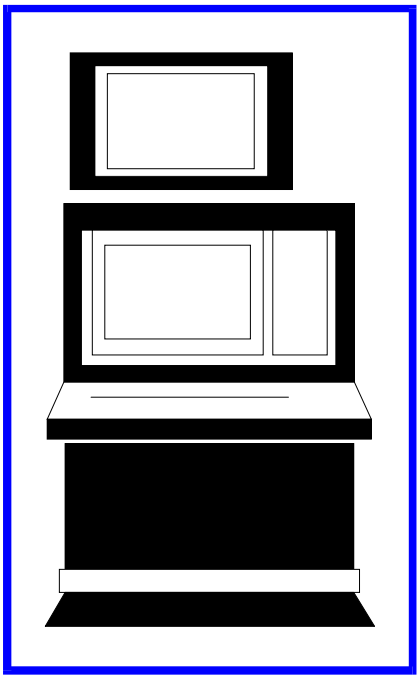
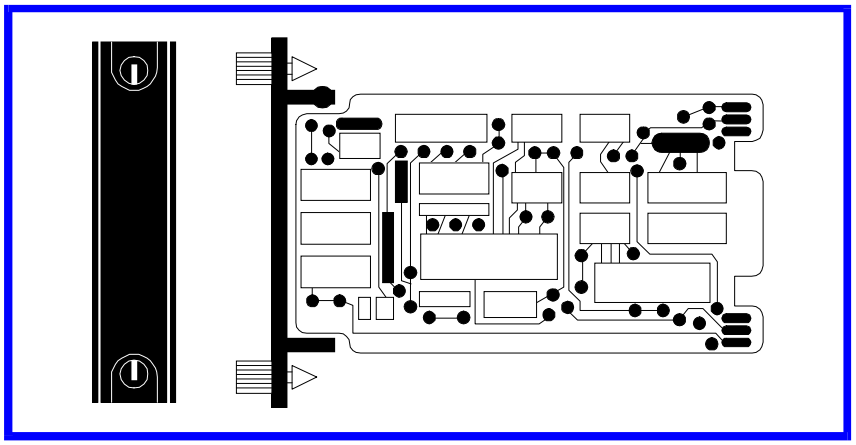
Bailey® infi 90®

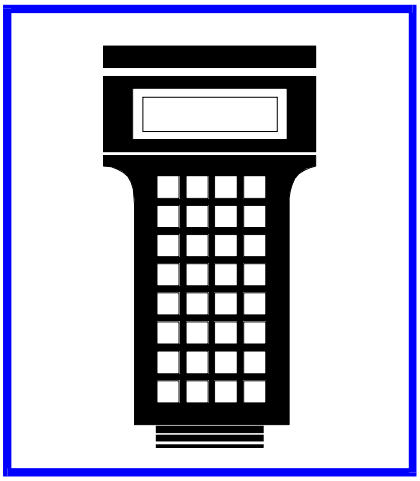
E96-705

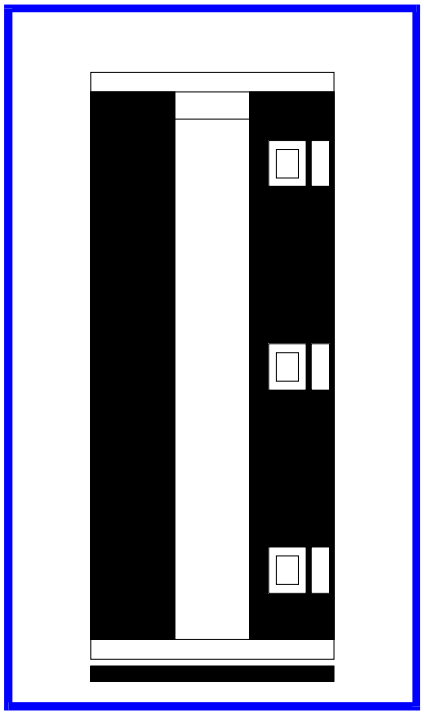
Instruction

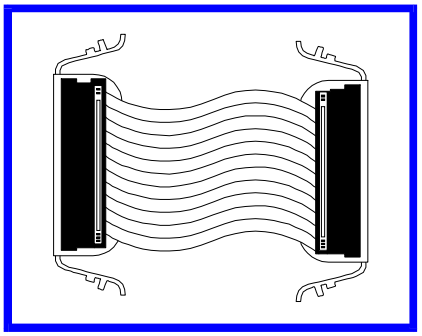
Personal Computer Software
LADDER Logic Configuration Utility (SLAD)
(Release 3.0)

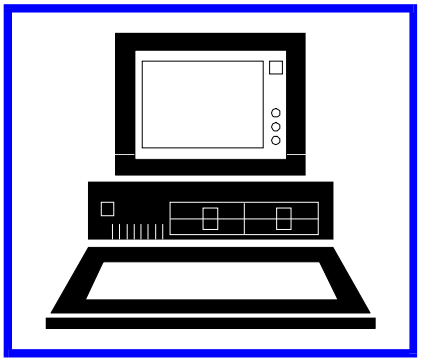












Bailey Process Control and Automation Solutions from Elsag Bailey Group

WARNING notices as used in this instruction apply to hazards or unsafe practices that could result in personal injury or death.

CAUTION notices apply to hazards or unsafe practices that could result in property damage.

NOTES highlight procedures and contain information that assists the operator in understanding the information contained in this instruction.

WARNING

INSTRUCTION MANUALS

DO NOT INSTALL, MAINTAIN, OR OPERATE THIS EQUIPMENT WITHOUT READING, UNDERSTANDING, AND FOLLOWING THE PROPER **Elsag Bailey** INSTRUCTIONS AND MANUALS; OTHERWISE, INJURY OR DAMAGE MAY RESULT.

RADIO FREQUENCY INTERFERENCE

MOST ELECTRONIC EQUIPMENT IS INFLUENCED BY RADIO FREQUENCY INTERFERENCE (RFI). CAUTION SHOULD BE EXERCISED WITH REGARD TO THE USE OF PORTABLE COMMUNICATIONS EQUIPMENT IN THE AREA AROUND SUCH EQUIPMENT. PRUDENT PRACTICE DICTATES THAT SIGNS SHOULD BE POSTED IN THE VICINITY OF THE EQUIPMENT CAUTIONING AGAINST THE USE OF PORTABLE COMMUNICATIONS EQUIPMENT.

POSSIBLE PROCESS UPSETS

MAINTENANCE MUST BE PERFORMED ONLY BY QUALIFIED PERSONNEL AND ONLY AFTER SECURING EQUIPMENT CONTROLLED BY THIS PRODUCT. ADJUSTING OR REMOVING THIS PRODUCT WHILE IT IS IN THE SYSTEM MAY UPSET THE PROCESS BEING CONTROLLED. SOME PROCESS UPSETS MAY CAUSE INJURY OR DAMAGE.

AVERTISSEMENT

MANUELS D'OPÉRATION

NE PAS METTRE EN PLACE, RÉPARER OU FAIRE FONCTIONNER L'ÉQUIPEMENT SANS AVOIR LU, COMPRIS ET SUIVI LES INSTRUCTIONS RÉGLEMENTAIRES DE **Elsag Bailey**. TOUTE NÉGLIGENCE À CET ÉGARD POURRAIT ÊTRE UNE CAUSE D'ACCIDENT OU DE DÉFAILLANCE DU MATÉRIEL.

PERTURBATIONS PAR FRÉQUENCE RADIO

LA PLUPART DES ÉQUIPEMENTS ÉLECTRONIQUES SONT SENSIBLES AUX PERTURBATIONS PAR FRÉQUENCE RADIO. DES PRÉCAUTIONS DEVRONT ÊTRE PRISES LORS DE L'UTILISATION DU MATÉRIEL DE COMMUNICATION PORTATIF. LA PRUDENCE EXIGE QUE LES PRÉCAUTIONS À PRENDRE DANS CE CAS SOIENT SIGNALÉES AUX ENDROITS VOULUS DANS VOTRE USINE.

PERTURBATIONS DU PROCÉDÉ

L'ENTRETIEN DOIT ÊTRE ASSURÉ PAR UNE PERSONNE QUALIFIÉE EN CONSIDÉRANT L'ASPECT SÉCURITAIRE DES ÉQUIPEMENTS CONTRÔLÉS PAR CE PRODUIT. L'AJUSTEMENT ET/OU L'EXTRACTION DE CE PRODUIT PEUT OCCASIONNER DES À-COUPS AU PROCÉDÉ CONTRÔLE LORSQU'IL EST INSÉRÉ DANS UNE SYSTÈME ACTIF. CES À-COUPS PEUVENT ÉGALEMENT OCCASIONNER DES BLESSURES OU DES DOMMAGES MATÉREILS.

NOTICE

The information contained in this document is subject to change without notice.

Elsag Bailey, its affiliates, employees, and agents, and the authors and contributors to this publication specifically disclaim all liabilities and warranties, express and implied (including warranties of merchantability and fitness for a particular purpose), for the accuracy, currency, completeness, and/or reliability of the information contained herein and/or for the fitness for any particular use and/or for the performance of any material and/or equipment selected in whole or part with the user of/or in reliance upon information contained herein. Selection of materials and/or equipment is at the sole risk of the user of this publication.

This document contains proprietary information of Elsag Bailey, Elsag Bailey Process Automation, and is issued in strict confidence. Its use, or reproduction for use, for the reverse engineering, development or manufacture of hardware or software described herein is prohibited. No part of this document may be photocopied or reproduced without the prior written consent of Elsag Bailey.

Preface

This manual describes the operation of the utilities provided with the LADDER Logic Configuration Utility (SLAD), software release 3.0.

The LADDER Logic Configuration Utility user is required to have a thorough familiarity with ladder logic, the INFI 90® Strategic Process Management System, and PC-DOS based computer operation. These utilities provide the user with the capability to create process control configurations off-line and to monitor and tune the process on-line.

Refer to related documents listed under ***REFERENCE DOCUMENTS*** in Section 1 for additional information.

® Registered trademark of Elsag Bailey Process Automation.

List of Effective Pages

Total number of pages in this instruction is 136, consisting of the following:

Page No.	Change Date
Preface	Original
List of Effective Pages	Original
iii through vii	Original
1-1 through 1-5	Original
2-1 through 2-2	Original
3-1 through 3-4	Original
4-1 through 4-8	Original
5-1 through 5-30	Original
6-1 through 6-5	Original
7-1 through 7-4	Original
8-1 through 8-2	Original
9-1 through 9-1	Original
A-1 through A-47	Original
B-1 through B-4	Original
C-1 through C-1	Original
D-1 through D-3	Original
E-1 through E-7	Original
F-1 through F-4	Original
Index-1 through Index-2	Original

When an update is received, insert the latest changed pages and dispose of the superseded pages.

NOTE: On an update page, the changed text or table is indicated by a vertical bar in the outer margin of the page adjacent to the changed area. A changed figure is indicated by a vertical bar in the outer margin next to the figure caption. The date the update was prepared will appear beside the page number.

Table of Contents

Page

SECTION 1 - INTRODUCTION 1-1
- OVERVIEW 1-1
- INTENDED USER 1-1
- INSTRUCTION CONTENT 1-1
- HOW TO USE THIS MANUAL 1-2
- COMPATIBILITY 1-2
- GLOSSARY OF TERMS AND ABBREVIATIONS 1-2
- REFERENCE DOCUMENTS 1-3
- CONVENTIONS 1-4
- REQUIREMENTS 1-4

SECTION 2 - INSTALLATION 2-1
- INTRODUCTION 2-1
- REQUIREMENTS 2-1
- INSTALLING THE SOFTWARE 2-1

SECTION 3 - USING LADDER 3-1
- INTRODUCTION 3-1
- STARTING THE LADDER PROGRAM 3-1
- MENU STRUCTURE AND OPERATION 3-3
- GENERAL OPERATING PROCEDURE 3-4
- EXITING THE LADDER PROGRAM 3-4

SECTION 4 - MONITORING A PROGRAM 4-1
- INTRODUCTION 4-1
- MONITORING LOGIC RUNGS 4-2
- MONITORING FUNCTION BLOCK RUNGS 4-5
- LISTS 4-7
- QUITTING THE MONITOR 4-8

SECTION 5 - CREATING OR MODIFYING A PROGRAM 5-1
- INTRODUCTION 5-1
- THE HELP DISPLAY 5-1
- LOGIC RUNGS AND FUNCTION BLOCK RUNGS 5-3
  - Logic Rungs 5-3
    - The Heading 5-4
    - Specifications: The Logic Circuit 5-4
    - Output: The Coil 5-5
    - The Comment Area 5-6
  - Function Block Rungs 5-6
    - The Heading 5-7
    - Specifications: The Input and Function Code 5-7
    - Output 5-8
    - The Comment Area 5-8
    - Working with External Data: I/O Function Blocks 5-8
- EDITOR COMMANDS 5-9
  - Block 5-10
  - Compile 5-12
  - Copy 5-14
  - Delete 5-15
  - Display 5-16

Table of Contents (continued)

Page

SECTION 5 - CREATING OR MODIFYING A PROGRAM (continued)
- Done 5-18
- Error List 5-19
- Function Code 5-20
- Insert 5-21
- Insert File 5-22
- Mark 5-23
- Save 5-24
- Tags 5-25
- Title 5-26
- Xref 5-27
- Xtag 5-28
- Modifying Rungs 5-29

SECTION 6 - DOWNLOADING A LADDER PROGRAM 6-1
- INTRODUCTION 6-1
- INTERFACING TO THE INFI 90 MODULE 6-1
- MODULE FUNCTIONS 6-3

SECTION 7 - PRINTING A LADDER PROGRAM 7-1
- INTRODUCTION 7-1
- SELECTING A FILE 7-1
- SETTING THE PORT SPECIFICATIONS 7-3
- TESTING THE PORT 7-4

SECTION 8 - LISTING THE DIRECTORY 8-1
- INTRODUCTION 8-1

SECTION 9 - CONFIGURING THE INFI 90 INTERFACE 9-1
- INTRODUCTION 9-1

APPENDIX A - TUTORIAL A-1
- INTRODUCTION A-1
- EDITOR A-2
  - Logic Rungs A-5
    - Heading A-6
    - Coil A-7
    - Logic Circuit A-8
    - Comment Area A-10
  - Function Block Rung A-11
  - I/O Function Blocks A-14
  - Creating New Rungs A-17
  - Displaying Rungs A-18
  - Deleting Rungs A-19
  - Save Ladder Program A-20
  - Function Block Language A-21
  - Compile a Ladder Program A-22
  - Quit the Editor A-23
- DOWNLOAD A PROGRAM A-24
- MONITOR A-26
  - Logic Rung A-29

Table of Contents (continued)

Page

APPENDIX A - TUTORIAL (continued)
Function Block Rung A-34
Lists A-38
Additional Practice A-40
Quit the Monitor A-40
PRINT A-40
INFI 90 INTERFACE DATA A-46

APPENDIX B - SUMMARY OF EDITOR COMMANDS B-1
EDITOR COMMANDS B-1

APPENDIX C - EDITOR ERROR CODES C-1
ERROR CODE LISTING C-1

APPENDIX D - LOGIC CIRCUITS D-1
LOGIC CIRCUIT DESCRIPTION D-1

APPENDIX E - REPLY CODES E-1
REPLY CODE LISTING E-1
Greater Than 100 ..... E-1
Less Than 100 ..... E-1

APPENDIX F - PROBLEM REPORTS F-1
PROBLEM REPORT LISTING F-1

List of Figures

No.	Title	Page
3-1.	The LADDER Main Menu Screen	3-3
4-1.	Monitor Opening Screen	4-1
4-2.	Tag List	4-2
4-3.	Example Monitor Rung Screen	4-3
4-4.	X-Reference Screen	4-4
4-5.	Force Screen	4-4
5-1.	Off-Line Editor Title Page	5-2
5-2.	Title Page with Help Display	5-2
5-3.	Logic Rung Example	5-4
5-4.	Off-Line Editor Comment Area Display	5-6
5-5.	Example Function Block Rung	5-7
5-6.	Compilation Error List Display	5-12
5-7.	List of Rungs with Same Function Code	5-20
5-8.	Tag List Screen	5-25
5-9.	Tag References Screen	5-27
5-10.	Modify Tag Screen	5-28
6-1.	INFI 90 Interface Data Screen	6-2
6-2.	Data File to Download	6-3

List of Figures (continued)

No.	Title	Page
6-3.	Data File Title Page/Module Functions Screen	6-4
7-1.	Printer Menu	7-1
7-2.	Print Specification Screen	7-2
7-3.	Specify Printer Interface Screen	7-3
8-1.	Directory File Specification Screen	8-1
8-2.	Example Directory Listing	8-2
8-3.	Example Read File Screen	8-2
9-1.	INFI 90 Interface Data Screen	9-1
A-1.	LADDER Main Menu	A-1
A-2.	Off-Line Editor Opening Screen	A-2
A-3.	Off-Line Editor Title Page	A-3
A-4.	Title Page with Help Display	A-4
A-5.	Logic Rung Sample Screen	A-5
A-6.	Comment Screen	A-6
A-7.	Display a New Contact in Series with RL-CC	A-9
A-8.	Function Block Rung Example	A-11
A-9.	Function Block Rung Display with More	A-13
A-10.	Input Module	A-15
A-11.	Output Module	A-16
A-12.	Compilation Error List Display Example	A-23
A-13.	Data File to Download	A-25
A-14.	Module Status Screen in Configure Mode	A-25
A-15.	Monitor File Name	A-27
A-16.	Monitor Module Status Screen	A-28
A-17.	Tag List Screen	A-28
A-18.	Monitor Sample Rung Screen	A-29
A-19.	Monitor X-Reference Screen	A-31
A-20.	Monitor Force Screen	A-32
A-21.	Monitor Function Block Rung (More) Screen	A-34
A-22.	Monitor Function Block Rung (End) Screen	A-35
A-23.	Monitor Descriptor Screen	A-36
A-24.	Forced Rung List	A-38
A-25.	List of Rungs with Same Function Codes	A-39
A-26.	Printer Menu	A-41
A-27.	Specify Printer Interface Screen	A-42
A-28.	Print Specification Screen	A-43
A-29.	INFI 90 Interface Data Screen	A-47
D-1.	Circuit Samples and Equations	D-2
D-2.	Circuit Sample	D-3

List of Tables

No.	Title	Page
1-1.	LADDER Module Support	1-2
1-2.	Glossary of Terms and Abbreviations	1-3
1-3.	Reference Documents	1-4
2-1.	LADDER Files	2-1

List of Tables (continued)

No.	Title	Page
3-1.	LADDER Start-up Parameters	3-1
5-1.	Contact Type Descriptions	5-5
5-2.	Text Editing Key Commands	5-29
5-3.	Modify Logic Rung Contact Key Commands	5-29
5-4.	Modify Logic Rung Coil Key Commands	5-30
5-5.	Modify Function Block Rung Inputs or Parameters Key Commands	5-30
5-6.	Modify Function Block Outputs Key Commands	5-30
6-1.	INFI 90 Interface Data	6-2
7-1.	File Print Specifications	7-2
7-2.	Printer Port Specifications	7-3
B-1.	Editor Commands	B-1
B-2.	Text Editing Commands	B-2
B-3.	Modify Logic Rung Contact Key Commands	B-2
B-4.	Modify Logic Rung Coil Key Commands	B-3
B-5.	Modify Function Block Rung Inputs or Parameters Key Commands	B-3
B-6.	Modify Function Block Outputs Key Commands	B-4
C-1.	Editor Error Codes	C-1
D-1.	Contact Type Descriptions	D-3
E-1.	Computer Interface Reply Codes	E-1
E-2.	Station Mode Settings	E-7
F-1.	Module and Extended Module Problem Reports	F-1

SECTION 1 - INTRODUCTION

OVERVIEW

The LADDER Logic Configuration Utility facilitates configuring INFI 90 processor modules using ladder logic diagrams. Traditional logic rungs for contacts and coils can be integrated with function code algorithms using an editor utility to create or modify ladder programs. A monitor utility is used to monitor and tune the configuration after it is downloaded to the module. Other utilities are included to configure the communications interface and to print the ladder program.

> **NOTE:** When used uppercase, the term LADDER refers to this software package (the LADDER Logic Configuration Utility). When used lower-case, the term ladder refers to the output of this software package (a ladder logic configuration file to be downloaded to an INFI 90 processor module).

INTENDED USER

This manual is intended for users of INFI 90 ladder programming who are familiar with ladder logic, the INFI 90 Strategic Process Management System, and PC-DOS based computers.

INSTRUCTION CONTENT

This manual contains nine sections and six appendices:

Introduction	Provides an overview of LADDER and general information.
Installation	Provides the installation procedure.
Using Ladder	Explains how to start and exit LADDER and describes the menu structure.
Monitoring a Program	Describes how to monitor and tune a ladder program.
Creating or Modifying a Program	Describes how to create and modify a ladder program.
Downloading a Ladder Program	Describes how to transfer ladder programs from the computer to a processor module.
Printing a Ladder Program	Describes how to print a listing of the ladder program.
Listing the Directory	Describes how to view a listing of the directory.
Configuring the INFI 90 Interface	Describes how to configure the computer communication interface.

Tutorial	Provides a detailed tutorial for using LADDER.
Summary of Editor Commands	Provides a quick reference guide for the editor commands.
Editor Error Codes	Gives a listing of editor errors.
Logic Circuits	Provides brief description of logic circuits.
Reply Codes	Provides a list of the interface unit reply codes.
Problem Reports	Provides a list of the problem report descriptions.

HOW TO USE THIS MANUAL

Read Section 1 for an overview of LADDER and this instruction. Proceed to Section 2 for information on installation. Refer to Section 3 for information on starting LADDER and the LADDER menu structure. The remaining sections provide procedures for using each utility. Appendix A provides a detailed tutorial using the included ***PCDEM.LAD*** ladder configuration.

COMPATIBILITY

The LADDER Logic Configuration Utility release 3.0 is compatible with configurations generated by all earlier releases. It supports the module types listed in Table 1-1.

Table 1-1. LADDER Module Support

Module	Description
CBC01	Loop Command Batch Controller
CLC01, CLC02, CLC03, and CLC04	Loop Command Controller
CSC01	Sequence Command Controller
IMAMM02, IMAMM03	Analog Master Module
IMLMM02	Logic Master Module
IMCOM02A, IMCOM02B, IMCOM03, IMCOM03A, IMCOM04, and IMCOM04A	Controller Module
IMMFC01, IMMFC02, IMMFC03, IMMFC04, and IMMFC05	Multi-Function Controller Module
IMMFP01, IMMFP02, and IMMFP03	Multi-Function Processor Module
NGCM01, NGCM02	Gateway Communication Module

GLOSSARY OF TERMS AND ABBREVIATIONS

Table 1-2 lists the terms and abbreviations used in this manual.

Table 1-2. Glossary of Terms and Abbreviations

Term	Definition
Baud rate	Rate at which data is transmitted over a serial bus in bits per second.
CAD	Computer aided design. A Bailey Controls Company computerized design package that allows an engineer or technician to create function block control logic.
Coil	The output of a logic rung in a ladder program.
Compile	To translate high-level language statements into a form that can directly activate the device hardware.
Configure mode	Refers to the processor module mode used when downloading a ladder program.
Contact	An element making up a logic circuit in a ladder program. There are four types, represented by relay contacts: normally open, normally closed, leading edge, and trailing edge.
Execute mode	Refers to the processor module mode used when executing and tuning a ladder program.
Function block	The occurrence of a function code at a block address of a module.
Function block rung	A ladder program rung containing a function block.
Function code	An algorithm which manipulates specific functions. These functions are linked together to form the control strategy.
.LAD	Unique file extension for ladder programs.
LADDER	A program developed by Bailey Controls Company to configure a processor module via ladder logic rather than using the CAD utility.
Ladder logic	A method of configuring an INFI 90 processor module represented by a ladder logic diagram. The configuration is given by a network of contacts arranged in horizontal rows called rungs between two vertical rails on the side of the diagram containing the symbolic configuration.
Ladder program	An INFI 90 processor module configuration generated by LADDER.
Logic rung	A ladder program rung containing a logic circuit.
PCDEM.LAD	An example ladder program file included with the SLAD 3.0 release.
Processor module	A module capable of accepting and executing a ladder program.
Rung	One element of a ladder program. It either contains a logic circuit or a function block.
SLAD	Nomenclature for Bailey Controls Company LADDER Logic Configuration Utility software package.
SPM	Serial port module.
Tag	An analog or digital process value defined as a control or monitor point.
Tune	Refers to changing the tunable parameters of a function block while the processor module is executing the ladder program.

REFERENCE DOCUMENTS

Table 1-3 lists documents useful when working with LADDER. Refer to the ***Function Code Application Manual*** and the appropriate module manual for your installation.

Table 1-3. Reference Documents

Document Number	Description
I-E92-500-3	CBC01 Batch Command Controller Instruction
I-E92-500-4	CSC01 Sequence Command Controller Instruction
I-E92-500-7	CLC03/04 Loop Command Controller Instruction
I-E96-200	Function Code Application Manual
I-E96-201	IMMFP01 Multi-Function Processor Module Instruction
I-E96-202	IMMFP02 Multi-Function Processor Module Instruction
I-E96-203	IMMFP03 Multi-Function Processor Module Instruction
I-E96-205	IMAMM03 Analog Master Module Instruction
I-E96-207	IMCOM03/04 Controller Module Instruction
I-E96-209	IMLMM02 Logic Master Module Instruction
I-E96-211	IMMFC03 Multi-Function Controller Module Instruction
I-E96-212	IMMFC04 Multi-Function Controller Module Instruction
I-E96-213	IMMFC05 Multi-Function Controller Module Instruction

CONVENTIONS

Menu selections, user input fields, and system messages are shown in *Mixed case italic type*.

Exact user keyboard entries are shown in **Mixed case bold type**. User supplied entries are shown in *italic type*.

Command lines are shown in **bold type**. Required command parameters are shown in *italic type*. Optional command parameters are shown [*bracketed*]. Example: **command** *parameter1* [*parameter2*].

File names in text appear in ***mixed case bold-italic type***.

Keyboard keys are shown as KEY. KEY-KEY means to press and hold the first key, then press the second key.

REQUIREMENTS

The computer must be an IBM AT® or compatible and have the following minimum configuration:

- DOS (disk operating system) version 3.1 or later.
- 640 kilobyte RAM (random access memory).
- A monochrome or color monitor.
- One 5-1/4 inch or 3-1/2 inch floppy disk drive.

® Registered trademark of International Business Machines Corporation.

- One hard disk drive (ten megabyte minimum capacity).
- One RS-232-C serial port (to communicate with INFI 90 modules).

An Epson compatible printer is optional.

SECTION 2 - INSTALLATION

INTRODUCTION

This section provides system requirement and installation information.

REQUIREMENTS

The computer must be an IBM AT or compatible and have the following minimum configuration:

- DOS (disk operating system) version 3.1 or later.
- 640 kilobyte RAM (random access memory).
- A monochrome or color monitor.
- One 5-1/4 inch or 3-1/2 inch floppy disk drive.
- One hard disk drive (10 megabyte minimum capacity).
- One RS-232-C serial port (to communicate with INFI 90 modules).

An Epson compatible printer is optional.

INSTALLING THE SOFTWARE

1. Create a directory for the LADDER software. Type:

   MD LADDER Enter

   CD LADDER Enter

2. Copy the LADDER software from the floppy disks to the hard disk. Use the following command for each disk:

   COPY A:*.* Enter

Table 2-1 is a list of the files included on the distribution disk and copied to the ladder directory during installation:

Table 2-1. LADDER Files

File	Description
LAC.SDL	Data file used by LADDER for function code information.
LAD.BAT	A batch file used to run the LADDER program.

Table 2-1. LADDER Files (continued)

File	Description
LM2.SDL	Data file used by LADDER for function code information.
MDT.DAT	Data file used by LADDER for function code information.
MFC.SDL	Data file used by LADDER for function code information.
PCDEM.LAD	Demo configuration as shown in Appendix A.
PCDEMZZZ.LAD	A backup copy of ***PCDEM.LAD***.
QAMM.LAD	Default configuration file for an IMAMM03 module.
QCBC.LAD	Default configuration file for a CBC01 module.
QCLC.LAD	Default configuration file for CLC03 and CLC04 modules.
QCOM.LAD	Default configuration file for IMCOM03 and IMCOM04 modules.
QCSC.LAD	Default configuration file for a CSC01 module.
QGCM.LAD	Default configuration file for an INGCM01 module.
QLMM02.LAD	Default configuration file for an IMLMM02 module.
QMFC.LAD	Default configuration file for IMMFC05, IMMFP01, IMMFP02 and IMMFP03 modules.
SLAD.EXE	LADDER program - small model.
SLADX.EXE	LADDER program - large model (extended memory version).
SLAD30.DOC	An ASCII file containing SLAD 3.0 release notes.
STT.DAT	Data file used by LADDER for function code information.
TDF.DAT	Data file used by LADDER for function code information.

This completes LADDER software installation. Refer to Section 3 for information on starting LADDER.

SECTION 3 - USING LADDER

INTRODUCTION

This section provides information on starting, using, and exiting the LADDER Logic Configuration Utility (LADDER).

> **NOTE:** When used uppercase, the term LADDER refers to this software package (the LADDER Logic Configuration Utility). When used lower-case, the term ladder refers to the output of this software package (a ladder logic configuration file to be loaded into an INFI 90 module).

STARTING THE LADDER PROGRAM

The LADDER program is provided in two versions: A standard memory version and an extended memory version. The standard version of LADDER does not require extended memory and has a maximum of 3,000 rungs. The extended memory version requires extended memory in your computer. With three megabytes of extended memory, up to 10,000 rungs can be used.

The following command line is used to start the standard memory version of LADDER:

SLAD *[parameter1] . . . [parameter13]*

The following command line is used to start the extended memory version of LADDER:

SLADX *[parameter1] . . . [parameter13]*

Any combination of the parameters described in Table 3-1 can be used with either version.

Table 3-1. LADDER Start-up Parameters

Parameter	Default	Description
CM-OFF CM-ON	CM-ON	Screen color mode switch. Use CM-OFF for monochrome or plasma monitors.
CP-OFF CP-ON	CP-OFF	Compressed printing switch. Use CP-ON to print on 8.5x11-inch paper on an Epson compatible printer. The default requires 11x17-inch paper.
IB-*nnnnn*	IB-9600	Interface baud rate[1]: 300, 1200, 2400, 9600, 19200.
IC-COM1 IC-COM2	IC-COM1	Interface serial port connection[1].
IL-*nnn*	IL-0	Interface loop number[1]: 0 to 250.

Table 3-1. LADDER Start-up Parameters (continued)

Parameter	Default	Description
IT-NONE IT-SPM IT-CIU01 IT-CIU02 IT-CIU03 IT-CIU04	IT-NONE	Interface type[1]. The following are equivalent: CIU01 - INPCI01 CIU03 - INPCI02 CIU04 - INICI01
NR-*nnnnn*	NR-2500	Number of rungs[2]: 700 to 10000. The default is the same for both the standard version and extended memory version. The maximum for the standard version is 3000. The maximum extended version is 10000.
OS-DOS OS-QNX	OS-DOS	Operating system.
PB-*nnnnn*	PB-9600	Printer baud rate: 300, 1200, 2400, 9600.
PC-COM1 PC-COM2 PC-LPT1	PC-LPT1	Printer port connection.
PD-7 PD-8	PD-8	Printer port number of data bits.
PP-NONE PP-ODD PP-EVEN	PP-NONE	Printer port parity.
PS-1 PS-2	PS-1	Printer port number of stop bits.

NOTES:

1. The program will attempt to initialize the interface device on program start-up if the **IB**, **IC**, **IL**, and **IT** parameters are all given.
2. The number of rungs that can be used is dependent on the amount of free system memory. In order to allocate for more than 2,500 rungs, more than 585,000 bytes of memory must be available. This is possible if using PC-DOS 5.0, in which case up to 615,000 bytes can be made available.

Example 1 To start LADDER using a color monitor, 11×17-inch paper, and 2,500 rungs (all the default values), type:

SLAD Enter

Example 2 To start the LADDER using a monochrome monitor, 8.5×11-inch paper, and 5,000 rungs, type:

SLADX CM-OFF PC-ON NR-5000 Enter

Example 3 To start LADDER and initialize the interface, type:

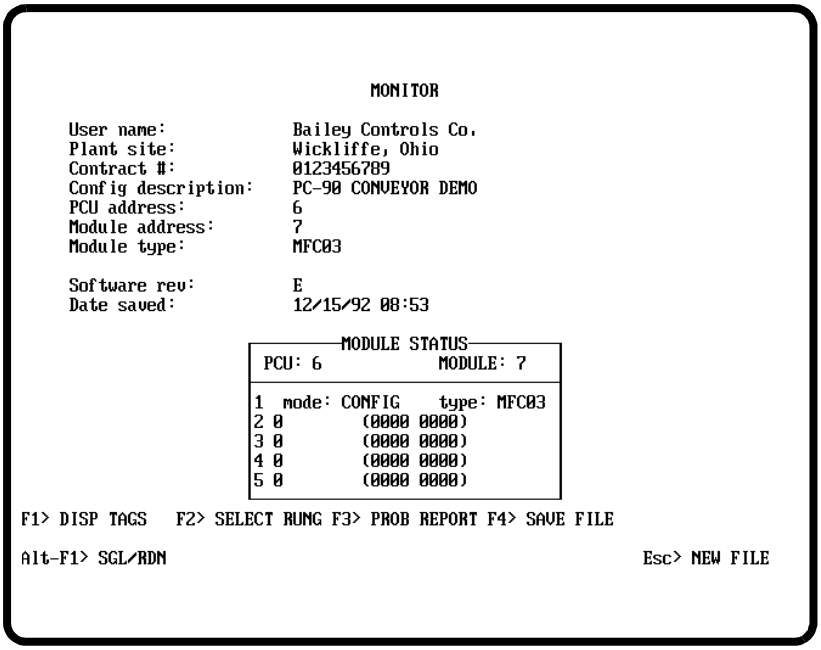
SLAD IB-19200 IC-COM2 IL-1 IT-CIU04 Enter

Example 4 LADDER supports pathing to program data files. Therefore, the program does not need to be started in the directory where the LADDER program and its data files reside. To start LADDER from any directory, specify the full path name:

C:\LADDER\SLAD Enter

MENU STRUCTURE AND OPERATION

When the LADDER program is started, the main menu screen appears (Figure 3-1). Menu items appear across the bottom lines of the menu. Select a menu item by pressing the appropriate function key.



