



XP8300, XP8400, and SE1100

Relay Output Boards

User's Manual

Revision B



XP8300, XP8400, and SE1100 User's Manual

Part Number 019-0054 • Revision B

Last revised on August 2, 1999 • Printed in U.S.A.

Copyright

© 1999 Z-World • All rights reserved.

Z-World reserves the right to make changes and improvements to its products without providing notice.

Trademarks

- Dynamic C[®] is a registered trademark of Z-World
 - Windows[®] is a registered trademark of Microsoft Corporation
 - PLCBus[™] is a trademark of Z-World
 - Hayes Smart Modem[®] is a registered trademark of Hayes Microcomputer Products, Inc.
-

Notice to Users

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

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

All Z-World products are 100 percent functionally tested. Additional testing may include visual quality control inspections or mechanical defects analyzer inspections. Specifications are based on characterization of tested sample units rather than testing over temperature and voltage of each unit. Z-World may qualify components to operate within a range of parameters that is different from the manufacturer's recommended range. This strategy is believed to be more economical and effective. Additional testing or burn-in of an individual unit is available by special arrangement.

Company Address



Z-World

2900 Spafford Street
Davis, California 95616-6800
USA

Telephone: (530) 757-3737
Facsimile: (530) 753-5141
Web Site: <http://www.zworld.com>
E-Mail: zworld@zworld.com

TABLE OF CONTENTS

About This Manual	vii
--------------------------	------------

XP8300

Chapter 1: Overview	13
Features	15
Chapter 2: Getting Started	17
Connecting Expansion Boards to a Z-World Controller	18
XP8300 Configuration	19
Setting Board Addresses	20
Chapter 3: Software Reference	21
Relay Board Addresses	22
Physical Addresses	22
Logical Addresses	22
Software	23
Dynamic C Libraries	23
How to Use the Relay Boards	24
Reset Boards on PLCBus	24
Address Target Board	25
Operate Relays	25

XP8400

Chapter 4: Overview	29
Features	31
Chapter 5: Getting Started	33
Connecting Expansion Boards to a Z-World Controller	34
XP8400 Configuration	35
Setting Board Addresses	36

Chapter 6: Software Reference	37
Relay Board Addresses	38
Physical Addresses	38
Logical Addresses	38
Software	39
Dynamic C Libraries	39
How to Use the Relay Boards	40
Reset Boards on PLCBus	40
Address Target Board	41
Operate Relays	41

SE1100

Chapter 7: Overview	45
Features	47
Chapter 8: Getting Started	49
Connecting an SE1100 to a Z-World Controller	50
SE1100 Configuration	52

Chapter 9: Software Reference	53
--------------------------------------	-----------

APPENDICES

Appendix A: PLCBus	57
PLCBus Overview	58
LCD Bus Signals	59
PLCBus Signals	59
Registers and Addresses	60
Device Allocation	63
4-Bit Devices	63
8-Bit Devices	63
Expansion Bus Software	64
Driver Calls	64

Appendix B: Specifications	69
XP8300 Relay Expansion Board	70
XP8400 Relay Expansion Board	71
SE1100 Relay Expansion Board	72
Quick-Release Connectors	73
Appendix C: Connecting and Mounting	
Multiple Boards	75
Connecting Multiple Boards	76
Mounting	78
Appendix D: Simulated PLCBus Connections	79
BL1000	80
BL1100	81
BL1300	81
BL1400 or BL1500	82
Appendix E: Advanced Programming	83
Controlling a Relay	84
PLC_EXP.LIB	85
PBUS_TG.LIB	86
PBUS_LG.LIB	86
DRIVERS.LIB	87
Sample Projects	88
PLCBus Controllers	88
Instructions	88
Sample Program	89
Controllers with Simulated PLCBus	90
Instructions for BL1000 and BL1100	90
Sample Program for BL1000 and BL1300	91
Index	93

Blank

ABOUT THIS MANUAL

This manual provides instructions for designing a controller system that uses relay boards. Instructions are also provided for using Dynamic C[®] functions.

Assumptions

Assumptions are made regarding the user's knowledge and experience in the following areas:

- Ability to design and engineer the target system that is controlled by a controller with analog-to-digital conversion expansion boards.
- Understanding of the basics of operating a software program and editing files under Windows on a PC.
- Knowledge of the basics of C programming.



For a full treatment of C, refer to the following texts.

The C Programming Language by Kernighan and Ritchie
C: A Reference Manual by Harbison and Steel

- Knowledge of basic Z80 assembly language and architecture for controllers with a Z180 microprocessor.



For documentation from Zilog, refer to the following texts.

Z180 MPU User's Manual
Z180 Serial Communication Controllers
Z80 Microprocessor Family User's Manual

- Knowledge of basic Intel assembly language and architecture for controllers with an Intel[™]386 EX processor.



For documentation from Intel, refer to the following texts.

Intel[™]386 EX Embedded Microprocessor User's Manual
Intel[™]386 SX Microprocessor Programmer's Reference Manual

Acronyms

Table 1 lists and defines the acronyms that may be used in this manual.







Table 1. Acronyms

Acronym	Meaning
EPROM	Erasable Programmable Read-Only Memory
EEPROM	Electrically Erasable Programmable Read-Only Memory
LCD	Liquid Crystal Display
LED	Light-Emitting Diode
NMI	Nonmaskable Interrupt
PIO	Parallel Input/Output Circuit (Individually Programmable Input/Output)
PRT	Programmable Reload Timer
RAM	Random Access Memory
RTC	Real-Time Clock
SIB	Serial Interface Board
SRAM	Static Random Access Memory
UART	Universal Asynchronous Receiver Transmitter

Icons

Table 2 displays and defines icons that may be used in this manual.

Table 2. Icons

Icon	Meaning	Icon	Meaning
	Refer to or see		Note
	Please contact	Tip	Tip
	Caution		High Voltage
	Factory Default		

Conventions

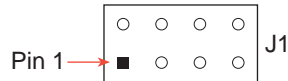
Table 3 lists and defines the typographical conventions that may be used in this manual.

Table 3. Typographical Conventions

Example	Description
while	Courier font (bold) indicates a program, a fragment of a program, or a Dynamic C keyword or phrase.
// IN-01...	Program comments are written in Courier font, plain face.
<i>Italics</i>	Indicates that something should be typed instead of the italicized words (e.g., in place of <i>filename</i> , type a file's name).
Edit	Sans serif font (bold) signifies a menu or menu selection.
...	An ellipsis indicates that (1) irrelevant program text is omitted for brevity or that (2) preceding program text may be repeated indefinitely.
[]	Brackets in a C function's definition or program segment indicate that the enclosed directive is optional.
< >	Angle brackets occasionally enclose classes of terms.
a b c	A vertical bar indicates that a choice should be made from among the items listed.

Pin Number 1

A black square indicates pin 1 of all headers.



Measurements

All diagram and graphic measurements are in inches followed by millimeters enclosed in parenthesis.

Blank

XP8300



Blank



CHAPTER 1: OVERVIEW

Chapter 1 gives an overview of the XP8300 relay board and its specific features.

Z-World's XP8300 expansion boards provide a simple way to add relays to a control system built around a Z-World controller. These relay output boards can be connected on the PLCBus in conjunction with other expansion boards. The actuation voltage for the board's relays comes from the controller via the PLCBus port. The XP8300's six relays are high-power relays.

Figure 1-1 illustrates a system of expansion boards mounted on a DIN rail and connected to a controller. Chapter 2, "Getting Started," provides instructions and illustrations for connecting a relay board to a controller's PLCBus port. Appendix D, "Simulated PLCBus Connection," provides instructions and illustrations for connecting relay boards to a specific controller that does not have a PLCBus port.

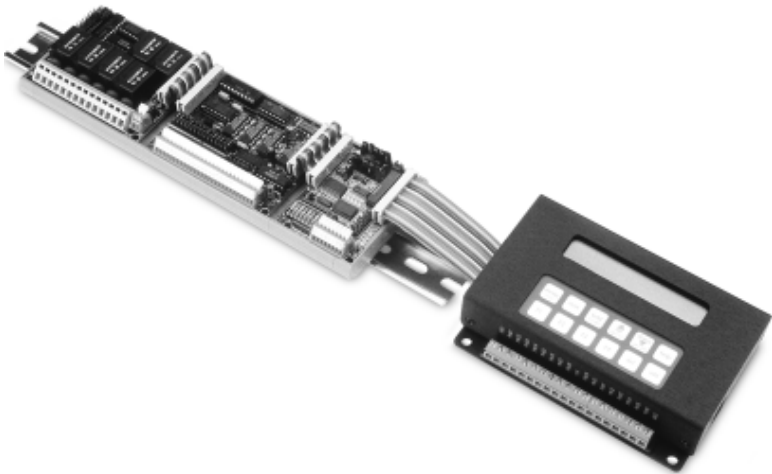


Figure 1-1. Expansion Board System

Features

The XP8300 board has six 24 V high-power relays installed as standard equipment: two are configured as SPDT and four are configured as SPST. All the relays have wire terminals that allow easy connections to external devices. Each relay is protected with a 10 A fuse. To help eliminate noise transients, a metal oxide varistor (MOV) is attached between pin 1 and pin 3 of each relay.

The inputs (pin 1) and normally open output contacts (pin 3) for all relays on an XP8300 board are accessible on headers H1 and H2. The normally closed outputs (pin 4) for relays 4 and 5 are available at header H4.

The XP8300 also has six LEDs that correspond to the six relays. An LED turns on when the corresponding relay's coil is energized. However, an illuminated LED does not verify that the contacts within the relay actually switch.

The XP8310 is a 12 V version of the XP8300.

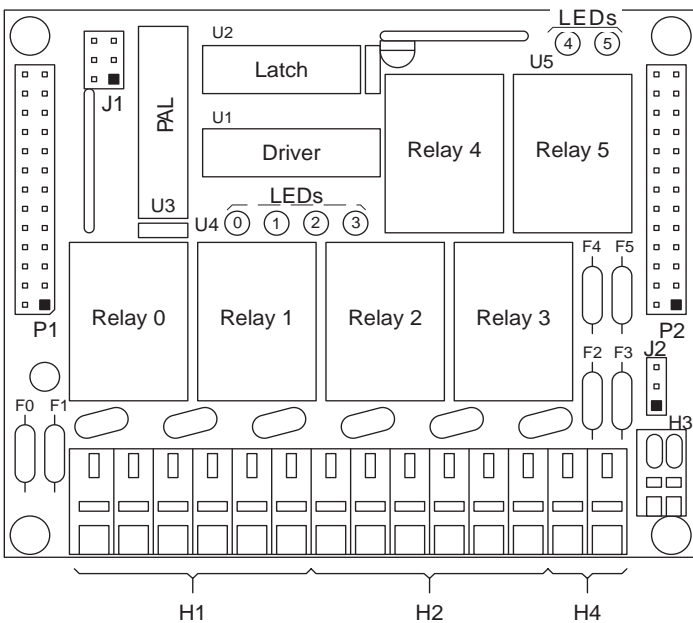


Figure 1-2. XP8300 Relay Expansion Board Layout

Blank



CHAPTER 2: **GETTING STARTED**

Connecting Expansion Boards to a Z-World Controller

Use the 26-conductor ribbon cable supplied with the expansion board to connect the expansion board to the PLCBus on a Z-World controller. See Figure 2-1. The expansion board's two 26-pin PLCBus connectors, P1 and P2, are used with the ribbon cable. Z-World recommends using the cable supplied to avoid any connection problems.

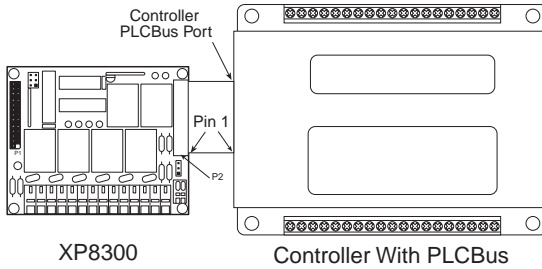


Figure 2-1. Connecting XP8300 Expansion Board to Controller PLCBus



Be sure power to the controller is disconnected before adding any expansion board to the PLCBus.

Follow these steps to connect an expansion board to a Z-World controller.

1. Attach the 26-pin ribbon cable to the expansion board's **P2** PLCBus header.
2. Connect the other end of the ribbon cable to the PLCBus port of the controller.



Be sure pin 1 of the connector cable matches up with pin 1 of both the controller and the expansion board(s).

3. If additional expansion boards are to be added, connect header **P2** on the new board to header **P1** of the board that is already connected. Lay the expansion boards side by side with headers P1 and P2 on adjacent boards close together, and make sure that all expansion boards are facing right side up.



See Appendix C, “Connecting and Mounting Multiple Boards,” for more information on connecting multiple expansion boards.

Controllers with simulated PLCBus ports require special expander cables, but are as easily connected. Appendix D, “Simulated PLCBus Connection,” gives detailed illustrated instructions for connecting relay boards to controllers without PLCBus ports.

XP8300 Configuration

The XP8300 board holds six high-power relays. Each XP8300 relay has the following specifications:

- Standard coil voltage 24 V DC.
- Contact ratings:
 - 10 A at 24 V DC
 - 10 A at 120 V AC
 - 7 A at 250 V AC resistive maximum.

