4-14
PPP Server Settings

There are a seven (7) PPP Server configuration parameters. For most installations, only the user accounts will need to be modified (user, password and assigned IP address.)

In some cases a users' preference may be to change the default interface IP address (MYIPADDR=.) If so, the IP address assigned to the dial-in user accounts must be assigned so it is on the same Network ID but has a unique Node ID. In the default configuration shown in Figure 4-14, the subnetmask is set to 255.255.255.0 which indicates that the first three digits of the IP address is the Network ID and the last digit is the Node ID. Therefore, the dial in caller's IP is set to **x.y.z.50,** which is different than the interface's assignment of **x.y.z.48**, which would then be a valid IP address.

The PPP Server support is started by a shell script (/etc/init.d/pppserver.sh.) To start the pppclient automatically during the boot cycle, enable the pppserver script in **runlevel.conf**. (see runlevel.conf)

If the pppserver interface is a modem, **mgetty** must be enabled in the inittab.

If both the pppclient and pppserver are enable, Gatecraft Linux will assign the dial in modem as network interface **ppp1**. If the pppclient is not enabled, the dial in modem will be assigned as network interface **ppp0**. This can be confirmed using the Linux command **ifconfig**.

INITTAB

The **inittab** file describes which processes are started at bootup and during normal operation. This file, **/etc/init.d/rc, /etc/init.d/rcS,** is called by the init binary during system initialization. If the PPP Server is being activated, initab will need to be edited to enable the modem's auto answer feature.

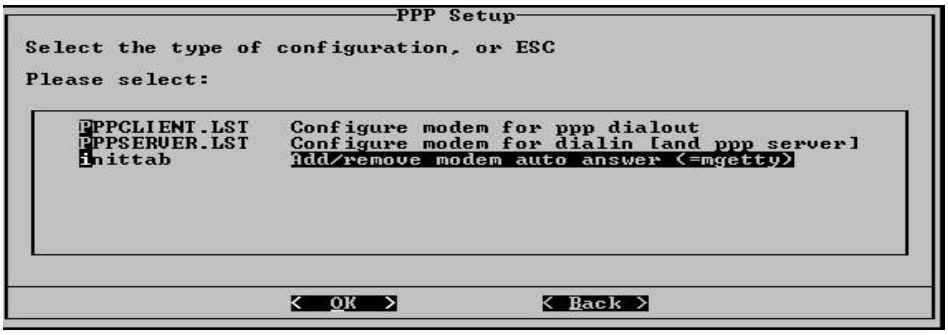

Figure 4-15
PPP Setup Menu

To enable the modem's auto answer feature, the **mgetty** daemon must be enabled. This is accomplished by removing the leading # (number symbol character) in the last line in the inittab file as shown in Figure 4-16.

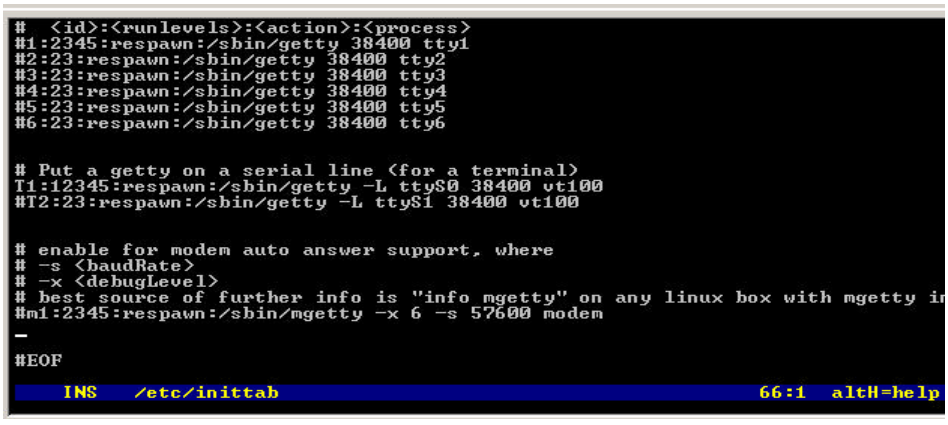

Figure 4-16
Modem Auto Answer in Inittab

After editing inittab, shutdown (init 0) and reboot Gatecraft Linux.

FIREWALL CONFIGURATION

The Gatecraft Firewall provides expert network traffic management using packet filtering. The first step in configuring the Gatecraft Firewall is to select a pre-defined rule set or create a new one by cloning the raw (default) set or one of the pre-defined sets. The selected rule set will be the one that is enforced when the system is rebooted; provided the Firewall script is enabled in **runlevel.conf**. (see runlevel.conf)

SELECT OR CLONE A FIREWALL RULE SET

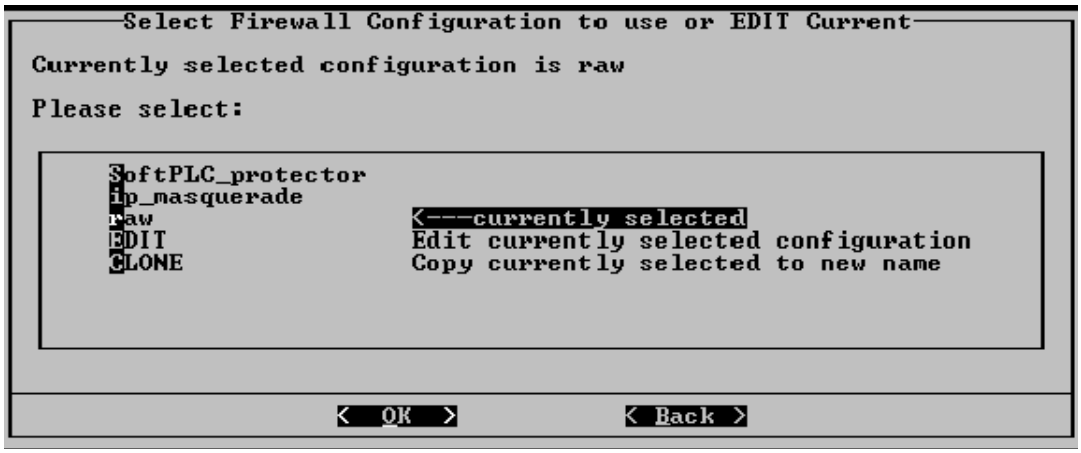

Figure 4-17
Firewall Configuration Selection

The SoftPLC_protector base rule set is designed to control traffic flow between the two ethernet ports, eth0 and eth1.