TPS0778A

Figure 3-1. The LADDER Main Menu Screen

F1> Monitor Monitor the operation of an INFI 90 module as it runs a ladder program.

F2> Edit Create a new ladder program or modify an existing one.

F3> Config Download (i.e., transfer) a ladder program from the LADDER computer to an INFI 90 module.

F4> Print Print a listing of a ladder program.

F5> Directory Display the file names and their creation dates located in a specified directory.

F6> CIU/SPM Specify the type of interface between the LADDER computer and the INFI 90 module.

Alt-F10> Exit Exit to the DOS prompt.

Submenus appear upon making a selection from the main menu. In some cases, the submenu items do not fit across the bottom line of the screen. This is indicated by the character + in the far right column of the bottom line of the screen. To see

more submenu items, press Tab or +. The remaining menu items appear in place of the current menu items.

GENERAL OPERATING PROCEDURE

The following is an outline procedure for using LADDER:

1. Use *EDIT* to create or modify a ladder program.

2. Run the editor command **COMPILE** to compile the ***.LAD*** file into a ***.CFG*** file that can be downloaded to the target module. This also checks for errors in the ladder program. If necessary, correct any errors and run **COMPILE** again.

3. Use *CIU/SPM* to configure the interface to the target module.

4. Use *CONFIG* to download the ***.CFG*** file to the target module.

5. Use *MONITOR* to monitor and tune the process.

Refer to the remaining sections of this instruction for information on using the LADDER utilities. Refer to Appendix A for a detailed tutorial on using the LADDER software.

EXITING THE LADDER PROGRAM

To exit LADDER, return to the main menu (Figure 3-1) and press Alt-F10.

SECTION 4 - MONITORING A PROGRAM

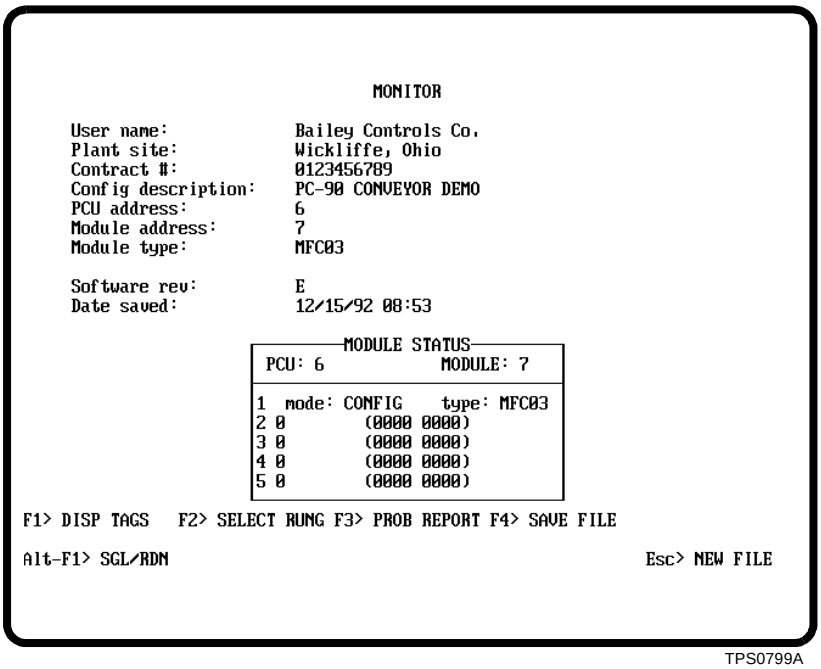
INTRODUCTION

The LADDER monitor utility serves two purposes:

- It observes the operation of an INFI 90 processor module as it runs a ladder program. The processor module actually runs a function block program which is translated from the ladder program. What appears on the screen is the ladder program with live data from the function block program in the processor module.

- It allows certain changes to be made to a program as it runs. Tunable parameters of function block rungs can be changed and contacts and coils of logic rungs can be forced.

Press F1 at the main menu to start the monitor. If the system interface data and the file name have not been entered, the system prompts for them. Figure 4-1 shows the monitor opening screen.



TPS0799A

Figure 4-1. Monitor Opening Screen

F1> Disp Tags Displays a list of all the tag names defined in the ladder program.

F2> Select Rung Permits selecting a rung for monitoring.

F3> Prob Report Displays module problem reports. Refer to Table F-1 for descriptions of the problem reports.

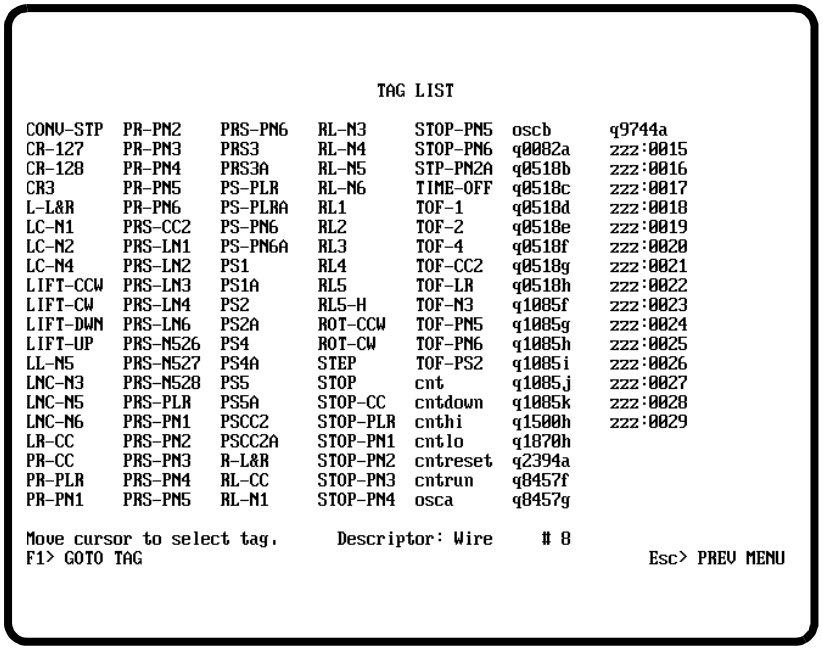
F4> Save File Saves changes made to the ladder program while in the monitor (by tuning or forcing) to the disk.

Alt-F1> SGL/RDN Permits monitoring of the module status of a single module and a redundant module.

Esc> New File Enables entry of a new ladder file name for monitoring.

MONITORING LOGIC RUNGS

Either *DISP TAGS* or *SELECT RUNG* can be used to select a rung for monitoring. Press F1 to display the tag list as shown in Figure 4-2. Select any rung by moving to the desired tag and pressing F1. The rung containing that tag appears on the screen. Once a rung is selected, a screen similar to Figure 4-3 appears.

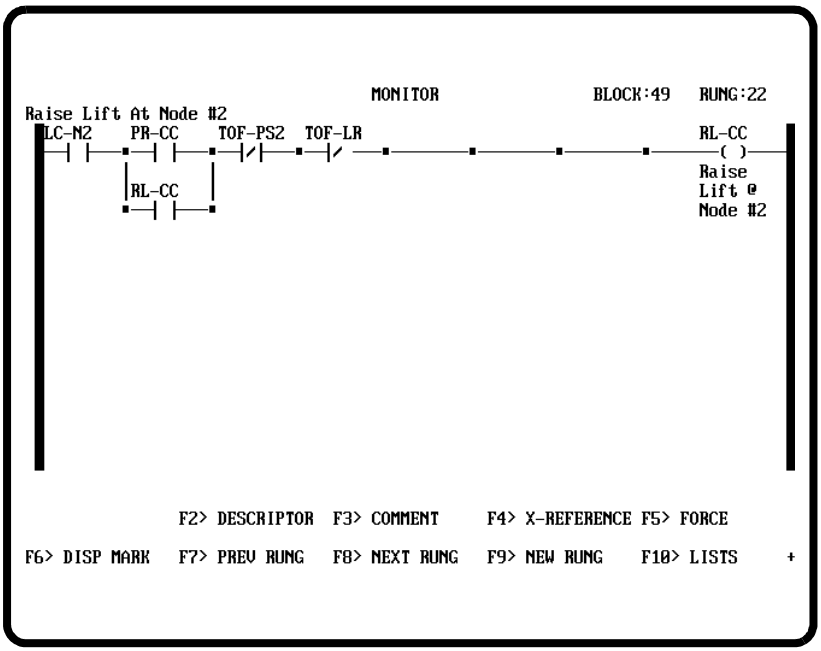


TPS0789A

Figure 4-2. Tag List

The value of each contact or coil is indicated in the space inside the symbol: dark equals zero, bright equals one. A contact represents a reference to an item of data (a coil or function block output) and, depending upon the type of coil, an operation to be performed on that item of data.

If an input or output symbol is shown in reverse video (dark symbol on light background), the corresponding value is not



TPS0790A

Figure 4-3. Example Monitor Rung Screen

being read from the module. If all the values are reversed, it can mean that the processor module is not in execute mode.

The monitor menu has a **+** character on the right side, indicating more menu items are available. To see the other menu items, press **Tab** or **+**.

For logic rungs, the menu items function as follows:

F2> Descriptor This changes the display so that each contact and the coil are replaced by their descriptions. A checkerboard pattern appears to help identify the grid positions. **F1** returns to the logic rung display.

F3> Comment This displays the comment area.

F4> X-Reference This finds all the rungs that refer to the coil of the displayed rung (cross reference). After pressing **F4**, the menu shown in Figure 4-4 appears. The following keys can be used:

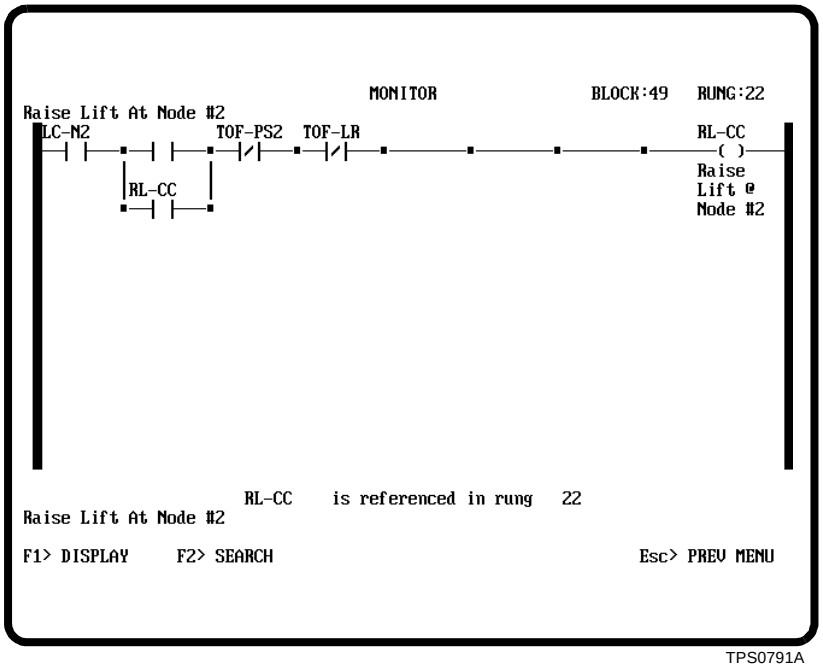
F1 Displays the currently identified rung.

F2 Searches for the next rung that refers to the tag. It looks for the next rung each time **F2** is pressed.

Esc Terminates the cross reference search.

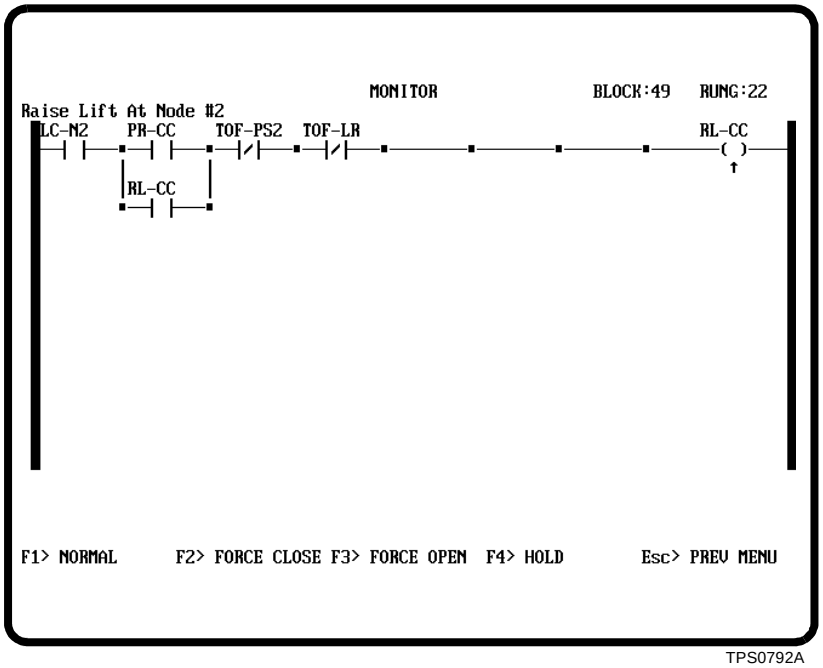
F5> Force This substitutes a constant value in place of the normal dynamic value of a contact or a coil.

1. Press F5 to enter the force mode. The screen now appears as shown in Figure 4-5. Initially the cursor points to the coil.



TPS0791A

Figure 4-4. X-Reference Screen



TPS0792A

Figure 4-5. Force Screen

2. Move the cursor using the cursor control keys. The function keys now perform the following operations:

F2	Forces the contact or coil closed (value = 1) and highlights the inside of the contact or coil.
F3	Forces the contact or coil open (value = 0) and unhighlights the inside of the contact or coil.
F4	Holds the coil at the current value. F4 applies only to a coil.
F1	Returns to normal.

NOTE: The tag name associated with the contact of coil appears bright when a force or hold condition is in effect.

3. Press Esc to leave the force mode.

F6> Disp Mark This function displays the list of marked rungs. The list is arranged in a 10 × 10 grid. The display shows only a 2 × 5 portion of the list. The entire list may be viewed by scrolling the display using any of the following keys: The four cursor keys, Page Up, Page Dn, Home, End, Ctrl - →, and Ctrl - ←

F7> Prev Rung Displays the previous rung.

F8> Next Rung Displays the next rung.

F9> New Rung Selects any rung for display. Specify the rung by its rung number, one of its output tags, or its block number (block number preceded by #, e.g., #501).

F10> Lists Refer to ***LISTS*** in this section.

M> Rung and G> Go To Mark The mark command (**M**) allows up to 100 rungs to be marked for easy recall using go to mark (**G**).

T> Trace Input and 0> Trace Out These two commands provide a convenient method of tracing the flow of data into a rung. **T** steps from the current rung to any one of its inputs. Successive use of the **T** command traces out one particular path of data into the rung.

Once a trace path has been established with the **T** command, the **O** command allows a trace back (i.e., output to input) along the path one step at a time. To use the command, display a rung, press **T**, then follow the screen prompts.

MONITORING FUNCTION BLOCK RUNGS

As with the logic rung, the status (good or bad) of an input or output is indicated by the normal or reverse video attribute of the input or output symbol. Normal video indicates that the

status is good (i.e., the value is being read from the processor module). Reverse video indicates that the status is bad (i.e., the value is not being read from the module).

The value of an input or output is indicated in the space inside the symbol. Boolean input and output values are indicated in the same way as contacts and coils: Dark indicates zero and light indicates one. For inputs and outputs with data type real, the numeric value of the input or output is displayed inside the symbol.

On each function block rung there are two more or end indicators on the display, one highlighted and one normal. The highlighted one indicates which side may be scrolled up or down (use ↑ and ↓ or Page Up and Page Dn. To move the highlighting from one side to the other, use ← and →

The menu for function block rungs is very similar to the menu for the logic rungs. The operations of the function keys are identical, except for these differences:

F2> Descriptor As for the logic rung, this function displays the description associated with each input. In this case, however, the input description is displayed in place of the input name. F1 returns to the function block rung display.

F3> Comment Identical to logic rung operation.

F4> X-Reference This is also very similar to cross reference for logic rungs. The only difference is that a logic rung has only one output, while a function block may have more than one. For a function block, the cross reference refers to the output that is currently visible at the top of the display.

F5> Tune This function allows changes to the values of tunable parameters of a function block. Tuning is similar to editing, but differs in two important ways:

- During editing, any parameter or input may be changed. During tuning, only tunable parameters may be changed.

  NOTE: When the monitor is in tune mode, it does not read data from the processor module. Instead, it displays *0* and *bad* status for all the values. The values are displayed normally when the tune mode is exited.

- Editing is done off-line. Changes made to the ladder program affect only the version of the program that is in the personal computer. Download the edited program to make the changes in the processor module. Tuning is done on-line. The tuning changes are made not only to the version of the ladder program that is in the computer, but also to the version of the program that is in the processor

module. The changes are made to the program as it is being run by the module.

F7> Prev Rung Displays the previous rung.

F8> Next Rung Displays the next rung.

F9> New Rung Selects any rung for display. Specify the rung by its rung number, one of its output tags, or its block number (block number preceded by #, e.g., #501).

F10> Lists Refer to ***LISTS*** in this section.

M> Rung and G> Go To Mark Mark rung (**M**) allows up to 100 rungs to be marked for easy recall using go to mark (**G**).

T> Trace Input and 0> Trace Out These two commands provide a convenient method of tracing the flow of data into a rung. **T** steps from the current rung to any one of its inputs. Successive use of the **T** command traces out one particular path of data into the rung.

Once a trace path has been established with the **T** command, the **O** command allows a trace back (i.e., output to input) along the path one step at a time. To use the command, display a rung, press **T**, then follow the screen prompts.

LISTS

This item is discussed separately because it pertains not just to an individual rung, but rather to the entire program. Press F10 and the following menu appears:

F1> All Tags
F2> Forced
F3> Func Code
Esc> Prev Menu

F1> All Tags This function displays the tag list.

F2> Forced This function displays the forced rung list. This is a list of rung numbers of all the rungs which have contacts or coils that are forced.

1. Press F2 to view the list of forced rungs. If no rungs are forced, the *End of List* message appears.

2. If the list is too long to fit on one screen, press F2 to scroll through the list.

3. To select a rung for display, press F1.

4. To terminate the display and return to the menu, press Esc.

F3> Func Code This function displays a list of all the rungs that have a specified function code.

QUITTING THE MONITOR

To quit the monitor utility, press Esc one or more times to return to the main menu.

SECTION 5 - CREATING OR MODIFYING A PROGRAM

INTRODUCTION

The program editor utility is used to create or modify a ladder program. The editor operates off-line, meaning it can be used without having a target module (the INFI 90 module that will run the program) connected to the computer.

To create or modify a ladder program, select *EDIT* from the LADDER main menu. A prompt appears requesting an optional path and a file name. Enter the path name if the file is not in the current working directory. Enter the name of a ladder program file. The file name has two parts consisting of an eight-character name and a three-character extension. The extension is always ***.LAD*** for ladder programs. Enter only the first part of the name, not the extension.

For example, if the ladder program is in the current working directory, type:

file_name Enter

If the ladder program is not in the current working directory, type:

drive_spec:\path\file_name Enter

where:

file_name	Name of the ladder program without the extension.
drive_spec	Drive (A, B, C, etc.).
path	Complete path to the file.

NOTE: The file name plus path is limited to 64 characters.

After the editor reads the file, the title page of the file appears (Figure 5-1). The title page provides general information about the program. Although not shown in Figure 5-1, a highlighted area appearing at the bottom of the screen is called the command line. A blinking cursor to the left of the command line indicates that the program is waiting for a command to be entered.

THE HELP DISPLAY

F9 activates the editor help display as shown in Figure 5-2. The information displayed depends on what the editor is doing

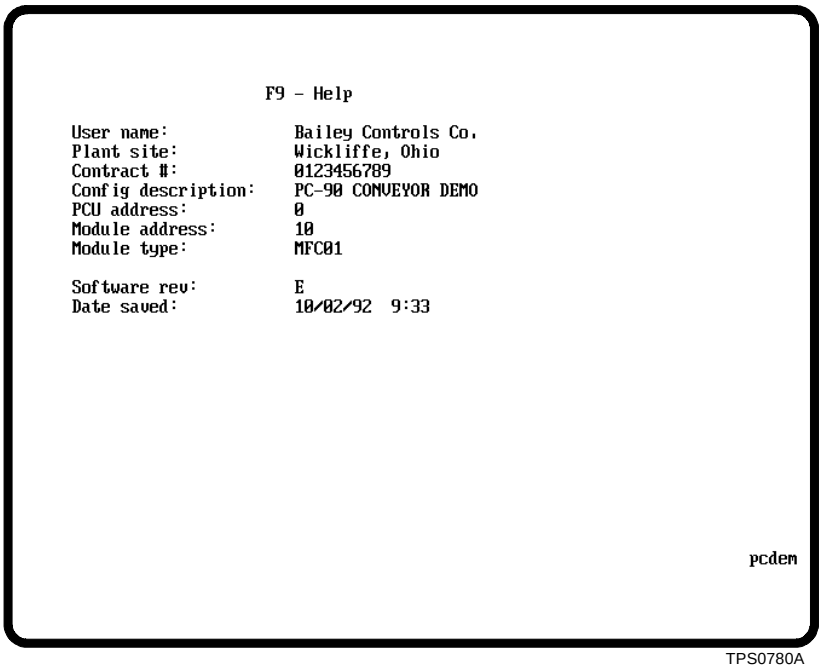


Figure 5-1. Off-Line Editor Title Page

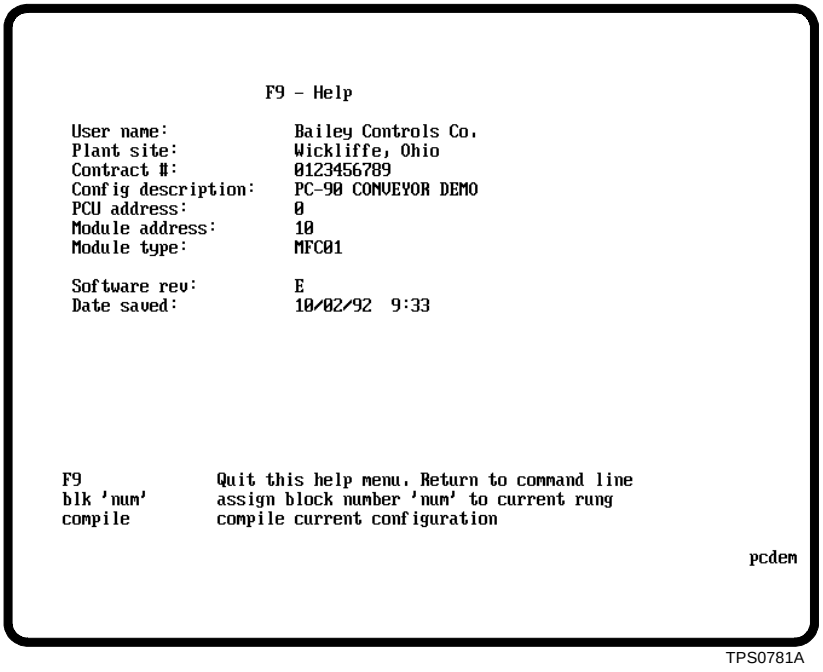


Figure 5-2. Title Page with Help Display

when help is requested. With the cursor on the command line, the help display provides a list of all the editor commands. Not all editor commands fit on the screen at once. To scroll through the list, use ↑, ↓, Page Up, or Page Dn. To leave the help display, press F9 again; the cursor returns to its original position. Appendix B contains a list of all the editor commands.

LOGIC RUNGS AND FUNCTION BLOCK RUNGS

A rung is the basic unit of a ladder program. There are two types of rungs: Logic rungs and function block rungs. For the purposes

of editing and displaying, a rung is divided into four areas: heading, specifications, outputs, and comments.

Heading - This is one comment line (up to 80 characters) that appears at the top of the rung display. The heading provides a brief explanation of the rung.

Specifications - For a logic rung, this area specifies a logic equation in terms of a series-parallel combination of contacts. For a function block, this area specifies the inputs and parameters of the function block.

Outputs - For a logic rung, this area contains the coil (i.e., the output of the logic equation). For a function block rung, this area contains the outputs of the function block.

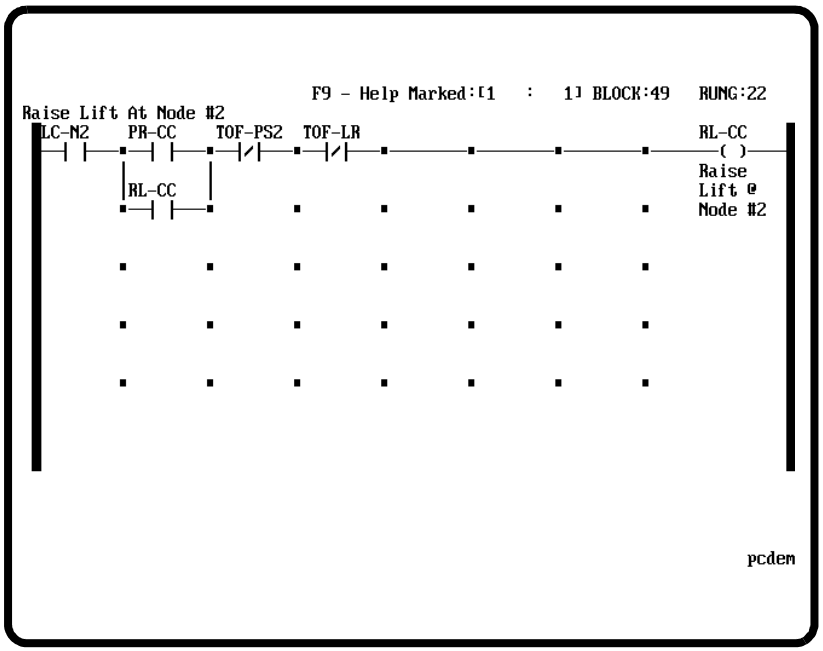
Comments - This is a free-form, multiline comment area that appears at the bottom of a rung on a printed listing or on a separate display screen. The comment area may contain any information that is relevant to the rung. The comment area is 77 characters wide by ten lines high.

Each rung is identified by a rung number in the upper right corner of a rung display. For example, if a ladder program consists of 100 rungs, the rungs are numbered one through 100. The rung numbers are assigned automatically by the editor as rungs are inserted and deleted. For example, assume a new rung is inserted at rung 50. The current rungs 50 through 100 are reassigned rung numbers 51 through 101.

Block numbers also appear in the rung display in the top center of the screen. On a color monitor, non-user assigned block numbers appear in red; user assigned block numbers appear in gray. Block numbers assigned by the program are only assigned when the ladder program is compiled. Therefore, new rungs inserted before a compile have a block number of zero. The file must be saved after compiling to permanently save the block numbers.

Logic Rungs

Figure 5-3 shows an example of a logic rung (rung 22 of the demo program, ***pcdem.lad***). Its areas are described in the following subsections.



TPS0783A

Figure 5-3. Logic Rung Example

The following keys are used to move between and within areas:

F2	Moves the cursor from one area to the next in sequence.
F1	Reverses the cursor movement.
↑, ↓, ←, →	Moves the cursor within an area.

THE HEADING

The heading is a single line of text. In rung 22 the heading is *Raise Lift At Node #2*. It is helpful to use the heading to give a brief description of the rung; however, it may be left blank.

SPECIFICATIONS: THE LOGIC CIRCUIT

The logic circuit specifies the operation of the rung in terms of a series-parallel connection of contacts. The contact grid is ten contacts wide by five lines high. The maximum number of contacts per rung is typically 15 (depending on the logic complexity).

Each contact represents a Boolean variable and has two attributes that describe how the value of the contact is to be determined. The tag associated with a contact specifies a variable which is output by a rung, (i.e., the coil of a logic rung or an output of a function block rung). The contact type specifies

an operation to be performed on the variable. The type is indicated by the symbol of the contact as shown in Table 5-1.

Table 5-1. Contact Type Descriptions

SYMBOL	VALUE OF VARIABLE	VALUE OF CONTACT	COMMENT
	0 1	0 1	NORMALLY OPEN
	0 1	1 0	NORMALLY CLOSED
	0 TO 1 OTHERWISE	1 0	LEADING EDGE
	1 TO 0 OTHERWISE	1 0	TRAILING EDGE

TP55174A

Applying these ideas to the example of Figure 5-3, the logic circuit can be interpreted as:

(value of LC-N2)

and

((value of PR-CC) or (value of RL-CC))

and

(inverse of the value of TOF-PS2)

and

(inverse of the value of TOF-LR)

OUTPUT: THE COIL

The coil represents the output of the rung (the result of the logic circuit), and looks like: ——( )——. A logic rung can have only one coil. The coil (or the value it represents) is identified by a tag name, as are all inputs and outputs. For rung 22 the tag name for the coil is *RL-CC*.

Also associated with the coil is a description. In this case:

Raise
Lift @
Node #2

The description is a comment - a block of text eight characters wide by three lines long. It may be left blank. In general, use the description to help identify the coil or to explain its use.

THE COMMENT AREA

The comment area is a block of text 77 characters wide by ten lines high. It may be left blank. Figure 5-4 shows an example of the comment area display.

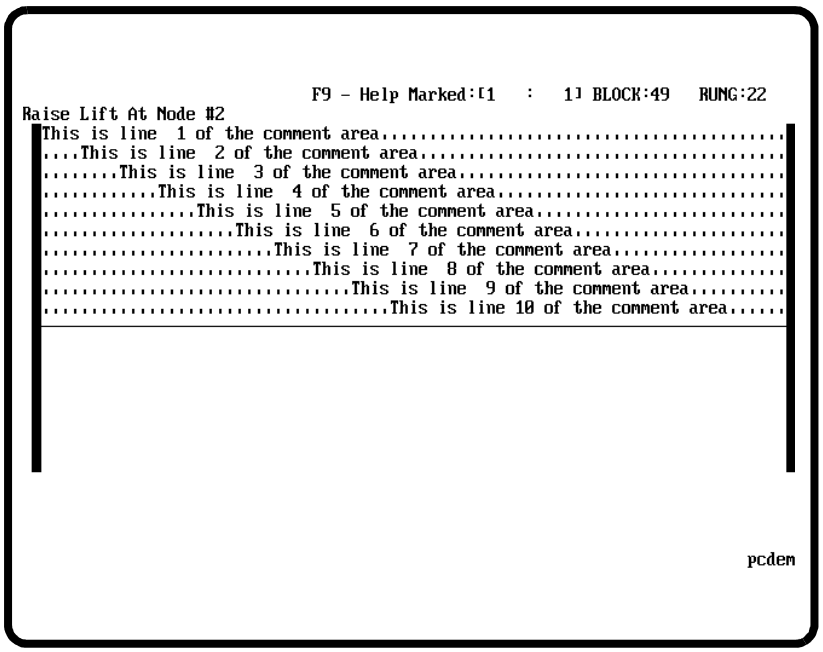


Figure 5-4. Off-Line Editor Comment Area Display

Function Block Rungs

A function block takes input values (outputs from other rungs), operates on those values as determined by its function code and a set of parameters, and produces output values. The output of a rung is an item of data. When a new rung is created, memory is set aside to hold this data value. The editor automatically keeps track of where data values are located in memory for each rung.

INFI 90 modules offer a wide variety of function blocks including such functions as timer, counter, drum programmer, limiter, comparator and PID controller. Refer to the ***Function Code Application Manual*** for a complete list of these functions.

Figure 5-5 shows a sample of a function block rung (rung eight from the ***PCDEM.LAD*** file). For this type of rung, the box in the middle of the display represents the function block. Inputs are to the left of the box, outputs to the right, and the function code and parameters are inside the box.