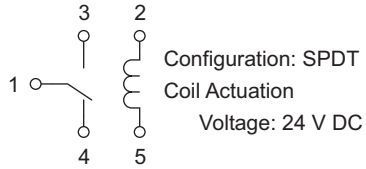



Figure 2-2. Relay Circuit

Pin 1 is the common. Pin 5 goes to a high-voltage/high-current driver on the relay board. Pin 2 is for the actuation voltage. Turning on the driver allows current to flow through the coil, switching on the relay. Pin 3 is the normally open contact. Pin 4 is the normally closed contact.

Each relay is protected by a 10 A fuse on pin 1. To help eliminate transients, a metal oxide varistor (MOV) is attached between pin 1 and pin 3 on each relay. An LED is connected in line with the coil on each relay, and lights up when current passes through the coil.



Although the relays are rated at up to 10 A, and are protected with 10 A fuses, the size of the traces on the printed circuit boards limits the current through each relay to 6 A.

Headers H1, H2, and H4 are used to connect external devices to the relays. Pin 1 and pin 3 connections for all relays are provided on headers H1 and H2. In addition, header H4 provides pin 4 connections for relays 4 and 5, allowing relays 4 and 5 to be used as SPDT relays. Relays 0 to 3 do not have their pin 4 available for external connection, and therefore can be used only as SPST relays.

Figure 2-3 illustrates the pinouts for the relay connection pins on headers H1, H2, and H4.

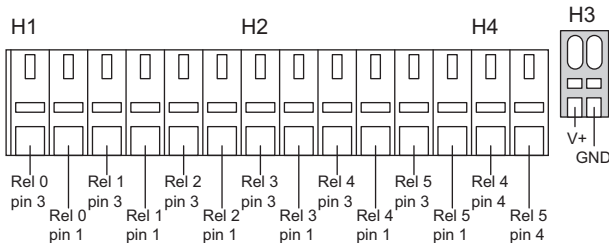


Figure 2-3. Relay Connection Pins

Jumper settings on header J2 determine the actuation voltage for the board's relays. When pins 1–2 are connected, the actuation voltage is supplied by the +24 V line on the PLCBus. When pins 2–3 are connected, the actuation voltage is supplied by the VCC line on the PLCBus.

When no pins on header J2 are connected, an actuation voltage must be supplied by connecting a 24 V power supply at sockets V+ and GND on header H3.



The XP8300 relays require an actuation voltage of 24 V, and the XP8310 relays require an actuation voltage of 12 V. These relays will not work with J2 pins 2–3 connected.



Apply a voltage on header H3 only when header J2 is not jumpered. Applying power to the board when J2 pins 1–2 or 2–3 are connected can damage the relay board and other boards on the bus.

Setting Board Addresses

Jumpers on header J1 (along with PAL encoding) determine the board's bus address. Figure 2-4 shows the jumper settings to set addresses 0–7.

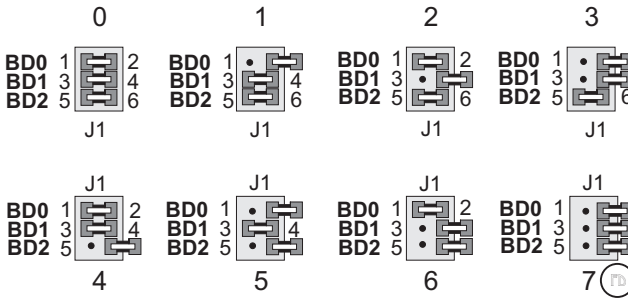


Figure 2-4. J1 Jumper Settings for XP8300 Board PLCBus Addresses



CHAPTER 3: SOFTWARE REFERENCE

Relay Board Addresses

Physical Addresses

Up to 64 addresses are possible on a single PLCBus. The 12-bit address of a particular relay board is determined by two factors: (1) the encoding of the PAL chip installed on the board, and (2) jumper settings on header J1. Since eight different PALs are available and J1 can be set eight different ways, 64 unique addresses are possible.

A 12-bit address can be conveniently placed on the bus using 4-bit addressing. A 12-bit physical address has the following format:

000z 000y pqr_x

Jumper bits are defined by the following pin settings:

z = 1 when J1 pins 5–6 are not connected

y = 1 when J1 pins 3–4 are not connected

x = 1 when J1 pins 1–2 are not connected

and

pqr is determined by the PAL.

The physical addresses correspond to the following PLCBus addresses.

000z—BUSADR0

000y—BUSADR1

pqr_x—BUSADR2

Logical Addresses

PLCBus expansion boards have “logical addresses.” Relay-specific software defines 64 integer board addresses, 0–63. The formula mapping physical address to logical address is defined by the following equation:

$$\text{logical address} = \text{pqr} \times 8 + \text{zyx}$$

The PAL encoding (pqr) and jumper bits (z, y, x) are defined above.

For example, a relay board that has PAL FPO4550 (pqr = 101) and J1 pins 5 and 6 connected (zyx = 011) would have the following addresses.

physical address: 000z 000y pqr_x = 0000 0001 1011 = 0x01B.

logical address: $101_{\text{B}} \times 8 + 011_{\text{B}} = 43 = 0x2\text{B}$.

Certain library functions expect a logical relay address.

Software

Dynamic C Libraries

Several Dynamic C function libraries are used with the routines defined in this section. Table 3-1 identifies which libraries are used with specific Z-World controllers.

Table 3-1. Dynamic C Libraries for Controllers

Library	Controller
EZIOCMMN.LIB	All controllers
EZIOPBDV.LIB	All controllers
EZIOTGPL.LIB	BL1000
EZIOLGPL.LIB	BL1100
EZIOMGPL.LIB	BL1400, BL1500
EZIOPLC.LIB	BL1200, BL1600, PK2100, PK2200
EZIOPLC2.LIB	BL1700
EZIOBL17.LIB	BL1700

Before using a library in an application, first include the library name in a **#use** command. For example, to use functions in the library **EZIOPLC.LIB**, insert the following line at the beginning of the program:

```
#use eziopl.c.lib
```

How to Use the Relay Boards

1. Send a reset command to all boards on the PLCBus.
2. Place the address of the target board on the PLCBus.
3. Operate the relays.

Reset Boards on PLCBus

These Dynamic C functions are used to initialize the PLCBus. Use these functions in a program before introducing any code to operate the relays.

- **VdInit()**

Initializes the timer mechanism.

LIBRARY: **VDRIVER.LIB**

- **void eioResetPlcBus()**

Resets all expansion boards connected to the PLCBus.

When using this function, initialize timers with **VdInit()** before resetting the PLCBus. All PLCBus devices must reset before performing any subsequent operations.

LIBRARY: **EZIOPLC.LIB**

- **void eioPlcRstWait()**

Provides a delay long enough for the PLCBus to reset.

This function provides a delay of 1–2 seconds to ensure devices on the PLCBus reset. Call this function after resetting the PLCBus.

LIBRARY: **EZIOBDV.LIB**

- **long int eioErrorCode**

Represents a global bit-mapped variable whose flags reflect error occurrences.

This register for this variable is initially set to 0. If the application tries to access an invalid channel, the flag **EIO_NODEV** (the first bit flag) is set in this register. Note that the other bits in **EIO_NODEV** deal with networked controllers.

Address Target Board

- **unsigned _eioPlcRelayAddr(unsigned BrdAddr);**

Converts bit pattern 00000000 00**pqrabc** to **pqrc 000b 000a** where **pqr** is the PAL number and **abc** is the address of the selected board.

PARAMETERS: The low byte of **BrdAddr** should contain the logical address ($8 * PAL\# + Board\#$). The board number is 0–63 (0–7 if only the factory default PAL is used).

RETURN VALUE: The bit-mingled BUSADR address **pqrc 000b 000a** for the XP8300 board.

LIBRARY: **EZIOBPDV.LIB**

Operate Relays

- **int plcXP83Out(unsigned address, int state);**

Energizes a relay on an XP8300 expansion board.

PARAMETERS: **address** is $8 * Board\# + Relay\#$. The board number is 0–63 (0–7 if only the factory default PAL is used). The relay number range is 0–5.

state indicates whether the relay should be energized—the specified relay is energized when **state** is non-zero, but is *not* energized when **state** is zero.

RETURN VALUE: 0 if the specified XP8300 and relay exist, otherwise –1. If the specified relay/board do not exist, the global variable **eioErrorCode** is bit-ored with the constant **EIO_NODEV**.

LIBRARY: **EZIOBPDV.LIB**

The **plcXP83Out** driver implements other function calls such as **eioPlcAdr12**, **eioPlcAdr4**, **eioReadD0**, **eioReadD1**, and **eioWriteWR**.



Refer to Appendix A, “PLCBus,” for a description of these other functions.

Blank

XP8400



Blank



CHAPTER 4: OVERVIEW

Chapter 4 gives an overview of the XP8400 relay board and its specific features.

Z-World's XP8400 expansion boards provide a simple way to add relays to a control system built around a Z-World controller. These relay output boards can be connected on the PLCBus in conjunction with other expansion boards. The actuation voltage for the board's relays comes from the controller via the PLCBus port. The XP8400's eight relays are low-power relays.

Figure 4-1 illustrates a system of expansion boards mounted on a DIN rail and connected to a controller. Chapter 5, "Getting Started," provides instructions and illustrations for connecting a relay board to a controller's PLCBus port. Appendix D, "Simulated PLCBus Connection," provides instructions and illustrations for connecting relay boards to a specific controller that does not have a PLCBus port.

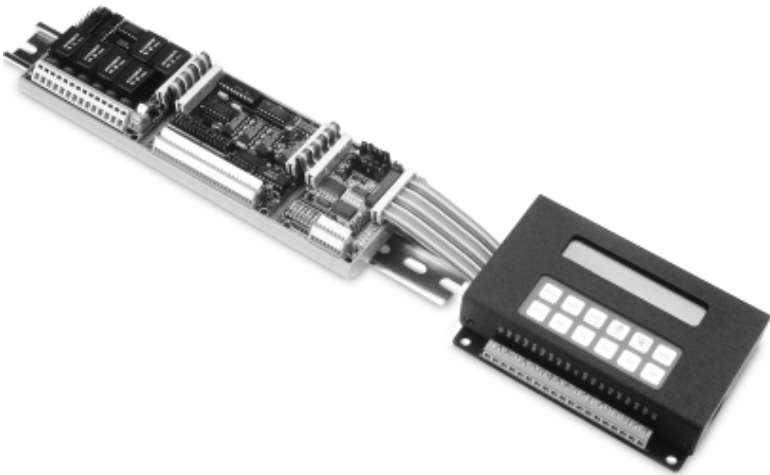


Figure 4-1. Expansion Board System

Features

The XP8400 expansion board has eight low-power (500 mA) SPST DIP relays that are normally open. Optional 12 V relays are available.

A single LED on the board lights up whenever it has been addressed via the PLCBus.

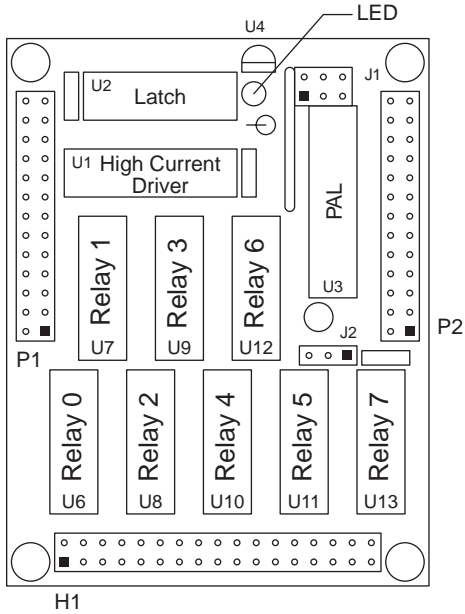


Figure 4-2. XP8400 Relay Expansion Board Layout

Blank



CHAPTER 5: **GETTING STARTED**

Connecting Expansion Boards to a Z-World Controller

Use the 26-conductor ribbon cable supplied with the expansion board to connect the expansion board to the PLCBus on a Z-World controller. See Figure 5-1. The expansion board's two 26-pin PLCBus connectors, P1 and P2, are used with the ribbon cable. Z-World recommends using the cable supplied to avoid any connection problems.

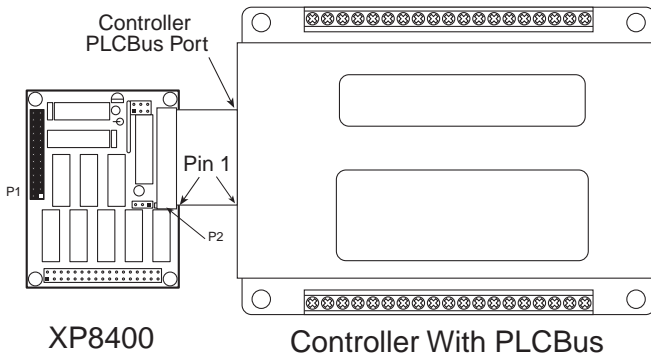


Figure 5-1. Connecting XP8400 Expansion Board to Controller PLCBus



Be sure power to the controller is disconnected before adding any expansion board to the PLCBus.