The ip_masquerade base rule set is designed to control traffic flow between a local ethernet port (eth0) and the Internet via the 2nd ethernet port (eth1) or a modem.

The raw or default base rule set has not been configured for any defined purpose.

Use the UP and DOWN arrow keys to position on the desired rule set or operation. Then press the **Enter** key to select it as the currently selected configuration. In this example, the raw base configuration is selected.

FIREWALL SETUP

There are five (5) basic configuration files that may need to be configured for a selected Firewall configuration: **interface**, **policy**, **rules**, **masq** and **blacklist**. Each of these configuration files has extensive comments that explain the purpose and use in great detail. A brief summary of the use of each file is provided below.

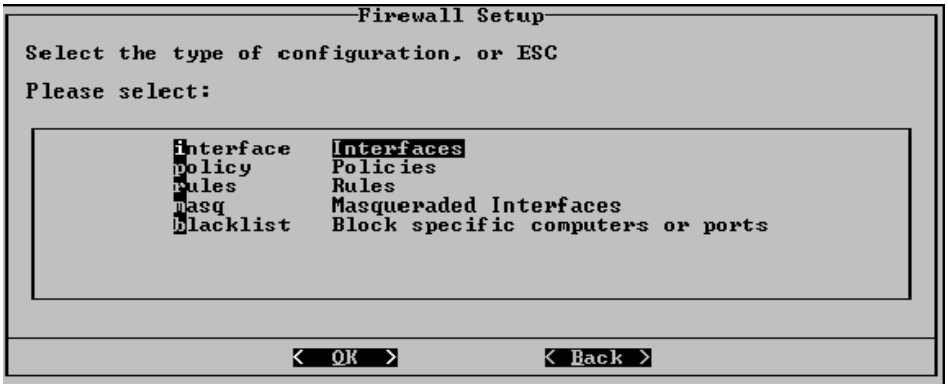

Figure 4-18
Firewall Setup Menu

Interface - An entry must be made in this file for each network interface to be used in the firewall system. There are four (4) possible fields for each entry.

ZONE	Zone for this interface. There are two predefined zones, **loc** or local network and **net** for a 2nd internal or external network
INTERFACE	Name of the interface, each may be listed once
BROADCAST	The actual broadcast address for the sub-network to which the interface belongs or the keyword **detect** may be used to automatically determine the broadcast address
OPTIONS	A comma separated list of options such as: **dhcp** and **black list**

Example: To assign the local network to eth0 and 2nd network or Internet connection to eth1.

#ZONE	INTERFACE	BROADCAST	OPTIONS
loc	eth0	detect	
net	eth1	detect	

Policy - This file defines what to do with a new connection request when no match is found in the **Rules** file to be defined in step 3. The most commonly used choices are: **ACCEPT**, **DROP**, and **REJECT**. There are five (5) columns or fields in for each entry and include:

SOURCE	Source zone (loc net, fw...)	
DEST	Destination zone (loc net) **all** or **fw** (firewall)	
POLICY	Policy if no match is found in the **rules** file. This will be:	
	ACCEPT	
	DROP	(ignore silent)
	REJECT	sends RST for TCP and port unreachable for others
	CONTINUE	
	NONE	
LOG LEVEL	(optional) If supplied each connection handled under the default POLICY is logged at that level	
LIMIT:BURST	(optional) If supplied specifies the maximum TCP connection rate and the size of an acceptable burst.	

Example 1: To allow all the local zone (eth0) to initiate communications to the 2nd zone (eth1):

#SOURCE	DEST	POLICY	LOG_LEVEL	LIMIT:BURST
loc	net	ACCEPT		
net	all	DROP		

Example 2: Let the **Rules** file, to be defined in step 3 below, manage all network traffic between the zones. Failed connections from the local zone **loc** will receive an error response while **net** initiated failed connections will not.

#SOURCE	DEST	POLICY	LOG_LEVEL	LIMIT:BURST
loc	net	REJECT		
net	all	DROP		

Rules - Rules in this file govern connection establishment and are processed ahead of the policy file of step 2 above. There are five (5) required fields for each entry and four (4) optional fields at the end of each line. The required fields are:

Action	**ACCEPT**	
	DROP	
	REJECT	
	DNAT	Port Destination Network Address Translation or Port Forwarding, is used to forward a connection to another system.
SOURCE	Source hosts to which this rule applies. This may be a single host, several hosts, a zone (loc, net, fw), or all zones (all.)	
DEST	Location of server to which this rule applies.	
PROTO	Specific protocol this rule defines. This must be **tcp**, **udp**, **icmp**, **a number**, or **all**	
DEST PORT(S)	Destination Ports. A comma-separated list of port names from **/etc/services**, port numbers or port ranges.	

If masquerade is used on a local zone, DNAT must be used to allow a connection from the Internet to a specific machine.

Example 1: Allow ssh, ftp and ping to firewall from local zone.

#ACTION	SOURCE	DEST	PROTO	PORT(S)
ACCEPT	loc	fw	tcp	ssh
ACCEPT	loc	fw	tcp	ftp
ACCEPT	loc	fw	tcp	ftp-data
ACCEPT	loc	fw	icmp	echo-request

Example 2: If the **net** zone is connected to the Internet, allow programs running on the firewall to have DNS access using the tcp or udp protocol.

#ACTION	SOURCE	DEST	PROTO	PORT(S)
ACCEPT	fw	net	tcp	domain
ACCEPT	fw	net	ucp	domain

Example 3: Limit traffic between the loc and net zones by IP address

#ACTION	SOURCE	DEST	PROTO	PORT(S)
ACCEPT	loc: 192.100.100.24	net	tcp	http, ftp
ACCEPT	loc: 192.100.100.201	net	ucp	smtp, domain

Masq - Use this file to define dynamic NAT (Masquerading) and to define source NAT (SNAT, or **static** IP Masquerading.)