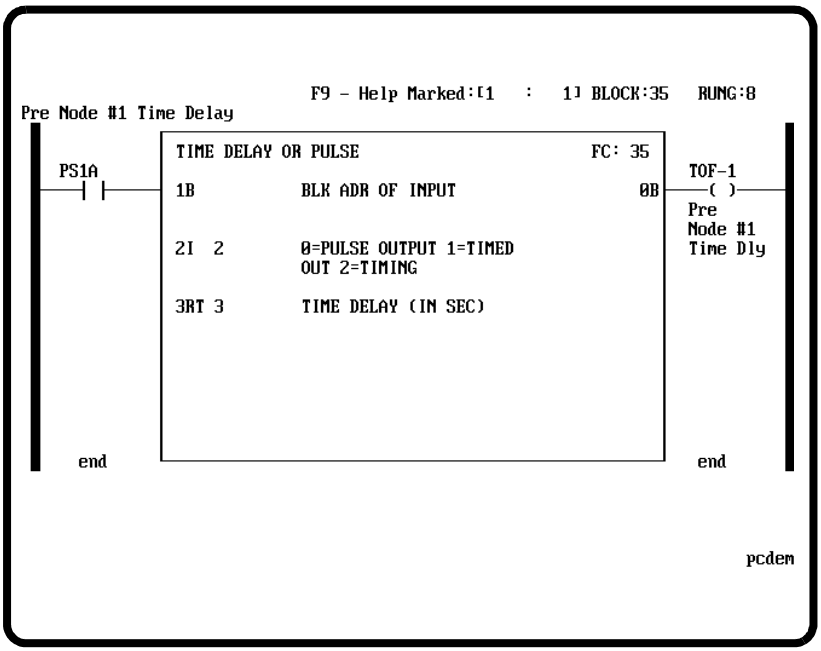


Figure 5-5. Example Function Block Rung

THE HEADING

The heading area is identical to that of a logic rung. It is a single line of text. In rung eight the heading is *Pre Node #1 Time Delay*. It is helpful to use the heading to give a brief description of the rung; however, it may be left blank.

SPECIFICATIONS: THE INPUT AND FUNCTION CODE

Rung eight has a Boolean input indicated by the symbol:

——| |——

to the left of the function code box. (An analog input is indicated by the symbol ——|| ||——.) The function code is FC 35, *TIME DELAY OR PULSE.*

An input to a function block rung has only one attribute - the tag. The tag identifies a variable which is the output of a rung (i.e., the coil of a logic rung or the output of a function block rung). The input variable can come from any rung. The only restriction is that the data types must agree (i.e., a Boolean input must refer to a Boolean output and a real input must refer to a real output).

The *1B* inside the function code box represents the specification number (*1*) and the input type (B = Boolean; R = real). *BLK ADR OF INPUT* is the specification name.

The next line indicates a parameter (notice that there is no symbol to the left of the box). The *2* is the specification number; *I* is the type (I = integer, R = real, B = Boolean). The *2* to the right indicates the value of the parameter, followed by the specification name.

On the next line, *3RT* represents the next parameter. The specification number is *3* and the *R* indicates its type is real. The *T* indicates a tunable parameter. This means the parameter may be tuned, or changed, while the program is being run in the execute mode. The value of the parameter is 3.0. As shown in the specification description, this parameter specifies the length, in seconds, of the time delay.

OUTPUT

To the right of the function code box is the Boolean output of the block, represented by the symbol ——( )——. This output is

identified by the tag *TOF-1*. Since the output is a unique item of data, this tag cannot be used again on the output (right) side of any rung. It could be used on the input (left) side of the rung as often as needed. This is the general rule for a tag; it may appear any number of times on the left side of rungs, but only once on the right side.

Inside the block at the output side is *0B*. The *0* indicates the output number (output numbering starts at 0); the *B* indicates the output type (Boolean). The output type is also indicated by its symbol:

—( )— Boolean.

—(( ))— real.

THE COMMENT AREA

The comment area is identical to that of a logic rung. It is a block of text 77 characters wide by ten lines high. It may be left blank. Figure 5-4 shows an example of the comment area display.

WORKING WITH EXTERNAL DATA: I/O FUNCTION BLOCKS

External data can come into a ladder program from a variety of sources. For example, process data may be read via special hardware modules called I/O modules, or data may be read from other INFI 90 modules via the module bus or plant loop.

Similarly, data can go to a variety of destinations. There is a specific function code for each type of I/O operation. For example, there is a function code to read a group of eight digital input values from a digital input module, another function

code to write a group of eight digital outputs to a digital output module, another to read a digital input over the plant loop, and so on.

EDITOR COMMANDS

The following subsections describe the editor commands.

NOTES:

1. Commands are shown in uppercase for clarity. However, they may be entered either in upper or lowercase.

2. Refer to Appendix B for a quick reference guide to the editor commands.

3. Refer to Appendix A for detailed usage of the editor commands.

BLOCK

PURPOSE: Assigns a block number to a rung for the function block language.

COMMAND: **BLK** *nnnn*

where:

nnnn The block number.

PROCEDURE:

1. Move the cursor to the command line by pressing F10
2. Type **BLK** *nnnn* Enter to assign block number *nnnn* to the current rung.

DISCUSSION: The rules and concepts discussed so far can be thought of as the ladder program language. For INFI 90 there is another concept which can be thought of as function block language. The languages are similar. In fact, there is a one-to-one correspondence between rungs in a ladder program and function blocks in a function block program. The primary difference between the two is how things are identified.

In a ladder program, each rung is identified by a rung number; inputs and outputs are identified by tag names. In a function block program each rung is identified by a block number. The outputs of a block are also identified by block number. For example, for block number 100, the first output is block number 100. If it has a second output, that output is numbered 101 and so on. Inputs also are identified by block number.

The language of INFI 90 modules is function blocks. LADDER automatically translates a ladder program into a function block program. For example, when a ladder program is downloaded to an INFI 90 module, LADDER translates rung numbers and tag names into block numbers.

There is one situation, however, where block numbers may be a concern. This has to do with a program in one module reading data from another module. There are several function codes that read data from another module. For example, function code 64 reads a list of digital values from another module in the same PCU, and function code 45 reads a digital input from a module in a different PCU.

In all of these function codes the source data item is identified by block number. Even if the source module or the receiving module is programmed in ladder language, the source data is identified by block number.

***BLOCK** (continued)*

Therefore, for a module that is a data source to other modules, it is necessary to know the block numbers of the blocks that are being read by other modules. Also, it is desirable to be able to set the block numbers of those blocks. The block number command assigns a block number to the current rung.

When a ladder program is translated into a function block program, block numbers are assigned in the following way: The rungs are scanned in order starting with rung 1. Each rung is given the next available block number or range of block numbers (if the block has multiple outputs).

Block numbers begin with zero. When using the **BLOCK** command to assign a block number, ensure that the block number is available.

COMPILE

PURPOSE: This command compiles the ladder program into a *.CFG* file which can be downloaded to the target processor module. It also analyzes the ladder program to find errors.

COMMAND: **COMPILE**

PROCEDURE:

1. Move the cursor to the command line by pressing F10
2. Type **COMPILE** Enter.

If no errors are found, the message *Compilation complete - no errors detected* appears. The file is compiled into a ***.CFG*** file of the same name. Otherwise, a list of errors is displayed. Figure 5-6 shows an example listing. If the error list is too long to display all at once, the display may be scrolled up and down with the cursor control and page keys.

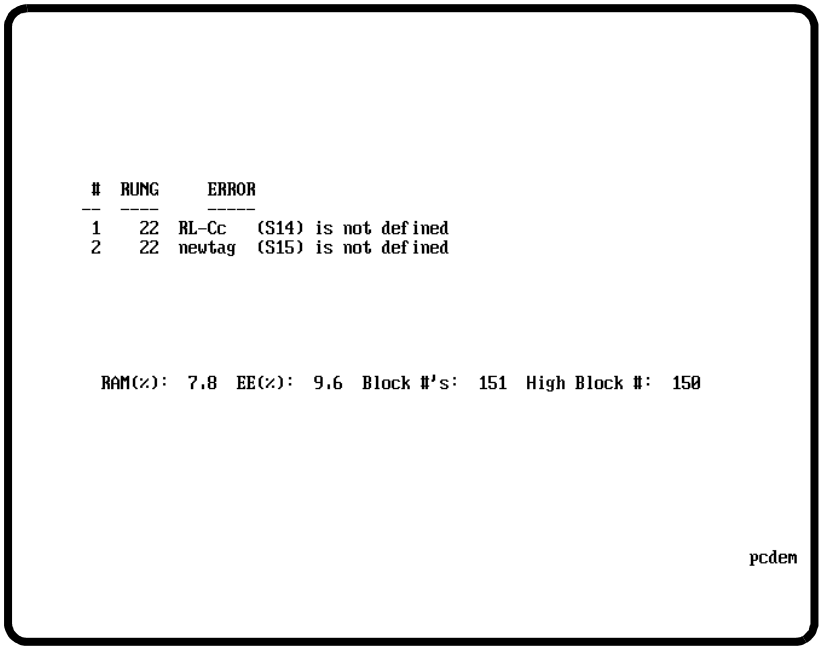


Figure 5-6. Compilation Error List Display

Normally, correct the errors one at a time. Use F10 to return to the command line, and use the display command to call up the rung with the error. After correcting the error, return to the error list again by using the command **ERR** and repeat the process until all errors are corrected.

***COMPILE** (continued)*

The following types of errors are found during compilation:

block # must be >>= num. An invalid block number was assigned to a rung (using the **BLOCK** command). The lowest available block number is *num*.

null tag. No tag has been assigned to the input of a rung.

tag is already defined. The same tag name is assigned to the output of two or more rungs.

tag is undefined. A tag is used as an input but is not defined (i.e., it is not assigned to the output of any rung).

tag is wrong data type. The tag is assigned to the output of a rung and to the input of a rung, but the data type of the output is not the same as the data type of the input.

COPY

PURPOSE: Copies a set of rungs.

COMMAND: **C** [*pointer*] [*pointer*]

where:

pointer One of the following pointers:

{ = First rung.

} = Last rung.

. = Current rung.

[= Mark.

] = Mark.

PROCEDURE:

1. Move the cursor to the command line by pressing F10

2. To copy rungs *pointer1* through *pointer2* starting at the current rung number, type the following:

   C *pointer1 pointer2* Enter

After copying the rungs, it renumbers all the following rungs.

Example: **C [ ]** Enter copies the rungs marked with the MARK command.

DELETE

PURPOSE: This command deletes a single rung or a set of rungs.

COMMAND: **DELETE** [*pointer*] [*pointer*]

where:

pointer One of the following pointers:

{ = First rung.

} = Last rung.

. = Current rung.

[= Mark.

] = Mark.

PROCEDURE:

1. Move the cursor to the command line by pressing **F10**
2. Type one of the following:

   DELETE **Enter** to delete the current rung.

   DELETE *pointer1* *pointer2* **Enter** to delete all rungs from *pointer1* to *pointer2*.

After deleting the rungs, it renumbers all the following rungs.

Examples: **DELETE { . Enter** deletes from the first rung to the current rung.

DELETE { } Enter deletes all rungs.

DELETE [ ] Enter deletes the marked rungs.

DISPLAY

PURPOSE: Selects a rung for display. There are four command types.

COMMAND: **D** [*num*] [*tag*] [*pointer*]

DB *nnnn*

N

P

where:

num A rung number.

tag A tag name.

pointer One of the following pointers:

{ = First rung.

} = Last rung.

. = Current rung.

[= Begin mark.

] = End mark.

nnnn = Block number.

nnnn The block number.

The following function keys operate as shortcut commands:

F7, Page Up Display previous rung (same as **P** Enter).

F8, Page Dn Display next rung (same as **N** Enter).

PROCEDURE:

1. Move the cursor to the command line by pressing F10
2. Type the command and press Enter.

Examples: **D 22** Enter displays rung number 22.

D RL-CC Enter displays the rung with an output named RL-CC.

***DISPLAY** (continued)*

D } Enter displays the last rung in the file.

DB 101 Enter displays the rung having a block number of 101.

DONE

PURPOSE: Quits the editor utility and returns to the editor utility opening screen.

COMMAND: **DONE [NOSAVE]**

where:

NOSAVE Quit without saving the file. The default is to save the file before quitting.

PROCEDURE:

1. Move the cursor to the command line by pressing F10.

2. Type one of the following:

   DONE Enter to save the file and quit.

   DONE NOSAVE Enter to quit without saving the file.

3. Either command returns to the editor utility opening screen, with the program asking for a file name. Specify another file to be edited by typing its name or leave the editor by pressing Esc. If Esc is pressed, the LADDER main menu appears.

ERROR LIST

PURPOSE: Displays a list of errors found during compilation.

COMMAND: **ERR**

PROCEDURE:

1. Move the cursor to the command line by pressing F10

2. Then type **ERR** Enter. The error list is displayed, similar to Figure 5-7.

During compilation, if errors are found the compile command generates and displays a list of errors. As errors are corrected this command can be used to recall the error list in order to view and correct remaining errors.

FUNCTION CODE

PURPOSE: Displays a list of all rungs with a specified function code.

COMMAND: **FC** *nnn*

where:

nnn A function code number.

PROCEDURE:

1. Move the cursor to the command line by pressing **F10**

2. To view a list of all rungs using function code 35, type **FC 35** **Enter**. A screen similar to Figure 5-7 appears.

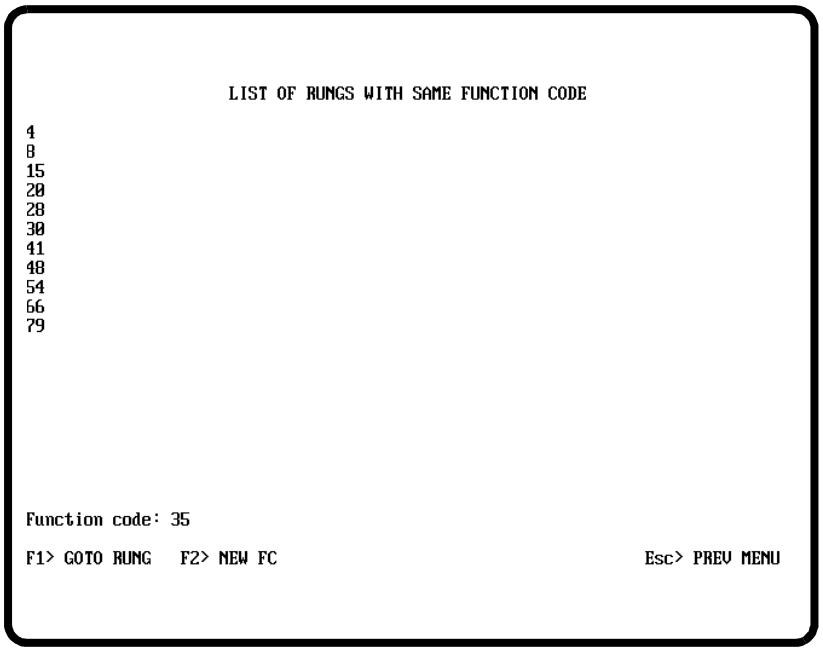


Figure 5-7. List of Rungs with Same Function Code

INSERT

PURPOSE: Creates and inserts a new rung.

COMMAND: **I** [*nnn*] [**A**]

where:

nnn	A function code rung with function code number *nnn*. The default is a logic rung.
A	Insert the new rung after the current rung. The default is to insert the new rung at the current rung.

PROCEDURE:

1. Move the cursor to the command line by pressing F10
2. Then type **I** [*nnn*] [**A**] Enter.

Examples:

I Enter inserts a logic rung at the current rung.

I A Enter inserts a logic rung after the current rung.

I 35 Enter inserts a function code 35 rung at the current rung.

I 35 A Enter inserts a function code 35 rung after the current rung.

All of these commands create and insert a new rung and adjust the rung numbers of subsequent rungs.

INSERT FILE

PURPOSE: Inserts a file at the current rung.

COMMAND: **I** *file_name*

where:

file_name The file to insert.

PROCEDURE:

1. Move the cursor to the command line by pressing F10
2. Then type **I** *file_name* Enter.

The file is inserted at the current rung. Rung numbers are adjusted accordingly.

MARK

PURPOSE:

The **MARK** command is used to mark a set of rungs for which to perform a particular operation. **[** marks the beginning of a set of rungs and **]** marks the end of a set of rungs. The mark command is useful to delete a series of rungs.

COMMAND:

M [

M]

PROCEDURE:

1. Display the rung that is to be the beginning of the marked set.

2. Type **M [** Enter.

3. Display the rung that is to be the end of the marked set.

4. Type **M]** Enter.

Example: Use the following command sequence to mark rungs 50 to 59:

D 50 Enter

M [Enter

D 59 Enter

M] Enter

To delete this marked set, type:

DELETE [ ] Enter

SAVE

PURPOSE: This command saves a ladder program to disk. It is recommended to save often to prevent data loss (for example, due to a power failure).

COMMAND: **SAVE** [*file_name*]

where:

file_name The name of the saved file. If the ladder program already has a name, it is not necessary to specify a file name. If the file already has a name, it can be saved with a new name to make a backup.

PROCEDURE:

1. Use F10 to move the cursor to the command line.
2. Type **SAVE** Enter.

Specifying a file name is commonly used to make a backup copy of the file. When a file is first opened and before it is changed, use the **SAVE** *file_name* command to make a backup copy. Then as the program is edited, use the **SAVE** command to periodically update the main file.

TAGS

PURPOSE: Displays a list of all tags.

COMMAND: **TAGS**

PROCEDURE:

1. Move the cursor to the command line by pressing F10

2. Then type **TAGS** Enter. A screen similar to Figure 5-8 appears.

```
                                    TAG LIST
CONV-STP  PR-PLR    PRS-PM2   PS5A      STEP      TOF-PM5   q0518h    zzz:0020
CR-127    PR-PM1    PRS-PM3   PSCC2     STOP      TOF-PM6   q1085f    zzz:0021
CR-128    PR-PM2    PRS-PM4   PSCC2A    STOP-CC   TOF-PS2   q1085g    zzz:0022
CR3       PR-PM3    PRS-PM5   R-L&R     STOP-PLR  cnt       q1085h    zzz:0023
L-L&R     PR-PM4    PRS-PM6   RL-CC     STOP-PM1  cntdown   q1085i    zzz:0024
LC-M1     PR-PM5    PRS3      RL-M1     STOP-PM2  cnthi     q1085j    zzz:0025
LC-M2     PR-PM6    PRS3A     RL-M3     STOP-PM3  cntlo     q1085k    zzz:0026
LC-M4     PRS-CC2   PS-PLR    RL-M4     STOP-PM4  cntreset  q1500h    zzz:0027
LIFT-CCW  PRS-LM1   PS-PLRA   RL-M5     STOP-PM5  cntrun    q1870h    zzz:0028
LIFT-CW   PRS-LM2   PS-PM6    RL-M6     STOP-PM6  osca      q2394a    zzz:0029
LIFT-DWN  PRS-LM3   PS-PM6A   RL1       STP-PM2A  oscb      q8457f
LIFT-UP   PRS-LM4   PS1       RL2       TIME-OFF  q0082a    q8457g
LL-M5     PRS-LM6   PS1A      RL3       TOF-1     q0518b    q9744a
LNC-M3    PRS-M526  PS2       RL4       TOF-2     q0518c    zzz:0015
LNC-M5    PRS-M527  PS2A      RL5       TOF-4     q0518d    zzz:0016
LNC-M6    PRS-M528  PS4       RL5-H     TOF-CC2   q0518e    zzz:0017
LR-CC     PRS-PLR   PS4A      ROT-CCW   TOF-LR    q0518f    zzz:0018
PR-CC     PRS-PM1   PS5       ROT-CW    TOF-M3    q0518g    zzz:0019

  136 Tags Total     Display first page(Ret)  or stop (F10)
```

TPS0811A

Figure 5-8. Tag List Screen

TITLE

PURPOSE: Displays the title page for editing.

COMMAND: **TITLE**

PROCEDURE:

1. Move the cursor to the command line by pressing F10

2. Then type **TITLE** Enter. The title page appears similar to Figure 5-1.

XREF

PURPOSE: Displays all references to a tag.

COMMAND: **XREF** [*tagname*]

where:

tagname Name of a tag.

PROCEDURE:

1. Move the cursor to the command line by pressing F10

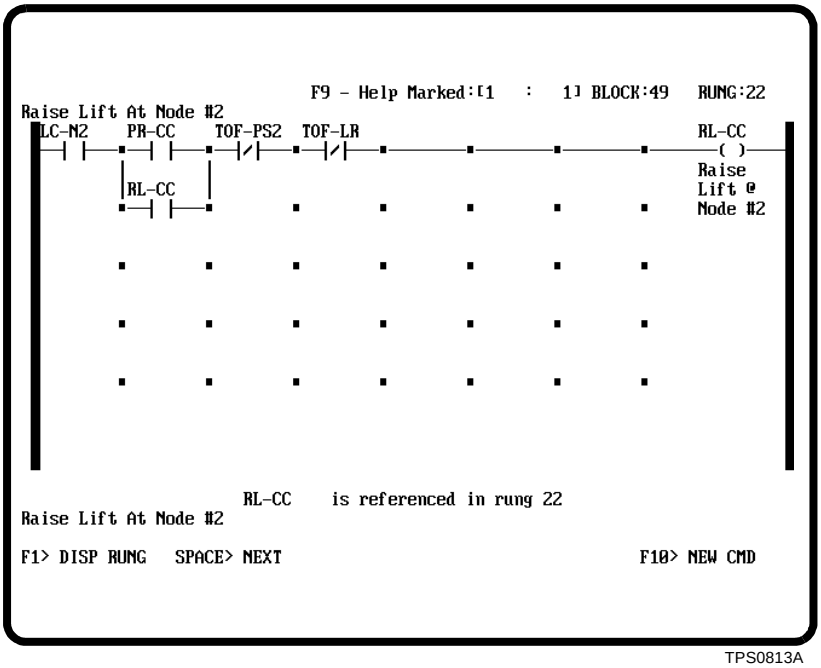
2. Then type **XREF** *tagname* Enter or **XREF** Enter. If the tag name is not specified then the tag name shown at the output of the rung is used. See to Figure 5-9.

3. Do one of the following:

   To display the next reference, press Space Bar

   To trace the reference to the indicated rung, press F1

   To type in a new editor command, press F10



TPS0813A

Figure 5-9. Tag References Screen

XTAG

PURPOSE: Modify tags.

COMMAND: **XTAG**

PROCEDURE:

1. Move the cursor to the command line by pressing F10

2. Then type **XTAG** Enter. The screen shown in Figure 5-10 appears.

3. Modify the tag as needed.

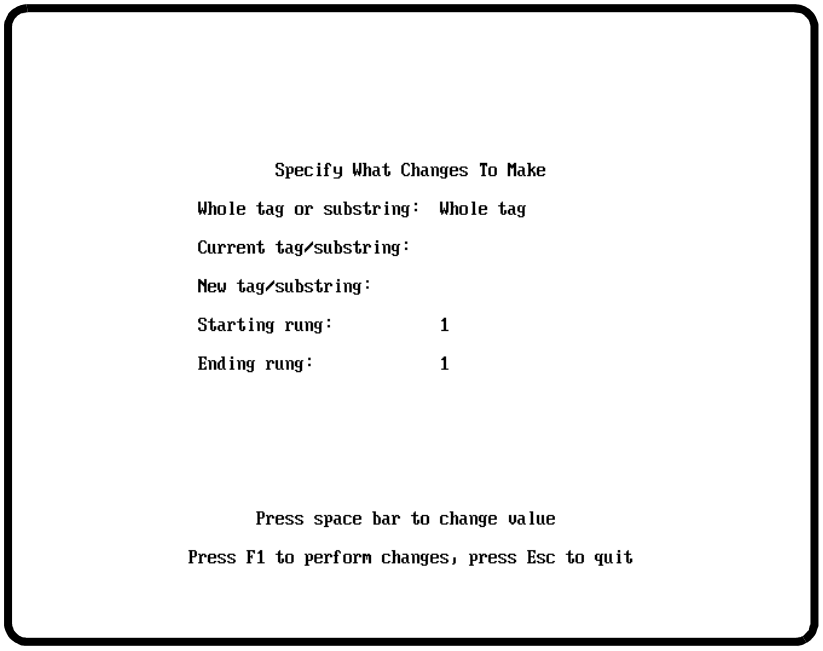


Figure 5-10. Modify Tag Screen

Modifying Rungs

Rungs are modified using the key commands listed in the following tables. Refer to Table 5-2 for text editing commands. These commands apply whenever text needs to be modified (e.g., heading, comments, tags, parameter values, etc.). Logic rung contacts are modified using the commands listed in Table 5-3. Logic rung coils are modified using the commands listed in Table 5-4. Function block rung inputs and parameters are modified using the commands listed in Table 5-5. Refer to Table 5-6 for commands used to modify function block outputs.

Table 5-2. Text Editing Key Commands

Key Command	Action
Clear from cursor to end of line.	Alt - F7
Clear line.	Alt - F8
Delete character.	Del
Delete column in comment area.	Alt - F4
Delete line in comment area.	Alt - F6
Go to command.	F10
Go to next area.	F2
Go to previous area.	F1
Insert column in comment area.	Alt - F3
Insert line in comment area.	Alt - F5
Restore saved line.	Alt - F10
Save line.	Alt - F9
Toggle insert and overtype mode.	Ins

Table 5-3. Modify Logic Rung Contact Key Commands

Key Command	Action
Delete column.	Alt - F4
Delete contact.	Del
Delete line.	Alt - F6
Go to area *n*.	Ctrl - Fn
Go to command.	F10
Go to next area.	F2
Go to previous area.	F1
Horizontal line.	-
Initialize.	Alt -I
Insert column.	Alt - F3
Insert contact.	Ins
Insert line.	Alt - F5
Modify tag.	T
Normally closed contact.	/

Table 5-3. Modify Logic Rung Contact Key Commands (continued)

Key Command	Action
Normally open contact.	**[**
Open.	**+**
Restore line.	**Alt** - **F10**
Save line.	**Alt** - **F9**
Toggle left connect.	**>**
Toggle right connect.	**<**
Toggles between the four contact types.	**?**

Table 5-4. Modify Logic Rung Coil Key Commands

Key Command	Action
Go to area *n*.	**Ctrl** - **Fn**
Go to command.	**F10**
Go to next area.	**F2**
Go to previous area.	**F1**
Modify descriptor.	**D**
Modify tag.	**T**

Table 5-5. Modify Function Block Rung Inputs or Parameters Key Commands

Key Command	Action
Clear line.	**Alt** - **F8**
Clear to end.	**Alt** - **F7**
Delete character.	**Del**
Enter data.	**Enter**
Go to command.	**F10**
Go to next area.	**F2**
Go to previous area.	**F1**
Page up.	**Page Up**
Page down.	**Page Dn**
Toggle insert or overtype mode.	**Ins**

Table 5-6. Modify Function Block Outputs Key Commands

Key Command	Action
Go to command.	**F10**
Go to next area.	**F2**
Go to previous area.	**F1**
Modify descriptor.	**D**
Modify tag.	**T**
Page down.	**Page Dn**
Page up.	**Page Up**

SECTION 6 - DOWNLOADING A LADDER PROGRAM

INTRODUCTION

This LADDER utility is used to transfer a ladder program from the computer to an INFI 90 module after it has been created or modified. The following outline can be used to download a ladder program:

1. Ensure that the ladder program compiled error free. Refer to ***COMPILE*** in Section 5.

2. Ensure that the INFI 90 interface is properly configured, refer to ***INTERFACING TO THE INFI 90 MODULE*** in this section.

3. Enter the appropriate ladder program file name to download when the prompt appears.

4. Put the module in the configure mode with *F1> CFG MODE*.

5. Use either *F6> LAD i* → mod or *F7> LAD m* → mod to download the ladder program.

6. Put the module back into execute mode with *F2> EXE MODE*.

Refer to ***MODULE FUNCTIONS*** in this section for procedures to use these functions and for other operations that can be performed on the modules.

INTERFACING TO THE INFI 90 MODULE

This section provides information on interfacing the computer to the INFI 90 module.

1. From the LADDER main menu, select *CONFIG* to initiate the configuration utility. The *INFI 90 INTERFACE DATA* screen appears as shown in Figure 6-1. If the interface data has already been specified, this screen does not appear.

2. Specify the interface data that LADDER needs to communicate with an INFI 90 processor module. Refer to Table 6-1.

3. When all the interface data is specified, press F1

If the computer is unsuccessful in trying to communicate with the SPM or CIU interface, the *CAN'T COMMUNICATE WITH MODULE* message is displayed indicating there is a problem that must be corrected before going on.

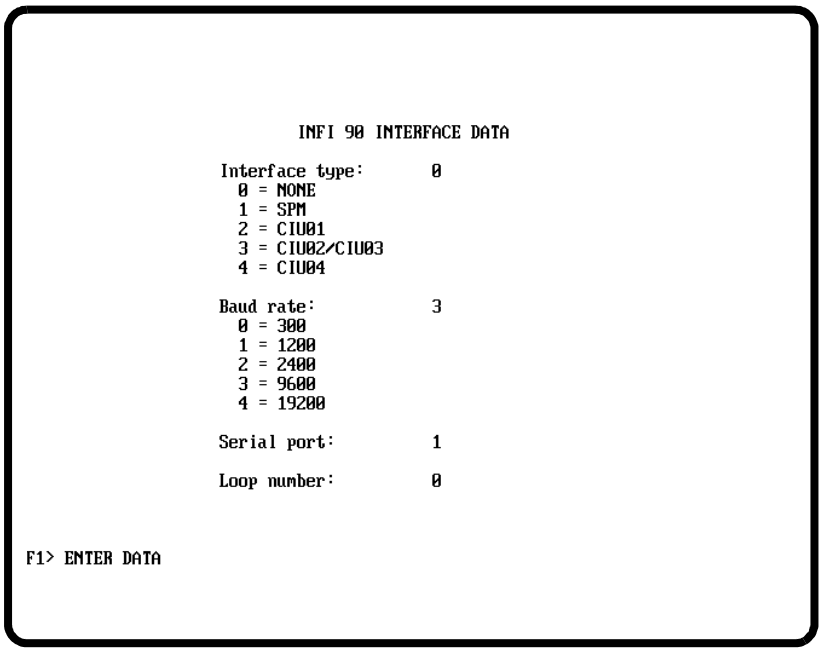


Figure 6-1. INFI 90 Interface Data Screen

Table 6-1. INFI 90 Interface Data

Field	Description
Interface type:	Specify whether the computer is connected to a serial port module (SPM) or to a computer interface unit (CIU). The serial port module communicates with any processor module in the same process control unit (PCU) as the serial port module. The computer interface unit communicates with any processor module connected to the plant loop. If there are no INFI 90 modules, specify *0* (none) to continue.
Baud rate:	Specify the speed (bits per second) that the computer will use to communicate with the SPM or CIU interface. Determine the baud rate of the SPM or CIU interface and specify that rate.
Serial port:	This specifies which serial port of the computer is connected to the SPM or CIU interface.
Loop number:	Specify the loop number with which to communicate.

Make sure that the processor module has the right module address and that the module is in normal operating mode. A green LED indicator should be visible through the front plate of the module. Press F1 again. If still unsuccessful, try different values for the interface data or specify **0** for no interface (off-line operation).

4. Once the computer has successfully communicated with the SPM or CIU interface, it requests a file name (Figure 6-2). If a ladder program is already in the computer memory, its name is displayed. Use that name or specify another file.

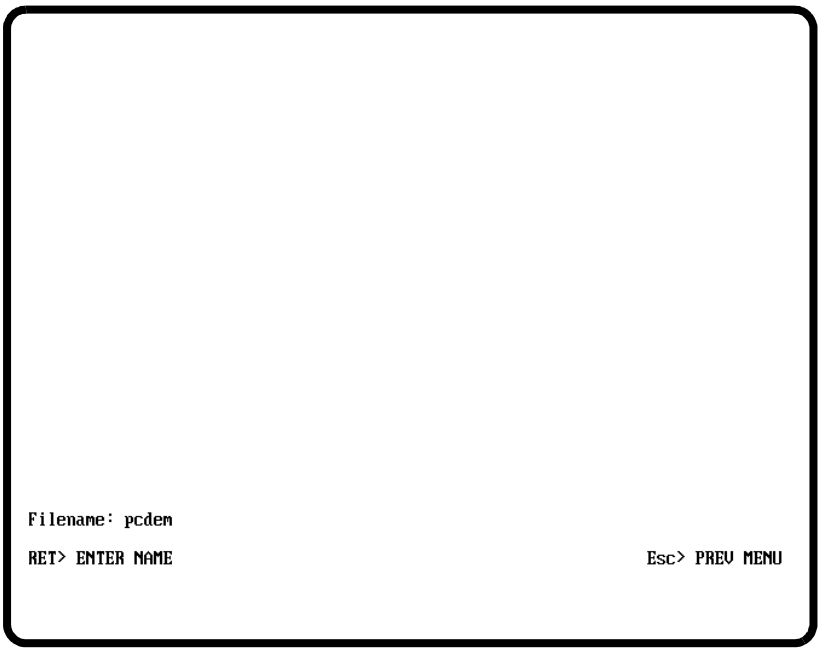


Figure 6-2. Data File to Download

5. Press Enter to continue.

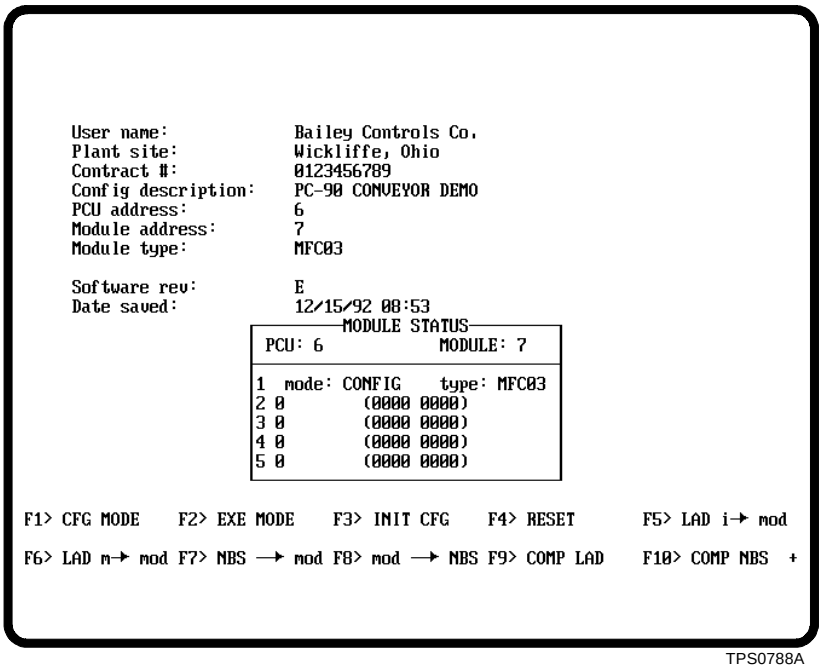
MODULE FUNCTIONS

After entering the data file to download, its title page appears in a screen along with the current status of the module associated with that program and a menu for selecting module functions (Figure 6-3). For more information on the module status refer to the appropriate product instruction listed under ***REFERENCE DOCUMENTS*** in Section 1. The menu provides selections for the functions that can be performed on the module.