Follow these steps to connect an expansion board to a Z-World controller.

1. Attach the 26-pin ribbon cable to the expansion board's **P2** PLCBus header.
2. Connect the other end of the ribbon cable to the PLCBus port of the controller.



Be sure pin 1 of the connector cable matches up with pin 1 of both the controller and the expansion board(s).

3. If additional expansion boards are to be added, connect header **P2** on the new board to header **P1** of the board that is already connected. Lay the expansion boards side by side with headers P1 and P2 on adjacent boards close together, and make sure that all expansion boards are facing right side up.



See Appendix C, “Connecting and Mounting Multiple Boards,” for more information on connecting multiple expansion boards.

Controllers with simulated PLCBus ports require special expander cables, but are as easily connected. Appendix D, “Simulated PLCBus Connection,” gives detailed illustrated instructions for connecting relay boards to controllers without PLCBus ports.

XP8400 Configuration

The XP8400 board holds eight low-power DIP relays. The relays installed by Z-World as standard equipment have the specifications shown in Figure 5-2.

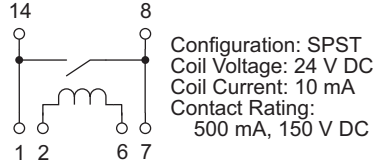


Figure 5-2. Relay Circuit

Header H1 (34 pins) connects external devices to the relays.

Contacts 1, 7, 8, and 14 of each relay are accessible through header H1. The relays are numbered 0–7. Contacts for relay 0 are called CT01, CT07, CT08, and CT014. The contacts for other relay 1 are named CT11, CT17, CT18, and CT114, and the contacts for the other relays are named similarly.

Figure 5-3 shows the pinouts for the relay connection pins for each relay.

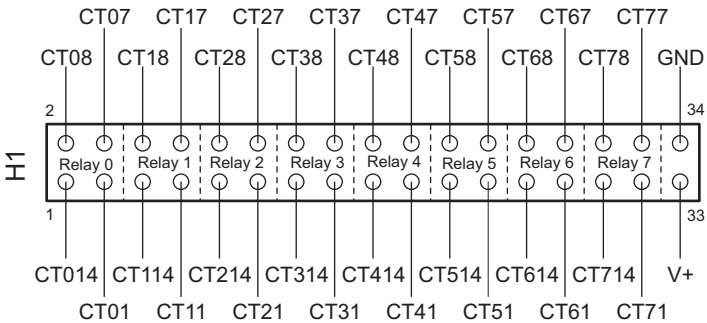


Figure 5-3. Relay Connection Pins

Jumper settings on header J2 determine the actuation voltage for the board’s relays. When pins 1–2 are connected, the actuation voltage is supplied by the +24 V line on the PLCBus. When pins 2–3 are connected, the actuation voltage is supplied by the VCC line on the PLCBus.

When no pins on header J2 are connected, an actuation voltage must be supplied by connecting a 24 V power supply at sockets V+ and GND on header H3.



The XP8400 relays require an actuation voltage of 24 V. These relays will not work with J2 pins 2–3 connected.



Apply a voltage on header H3 only when header J2 is not jumpered. Applying power to the board when J2 pins 1–2 or 2–3 are connected can damage the relay board and other boards on the bus.

Setting Board Addresses

Jumpers on header J1 (along with PAL encoding) determine the board's bus address. Figure 5-4 shows the jumper settings to set addresses 0–7.

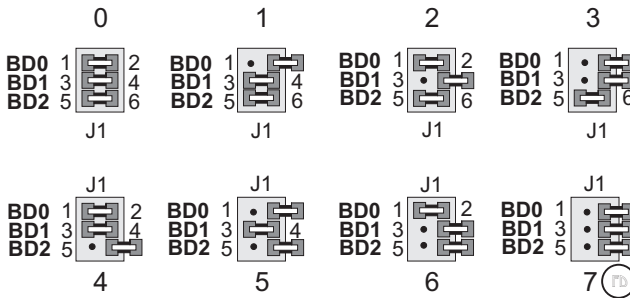


Figure 5-4. J1 Jumper Settings for XP8400 Board PLCBus Addresses



CHAPTER 6: **SOFTWARE REFERENCE**

Relay Board Addresses

Physical Addresses

Up to 64 addresses are possible on a single PLCBus. The 12-bit address of a particular relay board is determined by two factors: (1) the encoding of the PAL chip installed on the board, and (2) jumper settings on header J1. Since eight different PALs are available and J1 can be set eight different ways, 64 unique addresses are possible.

A 12-bit address can be conveniently placed on the bus using 4-bit addressing. A 12-bit physical address has the following format:

000z 000y pppx

Jumper bits are defined by the following pin settings:

z = 1 when J1 pins 5–6 are not connected

y = 1 when J1 pins 3–4 are not connected

x = 1 when J1 pins 1–2 are not connected

and

ppp is determined by the PAL.

The physical addresses correspond to the following PLCBus addresses.

000z—BUSADR0

000y—BUSADR1

pqr_x—BUSADR2

Logical Addresses

PLCBus expansion boards have “logical addresses.” Relay-specific software defines 64 integer board addresses, 0–63. The formula mapping physical address to logical address is defined by the following equation:

$$\text{logical address} = \text{pqr} \times 8 + \text{zyx}$$

The PAL encoding (pqr) and jumper bits (z, y, x) are defined above.

For example, a relay board that has PAL FPO4550 (pqr = 101) and J1 pins 5 and 6 connected (zyx = 011) would have the following addresses.

physical address: 000z 000y pqr_x = 0000 0001 1011 = 0x01B.

logical address: $101_{\text{B}} \times 8 + 011_{\text{B}} = 43 = 0x2\text{B}$.

Certain library functions expect a logical relay address.

Software

Dynamic C Libraries

Several Dynamic C function libraries are used with the routines defined in this section. Table 6-1 identifies which libraries are used with specific Z-World controllers.

Table 6-1. Dynamic C Libraries for Controllers

Library	Controller
EZIOCMMN.LIB	All controllers
EZIOPBDV.LIB	All controllers
EZIOTGPL.LIB	BL1000
EZIOLGPL.LIB	BL1100
EZIOMGPL.LIB	BL1400, BL1500
EZIOPLC.LIB	BL1200, BL1600, PK2100, PK2200
EZIOPLC2.LIB	BL1700
EZIOBL17.LIB	BL1700

Before using a library in an application, first include the library name in a **#use** command. For example, to use functions in the library **EZIOPLC.LIB**, insert the following line at the beginning of the program:

```
#use eziopl.lib
```

How to Use the Relay Boards

1. Send a reset command to all boards on the PLCBus.
2. Place the address of the target board on the PLCBus.
3. Operate the relays.

Reset Boards on PLCBus

These Dynamic C functions are used to initialize the PLCBus. Use these functions in a program before introducing any code to operate the relays.

- **VdInit()**

Initializes the timer mechanism.

LIBRARY: **VDRIVER.LIB**

- **void eioResetPlcBus()**

Resets all expansion boards connected to the PLCBus.

When using this function, initialize timers with **VdInit()** before resetting the PLCBus. All PLCBus devices must reset before performing any subsequent operations.

LIBRARY: **EZIOPLC.LIB**

- **void eioPlcRstWait()**

Provides a delay long enough for the PLCBus to reset.

This function provides a delay of 1–2 seconds to ensure devices on the PLCBus reset. Call this function after resetting the PLCBus.

LIBRARY: **EZIOBDV.LIB**

- **long int eioErrorCode**

Represents a global bit-mapped variable whose flags reflect error occurrences.

This register for this variable is initially set to 0. If the application tries to access an invalid channel, the flag **EIO_NODEV** (the first bit flag) is set in this register. Note that the other bits in **EIO_NODEV** deal with networked controllers.

Address Target Board

- **unsigned _eioPlcRelayAddr(unsigned BrdAddr);**

Converts bit pattern 00000000 00**pqrabc** to **pqrc 000b 000a** where **pqr** is the PAL number and **abc** is the address of the selected board.

PARAMETERS: The low byte of **BrdAddr** should contain the logical address ($8*PAL\# + Board\#$). The board number is 0–63 (0–7 if only the factory default PAL is used).

RETURN VALUE: The bit-mingled BUSADR address **pqrc 000b 000a** for the XP8400 board.

LIBRARY: **EZIOBPDV.LIB**

Operate Relays

- **int plcXP84Out(unsigned address, int state);**

Energizes a relay on an XP8400 expansion board.

PARAMETERS: **address** is $8*Board\# + Relay\#$. The board number is 0–63 (0–7 if only the factory default PAL is used). The relay number range is 0–7.

state indicates whether the relay should be energized—the specified relay is energized when **state** is non-zero, but is *not* energized when **state** is zero.

RETURN VALUE: 0 if the specified XP8400 and relay exist, otherwise –1. If the specified relay/board do not exist, the global variable **eioErrorCode** is bit-ored with the constant **EIO_NODEV**.

LIBRARY: **EZIOBPDV.LIB**

The **plcXP84Out** driver implements other function calls such as **eioPlcAdr12**, **eioPlcAdr4**, **eioReadD0**, **eioReadD1**, and **eioWriteWR**.



Refer to Appendix A, “PLCBus,” for a description of these other functions.

Blank

SE1100



Blank



CHAPTER 7: OVERVIEW

Chapter 7 gives an overview of the SE1100 relay board and its specific features.

Z-World's SE1100 expansion boards provide a simple way to add relays to a control system built around a Z-World controller. These relay output boards can be connected to the digital outputs of any Z-World controller. The SE1100 adds expansion capability even to boards without a Z-World PLCBus interface. The SE1100's four SPDT relays are high-power relays. Figure 7-1 illustrates a system of expansion boards mounted on a DIN rail and connected to a controller. Chapter 8, "Getting Started," provides instructions and illustrations for connecting the SE1100 relay board to a controller's digital outputs.

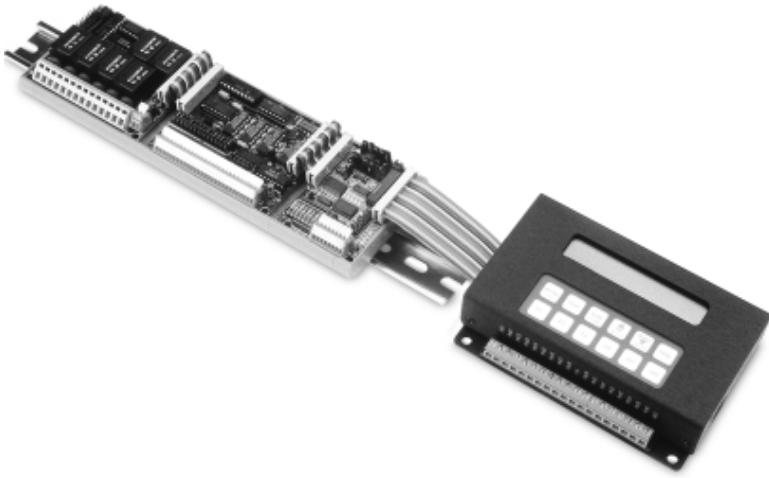


Figure 7-1. Expansion Board System

Features

The SE1100 relay board is designed to interface to the digital outputs of any Z-World controller. The board's four relays have a 6.3 A fuse connected to the common pin for overcurrent protection. In addition to the fuses, a snubber circuit across the common and the normally open/normally closed pins suppresses voltage spikes across the contacts. All of the signals from the four SPDT relays are brought out to header J1.

The interface voltage has a range of 5 V to 24 V. A 24 V DC supply is needed to power the relays. When driving the relays with high-voltage drivers, an SE1100 can be located up to 15 m (50 feet) from the controller. The opto isolation between the controller and the relays provides an extra level of assurance to guard against noise from high-voltage transients.

The LEDs on the relay board indicate the status of the relays. When an LED is on, the relay associated with that LED is energized. When an LED is off, the relay is in a default state. The default state is for the common terminal to be connected to the normally closed terminal.

The onboard linear regulator provides the regulated +5 V to all the logic elements. The relays and the LEDs are driven with the unregulated DC input voltage. Altogether, the SE1100 draws approximately 80 mA from the DC power supply input when all the relays are turned on.

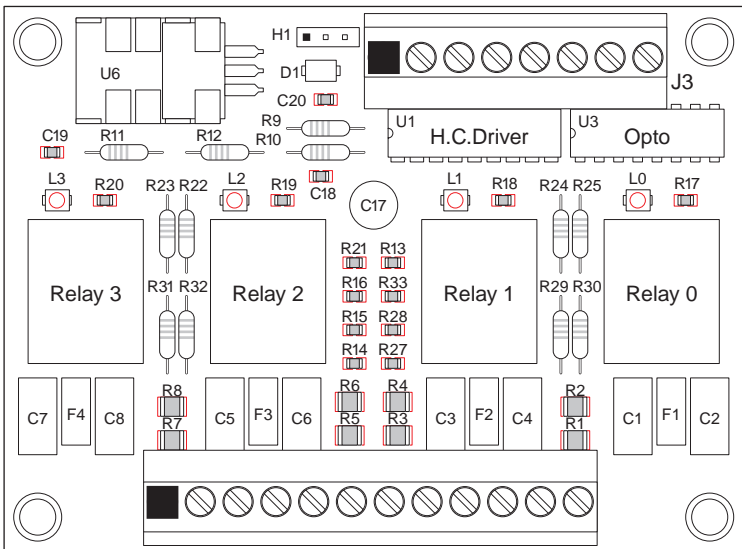


Figure 7-2. SE1100 Relay Expansion Board Layout

Blank



CHAPTER 8: **GETTING STARTED**

Connecting an SE1100 to a Z-World Controller

Connect the SE1100 to the digital outputs of any Z-World controller through the quick-release connector J3.