INTERFACE	Outgoing interface, usually the one connected to the Internet
SUBNET	Interface or subnet that you wish to masquerade
ADDRESS	Optional: used when you want to specify the source address explicitly

Example: A simple masquerading setup where eth1 connects to a DSL or cable modem and eth0 connects to your local network.

#INTERFACES	SUBNET	ADDRESS
eth1	eth0	

Blacklist - This file contains a list of IP addresses, sub-networks, or MAC addresses. When a packet arrives on an interface that has the **blacklist** option specified in the interface's file (see Firewall setup: Interface section), its source IP address is checked against this file and disposed of accordingly. There is one (1) required and 2 optional fields.

ADDRESS/SUBNET	Host address, subnet or MAC address to be blocked. MAC address must be prefixed with **~** (tilde) and use **-** (hyphen) as a separator. For example: ~00-A0-C9-15-39-78
PROTOCOL	(Optional) If specified must be a protocol number or protocol name from **/etc/protocols**
PORT	(Optional) The port may only be specified when the Protocol is tcp (6) or udp (17.) The assignment may be a comma separated list of port numbers or service names from **/etc/services**.

Example: Suppose you want to prevent Jimmy Shimmy's laptop (192.100.100.24) from downloading files via FTP. Jimmy's laptop is assigned to the local zone which has the **blacklist** option enabled in the **Interface** configuration file. The entry in the **blacklist** configuration file would be:

#ADDRESS/SUBNET	PROTOCOL	PORT
192.100.100.24	tcp	ftp

FREQUENTLY ASKED QUESTIONS

How to set the system time:

date [MMDDhhmmCCYY.ss]

Example: Set system date to November 11, 2004 10:30:00 AM

date 111610302004.00

How to set the hardware clock to the current system time:

hwclock -w -u

What are the reserved non-routable (Private) IP addresses per RFC 1918?

10.0.0.0 - 10.255.255.255

172.16.0.0 - 172.31.255.255

192.168.0.0 - 182.168.255.255

Where can I find a list of the service-names and the assigned ports for the common network services, such as: ssh, ftp, http?

See **/etc/services on Gatecraft Linux**

e.g.: **cat /etc/services | less**

CHAPTER 5 -
SOFTPLC OPERATION/TROUBLESHOOTING

SOFTPLC MODES OF OPERATION

PROGRAM	SoftPLC does not access I/O nor solve ladder logic. Logic and data table can be changed. Data table size can be modified; new files can be added and/or existing files can be resized.
TEST	SoftPLC reads the inputs, solves the ladder logic, but does NOT write to the outputs. As in RUN mode, data table can not be resized in TEST mode; however, ladder logic can be changed.
RUN	SoftPLC goes through repetitive, synchronous scan cycles that read inputs, solve the ladder logic, and write outputs. Any requests from O•N•E channels are also serviced. Data table can NOT be resized, but values can be changed; ladder logic cannot be modified.
FAULT	SoftPLC automatically goes to FAULT mode as a result of a runtime fault. The execution of the ladder program is halted and outputs are shut down. The Status file is updated with the cause of the fault and where in the ladder program the fault occurred.

Changing the SoftPLC Mode

TOPDOC NexGen can be used to put SoftPLC in either **RUN**, **PROGRAM**, or **TEST** mode. If the SoftPLC CPU has an attached Keyswitch, the keyswitch sets the mode. If the keyswitch is in the **REMOTE** position, then TOPDOC NexGen sets the mode.

HOW SOFTPLC OPERATES

INITIALIZATION

- SoftPLC Runtime loads.
- Initializes I/O drivers and TLM's (loadable instructions.)
- Checks for SoftPLC copy protection key. If one does not exist, SoftPLC is started in DEMO mode without TLM's.
- Creates the data table in RAM memory.
- Loads the APP (program file) specified in STARTUP.LST.
- Selects mode specified in STARTUP.LST, or if hardware keyswitch exists, reads keyswitch to determine mode.
- Updates time/date in Status file to computer's realtime clock.
- Opens the specified O•N•E communication channels.

SCAN CYCLE

After initialization, SoftPLC enters a SCAN loop and exits only upon fault, user command (e.g.: changing mode with TOPDOC or Keyswitch) or power loss. In the SCAN loop, SoftPLC:

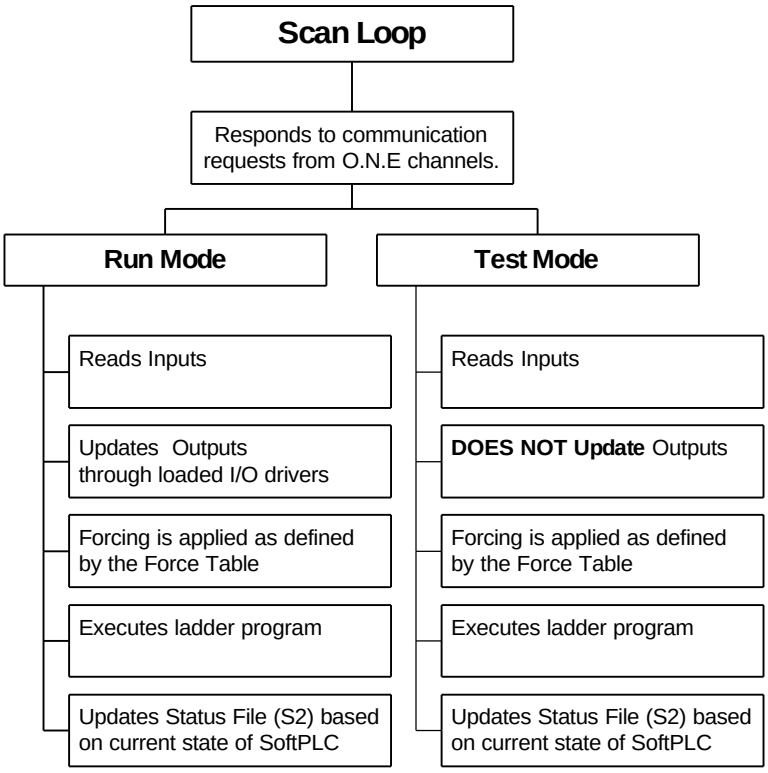

Figure 5-1
Scan Cycle

DENINSTALLATION

SoftPLC execution can be stopped by using a keyboard. When this happens, SoftPLC performs the following:

- Frees RAM memory allocated for data table
- Closes O•N•E Communication channels
- Instructs I/O driver(s) to shutdown and clear all outputs
- Removes I/O Drivers from memory
- Exits to Gatecraft Linux

USING SOFTPLC'S SSH AND FTP SERVERS

Since most SoftPLC systems include only a flash disk and no keyboard or monitor, the easiest way to maintain files on the flash or for troubleshooting is often through SoftPLC's embedded SSH and FTP Servers.