Not all menu items fit on the bottom line of the screen as indicated by the character + in the rightmost column. The menu may be scrolled by pressing Tab or **+**.

> **NOTE:** Functions F7, F8, and F10 are the only functions provided by the LADDER Logic Configuration Utilities that pertain to BASIC or C programs. The normal BASIC and C programming functions (e.g., edit, list, run, etc.) are performed via a dumb terminal directly connected to the module. Once a BASIC or C program has been written in this way, F7 can be used to save the program on disk and F8 to restore it to the module.

F1> Cfg Mode Requests the module to go to configure mode. If the module was in execute mode, it stops running the ladder program and shuts down its outputs. In configure mode, the module accepts a new program or unrestricted changes to the existing program.



TPS0788A

Figure 6-3. Data File Title Page/Module Functions Screen

F2> Exe Mode Requests the module to go to execute mode and to begin running the ladder program (and the BASIC or C program, if there is one). Before starting to run the ladder program, the module examines it for certain kinds of errors. If it finds an error, the module goes to error mode instead of execute mode, and the *MODULE STATUS* display shows the error information. If no error is found, the module enters execute mode and begins running the program.

F3> Init Cfg Commands the module to initialize (i.e., erase its configuration, ladder program and BASIC or C programs).

F4> Reset Commands the module to perform a reset operation. The consequences of a reset depend on the state of the module (execute or configure mode and for some modules, primary or backup).

F5> Lad I → Mod Performs two operations. First it commands the module to initialize its program memory (same as F3). Then it downloads the current ladder program from the computer to the module (current ladder program means the ladder program currently in memory).

F6> Lad M → Mod Downloads the current ladder program to the module similar to F5. However, it does not perform the initialize operation. This has two consequences: First, if the module has a BASIC or C program, the program is not affected. Second, if the module already has a ladder program, the program from the computer is merged with the existing program.

F7> NBS → Mod Downloads a BASIC or C program from disk to the module. The path and file name of the current ladder program are used for the BASIC or C program on the disk. The extension ***.NBS*** is substituted for ***.LAD***.

Example: If the current ladder program is ***A:\ladder\pcdem.LAD***, the BASIC program name will be ***A:\ladder\pcdem.NBS***.

F8> Mod → NBS Uploads a BASIC or C program from the module to the disk. The name of the BASIC or C program is derived from the current ladder program (as in F7).

F9> Comp Lad Compares the program in the module to the current ladder program (using the ladder program as the original). Starting with rung one, each rung in the ladder program is compared with the corresponding function block in the module. When differences are found, the relevant rung and block data are displayed, one rung at a time.

> **NOTE:** This comparison does not necessarily involve every function block in a module - only those which correspond to rungs in the ladder program.

F10> Comp NBS Compares a BASIC or C program in the module to a BASIC or C program on disk. The name of the disk file is determined as in F7 and F8. This function will determine only if the disk-resident and module-resident programs are identical or not. It will not report what the differences are or where they occur.

Alt-F1> SGL/RDN Some module types (e.g., MFC and MFP modules) allow modules to be configured in redundant pairs. At any given time, one module acts as the primary and the other as a hot backup. Normally, LADDER software communicates only with the primary. However, with the proper setup of the modules, it is possible for the LADDER software to communicate with the backup - even to reprogram the backup while the primary is running.

The *SGL/RDN* function switches the LADDER software between single and redundant modes. In single mode, the LADDER software communicates only with the primary module.

In redundant mode, the LADDER software communicates with either the primary or the backup. The module address of the backup is always one greater than the module address of the primary.

In single mode, the screen shows the module status for only the primary, and the commands discussed above (F1, F2 etc.) are directed to it. In redundant mode, the screen shows the module status for both the primary and the backup. The commands in this case can be directed to either the primary or the backup. The current target module is indicated by the blinking asterisk (*) inside the module status box. The target can be switched to the backup with → and to the primary with ←

SECTION 7 - PRINTING A LADDER PROGRAM

INTRODUCTION

This LADDER utility prints a listing of a ladder program. Select *PRINT* from the main menu. The *PRINTER* menu appears (Figure 7-1). If printing to a printer for the first time, refer to ***SETTING THE PORT SPECIFICATIONS*** in this section to set up the printer port for your printer. Refer to ***TESTING THE PORT*** in this section for information on testing the printer. Refer to ***STARTING THE LADDER PROGRAM*** in Section 3 for print start-up options.

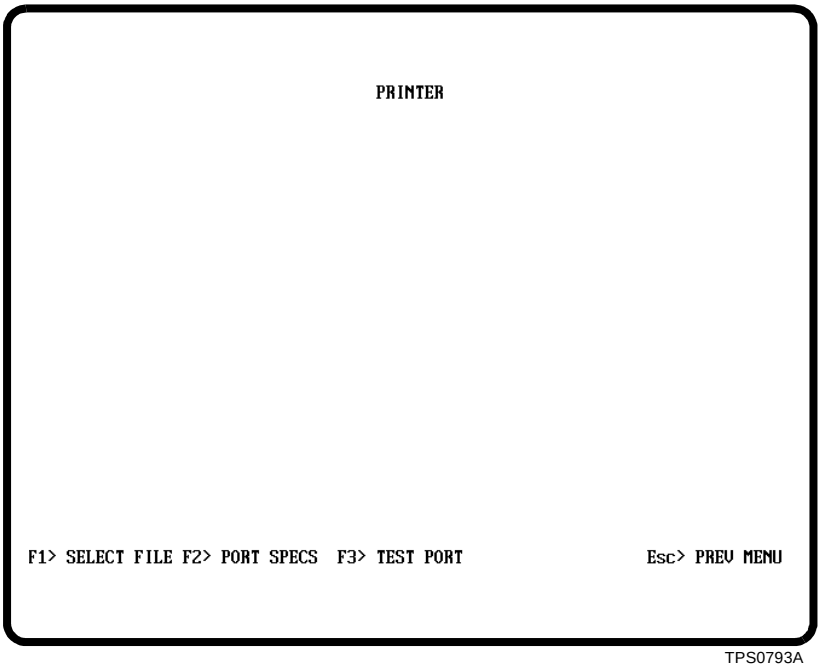


Figure 7-1. Printer Menu

SELECTING A FILE

Use this function to provide print utility specifications.

1. Choose *SELECT FILE* from the *PRINTER* menu. The program prompts for the file name.

2. Enter the path if the file is not in the current directory. After entering the file name, press **Enter**.

3. The file title page appears. Press **Space Bar** to continue.

4. The screen shown in Figure 7-2 appears. Refer to Table 7-1 to fill in the file print specification fields.

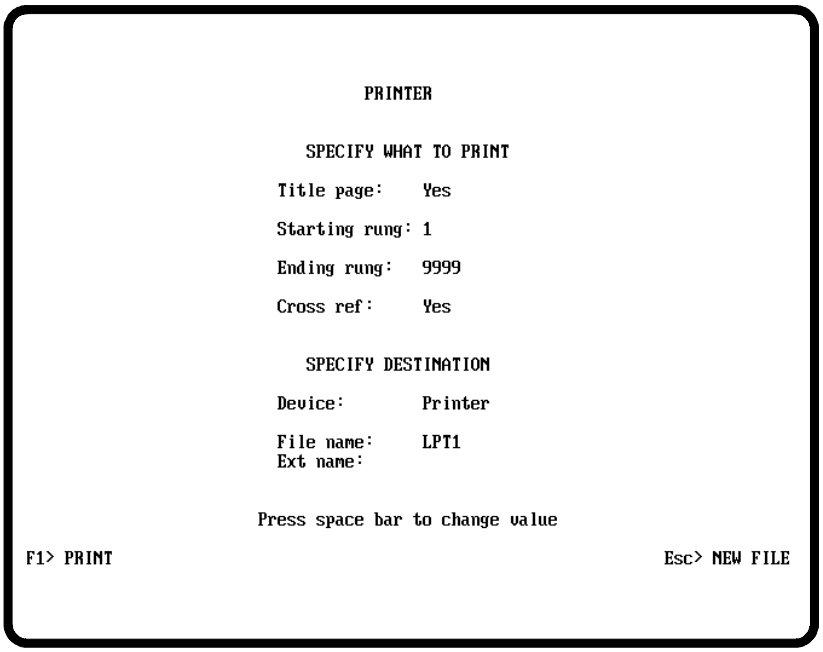


Figure 7-2. Print Specification Screen

Table 7-1. File Print Specifications

Field	Description
SPECIFY WHAT TO PRINT	
Title page:	Use the spacebar to select Yes or No.
Starting rung:	Specify the rung number of the first rung to be printed. Specify 0 to not print any rungs.
Ending rung:	Specify the last rung number to be printed. A number larger than the highest rung number is equivalent to the highest rung number.
Cross ref:	Use the space bar to select Yes or No for cross-reference. The cross-reference is an alphabetic list of all the tags defined in the ladder program and includes: Rung number. Block number. Description. Function code. List of all rungs which refer to the tag.
SPECIFY DESTINATION	
Device:	Specify whether the listing is to be printed (destination device = Printer) or stored on disk (destination device = Disk). Refer to Appendix A for a sample listing of the ***pcdem*** file.
File name:	This field may be edited to contain the desired file name if the destination device is *Disk*. Otherwise, it contains the specified printer port (Figure 7-3).
Ext name:	If printing to a file, enter the DOS file extension.

5. After entering the field information appropriately, press F1 to start printing.

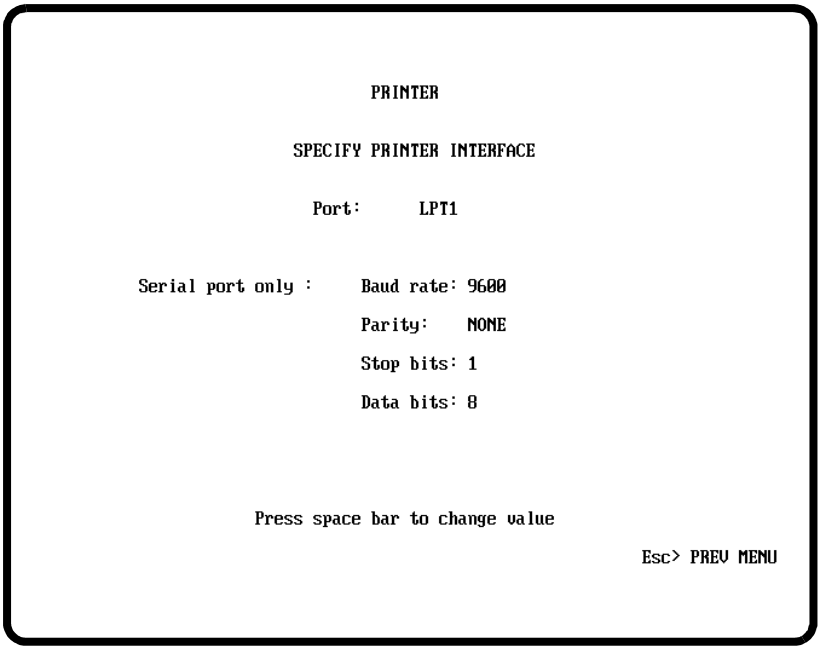
6. When printing is finished, press Esc. The program asks for another file name. Press Esc to return to the *PRINTER* menu, and once more to return to the main menu.

SETTING THE PORT SPECIFICATIONS

This selection specifies printer attributes used to print the ladder program.

1. Select *PORT SPECS* from the *PRINTER* menu. The screen shown in Figure 7-3 appears.

2. Refer to Table 7-2 to fill in the fields appropriately. Press Esc when done to return to *PRINTER* menu.



TPS0795A

Figure 7-3. Specify Printer Interface Screen

Table 7-2. Printer Port Specifications

Field	Description
Port:	Specifies a parallel interface (Centronics type) or a serial interface (RS-232-C) for the printer. Press the space bar to select the appropriate choice for your printer: LPT1, COM1, or COM2.
Serial port only:	
Baud rate:	300, 1200, 2400, 9600.
Parity:	Odd, even, or none.
Stop bits:	1, 2.
Data bits:	7, 8.

TESTING THE PORT

This function writes a test pattern to the printer to check that the printer is properly set up and connected. Select *TEST PORT* from the *PRINTER* menu. The printer should advance the paper to a new page and print the following text on one line:

<< = column 1 column 132 = >>

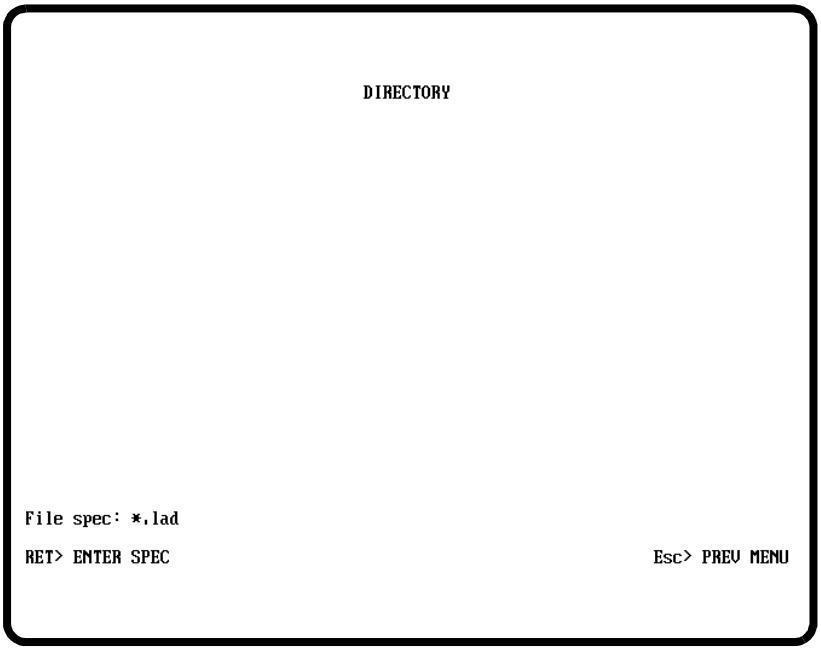
SECTION 8 - LISTING THE DIRECTORY

INTRODUCTION

The directory utility provides a directory listing of the current working directory.

1. Select *DIRECTORY* from the LADDER main menu. The screen shown in Figure 8-1 appears.

2. Press Enter to accept the default file specification (***.lad***) or type in a new file specification. The directory listing appears. An example is shown in Figure 8-2.



TPS0796A

Figure 8-1. Directory File Specification Screen

The key commands perform the following operations:

F1> New Spec Permits changing the file specification as in Figure 8-1.

Space> Next Pg Space Bar shows the next page of the directory listing.

Ret> Read File After using the cursor keys to highlight a file, Enter displays the selected file title page as shown in Figure 8-3 and sets the selected file as the default file to use for the editor utility.

Esc> Prev Menu Returns to the LADDER main menu.

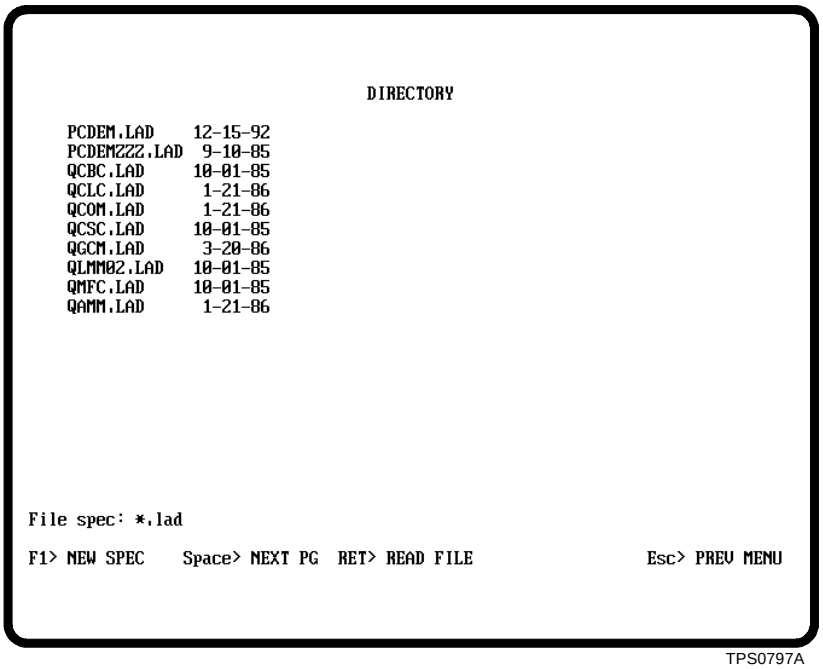


Figure 8-2. Example Directory Listing

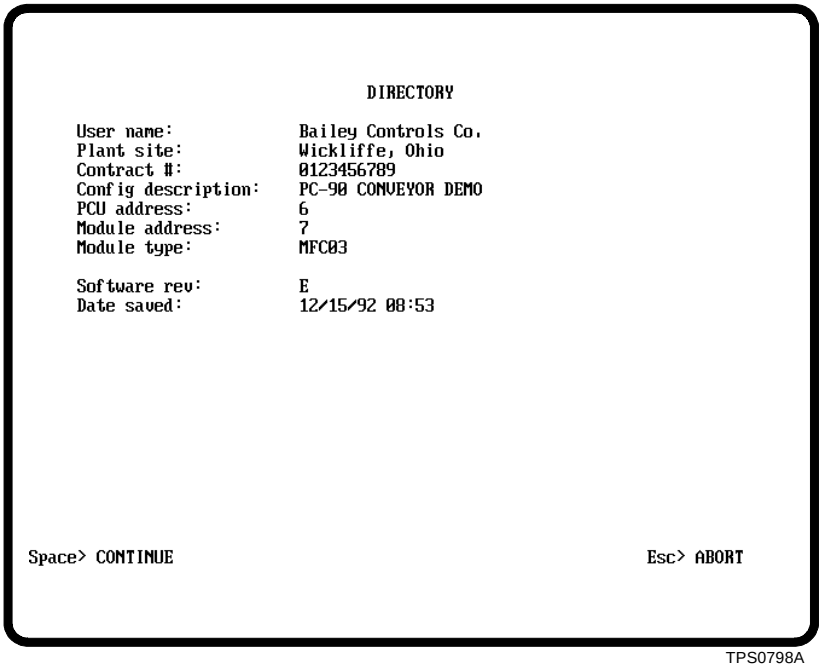


Figure 8-3. Example Read File Screen

SECTION 9 - CONFIGURING THE INFI 90 INTERFACE

INTRODUCTION

This section describes the procedures to configure the interface to the INFI 90 processor module.

1. From the LADDER main menu, select *CIU/SPM* to initiate the configuration utility. The *INFI 90 INTERFACE DATA* screen appears as shown in Figure 9-1.

2. Specify the interface data that LADDER needs to communicate with an INFI 90 processor module. Refer to Table 6-1.

3. When all the interface data is specified, press F1 The utility stores the interface data and returns to the LADDER main menu.

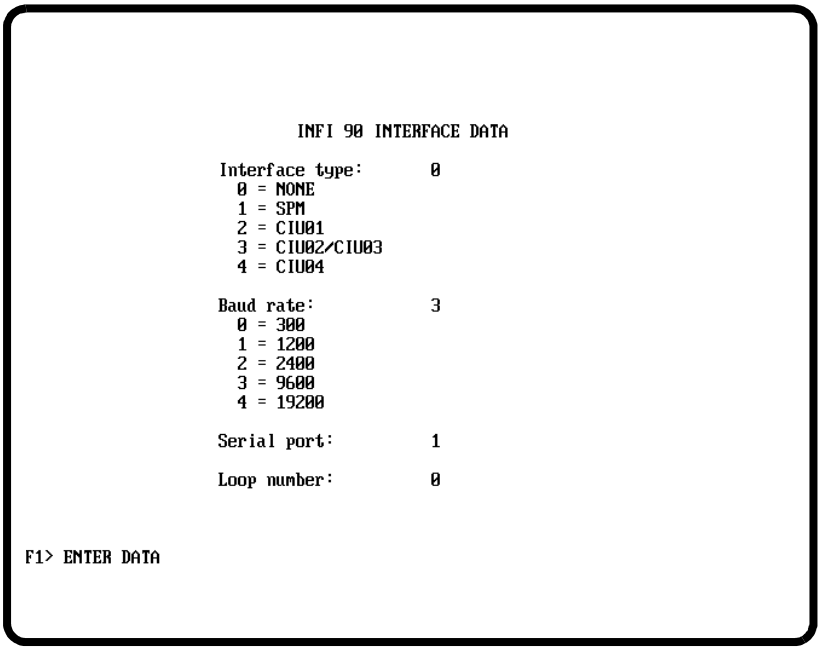


Figure 9-1. INFI 90 Interface Data Screen

APPENDIX A - TUTORIAL

INTRODUCTION

This section of the manual provides a tutorial for the LADDER Logic Configuration Utility. It discusses all major features of the utility using the demonstration ladder program file ***PCDEM.LAD*** provided with the software package. The tutorial assumes that the software package is installed as described in Section 2 and is connected to an INFI 90 processor module. To begin the tutorial, start the LADDER Logic Configuration Utility by typing the following at the DOS prompt (refer to ***STARTING THE LADDER PROGRAM*** in Section 3 for start-up options):

SLAD Enter

The main menu screen appears as shown in Figure A-1. Menu selections appear on the bottom two lines of the screen. Function keys are used to select these menu items which are described in the following paragraphs.

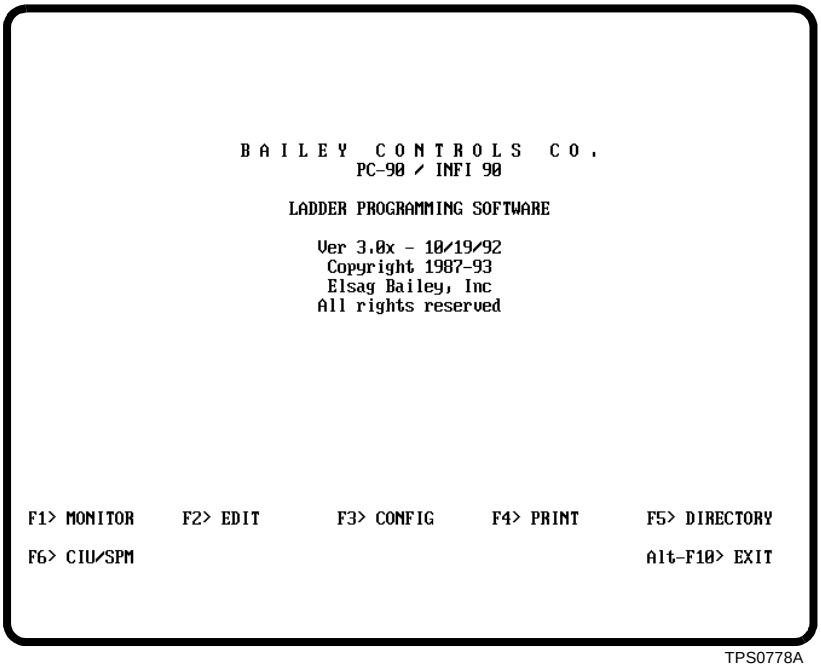


Figure A-1. LADDER Main Menu

F1> Monitor Monitor the operation of a INFI 90 module as it runs a ladder program.

F2> Edit Create a new ladder program or modify an existing one.

F3> Config Download a ladder program from the computer to an INFI 90 module.

F4> Print Print a listing of a ladder program.

F5> Directory List names and creation dates of files located in a specified directory.

F6> CIU/SPM Specify the type of interface between the computer and the INFI 90 module.

EDITOR

This section describes the use of the editor to modify or create a new ladder program. It is an off-line editor, meaning that it can be used without having the target module (the INFI 90 module that runs the program) connected to the personal computer.

1. Press F2 to select *EDIT*. The display shown in Figure A-2 appears on the screen.

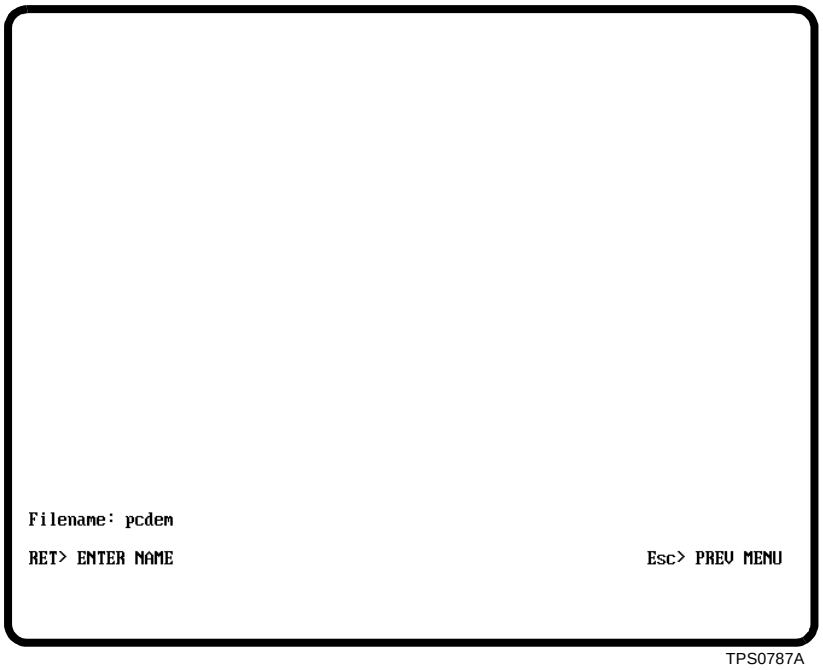


Figure A-2. Off-Line Editor Opening Screen

A prompt appears requesting the name of a ladder file. A ladder file is a unit of storage on a disk that contains a ladder program. The file has a name of up to eight characters followed by a period and a three-character extension. The extension is always ***.LAD***. Type in only the eight characters of the name, not the extension. In addition to the file name, a path may be specified. The path specifies where the file resides. It consists of a

drive specification (e.g., a:, b:, or c:) and a directory specification (e.g., \ladder). For example, the path and file name **C:\LADDER\PCDEM** specifies drive C, directory LADDER, and file ***PCDEM.LAD***.

2. A demonstration ladder file, ***PCDEM.LAD***, is supplied along with the ladder software. This file is used to demonstrate the operation of the software package. Type **PCDEM** Enter.

After the editor reads the file, it displays the title page of the file (Figure A-3). The title page provides general information about the program. Although not shown in Figure A-3, the command line is a highlighted area at the bottom of the screen. The blinking box at the left of the command line is the cursor and points to a location on the screen.

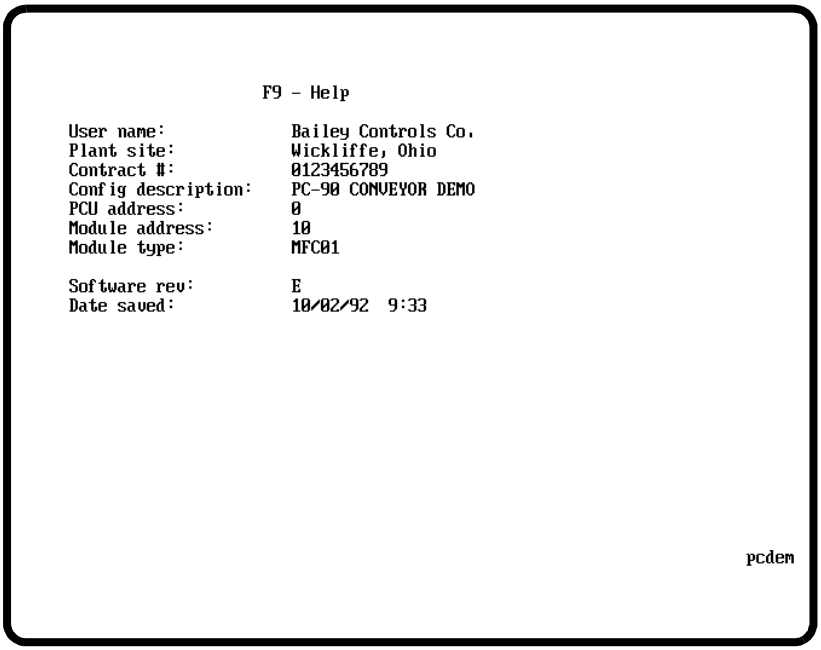


Figure A-3. Off-Line Editor Title Page

Appendix B lists and describes all of the editor commands. However, there is another way to view this list of commands. The editor provides a context sensitive help display. This means that the help display depends on what the cursor is pointing to at the time help is requested.

3. For example, with the cursor initially on the command line, press F9. The help display shown in Figure A-4 appears. Since the cursor was on the command line, the list of commands is displayed. The help display is three lines long so commands appear three at a time. Use the following keys to view the help display:

↓, Page Dn To scroll down the list.

↑, Page Up — To scroll back up the list.

F9 — To remove the help display. Press F9 again and the cursor returns to its original position.

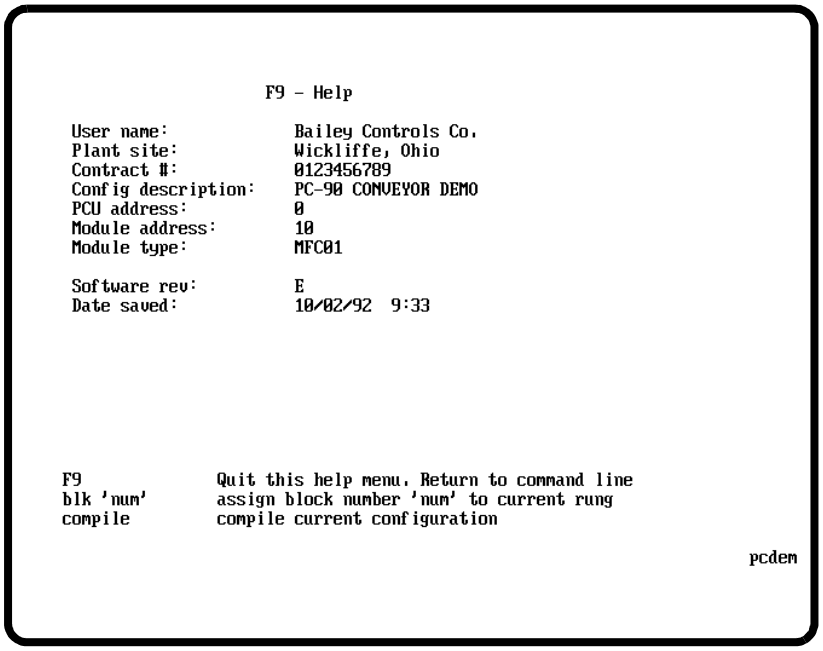


Figure A-4. Title Page with Help Display

4. Rung 22 is to be displayed. Refer to the help display which shows that a command to display a rung is **D** *nn* (where *nn* represents a rung number). On the command line type **D 22** Enter. Rung 22 appears as shown in Figure A-5.

A rung is the basic unit of a ladder program. There are two types of rungs: The logic rung and the function block rung (the rung currently on display is a logic rung). For editing and displaying purposes, a rung is divided into four areas:

Heading - This is a single comment line (up to 80 characters) that appears at the top of the rung display. The function of the heading is to provide a brief explanation of the rung.

Specifications - For a logic rung, this area specifies a logic equation in terms of a series and parallel combination of contacts. The contact grid is seven contacts wide by five lines high. However, the maximum number of contacts per rung is typically about 15 (the number depends on the complexity of the logic). For a function block rung this area specifies the inputs and parameters of the function block (described later).

Outputs - For a logic rung this area contains the coil (i.e., the output of the logic equation). For a function block rung this

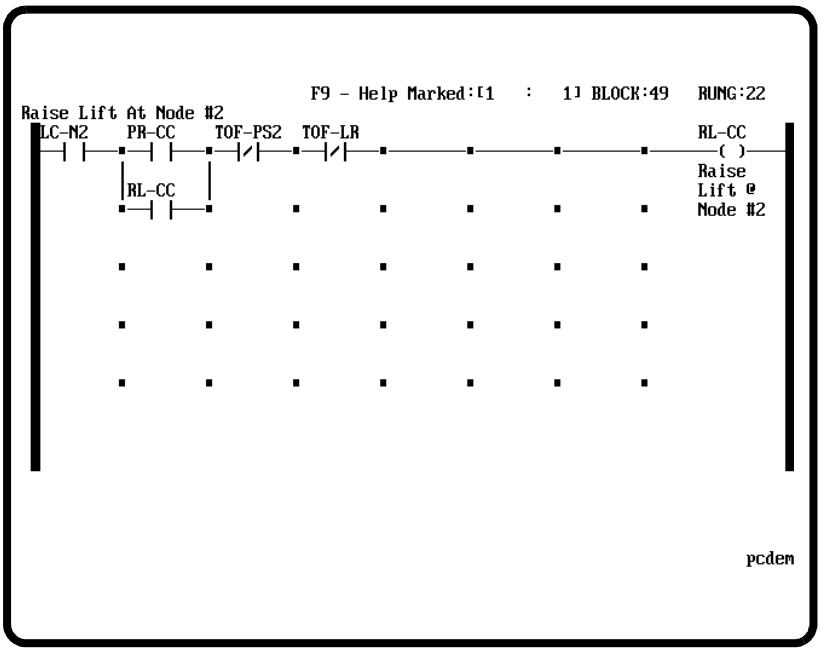


Figure A-5. Logic Rung Sample Screen

area contains the outputs of the function block (described later).