Figure 8-1 shows the pinout.

The four relays are optically isolated from the digital outputs on the host controller. RET1 provides a return for REL0 and REL1; RET2 provides a return for REL2 and REL3.

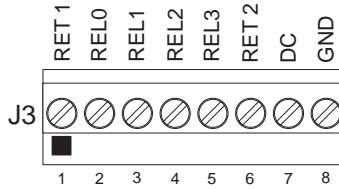


Figure 8-1. J3 Screw Terminal Addresses

High-Current Sinking Driver Connection

1. Wire RET1 and RET2 on J3 to K on the host controller.



K is connected to the +DC power supply on the host controller. K should not exceed 25 V.

2. Connect the four high-current outputs from the host controller to REL0, REL1, REL2, and REL3 on J3.
3. GND and DC on J3 may either be connected to a separate 24 V power supply, or they may be connected to GND and +DC on the host controller..

Figure 8-2 illustrates a typical SE1100 connection to a host controller with sinking high-current outputs.

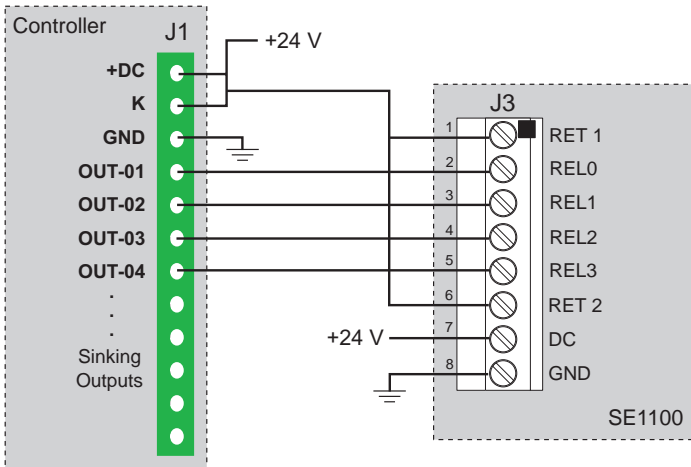


Figure 8-2. Connecting SE1100 to Controller with Sinking High-Current Outputs

High-Current Sourcing Drivers or TTL/CMOS Connection

1. Wire RET1 and RET2 on J3 to GND on the host controller.



K is connected to the +DC power supply on the host controller.
K should not exceed 25 V.

2. Connect the four high-current outputs or the TTL/CMOS outputs from the host controller to REL0, REL1, REL2, and REL3 on J3.
3. GND and DC on J3 may either be connected to a separate 24 V power supply, or they may be connected to GND and +DC on the host controller..

Figure 8-3 illustrates a typical SE1100 connection to a host controller with sourcing high-current outputs.

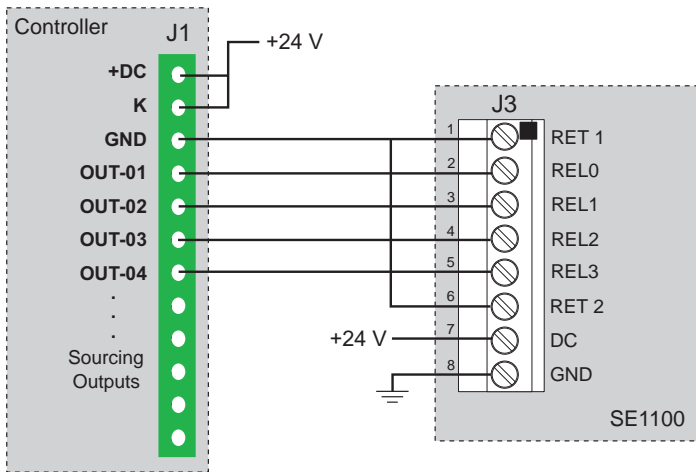


Figure 8-3. Connecting SE1100 to Controller with Sourcing High-Current Outputs

SE1100 Configuration

The SE1100 board holds four high-power relays. Each SE1100 relay has the following specifications:

- Standard coil voltage 24 V DC.
- Contact ratings:
 10 A at 24 V DC
 or 120 V AC,
 7 A at 250 V AC resistive
 maximum.

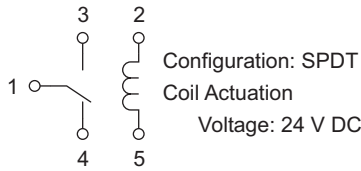


Figure 8-4. Relay Circuit

Pin 1 is the common. Pin 5 goes to a high-voltage/high-current driver on the relay board. Pin 2 is for the actuation voltage. Turning on the driver allows current to flow through the coil, switching on the relay. Pin 3 is the normally open contact. Pin 4 is the normally closed contact.

Each relay is protected by a 6.3 A fuse on pin 1. To help eliminate transients, a resistor/capacitor pair is attached between pin 1 and pin 3 on each relay. An LED is connected in line with the coil on each relay, and lights up when current passes through the coil.



Although the relays are rated at 10 A, they are protected with 6.3 A fuses because the size of the traces on the printed circuit boards limits the current through each relay to 6 A.

Header J1 is used to connect external devices to the relays.

Figure 8-5 illustrates the pinouts for the relay connection pins on header J1.

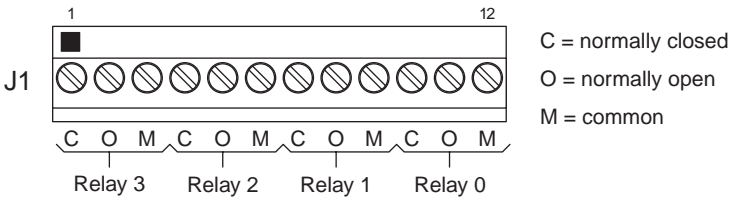


Figure 8-5. Relay Connection Pins



CHAPTER 9: SOFTWARE REFERENCE

There are no software drivers unique to the SE1100 expansion boards. Since the SE1100 is driven by the digital outputs of the host controller it is connected to, the drivers associated with the host controller's digital outputs will operate the relays on the SE1100.

The following sample program shows how to use the SE1100 with Z-World's BL1700 controller.

17SE1100.C

```
/* REL0 to U2, 0
   REL1 to U2, 1
   REL2 to U2, 2
   REL3 to U2, 3
   RET1 to RET2 to DC to BL1700, DCIN
   GND to U2, GND
*/

#include <vdriver.lib>
#include <eziobl17.lib>

main(){
    unsigned long t;
    VDInit();           // hits watchdog periodically
    eioBrdInit(0);     // initialize board
    while(1){
        t = MS_TIMER;
        printf("on\n");
        while((MS_TIMER - t) < 1000L){
            output(0x4100, 1);
            output(0x4100, 3);
            output(0x4100, 5);
            output(0x4100, 7);
        }
        t = MS_TIMER;
        printf("off\n");
        while((MS_TIMER - t) < 1000L){
            output(0x4100, 0);
            output(0x4100, 2);
            output(0x4100, 4);
            output(0x4100, 6);
        }
    }
}
```

APPENDICES



Blank



*APPENDIX A: **PLCBus***

Appendix A provides the pin assignments for the PLCBus, describes the registers, and lists the appropriate software drivers.

PLCBus Overview

The PLCBus is a general-purpose expansion bus for Z-World controllers. The PLCBus is available on the BL1200, BL1600, BL1700, PK2100, PK2200, and PK2600 controllers. The BL1000, BL1100, BL1300, BL1400, and BL1500 controllers support the XP8300, XP8400, XP8600, and XP8900 expansion boards using the controller’s parallel input/output port. The BL1400 and BL1500 also support the XP8200 and XP8500.

The ZB4100’s PLCBus supports most expansion boards, except for the XP8700 and the XP8800.

The SE1100 adds expansion capability to boards with or without a PLCBus interface. Table A-1 lists Z-World’s expansion devices that are supported on the PLCBus.

Table A-1. PLCBus Expansion Board Features

Board	Features
EXP-A/D12	Eight channels of 12-bit A/D converters
SE1100	Four SPDT relays, digital interface for all controllers
XP8100 Series	32 digital inputs/outputs
XP8200	“Universal Input/Output Board”—16 universal inputs, 6 high-current digital outputs
XP8300	Two SPDT and four SPST high-power relays
XP8400	Eight low-power SPST DIP relays
XP8500	11 channels of 12-bit A/D converters
XP8600	Two channels of 12-bit D/A converters
XP8700	One full-duplex asynchronous RS-232 port
XP8800	One-axis stepper motor control
XP8900	Eight channels of 12-bit D/A converters

Multiple expansion boards may be linked together and connected to a Z-World controller to form an extended system.

Figure A-1 shows the pin layout for the PLCBus bus connector.

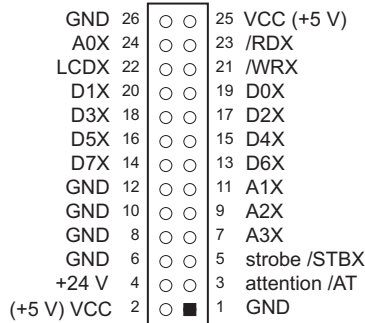


Figure A-1. PLCBus Pin Diagram

Two independent buses exist on a 26-conductor connector: the LCD bus and the PLCBus.

LCD Bus Signals

The LCD bus consists of the following signals:

- LCDX — positive-going strobe.
- /RDX — negative-going strobe for read.
- /WRX — negative-going strobe for write.
- A0X — address line for LCD register selection.
- D0X-D7X — bidirectional data lines (shared with expansion bus).

The LCD bus is used to connect Z-World's OP6000 series interfaces or to directly drive certain small liquid crystal displays. Figure A-2 illustrates the connection of an OP6000 interface to a BL2100 controller.

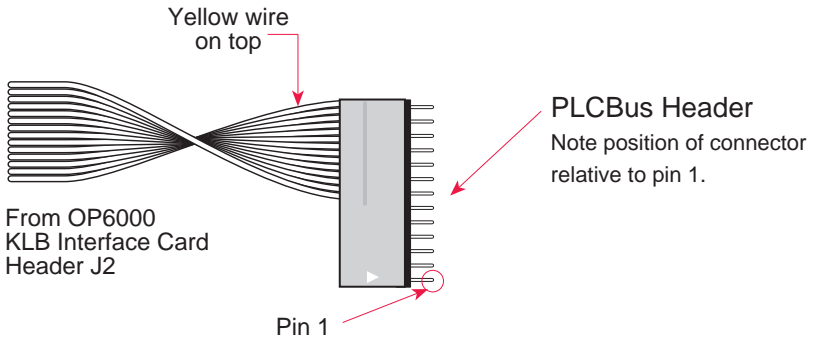


Figure A-2. OP6000 Connection to BL2100

PLCBus Signals

The PLCBus may be used as a 4-bit bus (D0X–D3X) or as an 8-bit bus (D0X–D7X). Whether it is used as a 4-bit bus or an 8-bit bus depends on the encoding of the address placed on the bus. Some PLCBus expansion cards require 4-bit addressing and others (such as the XP8700) require 8-bit addressing. These devices may be mixed on a single bus.

The PLCBus consists of the following signals.

- /STBX — negative-going strobe.
- A1X–A3X — three control lines for selecting bus operation.
- D0X–D3X — four bidirectional data lines used for 4-bit operations.
- D4X–D7X — four additional data lines for 8-bit operations.
- /AT — attention line (open drain) that may be pulled low by any device, causing an interrupt.

Registers and Addresses

There are eight registers corresponding to the modes determined by bus lines A1X, A2X, and A3X. Writing or reading one of these registers takes care of all the bus details. Functions are available in Dynamic C libraries to read from or write to expansion bus devices.

To communicate with a device on the expansion bus, first select a register associated with the device. Then read from or write to the register. The register is selected by placing its address on the bus. Each device recognizes its own address and latches itself internally. The registers are listed in Table A-2.

A typical device has three internal latches corresponding to the three address bytes. The first is latched when a matching BUSADR0 is detected. The second is latched when the first is latched and a matching BUSADR1 is detected. The third is latched if the first two are latched and a matching BUSADR2 is detected. If 4-bit addressing is used, then there are three 4-bit address nibbles, giving 12-bit addresses. In addition, a special register address is reserved for address expansion. This address, if used, would provide an additional four bits of addressing when using the 4-bit convention.