FTP stands for File Transfer Protocol, and is a way to transfer or edit files on a TCP/IP network. SSH stands for Secure Shell and is an encrypted communication protocol that allows you to login to the SoftPLC over the network, to execute commands in the SoftPLC, and to move files between SoftPLC and your computer.

FTP

Example uses of the FTP Server are as a convenient way to update SoftPLC system files or to maintain a complete backup of your SoftPLC's on a file server or other computer.

You communicate to SoftPLC's FTP Server through any FTP Client. If you are using Linux for your operating system, we recommend the freeware program **gFTP.** If you are using Windows for your operating system, a very basic FTP Client program is included, however we do not recommend using it. Our recommendation is to use **Filezilla**, which is included on the SoftPLC Product CD. Use of Filezilla is described below.

Only a single FTP session is supported by SoftPLC. There is a built-in 120 second timeout period if the FTP Client should go inactive while still logged in. If your FTP Client stops communicating, you have probably exceeded the timeout and will need to re-connect.

You can apply security to this connection by using a login password. This is set up in the Network Configuration Editor file, FTP tab.

USING FILEZILLA CLIENT PROGRAM

To install Filezilla from the SoftPLC Product CD choose **SoftPLC Dev. Software and Tools**, then **Filezilla**. Follow the installer program instructions. Once you have Filezilla installed, you can launch it by clicking on the Filezilla icon.

You will first need to set up a **Site Session**. Select **File**, **Site Manager**, then **New Site**. Enter the following (as appropriate for your SoftPLC system parameters):

- Host **192.100.100.101**
- Port: **21**

Select the **Normal** logon type. Enter **<anything>** in the user field and the appropriate password that was setup for FTP in NETWORK.LST.

Select **Connect** to connect to SoftPLC, or **Save and Exit** to permanently save this information.

Once connected, using Filezilla is fairly straightforward. You can view files in either system by changing to the correct directories. Highlight a file, then click on **VIEW** to see the contents.

To transfer a file from one system to the other, display the correct directory in the receiving system, then highlight the file you want to transfer. Drag the file name to transfer.

To edit a file, highlight it, then click the right mouse button to bring up a menu. Select **EDIT**. If the file is remote, it will be transferred to the local system automatically, and after you **Edit/Save**, the revised file will be sent back to the remote system.

SSH

You can use SSH to transfer files to SoftPLC, edit files in the SoftPLC, start/stop SoftPLC, view SoftPLC's log files, and other command line operations.

USING AN SSH CLIENT PROGRAM

If you are using Linux for your operating system, use **ssh** to establish a Secure Shell. If you are using Windows, you need to install **PuTTY** or another SSH client of your choosing. PuTTY can be found on your SoftPLC Product CD under SoftPLC Dev. Software and Tools. To install PuTTY, from the SoftPLC Product CD just choose it and follow the instructions. Once you have PuTTY installed, you can launch it by clicking on the PuTTY icon.

To establish a Secure Shell (SSH) PuTTY connection to the SoftPLC on ethernet:

1. Connect the client computer to SoftPLC ethernet port using an ethernet cross connect cable or an ethernet patch cable if both CPU's are connected to an ethernet switch.

2. Make an SSH connection.

   (Windows Option) Start PuTTY. Select SSH enter the IP address of the SoftPLC, and press Open to connect.

   login=**root**
   password=**softplc**

   (Linux Option) Open an xterm window and connect to the SoftPLC using SSH When asked for the password respond with softplc.

USING THE CONSOLE

The Gatecraft Linux console provides a way for you to troubleshoot and perform low-level maintenance on SoftPLC through a serial connection. This is extremely useful if ethernet communications are not working or if SoftPLC will not load and run.

To establish a serial connection if you are using Linux as your operating system, we recommend you use **minicon**. To establish a serial connection if you are using Windows for your operating system, use **HyperTerminal**:

1. Connect the client computer to the SoftPLC's serial console using the Null Modem cable on COM1.

2. Start **HyperTerminal** or **minicon** with the communication parameters set to

   38400 baud,
   8 bits,
   No Parity,
   1 Stop bit

3. Login

   login=**root**
   password=**softplc**

TROUBLESHOOTING GUIDELINES

Once a SoftPLC system has been successfully installed, most troubleshooting will be done with TOPDOC NexGen or via standard hardware troubleshooting techniques. Most problems after installation occur due to component failures in the I/O, cabling, etc.

Proper installations should include power protection and noise/surge suppression equipment. Some intermittent problems may be due to improper system protection, particularly when using remote I/O and ethernet communications.

As a SoftPLC system is in development/testing stages, errors in configuration are the most common problems users encounter.

This section briefly describes how to determine the cause of most common problems. Because SoftPLC systems can be used with so many different types of hardware, networks, etc., detailed troubleshooting is beyond the scope of this Reference Guide. If, after trying the suggestions in this section, you still need assistance, contact SoftPLC technical support.

COMMONLY USED TOPDOC TROUBLESHOOTING TOOLS

TOPDOC includes a number of functions to help troubleshoot SoftPLC systems. These include I/O forcing, search/trace functions, watch windows, and more. Additionally, the following functions are very useful:

- The **Detect on Net** function can be used to verify ethernet communications, the IP address of the SoftPLC and whether the SoftPLC is running.
- If a SoftPLC was running, but has stopped, you can use the **Mode** button to check the SoftPLC operating mode, to be sure it is not Faulted or in Program mode.
- The PLC Configurator can be used to determine whether a configuration setting is the cause of the problem.