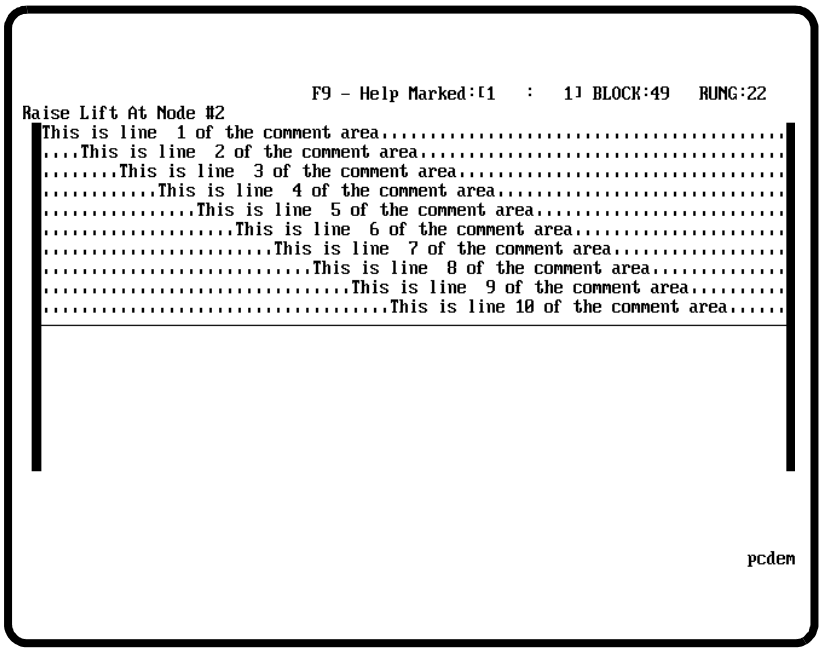
Comments - This is a free-form, multiline comment area that appears at the bottom of a rung (on a printed listing) or on a separate display screen. It is 77 characters wide by ten lines high. The comment area may contain any information that is relevant to the rung. Figure A-6 shows the comment screen.

Each rung is identified by a rung number. For example, if a ladder program consists of 100 rungs, the rungs are numbered one through 100. The rung numbers are assigned automatically by the editor as rungs are inserted and deleted. For example, if a new rung were inserted at rung 50, then the existing rungs 50 through 100 would be reassigned rung numbers 51 through 101.

Logic Rungs

This section examines rung 22 and demonstrates features of the editor. As mentioned earlier, a logic rung has four areas. Move the cursor between the areas as follows:

1. Press F2 and the cursor moves to the heading area.
2. Press F2 again and the cursor moves to the logic circuit.
3. Press F2 again and the cursor moves to the coil.



TPS0782A

Figure A-6. Comment Screen

4. With the cursor at the coil press F2 and the cursor moves to the comment area. The logic circuit and the coil are replaced on the display by the comment area (Figure A-6).

5. Now press F2 and the cursor returns to the heading.

F1 moves the cursor between the areas, but in the opposite direction.

HEADING

The heading is a single line of text. It is helpful if the heading contains a brief description of the rung, but it may be left blank.

1. With the cursor in the heading, press F9 to view the help display. The help display shows the currently active keys and the functions associated with each key. For example, Ins is active. Its function is to toggle between insert mode and replace mode. Now edit the heading.

2. Press F9 to return the cursor to the heading.

3. Press → until the cursor reaches the # character.

4. Press Ins several times and observe that the size of the cursor changes. The larger cursor indicates replace mode; the smaller cursor indicates insert mode.

5. Select replace mode and press **?**. Observe that # is replaced by *?*.

6. Move the cursor back to *?* using the ← key. Press Del and the *?* is deleted and characters to the right of the cursor move one position to the left.

7. Try the other keys listed in the help display. A key combination such as Alt-F7 means hold down Alt while pressing F7.

COIL

The coil represents the output of the rung (i.e., the result of the logic circuit). A logic rung can have only one coil. The coil (or the value it represents) is identified by a tag name, in this case, *RL-CC*. A tag consists of from one to eight characters (i.e., letters, digits, and special characters). The first character must not be a digit or # and there can be no imbedded spaces. Also, letters are distinguished by uppercase and lowercase (e.g., *A* is different from *a)*.

Also associated with the coil is a description. The description is a comment consisting of a block of text eight characters wide by three lines long. It may contain any information relevant to the coil or be left blank. In general, the description should be used to help to identify the coil or explain its use. In this case, the description is:

Raise
Lift @
Node #2

1. Press F2 to move the cursor to the coil area of the rung.

2. As an aid when modifying a tag, enable the help display, if desired, by pressing F9.

3. With the cursor at the coil, press **T** and the cursor moves to the tag. Now the tag can be edited in the same manner as the header. For practice, change the last character from *C* to *c*.

4. Now press Enter and the cursor moves to the description.

5. Press F9 to view the help display if desired. Return the cursor to the description.

6. Use the ↑ and ↓ keys to move the cursor from line to line. For practice, change *@* to *at*. Then press Enter until the cursor returns to the coil.

LOGIC CIRCUIT

The logic circuit specifies the operation of the rung in terms of series-parallel connections of contacts. Each contact represents a Boolean variable. A contact has two attributes that describe how the value of the contact is to be determined.

- The tag associated with a contact specifies a variable which is output by a rung (i.e., the coil of a logic rung or an output of a function block rung).

- The contact type specifies an operation to be performed on the variable. The type is indicated by the symbol used for the contact as shown in Figure 5-4.

Applying these ideas to the example of Figure A-5, the logic circuit can be interpreted as:

(value of *LC-N2*)

and

((value of *PR-CC*) or (value of *RL-CC*))

and

(inverse of the value of *TOF-PS2*)

and

(inverse of the value of *TOF-LR*)

To practice editing the logic area, follow these steps:

1. With the cursor at the coil, press F1 to move the cursor to the logic circuit area.

2. To edit the logic circuit, use the arrow keys to see how the cursor moves. Look at the help display to see what keys are active.

3. Move the cursor to the contact with tag *RL-CC*. Change the contact type by pressing **/** once, **?** several times, and then **[** once.

4. The tag is edited in the same way as the tag of the coil. Try changing *RL-CC* to *RL-Cc*.

5. Next, insert a new contact in series with *RL-Cc*:

   a. Move the cursor one position to the right of *RL-Cc*.

   b. Press Alt-F3 to insert a blank column.

c. Press **[** to place a contact.

d. Type **newtag** Enter.

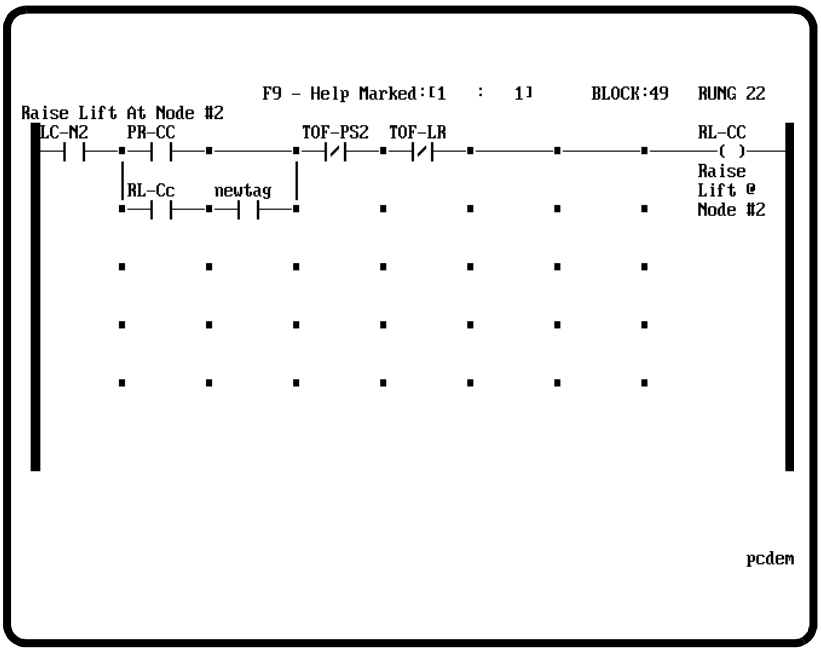
e. Move the cursor up one position.

f. Press **-** to place a horizontal connector.

g. Press **<** to remove the vertical connector to the left of *newtag*.

h. Press **>** to place a vertical connector to the right of *newtag*.

The result should look like Figure A-7. The order in which changes are made to a logic circuit is not important. There are certain rules that the circuit must adhere to (refer to Appendix D); however, the editor does not check the circuit until the cursor is moved out of the circuit area.



TPS0800A

Figure A-7. Display a New Contact in Series with RL-CC

6. Press F2. The cursor should move to the coil, indicating that the circuit has no logical errors.

7. Put an error in the circuit:

a. Press F1 to move the cursor back to the circuit.

b. Move the cursor to the position above *newtag*.

c. Press **<** to place a vertical connector to the left of *newtag*.

d. Press F2 again.

The message ****ERROR 9103**** should appear indicating that the editor found a rule violation at row 1, column 3.

8. Remove the error by pressing **<** to remove the vertical connector and then press F2.

COMMENT AREA

The comment area is a block of text 77 characters wide by ten lines high. It may contain information or it may be left blank.

1. Press F2 to move the cursor to the comment area.

2. Use the cursor keys to see how the cursor moves within the comment area. Also, try Home, End, →, ← and Enter

All the text editing functions used so far (insert, delete, etc.) work in the comment area and can be practiced now.

3. To try some new functions, look at the help display and move the cursor to the *5* of line 5.

4. Press Alt-F5 to insert a blank line at the cursor position; lower lines move down and line 10 is lost.

5. Press Alt-F6 to delete the line at the cursor; lower lines move up and the bottom line is blank.

6. Press Alt-F3 to insert a blank column at the cursor position. Columns on the right of the cursor move one position to the right and the last column is lost.

7. Press Alt-F4 to delete a column at the cursor. Columns on the right of the cursor move one position to the left and the last column is blank.

8. To move or duplicate a line:

a. Move the cursor to line 1 and press Alt-F9. Although nothing visible happened, the editor made an internal copy of line 1.

b. Move the cursor to line 4, select insert mode (indicated by a small cursor), and press Alt-F10. A copy of line 1 is inserted.

c. Move the cursor down one line, select replace mode (indicated by a larger cursor) and press Alt-F10. The copy of line 1 replaced the line at the cursor.

Function Block Rung

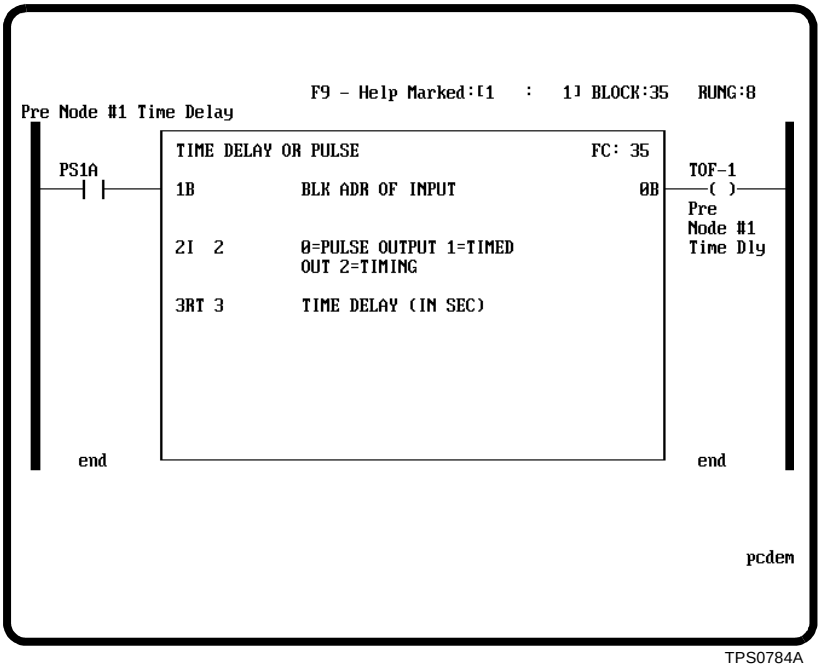
The examples so far have used only the logic rung. The function block rung is used next. A function block may be thought of as a software block box that takes input values (which are outputs from other rungs), operates on those values in a manner determined by its function code and a set of parameters, and produces output values. Be aware that the output of a rung is not just a value but also an item of data. When a new rung is created, a memory location is reserved to store the value and the value is placed into this location. The editor automatically keeps track of internal memory usage.

INFI 90 modules offer a wide variety of function blocks including such functions as timer, counter, drum programmer, comparator, limiter, and PID controller. Refer to the ***Function Code Application Manual*** for a complete list of these functions.

For an example of the function block rung, perform the following steps to display rung eight:

1. Press F10 until the cursor returns to the command line.

2. Type **D 8** Enter. The screen should look like Figure A-8.



TPS0784A

Figure A-8. Function Block Rung Example

For this form of rung, the box in the middle of the display represents the function block. Inputs are to the left of the box, outputs to the right, and the function code and parameters are inside the box.

The top line inside the box identifies the function code. On the left is the name of the function code, *TIME DELAY OR PULSE*, and on the right is the function code number, 35.

To the left of the box is the symbol ——| |—— which represents a Boolean input to the block. The input in this case is identified by the tag *PS1A*. In general, this input could come from any rung (i.e., a coil or the output of a function block rung). The only restriction is that it must be of type Boolean as opposed to type real (i.e., analog).

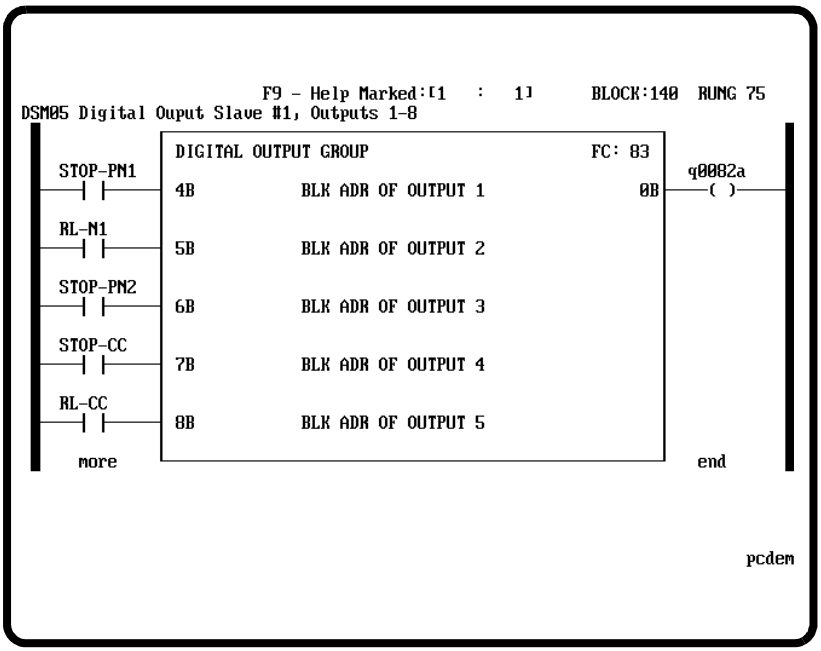
On the same line but inside the box is *1B*. The *B* indicates the input type (B = Boolean, R = real). The *1* indicates the specification number (inputs and parameters are collectively referred to as block specifications and are identified by a specification number). To the right of *1B* is *BLK ADR OF INPUT*. This is the specification name. In this case, since there is only one input, it is not very useful. For blocks with more than one input, however, it is helpful. For a memory block, the specification names would distinguish between the set input and the reset input.

Examine the line which begins with *2I*. This line represents a parameter since there is no symbol to the left of the box. The *2* identifies the specification number and the *I* identifies its type (I = integer, R = real, B = Boolean). Next on the right is *2* which indicates the value of the parameter. To the right of the value is the specification name, *0= PULSE OUTPUT*, *1 = TIMED OUT*, *2 = TIMING*.

Now examine the line which begins with *3RT*. This line represents the next parameter. Its specification number is 3 and its type is real. The *T* indicates that the parameter is tunable. This means that the parameter may be changed even while the program is being run by the module (i.e., the module is in execute mode). The value of the parameter is 10.0. The description shows that this parameter specifies the length, in seconds, of the time delay.

In this example, there are three specifications. There is room on the display for a total of five specifications. Since some function blocks have more than five specifications, the display may be scrolled up and down the list by use of the arrow and page keys.

The *end* on the bottom left of the display indicates that the end of the specification list is on display. If the end of the specification list is not visible, *more* is displayed in place of *end* as in Figure A-9.



TPS0801A

Figure A-9. Function Block Rung Display with More

To the right of the box in Figure A-8, the symbol ——( )—— represents a Boolean output of the block (i.e., same as a coil). This output is identified by the tag *TOF-1*. This tag identifies a specific item of data, so the tag must not be used again on the output (right) side of any rung. However, this tag may be used as an input or contact by any rung. In other words, tag *TOF-1* may appear any number of times on the left side (input) of any or all rungs but only once on the right side (output).

Below the output symbol is its description. In this case it is:

Pre
Node #1
Time D1y

Associated with this output is *0B*. The *0* indicates the number of this output (output numbering starts with zero). The *B* indicates the output type (B = Boolean, R = real). The output type is also indicated by its symbol:

——( )—— Represents a Boolean output type.

——(( )—— Represents a real output type.

There is room on a function block display for two outputs. Some function blocks have more outputs than can be displayed at once. As with the specifications, the outputs may be scrolled up and down; *end* is displayed if the last output is visible, otherwise *more* is displayed. The function block rung also has a heading and a comment area, just as the rung block.

Make some changes to this rung to practice editing. (The help display is available at any time by pressing F9)

3. Press F2 to move the cursor to the heading. Edit the heading of a rung block in the same way as the function block heading.

4. Press F2 again and the cursor moves to the first specification. Use a different value as the input to this block by entering the tag name here. For example, type **newtag**.

5. Now press Enter or ↓ and the cursor moves to the next specification. Leave this specification unchanged. Press Enter or ↓ to move the cursor to the last specification.

6. Change the 3.0 to 2. Several different keystroke sequences to accomplish this are shown in the following steps (starting with 3.0 and the cursor at the left).

   a. **2** Space Bar Space Bar

   b. Space Bar Space Bar **2**

   c. **2** Del Del

   d. **2** Del (hold down Del until the *0* is cleared)

   e. **2** Alt-F7

   f. Alt-F7 **2**

7. Move the cursor to the output by pressing F2

8. Change the tag for this output: Press **T**, type in the new tag and press Enter.

9. To change the description press **D**, make the changes, and press Enter until the cursor returns to the coil.

If this block had another output, the cursor would be moved to it by pressing ↓ or Enter.

10. Press F2 and the cursor moves to the comment area. The comment area of a function block rung is edited in the same manner as the comment area of a relay-logic rung. Press F2 again and the cursor moves back to the heading. Press F10 and the cursor moves to the command line.

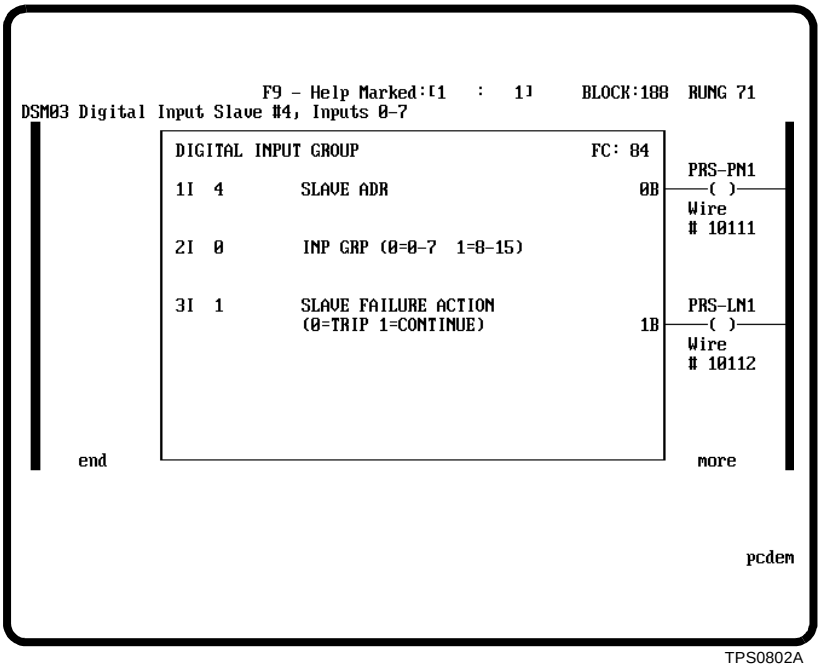
I/O Function Blocks

The preceding examples used function block rungs and logic rungs to show how to specify the flow of data among rungs. A tag name was assigned to an output of a rung and specified as

the input to another rung. The next examples look at processing inputs and outputs to the ladder program - the input and output function blocks.

External data can enter a ladder program from a variety of sources. Process data may be read via special hardware modules called I/O modules or read from other INFI 90 modules via the module bus or plant loop. Similarly, data can go to a variety of destinations. There is a specific function code for each type of input and output. For example, one function code reads a group of eight digital input values from a digital input module. One function code writes a group of eight digital outputs to a digital output module. Another reads a digital input over the plant loop, and so on. Two of these will be covered in more detail to illustrate the concept.

1. With the cursor on the command line, type **D 71** Enter. The screen shown in Figure A-10 appears. Rung 71 is a *DIGITAL INPUT GROUP* function code. It reads a group of eight digital inputs from a digital input module and makes those values available to the ladder program.



TPS0802A

Figure A-10. Input Module

2. Press F2 until the cursor is at the output named *PRS-PN1*. *PRS-PN1* and *PRS-LN1* are outputs 0 and 1 from this block.

3. This block has eight outputs; press Page Dn several times to examine them. These outputs correspond to the eight values read from the module. The module these values came from is determined by specifications 1 and 2. Specification 1 identifies the module, *SLAVE ADR*. Specification 2 identifies the group of

eight inputs (group 0 or group 1) on that module, *INP GRP*. For this example, *PRS-PN1* corresponds to the first input of group 0 of module 4, *PRS-LN1* corresponds to the second input of group 0 of module 4, and so on.

Notice that there are no inputs shown on the left side of the block. Input symbols represent inputs to a rung that are outputs from other rungs. In this case, the input values come from a module, not from other rungs. Think of this block as a source of data which does not require inputs, not as a processor of data which would require inputs.

4. Go to another rung: Press F10 and type **D 75** Enter. See Figure A-11, rung 75 is a *DIGITAL OUTPUT GROUP* function code. The function of this rung is the opposite of the function of the previous rung. This rung takes eight digital values from other rungs and writes them to a digital output module.

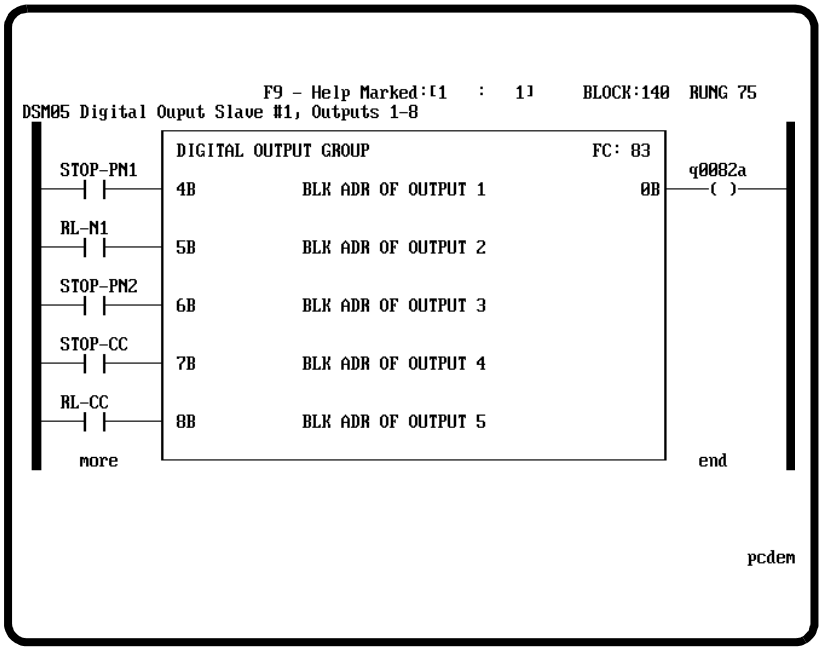


Figure A-11. Output Module

5. Press F2 twice. The cursor should be at the first input. Notice that five inputs are shown.

6. Now press Page Dn and the remaining three inputs are displayed. Two other specifications are also displayed - *SLAVE ADDR* and *OUTPUT GROUP*. These specifications specify the group and the digital output module to which the eight values are to be written.

For example (as can be seen by pressing Page Up, the value associated with tag *STOP-PN1* is written to the first output of

group 0 of module 1, the value of *RL-N1* is written to the second output of group 0 of module 1 and so on.

The block has one output that indicates the overall status of the digital output module (good or bad). Note that this output does not represent field wiring or a physical output. The inputs and outputs shown in the rungs represent items of data that exist within the ladder program.

Creating New Rungs

The following steps describe how to create a new rung.

1. If necessary, press F10 to move the cursor to the command line.

2. View the help display (F9) and scroll down to the **I** commands. There are four of them:

I	Insert logic at current rung.
I A	Insert logic after current rung.
I *nnn*	Insert function code *nnn* at current rung.
I *nnn* **A**	Insert function code *nnn* after current rung.

All of these commands insert a rung and increment the rung numbers of subsequent rungs. The first command inserts a logic rung at the current rung number (number of the rung on display). The second command inserts a logic rung after the current rung number. For example, if rung 20 is on display, **IA** inserts a logic rung at rung number 21.

The third command inserts a function block rung at the current rung number and the fourth command inserts a function block rung after the current rung number. In the third and fourth commands *nnn* stands for a function code number. The command **I 35** means to insert a function block rung whose function code is 35 at the current rung number.

Use these commands to add rungs to the demonstration program.

3. First, display rung 4:

   a. Press F10.

   b. Type **D 4** Enter.

4. Now type **I** Enter and the new rung appears on the screen. The rung that used to be 4 became 5, 5 became 6, and so on.

Rung 4 may be edited as discussed previously. The cursor should be at the heading.

5. Type **new logic rung**.

6. Move the cursor to the contact using F2 and edit the contact:

   a. Make this a normally closed type by pressing ***/***.

   b. Make the tag *newfb* by pressing **T**, then typing **newfb** Enter.

7. Move the cursor to the coil and notice that the coil already has a tag. When a rung is inserted, the editor automatically creates a unique tag for its output. This tag can be used, but normally a new one is assigned. Make it *new/rung*: Press **T**, type **new/rung** Enter. Practice changing this rung.

8. Insert a timer after this rung:

   a. Press F10 to move the cursor to the command line. It takes longer for the command line to appear because the editor is busy inserting the new rung.

   b. Type **I 35 A** Enter. The display changes to rung 5 and its function code is 35.

9. Edit the timer rung:

   a. Type **new function block rung** in the heading and press F2.

   b. Type **new/rung** for the input tag and press Enter twice.

   c. Type **1.0** for the time delay.

   d. Move the cursor to the coil by pressing F2 and make its tag *newfb* by pressing **T**, then typing **newfb** Enter.

Practice by making other changes to this rung.

Displaying Rungs

Previously a command like **D 5** was used to display a rung. There are several other commands that select a rung for display. Move the cursor to the command line and look at the help display. It displays the following list:

D *num* Display rung number *num.*

D *tag* Display rung which defines *tag.*

D *@* Display rung *@*.

DB *nnnn* Displays rung with block number *nnnn*.

N Display next rung.

P Display previous rung.

The first command is the one used previously. The second command displays the rung which has an output named *tag*. Try the command **D new/rung**. Rung 4 appears.

In the third command @ represents one of the following pointer characters:

{ Points to first rung.

} Points to last rung.

. Points to current rung.

[Points to rung specified by mark command.

] Points to rung specified by mark command.

For example:

D } Enter Displays the last rung.

D { Enter Displays the first rung.

D . Enter Displays the current rung.

Deleting Rungs

This section discusses the commands used to delete rungs. In the help display the following appear:

DELETE Delete current rung.

DELETE *@ @* Delete from @ through @.

The first command deletes a single rung, the one on display, and renumbers the following ones. The second command deletes a block of rungs (and renumbers the following ones). The start and end of the block are specified by using the pointer characters:

{ First rung.

} Last rung.

. Current rung.

[[mark.

]] mark.

For example:

DELETE { . Enter — Deletes from rung one through the current rung.

DELETE . } Enter — Deletes from the current rung through the last rung.

The pointers [and] are set by the mark command. The command **M [** assigns the [pointer to the current rung; **M]** assigns] to the current rung. To delete rungs 50 through 59, any of the following sequences of commands could be used:

D 50 Enter **M [** Enter **D 59** Enter **DELETE [.** Enter

D 59 Enter **M]** Enter **D 50** Enter **DELETE .]** Enter

D 59 Enter **M]** Enter **D 50** Enter **M [** Enter **DELETE [ ]** Enter

Save Ladder Program

In the first example with the editor, the name of a disk-resident file, ***pcdem***, was used. This file contained the ladder program to be edited. The editor then copied the program from disk into memory. The editing completed so far has affected only the memory version of the program. The disk file has not been changed.

To undo all the changes made in this editing session, simply stop the editor and leave the disk file unchanged. This also means all the changes would be lost if some unusual event occurred, such as a power failure. For this reason, it is a good idea to write the current version of a program from memory to disk from time to time as it is edited.

The following save commands appear in the command help display:

SAVE — Write memory to disk (current file name).

SAVE *file_name* — Write memory to disk (file name = *file_name*).

The first command saves the current version of the ladder program from memory back to disk with the same file name as when it was loaded. The second command saves the current version of the program from memory to a file specified by *file_name*.

This second command is commonly used to make a backup copy of a file. It is recommended to do the following: When a file is first read into memory, but before making a change to it, use the **SAVE** *file_name* command to make a backup copy. Use a name that is readily recognizable as a backup to a particular file. For example, for the file ***pcdem*** the command **SAVE PCDEM** could be used to make the backup copy. Then, as the program is edited, use the **SAVE** command occasionally to update the main file.

Function Block Language

The concepts and rules discussed so far can be thought of as a programming language - the INFI 90 ladder language. For INFI 90 there is another concept which can be thought of as the INFI 90 function block language. These languages are very similar. In fact, there is a one to one correspondence between rungs in a ladder program and function blocks in a function block program. The primary difference between the two is the way in which things are identified. In a ladder program, each rung is identified by a rung number; inputs and outputs are identified by tag names.

In a function block program each block is identified by a block number. The outputs of a block are also identified by block number. For example, for block number 100, the first output is block number 100. If it has a second output, it is 101 and so on. Since outputs are identified by block number, inputs must also be identified by block number. Another difference between the two types of programs is that the function block program does not contain any of the comments from the ladder program.

The native language of INFI 90 modules is function blocks. In general, the LADDER software automatically translates a ladder program into a function block program. For example, when a ladder program is downloaded to an INFI 90 module, the LADDER software translates rung numbers and tag names into block numbers.

There is one situation, however, where block numbers may be a concern. This has to do with a program in one module reading data from another module. There are several function codes that read data from another module. Function code 64 reads a list of digital values from another module in the same PCU and function code 45 reads a digital input from a module in a different PCU. In all these function codes the source data item is identified by block number. Even if the source module or the receiving module is programmed in ladder language, the data is identified by its block number. Therefore, for a module that acts as a source of data to other modules, it is necessary to know the block numbers of those blocks that are being read by other modules. In fact, it is desirable to be able to set the block

numbers of those blocks. The following command is in the command *HELP* display. It assigns the specified block number to the rung on display.

BLK *nnnn* Assign block number *nnnn* to current rung.

When a ladder program is translated into a function block program, block numbers are assigned in the following way. The rungs are scanned in rung number order, starting with rung one. For each rung, if the block number has not been assigned by the **BLK** command, it is assigned the next available block number (last block number used plus one) or range of block numbers (if the block has multiple outputs). The first available block number is zero. When assigning a block number with the **BLK** command, take into account the preceding rungs and use an available block number or else the program will try to fold back on itself.

Try assigning a block number to the current rung. Then clear the assignment by giving the **BLK** command with no number.