Table A-2. PLCBus Registers

Register	Address	A3	A2	A1	Meaning
BUSRD0	C0	0	0	0	Read data, one way
BUSRD1	C2	0	0	1	Read data, another way
BUSRD2	C4	0	1	0	Spare, or read data
BUSRESET	C6	0	1	1	Read this register to reset the PLCBus
BUSADR0	C8	1	0	0	First address nibble or byte
BUSADR1	CA	1	0	1	Second address nibble or byte
BUSADR2	CC	1	1	0	Third address nibble or byte
BUSWR	CE	1	1	1	Write data

If eight data lines are used, then the addressing possibilities of the bus become much greater: more than 256 million addresses according to the conventions established for the bus.

Place an address on the bus by writing (bytes) to BUSADR0, BUSADR1, and BUSADR2 in succession. Since 4-bit and 8-bit addressing modes must coexist, the lower four bits of the first address byte (written to BUSADR0) identify addressing categories, and distinguish 4-bit and 8-bit modes from each other.

There are 16 address categories, as listed in Table A-3. An “x” indicates that the address bit may be a “1” or a “0.”

This scheme uses less than the full addressing space. The mode notation indicates how many bus address cycles must take place and how many bits are placed on the bus during each cycle. For example, the 5×3 mode means three bus cycles with five address bits each time to yield 15-bit addresses, not 24-bit addresses, since the bus uses only the lower five bits of the three address bytes.

Table A-3. First-Level PLCBus Address Coding

First Byte	Mode	Addresses	Full Address Encoding
— — — — 0 0 0 0	4 bits \times 3	256	0000 xxxx xxxx
— — — — 0 0 0 1		256	0001 xxxx xxxx
— — — — 0 0 1 0		256	0010 xxxx xxxx
— — — — 0 0 1 1		256	0011 xxxx xxxx
— — — x 0 1 0 0	5 bits \times 3	2,048	x0100 xxxxx xxxxx
— — — x 0 1 0 1		2,048	x0101 xxxxx xxxxx
— — — x 0 1 1 0		2,048	x0110 xxxxx xxxxx
— — — x 0 1 1 1		2,048	x0111 xxxxx xxxxx
— — x x 1 0 0 0	6 bits \times 3	16,384	xx1000 xxxxxx xxxxxx
— — x x 1 0 0 1		16,384	xx1001 xxxxxx xxxxxx
— — x x 1 0 1 0	6 bits \times 1	4	xx1010
— — — — 1 0 1 1	4 bits \times 1	1	1011 (expansion register)
x x x x 1 1 0 0	8 bits \times 2	4,096	xxxx1100 xxxxxxxx
x x x x 1 1 0 1	8 bits \times 3	1Meg	xxxx1101 xxxxxxxx xxxxx xxx
x x x x 1 1 1 0	8 bits \times 1	16	xxxx1110
x x x x 1 1 1 1	8 bits \times 1	16	xxxx1111

Z-World provides software drivers that access the PLCBus. To allow access to bus devices in a multiprocessing environment, the expansion register and the address registers are shadowed with memory locations known as shadow registers. The 4-byte shadow registers, which are saved at predefined memory addresses, are as follows.

SHBUS0	SHBUS0+1	SHBUS1 SHBUS0+2	SHBUS1+1 SHBUS0+3
Bus expansion	BUSADR0	BUSADR1	BUSADR2

Before a new addresses or an expansion register value is output to the bus, its value is stored in the shadow registers. All interrupts that use the bus save the four shadow registers on the stack. Then, when exiting the interrupt routine, it restores the shadow registers and outputs the three address registers and the expansion registers to the bus. This allows an interrupt routine to access the bus without disturbing the activity of a background routine that also accesses the bus.

To work reliably, bus devices must be designed according to the following two rules.

1. The device must not rely on critical timing such as a minimum delay between two successive register accesses.
2. The device must be capable of being selected and deselected without adversely affecting the internal operation of the controller.

Device Allocation

4-Bit Devices

Digital output devices, such as relay drivers, should be addressed with three 4-bit addresses followed by a 4-bit data write to the control register. The control registers are configured as follows:

bit 3	bit 2	bit 1	bit 0
A2	A1	A0	D

The three address lines determine which output bit is written. The output is set as either 1 or 0, according to D. If the device exists on the bus, reading the register drives bit 0 low. Otherwise bit 0 is a 1.

For digital input, each register (BUSRD0) returns four bits. The read register, BUSRD1, drives bit 0 low if the device exists on the bus.

Table A-4 provides the address allocations for the registers of 4-bit devices.

Table A-4. Allocation of Registers

A1	A2	A3	Meaning
000j	000j	xxxj	digital output registers, 64 registers $64 \times 8 = 512$ 1-bit registers
000j	001j	xxxj	analog output modules, 64 registers
000j	01xj	xxxj	digital input registers, 128 registers $128 \times 4 = 512$ input bits
000j	10xj	xxxj	analog input modules, 128 registers
000j	11xj	xxxj	128 spare registers (customer)
001j	xxxj	xxxj	512 spare registers (Z-World)

j controlled by board jumper

x controlled by PAL

8-Bit Devices

Z-World's XP8700 and XP8800 expansion boards use 8-bit addressing. Refer to the XP8700 and XP8800 manual.

Expansion Bus Software

The expansion bus provides a convenient way to interface Z-World's controllers with expansion boards or other specially designed boards. High-level software drivers are easier to use than low-level drivers, but high-level drivers are less efficient in some cases. Table A-5 lists software libraries particular to each controller.

Table A-5. Dynamic C PLCBus Libraries

Library	Controller
DRIVERS.LIB	All controllers
EZIOTGPL.LIB	BL1000
EZIOLGPL.LIB	BL1100
EZIOMGPL.LIB	BL1400, BL1500
EZIOPLC.LIB	BL1200, BL1600, PK2100, PK2200, ZB4100
EZIOPLC2.LIB	BL1700
PBUS_TG.LIB	BL1000
PBUS_LG.LIB	BL1100, BL1300
PLC_EXP.LIB	BL1200, BL1600, PK2100, PK2200

Driver Calls

There are 4-bit and 8-bit drivers. The 4-bit drivers employ the following calls.

- **void eioResetPlcBus ()**

Resets all expansion boards on the PLCBus. When using this call, make sure there is sufficient delay between this call and the first access to an expansion board.

LIBRARY: EZIOPLC.LIB, EZIOPLC2.LIB, EZIOMGPL.LIB.

- **void eioPlcAdr12(unsigned addr)**

Specifies the address to be written to the PLCBus using cycles BUSADR0, BUSADR1, and BUSADR2.

PARAMETER: **addr** is broken into three nibbles, and one nibble is written in each BUSADR_x cycle.

LIBRARY: EZIOPLC.LIB, EZIOPLC2.LIB, EZIOMGPL.LIB.

- **void set16adr(int adr)**

Sets the current address for the PLCBus. All read and write operations access this address until a new address is set.

PARAMETER: **adr** is a 16-bit physical address. The high-order nibble contains the value for the expansion register, and the remaining three 4-bit nibbles form a 12-bit address (the first and last nibbles must be swapped).

LIBRARY: **DRIVERS.LIB.**

- **void set12adr(int adr)**

Sets the current address for the PLCBus. All read and write operations access this address until a new address is set.

PARAMETER: **adr** is a 12-bit physical address (three 4-bit nibbles) with the first and third nibbles swapped.

LIBRARY: **DRIVERS.LIB.**

- **void eioPlcAdr4(unsigned addr)**

Specifies the address to be written to the PLCBus using only cycle BUSADR2.

PARAMETER: **addr** is the nibble corresponding to BUSADR2.

LIBRARY: **EZIOPLC.LIB, EZIOPLC2.LIB, EZIOMGPL.LIB.**

- **void set4adr(int adr)**

Sets the current address for the PLCBus. All read and write operations access this address until a new address is set.

A 12-bit address may be passed to this function, but only the last four bits will be set. Call this function only if the first eight bits of the address are the same as the address in the previous call to **set12adr**.

PARAMETER: **adr** contains the last four bits (bits 8–11) of the physical address.

LIBRARY: **DRIVERS.LIB.**

- **char _eioReadD0()**

Reads the data on the PLCBus in the BUSADR0 cycle.

RETURN VALUE: the byte read on the PLCBus in the BUSADR0 cycle.

LIBRARY: **EZIOPLC.LIB, EZIOPLC2.LIB, EZIOMGPL.LIB.**

- **char _eioReadD1()**
 Reads the data on the PLCBus in the BUSADR1 cycle.
 RETURN VALUE: the byte read on the PLCBus in the BUSADR1 cycle.
 LIBRARY: **EZIOPLC.LIB, EZIOPLC2.LIB, EZIOMGPL.LIB.**
- **char _eioReadD2()**
 Reads the data on the PLCBus in the BUSADR2 cycle.
 RETURN VALUE: the byte read on the PLCBus in the BUSADR2 cycle.
 LIBRARY: **EZIOPLC.LIB, EZIOPLC2.LIB, EZIOMGPL.LIB.**
- **char read12data(int adr)**
 Sets the current PLCBus address using the 12-bit **adr**, then reads four bits of data from the PLCBus with BUSADR0 cycle.
 RETURN VALUE: PLCBus data in the lower four bits; the upper bits are undefined.
 LIBRARY: **DRIVERS.LIB.**
- **char read4data(int adr)**
 Sets the last four bits of the current PLCBus address using **adr** bits 8-11, then reads four bits of data from the bus with BUSADR0 cycle.
 PARAMETER: **adr** bits 8-11 specifies the address to read.
 RETURN VALUE: PLCBus data in the lower four bits; the upper bits are undefined.
 LIBRARY: **DRIVERS.LIB.**
- **void _eioWriteWR(char ch)**
 Writes information to the PLCBus during the BUSWR cycle.
 PARAMETER: **ch** is the character to be written to the PLCBus.
 LIBRARY: **EZIOPLC.LIB, EZIOPLC2.LIB, EZIOMGPL.LIB.**
- **void writel2data(int adr, char dat)**
 Sets the current PLCBus address, then writes four bits of data to the PLCBus.
 PARAMETER: **adr** is the 12-bit address to which the PLCBus is set.
dat (bits 0-3) specifies the data to write to the PLCBus.
 LIBRARY: **DRIVERS.LIB.**

- **void write4data(int address, char data)**

Sets the last four bits of the current PLCBus address, then writes four bits of data to the PLCBus.

PARAMETER: **adr** contains the last four bits of the physical address (bits 8–11).

dat (bits 0–3) specifies the data to write to the PLCBus.

LIBRARY: **DRIVERS.LIB**.

The 8-bit drivers employ the following calls.

- **void set24adr(long address)**

Sets a 24-bit address (three 8-bit nibbles) on the PLCBus. All read and write operations will access this address until a new address is set.

PARAMETER: **address** is a 24-bit physical address (for 8-bit bus) with the first and third bytes swapped (low byte most significant).

LIBRARY: **DRIVERS.LIB**.

- **void set8adr(long address)**

Sets the current address on the PLCBus. All read and write operations will access this address until a new address is set.

PARAMETER: **address** contains the last eight bits of the physical address in bits 16–23. A 24-bit address may be passed to this function, but only the last eight bits will be set. Call this function only if the first 16 bits of the address are the same as the address in the previous call to **set24adr**.

LIBRARY: **DRIVERS.LIB**.

- **int read24data0(long address)**

Sets the current PLCBus address using the 24-bit address, then reads eight bits of data from the PLCBus with a BUSRD0 cycle.

RETURN VALUE: PLCBus data in lower eight bits (upper bits 0).

LIBRARY: **DRIVERS.LIB**.

- **int read8data0(long address)**

Sets the last eight bits of the current PLCBus address using address bits 16–23, then reads eight bits of data from the PLCBus with a BUSRD0 cycle.

PARAMETER: **address** bits 16–23 are read.

RETURN VALUE: PLCBus data in lower eight bits (upper bits 0).

LIBRARY: **DRIVERS.LIB**.

- **void write24data(long address, char data)**

Sets the current PLCBus address using the 24-bit address, then writes eight bits of data to the PLCBus.

PARAMETERS: **address** is 24-bit address to write to.

data is data to write to the PLCBus.

LIBRARY: **DRIVERS.LIB**.

- **void write8data(long address, char data)**

Sets the last eight bits of the current PLCBus address using address bits 16–23, then writes eight bits of data to the PLCBus.

PARAMETERS: **address** bits 16–23 are the address of the PLCBus to write.

data is data to write to the PLCBus.

LIBRARY: **DRIVERS.LIB**.