TROUBLESHOOTING USING THE SYSTEM LOG

SoftPLC maintains a system log (SYSLOG) of activities relating to SoftPLC startup, shutdown, communication accesses, and certain runtime errors. By reviewing the SYSLOG, many SoftPLC problems can be quickly detected, particularly configuration and certain hardware errors.

The default is for the SYSLOG to reside in RAM memory. You can redirect SYSLOG to a file. Since the log is often too large to see on a single screen display, the procedure below recommends you look at the SYSLOG in a file.

To read SYSLOG, you need to access the Gatecraft Linux command prompt. You can do this on ethernet using SSH or serially, using Hyperterminal. Accessing the command prompt is described earlier in this Chapter - Using an SSH Client Program or Using the Console.

The command to read SYSLOG is **logread**. Some useful options for this command are:

logread >syslog.txt	Redirects SYSLOG to a text file called **syslog.txt** on the SoftPLC. You can then use **EDIT** to view the file.
logread |less	Will pause the screen display so that you can scroll up/down within pages of the log file. To exit, press **q** anytime you have a **:** prompt.

```
Jun 20 13:48:48 SOFTPLC1 syslog.info syslogd started: BusyBox v1.1.0 (2006.03.16-05:47+0000)
Jun 20 13:48:48 SOFTPLC1 authpriv.info dropbear[387]: Running in background
Jun 20 13:48:48 SOFTPLC1 user.info : SoftPLC : version 4.0.060608
Jun 20 13:48:48 SOFTPLC1 user.info :  Copyright(C) 1993-2006 SoftPLC Corporation
Jun 20 13:48:48 SOFTPLC1 user.info :  ALL RIGHTS RESERVED.
Jun 20 13:48:48 SOFTPLC1 user.info : SoftPLC : serial no. 2000-08270, &LOADLOC=0xb7d5dbb0
Jun 20 13:48:49 SOFTPLC1 user.info : SoftPLC : loaded "/SoftPLC/app/WEBDEMO/SOFTPLC.APP"
Jun 20 13:48:49 SOFTPLC1 user.info :  No TLM's loaded.
Jun 20 13:48:49 SOFTPLC1 user.info : ONE     : version 4.0.060606
Jun 20 13:48:49 SOFTPLC1 user.info : ONETCP  : version 4.0.060606
Jun 20 13:48:49 SOFTPLC1 user.info : ONECOMAB: version 4.0.060606
Jun 20 13:48:49 SOFTPLC1 user.info : SoftPLC : now running in REM_RUN mode.
Jun 20 13:48:54 SOFTPLC1 user.notice root: Weblet  : version 4.0, running from config context: /SoftPLC/web/lst/usr/
Jun 20 13:48:55 SOFTPLC1 authpriv.info dropbear[426]: Child connection from 192.100.100.25:5124
Jun 20 13:49:05 SOFTPLC1 authpriv.notice dropbear[426]: password auth succeeded for 'root'
Jun 20 13:51:37 SOFTPLC1 user.info : SoftPLC : shutting down.
Jun 20 13:51:37 SOFTPLC1 user.info : SoftPLC : closing ONE channel 0
Jun 20 13:51:37 SOFTPLC1 user.info : ONETCP  : shutting down.
Jun 20 13:51:37 SOFTPLC1 user.info : SoftPLC : closing ONE channel 1
```

Figure 5-2
Example SYSLOG

MANUALLY STARTING/SHUTTING DOWN SOFTPLC

There may be times you will want to Shut Down or Restart SoftPLC without cycling power. For example, any time you change the SoftPLC's configuration or other system files you need to restart SoftPLC before the change can take effect. You may want to remotely Shut Down SoftPLC to transfer new files or other maintenance. Manual/Remote Shut Down/Restart can be done from a Gatecraft Linux command prompt. You can do this on ethernet using SSH or serially using Hyperterminal. Accessing the command prompt is described earlier in this Chapter - Using an SSH Client Program or Using the Consol.

Following are the commands you use:

/etc/init.d/softplc.sh stop	Shuts Down SoftPLC (see Deinstallation)
/etc/init.d/softplc.sh start	Starts SoftPLC
/etc/init.d/softplc.sh restart	Starts SoftPLC without having to perform the manual **Stop** and **Start** (useful when new files have been transferred to flash and you want changes to become effective)

COMMON PROBLEMS

SoftPLC won't start after changing a configuration file/parameter:

- check configuration using TOPDOC PLC Configurator **Fetch** options
- use logread to see error

SoftPLC is running, but there are communication problems:

- verify whether device IP address resides on same subnet as SoftPLC
- use PING.EXE to check low-level ethernet communications
- verify serial port settings match (eg: baud rate, checksum...)
- check cable connections

SoftPLC is running but I/O stops working:

- is SoftPLC in Fault mode?
- check I/O card seating
- check cables
- verify I/O configuration

CHAPTER 6 -
MEMORY ORGANIZATION/DATA TABLE ADDRESSING

SOFTPLC USER MEMORY

Your application program consists of four main parts - the **ladder logic program**, the **data table** where the real I/O values and other parameter data is stored, the **property files**, and the application **documentation**.

TOPDOC and SoftPLC maintain all this information in a file called **SOFTPLC.APP**, which is uploaded from or downloaded to SoftPLC with TOPDOC. The folder where each SOFTPLC.APP is located determines its name. In addition, your online **RUN** mode changes from TOPDOC are limited to the information in these parts of memory. All other SoftPLC files need to be maintained through the included SSH/FTP servers or direct manipulation of the disk files when SoftPLC is shut down.

PROGRAM MEMORY ORGANIZATION

The application program ladder logic can be nicely divided into a main program and many subroutines. Each of these is referred to as a **program file**. Program files are numbered starting at P2 (program files 0 and 1 are reserved by SoftPLC and are not accessible to users.) SoftPLC's can have subroutines numbered P3 through P9999.