Compile a Ladder Program

The **COMPILE** command compiles a ***.LAD*** file into a ***.CFG*** for downloading into a target module. It also analyzes the ladder program to find certain kinds of errors. These errors are as follows:

Block # must be >> = num. An invalid block number was assigned to a rung (via the **BLK** command). The lowest available block number is *num*.

null tag. No tag has been assigned to the input of a rung.

tag is already defined. The same tag name is assigned to the output of two or more rungs.

tag is undefined. A tag is used as an input but is not defined (i.e., not assigned to the output of any rung).

tag is wrong data type. The tag is assigned to the output of a rung and to the input of a rung but the data type of the output is not the same as the data type of the input.

Try the **COMPILE** command now.

1. Type **COMPILE** Enter.

If no errors are found, a message is displayed saying *Compilation complete - no errors detected* and a ***.CFG*** file is created with the same name as the original. Otherwise, an error list appears similar to Figure A-12.

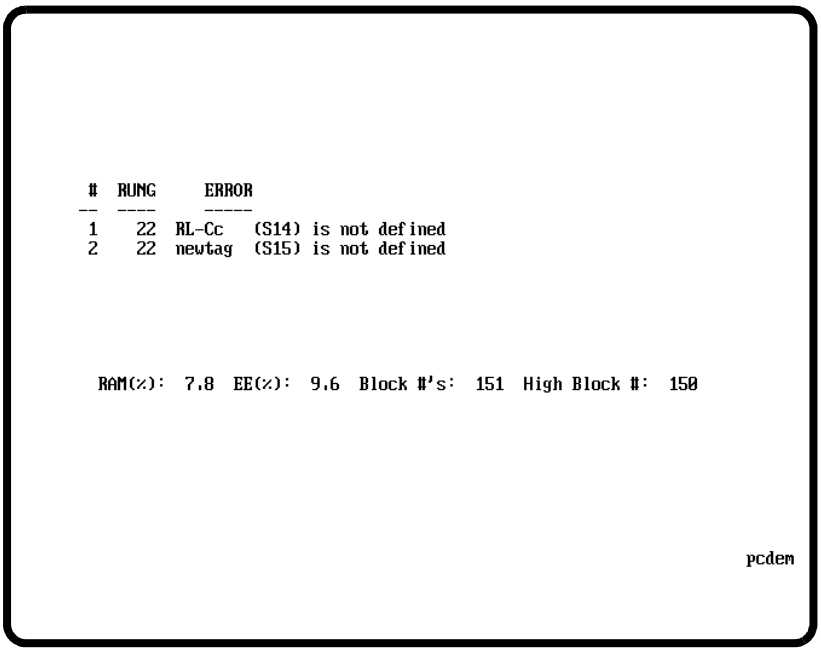


Figure A-12. Compilation Error List Display Example

Sometimes the error list is too long to display at once. In that case, the display may be scrolled up and down with the arrow and page keys.

2. If an error list is displayed, press ↓ and observe that the display scrolls down one position (error 2 is now at the top of the display).

3. Now press F10 to return the cursor to the command line. Pick one of the errors, display that rung and correct the error. Display a rung, (e.g., **D 23)**.

4. Now display the error list again by giving the command **ERR.** Notice that error 2 is still at the top of the display. This feature helps keep track of which error is being worked on. After an error is fixed, scroll that message off the display and work on the next one.

Quit the Editor

In the command help display are the following two quit commands:

DONE Done with file (write to disk)

DONE NOSAVE Done with file (do not write to disk)

Both commands quit the editor. The first command tells the editor to save the program from memory to disk (equivalent to a

SAVE) and then quit. The second command tells the editor to quit without doing a save.

1. Type **DONE NOSAVE** Enter. This returns to the editor opening screen, with the editor asking for a file name (Figure A-2). Since ***pcdem*** is still the active file, its name is displayed in the file name window.

2. At this point specify another file to be edited by typing its name or leave the editor by pressing Esc Press Esc and the main menu appears (Figure A-1).

DOWNLOAD A PROGRAM

This section explains how to transfer a ladder program from the computer to an INFI 90 processor module.

1. From the main menu screen press F3 to select *CONFIG*.

   NOTE: A screen may appear requesting INFI 90 interface data. If so, go to ***INFI 90 INTERFACE DATA*** in this section and return here. If not, continue with the next step.

2. The configuration program requests the name of a ladder file as shown in Figure A-13. Since the ***pcdem*** program is still in memory, its name is displayed in the file name field. The original, unedited version of ***pcdem*** is to be downloaded, not the edited version that is still in memory. Since ***pcdem*** is the active file name, the configuration program will not read it from disk. To remedy this, specify another file name and then specify ***pcdem***:

   a. Press Alt - F8.

   b. Type **x** Enter. Since there is no file named ***x*** on this disk, the message *CAN'T READ FILE* should appear along with a new menu.

   c. Press F2 (change name).

   d. Type **pcdem** Enter. The configuration program reads **pcdem** into memory. The display should look like Figure A-14.

If *NONE* was specified for interface type, the module status box will say *CAN'T COMMUNICATE WITH MODULE*. This is normal. Read the rest of this section, but do not perform the operations. However, if the *CAN'T COMMUNICATE WITH MODULE* message is displayed and *SPM* or *CIU* was specified, there is a problem that should be corrected before going on. Make sure that the processor module has the right module address (as specified in the title page) and that the module is in normal operating mode. The green LED indicator should be visible

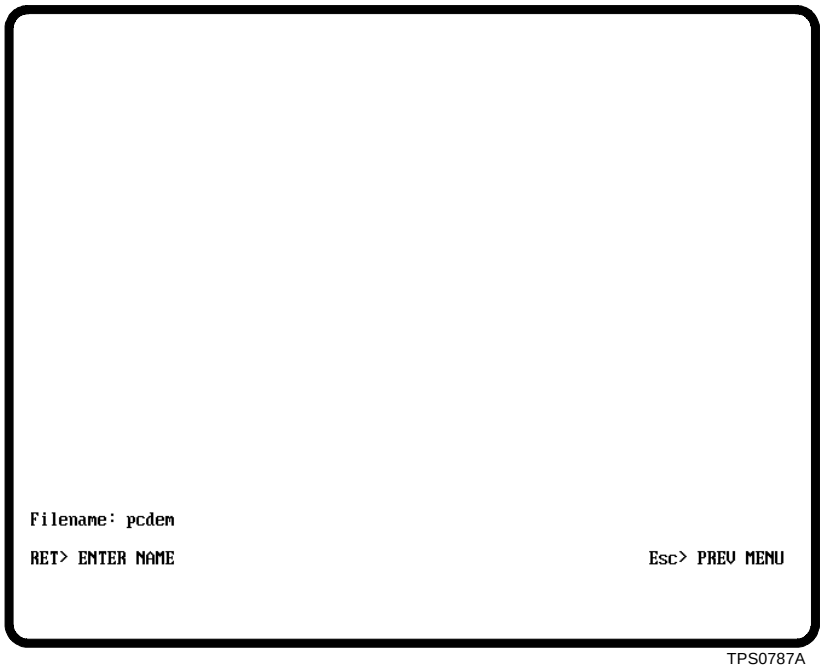


Figure A-13. Data File to Download

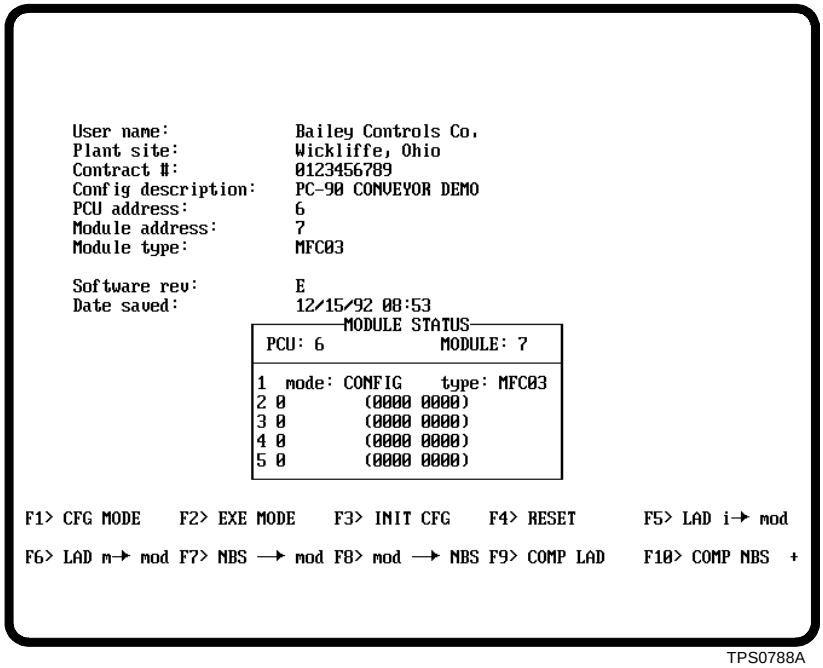


Figure A-14. Module Status Screen in Configure Mode

through the front plate of the module, either steady or flashing.

3. The menu items at the bottom of the display indicate the functions that can be performed now. Press F1 for *CFG MODE*. This tells the processor module to go to configure mode. When

the operation is completed, the module status display should indicate *CONFIGURE* mode and the green LED indicator on the processor module should be flashing. The module is idle in the sense that it is not running a program. It will now accept a new program or unrestricted changes to the existing one.

If the module was in execute mode when F1 was pressed, it performs certain shutdown actions before going to configure mode. For example, the digital outputs on all IMCIS02 I/O modules are de-energized (all analog output values are left unchanged) and all digital output modules are instructed to set their outputs to the predefined default values.

4. Press F5 to download the ladder program from the computer to the processor module. The program is sent to the module block by block (recall that the ladder program is translated into a function block program before being downloaded).

5. When the download is complete, press F2. This tells the module to go to execute mode and to begin running the new program. The module status display should indicate *EXECUTE* mode and the green status LED indicator on the processor module should be steady. This completes the download procedure. Press Esc and the main menu screen appears.

If, instead of *EXECUTE* mode, the status display indicates *ERROR* mode (and the green LED indicator on the processor module is flashing), then the processor module has found an error in the program and will not go into execute mode. This might mean that a step was omitted in the download procedure or that the ***pcdem*** file on disk has been modified.

6. Try the download procedure again. If the result is the same, restore ***pcdem*** to its original state using these steps and try again:

   a. Press Esc to return to the main menu.

   b. Press Alt - F10 to exit LADDER software and return to the DOS prompt.

   c. Type **COPY A:PCDEMZZZ.LAD A:PCDEM.LAD** Enter.

   d. Type **LAD** Enter to restart the LADDER software and then try downloading the file.

MONITOR

The monitor function has two major purposes:

- Allows observation of a processor module as it runs a ladder program. To be precise, the processor module actually runs a function block program which is translated from the

ladder program. What appears is the ladder program with live data from the function block program in the processor module.

- Allows the following changes to be made to a program as it runs: Tunable parameters of function block rungs can be changed and contacts and coils of logic rungs can be forced.

1. Press F1 to start the monitor. The monitor might request INFI 90 interface data to be specified. If so, refer to ***INFI 90 INTERFACE DATA*** of the tutorial and then return here. If this data has already been specified from earlier in this tutorial, the monitor does not request it again.

2. Next, the monitor requests the name of a ladder file as shown in Figure A-15. Since the ***pcdem*** program is still in RAM, its name is displayed in the file name window. To monitor another program, type in its name in place of ***pcdem***. In this case, use ***pcdem*** and just press Enter. The screen looks like Figure A-16.

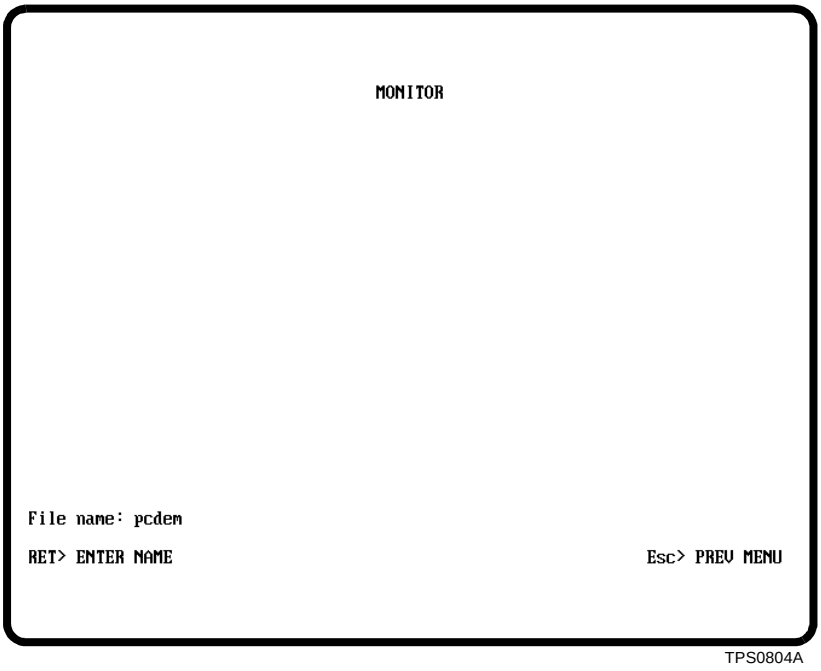
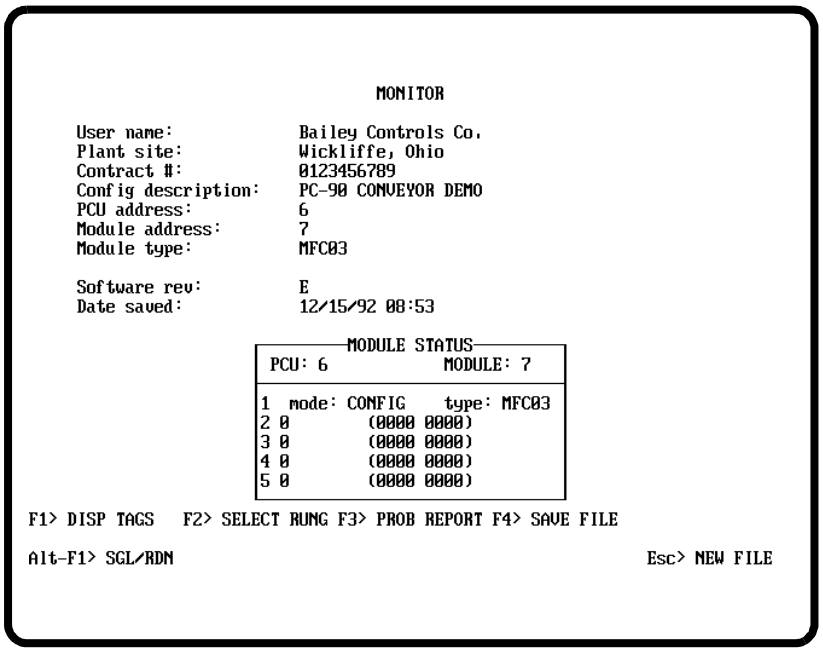


Figure A-15. Monitor File Name

3. Press F1 to display the tag list as shown in Figure A-17. This is a complete list of all the tag names defined in the ladder program. The tags are arranged in alphabetic order (uppercase letters precede their lowercase counterparts). In the demo program the entire list fits on one screen. If the list were too long, another item would appear on the menu that would allow scrolling through the list one screen at a time.



TPS0799A

Figure A-16. Monitor Module Status Screen

The cursor is in the upper left corner. Move the cursor with the arrow keys and notice that the field labeled *Descriptor:* changes. As the cursor points to a tag, the tag description appears in this field. For example, the description for tag *RL-CC* is *Raise Left @ Node #2*.

4. With the cursor at *RL-CC*, press F1 to display the rung which has *RL-CC* as an output. This is the same logic rung examined earlier with the editor (Figure A-18).

If *NONE* was specified earlier for interface type, the message ****CAN'T READ BLOCK**** appears near the bottom of the screen.

That is normal, continue. The following description of the monitor assumes that the *SPM* or *CIU* interface was specified. If *NONE* was specified, the actions of the monitor differ from this description.

If the *SPM* or *CIU* was specified and the *CAN'T READ BLOCK* message appears, it indicates a problem. Either the computer cannot communicate with the processor module (e.g., module is unplugged or its address is wrong) or the processor module is in error mode (refer to ***DOWNLOAD A PROGRAM*** in this tutorial).

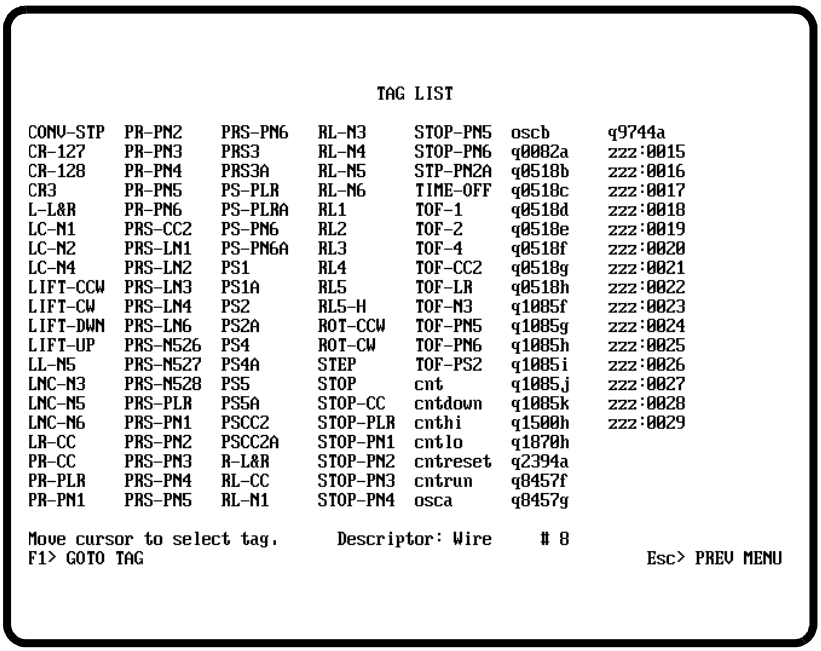


Figure A-17. Tag List Screen

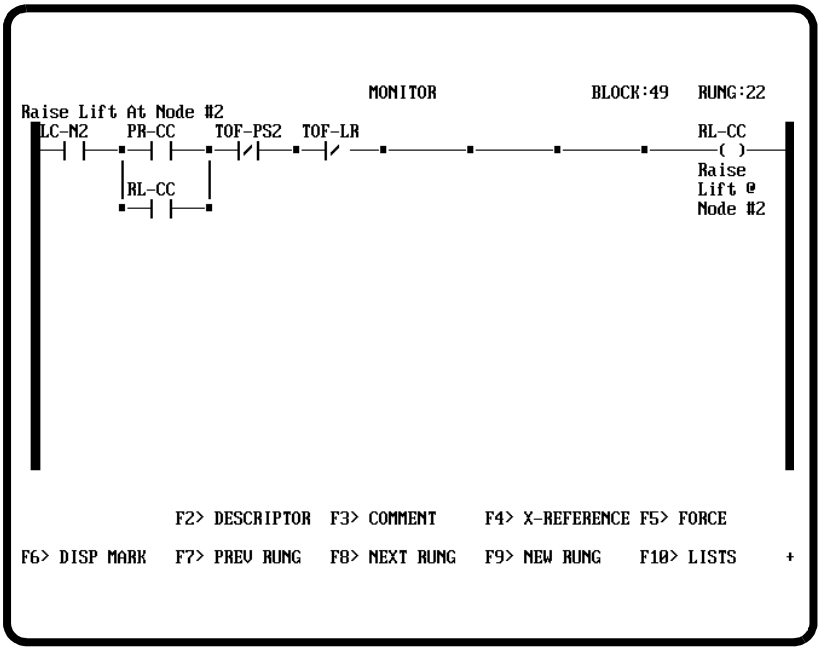


Figure A-18. Monitor Sample Rung Screen

Logic Rung

Rung 22 should be on the display. Observe the symbols for the contacts and coil. If a symbol looks normal (i.e., —| |— or —( )—), it indicates that the corresponding value is being

properly read from the processor module. If a symbol is shown in reverse video (i.e., light and dark reversed), it indicates that the corresponding value is bad (i.e., not being read from the module). If all the values are bad, it probably means that the processor module is not in execute mode.

The value for each contact and coil is indicated in the space inside the symbol, dark means zero and bright means one. Keep in mind that a contact represents a reference to an item of data (a coil or function block output) and, depending on the type of the coil, an operation to be performed on that item of data.

Examine contact *TOF-PS2*. Its type is normally closed (i.e., its operation is invert). Therefore, if the value of contact *TOF-PS2* is one (bright), then the value of output *TOF-PS2* must be zero (dark).

Now look at the menu at the bottom of the screen. On the far right side is the character **+**, indicating that the menu contains more items than are displayed. To see the rest of the menu press **Tab** or **+** several times. The item *F10> LISTS* is discussed in a later section.

F2> Descriptor

This changes the display so that each contact and the coil is replaced by its description.

1. Press **F2**. The checkerboard pattern appears to help identify the grid positions. Notice that in each grid position where there was a contact, there is now the description of the contact.

2. Press **F1** to return to the logic display.

F3> Comment

This displays the comment area. Press **F3**, observe the comment display, and then press **F1**.

F4> X-Reference

This function finds all the rungs that refer to the coil of the displayed rung. In this case, for example, find all the logic rungs that use *RL-CC* as a contact and all the function block rungs that use *RL-CC* as an input.

1. Press **F4**. Two lines of text appear near the bottom of the screen (Figure A-19). The first line indicates that *RL-CC* is referenced in rung 22 and the second line shows the heading of that rung (in this particular case, rung 22 refers to its own coil). The menu items are now:

F1> DISPLAY
F2> SEARCH
Esc> PREV MENU

2. Press **F2** to continue the search (i.e., to find the next rung that refers to *RL-CC*). In this case, it is rung 23 and the heading of rung 23 is *Check if Node #2 Lift is Raised.* Each time **F2**

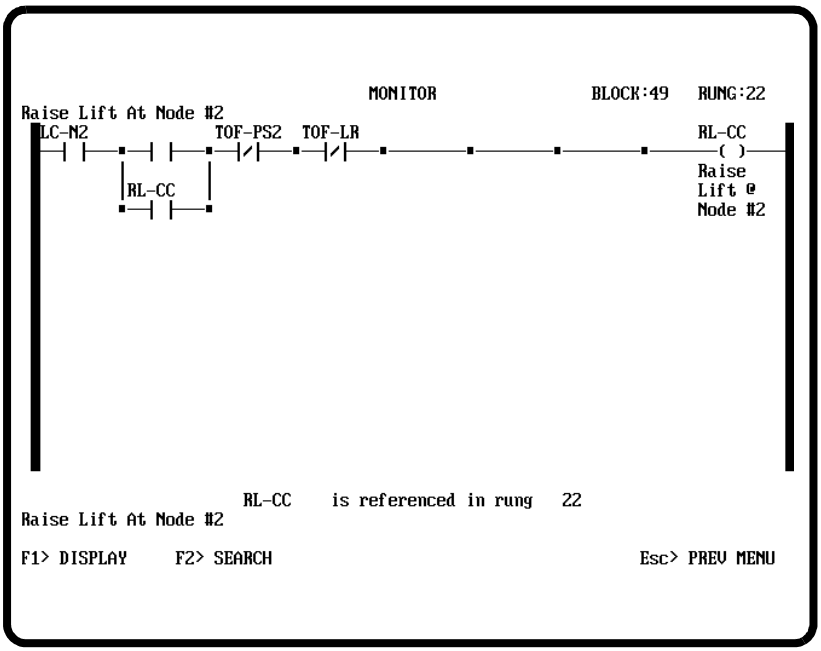


Figure A-19. Monitor X-Reference Screen

is pressed, another rung number and its heading are displayed. When all the rungs which reference *RL-CC* have been displayed, the message ****NO MORE REFERENCES**** appears. Pressing F2 again starts the sequence.

3. During the search, pressing F1 displays the currently identified rung. Press F2 until rung 23 is identified. Now press F1 and rung 23 appears. Rung 23 refers to *RL-CC*. Return to rung 22 by pressing F7.

4. Terminate the cross reference search at any time by pressing Esc. To demonstrate this, press F4 to enter cross reference mode and then Esc to terminate it.

F5> Force

This function forces a contact or coil. Force means to substitute a constant value in place of the normal dynamic value.

1. Press F5 to enter the force mode, shown in Figure A-20. Initially the cursor points to the coil.

2. Observe the menu and then press F2 to force the coil closed (i.e., value = 1). Notice that the inside of the contact turned light (indicating that its value is now 1) and that the tag name became brighter (indicating that the coil is forced).

3. Press F3 to force the coil open (i.e., value = 0). The inside of the coil turns dark (indicating that its value is now 0).

4. Press F1 to return the coil to normal and the tag returns to normal intensity.

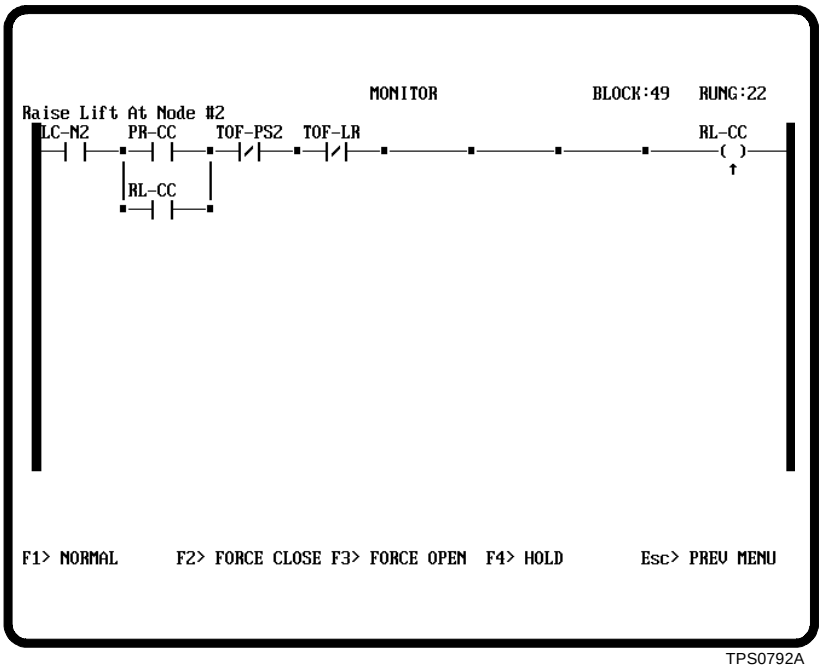


Figure A-20. Monitor Force Screen

5. Press F4 to force the coil to hold at its current value. Press F1 to return it to normal.

6. The cursor can be moved by using the arrow keys. Experiment with the cursor movement, then point it to one of the contacts.

7. Press F2 to force it closed (1).

8. Press F3 to force it open (0). Press F1 to return it to normal. Force several of the contacts, then press Esc to exit force mode.

F6> Disp Mark This function displays the list of marked rungs. The list is arranged in a 10 × 10 grid. The display shows only a 2 × 5 portion of the list. The entire list may be viewed by scrolling the display using any of the following keys: The four cursor keys, Page Up, Page Dn, Home, End, Ctrl - →, and Ctrl - ←

F7> Prev Rung This causes the display to move to the previous rung (i.e., current rung minus one). Press F7 to display rung 21.

F8> Next Rung This causes the display to move to the next rung (i.e., current rung plus one). Press F8 to display rung 22.

F9> New Rung This selects any rung for display. Specify the rung by its rung number, one of its output tags, or its block number.

1. To display rung 8, press F9, then type in **8** Enter.

2. To display tag RL-CC, press F9, then type **RL-CC** Enter.

3. To display block number 48 press F9 then type **#48** Enter. Rung 21 (block 48) is displayed.

M> Rung and G> Go To Mark

The mark command can mark up to 100 rungs for easy recall with the go to mark command.

1. Press **M** and the following prompt appears: *Select marker number (0-99)*.

2. Enter a value from 0 to 99 and that number marks the current rung.

3. To demonstrate the recall, display some other rung (e.g., press F8).

4. Press **G** then type **1** Enter and the marked rung is recalled.

T> Trace Input and O> Trace Out

These two commands provide a convenient method of tracing the flow of data into a rung. **T** command steps from the current rung to any one of its inputs. Successive use of **T** trace out one particular path of data into the rung. Once a trace path has been established with **T**, **O** allows a trace back (i.e., output to input) along that path one step at a time.

As an example, use rung 22.

1. To trace the data that feeds into contact *PR-CC*, press **T** and the cursor appears at the upper left contact.

2. Near the bottom of the screen appears the prompt *move cursor to desired contact*. Use the arrow keys to move the cursor to *PR-CC*.

3. Press Enter and rung 21 appears (coil of rung 21 is *PR-CC*). This establishes the first link in the trace path.

4. Since rung 21 has two contacts (PSCC2 and TOF-CC2), one or the other must be chosen for the next link. Choose PSCC2. Press **T** and Enter and rung 18 appears. It has only one contact (PRS-CC2), so there is no choice.

5. Press **T** and Enter and rung 71 appears. Since this is a digital input group function block and therefore a source of data, this is the end of one trace path.

6. Step back along this path. Press **O** and rung 18 appears.

7. Press **O** again and rung 21 appears.

8. Instead of stepping back to rung 22, continue the trace but along a different path (i.e., *TOF-CC2* instead of *PSCC2*). Press **T**, move the cursor to *TOF-CC2*, and press Enter. Rung 20 appears. It has only one input (*PSCC2A*).

9. Continue the trace by pressing **T** and Enter. Rung 19 appears. It has only one input (*PSCC2A*).

10. Continue by pressing **T** and Enter. Rung 18 appears. It has only one input, *PRS-CC2*.

11. Press **T** and Enter and rung 71 appears. This is the end of another path.

12. Step back along this path. Press **O** to go back to rung 18.

13. Continue pressing **O** to go back to rung 19, then rung 20, then rung 21, then back to the starting point, rung 22.

Function Block Rung

This section deals with the monitoring of a function block rung. Display rung 80 (F9 **80** Enter), which should look like Figure A-21. As with the logic rung, the status (good or bad) of an input or output is indicated by the normal or reverse video attribute of the input or output symbol. Normal video indicates that the status is good (i.e., the value is being read from the processor module). Reverse video indicates that the status is bad (i.e., the value is not being read from the module).

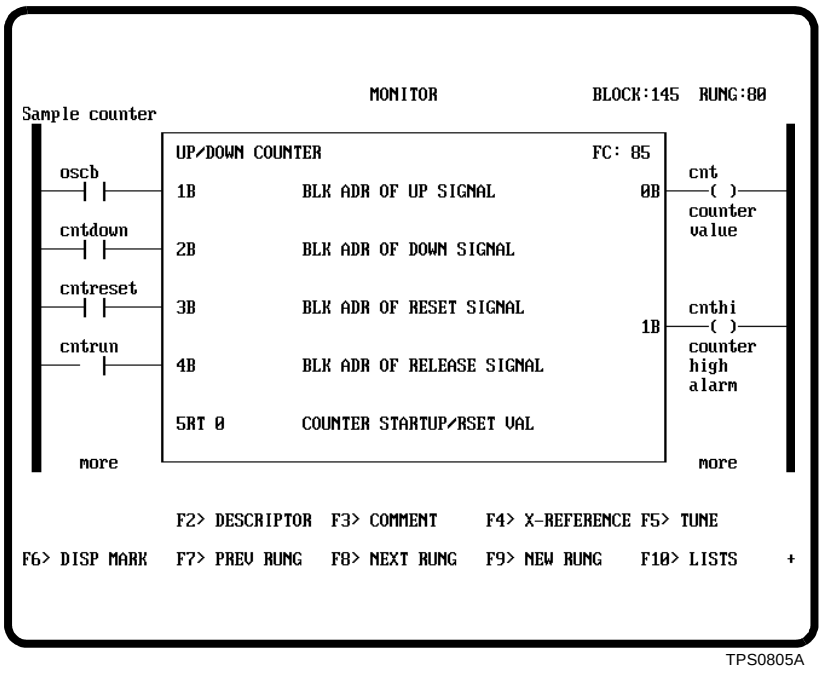


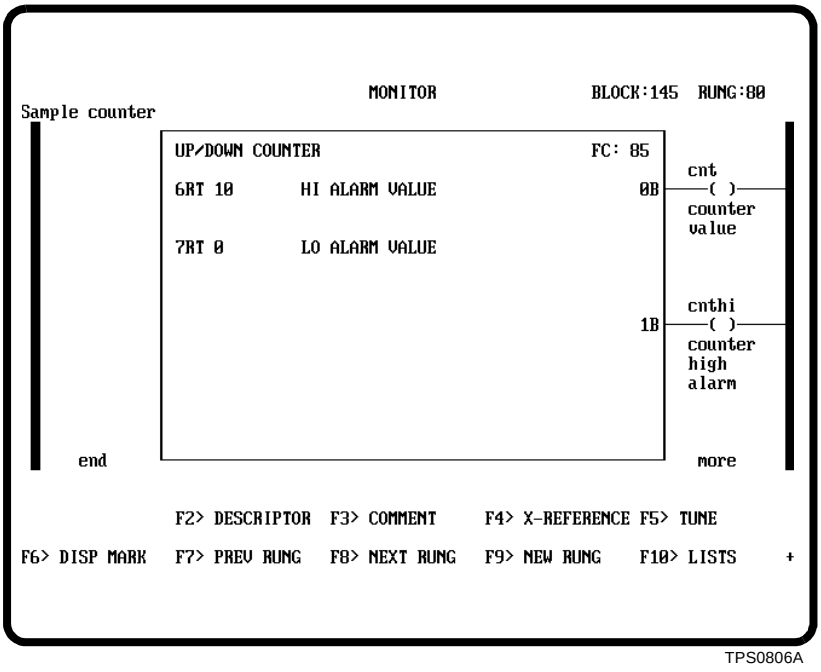
Figure A-21. Monitor Function Block Rung (More) Screen

The value of an input or output is indicated in the space inside the symbol. Boolean input and output values are indicated in the same way as contacts and coils - dark indicates zero and light indicates one. For inputs and outputs with data type real,

the numeric value of the input or output is displayed inside the symbol.