*APPENDIX B: **SPECIFICATIONS***

XP8300 Relay Expansion Board

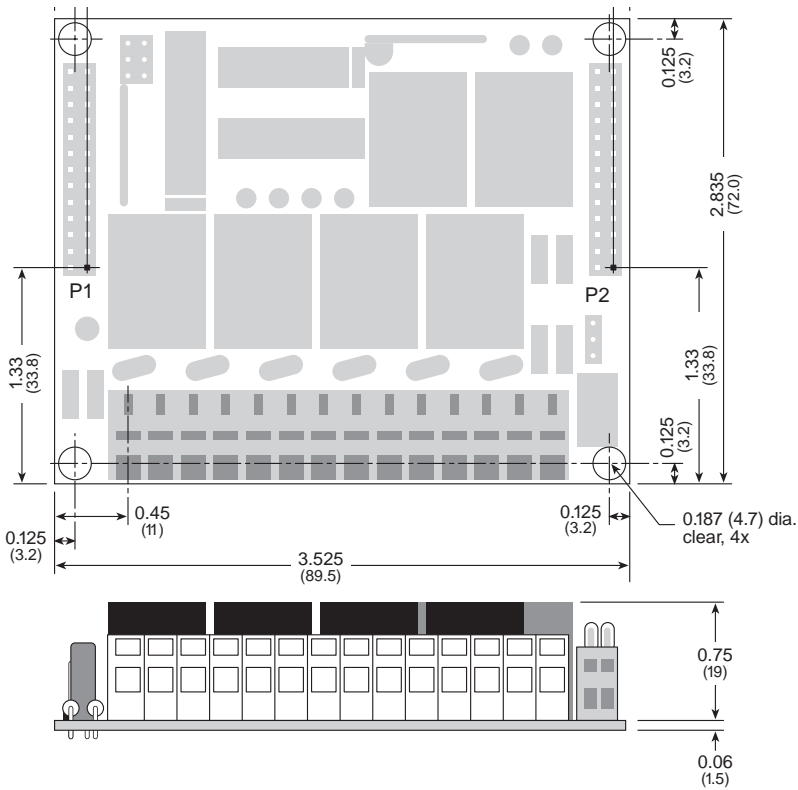


Figure B-1. XP8300 Dimensions

Table B-1. XP8300 Specifications

Feature	Specification
Board Size	2.835" × 3.525" × 0.78" (72.0 mm × 89.5 mm × ~20 mm)
Operating Temperature	-40°C to +70°C
Humidity	5% to 95%, noncondensing
Input Voltage and Current	24 V DC, 100 mA
Relays	6 SPDT relays—2 used as SPDT relays and 4 used as SPST relays 6 A at 250 V AC or 6 A at 24 V DC

XP8400 Relay Expansion Board

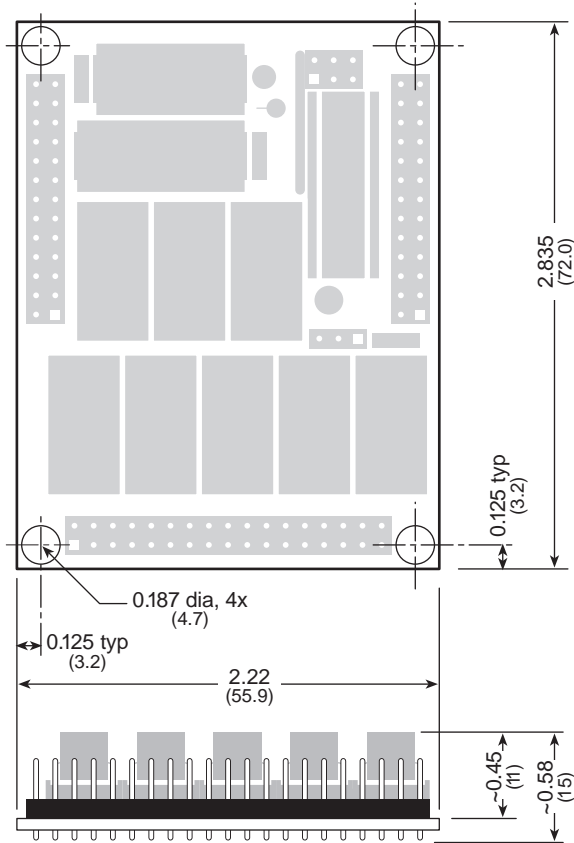


Figure B-2. XP8400 Dimensions

Table B-2. XP8400 Specifications

Feature	Specification
Board Size	2.835" × 2.22" × ~0.58" (72.0 mm × 55.9 mm × ~15 mm)
Operating Temperature	-40°C to +70°C
Humidity	5% to 95%, noncondensing
Input Voltage and Current	24 V DC, 80 mA
Relays	8 SPST DIP relays 500 mA at 150 V DC

SE1100 Relay Expansion Board

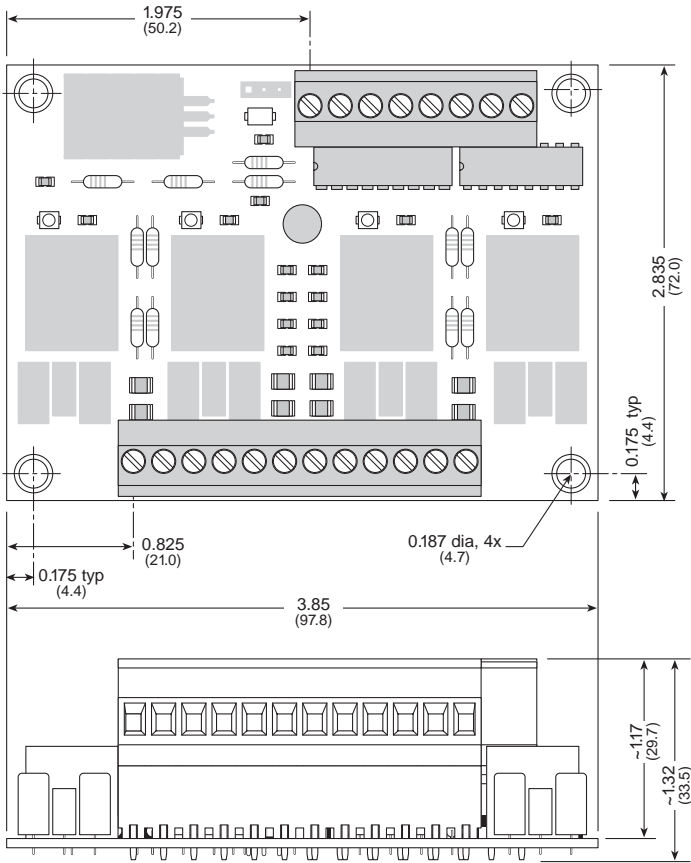


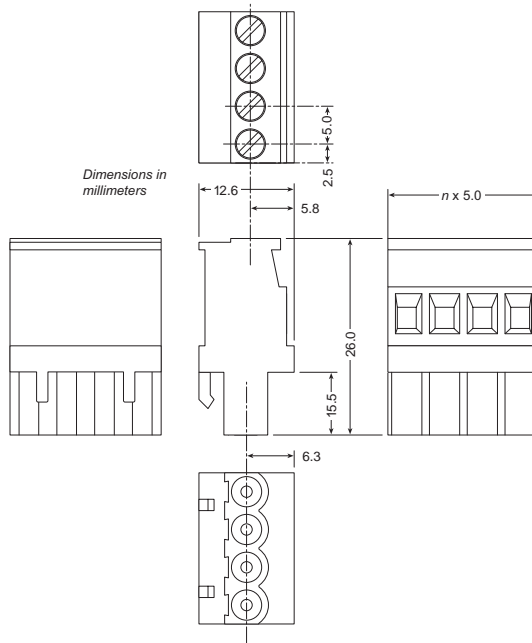
Figure B-3. SE1100 Dimensions

Table B-3. SE1100 Specifications

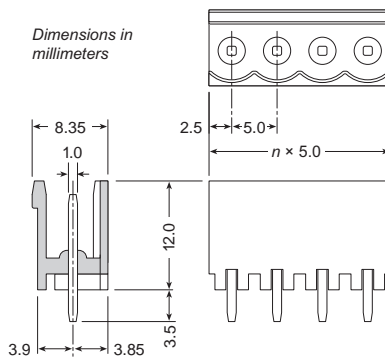
Feature	Specification
Board Size	2.835" × 3.85" × 1.32" (72.0 mm × 97.8 mm × 33.5 mm)
Operating Temperature	-40°C to +70°C
Humidity	5% to 95%, noncondensing
Input Voltage and Current	24 V DC, 80 mA
Relays	4 SPDT relays 6.3A at 250 V AC or 6.3A at 24 V DC

Quick-Release Connectors

The SE1100 comes equipped with quick-release connectors that allow for quick connection/disconnection. Figure B-4 illustrates the connectors and provides their dimensions. Table B-4 provides the specifications.



(a) Quick-Release Female Connector



(b) Quick-Release Male Connector

Figure B-4. Quick-Release Connectors

**Table B-4. Quick-Release Connectors
Specifications**

Feature	Specification
Maximum Voltage, Current	15 A @ 300 V
Insulation Resistance	100 GΩ
Wire	AWG #12–#26 stranded #14–#26 solid
Stripping Length	310 inches
Withdrawal Force	Meets UL 486
Torque	7 inches per pound



*APPENDIX C: **CONNECTING AND MOUNTING MULTIPLE BOARDS***

Connecting Multiple Boards

Eight or more expansion boards can be connected (“daisy chained”) at one time. The actual number of expansion boards may be limited by capacitive loading on the PLCBus.

Be sure that each expansion board has a unique address to prevent communication problems between the controller and the expansion board.

Follow these steps to install several expansion boards on a single PLCBus.

1. Place all expansion boards right side up.
2. Use the ribbon cable supplied with the boards.
3. Connect one board to the main controller.
4. Connect another expansion board to the first expansion board, connecting each board’s header P1 to the adjacent board’s header P2.

Figure C-1 illustrates a controller with expansion boards attached.

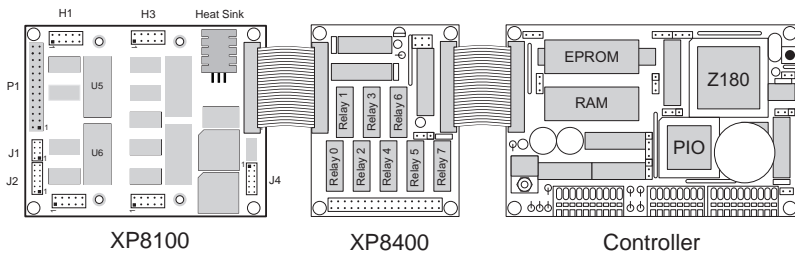


Figure C-1. Connecting Multiple Expansion Boards



Do not twist the ribbon cable or mount the expansion boards upside down! Damage may occur. Be sure Pin 1 of P1 and P2 of each board matches up with Pin 1 of the previous board. Pin 1 should be at the lower right when the expansion board is right side up, that is, the board markings are right side up.

When several expansion boards are connected, there may be a voltage drop along the network of expansion boards. No action is necessary as long as the digital voltage, VCC, is greater than 4.9 V on the last board.



VCC can be measured at pin 2 on header P1, and GND is pin 1 on header P1.

There are two ways to compensate for the voltage dropoff. The easiest way is to connect +5 V DC and ground from the host controller to pins 2 and 1 of header P1 on the last expansion board. Another solution, which can approximately double the number of boards that could otherwise be connected to a single controller, is a Y cable available from Z-World. Figure C-2 illustrates the use of the Y cable.

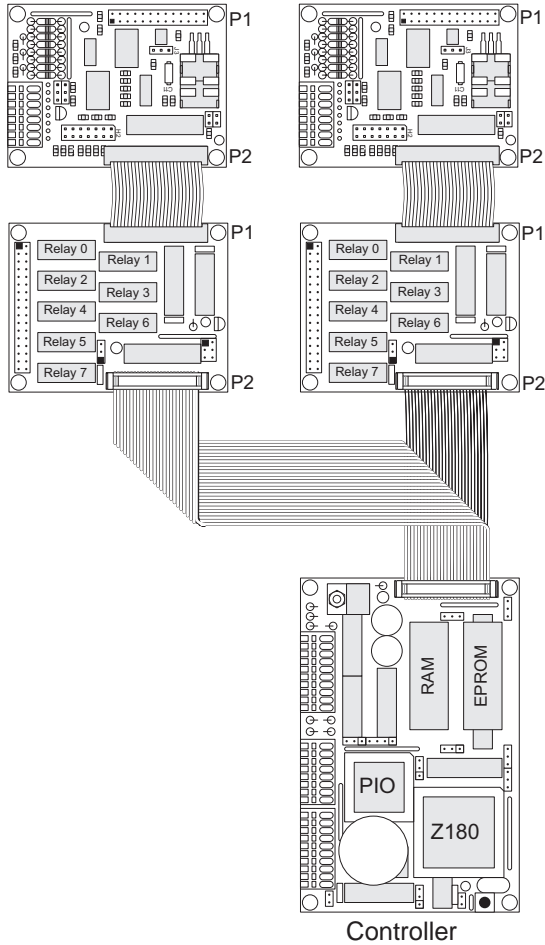


Figure C-2. Use of Y Cable to Connect Multiple Expansion Boards



For more information, call your Z-World Technical Support Representative at (530) 757-3737.

Mounting

The XP8300, XP8400, and SE1100 expansion boards can be installed in modular plastic circuit-board holders attached to a DIN rail, a widely used mounting system, as shown in Figure C-3.

The circuit-board holders are 77 mm wide and come in lengths of 11.25 mm, 22.5 mm, and 45 mm. The holders, available from Z-World and from other suppliers, snap together to form a tray of almost any length. Z-World's expansion boards are 72 mm wide and fit directly in these circuit-board holders.

Z-World's expansion boards can also be mounted with plastic standoffs to any flat surface that accepts screws. The mounting holes are 0.125 inches (1/8 inch) in from the edge of a board, and have a diameter of 0.190 inches.