<table>
<tr><td rowspan="2">Reserved by SoftPLC</td><td>P0</td><td></td><td></td></tr>
<tr><td>P1</td><td></td><td></td></tr>
<tr><td rowspan="4">Main Program</td><td rowspan="4">P2</td><td>P2/0</td><td rowspan="4">Rungs</td></tr>
<tr><td>P2/1</td></tr>
<tr><td>. . .</td></tr>
<tr><td>P2/x</td></tr>
<tr><td rowspan="4">Subroutines</td><td>P3</td><td>P3/0-P3/x</td><td rowspan="4">Rungs</td></tr>
<tr><td>P4</td><td>P4/0-P4/x</td></tr>
<tr><td>. . .</td><td></td></tr>
<tr><td>P9999</td><td>P9999/0-P9999/x</td></tr>
</table>

Figure 6-1
SoftPLC Program Memory Organization

Each program file can include up to 10,000 rungs, numbered 0 to 9999. The naming convention is **<program file #>/<rung number>**. For example, P4/65 is the 66th rung of Program File 4.

UNDERSTANDING THE DATA TABLE

SoftPLC stores data values (real I/O status, stored constants, and instruction parameters) in an area of memory referred to as the **data table**. The data table can be divided into different areas to accommodate different types and formats of data values.

The data table can be thought of as a big array made up of **bits**, which are the smallest division of the data. A bit can have a value of zero (0) or one (1.) You can access any bit in the data table with a formatted address. The data table array is organized 16 bits wide, with each row of 16 bits making up a **data table word**. Every data table word can be accessed with a formatted address. The data can be viewed in a number of different formats based on how you need to use it (signed integer, hexadecimal, octal, etc.) More complex **structures** are simply combinations of these bits and words, organized to make their meaning easier for you.

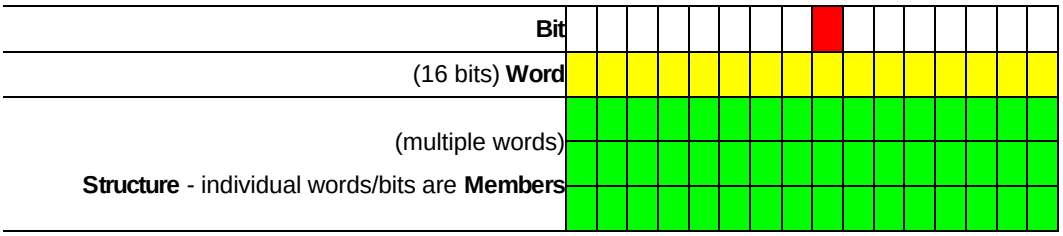

Figure 6-2
SoftPLC Data Table Components

The data table can be configured with many **data table files** of a number of different **data types**. A **data table file** is a group of logically related data of the same type. SoftPLC's can have data table files numbered 0 through 9999.

Each data table file is made up of a group of elements. SoftPLC's can have up to 9999 elements per data table file.

For some data types, the individual elements are single words. Some data types consist of elements that are multiple words. Finally, some data types are composed of a group of elements, referred to as a **structure**. A structure is subdivided into **members** at the bit or word level. You can access both structures and their members with a formatted address.

In addition to SoftPLC's actual data table file organization, you should organize the data within files into data blocks that help you group and organize related program data. You can address these data blocks in ladder instructions with a leading # sign, the starting address within the data block, and a length. These are called **instruction files** and are used in **File**, **Shift Register**, and **Sequencer Instructions**.

Data Organization Tips

1. Group data by similar kind, such as batch recipes or data to be transmitted to another SoftPLC or PC application.

2. Leave gaps to allow for online programming changes because the data table configuration (number of files and file sizes) cannot be changed in RUN mode.

3. (A-B Remote I/O only) Group block transfer data for smart modules.

DATA TABLE FILE TYPES/SIZES

Table 6-1 lists the SoftPLC data types available, as well as other information about each type. Note that there are 3 reserved data table file numbers 0, 1, 2. SoftPLC determines the size of these files. All other file numbers and sizes are user configurable.

In addition to the data table types found in the following tables, you can sometimes use integer program constants which are not stored in the data table, but in the instruction. TOPDOC NexGen will prompt when constants are allowed during instruction entry.

Table 6-1
Data Table File Types

FILE TYPE	CHARACTER	NUMBER (Default file)	MAX ELEMENTS/FILE	WORDS/ELEMENT
Output Image	O	0	8 (SPZ-LT) 64 (SPZ-1K) 128 (SPZ-2K) 512 (SPZ-8K)	1
Input Image	I	1	8 (SPZ-LT) 64 (SPZ-1K) 128 (SPZ-2K) 512 (SPZ-8K)	1
Status	S	2	32 fixed	1
Bit (binary)	B	3-9999 (3)	10000	1
Timer	T	3-9999 (4)	10000	3
Counter	C	3-9999 (5)	10000	3
Control	R	3-9999 (6)	10000	3
Integer	N	3-9999 (7)	10000	1
Floating Point	F	3-9999 (8)	10000	1
BCD	D	3-9999	10000	1
Message	MG	3-9999	10000	56
PID	PD	3-9999	10000	54
String	ST	3-9990	10000	82
Block Transfer	BT	3-9999	10000	56

Table 6-2 shows the valid data range for each data type.

Table 6-2
Valid Data Ranges

DATA TYPE	VALID DATA RANGE
Integer (also includes N, O, I, A, D, T, C, R)	-32768 to 32767
Floating Point	$\pm 1.1754944e^{-38}$ to $\pm 3.4028237e^{+38}$
Constant	-32768 to 32767

DATA TABLE ADDRESSING

Each data table location can be referenced by its address. You can directly address any file (or group of elements within a file), any structure, any member, any word, or any bit in the data table. In addition, a feature called **indirect addressing** can be used to allow value substitution at run time (described later in this chapter.)

Figure 6-3 shows how addresses are built, and the following tables provide additional details and examples.