1. Examine the two *more* indicators on the display. One is brighter than the other. Alternately, press ← and → and the brightness shifts from side to side.

2. The brighter *more* (or *end*) indicates which side may be scrolled (via the ↑, ↓, Page Up, and Page Dn keys). Experiment by scrolling each side. Figure A-22, for example, shows the left side scrolled to the end.



TPS0806A

Figure A-22. Monitor Function Block Rung (End) Screen

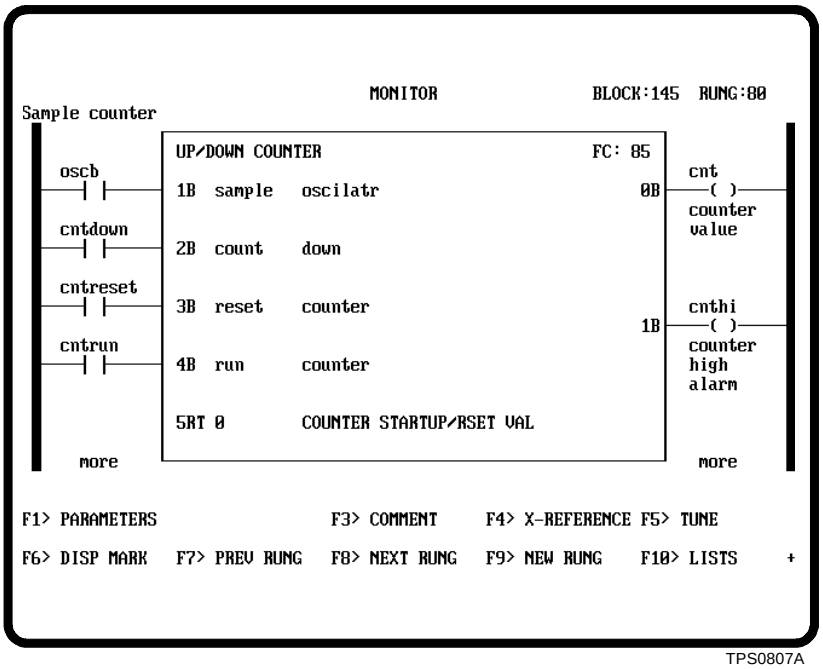
3. Next look at the menu (use the Tab or + key to scroll the menu). It is very similar to the menu for the logic rung and most functions are identical.

F2> Descriptor

As for the logic rung, this function displays the description associated with each input. In this case, however, the input description is displayed in place of the input name.

1. Press F2 and observe that *BLK ADR OF UP SIGNAL* is replaced by *sample oscilatr*. This is the description for tag *oscb* shown in Figure A-23.

2. Press F1 to return to the normal display.



TPS0807A

Figure A-23. Monitor Descriptor Screen

F3> Comment This function displays the comment area.

F4> X-Reference This is very similar to cross reference for logic rungs. The only difference is that a logic rung has only one output and a function block may have more than one. For a function block the cross reference refers to the output that is currently visible at the top of the display.

1. To demonstrate this, scroll the outputs so that *cnt* is displayed at the top.

2. Press F4. The message ****NO MORE REFERENCES**** appears indicating that the output *cnt* is not referenced in any rung.

3. Press Esc to terminate cross reference mode.

4. Scroll the outputs so that *cnthi* is on top, then press F4. The cross reference display indicates that *cnthi* is referenced in rung 82.

5. Press Esc to terminate cross reference mode.

F5> Tune This function changes the value of a function block tunable parameter. Tuning is similar to editing but differs in two important ways. During editing, any parameter or input may be changed. During tuning, only tunable parameters may be changed.

Editing is done off-line. This means that the changes made to the ladder program affect only the version of the program that resides in the personal computer. To get those changes into the processor module, the edited program must be downloaded.

Tuning is done on-line. This means that the tuning changes are made not only to the version of the ladder program that resides in the personal computer but also to the version of the program that resides in the processor module. The changes are made to the program as it is being run by the module.

1. To demonstrate, press F5. The cursor (and data window) appear at specification 5 (*5* is the specification number, *R* means real data type, and *T* means tunable).

2. Move the cursor to other tunable specifications with the ↑, ↓, Page Up and Page Dn keys. As practice, change specification 6 (*High Alrm Value*) from 10.0 to 5.0.

Notice that when F5 was pressed, all the input and output symbols changed to bad status. This does not indicate a problem. When the monitor is in tune mode, it does not read data from the processor module as it normally does. Instead, it displays 0 and bad status for all the values. The values will be displayed as normal when tune mode is exited.

3. As new values are entered on the screen, only the version of the program in the personal computer memory is changed. The program in the processor module is not immediately changed. Press Esc to terminate tune mode and all the new specification values are sent to the processor module.

F6> Disp Mark This function displays the list of marked rungs. The list is arranged in a 10 × 10 grid. The display shows only a 2 × 5 portion of the list. The entire list may be viewed by scrolling the display using any of the following keys: The four cursor keys, Page Up, Page Dn, Home, End, Ctrl - →, and Ctrl - ←

F7> Prev Rung This causes the display to move to the previous rung (i.e., current rung minus one) which is the same as F7 for logic rung.

F8> Next Rung This causes the display to move to the next rung (i.e., current rung plus one) which is the same as F8 for logic rung.

F9> New Rung This allows any rung to be selected for display (same as F9 for logic rung).

M> Mark Rung and G> Go To Mark These allow a rung to be marked and recalled (same as **M** and **G** for logic rung).

T> Trace Input and O> Trace Out These allow the flow of data into a rung to be traced (same as **T** and **O** for logic rung).

Lists

The *F10> LISTS* menu item is discussed in a separate section because it pertains not just to an individual rung but rather to the entire program. Press F10 and the following menu appears.

F1> ALL TAGS
F2> FORCED
F3> FUNC CODE
Esc> PREV MENU

These menu items are explained in the following subsections:

F1> All Tags

Displays the tag list which is a list of all the tag names defined in the ladder program.

1. Press F1. This is the same tag list function discussed in the beginning of ***MONITOR*** in this tutorial (Figure A-17).

2. Press Esc to return to the *LISTS* menu.

F2> Forced

This function displays the forced rung list. This is a list of rung numbers of all the rungs which have contacts or coils that are forced.

1. Press F2. If no rungs are forced, *End of List* appears. Otherwise, the forced rung list appears. See Figure A-24 as an example.

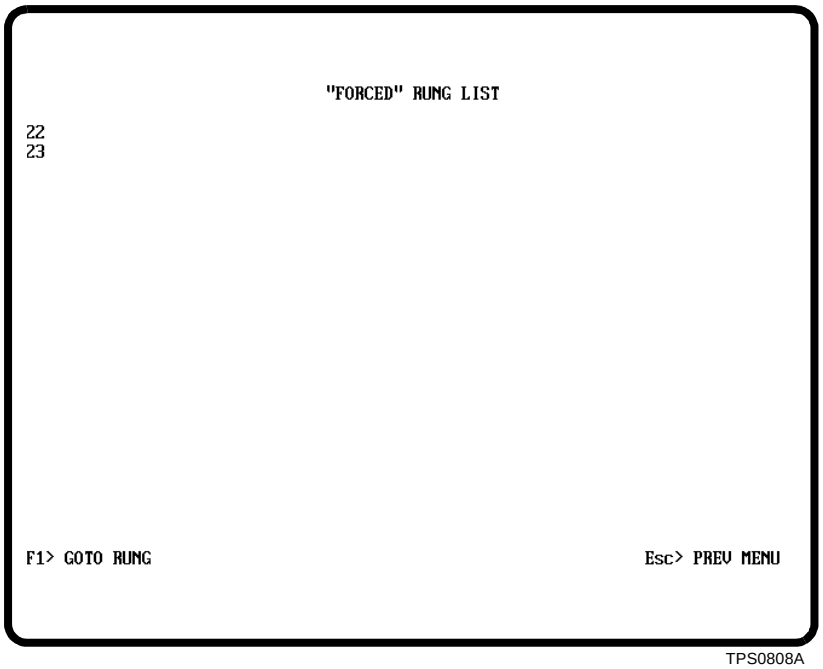


Figure A-24. Forced Rung List

2. If the list is too long to fit on one screen, F2 scrolls through the list.

3. Terminate the display and return to the *LISTS* menu by pressing Esc or select a rung to display by pressing F1.

4. Press F1 and the prompt *Enter rung number or tag (or Esc)* appears.

5. Any rung number or tag name may be specified for display (not just forced rungs). For example, type **22** Enter and rung 22 appears as shown in Figure A-18.

6. Press F10 to return to the *LISTS* menu.

F3> Func Code

This function allows a function code to be specified and then displays a list of all the rungs that have that function code.

1. Press F3 and the following prompt appears: *Function code:*.

2. To find out which rungs are timers (function code 35), type **35** Enter and the list appears as shown in Figure A-25.

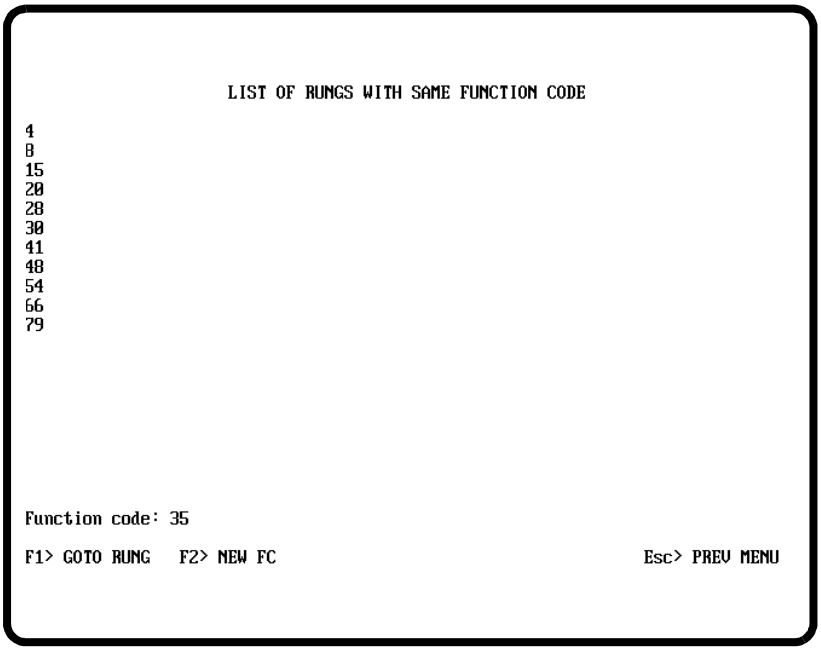


Figure A-25. List of Rungs with Same Function Codes

3. Select a rung to display (F1), select another function code (F2), or return to the *LISTS* menu (Esc). For example, to select another rung, press F1 and type **5** Enter.

Additional Practice

This concludes the discussion of the monitor. For additional practice, the following steps are some suggestions.

1. Display rung 80. This is an up or down counter which should be periodically counting up and resetting to 0 (observe output 0).

2. Mark the rung.

3. Scroll through the specifications and outputs.

4. Change the period by tuning specification 6.

5. Use the trace function to find the logic rung which drives the up signal and force it to a constant value. The counter should stop counting. Undo the force to start the counter again.

Quit the Monitor

To leave the monitor, display any rung and press Esc. The screen should look like Figure A-16 with these menu items:

F1> Disp Tags This function displays the tag list and returns to the monitor (as discussed at the beginning of ***MONITOR*** in this tutorial).

F4> Save File Any changes made to the ladder program while in the monitor (i.e., by tuning or forcing) affected the version of the program in the processor module and the version in the personal computer memory. The version of the program on disk, however, is not automatically updated. This function writes the program from memory to disk.

Esc> New File This function exits from the monitor or selects another file for monitoring.

1. Press Esc and the monitor requests another file name. To monitor another file, enter its name and return to the monitor.

2. Exit the monitor by pressing Esc. The main menu appears (Figure A-1).

PRINT

The primary function of this program is to print a listing of a ladder program. From the LADDER main menu, press F4 to select *PRINT*. The *PRINTER* menu appears as shown in Figure A-26.

NOTE: Refer to Section 3 for printer start-up options.

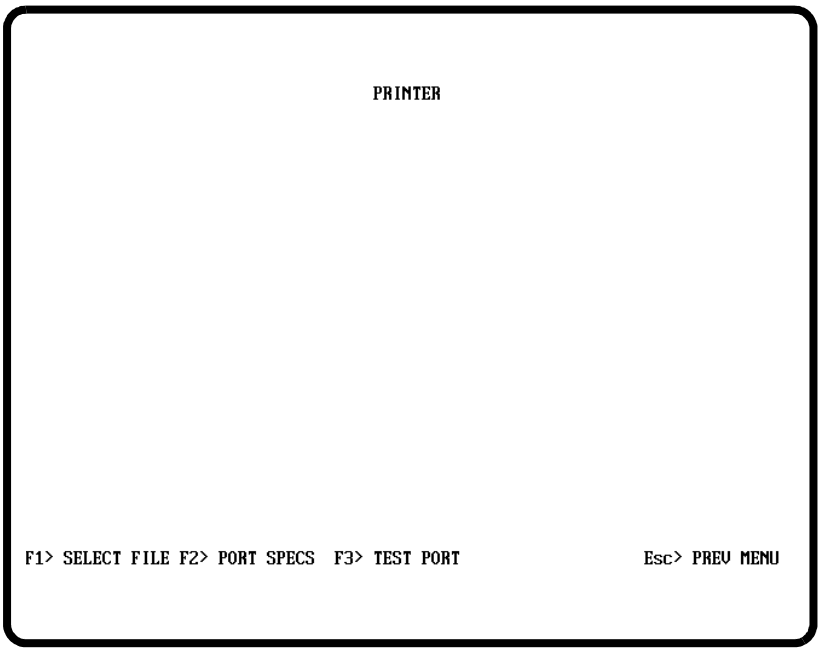


Figure A-26. Printer Menu

F2> Port Specs

This function sets certain attributes of the printer that are used to print the ladder program. Press F2 and the *SPECIFY PRINTER INTERFACE* screen appears as show in Figure A-27.

1. The first attribute is labeled *Port*. This specifies whether the printer has a parallel interface (Centronics type) or a serial interface (RS-232-C). Press Space Bar to toggle between *LPT1*, *COM1* and *COM2*. Select the appropriate value for your printer.

2. The remainder of the attributes apply only to a serial interface. If parallel was specified, skip this paragraph. Otherwise, use

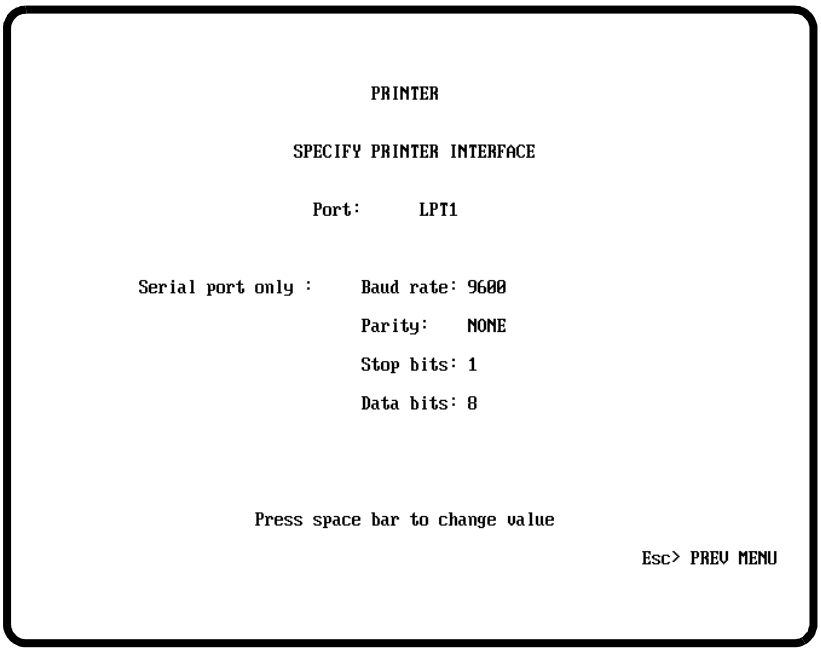
↑ or ↓ to move the cursor. Select the appropriate value for each attribute by pressing Space Bar until the desired value appears.

3. Press Esc to return to the *PRINTER* menu.

F3> Test Port

This function writes a test pattern to the printer. The purpose is to check that the printer is properly set up and connected. Press F3 and the printer should advance the paper to the start of a new page and print the following text on one line:

<< = column 1 column 132 = >>



TPS0795A

Figure A-27. Specify Printer Interface Screen

F1> Select File

Use this function to specify which ladder program and what parts of it are to be printed.

1. Press F1 and the print program requests a file name. Type **pcdem** Enter.

2. The file title page appears. Press Space Bar and the screen shown in Figure A-28 appears. Refer to Table 7-1 to fill in the fields appropriately.

3. After specifying what to print and the destination, press F1 to initiate printing.

4. When the printing is complete, press Esc. The print program asks for another file name. Press Esc again to return to the *PRINTER* menu.

5. Press Esc one more time to return to the LADDER main menu.

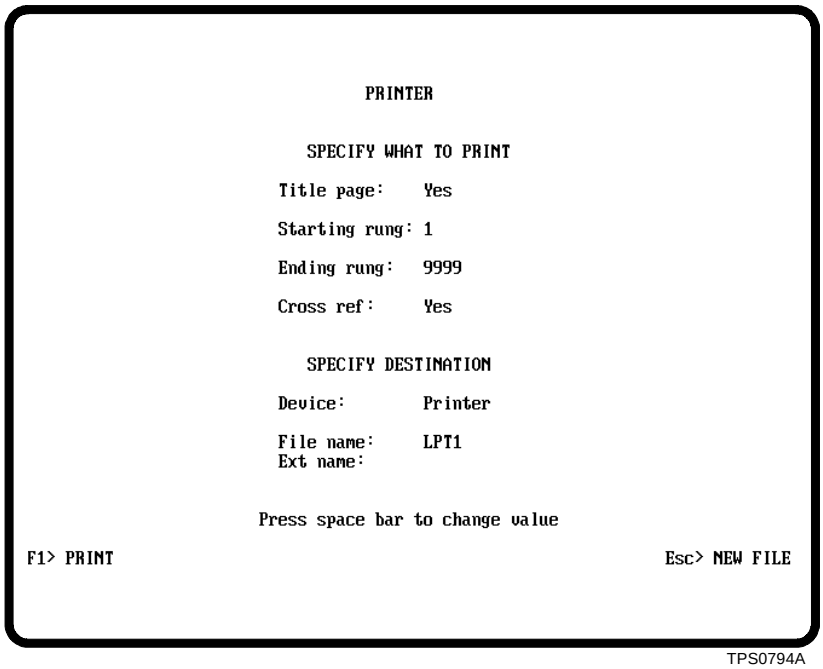


Figure A-28. Print Specification Screen

The following is a sample listing for ***pcdem*** (rungs 78-80).

```
FILENAME: pcdem
          User name:  Bailey Controls Co.
          Plant site:  Wickliffe, Ohio
          Contract #:  0123456789
          Config description:  PC-90 CONVEYOR DEMO
          PCU address:  00
          Module address:  10
          Module type:  MFC01
          Software rev:  E
          Date/time saved:    10-02-92  9:33


BLK: 143 Sample oscillator (first rung)                              RUNG  78
$ oscb                                                        osca  $
$---]\[-------------------------------------------------------------( )--------- $                 79
$ sample                                                      sample $
$ oscilatr                                                    oscilatr $
$                                                                    $
$                                                                    $

BLK: 144 Sample oscillator (second rung)                             RUNG  79
          ------------------------------------------------------------------
          | TIME DELAY OR PULSE                    FC: 35  |
  osca    |                                                | oscb                  78   80
 ----| |----- | 1B  sample    BLK ADR OF INPUT          0B |----( )-----
          |     oscilatr                                   | sample
          |                                                | oscilatr
          |                                                |
          | 2I   0        0=PULSE OUTPUT 1=TIMED           |
          |               OUT 2=TIMING                     |
          |                                                |
          | 3R T 2.00     TIME DELAY (IN SEC)              |
          |                                                |
          |                                                |
          ------------------------------------------------------------------
```

BLK: 145 Sample counter RUNG 80

```
                ------------------------------------------------------------------
                | UP/DOWN COUNTER                          FC: 85  |
  oscb          |                                                  | cnt
 ----| |-----   | 1B  sample    BLK ADR OF UP SIGNAL            0 |-((     ))-
                |       oscilatr                                   | counter
                |                                                  | value
cntdown         |                                                  |
----| |-----    | 2B  count      BLK ADR OF DOWN SIGNAL            |
                |       down                                       |
                |                                                  | cnthi                                  82
cntreset        |                                               1B |----( )-----
----| |-----    | 3B  reset      BLK ADR OF RESET SIGNAL           | counter
                |       counter                                    | high
                |                                                  | alarm
cntrun          |                                                  |
----| |-----    | 4B  run        BLK ADR OF RELEASE SIGNAL         |
                |     counter                                      | cntlo
                |                                               2B |----( )-----
                |                                                  | counter
                | 5R T 0.00     COUNTER STARTUP/RSET VAL           | lo
                |                                                  | alarm
                |                                                  |
                | 6R T 10.00    HI ALARM VALUE                     |
                |                                                  |
                |                                                  |
                | 7R T 0.00     LO ALARM VALUE                     |
                |                                                  |
                |                                                  |
                ------------------------------------------------------------------
```

TAG	TAG DESCRIPTION	RUNG	BLK	FUNC	REFERENCES
CONV-STP	Wire # 8	74	132	84DIS	3
CR-127	TRANSFER Pallet	60	97	110	62
CR-128	Pass Pallet	61	98	110	63
CR3		5	32	110	76*
L-L&R	Lower Lift & Rotate	36	73	110	77*
LC-N1	Lift Not Clear Node #1	10	37	110	11 12
LC-N2	Lift Clear Node #2	17	44	110	22 25
LC-N4	Lift Clear Node #4	51	88	110	56 57
LIFT-CCW	Wire # 10121	73	128	84DIS	33
LIFT-CW	Wire # 10120	73	127	84DIS	33
LIFT-DWN	Wire # 10118	73	125	84DIS	45
LIFT-UP	Wire # 10119	73	126	84DIS	33
LL-N5	Lower Lift	63	100	110	62 63 76*
LNC-N3	Lift Not Clear Node #3	43	80	110	45 50
LNC-N5	Lift Not Clear	58	95	110	59 60 61 62 63
LNC-N6	Lift NOT Clear	68	105	110	69 70
LR-CC	Lift Raised X Conv	23	50	110	25
PR-CC	Pallet @ Cross Con N#2	21	48	110	22 62
PR-PLR	Pallet @ Pre Lift Rotate	31	58	110	
PR-PN1	Pallet @ Prestop Node #1	9	36	110	70
PR-PN2	Pallet @ Pre Node #2	16	43	110	12
PR-PN3	Pallet @ Pre Node #3	42	79	110	32 33
PR-PN4	Pallet @ Pre Node #4	49	86	110	50
PR-PN5	Pallet Ready	55	92	110	57
PR-PN6	Pallet @ Pre Node #6	67	104	110	63
PRS-CC2	Wire # 10114	71	111	84DIS	18 24
PRS-LN1	Wire # 10112	71	109	84DIS	10
PRS-LN2	Wire # 10115	71	112	84DIS	17
PRS-LN3	Wire # 10122	71	114	84DIS	43
PRS-LN4	Wire # 10124	72	116	84DIS	51
PRS-LN6	Wire # 10130	72	122	84DIS	68
PRS-N526	Wire # 10126	72	118	84DIS	58
PRS-N527	Wire # 10127	72	119	84DIS	60
PRS-N528	Wire # 10128	72	120	84DIS	61
PRS-PLR	Wire # 10116	71	113	84DIS	26
PRS-PN1	Wire # 10111	71	108	84DIS	6 11
PRS-PN2	Wire # 10113	71	110	84DIS	13 25
PRS-PN3	Wire # 10117	73	124	84DIS	39 44
PRS-PN4	Wire # 10123	73	131	84DIS	46 56
PRS-PN5	Wire # 10125	72	117	84DIS	52 59

TAG	TAG DESCRIPTION	RUNG	BLK	FUNC	REFERENCES
PRS-PN6	Wire # 10129	72	121	84DIS	64 69
PRS3	Prox Sw Pre Node #3	39	76	110	40 42
PRS3A		40	77	33	41
PS-PLR	L&R Prox Switch	26	53	110	29 31 32
PS-PLRA		29	56	33	30
PS-PN6	Pallet Ready	64	101	110	65 67
PS-PN6A		65	102	33	66
PS1	Pallet Ready @ Pre N#1	6	33	110	7 9
PS1A		7	34	33	8
PS2	Pallet Ready @ Pre N#2	13	40	110	14 16
PS2A		14	41	33	15
PS4	Pallet Ready @ PN #4	46	83	110	47 49
PS4A		47	84	33	48
PS5	Pallet Ready @ PN #5	52	89	110	53 55
PS5A		53	90	33	54
PSCC2	Pallet Ready @ CC #2	18	45	110	19 21
PSCC2A		19	46	33	20
R-L&R	Raise Lift & Rotate	35	72	110	77*
RL-CC	Raise Lift @ Node #2	22	49	111	22 23 24 75*
RL-N1	Raise Lift Node #1	12	39	110	11 75*
RL-N3	Raise Lift	50	87	110	43 50 75*
RL-N4	Raise Lift	57	94	110	56 76*
RL-N5	Raise Lift	62	99	111	59 62 63 76*
RL-N6	Raise Lift	70	107	110	68 70 76*
RL1		34	61	161	35
RL2		34	62	161	37
RL3		34	63	161	38
RL4		34	64	161	36
RL5		34	65	161	33 44
RL5-H	Step #5 Hold	44	81	110	44 45
ROT-CCW	Rotate CCW	38	75	110	77*
ROT-CW	Rotate CW	37	74	110	77*
STEP		33	60	110	34
STOP	Stop Conv	3	30	110	3 4 11 24 25 32 45 56 59 69
STOP-CC	Lower Stop @ X Conv	24	51	110	75*
STOP-PLR	Lower Stop @ Pre L&R	32	59	110	45
STOP-PN1	Lower Stop Pre Node #1	11	38	110	75*
STOP-PN2	Lower Stop MC #2	25	52	110	27 75*
STOP-PN3	Lower STOP @ PN #3	45	82	111	34 77*
STOP-PN4	Lower Stop	56	93	110	77*
STOP-PN5	Lower Stop	59	96	110	76*
STOP-PN6	Lower Stop	69	106	110	76*
STP-PN2A		27	54	33	28
TIME-OFF	SHUT DOWN CONV	4	31	35	3 5
TOF-1	Pre Node #1 Time Dly	8	35	35	9
TOF-2		15	42	35	16
TOF-4		48	85	35	49
TOF-CC2		20	47	35	21
TOF-LR		30	57	35	22 23 25 31
TOF-N3	Pre Node #3 Delay	41	78	35	32 45
TOF-PN5		54	91	35	55
TOF-PN6		66	103	35	67
TOF-PS2		28	55	35	22
cnt	counter value	80	145	85	
cntdown	count down	81	148	110	80
cnthi	counter high alarm	80	146	85	82
cntlo	counter lo alarm	80	147	85	
cntreset	reset counter	82	149	110	80
cntrun	12345678 90123456 7890123	83	150	50	80
osca	sample oscilatr	78	143	110	79
oscb	sample oscilatr	79	144	35	78 80
q0082a		75	140	83DOS	
q0518b		74	133	84DIS	
q0518c		74	134	84DIS	
q0518d		74	135	84DIS	
q0518e		74	136	84DIS	
q0518f		74	137	84DIS	
q0518g		74	138	84DIS	
q0518h		74	139	84DIS	
q1085f		34	66	161	
q1085g		34	67	161	

TAG	TAG DESCRIPTION	RUNG	BLK	FUNC	REFERENCES
q1085h		34	68	161	
q1085i		34	69	161	
q1085j		34	70	161	
q1085k		34	71	161	
q1500h		71	115	84DIS	
q1870h		72	123	84DIS	
q2394a		77	142	83DOS	
q8457f		73	129	84DIS	
q8457g		73	130	84DIS	
q9744a		76	141	83DOS	
zzz:0015	time of previous cycle234	1	15	82	
zzz:0016	time of current cycle	1	16	82	
zzz:0017	CPU time (%)	1	17	82	
zzz:0018	chkpnt overrun (cycles)	1	18	82	
zzz:0019	cyc time overrun	1	19	82	
zzz:0020	Time of day (hours)	2	20	90	
zzz:0021	Time of day (min)	2	21	90	
zzz:0022	Time of day (sec)	2	22	90	
zzz:0023	Time sync (1=ok)	2	23	90	
zzz:0024	Year	2	24	90	
zzz:0025	Month 2	25	90		
zzz:0026	Day of month	2	26	90	
zzz:0027	Day of week	2	27	90	
zzz:0028	Rev lev2	28	90		
zzz:0029		2	29	90	

INFI 90 INTERFACE DATA

This utility accepts and records interface data for communicating with an INFI 90 processor module.

1. If this section is entered from the ***DOWNLOAD A PROGRAM*** section or the ***MONITOR*** section, the *INFI 90 INTERFACE DATA* page should be on the screen (Figure A-29). Otherwise, the main menu (Figure A-1) should be on the screen. Press F6 to go to the interface data page (Figure A-29).

2. Specify the interface data that LADDER needs to communicate with an INFI 90 processor module (refer to Table 6-1).

3. When all the interface data is specified, press F1

If the computer is unsuccessful in trying to communicate with the SPM or CIU interface, the *CAN'T COMMUNICATE WITH MODULE* message is displayed indicating there is a problem that must be corrected before going on.

Make sure that the processor module has the right module address and that the module is in normal operating mode. A green LED indicator should be visible through the front plate of the module. Press F1 again. If still unsuccessful, try different values for the interface data or specify *NONE* (no interface connection).

If this section was entered from the ***DOWNLOAD A PROGRAM*** section or the ***MONITOR*** section, return there and continue. Otherwise, the main menu (Figure A-1) should be on the screen.

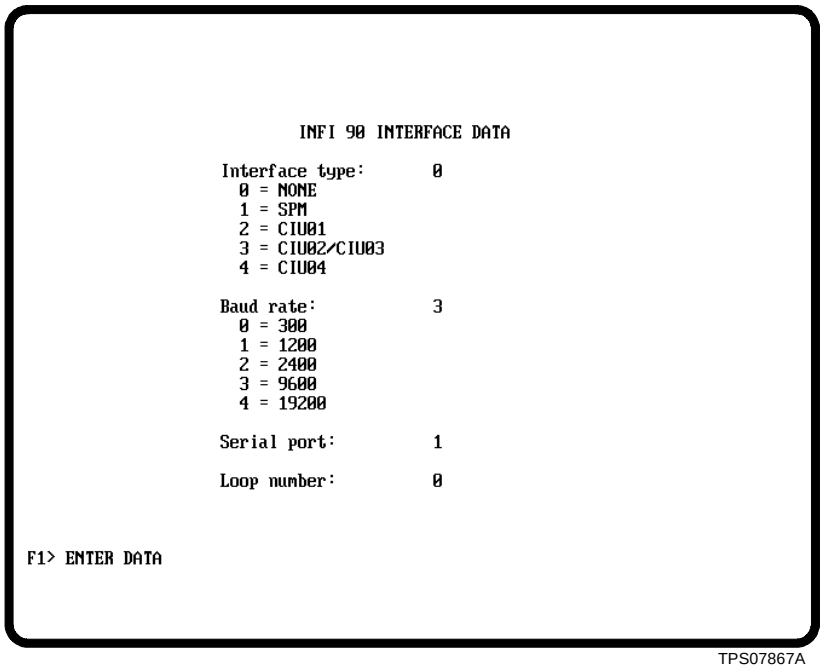


Figure A-29. INFI 90 Interface Data Screen

APPENDIX B - SUMMARY OF EDITOR COMMANDS

EDITOR COMMANDS

This appendix provides tables listing the editor utility commands. Refer to Table B-1 for a list of the editor command line commands. Refer to Table B-2 for text editing commands. These commands apply whenever text needs to be modified (e.g., heading, comments, tags, parameter values, etc.). Logic rung contacts are modified using the commands listed in Table B-3. Logic rung coils are modified using the commands listed in Table B-4. Function block rung inputs and parameters are modified using the commands listed in Table B-5. Refer to Table B-6 for commands used to modify function block outputs.