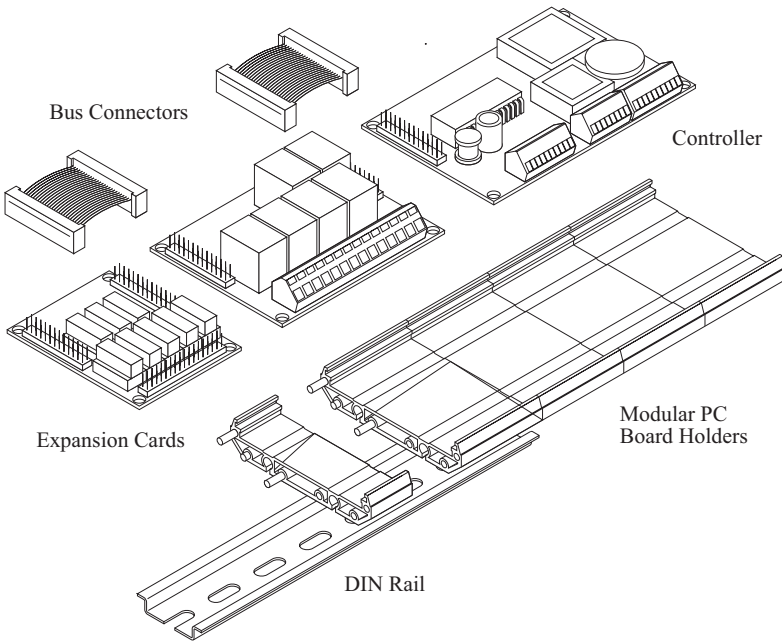


Figure C-3. Mounting Expansion Boards on DIN Rail



For information on ordering DIN rail mounts, call your Z-World Sales Representative at (530) 757-3737.



APPENDIX D:
SIMULATED PLCBUS CONNECTIONS

BL1000

The XP8300 and XP8400 expansion boards may be connected to a BL1000 using an expander cable (Z-World part number 540-0015). Fasten the cable's 20-pin connector to header J9 as shown in Figure D-1. Pins 1 and 2 of the connector must hang over the end of the header. Fasten the cable's PLCBus connector to header P1 or P2 of the expansion board, observing the orientation of pin 1, as shown.

Picks up VCC, GND, and PB0–PB7. Leaves PA0–PA7 available.

PIO Signal	PLCBus Signal
PB0 (J9:17)	D1X
PB1 (J9:15)	D0X
PB2 (J9:13)	D3X
PB3 (J9:11)	D2X
PB4 (J9:9)	A1X
PB5 (J9:7)	A2X
PB6 (J9:5)	A3X
PB7 (J9:3)	/STBX
+5 V (J9:1)	+5 V

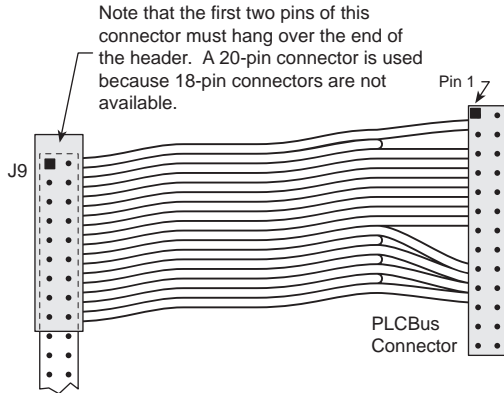


Figure D-1. BL1000 Expander Cable Connection

Software for interfacing the BL1000's PIO port to a PLCBus port may be found in the Dynamic C `PBUS_TG.LIB` library.



Use an external power supply with expansion boards connected to the BL1000. There is no provision in the special cable to supply +24 V from the controller to header P1 or P2 on the expansion boards.

BL1100

The XP8300 and XP8400 expansion boards may be connected to a BL1100 using an expander cable (Z-World part number 540-0015). Fasten the cable's 20-pin connector to the combined headers J010 and J10 as shown in Figure D-2. Pins 1 and 2 of the expander cable connector must hang over the end of the combined header. Fasten the cable's PLCBus connector to header P1 or P2 of the expansion board. Note the orientation of pin 1.

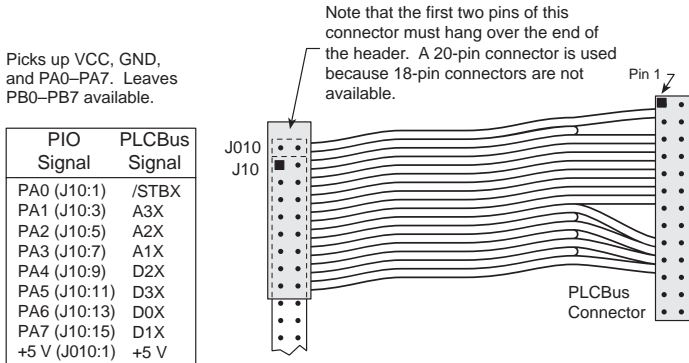


Figure D-2. BL1100 Expander Cable Connection

Software for interfacing the BL1100's PIO port to a PLCBus port may be found in the Dynamic C `PBUS_LG.LIB` library.



Use an external power supply when connecting expansion boards to the BL1100. There is no provision in the expander cable to supply +24 V from the controller to header P1 or P2 on the expansion boards.

BL1300

The XP8300 and XP8400 expansion boards may be connected to header P5 on the BL1300 using the same special cable used to connect them to the BL1000 or to the BL1100, as shown in Figure D-2. The first two pins of the special cable hang over the end of header P5 as before. However, the wire leading to pin 1 on the BL1300's header P5 must be cut, and may then be used to supply +5 V from an external source to the expansion board. Software from the Dynamic C `PBUS_LG.LIB` library may be used.



Use an external power supply with expansion boards connected to the BL1300. There is no provision in the special cable to supply +24 V from the controller to header P1 or P2 on the expansion boards.

BL1400 or BL1500

XP8300 and XP8400 expansion boards may be connected to header H3 on either the BL1400 or the BL1500. To add these expansion boards, the user must either make a custom cable or use an adapter board (Z-World part number 101-0050). To assist with making the connection via a ribbon cable, Table D-1 maps the signals from the controller's PIO to the expansion board PLCBus. Dynamic C's `EZIOMGPL.LIB` library may be used for programming.

Table D-1. PIO to PLCBus Signal Map

BL1400/BI1500		Expansion Board	
H3 Pin No.	PIO Port Signal	Pin No.	PLCBus Signal
1	VCC (+5 V)	2	VCC (+5 V)
2	PA0	5	/STBX
3	PA1	19	D0X
4	PA2	20	D1X
5	PA3	17	D2X
6	PA4	18	D3X
7	PA5	11	A1X
8	PA6	9	A2X
9	PA7	7	A3X
10	GND	10	GND

The adapter board provides an easy way to add XP8300 or XP8400 expansion boards to either a BL1400 or a BL1500 controller. Power is supplied to the controller via the power jack and to the expansion board via a screw terminal. For specifics on how to install an adapter board with a specific controller, see that controller's user's manual.

Use the appropriate external voltage supply with expansion boards connected to the BL1400 and BL1500.

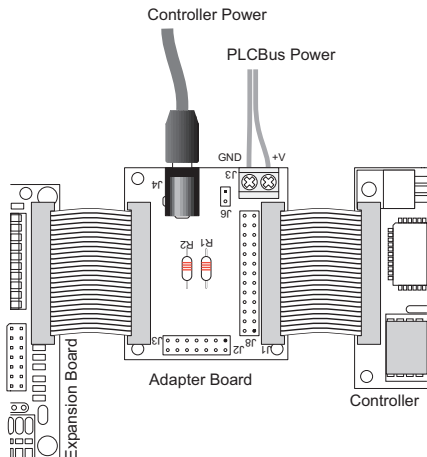


Figure D-3. Adapter Board Connections



APPENDIX E:
ADVANCED PROGRAMMING

Appendix E provides alternative programming methods for the XP8300 and XP8400 relay expansion boards.

Controlling a Relay

Once a relay's address is placed on the bus (the most recent address on the bus remains in effect), relays can be switched indefinitely. Use the BUSWR bus cycle to place four bits of data on the bus. Table D-1 shows the relay physical addresses and states.

Table D-1. Relay Addresses and States

Relay	Data Bits			
	D3	D2	D1	D0
0	0	0	0	0 = relay off
1	0	0	1	1 = relay on
2	0	1	0	
3	0	1	1	
4	1	0	0	
5	1	0	1	
6	1	1	0	
7	1	1	1	

A formula for turning on a relay is

```
relay# << 1 | 1
```

The following code fragments illustrate how to turn on a relay using this formula for a BL1200, PK2200, or PK2100.

```
#define ON 1
#define OFF 0
#define BOARD 0x0301 // board address is
// 0x0103

#define REL3 6 // 3 << 1 = 6
set12adr( BOARD ); // select the board
output( BUSWR, REL3|ON ); // turn relay 3 on
```

(or)

```
write12data( BOARD, REL3|ON );
```

For a BL1100 or BL1000 use the following code.

```
#define ON 1
#define OFF 0
#define BOARD 0x0301 // board address is
// 0x0103

#define REL3 3
PBus_Addr( BOARD ); // select the board
PBus4_Write( REL3|ON ); // turn on relay 3
```

PLC_EXP.LIB

The `PLC_EXP.LIB` library supports PLCBus controllers when operating PLCBus expansion boards. This library provides general bus functions and specific functions for the XP8300, XP8400, XP8600, and Exp-A/D12 expansion boards.

There are four groups of functions in this library. Table D-2 lists the two groups used by relay boards. Analogous functions exist in other libraries.

Table D-2. PLC_EXP.LIB Groups

Group	Functions
General	<code>plc_poll_node</code> , <code>Reset_PBus</code> , <code>Reset_PBus_Wait</code>
Relay	<code>plc_set_relay</code>

- **`int plc_poll_node(int board)`**
Returns 1 if the board identified by physical address `board` can be found on the PLCBus and 0 if not.
- **`void plc_set_relay(int board, int relay, byte state)`**
Switches a relay on an XP8300 or XP8400 board.
PARAMETERS: `relay` must be from 0–7 (0–5 on an XP8300 board).
`state` must be 1 (on) or 0 (off).
`board` must be a logical board address (0–63).
- **`void Reset_PBus()`**
`void rset_pbuss_wait()`
The function `Reset_PBus` resets the PLCBus. The function `Reset_PBus_Wait` provides the necessary delay (~450 ms) for the bus to reset.
- **`int plcrel_addr(int board)`**
Returns the (nibble-interchanged) bus address for a relay board identified by a logical address (0–63).

PBUS_TG.LIB

The **PBUS_TG.LIB** library allows the BL1000 to operate Z-World's relay boards and the XP8600 expansion boards. The **PBUS_TG.LIB** library does not support any other expansion boards.

The functions in this library are identical (except for internal details) to those in the **PBUS_LG.LIB** library.

PBUS_LG.LIB

The **PBUS_LG.LIB** library allows the BL1100 to operate Z-World's relay boards and the XP8600 expansion boards. This library does not support any other expansion boards.

There are three groups of functions in this library. The two groups used by relay boards are listed in Table D-3. Analogous functions exist in other libraries. For example, **reset_pbus** in **PLC_EXP.LIB** is used with controllers with a PLCBus and performs the same function as **Reset_PBus** in this library, which is used with the BL1100 and the BL1300.

Table D-3. PBUS_LG.LIB Groups

Group	Functions
General	PBus12_Addr, PBus4_Write, PBus4_Read0, PBus4_Read1, PBus4_ReadSp, Reset_PBus, Reset_PBus_Wait, Poll_PBus_Node
Relay	Relay_Board_Addr, Set_PBus_Relay

- **void PBus12_Addr(int addr)**

Places a 12-bit address on the PLCBus, in 4-bit mode. That is, it places three 4-bit nibbles on the bus. The first and third nibbles of **addr** must be interchanged: if the bus address is 0x125, **addr** must be 0x521.

- **int PBus4_Read0 ()**
int PBus4_Read1 ()
int PBus4_ReadSp ()

Carries out a bus read cycle. These functions correspond to bus cycles BUSRD0, BUSRD1 and BUSSPARE, respectively.

- **void PBus4_Write(byte value)**

Carries out a BUSWR cycle.

- **int Poll_PBus_Node(int addr)**
Returns 1 if there is a board at **addr** on the PLCBus, and 0 if not. The first and third nibbles of **addr** must be interchanged: if the bus address is 0x125, **addr** must be 0x521.
- **int Relay_Board_Addr(int board)**
Returns the (nibble-interchanged) bus address for a relay board identified by a logical address (0-63).
- **void Reset_Pbus()**
void Reset_Pbus_Wait()
The function **Reset_PBus** resets the PLCBus. The function **Reset_PBus_Wait** provides the necessary delay (~450 ms) for the bus to reset.
- **void Set_PBus_Relay(int board, int relay, int state)**
Switches a relay on an XP8300 or XP8400 board. **relay** must be from 0–7. **state** must be 1 (on) or 0 (off). **board** must be specified by a logical board address (0–63).