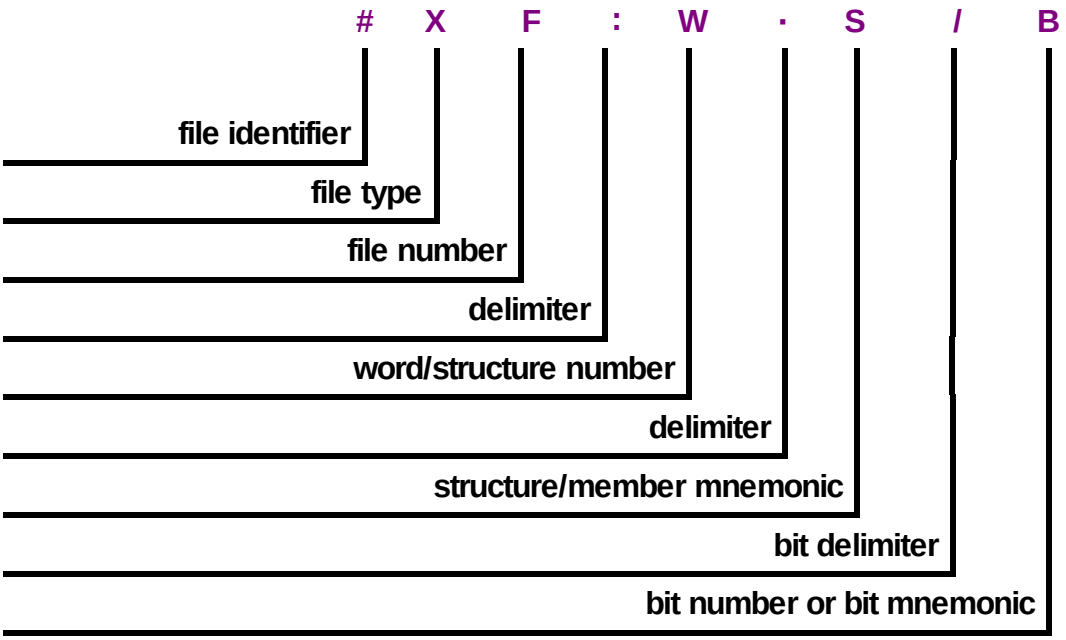

Figure 6-3
Logical Address Components

Table 6-3
Logical Addressing Detailed Description

#	Indicates the entire data table or instruction file addressed as an entity. Omit for bit, word, and structure addresses.		
X	**File Type**	A = ASCII B = Binary BT = Block Transfer C = Counter D = BCD F = Floating Point I = Input	MG = MessaGe N = iNteger O = Output PD = PID R = ContRol T = Timer ST = STring
F	**File Number**	0=output 1=input	2=status 3 ~ 9999=all other types
:	Colon delimiter separates file and word/structure numbers.		
W	**Word/Structure Number**	0 ~ 777 octal for input/output files 0 ~ 31 decimal for status file 0 ~ 9999 decimal for all other file types	
.	Period delimiter is used with structure-member mnemonics (e.g.: timer, counter, control, PD, MG, and BT addresses.)		
S	**Structure/Member Mnemonic**	Refer to specific structure definitions in the TOPDOC Help File	
		E.g.: Timer/Counter	Word 0 = .CTL Word 1 = .PRE Word 2 = .ACC
/	Bit delimiter separates bit number or bit mnemonic.		
B	**Bit Number** **or** **Bit Mnemonic**	00 ~ 07, 10~17 octal for input/output (I/O) files 00 ~ 159999 decimal for binary files (B types) when using direct bit address 00 ~ 15 decimal for all other file types **or** bit mnemonics for status bits within structures (e.g.: EN for enable bit of Timer/Counter.)	

I/O ADDRESSING

Addresses in the Output and Input image tables (files 0 and 1) are normally used to address input or output bits corresponding to real world devices and are always numbered in octal, not decimal. Therefore, I/O addresses are treated differently than all other data table addresses. Table 6-4 shows examples.

For I/O addresses, you do not need specify the file number (since it is always 0 or 1.) Simply use I or O, then the word number and/or bit address. (e.g. I07/17)

I/O word and bit addresses are number in octal, not decimal.

Table 6-4
Input/Output Addressing Examples

LEVEL	EXAMPLE
I1/14	Bit number 14 in octal of Input word 01. (The file number and colon are implied and do not need to be entered.)
O12/17	Highest bit, number 17 (octal), in Output word 12
I2	Input word 2, the 3rd element in the Input File

SoftPLC I/O drivers for different hardware I/O systems map the physical I/O addresses to the SoftPLC data table in different ways. Some analog or special I/O, maps to the I/O files, others to other data types (e.g.: N files.) Refer to the individual I/O driver documentation for your system for details on the relationship between a physical device and the logical I/O data table address.

ADDRESS MNEMONICS AND STATUS BITS

Address mnemonics let you address individual members of a structure at the word or bit level by an abbreviated name instead of a physical number. The available mnemonics varies based on the type of data and the instruction with which you are using the address. The ladder instruction information in the TOPDOC NexGen Help file indicates which mnemonics applies to each address as used in different instructions. Table 6-5 gives example mnemonics and Table 6-6 gives examples of addressing data at different levels.

Table 6-5
Example Mnemonics

INSTR/DATA LEVEL		WORD LEVEL		BIT LEVEL	
CTU	Counter (C)	**accumulated**	.ACC	**overflow**	.OV
FIFO	Control (R)	**length**	.LEN	**enable unload**	.EU
PID	PID (PD)	**setpoint**	.SP	**PV tracking**	.PVT

EXAMPLE ADDRESSING

Table 6-6
Addressing Data At Different Levels – Examples

LEVEL	EXAMPLE	DESCRIPTION
File	#N12	iNteger file, number 12 (# is used to indicate level=file)
Word (single word data types)	N23:4	iNteger file, number 23, **:** (colon) file delimiter, word four (5th element)
Element (float data type)	F8:6	Floating point file, number 8, word 6 (7th element) [Two 16 bit words are used to store the value]
Word Member (structure data type)	T14:5.ACC	Timer file, number 14, **:** (colon) file delimiter, element number 5, **.** (period) element delimiter, ACC word mnemonic
Bit (single word data types)	N17:34/9	iNteger file, number 17, **:** (colon) file delimiter, word number 34, **/** (forward slash) bit delimiter, bit number 9 (10^{th} bit)
Bit (binary files)	B3/275	Binary file number 3, **/** (forward slash) bit delimiter. 276^{th} element Binary files are continuous bit streams, so you can address by bit alone as shown here, or in the format above.
Bit (structure data type)	C5:2/DN	Counter file, number 5, **:** (colon) file delimiter, element number 2, **/** (forward slash) bit delimiter, DN bit mnemonic

INDIRECT ADDRESSING

In addition to directly addressing data table elements, you can indirectly address the file number, element number, and/or bit number part of an address. SoftPLC will use the value stored in the indirect part of the address to **resolve** the address at runtime.