Table B-1. Editor Commands

Command	Description	Example
blk *nnnn*	Assign block number *nnnn* to current rung.	blk 100
c @ @	Copy rungs @ through @ starting at current rung number.	c [ ]
compile	Compile a ladder program for error checking.	compile
d *num*	Display rung number *num*.	d 20
d *tag*	Display rung which defines *tag*.	d RL-CC
d @	Display rung pointed to by @.	d }
db *nnnn*	Display rung with block number *nnnn*.	db 100
delete	Delete current rung.	delete
delete @ @	Delete rungs @ through @.	delete [ ]
done	Done with file (save to disk).	done
done nosave	Done with file (do not save).	done nosave
err	Display errors from the last compilation error listing.	err
fc *nnn*	Display list of all rungs with function code *nnn*.	fc 35
i	Insert a logic rung at current rung number.	i
i a	Insert a logic rung after current rung number.	i a
i *nnn*	Insert a function code rung with function code number *nnn* at current rung number.	i 35
i *nnn* a	Insert a function code rung with function code number *nnn* after current rung number.	i 35 a
if *file_name*	Insert file *file_name* at current rung number.	if mylad
m [	Mark current rung with [.	m [
m]	Mark current rung with].	m]
n	Display next rung (same as F8).	n
p	Display previous rung (same as F7).	p
save	Write program to disk using current file name.	save
save *file_name*	Write program to disk as *file_name*.	save mylad

Table B-1. Editor Commands (continued)

Command	Description	Example
tags	Display list of all tags.	tags
title	Display/edit title page.	title
xref *tagname*	Display inputs used by tag.	xref zzz:0020
xtag	Modify tags.	xtag
Alt-F7	Clear from cursor to end of line.	
Alt-F8	Clear line.	
Alt-F9	Save current command.	
Alt-F10	Restore saved command.	
F3	Insert logic rung after current rung (i a).	
F6	Repeat previous command.	
F7	Display previous rung.	
F8	Display next rung.	
F9	Activate/de-activate help display.	

NOTE: The symbol @ represents one of the following rung pointers:

{ = First rung
} = Last rung
. = Current rung
[= Set by m [
] = Set by m]

Table B-2. Text Editing Commands

Key Command	Action
Clear from cursor to end of line.	Alt - F7
Clear line.	Alt - F8
Delete character.	Del
Delete column in comment area.	Alt - F4
Delete line in comment area.	Alt - F6
Go to command.	F10
Go to next area.	F2
Go to previous area.	F1
Insert column in comment area.	Alt - F3
Insert line in comment area.	Alt - F5
Restore saved line.	Alt - F10
Save line.	Alt - F9
Toggle insert or overtype mode.	Ins

Table B-3. Modify Logic Rung Contact Key Commands

Key Command	Action
Delete column.	Alt - F4
Delete contact.	Del
Delete line.	Alt - F6
Go to area *n*.	Ctrl - Fn

Table B-3. Modify Logic Rung Contact Key Commands (continued)

Key Command	Action
Go to command.	F10
Go to next area.	F2
Go to previous area.	F1
Horizontal line.	-
Initialize.	Alt - I
Insert column.	Alt - F3
Insert contact.	Ins
Insert line.	Alt - F5
Modify tag.	T
Normally closed contact.	/
Normally open contact.	[
Open.	+
Restore line.	Alt - F10
Save line.	Alt - F9
Toggle left connect.	>
Toggle right connect.	<
Toggles between the four contact types.	?

Table B-4. Modify Logic Rung Coil Key Commands

Key Command	Action
Go to area *n*.	Ctrl - Fn
Go to command.	F10
Go to next area.	F2
Go to previous area.	F1
Modify descriptor.	D
Modify tag.	T

Table B-5. Modify Function Block Rung Inputs or ParametersKey Commands

Key Command	Action
Clear line.	Alt - F8
Clear to end.	Alt - F7
Delete character.	Del
Enter data.	Enter
Go to command.	F10
Go to next area.	F2
Go to previous area.	F1
Page down.	Page Dn
Page up.	Page Up
Toggle insert or overtype mode.	Ins

Table B-6. Modify Function Block Outputs Key Commands

Key Command	Action
Go to command.	F10
Go to next area.	F2
Go to previous area.	F1
Modify descriptor.	D
Modify tag.	T
Page down.	Page Dn
Page up.	Page Up

APPENDIX C - EDITOR ERROR CODES

ERROR CODE LISTING

Table C-1 lists editor error codes.

Table C-1. Editor Error Codes

Error Code	Description
1	Block number already used.
2	Block number out of range.
3	No more block numbers available.
10	Invalid parameter in command.
11	Invalid number of parameters in command.
12	Invalid function code.
13	Rung memory full.
14	Unknown command.
15	Tag not found.
16	Invalid logic configuration.
17	Invalid rung number.
18	Invalid tag.
19	Invalid file name.
20	No ***.LAD*** file name defined.
23	Cannot read or write file.
24	File not a ***.LAD*** file.
26	Value out of range.
27	Invalid format.
28	Duplicate tags.
29	Cannot read ***.SDT*** files.
30	Cannot read ***.STT*** or ***.TDF*** files.
32	Start or end rung out of range.
33	Invalid module type.
34	Cannot read ***.LAD*** initialization file.
35	Cannot read ***.MDT*** file.
36	Too many tags.
39	Relay network too large.
40	Rung with block number not found.
9*r*0*c*	Error in contact definition: *r* = row number *c* = contact number.

APPENDIX D - LOGIC CIRCUITS

LOGIC CIRCUIT DESCRIPTION

The logic equation of a logic rung is represented graphically as a logic circuit. A logic circuit is composed of circuit elements and logic circuits connected in series and parallel within a seven (wide) by five (high) grid. The circuit elements are contacts, horizontal connectors, and vertical connectors (the left rail consists of vertical connectors).

A series circuit is represented by a horizontal sequence of elements and circuits. A series circuit must contain at least one contact.

A parallel circuit is represented by two or more series circuits connected according to the following rules:

1. The series circuits are equal in length and vertically aligned.

2. The parallel circuit is closed on each end by a continuous line of vertical connectors.

3. Given the left end of a parallel circuit the circuit ends with the nearest matching right end.

Evaluation of a circuit (translating the circuit to an equation) starts in the upper left corner of the grid. There must be a complete circuit from the upper left to upper right.

Figure D-1 contains some example circuits and the corresponding logic equations.

Each contact in a logic circuit specifies two things - an input variable and an operation on that variable. The variable is the coil of a logic rung or the output of a function block rung and is identified by its tag name. The operation is specified by the symbol used for the contact. There are four types of operations as described in Table D-1.

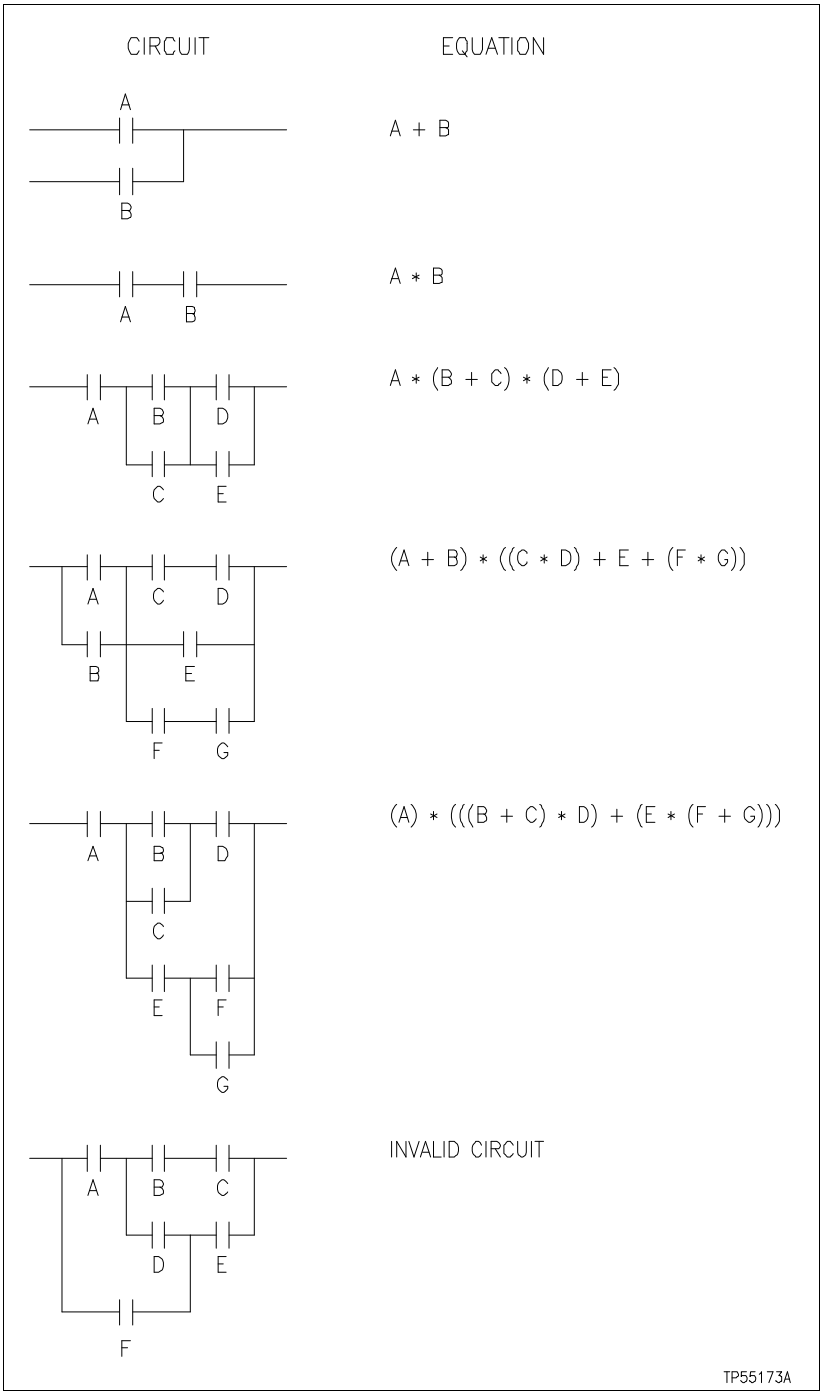


Figure D-1. Circuit Samples and Equations

Table D-1. Contact Type Descriptions

SYMBOL	VALUE OF VARIABLE	VALUE OF CONTACT	COMMENT
	0 1	0 1	NORMALLY OPEN
	0 1	1 0	NORMALLY CLOSED
	0 TO 1 OTHERWISE	1 0	LEADING EDGE
	1 TO 0 OTHERWISE	1 0	TRAILING EDGE

TP55174A

Consider the example of Figure D-2. CR will have the value of one if:

(value of PB1 changed from zero to one)

or

(Value of LS1 changed from one to zero)

or

(previous value of CR was one)

and

(value of LS2 is one)

and

(value of PB2 is zero)

Otherwise, the value of CR will be zero.

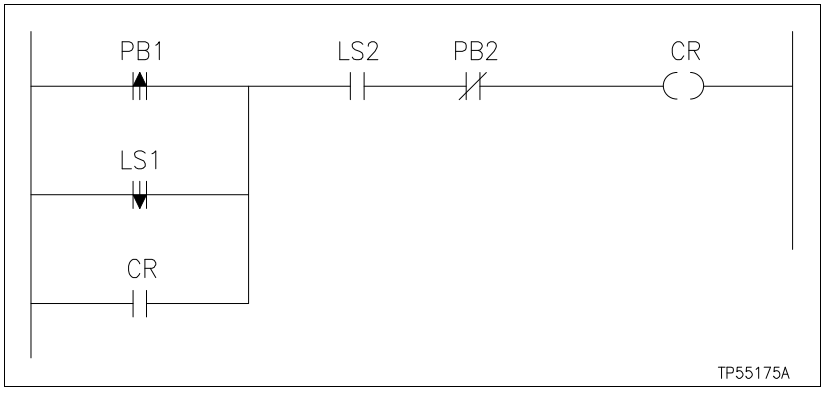


Figure D-2. Circuit Sample

APPENDIX E - REPLY CODES

REPLY CODE LISTING

The first byte of every reply from the computer interface to the host computer is the reply code. The reply code informs the host computer of the status of the command just issued. Table E-1 provides a list of reply codes. Reply codes are separated into two groups, reply codes greater than 100 and those less than 100.

Greater Than 100

The control module receiving the command from the host computer generates reply codes greater than 100. With these reply codes, the computer interface merely passes the reply code from the control module to the host computer. These reply codes reflect conflicts between the command issued by the host computer and the control module.

Less Than 100

Most nonzero reply codes less than 100 indicate a conflict between the incoming computer interface command and the internal operation of the computer interface. Not every nonzero reply code denotes an error.

Table E-1. Computer Interface Reply Codes

Number	Title	Description	Corrective Action
0	No error	Command successfully completed.	None required.
1	Waiting for loop	The command was started successfully. The computer interface is now waiting for a reply from the remote process control unit used in the command. The reply, when received, may be obtained using the **DEQUEUE** command with the same key field value as in the original command.	None required.
2	Improper format	Computer interface command not properly constructed. This usually indicates that the command contains too few or too many bytes.	Verify that the command corresponds exactly to computer interface format. Also verify that the checksum option is the same for both the computer interface and the host computer (both host and computer interface have checksum ON or OFF).

Table E-1. Computer Interface Reply Codes (continued)

Number	Title	Description	Corrective Action
3	Illegal command	This error denotes one of three possibilities: 1. The command code in this command is invalid. 2. The command issued is not valid for the current computer interface state. An example is using the **CIU CALLUP** command when password protection is not enabled. 3. Using the **Group** command for the first time without specifying the group of points.	Verify that the command code corresponds to a code listed in Table E-2. If this condition is met, the computer interface must be put into the proper state to execute the command. The **CIU RESTART** command options are frequently the cause of this error.
4	Index already established	The host computer has already established a point at this index. The host computer can only issue one establish command for each computer interface index.	Disestablish this index before using the establish command, or change the index of the new point.
5	Block already established another point	This point has already been established by the host computer using another index. The node, process control unit, module, block, and point type must be such that each index corresponds to unique data.	Use the index that is already established to access this point, or disestablish the existing index before establishing the point at a new index.
6	Command too long	The command supplied by the host computer is too long for the computer interface to process.	Shorten the command to fit within the computer interface format for that command.
7	Bad reply from node	The computer interface was unable to decipher the message sent to it.	Retry the message after verifying that it is properly constructed.
8	Export used as import	A command used to read data was issued for an index established as a data exporter.	Check the description of the point type established at the specified index. Establish it, if necessary, as the proper point type.
9	Repeat CIU restart command	The first **CIU RESTART** command executed properly. A second **CIU RESTART** command is required before continuing. This is a normal response for the computer interface.	Issue the second **CIU RESTART** command.
10	Undefined index	The command used an unestablished index.	Establish the index with the appropriate data.
11	Memory full	The capacity of the computer interface has been reached.	Reduce the load requirement on the computer interface.
12	Host communication error	The computer interface port used by the host computer detected a data transmission error, e.g., a framing error or a receiver overrun error.	Check the following: compatible baud rates; number of data bits; number of stop bits; parity; and noisy transmission line between the computer interface and the host computer.

Table E-1. Computer Interface Reply Codes (continued)

Number	Title	Description	Corrective Action
13	The INICT01 or INICT03 module not responding	The INICT01 or INICT03 module is unable to communicate with the INNIS01 module.	Check that modules are completely seated in the module mounting unit. Check all cable connections.
14	Import used as export	The host attempted to output data to an index that was established for reading data.	Establish index as a report point, or use another index.
15	Time-out of plant loop response	A reply to a command was not received within the time allotted.	Check the remote module addressed. Verify that the remote module configuration is correct, and it is in execute mode.
16	Number out of range	One or more of the fields in the commands are not within the acceptable limits for that command.	Check the limits on each field in the command and verify that each is within the proper range.
17	Illegal key	The key field value supplied in the command was not valid. Most commands require a key field of 0.	Correct the key field to correspond to the command requirements. This error condition also arises if this key value is already in use. Issue the **DEQUEUE** command before trying to use the key.
18	Need a restart command	The computer interface requires the **CIU RESTART** command to define its operating parameters. Issue this command before issuing any other command except **CIU CALLUP** and **ENVIRONMENT**.	Issue the **CIU RESTART** command before issuing other commands.
19	Module status used as import	An index established as a module status point was accessed as a regular read point. Module status points are not accessed by commands used for other read points.	Use the **READ MISCELLANEOUS STATUS** (**EXCEPTION**, **GROUP**, or **LIST**) command to access module status points.
20	Message active on loop	The computer interface is still waiting for a reply from the loop.	Try the **DEQUEUE** command again. It may be necessary to delay for a short time (2 seconds at most) between issuing the command and dequeuing the reply.
21	Import or export used as module status	An index established as a read or report point was accessed with a command intended for module status.	Use the **READ MISCELLANEOUS STATUS** (**EXCEPTIONS**, **LIST**, or **GROUP**) command to access read points. Data may be written to but not read from report points.
22	Exception report specifications lost	The specifications received for points established in the computer interface have been discarded by the computer interface. This is done when memory used to hold these specifications is needed for other more important functions such as establishing more points.	Use the **REGENERATE SPECS** command to bring specifications back into the computer interface.

Table E-1. Computer Interface Reply Codes (continued)

Number	Title	Description	Corrective Action
23	No message queued/ dequeue received	The **DEQUEUE** command was received from the host computer, but there were no outstanding commands from the host computer.	To receive the reply from an issued command with a nonzero key, issue the **DEQUEUE** command.
24	Reply too large	The reply received in response to a computer interface command is too large to be returned to the host computer.	Alter the command to generate a shorter reply.
25	Illegal station mode command	The host computer unsuccessfully attempted to change the mode of a control station. The most common causes of this error are: 1. The host computer is not allowed to change station modes (refer to the **CIU RESTART** command). 2. The mode change requested by the computer was illegal for the current state of the station.	Issue the **CIU RESTART** command with options set to allow the host computer to change station modes.
26	Illegal module number in command	The computer interface returns this code if the **READ PROBLEM REPORT** command specifies any module number other than 2 (the computer interface module number) in the same node in which the computer interface resides.	Try the command with the proper module number.
27	Time-out between bytes in command	The computer interface did not receive a complete, properly terminated command from the host computer within the allotted time-out period.	Verify the host computer properly formatted the command. Other possible causes of this error include mismatched baud rates, and noise generated by the host computer, modem, or transmission lines.
28	Index already established (by another computer)	Another process control unit in the system has a route established to this index in the computer interface. The remote node is expecting the host computer to output data from this index.	This index must be established as a report point. The report point type must correspond to the type used when establishing routes to this computer interface index. The configuration of the remote process control unit can also be changed to remove the exception report route from the computer interface.
29	Point type incompatible with command	The point type used in the command is not valid for this command. Generally, this conflict arises when attempting to write to a read point or read from a report point.	Verify the specified point type is allowed. If necessary, establish the index with the proper point type.

Table E-1. Computer Interface Reply Codes (continued)

Number	Title	Description	Corrective Action
30	Watchdog time-out	The computer interface did not receive a command from the host computer within the allowed time. This is specified with the **CIU RESTART** command.	When using the watchdog timer, the host computer must issue a command within the time period specified in the **CIU RESTART** command. Failure to do so causes the computer interface to take itself off-line. Set the watchdog timer in the **CIU RESTART** command to zero when not using this option.
31	Checksum compare error	The checksum supplied in the command differed from that calculated by the computer interface.	Verify that both the computer interface and the host computer either use or do not use checksums. If both the computer interface and the host computer use the checksum, a transmission error may have caused this error. Issue the command.
32	Destination node off-line	The computer interface cannot communicate with the process control unit referenced in the command.	Determine what caused the remote process control unit to go off-line. The remote process control unit must be on-line for proper command execution.
33	Call up command required	The computer interface is password protected. Issue the **CIU CALLUP** command before issuing any other commands.	If password protection is not desired, disable it and physically reset the module. Otherwise, issue the **CIU CALLUP** command with the proper password. This must be done separately for each port to be used.
34	Computer interface error	The computer interface has detected an internal software error. The data base in the computer interface may be corrupted.	Start the computer interface. If the error occurs again, notify Bailey Controls Company field service.
35	Computer interface busy	The computer interface is unable to process the incoming command because of a surge of internal activity. This may be the result of a previous command issued by the host computer, or the computer interface may have received a flood of data from the plant loop.	Issue the command.
36	INNIS01 off-line	The INNIS01 module is communicating with the loop but has gone off-line. The INNIS01 module is therefore unable to transfer data to or from the loop.	Issue the **CIU ONLINE/ OFFLINE** or **CIU RESTART** command to put the INNIS01 module back on-line.
37	Conflict with monitor mode	The command supplied by the host computer is disallowed in MONITOR MODE.	Change command or disable MONITOR MODE from the diagnostic port.

Table E-1. Computer Interface Reply Codes (continued)

Number	Title	Description	Corrective Action
38	Point type	The point type specified in the command is not the same as the established point type.	Check the point for point type. Issue command with proper point type.
39	Destination loop off-line	The computer interface is unable to communicate with the loop referenced in the command.	Determine what caused the computer interface of the remote loop to go off-line. The computer interface of the remote loop must be on-line for proper command execution.
40	Destination node busy	Destination node busy.	Retry.
41	Destination loop busy	Destination loop busy.	Retry.
42	Enhanced trend not established	The enhanced trend point has not been established as an index in the computer interface point table.	Establish the enhanced trend index before issuing this command.
100	Undefined message type	The message type is not valid for the target module.	Verify the message type and target module are correct.
101	Busy	The module cannot reply at this time.	Try the command at a later time.
102	Mode conflict	The mode required for the command does not agree with the current module mode.	Place the module into the correct mode.
103	Illegal data	The message data is out of legal range.	Verify the message data is correct.
104	Invalid block number	The function block number is not valid for the target module.	Verify the function block and the target module numbers are correct.
105	Undefined block number	The function block number specified is valid but not configured.	Verify the function block number is correct.
106	Block not readable	The function block specified is valid but has no readable parameters.	Verify the function block number is correct.
107	Invalid function code	The function code specified is not available for the target module.	Verify the function code is correct.
108	Function code and block number not compatible	The function code specified is not compatible with function block specified.	Verify the function code and function block are correct.
109	Insufficient memory to write block	There is insufficient RAM or NVRAM memory in the module to write the block.	Free up some memory space in the module.
110	Module not responding	The module is not operating correctly.	Check the condition of the module.
128	Waiting for module reply	The computer interface is waiting for the module to send a reply.	Wait until the module sends the reply.

Table E-2. Station Mode Settings

Setting Value	Description
0	Go to Local-Manual (Console/Station-Manual).
1	Go to Local Auto (Console/Station Auto).
2	Go to Local Cascade/Ration (Console/Station-Cascade/Ratio).
3	Go to Computer-Manual.
4	Go to Computer-Auto.
5	Go to Computer Cascade/Ratio.
6	Go to Local Level (Cascade/Station Level).
7	Go to Computer Level.
8	Go to Computer Back-up State.
9	Computer OK.
10	Go to Previous State.

APPENDIX F - PROBLEM REPORTS

PROBLEM REPORT LISTING

Table F-1 lists the module and extended module problem report descriptions.

Table F-1. Module and Extended Module Problem Reports

Report Type	Description	Byte Number	Byte Value
1	Communication problem (never mixed with other report types).	1	Report type (1)
		2	First node off-line
		3	Second node off-line
		4	Third node off-line
		5	Fourth node off-line
		n+1	n^{th} node off-line
2	Bad quality local I/O	1	Report type (2)
		2	Local block number
		3	
3	Bad quality module bus I/O	1	Report type (3)
		2	Local block number
		3	
		4	Remote module number
		5	Remote block number
		6	
4	Bad quality loop inputs	1	Report type (4)
		2	Local block number
		3	
		4	Remote module number
		5	Remote block number
		6	
		7	Remote node number
5	Communication failure with digital control station	1	Report type (5)
		2	Local block number
		3	
		4	Station address
6	Bad quality loop output	1	Report type (6)
		2	Local block number
		3	
7	Bad quality thermocouple signal (open thermocouple)	1	Report type (7)
		2	Local block number
		3	

Table F-1. Module and Extended Module Problem Reports (continued)

Report Type	Description	Byte Number	Byte Value
8	Bad quality cold junction RTD	1	Report type (8)
		2	Cold junction RTD block number
		3	
9	Unable to calibrate (bad quality)	1	Report type (9)
		2	Local block number
		3	
10	Out of service (bad quality)	1	Report type (10)
		2	Local block number
		3	
11	Lost exception report specs (bad quality)	1	Report type (11)
		2	Local block number
		3	
12	Bad quality I/O module	1	Report type (12)
		2	Local block number
		3	
		4	I/O module address
		5	Error type[1]
13	Remote I/O module error	1	Report type (13)
		2	Local block number
		3	
		4	I/O module address
		5	Error type
		6	Remote block number
		7	
		8	RMP expander bus address
		9	RSP serial link address
14	Remote digital station	1	Report type (14)
		2	Local block number
		3	
		4	Station address
		5	Remote block number
		6	
		7	RMP expander bus address
		8	RSP serial link address
15	Remote I/O control module	1	Report type (15)
		2	Local block number
		3	
		4	RMP primary I/O module address
		5	RMP backup I/O module address

Table F-1. Module and Extended Module Problem Reports (continued)

Report Type	Description	Byte Number	Byte Value
15 (cont.)		6	Error type: 1. RMP no response/wrong type 2. RSP node error
		7	Spare
		8	Spare
16	Remote I/O RMP/RSP	1	Report type (16)
		2	Byte count (18)
		3	Local block number
		4	
		5	RMP primary I/O module address
		6	RMP backup I/O module address
		7	Error type: 1. RMP no response/wrong type 2. RSP status OK
		8	RMP I/O module address with RSP errors
		9	RSP number 0 - 7 has problem
		10	RSP number 8 - 15 has problem
		11	RSP number 16 - 23 has problem
		12	RSP Number 24 - 31 has problem
		13	RSP number 32 - 39 has problem
		14	RSP number 40 - 47 has problem
		15	RSP number 48 - 55 has problem
		16	RSP number 56 - 63 has problem
		17	Spare
		18	Spare
17	Loop communications	1	Report type (17)
		2	Byte count = n+2
		3	First loop off-line
		4	Second loop off-line
		n + 2	n^{th} loop off-line
18	Loop and node communications byte count = 2 + (2*n) loop [i] = loop number node [i] = off-line node number	1	Report type (18)
		2	Byte count
		3	Loop [1]
		4	Node [1]
		5	Loop [2]
		6	Node [2]
		1 + (n*2)	Loop [n]
		2 + (n*2)	Node [n]

Table F-1. Module and Extended Module Problem Reports (continued)

Report Type	Description	Byte Number	Byte Value
19	INFI-NET® inputs	1	Report type (19)
		2	Byte count
		3	Local block number
		4	
		5	Remote loop number
		6	Remote node number
		7	Remote module number
		8	Remote block number
		9	
20	SER point data	1	SER point number
		2	
		3	Point attribute[2]
		4	Hour/Millisecond: Hour = 6 MSBs (if greater than 32 MSB, complete time stamp is invalid) Milliseconds = 10 LSBs
		5	
		6	Minute time stamp
		7	Second time stamp

NOTES FOR TABLE F-1:

1. Error type:
   1. No response or wrong type.
   2. Calibration error.
   3. Channel failure or out of range.
   4. Counter overflow.
   5. Blown fuse.

   (errors 6 - 15 are transmitter errors)
   6. Not responding
   7. Configuration mismatch.
   8. Wrong type.
   9. Input failure.
   10. Overpressured.
   11. Temperature range exceeded.
   12. Output fixed.
   13. Manifold fixed.
   14. Calibration.
   15. Communication inhibited.

2. Point attribute:
   Bit 7 (MSB) = Quality (0 = good, 1 = bad).
   Bit 6 = Alarm (0 = normal, 1 = bad or inactive).
   Bit 5 = Deleted (0 = normal, 1 = deleted from scan).
   Bit 4 = Spare.
   Bits 3 to 1 = 000 = standard SER point.
   001 = summary.
   010 = pre-fault.
   011 = post-fault.
   100 = snapshot.
   Bit 0 = SER point value (0 or 1).

® Registered trademark of Elsag Bailey Process Automation.

Index

C

CBC01 1-4
CLC03/04 1-4
Coil 1-3
Command line options 3-1
Compile 1-3
Configure mode 1-3
Configuring the INFI 90 interface 9-1
Contact 1-3
Creating a program 5-1
CSC01 1-4

D

Directory listing 8-1
Downloading a ladder program 6-3, A-24

E

Editor 5-1, A-2
- Commands 5-9, B-1
  - Block 5-10
  - Compile 5-12, A-22
  - Copy 5-14
  - Delete 5-15, A-19
  - Display 5-16, A-18
  - Done 5-18
  - Error list 5-19
  - Function code 5-20
  - Insert 5-21, A-17
  - Insert file 5-22
  - Mark 5-23
  - Save 5-24, A-20
  - Tags 5-25
  - Title 5-26
  - Xref 5-27
  - Xtag 5-28
- Error codes C-1
- Function block key commands 5-30, B-3, B-4
- Help 5-1
- Logic rung coil key commands 5-30, B-3
- Modify logic rung contact key commands 5-29, B-2
- Text editing key commands 5-29

Execute mode 1-3
Exiting LADDER 3-4

F

Files
- .CFG 5-12
- .LAD 1-3
- LAC.SDL 2-1
- LAD.BAT 2-1
- LM2.SDL 2-2
- MDT.DAT 2-2
- MFC.SDL 2-2
- PCDEM.LAD 1-3, 2-2
- PCDEMZZZ.LAD 2-2
- QAMM.LAD 2-2
- QCBC.LAD 2-2
- QCLC.LAD 2-2
- QCOM.LAD 2-2
- QCSC.LAD 2-2
- QGCM.LAD 2-2
- QLMM02.LAD 2-2
- QMFC.LAD 2-2
- SLAD.EXE 2-2
- SLADX.EXE 2-2
- STT.DAT 2-2
- TDF.DAT 2-2

Function block 1-3
Function code 1-3

G

Glossary 1-2

I

IMAMM03 1-4
IMCOM03/04 1-4
IMLMM02 1-4
IMMFC05 1-4
IMMFP01 1-4
IMMFP02 1-4
IMMFP03 1-4
INFI 90 interface data 6-2, A-46
Installation 2-1
Intended user 1-1
Interfacing to the target module 6-1

L

LADDER 1-3
- General procedure 3-4

Ladder logic 1-3
Ladder program 1-3
Logic circuits D-1
Logic rung 1-3

M

Manual conventions 1-4
Menu structure 3-3
Modifying a program 5-1

Index (continued)

Modifying rungs....5-29
Monitor utility.... 4-1, A-26
Monitoring
Function block rungs.... 4-5, A-34
Lists.... 4-7, A-38
Logic rungs.... 4-2, A-29

O

Overview....1-1

P

Printing a ladder program.... 7-1, A-40
Problem reports....F-1
Processor module....1-3

R

Reference documents....1-3
Reply codes....E-1
Requirements.... 1-4, 2-1
Rung.... 1-3
Descriptions.... 5-3
Function block rungs....5-6, A-11
Logic rungs....5-3, A-5
Number of.... 3-2

S

SLAD.... 1-3
SPM.... 1-3
Starting LADDER.... 3-1
Options.... 3-1

T

Tag.... 1-3
Tune.... 1-3
Tutorial....A-1

For prompt, personal attention to your instrumentation and control needs or a full listing of Bailey representatives in principal cities around the world, contact the Bailey location nearest you.

Australia
Elsag Bailey Pty. Limited
Regents Park, NSW
Phone: 61-2-645-3322
Telefax: 61-2-645-2212

Brazil
Bailey do Brasil
São Paulo
Phone: 55-11-548-4122
Telefax: 55-11-547-0315

Canada
Elsag Bailey (Canada), Inc.
Burlington, Ontario
Phone: 1-905-639-8840
Telefax: 1-905-639-8639

France
Elsag Bailey S.A.
Massy
Phone: 33-1-64-47-2000
Telefax: 33-1-64-47-2016

Germany
Bailey-F & P Automation GmbH
Overath
Phone: 49-220-473-90
Telefax: 49-220-473-979

Italy
Elsag Bailey
Genoa
Phone: 39-10-658-1
Telefax: 39-10-658-2941

Japan
Bailey Japan Company, Ltd.
Tagata-Gun, Shizuoka-Ken
Phone: 81-559-49-3311
Telefax: 81-559-49-1114

Jordan
Bailey Controls Jordan
Amman
Phone: 962-6-788-116
Telefax: 962-6-756-908

Mexico
Bailey Mexico S.A. de C.V.
Naucalpan
Phone: 52-5-557-6100
Telefax: 52-5-557-7022

Norway
Bailey Norge A.S.
Bergen
Phone: 47-55-222-000
Telefax: 47-55-222-010

People's Republic of China
Bailey Beijing Controls
Beijing
Phone: 86-10-401-0651
Telefax: 86-10-401-1643

Singapore
Elsag Bailey Pte. Ltd.
Singapore
Phone: 65-442-3200
Telefax: 65-442-2700

United Kingdom
Bailey Automation plc
Telford, Shropshire
Phone: 44-1952-670-477
Telefax: 44-1952-670-455

United States
Bailey Controls Company
Wickliffe, Ohio
Phone: 1-216-585-8500
Telefax: 1-216-585-8756

Venezuela
Bailey de Venezuela SA
Valencia
Phone: 58-41-329-196
Telefax: 58-41-327-632

Form I-E96-705A1 Litho in U.S.A. 196
Copyright © 1996 by Elsag Bailey Process Automation, As An Unpublished Work
® Registered Trademark of Elsag Bailey Process Automation
™ Trademark of Elsag Bailey Process Automation

Elsag Bailey
Process Automation

29801 Euclid Avenue • Wickliffe, Ohio 44092 • (216) 585-8500
Telex: 980621 • Telefax: (216) 585-8756 or (216) 943-4609