DRIVERS.LIB

The functions **set12adr**, **read12data**, and **write12data** in **DRIVERS.LIB** use 12-bit bus addresses. When using the functions in the drivers library, swap the first and third nibbles of the address before passing the address to the function. For example, if the address is 0x125, pass 0x521.

Sample Projects

The following two sample programs activate the relays on one or more XP8300 or XP8400 boards attached to a controller. Two versions of the program are shown: one for PLCBus controllers, and one for the BL1100 and BL1300.

The following instructions tell how to set up a system, write and compile a program, and run a sample program to operate relay boards on a bus.

PLCBus Controllers

Instructions

1. Power up the controller and make sure it is working properly. If you encounter problems, consult the controller's reference manual.
2. Disconnect power from the controller.
3. Using a PLCBus ribbon cable, connect header P2 of the relay board to the PLCBus on the controller. Make sure both boards are right-side up, with their input and output headers facing toward you. If you have additional relay boards, chain them to the first board with PLCBus ribbon cables.
4. Check the jumpers on headers J1 and J2 on the relay boards. With only one board, leave J1 unjumped. With more than one board, leave J1 unjumped on the first board and set J1 with a different and unique address on each additional board. On every relay board, connect pins 1–2 on J2. This connection causes each board to draw its relay-actuation voltage from the +24 V provided over the PLCBus by the controller.



When using the standard XP8300 or XP8400 with 24 V relays, the controller must be powered by a 24 V supply or 24 V must be brought in externally in order to actuate the relays reliably.

5. Power up the controller and bring up Dynamic C on your PC. If you encounter problems reestablishing communications between your PC and the controller, consult the controller's reference manual.
6. Open and run the sample program. Refer to the *Dynamic C Technical Reference* manual for detailed instructions on running a program.
7. The LEDs on the relay board(s) will begin flashing to indicate the relays are actuating.

Sample Program

The relay board demonstration program can be used to locate all XP8300 and XP8400 expansion boards. The program then loops, activating the relays on each board. For each board, the program concludes with an all-on/all-off sequence. To locate each board, the program polls all 64 possible addresses, then displays the logical address in Dynamic C's **STDIO** window for each board that responds.

```

/*****
  Relay Board Demo for XP8300 and XP8400
  *****/
#define ON 1
#define OFF 0
main(){
  int board,relay,found,list[64];
  Reset_PBus(); // always do this, first thing
  delay(1000); // pause 1000ms for reset
                // Locate relay boards. Build list
                // and print board IDs

  found=0;
  printf("\nLogical relay addresses found: ");
  for( board=0; board<64; board++ ){
    if( plc_poll_node(plcrel_addr(board)) ){
      list[found++] = board;
      printf(" %d ",board);
      if( found%10 == 0 ) printf("\n");
    }
  }

                // Activate relays on each board
                // found
  while( 1 ){ // loop forever
    for( board=0; board<found; board++ ){
      for( relay=0; relay<8; relay++ ){
        plc_set_relay(list[board],relay,ON);
        delay(333);
        plc_set_relay(list[board],relay,OFF);
      }
      for( relay=0; relay<8; relay++ ){
        plc_set_relay(list[board],relay,ON); // all
      }
      delay(750);
      for( relay=0; relay<8; relay++ ){
        plc_set_relay(list[board],relay,OFF); // all
      }
    }
  }
  delay( int ms ){ // Max delay time = 2375 ms
    unsigned int ival, i, j;
    ival = (int)(ms * 27.30667) + 1;
    for( i=0; i<ival; i++ ) j = j;
  }
}

```

Controllers with Simulated PLCBus

Instructions for BL1000 and BL1100

1. Power up the BL1000 or BL1100 and make sure it is working properly. If you encounter problems, consult the controller's technical reference manual.
2. Disconnect power from the controller.
3. Using the appropriate cable, connect the XP8300 or XP8400 to the PIO port on the controller. See Appendix D, "Simulated PLCBus Connection," for detailed information regarding this cable. With more than one relay board, chain the additional boards to the first one with PLCBus ribbon cables. Make sure all relay boards are positioned with headers facing the same direction.
4. Check header J1 on the relay board(s) for correct jumper setting(s). With only one board, leave J1 unjumped. With more than one board, leave J1 unjumped on the first board and set J1 with a different and unique address on each additional board.
5. Make sure that header J2 has no pins connected. Connect a wall transformer or equivalent 24 V direct current power supply to the V+ and GND terminals on header H3 (when using XP8300) or on header H1 (when using XP8400).
6. Power up the controller and bring up Dynamic C on the host PC. If a problem reestablishing communication occurs, consult *Dynamic C Technical Reference* manual.
7. Open and run the program. See the *Dynamic C Technical Reference* manual for details on opening and running programs.
8. The LEDs on the relay board(s) will begin flashing to indicate that the relays are actuating.

Sample Program for BL1000 and BL1300

The program locates all XP8300 and XP8400 boards attached to the PLCBus. The program then loops, activating the relays on each board. For each board, the program concludes with an all-on/all-off sequence. To locate boards, the program polls all 64 possible addresses. The integer (logical) address of each board that responds is displayed in Dynamic C's **STDIO** window.

```

/*****
                Relay Board Demo - for BL1100
*****/
#define ON 1
#define OFF 0
main(){
    int board,relay,found,list[64];
    Reset_PBus(); // always do this, first thing
    Stall(3000); // pause ~1sec for reset
                // Locate relay boards. Build list
                // and print board IDs

    found=0;
    printf("\nLogical relay addresses found: ");
    for( board=0; board<64; board++ ){
        if( Poll_PBus_Node(Relay_Board_Addr(board)) ){
            list[found++] = board;
            printf(" %d ",board);
            if( found%10 == 0 ) printf("\n");
        }
    } // Activate relays on each board
    // found
    while( 1 ){ // loop forever
        for( board=0; board<found; board++ ){
            for( relay=0; relay<8; relay++ ){
                Set_PBus_Relay(list[board],relay,ON);
                Stall(1000);
                Set_PBus_Relay(list[board],relay,OFF);
            }
            for( relay=0; relay<8; relay++ ){
                Set_PBus_Relay(list[board],relay,ON); // all
            }
            Stall(2000);
            for( relay=0; relay<8; relay++ ){
                Set_PBus_Relay(list[board],relay,OFF); // all
            }
        }
    }
}

```

Blank

Symbols

#use	23, 39
/AT	59
/STBX	59
20-pin connector	80, 81
4-bit bus operations	59, 60
5 × 3 addressing mode	61
8-bit bus operations	59, 61

A

A1X, A2X, A3X	59, 60
actuation voltage	
XP8300	20
XP8400	35
addresses	
encoding	61
logical	
XP8300	22
XP8400	38
modes	61
PLCBus	61
relay boards	22, 38
XP8300	22
XP8400	38
attention line	59

B

background routine	62
bidirectional data lines	59
BL1000	81
software	80
BL1100	80
software	80
BL1300	81
software	81

BL1400	82
software	82
BL1500	82
software	82
board addresses	
XP8300	
jumper settings	20
XP8400	
jumper settings	36
board layout	
SE1100	47
XP8300	15
XP8400	31
BUSADR0	60, 61
BUSADR1	60, 61
BUSADR2	60, 61
BUSADR3	66, 67
BUSRD0	63, 65, 66, 67
BUSRD1	63, 65
BUSWR	65, 84, 86

C

coil voltage	
SE1100	52
XP8300	19
XP8400	35
connecting expansion boards	
SE1100	50, 51
XP8300	18, 19
XP8400	34, 35
connecting nonPLCBus controllers	
+24 V	80
BL1000	80
BL1100	81
BL1300	81
BL1400	82
BL1500	82
cable	80

connectors	
quick-release	73
contact ratings	
SE1100	52
XP8300	19
XP8400	35
control registers	63

D

D0X–D7X	59
daisy chaining	76
digital inputs	
PLCBus	63
dimensions	
SE1100	72
XP8300	70
XP8400	71
DIN rail	14, 30, 46
DIP relays	58

E

Exp-A/D12	58
expansion boards	
installation	
adapter board for BL1400/	
BL1500	82
BL1000	80
BL1100	81
BL1300	81
BL1400	82
BL1500	82
reset	64

F

features	
SE1100	47
XP8300	15
XP8400	31
fuses	
SE1100	52
XP8300	19

H

headers	
SE1100	
J1	52
J3	50
XP8300	
H1	19
H2	19
H3	19
H4	19
XP8400	
H1	35

I

inport	65, 67
inputs	
digital	63
installation	
expansion boards 18, 34, 76, 77	
SE1100 expansion boards	50
interrupts	59, 62
routines	62

J

jumper settings	
XP8300	19
board addresses	20
J1	20
J2	20
XP8400	35
board addresses	36
J2	35

L

LCD	59
LCD bus	59
LCD connections	
PLCBus	59
LEDs	
SE1100	47
XP8300	15
XP8400	31

liquid crystal display. <i>See</i> LCD	
logical addresses	
XP8300	22
XP8400	38

M

memory-mapped I/O register	60
metal oxide varistor (MOV)	15
mode	
addressing	61
mounting	78
end caps	78

N

noise transients	
SE1100	47

O

operating relay boards	24, 40
outport	65, 67

P

P1	76, 80, 81
P2	76, 80, 81
PAL encoding	
XP8300	20
XP8400	36
pinout	
PLCBus	58
SE1100	52
XP8300	19
XP8400	35

PIO port

simulated PLCBus	90
PLCBus ...	58, 59, 60, 61, 62, 63
26-pin connector	
pin assignments	58
4-bit drivers	64
4-bit operations	59, 60, 61
8-bit drivers	67
8-bit operations	59, 61

PLCBus (continued)

addresses	61, 62
connecting cable	81
connector	80
control registers	63
devices	62, 63
expansion register	62
installing boards	76, 80
LCD connections	59
reading data	60
ribbon cables	76
rules for devices	62
writing data	60
Y cable	77

Q

quick-release connectors	73
--------------------------------	----

R

read PLCBus	65
reading data on the PLCBus	60
relay	
control	46
relays	
DIP	58
SE1100	
specifications	52
turning on	84
XP8300	
actuation voltage	20
specifications	19
XP8400	
actuation voltage	35
specifications	35
reset expansion boards	64
ribbon cables	76

S

sample programs

17SE1100.C	54
SE1100	54
XP8300/XP8400	88

SE1100	46, 58	software	
connection		PLCBus (continued)	
sinking driver	50	set24adr	67
sourcing driver	51	set4adr	65
features	47	set8adr	67
select address	64	write12data	66
shadow registers	62	write24data	68
simulated PLCBus		write4data	67
PIO port	90	write8data	68
software		SE1100	54
libraries	23, 39	XP8300	23
DRIVERS.LIB	22–23,	EIO_NODEV	24
38–39, 64, 87		eioErrorCode	24
EZIOCMMN.LIB	23, 39	eioPlcRelayAddr	25
EZIOLGPL.LIB	64	eioPlcRstWait	24
EZIOMGPL.LIB	64	eioResetPlcBus	24
EZIOBDV.LIB ..	24–25, 40–41	plcXP83Out	25
EZIOPL2.LIB	64	VdInit	24
EZIOPLC.LIB	24, 40, 64	XP8300/XP8400	
EZIOPLC2.LIB	23, 39	PBus12_Addr	86
EZIoTGPL.LIB	64	PBus4_Read0	86
PBUS_LG.LIB ..	23, 39, 80–81,	PBus4_Write	86
86, 90		Plc_poll_node	85
PBUS_TG.LIB ..	23, 39, 80–82,	Plc_set_relay	85
86		Plcrel_addr	85
PLC_EXP.LIB	23, 39, 85	Poll_PBus_Node	87
PLCBus	60	Relay_Board_Addr	87
VDRIVER.LIB	24, 40	Reset_PBus	86, 87
PLCBus	64–67	reset_pbus	85, 86
4-bit drivers	64	Reset_PBus_Wait	87
8-bit drivers	67	Set_PBus_Relay	87
eioPlcAdr12	64	set12adr	87
eioRead0	65	write12data	87
eioRead1	66	XP8400	
eioRead2	66	EIO_NODEV	40
eioResetPlcBus	64	eioErrorCode	40
eioWriteWR	66	eioPlcRelayAddr	41
read12data	38, 66	eioPlcRstWait	40
read24data	67	eioResetPlcBus	40
read4data	66	plcXP84Out	41
read8data	67	set12adr	38
set12adr	65	VdInit	40
set16adr	65		

specifications	
SE1100	72
quick-release connectors	74
XP8300	70
XP8400	71

V

V+	
XP8300	20
XP8300/XP8400	90
XP8400	36
VCC	
XP8300	20
XP8300/XP8400	90
XP8400	35

W

writing data on the PLCBus	60, 66
----------------------------	--------

X

XP8100	58
XP8200	58
XP8300	14
features	15
XP8310	
features	15
XP8400	30
features	31
XP8500	58
XP8600	58
XP8700	58, 59
XP8800	58
XP8900	58

Y

Y cables	77
----------------	----

Blank



Z-World

2900 Spafford Street
Davis, California 95616-6800 USA

Telephone: (530) 757-3737
Facsimile: (530) 753-5141
Web Site: <http://www.zworld.com>
E-Mail: zworld@zworld.com

Part No. 019-0054
Revision B