Therefore, you can use ladder logic to change the value stored at the substitute address. This is a valuable tool for applications such as indexing sequential batch recipe files. For example, changing from one batch to another could be done with an address such as #N[C5:2.ACC], where as the counter accumulates, you would reference #N10, #N11, #N12...

Rules:

1. The substitute address must be a word address of type N, T, C, R, B, I, O, or S. If you use a T, C, or R address, you must use a word-length member such as R6:0.POS or T4:0.ACC.

2. The indirect address is entered in brackets [ ].

3. You cannot use indirect addresses for I/O addresses.

Table 6-7
Indirect Addressing Examples

INDIRECT ADDRESS	VARIABLE
N7:[T4:0.ACC]	The accumulated value of T4:0 determines which word in file N7 is referenced by the instruction.
N[C5:2.ACC]:0	Word 0 of the file being referenced is determined by the accumulated value of C5:2.
B3/[N7:0]	The value in N7:0 determines which bit is referenced by the instruction.

APPENDIX A - GATECRAFT LINUX BASIC SHELL COMMANDS

cat file	Print contents of file to screen
cd directory	Change to directory
cp file1 file2	Copy file1 to file2

less file	Show contents of file
ls [options] [files]	List files, compressed (ls) long format (ls -l) show hidden files (ls -a) show hidden in long format (ls -al)

mkdir directory-names	Create directories
mv file1 file2	Move, rename file1 to file2
rm file	Remove file
rmdir directories	Remove empty directories

tar - [cx t zj v] -f archive-file	create or extract list table of archive file verbose and z (.gz) or j (.bz2) Ex: tar -xzvf tar.gz or tar -xjvf tar.gz2

ifconfig	List network interfaces

find [path ...] -name file	Search for file in a directory hierarchy. Ex: find/-name STARTUP.LST

which command	Locates a command in path

date [MMDDhhmmCCYY.ss]	Set system date.

hwclock -w -u	Sets the clock to the current system time.

	For specific Shell command options, use –help, for example: **ls –help**

For more information about Linux shell commands, the Internet is a great source for free material.

APPENDIX B - O·N·E SETUP FOR THE MSG INSTRUCTION

O·N·E SETUP DEFINITIONS

Table B-1
O·N·E Setup Definitions

Command Set	Application layer packet format carried by some wrapper protocol and media type. SoftPLC uses the DH+ command set.
Transaction	All transactions consist of a COMMAND packet sent out by an initiator and a REPLY packet send out in response by a responder.
Command	First part of a transaction.
Reply	Response to a Command.
Initiator	Node that initiates the transaction, whether it is a read, write, change mode, or other type of command.
Responder	Node that replies to a command as part of a transaction

O·N·E SETTINGS

Unlike other O·N·E channel types, the TCP/IP media type has some special requirements:

Table B-2
O·N·E Special Requirements

Each O·N·E channel definition is either a client or a server.
SoftPLC can only act as a responder on the server channel, it may not act as an initiator on that channel.
Each SoftPLC node must have at **most** one server channel.
SoftPLC can only act as an initiator on a client channel, it may not act as a responder on that channel.
Each peer node that you wish to talk to must be defined in a separate O·N·E client channel.

Example: Image of a 3 node ethernet consisting of 2 SoftPLC's and a PLC-5.

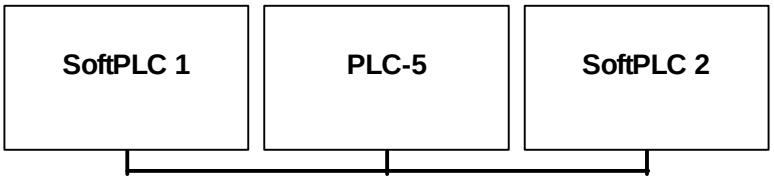

Figure B-1
TCP/IP MSG Instruction Example

SoftPLC 1 must read from and/or write to SoftPLC 2 using one or more MSG Instructions. SoftPLC 1 is the initiator and SoftPLC 2 is the responder. SoftPLC 1 must also be a responder to the PLC-5, as does SoftPLC 2. SoftPLC 2 is never an initiator. Here are the various O·N·E channel definitions that you would need to support this setup.

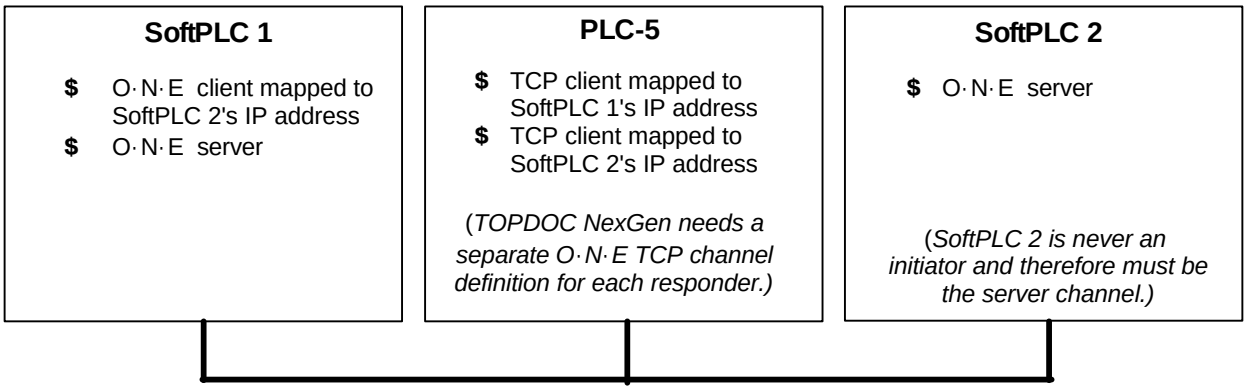

Figure B-2
Example O·N·E Channel Definitions